**Part 1 - PSS 3**

**Ian Valiante**

**Introduction**

This part of the project focuses on creating and analyzing three types of data—original, salted, and smoothed—using JFreeChart for visualizations and Apache Commons Math for statistical analysis. To tell the story of how data evolves, we’ve created four graphs: one for the original sine wave, one showing the noisy salted data, one with the smoothed version of that data, and a combined graph that puts everything together for easy comparison. The goal here is to show how data can go from clean and simple to messy and noisy, and then back to something much more meaningful.

**Implementation**

The program starts with the generation of a sine wave for the original series. This creates a predictable, smooth pattern that acts as our baseline. To make things more interesting (and realistic), Gaussian noise is added to each point of the sine wave, creating the salted series. This noise simulates the kind of randomness you might encounter in real-world data. Finally, to clean things up, a rolling mean is applied to the salted data. This smoothing technique uses a sliding window to calculate an average for the most recent points, effectively reducing noise while preserving the overall trend.

All three series—the original sine wave, the noisy salted data, and the smoothed series—are plotted together on a single chart. Each series is represented with a different line style so you can easily tell them apart. The chart itself is built using JFreeChart and displayed in a Java Swing application, making it simple to view and interact with.

**Results**

**Original Data**

We start with a simple sine wave. It’s smooth, predictable, and the perfect baseline to use as a reference. This sine wave serves as the “pure” version of the data and sets the stage for what’s coming next. It’s a great way to understand the impact of adding noise and applying smoothing techniques.

**A graph with red line

Description automatically generated**

**Salted Data**

To make things more interesting—and realistic—we add a bit of chaos in the form of Gaussian noise. This step creates the salted data. Now, instead of a clean wave, we have something that looks much more unpredictable, with random ups and downs. This noise simulates the kind of imperfections you’d encounter in real-world data, like measurement errors or environmental factors. While the pattern of the sine wave is still there, it’s hidden beneath the noise.

**A graph with a red line

Description automatically generated**

**Smoothed Data**

Next, we clean things up by applying a rolling mean to the salted data. This smoothing technique takes the average of recent data points within a sliding window, which helps to reduce the noise while keeping the general shape of the sine wave intact. The smoothed data brings us back to something that resembles the original wave but with just enough detail to reflect the natural variability. It’s a great example of how you can make messy data easier to interpret without completely losing the story it’s telling.

A graph with a red line

Description automatically generated

**Combined Graph**

Finally, we put everything together in a single graph. The original sine wave, the noisy salted data, and the cleaned-up smoothed data are all displayed side by side, each with its own unique line style. Seeing them together makes it easy to understand the transformations quickly. You can clearly see how the noise distorts the original signal and how smoothing brings it closer to its clean, original form. This graph tells the complete story in one place and really highlights the power of these techniques.

A graph with different colored lines

Description automatically generated

**Conclusion**

Overall, this project shows how powerful visualization, and statistical tools can be when it comes to making sense of data. JFreeChart makes it easy to turn numbers into clear, compelling visuals, while Apache Commons Math gives us the tools to clean and analyze that data. The four graphs bring everything together, showing how data can be transformed from something messy and imperfect into something much more meaningful and easier to understand.