STA 141C - HW 4 - Text Processing

Version 1.1

- See Canvas for due dates and grading rubric.
- Turn in a neatly typed 1-2 page report that answers the questions in clear English, using complete sentences.
- Use a format that Canvas can preview: pdf, html, Word are all fine. Raw Jupyter Notebook files and zip files do not work.
- Include submission scripts to run you code on the Gauss cluster. It's fine to develop your code locally, but you need to actually run it on Gauss.
- Code must run, and support your answers.

Overview

In this assignment we'll learn about line by line text processing using bash on a remote computer.

Data

I pulled the largest table from the usaspending database. I removed three columns that look like they're only used for indexing the website. I also went through every text column and replaced comma and newline characters with spaces.

So the data went from:

```
number,text
1,"Comma, city, oh yeah,,,"
2,"2 Newlines

inside, this one"
```

to:

```
number,text
1,"Comma  city  oh yeah   "
2,"2 Newlines  inside  this one"
```

In other words, you can now assume that newlines separate rows, and commas separate fields. This assumption greatly simplifies text processing with shell tools.

Bash

All of the following questions can be answered using combinations of the following commands:

Command	Summary
`unzip`	unzip a compressed `zip` file
`head`	output the first part of files
`tail`	output the last part of files
`tr`	translate or delete characters
`nl`	number lines of files
`cat`	concatenate files and print on the standard output
`cut`	remove sections from each lines of files (select column)
`wc`	print newline, word, and byte counts for each file
`grep`	print lines matching a pattern
`sort`	sort lines of text files
`uniq`	report or omit repeated lines

You're welcome to use more advanced bash commands including `sed`, `awk`, but do not use any programming languages.

Questions

1. Programming

Answer each of the following questions using a single sequence of bash commands piped together `|` and saved to a file using `>`. Evaluate them on the gauss cluster to get your answer. The questions in this section need only a one sentence answer, no need to elaborate.

1. What are the names and integer positions of the columns? Save as `colname_index.txt`. List the last three here, i.e. the first three will be:

   ```
   $ head colname_index.txt
    1  recipient_unique_id
    2  transaction_id
    3  action_date
   ```

2. How many characters are in the longest line? Save as `maxchars.txt`.

3. Find all the rows in the data where the string `bicycle` appears. Use a case insensitive match. Save as `bicycle.csv`. How many are there?

4. Find the set of unique funding agencies, meaning no duplicates. Save as `funding_agency_set.txt`. How many are there?

5. Find the description (`transaction_description`) and amount (`total_obligation`) of the 3 largest transactions. Save as `largest.csv`. List them here.

2. Reflecting

1. Explain in your own words the `sbatch` submission process. Start with moving code to the cluster, and finish with downloading a result.
2. Are the funding agency ID's in this assignment the same as in the first data set?
3. These two bash commands will produce the same output. Which is more efficient, and why?
    1. `cat file | sort | grep "pattern"`
    2. `cat file | grep "pattern" | sort`
4. Come up with your own question about this data set that you can answer with a single sequence of bash commands as you did with the questions in the first section. Run it and verify that it does what you expect. Show your code and explain what every step does.