BIEN 4290 - 2022
Laboratory Exercise [1]

Objectives:
- Give students practice in using shell scripts for automated analysis of data.
- Develop library routines in C++ to perform basic statistics for use in future projects.
- Give students an opportunity to use statistical and graphical summary of physiologic data.
- Expose students to EEG processing and brain electrophysiology.
- Enforce technical writing skills.

Data: In the **/lab/bien4290/ERP** directory there are two sets of data files. Each set consists of 60 data files. Each data file is a **1-second event-related potential (ERP)** collected from the primary visual cortex of a rat brain. The data is sampled at a rate of 500 Hz. The amplitude of the signals ranges from -2.16 to 6.76 mVolts, and from -1.48 to 2.37 mVolts for each of two sets, respectively.

The EEG data were collected from a rat while an anesthetic agent was administered in varying concentrations. Simultaneous recordings were made continuously from four different locations in the brain. One set of data, labeled **ERP00**, was recorded during administration of 0.0% halothane while the second set of recordings, labeled **ERP05**, were recorded during administration 0.5% halothane anesthesia. Each of the 60 epochs within a specific anesthetic level represents an evoked response to a flash of light.

You may refer to class notes, a physiology textbook or various journal articles to find out more about evoked responses, EEG and the effect of anesthetic agents on brain function. Before summarizing the data numerically, be sure to plot these data files (using any spreadsheet or Matlab program) to see what the data look like.

PART 0: Basic requirements:

- The program must be placed in its own directory (e.g. Lab1).
    - This directory and all files within it must be Git tracked (e.g. run git init within the directory, git add, etc.)
- While developing, it is my expectation that the code is committed to your local Git repo at least as often as major functions are completed, or as bugs are fixed. Each commit should have a relevant message.
- The program's class should use a unique namespace.
- All functions should check the validity of their inputs.
- Follow the General Formatting for Code guidelines in the syllabus, create a comment block at the top of your *.hpp file, using the comment for individual lines ("//"), or multiple lines ("/*" and "*/").
- Add header guards to the top and bottom of your header files, by using the preprocessor directives.
- **Be sure to compile your code using g++, instead of gcc!**
- **You may use any array type (C-style arrays, std::array, std::vector, etc) but be consistent.**

Part I: Statistical routines (Week 1) – Due February 8th (10 pts for meeting milestone)

In this portion of the laboratory, you will be writing simple statistical routines in C++ that will be compiled and stored in your own library for use in this lab as well as future labs. You will develop two **separate** programs, one that estimates the statistics of your data (including mean, standard deviation, and a histogram) and a second that estimates the correlation coefficient between data files.

The program should meet the following requirements:

- It should be a set of files named **[yourlastname][yourfirstinitial]_stats_test.cpp**, **[yourlastname][yourfirstinitial]_stats.cpp**, **[yourlastname][yourfirstinitial]_stats.hpp**.
- The program's class should use a unique namespace.
- The _stats.cpp/hpp files should **at least** have **public** functions that:
    - Calculate the min/max of input arrays (and return the min/max respectively).
    - Calculate the mean of input arrays (and return the mean).
    - Calculate the standard deviation of input arrays (and return the standard deviation).
    - Estimate a histogram from input arguments.
        - The histogram should be centered at the dataset mean. Each bin should have a width of 0.4*(sample standard deviation). The bins should begin and end at sample mean ±3*(sample stddev), respectively.
        - The function should take in at least an array as an argument, and output an array corresponding to quantity of data within each bin.
- The _stats_test.cpp file should have ONLY a main function that:
    - Expects one input argument corresponding to the path of an input data file.
    - Ensures that all of the data points are correctly read.
    - If the data file cannot be found, an error should be printed to the screen and data analysis should be terminated.
    - Calculates and writes the following quantities to the screen (being sure to provide a label). (e.g. the screen should display "Number of points: N").
        - Number of data points analyzed
        - Mean value of data
        - Standard deviation of the data
        - Minimum of data
        - Maximum of data
        - Histogram (use stars (for vertical) or equals symbols (for horizontal) to create the bars for each bin. Each star should represent a specific quantity- this quantity should be displayed below the histogram.

The correlation program should meet the following requirements:

- It should be a pair of files named **[yourlastname][yourfirstinitial]_corr_test.cpp**, **[yourlastname][yourfirstinitial]_corr.cpp**, **[yourlastname][yourfirstinitial]_corr.h**.
- The program's class should use a unique namespace, or the same namespace as the previous program.
- It _corr.cpp/hpp files should have a class function that:
    - Estimate the correlation coefficient between the two data sets. The equation for the correlation coefficient is as follows:

$$r_{xy} = \frac{\sum_{i}^{N}(x_i y_i) - N\overline{xy}}{\sqrt{\left(\left(\sum_{i}^{N} x_i^2\right) - N\overline{x}^2\right)\left(\left(\sum_{i}^{N} y_i^2\right) - N\overline{y}^2\right)}}$$

- _corr_test.cpp should have ONLY a main function that:
    - Expects two input arguments corresponding to the paths of two data files.
    - Ensures that all of the data points from both files are correctly read and parsed.
    - If a data file cannot be found, error should be printed to the screen and data analysis should be terminated.
    - Writes the following quantities to the screen:

- Correlation value (just the number with no accompanying text)

You must detail a test plan for verifying proper performance of your code (may include using simulated data and error conditions).

Part II: Data analysis and Shell scripting (Week 2)

Once you have verified that your programs in part 1 are working properly, put the working, compiled program in your bin of C++ programs and do not edit the program anymore.

You will now use shell scripting and your C++ algorithms to analyze the EEG data in **/lab/bien4290/ERP**. Copy the EEG data should be kept in a subdirectory in your **Lab 1** folder called **EEG_DATA**.

A. ***Histogram analysis:*** For a specific anesthetic concentration, you will combine your data from all 60 data files into one data set. You will then estimate a histogram for that single, combined data set.
Your shell script should perform the following:

- Prompt user for the concentrations level to analyze.
- Automatically concatenate all 60 data files from the appropriate concentration level into one data file (you choose a file name). This new data should remain in the directory, **EEG_DATA**
- Copy the new data file to a file called **catted_input.dat**
- Execute the histogram program for this new data set. (The histogram program should remain in your C library)
- Write the output of the histogram program to a file (you choose a name that reflects concentration level and histogram results). This out file should be placed in **EEG_DATA**.
- Print to the screen where the output is stored, and verify that this is true.

Perform an analysis on paired halothane concentrations.

B. ***Correlation analysis:*** For a specific anesthetic concentration, you will analyze the correlation coefficient between consecutive data files (evoked response). In other words, you will calculate a correlation coefficient for epoch 0 vs. epoch 1, and then epoch 1 vs. epoch 2, and then epoch 2 vs. epoch 3, etc. You will estimate a total of 59 correlation coefficients for a specific anesthetic concentration.
Your shell script should perform the following:

- Prompt user for the concentration to analyze (00 or 05).
- Use a loop to automatically analyze two consecutive data files (starting with epoch 0 and epoch 1) at a time until all pairs for data files have been analyzed as described above.
  - Append the output to a file named **conc_corr_[concentration value].csv**, where each row consists of a label describing the two epochs that are being compared (e.g. 0 vs 1) and the correlation value.
  - The label and the value should be in separate "columns" as defined by a csv: **in other words, each row's values must be separated by a comma!**
- The correlation program should remain in your C++ library.

Perform this correlation analysis for the anesthetic concentrations. Plot the output for each concentration level using spreadsheet software or MATLAB. Note that your graph should plot correlation coefficient vs. epoch number.

Lab Report:

(LIMIT 4 PAGES of 11-pt text, not including graphs and tables). DO NOT EXCEED.

Your technical communication will consist of a **formal lab report. No abstract** is required for this report.

In your **introduction**, tell the read a few words about the type of data you are analyzing and how you are going to analyze it in this laboratory. You might also tell the reader why such analysis might me important (this is a good place to cite references to the literature regarding EEG, evoked potentials, anesthesia, etc). Cite at least 2 references. You might also want to discuss the advantages of shell scripting in this study of EEG.

In your **methods**, be sure to give a description of how the data were collected and then briefly describe the statistics that you used to analyze the data (also tell reader what files you analyzed/ number of data points analyzed?). Describe your rationale for the design of both your C++ programs and shell scripts (efficiency, readability, error handling, flexibility, etc.). Describe how you verified proper performance of your C++ code. **"It compiles" is not a sufficient answer.**

In your **results**, briefly describe your success in writing the C++ programs and shells scripts and how you verified proper performance of the code. Provide illustrative plots and/or tables for the EEG data analyzed and the statistics that you performed. For each plot and table, provide an explanation **(just the facts!** No interpretation!) for the reader of what you observe about the data. Do not just insert a collection of plots and table with no explanation.

In your **discussion/conclusion**, summarize the importance of the work you performed (similar to introduction), summarize the key observations (no need to repeat numbers, just general trends), provide an interpretation of why you observed what you observed (again—citations to the literature may be useful here). Did the graphical analysis and/or statistics provide any surprising results? If you had certain difficulties writing code to perform data analysis, you might elaborate on the problems you encountered and how you dealt with the issue.

****Note**: In addition to the formal lab report, please hand in **all C++ code (including the git folder!), scripts, and output files** required for this lab in a zip file, and the **graphs, plots, and/or tables** as an appendix to your lab report. That is, you should turn a one document that contains the formal report, and a **separate** zip file with the code.

Your final document must be submitted to the D2L drop box by 2:05 pm on Tuesday Feb 15th.

You will also demo your working shell scripts and C++ programs at the beginning of lab on Tuesday Feb 23rd to your TAs.

Your final grade will be weighted 70% for the code (equal contributions for C++ code and shell scripts) and 30% for the final report.