The Opioid Files

Tumble down the rabbit hole into Wonderland, where nothing is as it seems. Join Alice as she meets a clock-watching White Rabbit, a grinning Cheshire Cat, an eccentric Mad Hatter and a very cross Queen of Hearts.

In the land of Big Data, the usual techniques don't quite work and we must be prepared for surprises. The Opioid Files exercise is designed to have you experience some of the differences and workarounds.

Background

Over the summer of 2019, The Washington Post published a series of reports on the distribution of opioids in the US. Their data team published a portion of a database that tracks the path of every opioid pain pill, from manufacturer to pharmacy, in the United States between 2006 and 2012.

Details of how the data may be accessed are available here. This file represents the national data, it is about 7GB *compressed.* Uncompressed, it is about 75 GB in size. The downloaded file is available on the eecs Linux servers[1] as `/comp/119/arcos_all_washpost.tsv.gz.`

[1] Visit http://systems.eecs.tufts.edu/managing-your-password/ to obtain a login and password, then ssh linux.eecs.tufts.edu to access your linux account.

Problems[2]

1. [5 points] Find the column names in the Opioid dataset. The "normal" way would have been to `gunzip` the .gz file and run `head -1` on the result, but you likely don't have enough disk space. Conveniently, `zcat` can read the file and write the unzipped contents into stdout, which you can pipe into `head -1`.

2. [5 points] Find the number of rows in the Opioid dataset, using `zcat` output piped into `wc`. Make sure to take the header row into account.

3. [20 points] Find the number of rows for each year in the dataset. As above, we don't have enough space to unzip. So here's a potential strategy: Use the `shuf` command to extract, say, 5000 rows from the output of `zcat`. Find the proportion of rows for each year in this extract. You won't get an exact count but knowing the total number of rows, can you estimate the number of rows for each year?

Submit all three problems as a single PDF file into Gradescope. Be sure to include the commands you issued and also the programs you wrote. *Be sure to not paste screenshots into what you submit.*

[2] According to the rubric, this problem is worth 3%. A factor of 10 will be applied when totalling the score.