Matthew Roesch

Probability And Applied Stats: Project 1

Overall, these programs were a great way to get be back into the swing of things with coding. At the beginning of the semester, I felt as if my skills were a bit rusty since I hadn’t really coded during the summer, and the semester before that I didn’t have much coding practice on this scale. Honestly this felt a bit like I was back in my high school java class, only using much more advanced concepts. I say that not to make it seem like this was very easy to do, but to make a comparison on how the classes felt. I had an incredible teacher during high school who really made coding easy and fun to learn and perform. We would start by learning what each new thing would do, then we would create a fun project that utilizes it. I’m glad this class seems to follow a similar pattern, as I think it is very beneficial for learning, at least to me.

When I started coding the statistics formulas, I originally had most of them be in their own individual classes, with their own testers. While I did like this as it made it easy to keep track of what I had already done, it was a sloppy way to code it. Luckily, because of the way I had made the classes meant that I wouldn’t had to do too much work to change them from classes to methods. For the most part, it was just changing inputs in the constructor to be parameters for the methods. The biggest problem was having to change formulas that use combinations. Previously, I just created a combination object and called a method to return the combination. So, when reworking everything, I had to change any formula that used combinations to no longer create an object, and instead just call the method in the stats library class. This wasn’t a hard thing to do, just tedious. But from now on I’ll be thinking of whether or not specific uses require being its own class, or just a method.

Now the specific projects are what I was both most excited about and most annoyed by. Starting off with the door game really felt like I was back in high school java. These were the types of projects that we would create, and it just kind of felt like home. I didn’t have any problems with this one, and it all went smoothly. Same to be said about the birthday tester. It was pretty simple to calculate the probability in a given size, just needed the amount of runs and people in the groups.

The fish market simulation is where things got tough for me. It had been a while since I worked with parent and child classes, so I needed a slight brush up on them, but that was easy and over with in just a few minutes. Creating the different types of seafood and even the market was fairly simple and straightforward. My problem was with outputting the data to a csv file. I believe I had only worked with them ever once before, about three years ago at nine in the morning during a zoom class. So that information definitely didn’t stick, and it hadn’t really come up since, so I had essentially had no experience with readers or writers. And surprisingly I couldn’t find any easy and comprehensive guides on how they work, so this was a big roadblock. I spent a lot of time trying to solve errors and problems with the code and with excel, which was quite frustrating. Eventually I figured it out, and with a bit of help from you, was able to fix the main error causing it to not output, which of course was that I just missed a close statement. It might not be the best implementation of a writer, but as it is essentially my first and it works, I am sufficiently happy with it.

After figuring out the readers and writers the programs involving graphs became not only possible, but much easier. The plotter class is fairly straightforward, just write the output of the formula, and the salter class just adds or subtracts a random amount from a given datapoint. The smoother class did give me a bit of a struggle though. I had to think about how to store the data of the nearby points, and originally, I planned to just read the file over again until I got to the desired value, then save it somewhere to be used for averaging later. But that seemed pretty inefficient, so I decided to read the file once, and save all the values into two array list, one holding the x values and one holding the y values. Then when averaging each point, I could just reference back to the list, without needing to read the file again and again.

All in all, I’m quite happy with how the project turned out. My stats library class feels quite robust, and I like the way it turned out. The side projects were my favorite though. Some were definitely challenging, and they introduced me to some new concepts which I feel are extremely useful in programming.