Simple Statistics

Portfolio Exercise

This portfolio exercise covers…

- The C# language.

Brief

Using C#, we are going to set up a simple class for doing some basic statistical analysis on an array of numbers. There's nothing to do with mobile apps here, just getting used to the language.

- Start a new project, select console app and call it "Statistics". A console app just means that we'll be dealing with C# alone, with no user interface except some text on the screen
- Add a new class to the project and call it "NumberCruncher".
- Add the following two properties to the class:
    - An array of doubles, with a size of 1000, called "data". This array will store some kind of data - temperatures, weights… it doesn't matter as long as they're numbers.
    - A integer called "count". This will keep track of how many numbers have been loaded into the data array.
- Create an instance of your NumberCruncher class called "test" in the main() method of your program and run just to make sure it's all working.
- Now, add the following methods. After each method, add some code to main() to check the new code working. So, after you've finished the addNumber() method, use it to add a few numbers to your test class. You don't have to do them in order so feel free to jump around.
    - **void addNumber(double)** - This adds a number into the next slot in the array and updates the count.
    - **void displayData()** - Displays all stored numbers to the console. This is actually a bad idea (why..?) but will be useful for our own testing.
    - **void removeLastNumber()** - Removes the last number in the array and updates the count.
    - **void removeNumberAt (int)** - Removes the number at the index provided and shuffles all the numbers that come after it down to fill the gap. Don't forget to update the count.
    - **double total()** - Returns the total of the numbers in the array.
    - **double average()** - Returns the average of the numbers in the array.
    - **double mean()** - Returns the mean of the numbers in the array. Since the mean is just the average, we can call the average() method. (In which case, why would we want both mean() and average()?)
    - **double minimum()** - Returns the lowest number in the data.
    - **double maximum()** - Returns the highest number in the data.

- **double range()** - Returns the range of the data (highest minus lowest).

Challenge

Create a method called mode() that returns the mode of the numbers (that is, the number that occurs the most often in the data). This is a difficult problem so don’t worry if you can’t do it. However, you should make a solid attempt.

Submission

Show the lecturer your working code before you submit. Submit via Blackboard under “Assessment” in the sidebar.