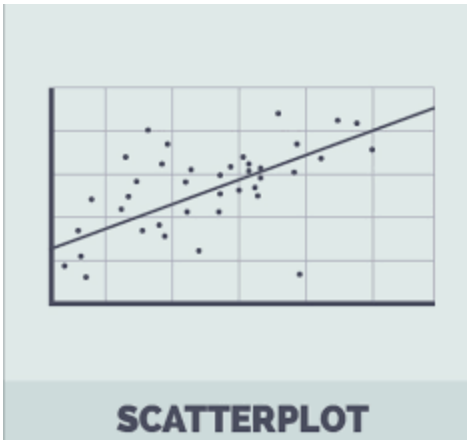
Java Programming
Storage Project - Scatter Plots

Required Skills: 1D/2D Arrays, ArrayLists, LinkedLists, Sorting & Searching, File Reading.



Assignment:In this project, you will take a series of points (x, y) showing a relationship between two quantities. The resulting output will be a scatterplot of the data as well as some statistical describers.

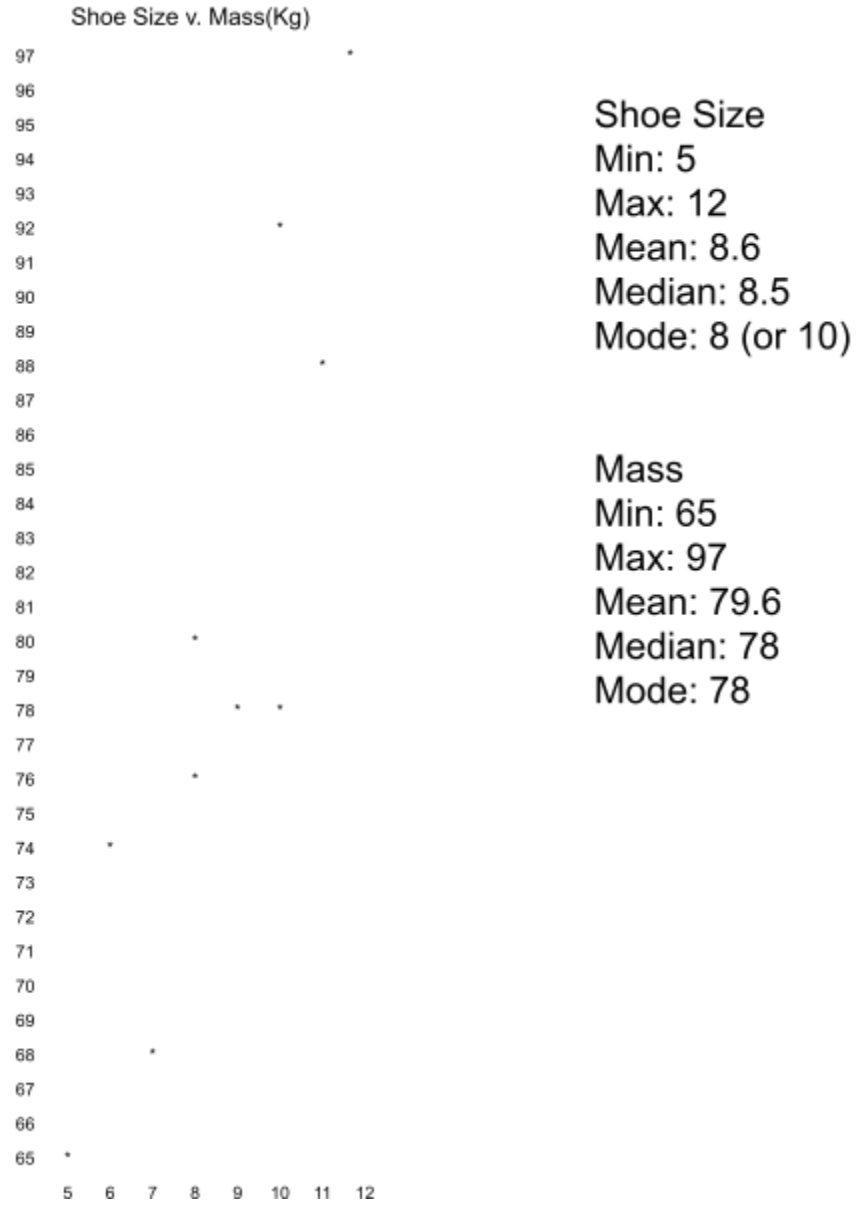
Requirements:

1. The data should be taken from a file as a comma separated list of points. x_1, y_1, x_2, y_2, x_3, y_3, …
2. The data should be transformed into an array of ints.
3. There should be a library file called ScatterPlot.java that contains the following methods:
    a. getXs - takes the original data in and returns an ArrayList of the x values
    b. getYs - takes the original data in and returns an ArrayList of the y values
    c. plot - takes the original data and returns a 2D array of Strings representing the scatter plot of the points. The first column will show the y values (all values from min to max) and the last row will show the x values(all values from min to max). The points should be represented using a *.
4. There should be a library file called Statistics.java that contains the following methods:
    a. getMax - takes an ArrayList of ints and returns the maximum value
    b. getMin - takes an ArrayList of ints and returns the minimum value
    c. mode - takes an ArrayList of ints and returns the mode. If there are multiple modes, return any one. (Hint: may be easier if sorted)
    d. mean - takes an ArrayList of ints and returns the mean
    e. median - takes an ArrayList of ints and returns the median (Hint: easier if sorted)
5. The output should have a scatterplot with title and the statistical measures of each of the measures.

Sample Input: The file will have three lines of data. The first line is the x variable, the second line is the y variable and the third line is the data.

Shoe Size
Mass(kg)
5, 65, 12, 97, 7, 68, 10, 92, 10, 78, 9, 78, 8, 76, 11, 88, 6, 74, 8, 80

Sample Output:
(the output will be stacked in the actual program - shown side by side here for ease of viewing)



Shoe Size
Min: 5
Max: 12
Mean: 8.6
Median: 8.5
Mode: 8 (or 10)

Mass
Min: 65
Max: 97
Mean: 79.6
Median: 78
Mode: 78

Grading:

	2	1	0
Coding Style	All code is well written and readable. Variable and method names are meaningful. Code is well organized, indented, and commented where necessary. Functions and events work well with each other.	Code is functional. Variable and method names are meaningful. There may be some confusing sections of code, but a majority of the code works as intended.	Code is hard to follow and more often than not does not work. There is a general lack of structure and readability in the code.
Modularization	Libraries and methods are used effectively and appropriately. The main method is used mostly as a control to take input, call methods and display output.	Libraries and methods are used, but may be over/under used. The main method contains some code that should be wrapped into a method.	Libraries/methods are not used or not used enough. The main method contains a majority of the code for the project.
Efficiency	Code is written efficiently. This includes all free form code and any code written in methods. Algorithms and code have clearly Pbeen planned, evaluated, and edited.	Code is mostly written efficiently. There is evidence of algorithmic planning, though the execution may not be in the most efficient, clearest form.	A majority of the code is not written efficiently. Code was clearly not planned or edited, just written.
Interaction	The file is read appropriately and without error. The output is clear and easy to read. Outputs are all in the specified format.	The file is read appropriately and without error. Output may be correct, but is difficult to read and may not be in the specified format. There may or may not be safeguards in place to correct user input.	The file input does not work.
Arrays	Arrays are used appropriately and with correct syntax and logic. Choices are made where to use arrays based on efficiency of code and run time analysis (big O). Arrays passed into methods are not altered unless that is the desired effect.	Arrays are used with correct syntax and logic. They may not be used appropriately at all times. Arrays passed into methods may be altered accidentally causing unwanted side effects.	Arrays are not used or not used appropriately most of the time.
Lists	Lists are used appropriately and with correct syntax and logic. Choices (ArrayList v LinkedList) are made to use lists based on efficiency of run code and run time analysis (big O). List methods are written generically to take both Array or Linked, based on programmers preference AND do not modify the list, unless that is the desired effect.	Lists are used with correct syntax and logic. Appropriate choices for the type of list may not be made at all times. Methods that use lists may not work for all list types or may alter the list causing unwanted side effects.	Lists are not used or not used appropriately most of the time.
Sorting/ Searching	When sorting and searching, appropriate algorithms are always selected based on the programmers knowledge of data size, purpose and efficiency (Big O). Sorting and searching methods are called from previously created library files.	When sorting and searching appropriate algorithms are often selected based on data size, purpose and efficiency (Big O). Sorting and searching methods are not called from previously created library files.	If sorting/searching is present, the appropriate choices are most often not made. Sorting and searching may not be implemented in method form, and are not called from previously created library files.
Correctness	Outputs for the program are correct for all runs of the program.	A majority of outputs are correct for all runs of the program. There may be minor calculation errors.	Few to no outputs are correct. There may be major, and far reaching, calculation errors.
Final Product	The final product shows significant student learning and reflection. Design decisions show thoughtfulness and provide a logical and smooth flow of information. There is evidence of exemplary student effort.	The final project shows some evidence of student learning and reflection. Design decisions show some thoughtfulness and provide all a logical flow of information. There is evidence of student effort.	The final project shows minimal student learning and/or reflection. Design decisions seem unfounded and expedient. There is minimal evidence of student effort

---

18 -> 100
17 -> 95
15/16 -> 90

14 -> 85
12/13 -> 80

10/11 -> 75
9 ->70

8 -> 65
6/7 -> 60

5 -> 50
4 -> 40
3 -> 30
2 -> 20
1 -> 10
0 -> 0

NOTES: