Problem Set 2

Due Friday Sep. 16, 10 am

Comments

- This covers material in Units 3 and 4.
- It's due at 10 am (Pacific) on September 16, both submitted as a PDF to Gradescope as well as committed to your GitHub repository.
- Please see PS1 for formatting and attribution requirements.
- 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:
  1. set the terminal to be the Ubuntu subsystem if you are on Windows;
  2. if using RStudio, you may only be able to run bash chunks if you have the notebook mode on;
  3. using single versus double quotes in your Rmd/Rtex 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 the Ed forum if you have trouble.
- You will probably need to use `sed` in a basic way as we have used it so far in class and in the bash tutorial. You should not need to use more advanced functionality nor should you need to use `awk`, but you may if you want to.

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 or wanted to make changes based on Chris' solutions or your discussions with other students.

   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 problem is based on real events…). 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 the Global Historical Climatology Network. 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.

1. Download yearly climate data for a set of years of interest into a temporary 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.

2. 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.

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

4. 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 (or you should download into your operating system's temporary file location). Hint: to check for equality in an if statement, you generally need syntax like: `if [ "${var}" == "7" ]`.

3. On Friday, September 16, 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!):

        1. The Preface, the Basic Reproducible Workflow Template chapter and Lessons Learned chapters of this Berkeley-produced book on reproducible research, found online via UC Library search: The Practice of Reproducible Research. The book is written from the perspective of learning about reproducibility practices using case studies
        2. This article is written from the perspective of scientific research: Millman and Perez.
        3. This article is written from the perspective of scientific software development: Wilson et.al.
        4. This article is written from the perspective of social science research: Chapter 11 of Christensen et al. Transparent and Reproducible Social Science Research. (You can also see the entire book online via UC Library search
        5. This article is also written from the perspective of social science research: Gentzkow and Shapiro.

When reading, please think about the following questions:

1. Are there practices suggested that seem particularly compelling to you? What about ones that don't seem compelling to you?
2. 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)?
3. 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?
4. 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. Please skim through this economics paper, 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 and think about whether that code makes it easy to reproduce what they did in the paper. Based on the Unit 4 class notes and the item you read above in (a), make a short list of strengths and weaknesses of the reproducibility of the authors' materials and provide that as your solution to this problem. Your task in Section on Friday, September 16 will be to discuss with your group and in Section as a whole and come up with a consensus answer. Some things to think about when you look at their materials:

   1. From the information above and the documentation provided, can you quickly identify where in the code (if present at all) the authors:
      1. collected their data,
      2. cleaned/processed their data,
      3. calculated various statistics, and
      4. produced the plots in their paper.
   2. In terms of coding what elements of their project do you like? Consider: Documentation, comments, organization, naming, workflow, data provenance, etc.
   3. Similarly, what do you think could be improved?
   4. Without examining the code itself, can you quickly discern the purpose of each file?
   5. Without examining the code in detail, can you quickly tell what each block of code does?
   6. Are there exceptions to your answers above?
   7. In what way do the authors document their workflow? Do you think this method is effective?