University of Glasgow | School of Computing Science

Assessed Coursework

Course Name	**Programming Methodology (PM)**			
Coursework Number	**2**			
Deadline	**Time:**	**23:59**	**Date:**	**Sun 24 Nov 2019**
% Contribution to final course mark	**18%**			
Solo or Group ✓	**Solo**		**Group**	✓
Anticipated Hours	**10 per group member**			
Submission Instructions	**1. Submit via SIT_Xsite a Zip or compressed tar file containing your source code directory (UoG-PM.tgz or UoG-PM.zip), plus one PMReportSessionxGroupxx.pdf file** **2. Submit a hardcopy PMReportSessionxGroupxx.pdf into the drop box at level 4 UoG office SIT@NYP**			
Please Note: This Coursework cannot be Re-Done				

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below.

The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline
   a. the work will be assessed in the usual way;
   b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than five working days after the deadline will be awarded Grade F.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause to Admin-In-Charge

Penalty for non-adherence to Submission Instructions is 2 bands

You must complete an "Own Work" form

Programming Methodology (UoG-PM) 2019-20

2nd Assessed Exercise: Parameter/Probability Distribution Estimation for Linear Regression in C program

Introduction

The goal of this second AE exercise is to further improve yourselves with the design, implementation and performance testing of C programming with the same given dataset especially if implemented in Internet of Things (IoT) platform. With more in depth problem statement with the same dataset, you will be required to go through the same whole cycle of problem definition, problem analysis, algorithm and pseudocode design and implement and testing the algorithm in C programming.

Dataset and Software

You will be working on a dataset **Group1_8.txt** for **Group1 to Group 8** and **Group 9_15.txt for Group 9 to Group 15** that comprises of

- 10,000 rows and 2 columns of floating numbers. They represented the 10,000 points of x, y coordinates of a straight-line equation $y = mx + c$ where m is the gradient and c is the y intercept. The first and second column separated by a comma is the x and y coordinate of each point respectively.
- the y coordinates of a straight-line equation are being corrupted by random distributed noise which is statistically independent random variable. In the summary, we can say that the straight-line equation is represented by

  $$y = mx + c + n$$

  where n is the random distributed noise with a mean μ and standard deviation σ

The same dataset will be uploaded again and be available in the SIT-Xsite Dropbox Folders under the course assessment folder and you should work on the same dataset that belongs to your grouping according to the grouping list in the **Final Assessed Exercise Grouping List.pdf.** The programming language is to be used is the same Standard C programming where Microsoft Visual Studio Code (VSC) is the baseline editor and GNU Compiler Collection (GCC) is the standard compiler. Your C Program is free to call/activate any plotting tool such as GNUplot at http://www.gnuplot.info

Assessed task

Your main task is triple fold

- find and calculate the parameters of the noise random variable n namely its mean μ and standard deviation σ respectively;
- find and plot the probability density distribution of noise random variable n; and
- Calculate the time required for your entire whole program from extracting the dataset, perform linear regression analysis and till finding out and plotting the probability density function of the noise random variable n.

Similarly, as in AE1, firstly looking at the dataset, you should first perform problem definition, problem analysis followed by the algorithm and pseudocode before proceed to the C programming coding. Your C program should be of originality as usual. Retaining the same baseline or improved version program as in AE1 as below

- Open the dataset file and extract the respective observation data point (x_1, y_1), $(x_2, y_2), (x_3, y_3) \cdots (x_{10,000}, y_{10,000})$ into array set.
- Write a function to perform a linear regression analysis and fits a straight line equation in the form of $y = mx + c$ to the set of observations $(x_1, y_1), (x_2, y_2) \cdots (x_{10,000}, y_{10,000})$. It should also compute the correlation coefficient, the coefficient of determination and the standard error of the estimate.
- The output program should plot the 10,000 points in the x, y plane and draw the estimated straight-line equation that will superimpose on these 10,000 points. The output program shall also print the result
    - The equation of the estimated straight line
    - correlation coefficient
    - coefficient of determination
    - standard error of the estimate

In additional, your program should cover the requirements need in AE2 which are

- write a function to plot the probability density distribution (pdf) of noise random variable n based on the given dataset.;
- write a function to obtain the parameter of the noise random variable n namely its mean μ and standard deviation σ respectively. Hence, compare with the plotted pdf and obtain the pdf function;
- time stamp the time required for your entire whole program from extracting the dataset, perform linear regression analysis till finding out and plotting the probability density function of the noise random variable n.

As such as possible, explore the use of pointer concept in your program and function.

What to handin

According to your Group number, use

1. SIT-Xsite Dropbox to submit a single zip or compressed tar file with the contents of the UoG-PM plus a separate PMReportSessionxGroupxx.pdf file and the scanned version of the signed declaration forms of all members in the group and
2. Provide a hardcopy PMReportGroupxx.pdf file attached with the original signed declaration forms of all group members at the Dropbox at level 4 UoG office at SIT@NYP

To aid in testing and assessing of your code, please make sure that:

- Your submission file is named UoG-PM.tgz or UoG-PM.zip.
- When uncompressed, your files will be in a folder named UoG-PM_SessionxGroupxx
- The folder should contain all the following
    - all the C source codes (.c or even .h codes) and the respective executable file (. exe file) that are necessary to take in the dataset and generate all the plots and the printed results
    - PMReportSessionxGroupxx.pdf where x and xx indicate your session and group number respectively according to the grouping list.
- Your PMReportSessionxGroupxx.pdf file should outline and contain
    - The name of the group member and the individual contribution.
    - your design solution in terms of
        - problem definition,
        - problem analysis,
        - algorithm and pseudocode.
    - all the source codes
    - all the plots and results
    - any interesting aspects of your solution (e.g., assumptions you've made, optimisations that you thought of, etc.),
- Please note that if your solution produces correct results, but this is only true under assumptions that were not explicitly stated in your report pdf file, there may be a deduction of marks.
- Your submission will be tested for plagiarism with a specialised off-the-shelf tool. Plagiarism cases will be dealt on a case-by-case basis but suffice to say there will be little tolerance.

How this exercise will be marked

Following timely submission, the exercise will be given a numerical mark between 0 (no submission) and 100 (perfect in every way). The numerical marks will then be converted to a band. The marking scheme is as follows:

- 30 marks for the quality and readability of the code; make sure that you use appropriate names for variables/function and that your source code is properly structured.
- 25 marks for comments and documentation in the code and your PMReportGroupxx.pdf file; make sure that you comment and document in your source files, at the very least, the basic steps taken to comment on the function and various variables used. Do not make an essay of your code; use your pdf file to discuss further details.
- 30 marks for a solution that produces the correct results; partial marks will be awarded if your solution is on the right track but produces incorrect results due to minor errors. Marks will be awarded based on originality of the codes. It is of utmost importance to demonstrate the team competency in originality in producing the various program modules and shall be generated by the team instead of getting from another available external libraries.
- 15 marks for any optimisations you have come up with, e.g., to reduce the number of linear regression computation time, faster time to read and print the results etc. especially as a way to show the efficiency of implementation if it will be implemented in power and memory limited IoT platform.