**Exploring Data Visualization and Manipulation with JFreeChart in Java**

**Yash Maisuria**

**Introduction:**

In the realm of data analysis and visualization, programming plays a vital role in extracting insights and presenting data in an accessible format. This essay delves into a Java code snippet that demonstrates data reading, visualization, and manipulation techniques using the JFreeChart library. The code showcases the use of JFreeChart to create plots, apply data transformations, and explore statistical properties of a dataset. Let's explore the code in detail.

**Code Overview:**

The code begins by importing necessary classes and libraries, including JFreeChart for charting and the Percentile class from Apache Commons Math for smoothing calculations. It defines a class named "JfreePlotter" that encapsulates various methods for data reading, plotting, and manipulation.

**Reading Data:**

The "readData" method takes a filename as input and returns a List<Double> containing the data read from the file. It uses a BufferedReader to read each line of the file, splits the line using a comma as a delimiter, and converts each token into a Double value. The parsed values are added to the data list, which is eventually returned.

**Plotting Data:**

The "plotData" method generates a line plot based on the provided data and saves it as an image file. It starts by creating an XYSeries object to hold the data points. It iterates through the data and adds each data point to the series with the corresponding x-coordinate. The XYSeries is then added to an XYSeriesCollection, which serves as the dataset for the chart. Using ChartFactory, a JFreeChart object is created with the specified title, axes labels, dataset, and orientation. Additional customization includes setting a white background, black line color, and disabling various features. Finally, the chart is saved as a PNG image file.

**Salting Data:**

The "saltData" method takes the original data, applies a salt effect to it, and calls the "plotData" method to generate a plot of the salted data. It creates a new list, "saltedData," and iterates through each value in the original data. It multiplies each value by a random factor within a range of ±5% to introduce variability. The resulting salted data is then passed to the "plotData" method for visualization.

**Smoothing Data:**

The "smoothData" method smoothes the data by applying a moving median filter with a window size of 5. It creates a new list, "smoothedData," and iterates through each data point. For each point, it extracts the surrounding values within a window of size 5 (if available) and calculates the median using the Percentile class from Apache Commons Math. The resulting smoothed data is then passed to the "plotData" method for visualization.

**Conclusion:**

This code snippet showcases the use of JFreeChart library in Java to read, visualize, and manipulate data. By leveraging JFreeChart's capabilities, the code demonstrates the creation of line plots, the introduction of variability through data salting, and the application of a moving median filter for data smoothing. These techniques provide insights into the patterns, trends, and statistical properties of the dataset. Through the power of programming and libraries like JFreeChart, Java enables the exploration and understanding of data, facilitating data-driven decision-making and analysis in various domains.