CS 510 Qualifying Exam

About this exam

This is the qualifying exam for CS 510 Computing for Scientists at Chapman University. This exam and its components are copyrighted to Lindsay Waldrop, Ph.D. and may not be reproduced publicly without written permission. (You are prohibited from posting this material on a homework help site such as Course Hero or Chegg.) The exam can be used by students in future CS 510 classes for study purposes.

Exam Agreement

This exam is an individual assessment and you should not receive or offer help on it from any other human. However, you may use any resource, either online or physical, to complete the work. This includes:

- Any help forum or website (e.g. StackOverflow) questions that already exist. (You MAY NOT ask a question on a forum and then receive specific help from a person.)
- Any notes, code, slides, papers, or previous feedback from the course and instructor.
- Any books, online or physical.
- Scholarly works such as papers.

It DOES NOT include:

- Help from any other student or person. This is an individual assessment.
- Asking for help on specific questions.
- Help from homework websites such as Course Hero or Chegg.

By cloning this repository, you are agreeing to the above terms.

Instructions for this exam

To begin the exam, accept the Github classroom assignment through the link provided during the exam period and follow the instructions to create your own repository using the provided code template.

This exam consists of two portions: essay questions and a coding practical.

1. **Essay Questions**: in the file labeled “Essay-questions.pdf”. Provide answers to these questions in the time period provided during the exam. Be sure to provide in-text citations and references to scholarly works that support your conclusions.
2. **Coding Practical**: in the folder labeled “practical”. This folder contains a coding project in R that completes a basic task. However, it has little documentation and runs slowly. Your tasks are to:

- improve the documentation and structure of the code so that it successfully reproduces, similar to the rules in the course (these are repeated below).
- refactor the code so that it is more reader friendly.
- optimize the code for memory usage and speed.
- add additional functionality to the code (noted in the practical/README.md file).

Do the best you can in the time you have. You should use git to add commits to your directory as you work, I will review this history so do NOT neglect to commit your work and document the changes as you go! I will be assessing the code in the same way as the final projects, using this form: https://forms.gle/UiCwmAJca16uMwED9, a copy of which is included in the main directory of this repository. (Note: You do not fill out this form, I do! But it is here for your reference.)

To submit your work:

- To submit your essay question answers, turn in a digital copy of your answers by the end of the exam period by emailing them to waldrop@chapman.edu. (This should be a stand-alone file in rft, doc, docx, or pdf. NO LINKS TO GOOGLE DOCS.) The essay questions are due by the end of the exam period at 6:30 pm PST on Aug 27 2021.
- To submit your code practical, turn in the repository via Github Classroom assignment. The coding practical is due 11:59 pm PST on Aug 29 2021.

BE ADVISED: No late work will be accepted, so please have your work be submitted on time.

Rules for Evaluating Reproducibility

Procedure for peer-reviewing code:

1. Download the release specified. Do not use another release or the most recent version on Github.
2. Open the zip file and enter the directory.
3. Read and follow the instructions in the README.
4. If no instructions are provided, either:
   - open RStudio project and source first R script.
   - setwd() and source first R script.