Code Documentation for OctavePSS Project

By Rachel Hussmann

Overview

The purpose of the OctavePSS project was to learn a new programming language and to create a Plotter, Salter, and Smoother program, just like we did in Java. This project creates data points for a log function, salts the y values by adding or subtracting random integers on a set bound, and smooths the data by taking a window of values and averaging them. This project has three classes and one script file.

How It Works

LogFunction

The LogFunction class contains methods that calculate the answer to a mathematical function and save the x and y values. This class has two variables:

- x – A vector of x values associated with the function
- y – A vector of y values associated with the function

This class has three methods:

- LogFunction
  - Parameters: int startingValue – The starting x value (inclusive), int finishingValue – The last x value (inclusive)
  - Functionality – Constructor for the LogFunction class, calculates the value of ln(x) and saves the x and y values in vectors
  - Returns: ln – The newly created LogFunction object
- get_x
  - Parameters: ln – The LogFunction object
  - Functionality – Returns the x values for the function
  - Returns: x – The vector holding the x values for the function
- get_y
  - Parameters: ln – The LogFunction object
  - Functionality – Returns the y values for the function
  - Returns: y – The vector holding the y values for the function

Salter

The Salter class contains a method that salts (adds a random value to) the data. The class contains one method:

- addSalt
    - Parameters: salt – The Salter object, yValues – The y values from the function that need to be salted, bound – The upper and lower bound of the random number generator (-bound to bound)
    - Functionality – Adds a random value to the y values to hide the original values
    - Returns: saltedValues – The salted y values

Smoother

The Smoother class contains a method that smooths out salted data and makes it more understandable. The class contains one method:

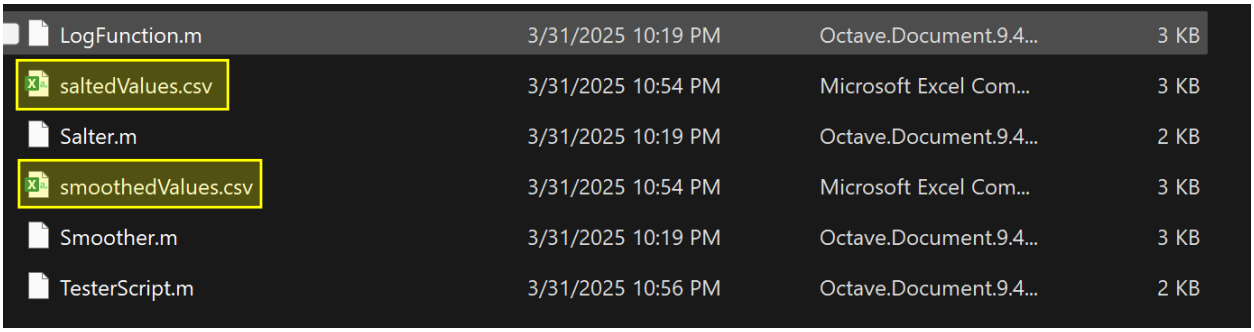
- smoothData
    - Parameter: smooth – The Smoother object, yValues – The salted y values, windowValue – The number of data points to be averaged to smooth the data, runs – The number of times the smoothData method should be run
    - Functionality – Takes the salted values and finds the average of a set number of values to smooth the data
    - Returns: smoothY – The smoothed y values

TesterScript

The TesterScript is the script file that runs the methods, much like a main method in Java. It tests the methods from the LogFunction, Salter, and Smoother classes. This script also saves the x and y values of the salted and smoothed function into a .csv file.

Output

The .csv files for the smoothed and salted values



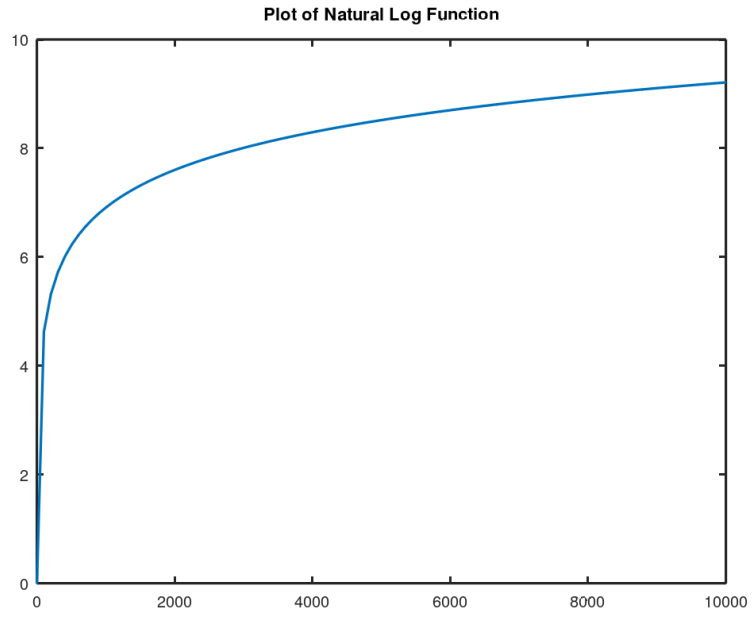
Some of the values from saltedValues.csv

1	102	203	304	405	506	607	708
186	352.625	-477.687	66.71703	-143.996	-359.773	266.4085	-12.4376

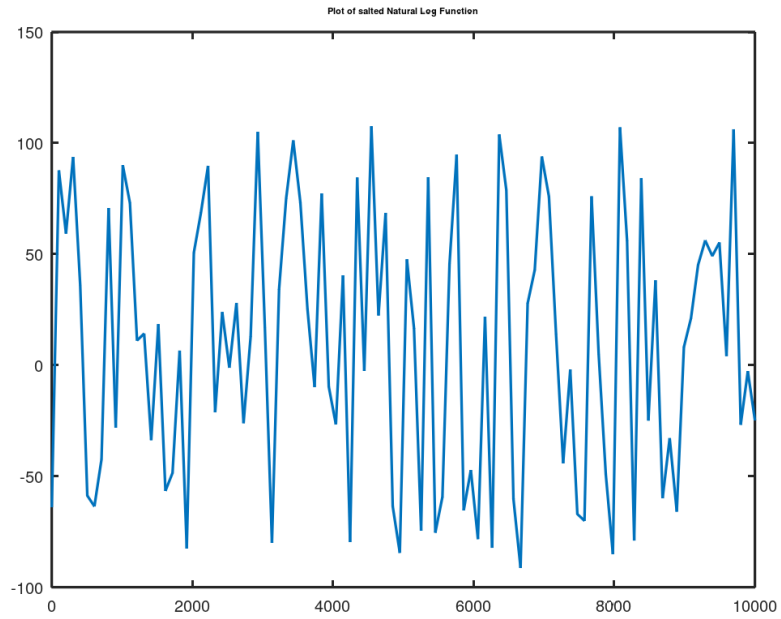
Some of the values from smoothedValues.csv

1	102	203	304	405	506	607	708	809	910
7.958568	8.203397	8.459909	8.732054	9.016596	9.308464	9.606058	9.908058	10.21527	10.53061

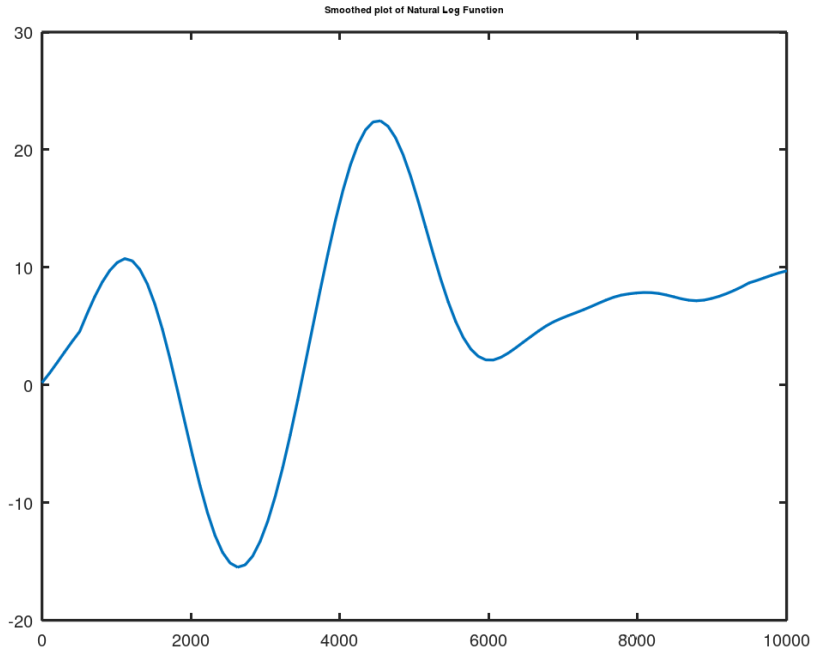
Graph of the Natural Log Function



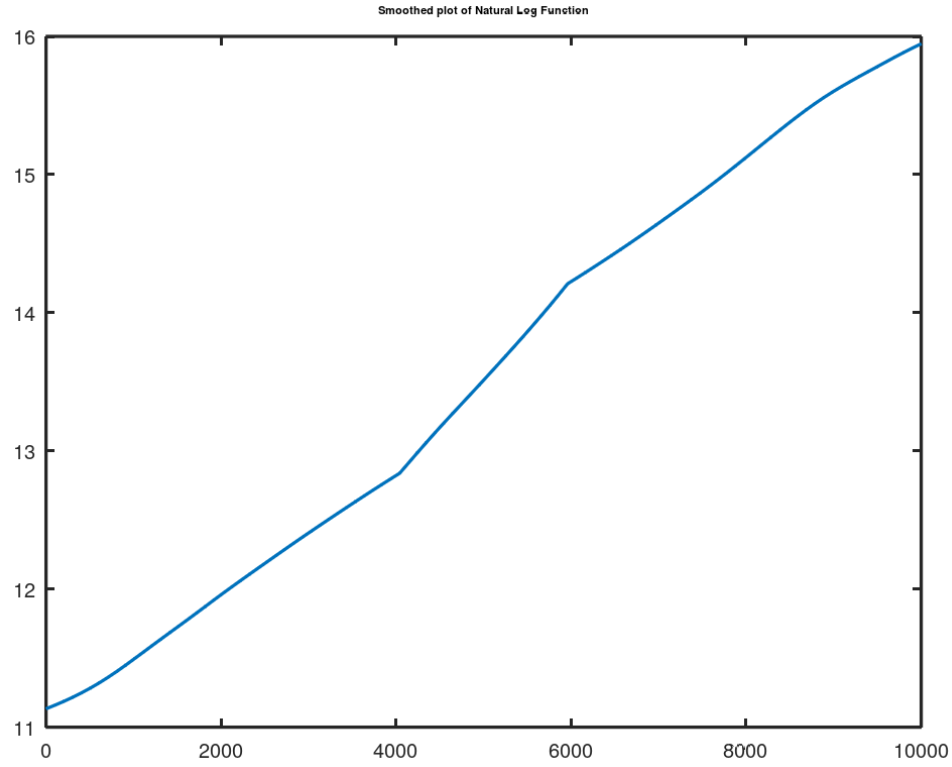
Graph of the Salted Log Function with bound from -100 to 100



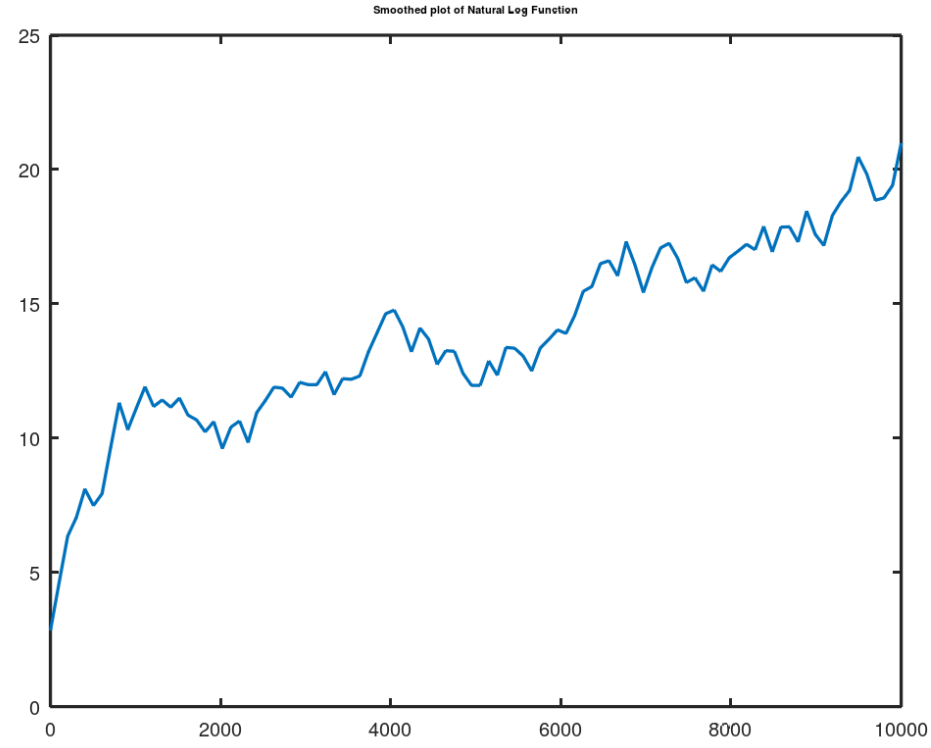
Graph of the Smoothed Log Function (bound -100 to 100) with windowValue = 5 and runs = 4



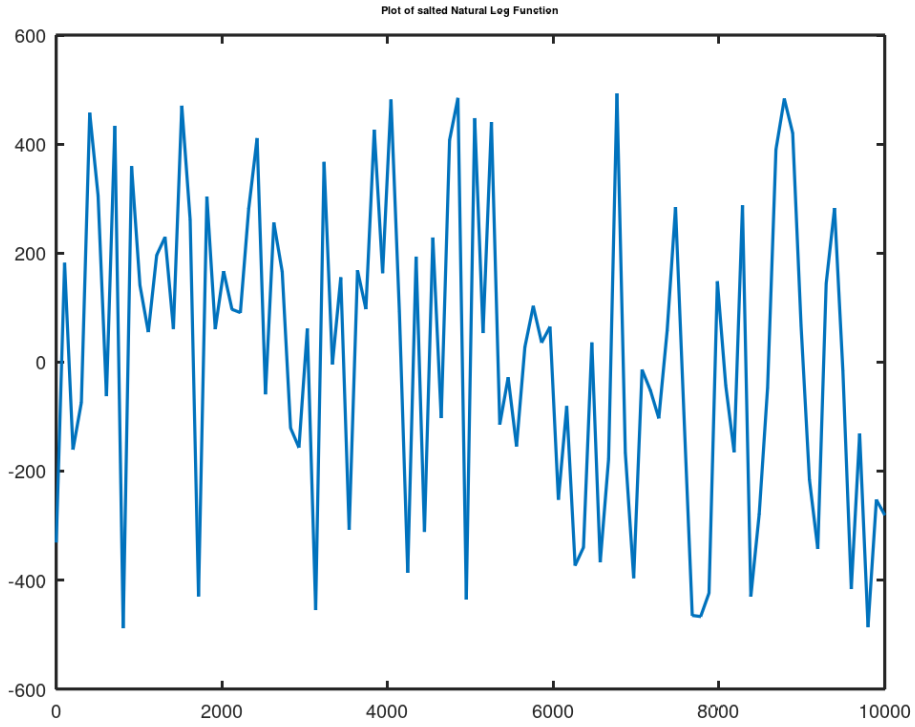
Graph of Smoothed Log Function (bound -100 to 100) with windowValue = 40 and runs = 3



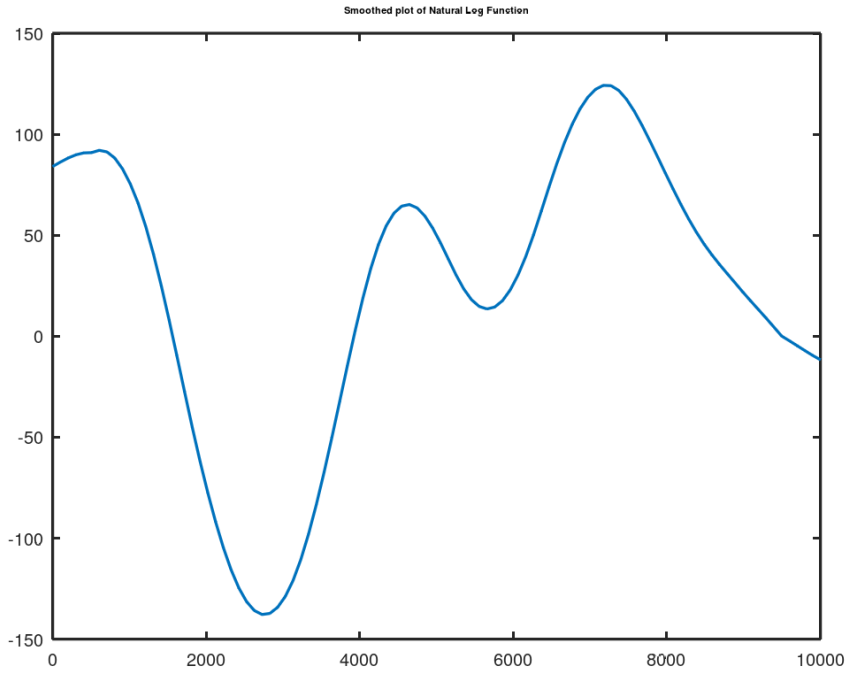
Graph of Smoothed Log Function (bound -100 to 100) with windowValue = 50 and runs = 1



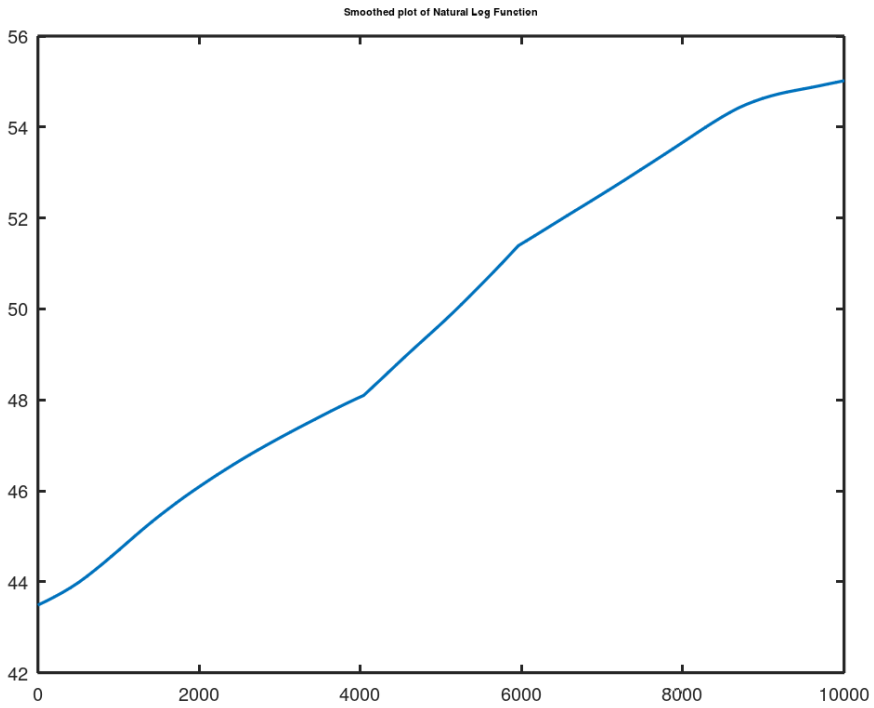
Graph of the Salted Log Function with bound from -500 to 500



Graph of the Smoothed Log Function (bound -500 to 500) with windowValue = 5 and runs = 4



Graph of Smoothed Log Function (bound -500 to 500) with windowValue = 40 and runs = 3



Graph of Smoothed Log Function (bound -500 to 500) with windowValue = 50 and runs = 1