Code Documentation for PSS Project

By Rachel Hussmann

Overview

The purpose of the PSS project was to create classes and methods that calculate a function, salt the y values of the function, smooth the salted y values, and save the x and y values to a csv file. This project has six classes.

How It Works

Exporter

The Exporter class is responsible for saving the x and y values of the calculated function. It has one method:

- createFile
    - Parameters: ArrayList<Integer> xValues – The ArrayList of x values that were used with the function to create the y values, ArrayList<Double> yValues – The ArrayList of y values that were calculated using the function, String nameOfFile – The desired name for the file
    - Functionality – Saves the x and y values to a .csv file
    - Returns: Nothing, but prints statements

Function

The Function class contains methods that calculates the answer to a mathematical function and saves the x and y values. This class has three methods:

- logFunction
    - Parameters: int startingValue – The starting x value (inclusive), int finishingValue – The last x value (inclusive), int origin – The origin of the random function (inclusive), int bound – The upper bound of the random function (exclusive)
    - Functionality – Calculates the value of ln(x) and saves the x and y values in ArrayLists
    - Returns: Nothing
- saveFunction

    - Parameters: ArrayList<Integer> xValues – The ArrayList of x values from the function, ArrayList<Integer> yValues – The ArrayList of y values calculated using the function and the x values
    - Functionality – Creates an Exporter object and saves the x and y values from the function into a .csv file
    - Returns: Nothing
- saveSaltedFunction
    - Parameters: ArrayList<Integer> xValues – The ArrayList of x values from the function, ArrayList<Integer> yValues – The ArrayList of y values calculated using the function and the x values, int origin – The origin of the random function (inclusive), int bound – The upper bound of the random function (exclusive)
    - Functionality – Creates an Exporter object, salts the data using a Salter object, and saves the x and salted y values into a .csv file
    - Returns: Nothing

Importer

The Importer class contains methods that accept a file and extract the data from the file. The class contains two methods:

- importFile
    - Parameter: String filepath – The filepath of the file that needs to be imported
    - Functionality – Imports the file and places each line of data in an ArrayList of strings. Sends the data to trimAndProcess and returns its result.
    - Returns: ArrayList<Double> - An ArrayList of doubles that holds the y value from the file
- trimAndProcess
    - Parameter: ArrayList<String> data – The ArrayList of strings that need to be processed
    - Functionality – Trims the extra information off of the imported file and separates the values into x and y values
    - Returns: ArrayList<Double> - An ArrayList of doubles that holds the y values

Salter

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

- addSalt

    - Parameters: ArrayList<Double> yValues – The ArrayList of y values calculated using the function and the x values, int origin – The origin of the random function (inclusive), int bound – The upper bound of the random function (exclusive)
    - Functionality – Adds a random value to the y values to hide the original values
    - Returns: ArrayList<Double> - The yValues that have been salted

Smoother

The Smoother class contains methods that smooth out salted data and make it more understandable. The class has two constructors, the default constructor and an additional constructor that accepts an ArrayList<Double> of the salted y values. The class contains two variables:

- ArrayList<Integer> xValues
- ArrayList<Double> yValues

The class also contains three methods:

- smoothData
    - Parameter: int windowValue – The number of data points to be averaged to smooth the data
    - Functionality – Takes the salted values and finds the average of a set number of values to smooth the data
    - Returns: Nothing
- saveSmoothedFunction
    - Parameters: None
    - Functionality – Saves the x values and the smoothed y values into a .csv file
    - Returns: Nothing
- runSmoother
    - Parameters: int windowValue – The number of values to the left and to the right of each value to find the average of, int numberOfRuns – The number of times to run the smoothData method
    - Functionality – Runs the smoothData method the number of times the user would like and saves the results of the smoothing
    - Returns: Nothing

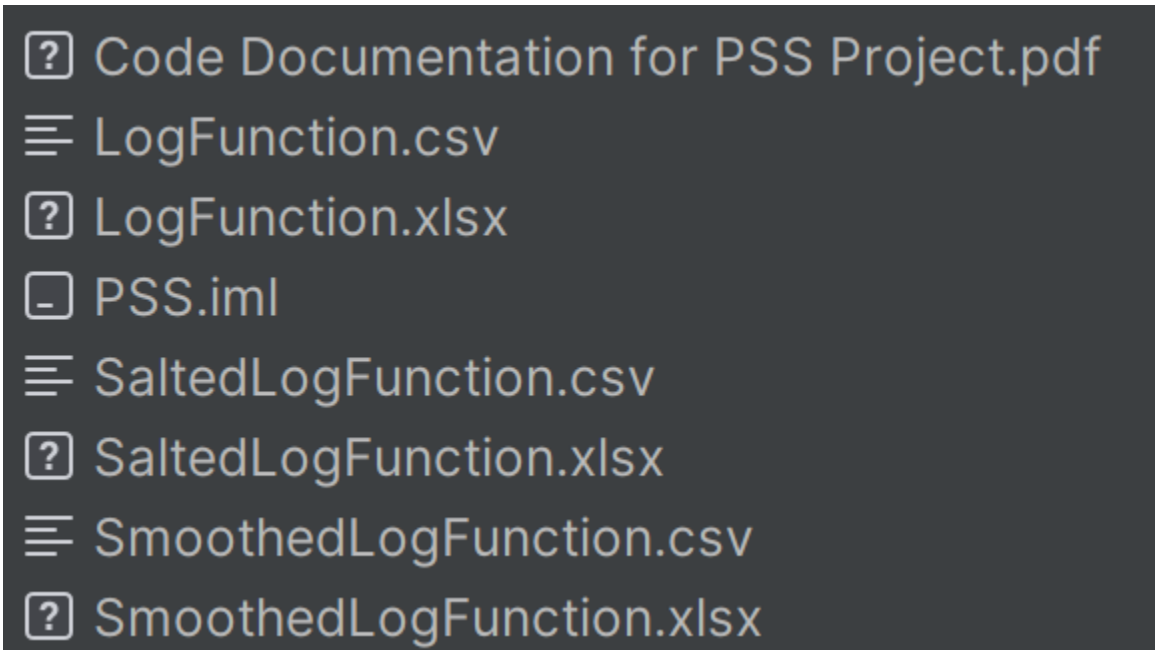
PSSTester

The PSSTester class contains the main method and is used to test the methods from the Exporter, Function, Importer, Salter, and Smoother classes.

Output

Screenshots

Screenshot of the files in the PSS directory



Contents of the LogFunction.csv

```
X,Y,
1,0.0000,
2,0.6931,
3,1.0986,
4,1.3863,
5,1.6094,
6,1.7918,
7,1.9459,
8,2.0794,
9,2.1972,
10,2.3026,
11,2.3979,
12,2.4849,
```

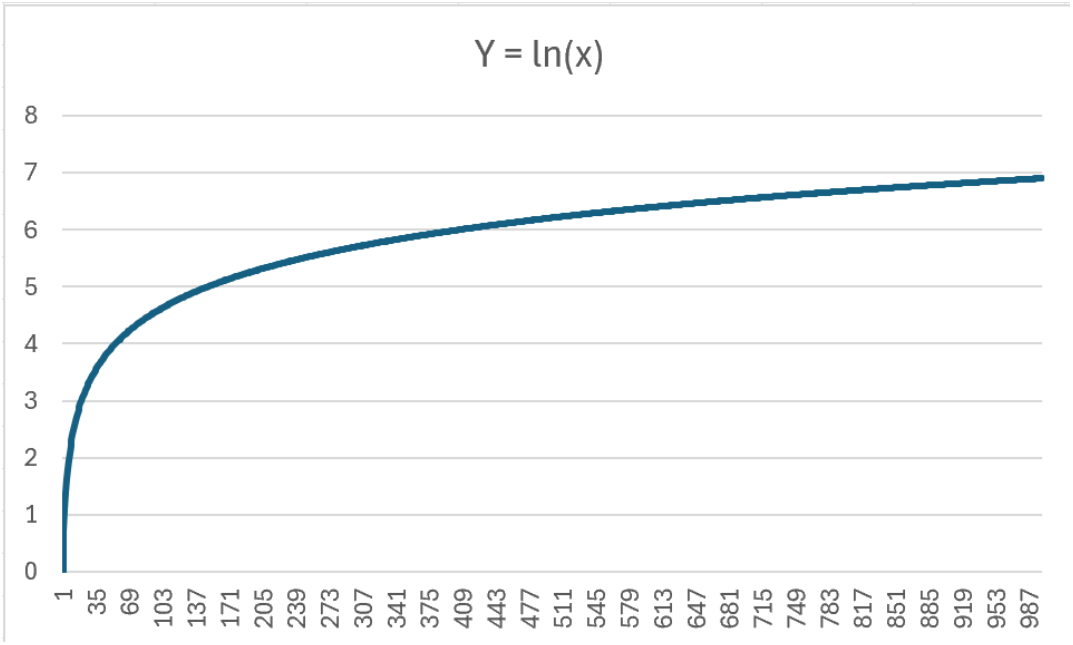
Contents of the SaltedLogFunction.csv

```
X,Y,
1,-446.0000,
2,127.6931,
3,-436.9014,
4,-238.6137,
5,-115.3906,
6,-177.2082,
7,-274.0541,
8,420.0794,
9,-400.8028,
10,-257.6974,
11,114.3979,
12,276.4849,
13,241.5649,
```

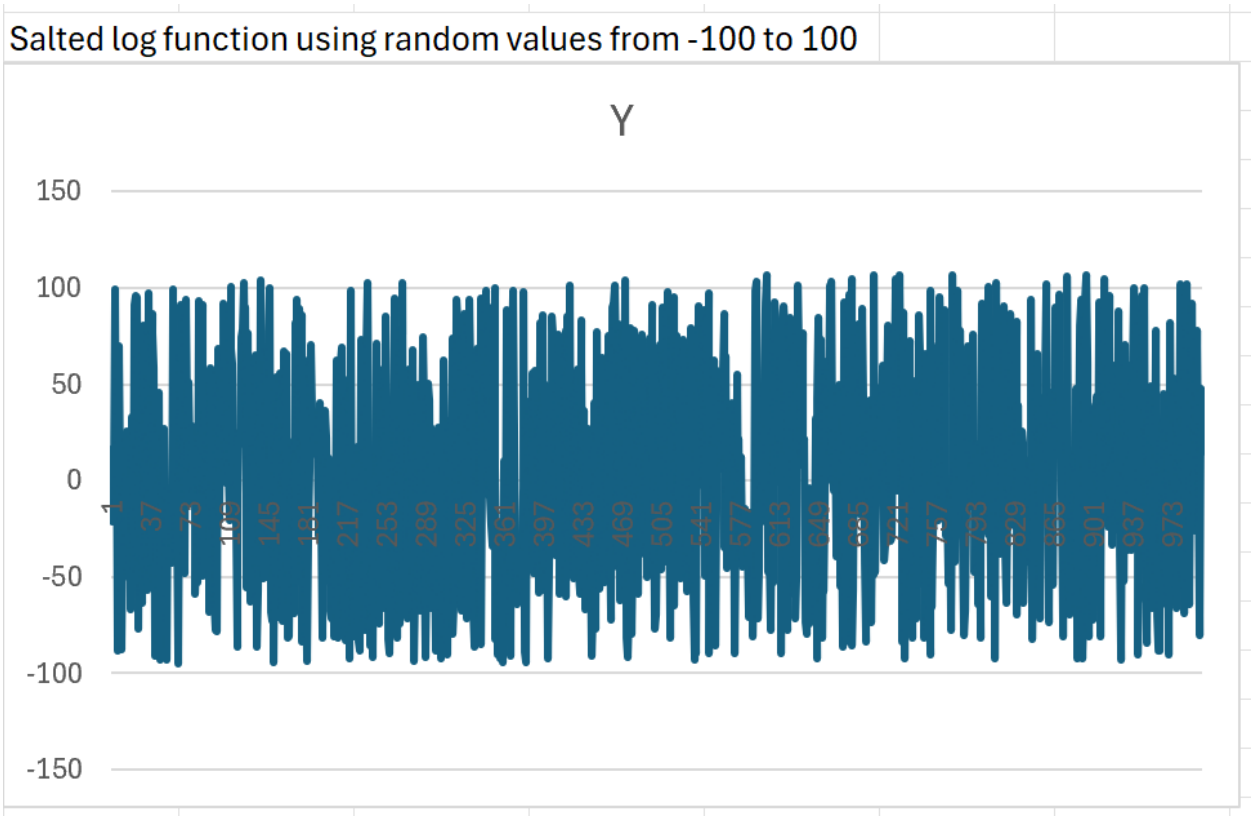
Contents of the SmoothedLogFunction.csv

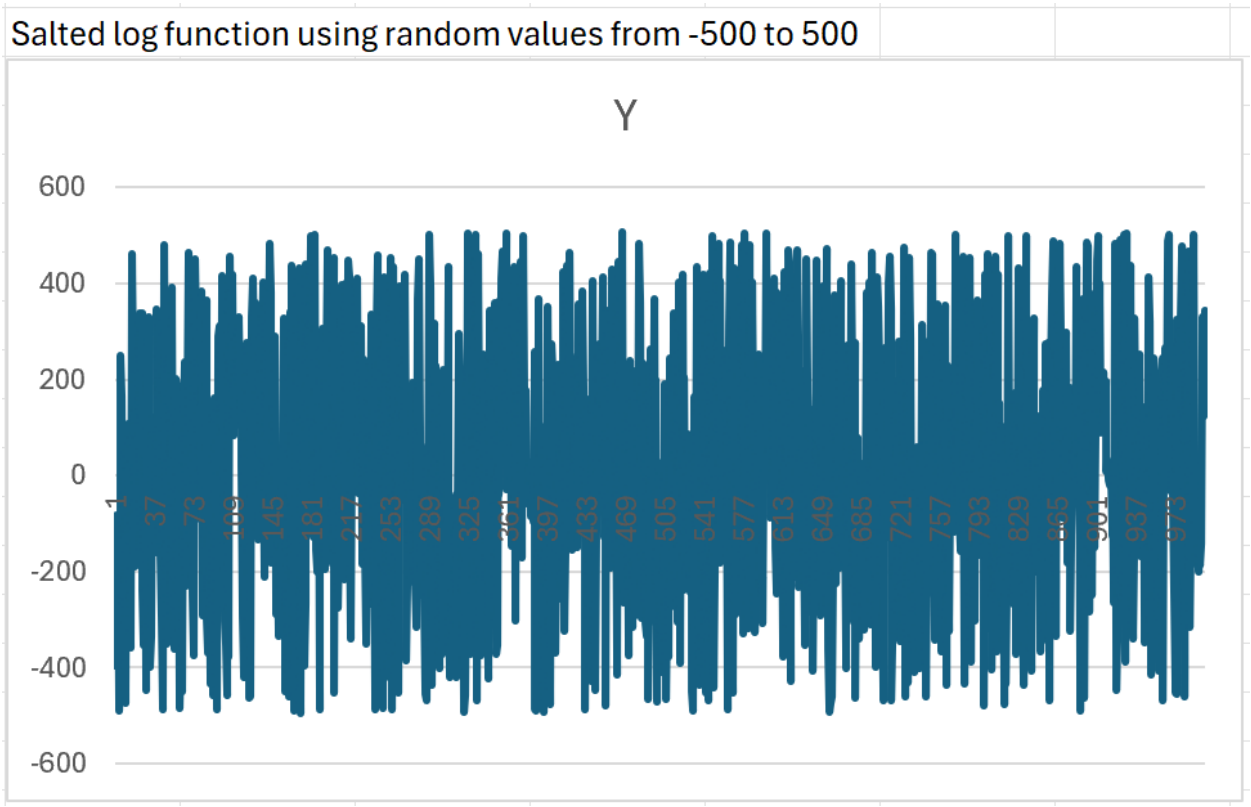
```
X,Y,
1,5.9951,
2,5.9935,
3,5.9920,
4,5.9904,
5,5.9889,
6,5.9873,
7,5.9857,
8,5.9842,
9,5.9826,
10,5.9810,
11,5.9795,
12,5.9779,
13,5.9763,
```

Excel chart of the log function

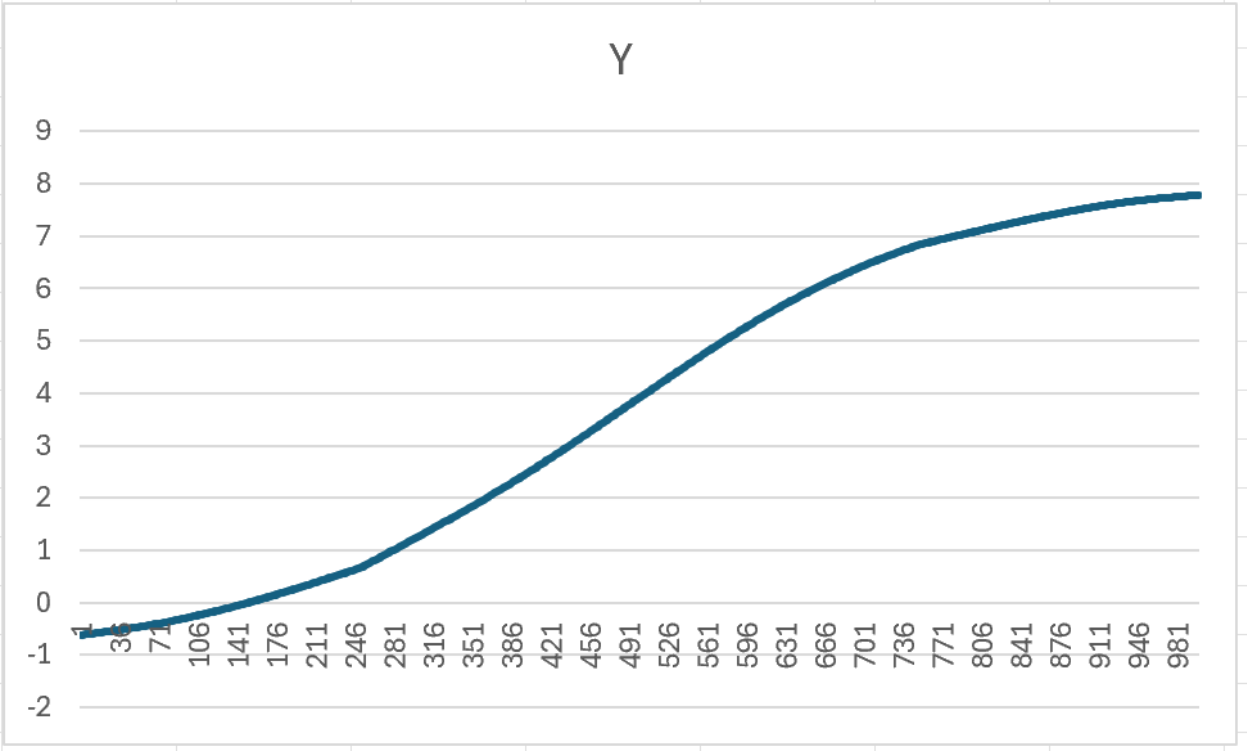


Experimenting with different values for different features

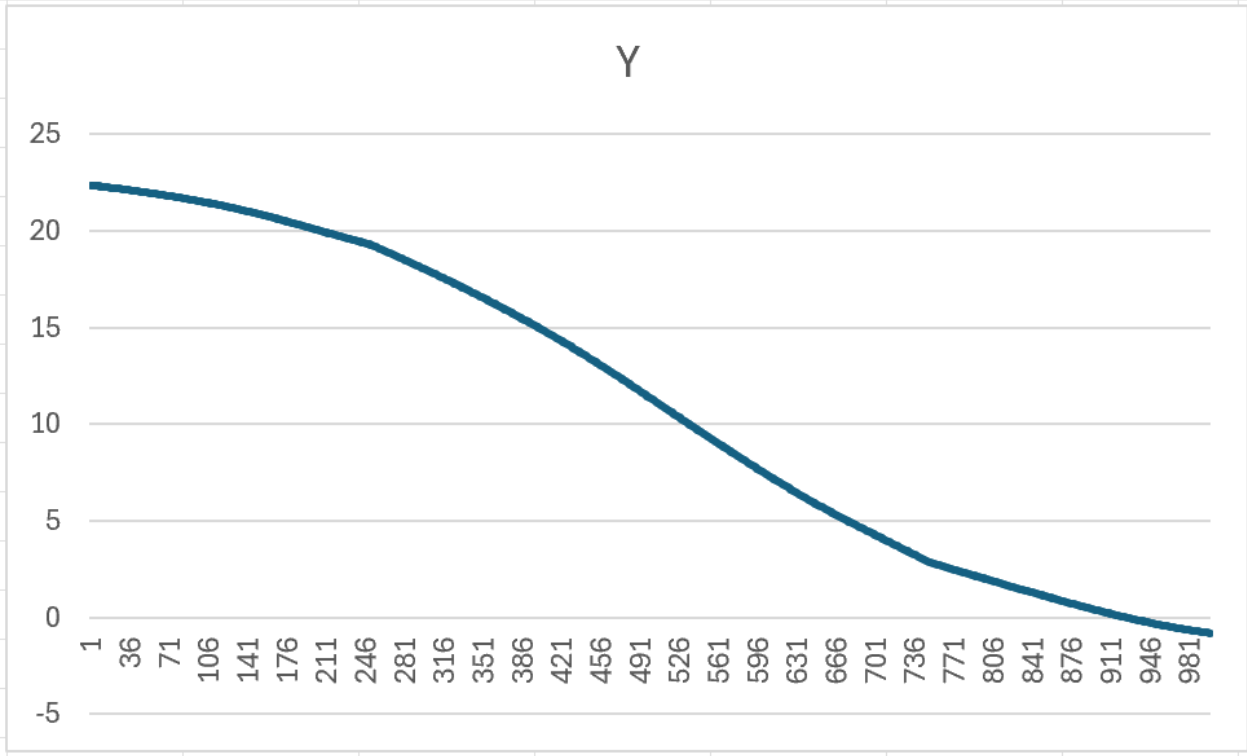




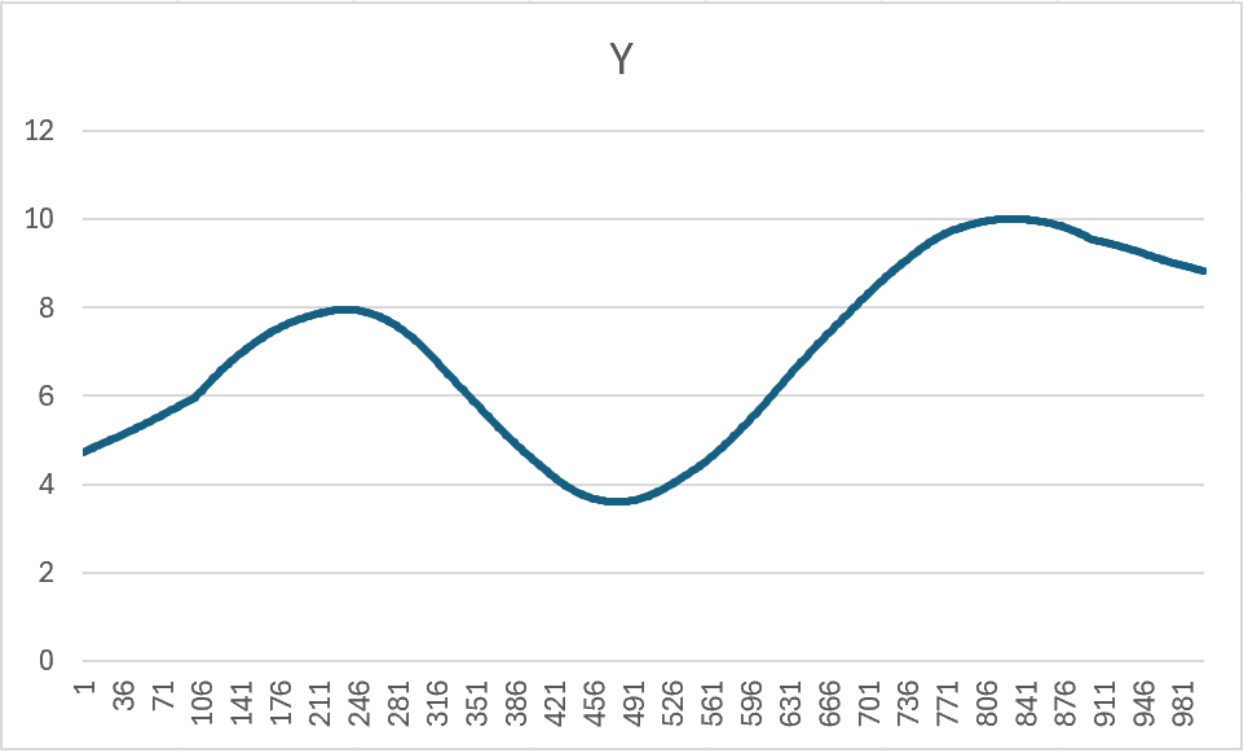
Smoothed log function using bounds -100 to 100 and window value 250, 3 runs



Smoothed log function using bounds -500 to 500 and window value 250, 3 runs



Smoothed log function using bounds -100 to 100 and window value 100, 3 runs



Smoothed log function using bounds -500 to 500 and window value 100, 3 runs