



School of Computer Science and Software Engineering

CITS4407 Open Source Tools and Scripting

Assignment 3

In this unit we have focused on open-source tools that are designed to filter and process textfiles and streams of text. Notably we have not focused on generating or manipulating images or graphs. However this knowledge is very useful when visualising large datasets, as most contemporary software (particularly web-based software) to produce images and graphs require their data and commands to be presented as plain text.

Some excellent visualisation examples include:

- [Google Charts](#),
- [D3](#), and
- [Highcharts](#).

The goal of this assignment is to assess your understanding of the use of the shell and open source tools to effectively visualise some large datasets. You will be assessed on the quality of your shellscripts to produce/manipulate the data for the visualisations, but not on how aesthetic or beautiful your visualisations are. Marks will also be awarded for attempting to identify and highlight some 'interesting' aspects of the data - not just a simple 2-D plot of 2 unrelated variables.

- The assignment contributes **30%** towards your final mark in CITS4407 this semester, and is to be completed as individual work.
- The deadline for this assignment is **12noon Friday 4th November**.
- Submit your assignment as either one or more files (in a single archive, or as individual files), using [cssubmit](#). Please DO NOT submit your scripts in a Microsoft Word file.
- You are welcome to undertake the project on your home or laptop computers. Please note, however, that all materials submitted for marking must be working on a CSSE OS-X or Linux machine by the due date.

Tasks

Remember, it's considered good practice to include some comments in programs and shellscripts, to explain the author's design and logic. Include your name and student number in a comment near the top of each of your shellscripts.

1. The source of the input data for this task is a sequence of captured wireless Ethernet (WiFi) frames. The contents of each frame have then been formatted to a textfile, providing details of each frame (one frame per line). Note that only the frame's *header* is captured, and none of its data-payload (which is likely encrypted, anyway). Thus, the only privacy concerns exposed by this data include which device was communicating with which other device, how often, how much, and at what time. No personal or private data is exposed.

The fields of each line (frame) include: time-of-day (in microseconds), the transmitting device's distinct MAC address, the receiving device's distinct MAC address, the length (in bytes) of the frame, the signal strength with which the frame was received, and a short English description of the frame.

Choose one (or more) visualisations from the list, above, to effectively display captured WiFi packets. Develop a shellscript to produce the one or more files required to create the visualisation.

2. The input data for this task is a typical web-server's logfile recording the requests made of the web-server's URLs. Each request is formatted, one per line, and each request's fields include: the requesting IP address, data and time of the request, the URL requested, the web-server's integer return code (indicating success or error), and the number of bytes transferred.

Choose one (or more) visualisations from the list, above, to effectively display web-server logfiles. Develop a shellscript to produce the one or more files required to create the visualisation.