Stat243: Problem Set 2, Due Friday Sep 18 and Monday Sep. 21

September 11, 2020

Comments:

- This covers material in Units 3 and 4.
- It's due in two parts:
  - Please do Problem 3 in advance of Section on Friday Sep. 18. In particular you should turn in your initial thoughts about problem 3b via the Google form indicated. The remainder of your answers to Problem 3 should be turned in with this problem set.
  - The full problem set is due at 5 pm (Pacific) on September 21, **both submitted as a PDF to Gradescope as well as committed to your Github repository** as documented in the *howtos/submitting-electronically.txt* file.
- Please note my comments in the syllabus about when to ask for help and about working together. In particular, **please give the names of any other students that you worked with on the problem set and indicate in the text or in code comments any ideas or code you borrowed from another student.**

Formatting requirements

1. Your electronic solution should be in the form of an R markdown file named *ps2.Rmd* or a LaTeX+knitr file named *ps2.Rnw* or *ps2.Rtex*, with bash and R code chunks included in the file (or read in from a separate code file).
2. Your PDF submission should be the PDF produced from your Rmd/Rtex/Rnw. Your Github submission should include the Rtex/Rmd/Rnw file, any code files containing chunks that you read into your Rtex/Rmd file, and the final PDF, all named according to the guidelines in *howtos/submitting-electronically.txt*.
3. Note that using chunks of bash code in Rmd/Rtex/Rnw can sometimes be troublesome, particularly on Windows machine. Some things to try if you are having trouble: (a) set the terminal to be the Ubuntu subsystem if you are on Windows; (b) if using RStudio, you may only be able to run bash chunks if you have the notebook mode on; (c) using single versus double quotes in your Rmd/Rtex/Rnw document may make a difference. If you can't produce a PDF that includes the bash chunk output, feel free to not run those chunks and just paste in the output you get manually. Please post on Piazza if you have trouble - the suggestions above are, unfortunately, only based on some vague memories of mine of what students' experience was last year.

4. You will probably need to use *sed* in a basic way as we have used it so far in class and in the tutorial on bash. You should not need to use more advanced functionality nor should you need to use *awk*, but you may if you want to.

5. Your solution should not just be code - you should have text describing how you approached the problem and what the various steps were.

6. Your code should have comments indicating what each function or block of code does, and for any lines of code or code constructs that may be hard to understand, a comment indicating what that code does. You do not need to show exhaustive output but in general you should show short examples of what your code does to demonstrate its functionality.

Problems

1. Add assertions and testing for your code from Problem 4 of PS1. You may use a modified version of your PS1 solution, perhaps because you found errors in what you did.

   (a) Add formal assertions using the *assertthat* package. You should try to catch the various incorrect inputs a user could provide and anything else that could go wrong (e.g., what happens if one is not online?).

   (b) Use the *testthat* package to set up a small but thoughtful set of tests of your functions. In deciding on your tests, try to think about tricky cases that might cause problems.

2. A friend of mine is planning to get married in Death Valley National Park in March (this happened last year before the pandemic...). She wants to hold it as late in March as possible but without having a high chance of a very hot day. This problem will automate the task of generating information about what day to hold the wedding on using data from https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/. All of your operations should be done using the bash shell except part (c). Also, ALL of your work should be done using shell commands that you save in your solution file. So you can't say "I downloaded the data from such-and-such website" or "I unzipped the file"; you need to give us the bash code that we could run to repeat what you did. This is partly for practice in writing shell code and partly to enforce the idea that your work should be reproducible and documented.

   (a) Download yearly climate data for a set of years of interest into a new (temporary) subdirectory within your working directory. Do not download all the years and feel free to focus on a small number of years to reduce the amount of data you need to download. Note that data for Death Valley is only present in the last few decades. As you are processing the files, report the number of observations in each year by printing the information to the screen, including if there are no observations for that year.

   (b) Subset to the station corresponding to Death Valley, to TMAX (maximum daily temperature), and to March, and put all the data into a single file. In subsetting to Death Valley, get the information programmatically from the *ghcnd-stations.txt* file one level up in the website. Do NOT type in the station ID code when you retrieve the Death Valley data from the yearly files.

   (c) Create an R chunk that takes as input your single file from (b) and makes a single plot of side-by-side boxplots containing the maximum daily temperature on each day in March.

   (d) Now generalize your code from parts (a) and (b). Write a shell function that takes as arguments a string for identifying the location, the weather variable of interest, and the time period (i.e., the years of interest and the month of interest), and returns the results. Your function should

detect if the user provides the wrong number of arguments or a string that doesn't allow one to identify a single weather station and return a useful error message. It should also give useful help information if the user invokes the function as: "get_weather -h". Finally the function should remove the raw downloaded data files.
Hint: to check for equality in an if statement, you generally need syntax like: `if [ "${var}" == "7" ]`.

3. On Friday, September 18, Section will consist of a discussion of good practices in reproducible research and computing. In preparation, please do the following:

   (a) Read **one** of these five items (you can read more if you want of course!):
      i. Read Chapter 11 of Christensen et al. Transparent and Reproducible Social Science Research (chapter 11 is here in bCourses) (You can also see the entire book online via Oskicat: https://california.degruyter.com/view/title/568658?tab_body=toc) - this one is written from the perspective of social scientists.
      ii. Gentzkow and Shapiro (http://www.brown.edu/Research/Shapiro/pdfs/CodeAndData.pdf) – also written from the perspective of social scientists.
      iii. Wilson et.al. (http://arxiv.org/pdf/1210.0530v3.pdf)
      iv. Millman and Perez (https://github.com/berkeley-stat243/stat243-fall-2014/blob/master/section/millman-perez.pdf)
      v. Read the Preface, the Basic Reproducible Workflow Template chapter and Lessons Learned chapters of the new-ish (Berkeley-produced) book. You can see the book online via Oskicat: The Practice of Reproducible Research.

      When reading, please think about the following questions:
      i. Are there practices suggested that seem particularly compelling to you? What about ones that don't seem compelling to you?
      ii. Do you currently use any of the practices described? Which ones, and why? Which ones do you not use and why (apart from just not being aware of them)?
      iii. Why don't researchers consistently utilize these principles/tools? Which ones might be the most/least used? Which ones might be the easiest/most difficult to implement?
      iv. What principles and practices described apply more to analyses of data, and which apply more to software engineering? Which principles and practices apply to both?

      As your answer to this problem, please write a paragraph or two where you discuss one or more of these questions. I'm not looking for more than 10-15 sentences, just evidence that you've read one of the items and considered it thoughtfully.

   (b) (You will have time to work on this during class time Wednesday September 16.) Please skim through the paper *ps/clm.pdf*, focusing on the Method section, but note that the idea is just to get the main idea of the analysis steps, not to understand the context fully or see all the details. Then look at the code the authors provide at https://github.com/andykrause/hhLocation and think about whether that code makes it easy to reproduce what they did in the paper. Based on the Unit 4 PDF and the item you read above in (a), make a short list of strengths and weaknesses of the reproducibility of the authors' materials and turn in via this Google form before noon on Friday Sep. 18. Your task in Section on Friday September 18 will be to discuss with your group and come up with a consensus answer that you should turn in as your answer to this problem. This could be a bulleted list of strengths and weaknesses or a few paragraphs of text. Consistent with the instructions, make sure to note in your answer who your group members were.
      Some things to think about when you look at their materials:

i. From the information above and the documentation provided, can you quickly identify where in the code (if present at all) the authors:
   A. Collected their data
   B. Cleaned/processed their data
   C. Calculated various statistics
   D. Produced the plots in their paper
ii. In terms of coding what elements of their project do you like? Consider: Documentation, comments, organization, naming, workflow, data provenance, etc.
iii. Similarly, what do you think could be improved.
iv. Without examining the code itself, can you quickly discern the purpose of each file?
v. Without examining the code itself, can you quickly tell what each block of code does?
vi. Are there exceptions to your answers above?
vii. In what way do the authors document their workflow? Do you think this method is effective?