Linda Mei

Project 1 Work Accomplished

In the “homework” histogram, I made a histogram for 1.3 since it was the first problem. Making the histogram was not difficult because I followed the steps to make the relative frequency histogram. Afterward, I compared the class example histogram to mine. Since my histogram looked pretty accurate to the example histogram, it was complete. Next, the ongoing coding assignment we had throughout the first half of the semester was the stats library. I coded solvers for variance, standard deviation, permutations, combinations, mean, median, mode, and binomial distribution and tested them against problems in the textbook to ensure they worked. Throughout most of my programs, I tried to make one method run the whole program instead of having multiple lines of code in the main method. Later, I will talk about updating the stats library.

The Monty Hall program was a bit unintuitive for me at first. The most important part for me was understanding how the game worked. After, I brainstormed in text and then coded. At some point, I had to go back and redo my code since I was not getting the probabilities that I was expecting to get from the results. The hardest part for me was applying the logic of how the player picks one random door, which could be a dud or the prize, and Monty Hall always has to show the dud prize. Keeping track of those variables in my program took me quite a while, but when the program worked, it was satisfying.

The Birthday program was more confusing than the Monty Hall program for me. It was crucial to understand how the birthday paradox worked and think of how to go about coding it before jumping in. I created a bunch of people with a random month and day for a birthday and added them to an ArrayList. Then, I used two for loops to check when all possible pairs of birthdays (with no duplicates) were equal. If the birthdays were the same, I would increment the count of how many times there was a matching birthday. I had the simulation test the number of successful trials divided by the total number of trials (successful trials were trials with at least one matching birthday). I also coded the probability of the matching birthdays solely based on the class size to compare with the probability from my simulations based on the user’s input of the class size and number of runs.

The FishMarket program was less confusing than the other two because the instructions were clear. The only parts I did not know how to was exporting and importing the .csv files. I did have to search for how to do those since I would not have been able to figure it out off the top of my head. I included the citations in the comment at the top of the FishMarket.java file (also included in my graph program piggybacking off my FishMarket program). For the histograms in steps five and six, I followed a similar process as the homework histograms and ensured that the histograms reflected the data. The weights for step six were somewhat confusing until I realized which part of my code was changing the random number generation of the seafood to make it unfair. From there, I adjusted the maximum value of the random number to be the user’s input. There is a mini report for the FishMarket program explaining the four graphs.

The last two things are the graph program: plotter, salter, and smoother and updating the stats library. I took what I heard in class about the windows and left and right points and tried to figure out what I could code to make it work. For the graph program, I brainstormed some ideas first. Then, I slowly translated the text into code. It was a lot of trial, error, and experimentation since this was my first time working with graphs and .csv files. Even though it took a while, I was happy I got it to work. Next, I updated the stats library by adding the geometric distribution, hypergeometric distribution, Poisson distribution, conditional probability, and Bayes’ theorem. I thought it was weird to work BigInteger and BigDecimal because I had to use many methods, and the code was longer. I did get it to work with a book problem, but it did take a while. I also included my set operations code (intersection, union, complement) from the set operations homework (assigned in a class) even though it is not part of Project 1. I thought it made the most sense to include everything we worked on this semester, so I kept it. Lastly, I updated the stats formula sheet for all the formulas in class and stopped at the end of chapter 4.3 in the textbook. The formula sheet includes everything up to the midterm.