Nick Fichera

*Probability and Applied Statistics: Project 1*

To start off, the first thing that had to be done was the relative frequency graph of the data. This was simple to me and the only thing that didn’t require programming as we just took data from the text and put it into excel to create a relative frequency histogram.

In addition, I begin implementing some of the basic functions for my stats library. Initially, this included mean, median, mode, standard deviation and variance, something we did towards the beginning of the semester. Being that it was the first assignment, it was not that difficult. The only one I struggled a little bit with was the median a little bit with the mode. The main problem with finding the median was represented by the case where an even number of values were present. This involved finding the left and right values that were right in the middle. This was something that took me a little bit to figure out, but I got it down.

Moving forward, we moved to set operations and a review of sets. This involved writing a program to find the union, intersection and complement of two sets, or of two subsets compared to one universal set. The only thing I somewhat struggled with is finding the complement of sets.

Next, we moved onto tools for counting sample points. This majorly involved two formulas: permutations and combinations. For this, I created a program to determine permutations and combinations. Inside the program existed a method to find the factorial of a number, since there was no java method in any libraries to do so (that I could find at least). Permutations and combinations utilize the factorial method in their own methods.

Continuing, conditional probability was also implemented. It includes a java program with two methods. The first method takes the probability of A intersect B, and the probability of B and divides them and returns the result. The second method takes the probability of A and the probability of B and multiplies them and returns the result (if A and B were independent).

Progressing, we run into the two laws of probability. Here I have four methods: multiplicative law, multiplicative law if independent, additive law, and additive law if independent. These were straightforward to implement. We also saw the law of total probability which is also implemented.

Now, we move onto our distributions. Before that, let’s start with the expected value and variance of a random discrete variable. This method takes an int array of Y values and a double array of their probabilities. It then multiples each Y by its probability, sums it, and returns it. Moving along, the first distribution being the binomial distribution. The parameters take in the number of runs, what our Y variable should be and the probability. It calls upon the combinations method that was defined in my permutations and combinations solver. Pretty straight forward. Now, we move onto the geometric and hypergeometric distributions. In the geometric distribution Y and probability are taken and the formula is used along with the Math.pow function. The hypergeometric series takes in the values it needs, does the formula, and then uses the combinations formula I implement, like the binomial distribution. Also, we have the Poisson distribution. This method computed the outcome by utilizing the Math.pow function for the numerator and my factorial solver for the denominator. Lastly, we have Tchebysheff's theorem. For this I had the mean, standard deviation, and value 1 and 2 which represented the bounds. I used the Math.abs function to determine the within number. The second method will find the value of C, like the example we did in class.

Finally, we run into the other programs we had to make. This included the birthday, door game and the Seafood programs. These were somewhat interesting to implement, and it was kind of nice to be able to program stuff that isn’t just implementing formulas from the book. All in all, I thought this project was interesting.