Brian Farrell

Big Data Exercise 2

1. [15 points] Find the column names in the Opioid dataset. The naive way is 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 can be piped into head -1.
    a. I used the following command to answer this question: gzcat arcos_all_washpost.tsv.gz | head -1
        i. Gzcat decompresses and outputs the content of the arcos_all_washpost.tsv.gsv file. I used gzcat because I ran this command on my mac laptop and gzact is the mac version of zcat. Head -1 outputs the header row of the data and resulted in the below output.
    b.
```
REPORTER_DEA_NO REPORTER_BUS_ACT       REPORTER_NAME
REPORTER_ADDL_CO_INFO   REPORTER_ADDRESS1
REPORTER_ADDRESS2      REPORTER_CITY   REPORTER_STATE
REPORTER_ZIP    REPORTER_COUNTY     BUYER_DEA_NO
BUYER_BUS_ACT   BUYER_NAME      BUYER_ADDL_CO_INFO
BUYER_ADDRESS1      BUYER_ADDRESS2  BUYER_CITY
BUYER_STATE     BUYER_ZIP       BUYER_COUNTY
TRANSACTION_CODE    DRUG_CODE       NDC_NO  DRUG_NAME
QUANTITY       UNIT    ACTION_INDICATOR    ORDER_FORM_NO
CORRECTION_NO   STRENGTH       TRANSACTION_DATE
CALC_BASE_WT_IN_GM  DOSAGE_UNIT     TRANSACTION_ID
```

Product_Name Ingredient_Name Measure MME_Conversion_Factor

Combined_Labeler_Name Revised_Company_Name Reporter_family dos_str

2. [15 points] Find the number of rows in the Opioid dataset by processing the zcat output, stripping the header row, and counting the remaining lines using wc.
    a. I ran the following command to find the number of rows minus the headers:
    b. zcat arcos_all_washpost.tsv.gz | awk 'NR > 1' | wc -l
    c. The resulting output of the above command was: 178,598,026.
    d. I used zcat to look into the unzipped file. Awk 'NR > 1' was used to skip the header row. And finally, wc -1 was used to count up the remaining number of rows in the file.
    e. I switched to my windows pc on this question so my commands would run faster, which is why I switched from gzcat to zcat.
3. [20 points] Find the names of all the drugs named in the dataset.
    a. I used the following python program to print all the unique drug names into a text file called "drug_names.txt"

```
import gzip



def get_drugs():

    drug_names = set()  # Set to avoid duplicates

    with gzip.open('arcos_all_washpost.tsv.gz', 'rt') as f:

        header = f.readline()
        drug_column_index = header.strip().split('\t').index('DRUG_NAME')  # Find the index for the drug name column
        for line in f:
            drug_names.add(line.strip().split('\t')[drug_column_index])  # Add drug names to the set
    return drug_names


def write_drugs(drug_names):
    with open('drug_names.txt', 'w') as f:
        for drug in (drug_names):  # for loop of the set the set
            f.write(drug + '\n')  # Write each drug name to the file


drug_names = get_drugs()
write_drugs(drug_names)
```

b. The output of the text file was:

   i. HYDROCODONE

   ii. OXYCODONE

4. [20 points] Estimate the number of rows for each year in the dataset. There may be enough space in the shell, but this exercise requires you to assume that that’s not the case. So here’s a potential strategy: Use the shuf command to extract, say, random 7,500 rows

from the output of zcat. Find the proportion of rows for each year in this extract. Assuming that the distribution of the random 7,500 rows is similar to the distribution in the whole file, estimate the number of rows for each year.

   a. I used the following python program to estimate the number of rows in each year.

```
import gzip
import random
from datetime import datetime


def estimate_years(file_path, sample_size=7500, total_rows=178598026):
    with gzip.open(file_path, 'rt') as f:
        header = f.readline()
        columns = header.strip().split('\t')
        transaction_date_index = columns.index('TRANSACTION_DATE')  # Find the column
index
        # Extract a random sample of rows
        sample = []
        for i, line in enumerate(f):
            if random.random() < sample_size / total_rows:
                fields = line.strip().split('\t')
```

```
if len(fields) > transaction_date_index:  # Ensure the column exists
    date_str = fields[transaction_date_index]  # Extract the date
    # print(f"Processing row {i + 1}: Date = {date_str}")  # Debug: Print the date being processed
    try:
        # Parse the date in "Month/Day/Year" format and extract the year
        date = datetime.strptime(date_str, '%m%d%Y')  #formatting the date
        year = date.year
        sample.append(year)
    except ValueError:
        # Skipping null dates
        print(f"Skipping invalid date: {date_str}")  # Debug: Print invalid dates
        continue
else:
    print(f"Skipping row {i + 1}: Missing TRANSACTION_DATE") #Used to debug rows
if len(sample) >= sample_size:
    break
    # Count the occurrences of each year in the sample
    year_counts = {}
    for year in sample:
        year_counts[year] = year_counts.get(year, 0) + 1

    # Step 3: Estimate the rows
    scaling_factor = total_rows / sample_size
    estimated_counts = {year: count * scaling_factor for year, count in year_counts.items()}

    # Print answer
    print("\nYear\tEstimated Rows")
    for year, count in sorted(estimated_counts.items()):
        print(f"{year}\t{count:.0f}")

if __name__ == "__main__":
    file_path = 'arcos_all_washpost.tsv.gz'  # Path to the dataset (I had trouble at first so I had to navigate straight to the directory)
    estimate_years(file_path)
```

b. The output of this file was:

i.	Year	Estimated Rows
ii.	2006	19741035
iii.	2007	23670192
iv.	2008	24432210
v.	2009	26003873
vi.	2010	27194526
vii.	2011	28623310
viii.	2012	28932880

c. I averaged the estimated rows and got 25,514,004 rows per year according to this sample of 7500 entries.

d. I attempted this first in shell, but kept running into issues. I have more experience writing python scripts, so I switched to it because I felt more comfortable debugging python.

5. [15 points] Obtain the count of rows for June 2012 by extracting all such rows from arcos_all_washpost.tsv.gz and running wc on the extracted rows.

    a. I ran the command: zgrep -P '\t06[0-9]{2}2012\t' arcos_all_washpost.tsv.gz | wc -l

        i. The result was: 2,323,389

6. [15 points] Estimate the count of rows for June 2012 based on answers to questions 1, 2, and 4 and compare that count with your findings from Q5.

    a. There are around 25,514,004 rows per year according to my answer to question four. Dividing this by 12, to get the monthly average, equals 2,126,167 rows for June 2012. This is around 200,000 off the actual number of rows in June 2012.

This is a pretty good estimate considering the data used was only a 7500 random sample of the 178 million rows.