Data Science Techniques (MAT 339)
Homework 3 - - 10 Points - - SOLUTIONS

Submit a **hard copy** of your work at the beginning of class on Wednesday, February 4th. There is no **electronic submission** for this homework.

Since we will be printing, we will try to conserve ink this time around by not using screen shots of the terminal or folders in dark mode. You can either change the terminal and folder windows to light mode through options, or copy the command you issued and the contents of its output into one document you use to organize the homework assignment. This can be LaTeX, Word, markdown, text, etc. Just be sure it has a white or light background so as not to overuse printer toner.

1. We have seen the use of `>` to redirect output of a command to a file, e.g. `echo "Hello World!" > hello.txt`. Going in the opposite direction, `<` redirects the contents of a file to use as input for a command. Use both of these redirection operators with the `sort` command (even though technically you can get away with using only one here) to take an unsorted file and output a sorted version of it all from one line on the terminal. To save time and space, choose a small file of around a dozen lines. Copy and paste the text from your command and the outputted file into your homework file (e.g. LaTeX, txt, docx, etc.). Do not screen-shot any black screens.

   SOLUTION: Here is the beginning of content of the file `ct5.csv`:

```
$ head -7 ct5.csv
Admin2,Province_State,Confirmed,Deaths,Recovered
Hartford,Connecticut,61606,2069,False
Fairfield,Connecticut,70347,1899,False
New Haven,Connecticut,61489,1763,False
New London,Connecticut,16194,353,False
Middlesex,Connecticut,8895,312,False
Litchfield,Connecticut,9616,250,False
...
```

   Now sort it into a new file and look at that file:

```
$ sort < ct5.csv > ct5sorted.csv
$ cat ct5sorted.csv
Admin2,Province_State,Confirmed,Deaths,Recovered
Fairfield,Connecticut,70347,1899,False
Hartford,Connecticut,61606,2069,False
Litchfield,Connecticut,9616,250,False
Middlesex,Connecticut,8895,312,False
New Haven,Connecticut,61489,1763,False
New London,Connecticut,16194,353,False
```

2. For this question, you are encouraged to find a website that has a number of data files (csv, txt, etc.) that follow the same naming convention since you will be asked to download several at a time using `curl` or `wget`. Alternatively you are welcome to use the Marine Cadastre AIS site shown in class at `https://hub.marinecadastre.gov/pages/vesseltraffic`.

   (a) Use a single call to `curl` or `wget` to download multiple files at a time using the command's special features to download multiple files (i.e. do not list out each file name you are downloading). See the course slides and/or documentation for those programs if you need help on how to do this. Include your command **and** your directory listing (e.g. using `ls`) showing the downloaded files as your response to this question.

   SOLUTION: We will use some dummy files posted to the `github_basics_A` repo:

   ```
   curl -O https://raw.githubusercontent.com/ifrommer/github_basics_A/refs/
   heads/main/data/ct[1-5].csv
   ls -l
   ```

   ```
   $ ls -l
   total 5
   -rw-r--r-- 1 IFrommer 1049089 384 Feb  9 14:22 ct1.csv
   -rw-r--r-- 1 IFrommer 1049089 384 Feb  9 14:22 ct2.csv
   -rw-r--r-- 1 IFrommer 1049089 384 Feb  9 14:22 ct3.csv
   -rw-r--r-- 1 IFrommer 1049089 384 Feb  9 14:22 ct4.csv
   -rw-r--r-- 1 IFrommer 1049089 384 Feb  9 14:22 ct5.csv
   ```

   (b) Use a command to display a small portion of one of the files you just downloaded but pipe that command to the `tee` command to echo the output to the screen and also direct it to a file at the same time. (See page 69 of *The Linux Command Line* book for information on this commmand.) Include your command **and** the contents of the output file as your response to this question.

   SOLUTION: `head ct1.csv | tee ct1-head.csv`

   ```
   $ cat ct1-head.csv
   Admin2,Province_State,Confirmed,Deaths,Recovered
   Hartford,Connecticut,61606,2069,False
   Fairfield,Connecticut,70347,1899,False
   New Haven,Connecticut,61489,1763,False
   New London,Connecticut,16194,353,False
   Middlesex,Connecticut,8895,312,False
   ...
   ```

   (c) Using a single call to a terminal command, find the number of lines in each of the files you downloaded together above. Include your command **and** its output as your response to this question. Reminder - do not use screen shots of black terminal windows - see instructions above.

```
$ wc -l *
10 ct1.csv
10 ct2.csv
10 ct3.csv
10 ct4.csv
10 ct5.csv
50 total
```

3. Chapter 9 of the excellent book *Automate the Boring Stuff with Python* by Al Sweigart covers **regular expressions**. It is available for free online at:

   `https://automatetheboringstuff.com/3e/chapter9.html`.

   (a) Read this chapter from its beginning up through the section ”A Review of Regex Symbols” (a little more than halfway down the web page). Note that while the code shown is for use in Python, the regular expressions covered are universal.
   I read the chapter

   (b) Answer these questions at the end of the chapter (scroll to bottom of web page): 8, 10, 11, 12, 13, 14

   ```
   8.The | character signifies matching either, or between two groups.

   10.The + matches one or more. The * matches zero or more.

   11.The {3} matches exactly three instances of the preceding group.
    The {3,5} matches between three and five instances.

   12.The \d, \w, and \s shorthand character classes match a single
    digit, word, or space character, respectively.

   13.The \D, \W, and \S shorthand character classes match a single
    character that is not a digit, word, or space character, respectively.

   14.The .* performs a greedy match, and the .*? performs a non-
   greedy match.
   ```

4. Spend some time looking over the files you downloaded earlier in this homework using `curl` or `wget` to get a feel for their contents using shell commands covered in class. Then based on that and what you learned from the reading in the previous question, come up with two or more **regular expressions** to search for within the files using `grep`. Within your homework submission file, include the commands, what each is searching for in plain terms, and all or a portion of the results. See slide 12 of lec03 for an example.
   SOLUTION: 1) Based on class slides - find lines with two-word strings in them:

```
$ grep -E ’^\S+ \S+$’ ct1.csv
New Haven,Connecticut,61489,1763,False
New London,Connecticut,16194,353,False
```

2) Find a single-digit entry in the csv.

```
$ grep -E ’(^|,)[0-9](,|$)’ ct1.csv
Unassigned,Connecticut,892,9,False
```

5. (**Extra Credit - up to 2 points**) Create a few **aliases** to save you time typing long commands. Define these inside your `.bashrc` file. See slide 17 of lec03 and/or page 52 of *The Linux Command Line*. Paste a snippet of your `.bashrc` file showing the aliases and the terminal showing you using them into your overall homework file.
   Answers will vary.