CIS351

Homework 4

Academic Integrity

- Remember the Academic Integrity Policies, of this course.

Objectives

- Read input data into an Arraylist.
- Compute statistics using an Arraylist.
- Process command-line arguments.

Additional Help

- Feel free to watch the video on Arraylist
- Java ArrayList API

Background

Standard deviation is a measure of *the spread* of a set of values around their mean. The higher the standard deviation, the more the values are scattered away from the mean. See the Wikipedia article for more details and examples.

Part 1: Understand the algorithm

Given a set of N values (x1 through xn) with mean x̄, the standard deviation σ is:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

Finding the standard deviation involves two steps: one that computes the mean and another that computes the average of the squared differences (i.e., the variance). At the end, take the square root of the variance to determine the standard deviation.

- Compute the mean and standard deviation for the following numbers by hand: {82, 88, 97, 80, 79, 92}
- Make sure you get 86.3 for the mean and 6.6 for the standard deviation. Think about the algorithm you used.

Part 2: Creating an Arraylist from input values

1. Create two Java classes for this homework: **StatDriver.java** with a main method, and **Stats.java** without a main method. The goal of the main method is to read a series of numbers from System.in, create an Arraylist containing those numbers, call the mean and standard deviation functions of the Stats class, and then report the results.

2. While executing your program, using the main written in `StatDriver.java`, user should be able to provide a string on the command line indicating whether they want to calculate the mean ("mean") or the standard deviation ("std"). If no argument

is provided, the mean should be calculated.

Help: the argument `args` *, in your main method parameter, is an array of Strings. Read more about it*
https://www.javatpoint.com/command-line-argument

3. Look at how the `StatDriver` class is executed from the terminal, using the java command.

```
→  Sol java StatDriver mean
4
8
1
7
3
0
-6
a
Mean: 2.43
→  Sol java StatDriver std
4
8
1
7
3
0
-6
a
StdDev: 4.37
```

Here, the `java` command is taking in the class name and a paramter *(i.e. mean or std)*. This parameter gets saved in args[0] position of the args parameter of `main` method. Hence, you can check `args[0].equals("mean")` to test if the user wants to compute mean. Similar checking can be done of `std` as well.

4. While entering the numbers from the terminal, as program input, the user should be able to indicate that no more numbers will be entered by pressing CTRL-d or any non-number character (ex.letters A-Z) in the terminal. ***See the example of the program interaction provided below to have a better understanding.***

5. To start coding, first create an Arraylist of double values: `ArrayList<Double> data` in your main method.

6. Now, use a loop to read the input values from the keyboard (as doubles) and store them into the Arraylist. No prompt should be given before reading the values. The loop should terminate when user indicates that no more numbers will be entered by pressing CTRL-d in the terminal.

The following lines of code will allow you to read the numbers from the terminal and will store it in an Arraylist named **data,** till user presses CTRL-d in the terminal:

Scanner input = new Scanner(System.in);

while (input.hasNextDouble()) {
       data.add(input.nextDouble());
    }

Note: In this example, the while loop works for anything other than numbers. User can either press ctrl-d or any non-number (ex.letters A-Z) in the keyboard to stop taking input.

7. Once the Arraylist is built, if the user wanted to compute mean, you will call the appropriate method (either mean or stdDev) of Stats class and pass this Arraylist as the argument.

8. Here are some example interaction with the completed program:

```
rahma2fx@L25001:~$ java StatDriver mean
1
```

```
2
3
4
Mean: 2.50
rahma2fx@L25001:~$ java StatDriver std
1
2
3
4
StdDev: 1.12
rahma2fx@L25001:~$ java StatDriver
1
2
3
4
Mean: 2.50
```

CAUTION: Do not move on until you know the input is working.

Part 3: Writing a method to calculate the mean

1. In the Stats.java file, create a public static method called **mean** that will take an Arraylist of double values as its parameter, and then calculate and return the mean of those numbers (which is of double datatype).

2. The method should ensure that the Arraylist is not null and that it has more than 0 elements. If either of these conditions exists, the method should return Double.NaN (not a number).

3. Test your mean method from the StatDriver.java program. The main method should output the return value as a single line, using the format "Mean: %.2f\n".

Part 4: Calculating the standard deviation

1. Add a public static method called **stdDev** in Stats.java. This method should compute the standard deviation of a given Arraylist of double numbers and return that double number.

2. Your stdDev method should call the mean method created in Part 3 above.

3. The method should ensure that the Arraylist is not null and that it has more than 0 elements. If either of those conditions exists, return Double.NaN (not a number).

4. Add to the driver program a single line of output following the mean output line. The format of the standard deviation line should be "StdDev: %.2f\n".

Test cases

If you want to test your program, after your implementation, compile both java classes and execute StatDriver.java. Provide the following data as input:

```
4
8
1
7
3
0
-6
a
```

For Mean, you should have: 2.43

For Std, you should have: 4.37

Grading Criteria

Total points: 100 points
Filled out Reflection Form - 10 pt
General Programming Correctness: Compile and execute - 10pt
Part 2 - 20pt
Part 3 - 30pt
Part 4 - 30pt

Submission Instructions

Create a zip folder containing the following and submit the zip folde rin Blackboard

1. **StatDriver.java**
2. **Stats.java**
3. filled out Reflection Form (https://farahman.github.io/reflection-form.pdf)
4. filled out cover sheet (https://farahman.github.io/CoverSheet.pdf)

Farzana Rahman / frahman@syr.edu