Project 2 Part 1

PSS1 contains 4 java class programs that make up the first section of this project. The objective of these classes is to plot, salt, and smooth data. Those classes are:

- **FunctionPlot.java** | File path: src\main\java\Part1_JavaPSSFIles\FunctionPlot.java
- This file contains 2 methods:
    - ***expFunc***(): which calculates $y = x^n$ and returns an arraylist containing 25 points from this function. Using the initialX parameter as the starting point for the graph, and the *n* parameter that the x-value is affected by. It stores the (x,y) values into an ArrayList of double [ ] type.
    - ***logFunc()***: which calculates $y = \log(x)$ and returns an array list containing 25 points from this function. Using the initial X parameter as the starting point for the graph, and storing the (x,y) values into an ArrayList of double [ ] type.
- **Salter.Java** | File path: src\main\java\Part1_JavaPSSFIles\Salter.java
    - This file contains two methods,
    - ***salter***: which in takes an array list parameter and randomly goes through the list replacing Y values.
        - This method modifies a provided dataset, replacing approximately 33% of the y-values with a random double between 0 and the highest value in the graph.
    - ***smoother():*** this in takes a salted dataset and smooths it out.
        - This method has a parameter called Window Value, which is the range of array list points that the smoothing method uses to calculate a mean of y-values to replace the active manipulated y-value. This effectively reduces the noise caused by the salting method.
- **DataHandler.java** | File path: src\main\java\Part1_JavaPSSFIles\DataHandler.java
    - ***csvOverWriter***(): this method takes in a array list dataset, and a file path string. It iterates through the dataset writing the coordinate points of each target point into a csv file that it creates at the file path string location. The reason it is called an over writer is due to it overwriting any pre-existing csv files with the same name.

    - ***csvCreator*():** this method works exactly the same as the over writer, except for the fact that it has a check to ensure the file does not already exist. If it does the method stops and prints an output letting the user know the file already exists.
    - ***dataReader():*** this method has a parameter of a file path string. It takes this file path which should be a csv file containing x and y coordinates. This goes line by line of the csv file, parsing the Strings into double values, and then adds them into an array list with a double[ ] type that the method returns when finished.
- **CsvTester** | File path: src\main\java\Part1_JavaPSSFIles\CsvTester.java
    - This is the tester file for part 1 of this project. It calls and provides example uses of all the previously mentioned methods.
    - Outputs for all of the previously mentioned methods are placed within ProjectOutput\ProjectPart1
    - This contains 3 csv files:
        - BaseplotOutPut.csv ,
        - SaltedPlotOutPut.csv ,
        - smoothedDataOutPut.csv
    - These show the effectiveness of all the previously mentioned classes.

PSS 2: Documentation is placed here: WriteUps\MatLab Documentation.pdf

- Script locations for PSS 2 are placed here: Project \MatLab Folder

PSS 3: This section contains 1 primary class dedicated to the function of creating graph .pngs using one of the dataset arrays created by the FunctionPlot.java class. That class is:

- **ChartImgCreator.java** |
  File Path: src\main\java\Part1_JavaPSSFIles\ChartImgCreator.java
    - ***lineChartMaker():*** The purpose of this method is to in take a dataset, graph title string, and a file path string to create a graph.png based off of that dataset. This utilizes JfreeCharts and add the dataset to a series collection and then creates a scatter chart of the dataset.

    - ***groupChartMaker():*** Similarly to the lineChartMaker() this takes In 3 datsets, a graph title string, and a file path string. It adds all 3 datasets to their own respective series before adding them to a series collection. Once they are in the series collection they create a group scatter graph. In order to differentiate between the three datasets, this method also utilizes the LineShapeAndRtenderer class which allows me to give different line thicknesses and colors to each individual graph.
    - ***groupChartApp():*** This an extra credit method that will be discussed in the Extra Credit Considerations documentation within the Write Ups folder.
- Outputs for this class are also present within the ProjectOutput\ProjectPart1
    - The files
        - ProjectOutput\ProjectPart1\BaseDataGraph.png
        - ProjectOutput\ProjectPart1\PSS Chart.png
    - Display the output of the methods above, BaseDataGraph.png being the output of groupChartMaker() and PSSChart.png being the output of lineChartMaker().

For Project Part 2 please open up the Part-2Contents folder within this write ups folder.