Data Technologies (STATS 220)
The University of Auckland
Paul Murrell

Course Description:
This course introduces a variety of computer technologies relevant to storing, managing, and processing data. The course has two aims: to teach software tools specific to the handling of data, and to teach and build confidence with general concepts of computer languages. It is useful for students with interests in applying statistics in business or research environments.

The course also aims to generally raise the students' awareness of the incredible range of tasks that a computer is capable of performing (in addition to providing concrete tools for performing specific tasks).

Topics: How to write computer code; publishing data on the world wide web (HTML); data description and semantic markup (XML); data storage (file formats, spreadsheets, databases); data management and summary (database queries, SQL); data processing (R).

Languages:
HTML (CSS), XML (DTD), SQL, and R. No prior experience with writing code needed.

Text: The course text is a CC-NC-SA licensed work,
https://www.stat.auckland.ac.nz/~paul/ItDT/

Pre-requisites: passing grade on stage 1 paper in Statistics or Computer Science.

There are two lectures per week, plus one lab per week.

Assessment:
lab work (.5% per lab for a total of 5%),
three assignments (15%),
mid-term test and final exam (total 80%).

Labs:
Lab 1: Modify an HTML document to indent/layout a section of code and to fix a simple error; the goal is to introduce ideas of code layout and commenting, plus reading and interpreting computer error messages.

Lab 2: Write an HTML document from scratch (by writing raw HTML and CSS code); the goal is to reinforce ideas of code layout and commenting, plus develop confidence with writing code, plus develop a basic understanding of HTML and CSS syntax and semantics.

Lab 3: Identify the file format for a selection of files that have had their file name suffixes stripped (by trying to open the files in different software) and perform a simple calculation with the data each file; the goal is to learn the difference between simple text files and complex binary files, what that implies in terms of the software required to open the file, and what that implies in terms of how easy it is to perform calculations with the data in the file.

Lab 4: Write an XML document with DTD from scratch; the goal is to start thinking about storing data in ways other than just in rows and columns, plus develop a basic understanding of XML and DTD syntax and semantics (the latter being an example of the expressiveness of code).

Lab 5: Convert a SIMPLE data set from a single flat text file format to a database in third normalised form; the goal is to think about data storage in a slightly more formal way, plus develop a basic understanding of how relational databases are structured.

Lab 6: Write SQL queries to access a pre-prepared MySQL database; the goal is to develop a basic understanding of SQL syntax and semantics, plus we reinforce the ideas of relational database structure.

Lab 7: Write R code to read a file and perform simple arithmetic on the contents; the goal is to develop a basic understanding of the syntax of R expressions, plus R functions to read text files, assigning values to symbols, and arithmetic operators.

Lab 8: Write R code to generate sequences and run simple for loops; the goal is to develop an awareness of the iterative capabilities of a general-purpose language. This is demonstrated by reading data from several files and calculating a summary statistic across all of the files.

Lab 9: Write R code to perform more sophisticated loops (more expressions in the body of the loop); the goal is to develop an understanding of fundamental data structures and performing tasks that require more steps than a single R function call.

Lab 10: Write R code to perform more sophisticated calculations; the goal is to develop an understanding of subsetting data structures, plus to demonstrate the idea of building more complex code solutions by starting small and slowly adding complexity.

Assignments:
Ass 1: Write an HTML document from scratch (like Lab 2 only harder)

Ass 2: Write an XML document with DTD from scratch and design a database in third normal form (like Labs 4 and 5 only harder).

Ass 3: Write R code (like Labs 7 to 10 only coherent in the sense of building a larger task in small steps)