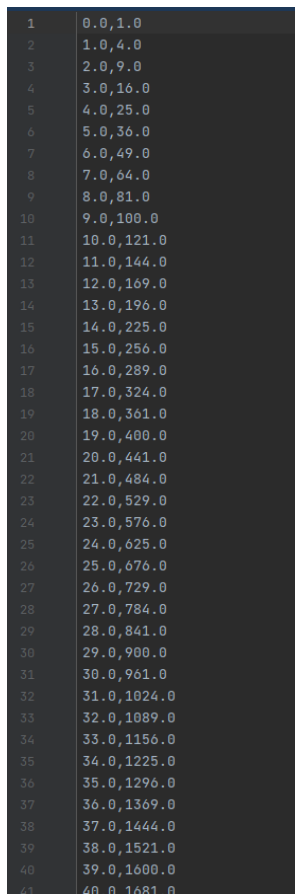
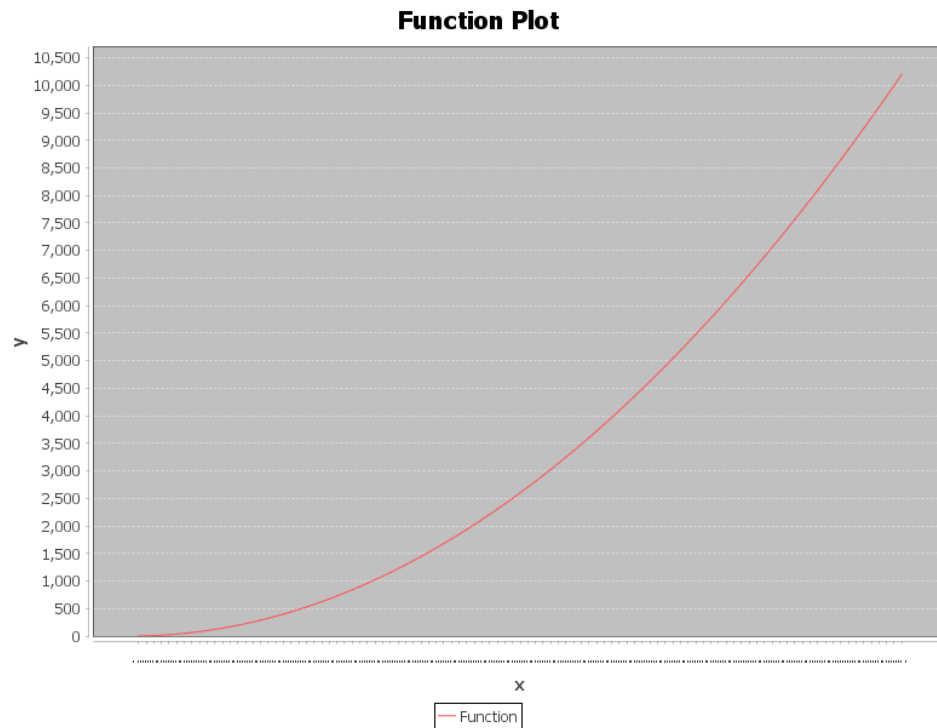


Importing Jars and Dependencies Report

Along with the normal java programming and then the MatLab programming for the plotter, salter, and smoother, the last thing we needed to learn was how to import from Apache Stats Library and JFree Charts to handle most of the code for us. To me, this was the hardest part of the entire project, and I was only able to find out how to plot the data, the salter and smoother unfortunately were too much for me to handle. Nonetheless, I did figure out the plotter and have some things to share about it. The first part required was finding where to download and import the Jar files to use the code's import statements. I was able to find these files able to download on the Apache statistics library and the J free charts websites. Once I had the files downloaded, I had to figure out how to add them to my java project on IntelliJ, which figuring out was not an easy process. In order to do this you need to open up the project structure, and head over to the modules setting and on the tab that says dependencies, that is where you add in the .jar files that I had previously downloaded. Once I figured out this seemingly easy step which took me longer than I would like to admit, I was able to add my imports and begin the code. I have a hefty amount of imports at the top of my code, the first of which are normal imports for CSV file writing, and the rest are from the Apache and jfree charts libraries. I use the Apache imports such as univariate function, polynomial function, and descriptive statistics to set my function, which is $x^2 + 2x + 1$ and then set the for loop that sets the range from 1-100 for the x values. The rest of the code is used to output this data into a CSV file, and also uses imports from JFree charts to output a Png file with its own graph, not needing to use excel for this step. Below I will include screenshots of the CSV file it produced along with the graph that came from the Png file.



1	0.0,1.0
2	1.0,4.0
3	2.0,9.0
4	3.0,16.0
5	4.0,25.0
6	5.0,36.0
7	6.0,49.0
8	7.0,64.0
9	8.0,81.0
10	9.0,100.0
11	10.0,121.0
12	11.0,144.0
13	12.0,169.0
14	13.0,196.0
15	14.0,225.0
16	15.0,256.0
17	16.0,289.0
18	17.0,324.0
19	18.0,361.0
20	19.0,400.0
21	20.0,441.0
22	21.0,484.0
23	22.0,529.0
24	23.0,576.0
25	24.0,625.0
26	25.0,676.0
27	26.0,729.0
28	27.0,784.0
29	28.0,841.0
30	29.0,900.0
31	30.0,961.0
32	31.0,1024.0
33	32.0,1089.0
34	33.0,1156.0
35	34.0,1225.0
36	35.0,1296.0
37	36.0,1369.0
38	37.0,1444.0
39	38.0,1521.0
40	39.0,1600.0
41	40.0,1681.0



This graph displays the same thing that the plotted data graph did for both the normal java code and the MatLab code. Personally, I like the way this graph looks and displays information better than how the excel graphs look for this. These imports are something I will look to be using in the future if I am tasked with similar projects.