Data Science Techniques (MAT 339)
Homework 6 - - 10 points

Submit a **hard copy** of your work at the beginning of class on Wednesday, February 25th. There is no **electronic submission** for this homework.

1. In this question you will read about the basics of a powerful automation tool known as `make` and install it on your system.
   (a) Read the `lec-makefiles.pptx` slides in the `lec05-automation-shell-scripts` folder of the course repo. **Optionally** also read the **Build Systems** section of this web page: https://missing.csail.mit.edu/2020/metaprogramming/
   (b) Install make in your system by following the instructions in `"Installing make for Git Bash.docx"` in the **Admin** folder of the course repo.
2. In this question you will create and run a very simple `makefile`.
   (a) Create a one-line text file named `name01.dat` that contains one name, e.g. "Bart".
   (b) Write a Python file named `hello.py` that reads in `name01.dat` and prints a greeting with that person's name, as in "Hello Bart!".
   (c) You may already know you can run Python files from the command line if your terminal has Python installed. Earlier in the semester you should have set up **Anaconda** to run within **Git Bash** enabling this capability. In the terminal, in the same folder as the two files you just created, type `python hello.py`. Include this terminal line and the output as your answer to this part.
   (d) Following the information in the slides you were to read above, create a simple `makefile` (you can name it `Makefile` or `Makefile.print-hello`) that runs `hello.py` and pipes its output (use `>`) to a file named `hello.txt`. The dependencies of `hello.txt` in the `makefile` should be `name01.dat` and `hello.py`. Include your `makefile` and successful execution of it as your answer to this part.
   (e) Without changing `name01.dat`, run the `makefile` again. Supply the output as your answer to this part.
   (f) Change the contents of `name01.dat` and run the `makefile`. Supply the output as your answer to this part.
3. THIS QUESTION IS OPTIONAL. In order to automate LaTeX file production using `make`, install **MiKTeX**, and optionally a LaTeX editor like **TeXstudio**. Links are available through a Google search.
4. Starting with the `covid-3.sh` shell script from the course repo (in the `lec05-automation-shell-scripts/ex-covid` folder), make the changes to meet the requirements described below. Feel free to make other changes (e.g. plot "Confirmed" instead of "Deaths") as long as they do not reduce the scope of the script.

(a) save the two csv files and the png file with the name of the state (e.g. `Maine.csv`, `Maine-trim.csv`, `Maine.png`) (Hint: use ${var-name} to refer to a variable name)

(b) pass some part of the date in to your script as an argument; this can be the year, the month, the day or some combination as long as your script will still be downloading some range of data files and achieving the same processing as before for the dates; for example, pass the month as a two digit code and obtain data for the 1st through the 10th day of that month

(c) come up with another alteration to the script of your own to make it more functional

Submit your script and displays of it running in the terminal and the png file it creates.

5. Creat your own shell script from scratch that utilizes several **terminal commands** along with **Python** and/or **sqlite3**. The level of complexity of your script should be close to that of the COVID one (the version from class). Choose something you are interested in and that has potential for expansion, because you will have the opportunity to expand it for the course final project. Answer these questions about your script:

   (a) What problem or issue does your script address?

   (b) List the terminal commands it uses:

   (c) What other programs does it use (e.g. Python, sqlite3, etc.) and what functionality do those programs add to the script?

   (d) How would you expand it if it was to become a final project?

   (e) Include your script and output of it running in the terminal plus any other output it creates if any (e.g. png file). If your script generates a large amount of output, only include a portion of that in your homework.