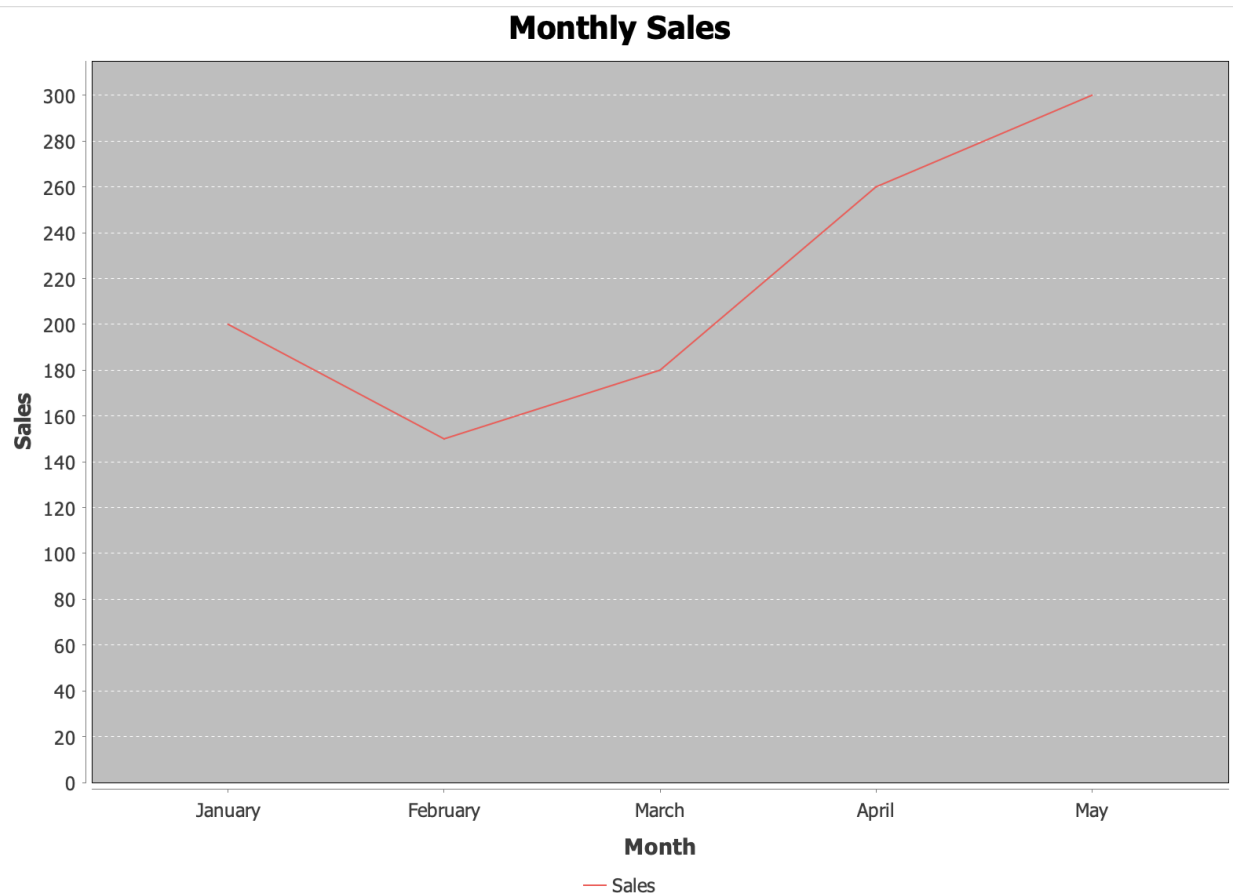


JFreeCharts and Apache

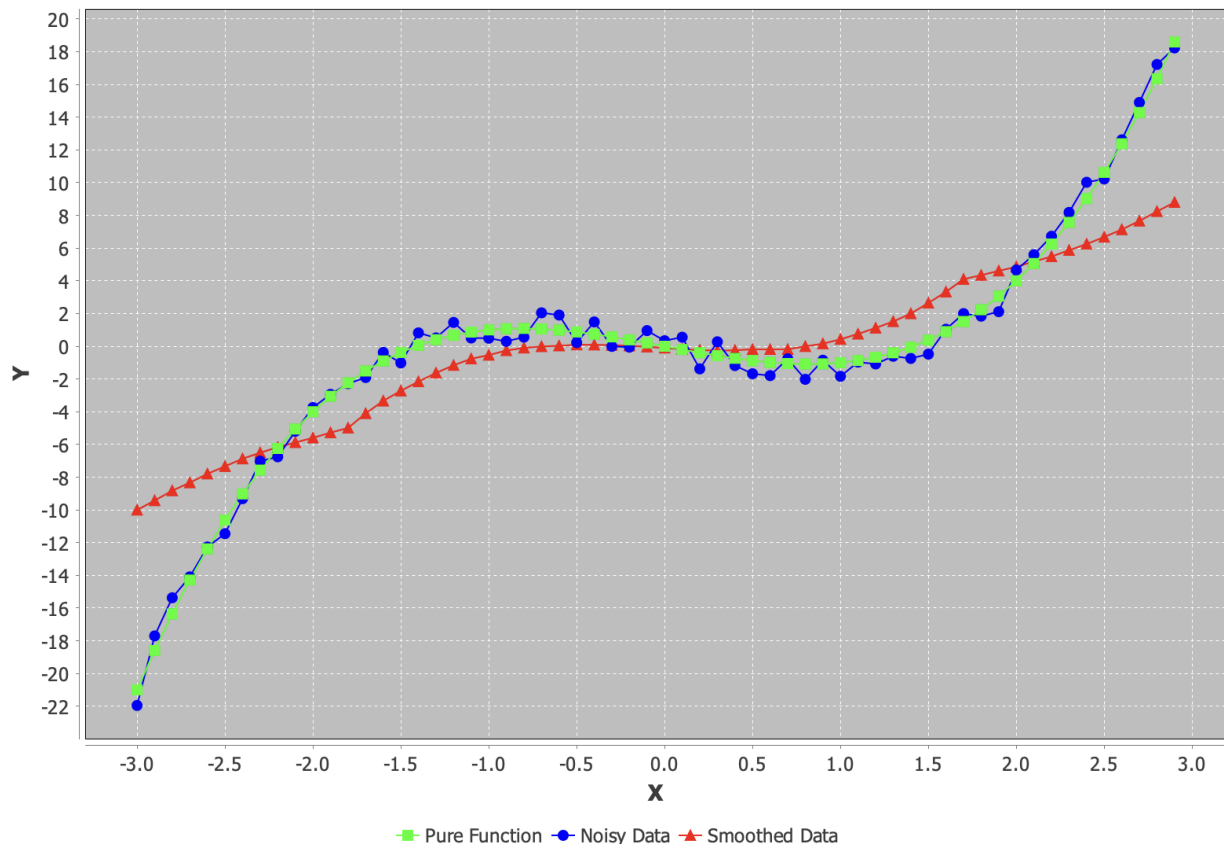
For this project, I used the external Java libraries JFreeChart and Apache Commons Math to work with the function $f(x)=x^3-2x$. The goal was to simulate a basic data processing pipeline: generate the function values, add noise (called “salting”), smooth the noisy data, and plot everything for comparison. I first created the original dataset by evaluating the function from -10 to 10 in steps of 0.1. To mimic noise, I added random values to each data point. This gave me the salted version of the curve.

The first graphic was what I generated after following a tutorial to better understand JFreeCharts. I used the tutorial on this url <https://www.baeldung.com/jfreechart-visualize-data> to help learn the basics of JFreeCharts.

Next, I used Apache Commons Math’s DescriptiveStatistics class to apply a moving average and smooth out the noise. The window size was fixed, and the smoothed result gave a cleaner look that followed the original shape more closely. I plotted all three versions—original, salted, and smoothed—on the same graph using JFreeChart. The original curve showed the expected cubic behavior, the salted one was noisy and jagged, and the smoothed curve cleaned it up and brought it closer to the original. This was a straightforward way to show how JAR libraries can be used in Java to handle basic data manipulation and visualization.



Data Smoothing Visualization



In addition to these, I used JFreeCharts extensively in my project to generate visuals. Most notably, I used it in my implementation of hash maps to help generate charts that visualized the efficiency and viability of my hashing function. JFreeCharts is significantly easier to use than what we did in part one of this project. Generating a csv file and having separate classes to smooth, salt, and plot data is very cumbersome and JFreeCharts made this all much simpler to complete. I wish I had it when doing the original portion of this project.