

CMEE Masters: Computing Coursework Assessment

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

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: Calum Pennington

Overall Project workflow

Found all the expected weekly directories in your parent directory.

You had .gitignore files throughout, with meaningful exclusions - good.

You had a readme file, with a list of topics covered in the week - Good. But you could also have added a description of what the overall project structure is.

In each of the weeks, you had detailed readme files, each listing the scripts and data of that week and what they do or are for – good. In the case of Data, only a brief overview of the data files would be fine; after all, the Code files would presumably contain sufficient info about what sorts of data are expected as input. Indeed, a few of the files needn't have been there or documented: FirstExample.blg, FirstExample.bbl, FirstExample.log, as these can be cleaned up after compilation of a latex document. And no need to document the files in Sandbox as they are purely experimental or in Results, as these will presumably be re-generated when scripts in the Code directory are run. As you become a seasoned programmer, you will learn to make the readme file descriptions more compact.

WEEK 1

Found directories Data, Sandbox, Code

Found 11 code files: ConcatenateTwoFiles.sh, CountLines.sh, variables.sh, CompileLaTeX.sh, csvtospace.sh, FirstExample.tex, MyExampleScript.sh, FirstBiblio.bib, UnixPrac1.txt, tabtocsv.sh, boilerplate.sh

Found the following extra files: FirstExample.pdf, FirstExample.blg, FirstExample.bbl

Some of these could be cleaned up after compilation of the latex document by modifying the compilelatex shell script. Also, the pdf could have been redirected to the results directory.

UnixPrac1.txt was fine, and the solutions really well explained. Do compare with the solutions on the master git repository.

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.

Marks for this week: 97

WEEK 2

Found the Code, Sandbox, Data, Results directories

Found regexs.py in parent directory! -.5 pts

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

No extra files; great!

oaks.py had an indentation error. -5 pts

loops.py had an indentation error. -5 pts

Good job with align_seqs_fasta.py. Do look at the solution on the repo as well.

lc1.py, lc2.py, dictionary.py, tuple.py were all fine. For more compact solutions, look at the solution on the repo. Some of them could have given a better formatted output. Again, check out the solutions on the master repository .

You did align_seqs_fasta.py — so 2.5 extra credit points added back on

All other scripts were fine. Compare with the solutions on the master repository.

Marks for this week: 95

WEEK 3

Found directories Practicals, Code, Data, Results

Found 23 code files: browse.R, AutoCorr_Latex.tex, PP_Regress.R, apply1.R, sample.R, run_get_TreeHeight.sh, GPDD_Map.R, boilerplate.R, TreeHeight.R, PP_Lattice.R, next.R, Vectorize1.R, SQLinR.R, break.R, basic_io.R, try.R, apply2.R, get_TreeHeight.R, TAutoCorr.R, Vectorize2.R, PP_Regress_loc.R, DataWrang.R, control.R

PP_Regress.R: was fine. Also have a look at the solution. Lot of overzealous commenting - OK at this stage though!

TreeHeight.R and associated scripts were fine. I like the fact that you used checks and printed feedback to screen. Good!

Vectorize1.R was fine — compare with the solution on the master repository.

Vectorize2.R was fine — again, compare with solution on the master repository

You had a `install.packages("sqldf")` command in SQLinR.R, but that was never going to work on somebody else's computer... -3 pts

TAutoCorr.R was more or less correct (look at the sampling approach in the solution) and again, zealously commented!

The report formatted in \LaTeX from the TAutoCorr.R: Very nice writeup and interpretations. You said that the temperature data did not look normally distributed, but didn't test that. Not the main reason to use a non-parametric correlation - temperature autocorrelation is the main reason

Overall, this week's practical work was very good.

Marks for this week: 96 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 good README

Found 23 code files: TestR.py, regexs.py, LV3.py, timeitme.py, blackbirds.py, Nets-Jan.py, Nets2.py, Nets.py, TestR.R, profileme.py, LV4.py, Nets3.py, fmr.R, using_os.py, LV1.py, DrawFW.py, re4.py, Nets.R, SQLinR.R, run_fmr_R.py, Db_SQLite.py, LV2.py, run_LV.sh

Found no extra files, great!

In LV*.py, you put in on-screen feedback and default values. Very good.

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

using_os.py worked, but compare with the more compact solution on the master repository.

You had a `install.packages("sqldf")` command in SQLinR.R, but that was never going to work on somebody else's computer... -3 pts

You had multiple (versions) Nets*.py scripts, the first of which had an error. 5 points off for that: you could have removed the wrong versions — version control is precisely so that you don't have to manually make copies of your script versions!

You did all the extra credit LV* scripts with profiling: + 7.5 pts. Think about how you could have written the script to run or batch-run these simulations using pure python (not Python + Bash).

Marks for this week: 92 pts

Overall Assessment

You did a very good job overall, including most extra credit Qs. You did a lot of zealous commenting/annotating in all weeks – good. But as you become a seasoned programmer, you will become more conservative with commenting!

However, I was really impressed by your efforts to understand as many details of the programming languages and programming as possible - I noticed that many of the comments were notes to yourself about that.

I found a few errors, especially in the Python weeks – have a look at them again, some of them might seem obvious now that you have been programming longer in Python and R. Overall, you seem to have become quite comfortable with both Python and R – that’s a good sign!

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. You delivered on most fronts, and if this is the first time you have done any kind of programming in UNIX, Python, & R, you did very well!

Provisional Grade

A	86
B	
C	
F	

These marks are based on two sets of marking criteria, both given at the end of the Silwood Biological Computing Notes: “Computing Coursework Assessment Criteria” and the “MARKING CRITERIA for EXAMS and ESSAYS and COURSEWORK”. Each week’s assessment will have lesser marks than what you would have seen originally because I would have taken some points for script errors as well as my assessment of that week’s overall performance.

Signed: Samraat Pawar

March 30, 2017