

Nick Fichera

Formal Report

Project 2

To begin, I had an interesting experience working on this project. Initially we were told to find a dataset and create questions from this. I discovered my dataset that I used from Google's dataset search engine. The dataset I found details records of airplane crashes around the world from 1908 to 2009. I found these results interesting, despite being a little grim, but I figured it was something that could work for my project. Additionally, I decided to create some java programs for my stats library. One of these programs calculates the probability an event lands in a uniform distribution. The other program allows users to create PTable objects, which is essentially a matrix but has features such as finding the marginal probability for the discrete case.

Next, I had to create a graphing program using JFreeCharts and Apache Commons. Honestly, this was the first time that I really had to go out and find libraries and attach them to the build path for my project (I use eclipse). This taught me an important skill that I could carry with me. Initially, I took some time to try to figure it out and eventually got there (details are in the JFreeChart report).

Moving forward, we then had to use Octave. Initially, I learned about functions and how to create them. The initial graphing was pretty simple. The salt program is structured kind of how I thought it would be, just a simple for loop that negates the salt value each iteration, which is very similar to how I implemented this in the first project. Smoothing is where I struggled a little bit. I couldn't figure out how to access the salted values from the salting function, so I just copy pasted the for loop from the salting project to my smoothing project. (I understand that this isn't the most efficient). I then wrote a for loop after to inspect the left and right values of salted values, average them and adding them to the graph. Theoretically, I believed that the code I wrote should produce a smooth function, however the result I was expecting did not exist (see octave report).

Next, we had to do a write up on the other distributions of Chapter 4. The firstly included the Normal distribution or the Gaussian distribution. This represents a bell curve, something that I have previously known about but is cool to see in this book. A distribution function is supplied which allows us to integrate to find the probability of an event under the bell curve. Next is the gamma distribution. This is the same as a bell curve except it is skewed to the right as y gradually decreases. Lastly, we have the beta distribution, which deals with skewed bell curves on each side, but more specifically, proportions.

Lastly, we had to implement a Poker project. Initially, I created the Card class where objects created from this can represent any card from a 52-card playing deck. I then created a Deck class where objects created from this represent a deck with 52 Card objects. This class

calculates the Monte Carlo, as well having methods to check if a given hand (an array of 5 distinct cards from Deck) has any winnings by using the evaluateHand method. The Draw class has a method to determine a winner between two hands, as well as simulating a one and two player game.

Overall, I enjoyed working on this project because it opened my mind up to new ideas and helped me reinforce my programming skills.