**Project 1 report and summary**

This project provided me with opportunities to both practice completing tasks that (to me) were already mostly routine, and to start experimenting with some potentially useful techniques that for whatever reason I had never found much use for.

The programming parts of the project were immediate, straightforward, and borderline trivial. These involved direct translation of formulas and probability calculations into Java code, while being careful to use the proper data types to obtain correct results (e.g. BigDecimal instead of double and BigInteger instead of int, etc.). The only programming task that was slightly more involved was the salting and smoothing of data, and the graphical results were intriguing. The programming aspect in the FishMarket program was also straightforward, but the exporting of the data as a csv file was something I had never done, and which I found somewhat interesting. I had never used Excel before being exposed to it in this class, and while it certainly is an annoying and frustrating software package at times, I could see how it can be very useful in various disciplines and lines of work. I did get a healthy amount of exposure, and practice with, creating graphs, histograms, and working with data sets.

The Monty Hall problem was one that I had already seen previously, although writing the program and tweaking the number of doors, as well as the number of iterations, allows for much better understanding of what is really happening. In particular, as the number of doors increases, the player benefits increasingly from switching from the door that was originally chosen. Incidentally, when introducing the Monty Hall to people for the first time, a useful and convincing technique would be to have them imagine the game being run with 1,000,000 doors. Since the host knows where the prize is hiding, the host will eliminate (miraculously!) 999,998 of the remaining doors.

The birthday problem was also one that I had encountered before, but in addition to solving it exactly and mathematically, I also solved it by writing a different, OOP-style program and simulating the problem.