Assignment 5 – CS 4536/536: Programming Language Design
Instructor: Rose Bohrer (pronouns: she/her)
Instructor Office: Fuller Labs 139

Overview

This assignment is about interpreting data from user studies, with an emphasis on Likert data.

How Long Will This Take?

My goal is that most students will spend at most 13 hours/week on this course and that few will spend more than 15 hours/week, in total.

If you have spent more than eight hours on this assignment and are not done, you are **expected** to reach out to the course staff for help (office hours, chat, or email). It is our job to help you and your job to seek help when appropriate.

Grading

This assignment is worth 100 points. 75 points are for the programming part and 25 points for the written part. Your grade for the programming part is the sum of scores for each passing test case. Your grade for the written part is 25 * the fraction of problems for which you submit an honest attempt at a solution.

What to Read While Working on This Assignment

In addition to the lecture readings, you will likely find the official documentation for Rust helpful. The homework handout lists specific research papers which you are expected to reference while answering the problems. You are welcome to refer to other resources about data analysis.

Programming Assignment

Download and extract the programming assignment from Canvas. Your assignment is to implement all functions to meet their specifications as given in the comments. Point breakdown:

- 15pts agreement_to_score
- 15pts index_to_prompt
- 15pts index_to_key
- 15pts acquiescence_bias
- 15pts score

The main learning objective of the programming assignment is to understand how to analyze the data resulting from a Likert scale, including scoring the data and computing their acquiescence bias. It is intended to be less time-consuming than the previous assignments, in order to leave time for the written assignment, exam review, and any revisions students might make to previous assignments per the course resubmission policy.

Written Assignment

25 pts Data analysis

Interpret the data collected from your study to answer the research question of the study. The methods will depend on which style of study you did. Only undertake the method(s) relevant to your study:

- Survey/Quantitative Questions: Use the mean or median function to interpret your data, depending on what type of data you have. You are welcome to also compute any other statistics that are relevant to your problem, but not required.
- Interview+Activity/Qualitative Questions: Read the Qualitative Studies chapter of the HCPL textbook to familiarize yourself with the technique of ***Thematic Analysis***. Perform a grounded theory analysis of your results. *An exhaustive analysis can be very time-consuming, so it is sufficient to analyze data from 3 people to receive credit.* Take the written responses or transcripts, as appropriate, and perform the 6 stages of thematic analysis. You are only required to include the themes and their definitions (and also, write the report). If you show additional work, however, we can give more detailed feedback.
- Regardless which method you use, clearly restate your original research question and clearly mark the main answer to that question which you've arrived at

Use the results of the analysis to briefly (e.g. <= 1 paragraph) provide and justify your answer to your initial research question.

Optional bonus: 10 pts Creative Coding

This is a programming problem, but is listed under the written problems because it is ungraded. Write a creative program to express yourself, using any of the creative technologies mentioned in the textbook:

- Processing
- Twine
- Inform
- Penrose
- Context-Free Art

If your art is visual, include a picture in your submission, otherwise include the source code and submit it with your written answer to the previous question as a .zip file. Please explicitly indicate whether you are okay with the instructor sharing your work with the other students in class for fun.

This activity is intended to be open-ended and fun. If you're having trouble coming up with an idea of your own, you can do either of the following to receive credit:

- Look up a tutorial for one of these technologies, complete it, and hand in the completed work
- Download an existing piece of art that someone has published in one of these tools, **cite that piece of work in your submission,** and modify it / experiment with it.

Optional: 0 points, but will help you with the final

Recall that all the content from all lectures is fair game for the final exam, though I most likely will not cover every single lecture. Closer to the exam, I will publish a list of potential exam questions. Out of respect for time, I do not assign full-on homework questions for each topic, but many students would like feedback on their work before the exam. So, here are some optional ungraded problems. If you choose to complete these, it does not contribute directly to your grade, but I will give you written feedback on them, to help you prepare for the final exam, well in advance. Of course, you are welcome to pick just a few of these questions that you want feedback on and submit those.

Design:

Perform a thematic analysis of this blog post about a newcomer's experience with Rust, written in 2015. https://internals.rust-lang.org/t/newcomer-to-rust-my-experience/1816

Show your work, i.e., codes and initial list of themes as well as final themes. Skip step 6 (writing the report).

Natural Language:

Name the five dimensions of the SPACE model and give an example of something you would measure when measuring each dimension.

Demonstrate the difference between verbosity and complexity by providing two examples: the first example should be something verbose but not complex, and the second should be complex but not verbose. The examples could be programming languages, PL features, or example programs.

Gender:

Name the 5 facets of the GenderMag methodology. For each one, describe it in one sentence and give two very brief examples of programming language design choices or programmer experiences that are relevant to that facet.

Disability

1. Suppose you have a repetitive strain injury (such as Carpal Tunnel Syndrome) and wish to use a specialized keyboard layout for programming. However, your boss only lets you use a standard keyboard layout. Suppose you have to choose between two programming languages X and Y, where X uses lots of punctuation and Y uses lots of keywords. Based on an ergonomic perspective on syntax, which language will aggravate your injury more, and why?
2. Using an indentation-sensitive language of your choice (such as Python or, if you know it, Haskell) write two programs which have different meanings but would be pronounced the same using a traditional screen reader.
3. Based on discussions in the course (e.g. lecture and book), what are reasons that a designer might consider hybrid visual-textual syntax? Select all that apply
   a. Social integration across the disability spectrum
   b. Increasing the accessibility of an existing visual language for visually disabled programmers
   c. Increased runtime efficiency, e.g., decreased memory usage
   d. Increased ability to rule out errors statically at compile-time, e.g., with a static type system

Diagrams / Visual Design:

1. In one sentence, describe the difference between "Programming Language Design" and "Visual Design"
2. The languages CSS and Penrose have different audiences. List one design decision in CSS and one design decision in Penrose where the audience mattered, e.g. the Penrose design decision should be a better fit for the Penrose audience than the CS audience.

Peer Reviews

There is no peer-review component in this assignment.