Evan Batzer

STA 141C

HW4

1. **Programming**
2. *What are the names and integer positions of the columns? Save as colname\_index.txt List the last three here:*

59. awarding\_subtier\_agency\_abbreviation

60. f unding\_subtier\_agency\_abbreviation

61. business\_categories

1. *How many characters are in the longest line? Save as maxchars.txt.*

There are 7031 characters in the longest line.

1. *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?*

There are 32122 lines that match the string “bicycle” (case insensitive) in this database.

1. *Find the set of unique funding agencies, meaning no duplicates. Save as funding\_agency\_set.txt. How many are there?*

There are 781 unique funding agency IDs (780 if not counting blank entries).

1. *Find the description (transaction\_description) and amount (total\_obligation) of the 3 largest transactions. Save as largest.csv. List them here.*
2. Amount: $235584095641.00 (235.6 billion)
   1. Description: *NULL*
3. Amount: $235584095641.00 (235.6 billion)
   1. Description: *NULL*
4. Amount: $235584095641.00 (235.6 billion)
   1. Description*: NULL*
5. **Reflecting**

*1. Explain in your own words the sbatch submission process. Start with moving code to the cluster, and finish with downloading a result.*

The sbatch submission progress begins with downloading code through cloning a repository via github or direct movement of files through a transfer protocol such as SSH file transfer protocol (SFTP). The latter is better capable of handling larger files but lacks the convenience of a downloadable repository. After code has been moved to the cluster, initiation of a shell script using *sbatch* requests that the head node queue and assign processing tasks to worker nodes within the cluster. Depending on size of the task, general cluster demand, and task organization (such as number of nodes used in parallel tasks), the worker node completes the specified task, producing an output summary and associated products of the shell script. These are stored within the cluster and can then be accessed locally, either by pushing changes to a remote Github repository or directly downloading files from the remote machine to a local directory using SFTP.

*2. Are the funding agency ID’s in this assignment the same as in the first data set?*

Yes, the funding agency IDs do seem to be the same in this set as our first dataset, though are far more unique funding agency IDs in the new dataset (780 non-blank entries vs. 243 in the first). When comparing with the “agencies.csv” dataset from previous homeworks, it is clear that many IDs have shared agency descriptions: 7 – “Library of Congress”, 89 – “Peace Corps”, etc. I found it difficult to determine what field was used to generate agency IDs in the previous csv file and chose to extract unique “funding\_toptier\_agency\_name” as the “funding\_subtier\_agency\_name” seemed to produce too many values with identical IDs. As a result, my solution is probably a bit coarse.

*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

The latter will be more efficient, as the *grep “pattern”* command will reduce the size of the object being accessed during the sort. As the order of these operations can be performed interchangeably, first subsetting the object to rows which contain the pattern then sorting will be far faster than sorting over all entries (including those that do not contain the pattern) then searching for the pattern amongst all the pre-sorted entries.

*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*

For this question, I decided to ask where funding related to US borders is sent, given the recent political discussion over a wall on our southern border.

First, I cut the data to relevant columns - obligation, description, and recipient state ID

*unzip -p ${DATAFILE} | cut --fields ${AMT},${DESC},${STATEID} --delimiter , |*

Then, searched for the term "border" (case insensitive)

*grep --ignore-case border |*

Then, using awk, delimited fields by "," and sum over unique values of the array. Source from https://stackoverflow.com/questions/10286522/group-by-sum-from-shell

*awk -F '[,]' '{arr[$3]+=$1}*

*END {for (key in arr) printf("%s\t%s\n", key, arr[key])}'|*

Finally, I sorted over the first column and output a new text document

*sort -k1,1 > borderfunding.txt*

The resulting table shows some interesting patterns. The state of Virginia receives the most funding for “border” related expenses, followed by California and Texas. This may be due to government agencies located in Virginia and enforcement efforts along the southern border. However, it’s surprising that Tennessee and Colorado are on this list – why are these interior states given so much border related funding? Are there military bases or other governmental agencies based in these states? Or is there a different use of the term “border” in spending descriptions that might not be related to country borders?

**Table 1:** Top 10 state and associated “border” related funding amounts.