Patrick Niederhauser
Project 1 report

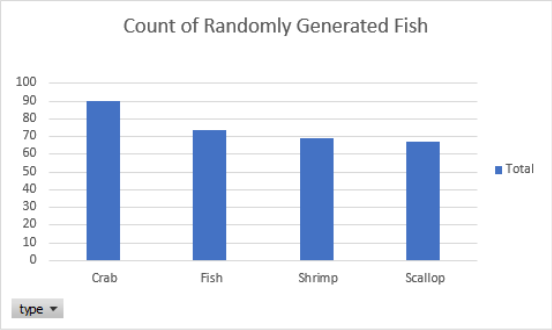
This project had multiple assignments that helped me build my coding skill tremendously in a short period of time. This project had 5 assignments that had varying difficulty levels, but all used different techniques and skills to help build our programming tool box. I appreciate the challenge that this project presented and I am thankful for having the opportunity to work on it.

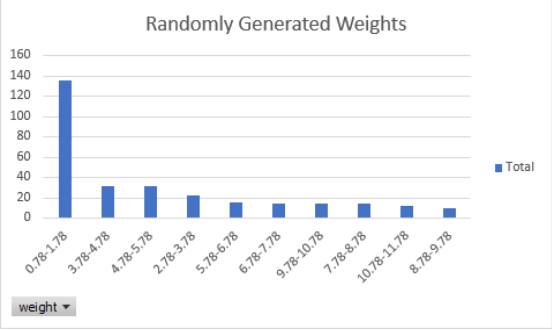
The first project I worked on was my monty hall simulation. This program focused on the statistical anomaly that is known as the monty hall theorem. This problem asked for us to prove the monty hall theorem using statistics. To simulate this problem I had to create two different methods. One where the user did not switch doors and the other where they did switch doors. Comparing these two answers would either prove or disprove the monty hall theorem. The first step on coding this project was to create a method in which the user did not switch doors. This was done by creating a random number generator and a for loop. I created two variables, one which contained the correct door and one which contained the first guess. I then compared these two variables and if they equaled one another I added 1 to our count. To calculate the win percentage I then did total win over total trials. To simulate switch doors I did the same thing but ran our for loop again and simulated the change in answer. If the changed answer was equal to our original door variable I then added 1 to our count. Our results proved the Monty Hall theorem to be true, returning around 60% winning percentage when switching doors and a 33% win percentage when not switching doors. This answer was a bit surprising to me because I figured the answer would be about the same, but it does make sense that switching doors gives you a higher chance. When you first pick a door you have a ⅓ chance to guess correctly and when you are switching doors you now have a ½ chance to pick the correct door, because you eliminated one of the choices.

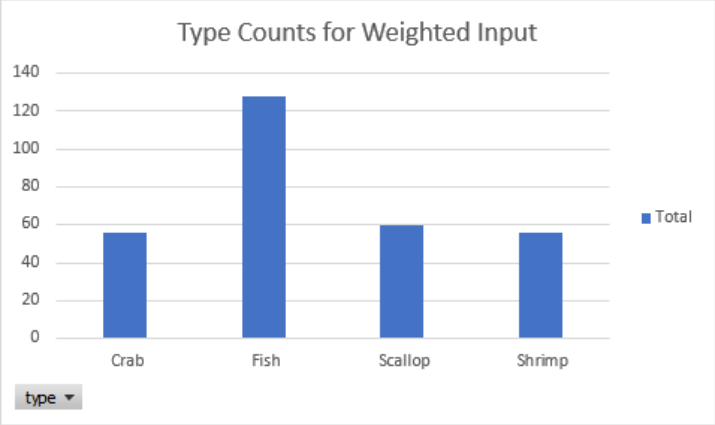
The next project I worked on was the birthday paradox problem. This paradox indicates that if there was a class room of x number of students what is the probability that you would share a birthday with someone else in the room. To start this problem I first had to create an object person that contains an integer of a random month and day (mm/dd). This was done by creating a random number from 1-12 for the month and then creating a random day based on the appropriate month. For example February would pick a random day between 1-28 days because February only has 28 days. I then created a new method which would create an object of type person inside of a for loop that would determine if two people would have the same birthday. This for loop took user input to determine how many times it would run the simulation and how many people were in the class. My results were a little bit surprising to me. I would have guessed that the odds of two people in a 24 person class sharing two birthdays would be very low, but it's actually almost 50%. I had to check my answer multiple times after I was done coding this project, because I thought my answer was unrealistic, but I guess that's why it's called a paradox.

The next project I worked on was the fish program. I already talked about the first program in my fish market report, so I will keep this short. This first market program created a unique fish object using inheritance. The subclass of the super classes were responsible for creating unique weights and prices for each creature that I added to my fish market. I then ran this simulation 300 times and graphed the results. I then added user imputed “weights” to the

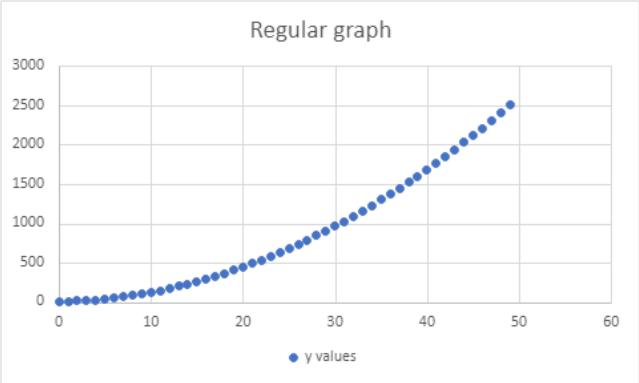
types and graphed the results again. I'm going to include my graph below so you can get an understanding of my answers. This project was definitely the most fun I had on a project thus far, I'm an active fisherman so doing anything with fish excites me. My answers were not too shocking considering a lot of it was straightforward, however visualizing my data was a very cool experience.

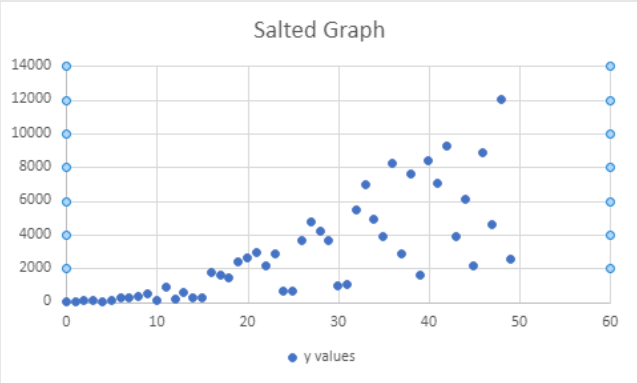


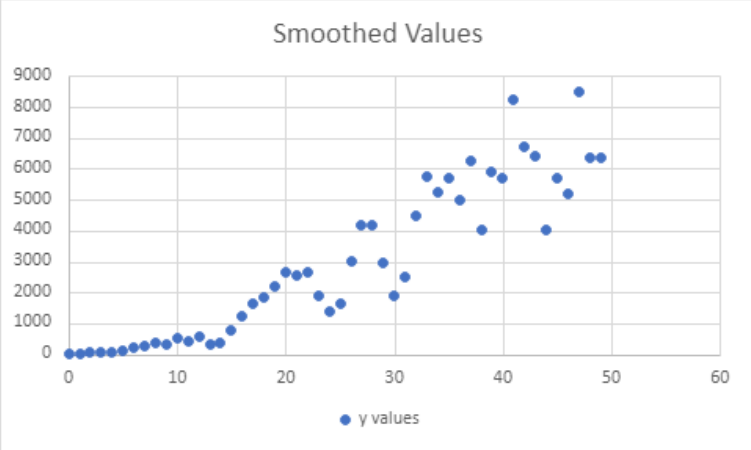




The next project I had to work on was the salter and smother graph. This project wanted me to create random graph data using a random formula, so I decided to use y=mx+b. I first created a graph method to add an x value and y value to my two array list. I then took user input to determine how many points I would graph, I then took that data and printed it on to a csv file. Now that we had our regular graph file, I wanted to salt our data. To do this I created a method called salter that took our graph csv file and read the data in line by line. I then salted the data and saved it to a new array list. I then called a new method that took our salted data and printed it to our csv file. Now that we have our salted data I wanted to smooth it out. I read in our salted data line by line under a new method called smoothed. I then found the average of three points and added them to our new array list. I then printed that array list using the same process described above. I then opened excel and graphed all three csv files. I included my graphs below. My answer wasn't too surprising, but it was interesting to see my data being graphed.







The last project I added was my stats library, this library included a collection of every stats formula we used thus far. I created some random number generator numbers to help me test some of my answers. This library helped me understand a lot of the stats that we learned throughout the class. This library is pretty straightforward and I programmed all my problems so that I can just plug homework variables in and the program would solve them for me. This stats library was very helpful and helped me understand a lot of our theories and problems that we discussed in class.