**Probability and Applied Stats**

**Plotter, Salter, Smoother #3 External Libraries**

For the final portion of our project, we had to create three more programs that do the same thing as the previous parts of this project. First, a plotter to create a chart for our favorite Pythagorean theorem formula. Then, a salter to take that data and completely discombobulate it. Finally, a smoother that attempts to repair the damages done by the salter. Given that this is now our third time creating programs that essentially do the same thing, I was thrilled when I realized how much simpler it would be using external libraries. Using JFreeChart and Apache Commons Math I was able to finish this portion of the project in record time. These libraries largely enhance one's ability to create charts and execute math commands in java. To begin, Apache Commons Math was especially helpful by allowing me to use FastMath which is designed to execute mathematics efficiently. It involves methods optimized for speed, certain operations that are implemented with higher precision, and functions not available in javas typical Math class. In my program it is used to calculate the hypotenuse from the fixed side and variable side. Another benefit is that Apache Commons Math is able to handle larger datasets more efficiently which confirms consistent output across all iterations. Then, JFreeChart generates a line chart of the Pythagorean theorem for me. I can change the labels by adjusting my code which makes it very simple. It makes it very simple to add and adjust the axis and legend, also allowing the user to adjust the color and style of the chart. I use JFreeChart in the salting and smoothing programs as well, creating charts that show the original data compared with the salted and smoothed. The charts created are eye pleasing and they get the message across; however, I would say that I prefer the charts generated by MATLAB. It was actually easier to generate the charts using JFreeChart and Apache Math Commons because java is the main programming language that I use. Since MATLAB was completely new to me this project, I struggled a little when it came to writing my code. On the other hand, as mentioned before I do prefer the charts MATLAB produces to the ones produced from JFreeChart. This is mainly due to the clean look that the MATLAB charts were presented with. I found it much easier to read the data from the MATLAB charts. This is also because of the tools that MATLAB provides for you when viewing their charts. One of the most valuable tools being the data tips tool which shows you the exact coordinates of where you left click. This accompanied with little things such as the ability to drag and pull around to wherever you wanted to view specifically on the chart give MATLAB a slight edge for me. Additionally, I usually use Visual Studio Code when programming, however I was having difficulty importing the external libraries into this IDE. I was able to import them after switching over to Eclipse where I was met with a more straightforward way to configure my project build path. After completing all three parts of this project I feel that I have gained fundamental knowledge that I will keep with me and put to use in the future. I am grateful because I feel that while MATLAB may be a little more niche and math related it is still a great skill to have in my arsenal. However, I am really looking forward to finding more external libraries I can import for various programs and seeing what different kinds of projects I am able to cook up.