

CSE 411: Advanced Programming Techniques
Fall 2017

Programming assignment 1: Programming Warm-up

Due: on September 6nd (Wednesday), 11:59pm; Total: 20 points

Imagine that you have a brilliant research idea for super efficient indexing of a large quantity of web pages. You have developed a prototype implementation based on the idea. Let us assume a previous system (by some other research group) can index webpages at the speed of 10,000 pages per second. You want to show through experiments that your new system can do better in terms of speed. On six different experiments (perhaps using different sets of webpages), your experiments show that the system's processing speed is:

11300
9890
10400
9900
10545
12334

Your research supervisor is uncertain whether this set of data shows the superiority of the new system and asks you to implement a program for performing a one-sample t-test to check whether with significance your system indexes webpages faster than 10,000 pages per second. Therefore, the first task of this assignment is to implement generic one-sample t-testing.

Imagine now you have another idea for optimizing your system. After implementing the new idea, the following pairs of data show the processing speed of the original system and the new system with the optimization, for six experiments:

11300 11400
9890 9800
10400 11345
9900 9739
10545 10787
12334 12555

Given this set of data, your supervisor asks you to implement a paired t-test to check whether there is a statistical significance that shows the new system is faster than the original one. Therefore, the second task of this assignment is to implement generic paired t-testing.

You may find that some aspects of this assignment seem vague. Assume (correctly) that your supervisor expects you to be professional in your response, even in the face of some potential ambiguity. For instance, if you don't know how to do t-tests, you should do a search and learn by yourself. In the same spirit, you should design your own input-output format.

Your implementation may be in any language, and must be implemented by you (**if your language includes a library for performing t-tests, you should not use it**). The implementation you turn in must:

1. be available in (relatively compact) source code form;
2. be compilable (if appropriate) and runnable on the CSE Sun lab machines;
3. be with good documentation (including comments in the code as well as a readme file documenting the interface of the program and information that can enable other people to run the code).

Note: it is sufficient to calculate t-values in your t-tests; the conversion from t-values to p-values is optional.

Submission format. You should submit a zipped package (that contains all your code and documentation) through CourseSite.

Your program will be graded not just by correctness, but also by efficiency, documentation, readability, style, robustness (gracefully handles unexpected input), and usability.