N.B all the commands shown below are case sensitive, so do NOT capitalise them!! The command prompts indicate that they are ready for you to enter code with a symbol, such as $, it is whatever symbol is last on the line of the command prompt when you first open it. When you see a line ending in this symbol, it means the command prompt has completed whatever it is doing and is ready for you to enter a new command.

1 Installation, mainly for Katie

How I installed on Andy's Windows machine, no Admin privileges necessary.

- In the Software Center, download Anaconda (find by searching for Python)
- In the Software Center, download Git (find by searching for Git)
- Get a copy of the code locally:
  - Open Git Bash terminal- this looks like a command prompt
  - Move the current working directory of the terminal to the folder you would like to store the application in- `cd full/path/to/folder`- (you can find the full path by going to the desired folder in Windows finder and copying and pasting the content of the bar at the top)
  - Copy the code from the web using `git clone https://github.com/kebarr/raman_analysis.git`
  - Go to the right git branch: `git checkout python3`
- Set up everything you need for the code to run:
  - Go to the Anaconda terminal- this allows us to download the necessary Python modules- by going to the Anaconda application folder and clicking the terminal app
  - We should now have another command prompt open

  - `cd` into the same directory as you went into in the Git terminal, but with `raman_analysis` at the end of the folder path. I.e. `cd full/path/to/folder/raman_analysis`
  - Into this command prompt, enter `conda create --name raman python=3.5` flask bootstrap has incompatibility issues with later python. install flask-bootstrap first (see below)
  - enter the commands for Anaconda to install the necessary modules, such as `conda install numpy`, `conda install pandas`, `conda install matplotlib`, `conda install scipy`, `conda install scikit-learn`, each of the commands (separated by commas) should be run on its own line and allowed to finish before the next command is run
  - Some modules need to be downloaded in a slightly different way: `conda install -c conda-forge flask`, `conda install -c conda-forge flask-caching`, `conda install -c conda-forge flask-bootstrap`
  - These lists are no exhaustive, if you do this then try to run it and get a `No module named...` error, then try `conda install module_name` and if that doesn't work, try `conda install -c conda-forge module_name`

- Now run the app!
- In the Anaconda prompt, type `python app.py`
- You know it will work when you get output including: `Running on http://127.0.0.1:5000/`
- To use the app, go to `http://127.0.0.1:5000/` in your browser (tested on Firefox and Safari, so if you have issues and are using a different browser, try one of these). The browser must have javascript enabled for the app to work, it should be by default.
- Once it is open in your browser, it should be fairly self explanatory to use
- When you are finished, kill the app by going to the Anaconda window and hitting control c, once it has died, the line the cursor is on will go back to a normal command prompt line

2 To get an updated version of the code

- Open the Git bash window (described above)
- `cd` into the `raman_analysis` folder
- In the command prompt, enter `git pull`
- If the app is running, in the anaconda window you should see `* Detected change in ...`. This means it can see the changes to the code and reload. Force refresh your browser so the changes are picked up there to
- If the app is not running, open the Anaconda prompt (described above) `cd` into the correct directory, and enter `python app.py` to run the app
- If the code changes the way it analyses the data, you need to delete the old analysis results, which have been saved for speed. From the Git command prompt , do: `cd uploads` then `del *pickle`, then go back into the main analysis directory by entering `cd ..`