

1. 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.

Ans:



```
samay@samay: ~  
vm-linux01(spashi01)30: ls -lh  
total 6.5G  
-rw-r--r--. 1 jsingh11 tal19 6.5G Jan 24 2022 arcos_all_washpost.tsv.gz  
-rwxr-xr-x. 1 jsingh11 tal19 1.5K Jan 26 2022 opioid_2006.tsv  
drwxr-xr-x. 5 jsingh11 tal19 4.0K Sep 6 13:53 public_html  
vm-linux01(spashi01)31: zcat -fq arcos_all_washpost.tsv.gz | head -1  
REPORTER_DEA_NO REPORTER_BUS_ACT REPORTER_NAME REPORTER_ADDL_CO_INFO REPORTER_ADDRESS1 REPORTER_ADDRESS2 REPORTER_CITY REPORTER_STATE REPORTER_ZIP REPORTER_CO  
UNTY 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_GMD  
OSAGE_UNIT TRANSACTION_ID Product_Name Ingredient_Name Measure MME_Conversion_Factor Combined_Labeler_Name Revised_Company_Name Reporter_family dos_str  
vm-linux01(spashi01)32: 
```

Figure 1: Columns printed in the terminal using zcat and head command

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

Ans:

Total Number of rows (including the column names): 178598027

Total Number of rows (excluding the column names): 178598026

```

null NORTH ARLINGTON NJ 7031 BERGEN S 9143 00406051201 OXYCODON
E 6.0null null 074287095 null 0000 03212008 2.6895 600.0 8
03071034 OXYCODONE HCL/ACETAMINOPHEN 5MG/325M OXYCODONE HYDROCHLORIDE TAB 1.5
SpecGx LLC Mallinckrodt Cardinal Health 5.0
PC0003044 DISTRIBUTOR CARDINAL HEALTH 110, LLC null 6012 EAST MOLLOY RD
null SYRACUSE NY 13211 ONONDAGA BC8045759 CHAIN PHARMACY
NEW JERSEY CVS PHARMACY, L.L.C. DBA: CVS/PHARMACY # 03136 440 BELLEVILLE TPKE.
null NORTH ARLINGTON NJ 7031 BERGEN S 9143 00406051201 OXYCODON
E 20.0 null null 074287113 null 0000 05302008 8.965 2
000.0 805097863 OXYCODONE HCL/ACETAMINOPHEN 5MG/325M OXYCODONE HYDROCHLORIDE TAB
1.5 SpecGx LLC Mallinckrodt Cardinal Health 5.0
PC0003044 DISTRIBUTOR CARDINAL HEALTH 110, LLC null 6012 EAST MOLLOY RD
null SYRACUSE NY 13211 ONONDAGA BC8045759 CHAIN PHARMACY
NEW JERSEY CVS PHARMACY, L.L.C. DBA: CVS/PHARMACY # 03136 440 BELLEVILLE TPKE.
null NORTH ARLINGTON NJ 7031 BERGEN S 9143 00591093301 OXYCODON
E 4.0null null 074287113 null 0000 05302008 2.6895 400.0 8
05097864 OXYCODONE.HCL/APAP 7.5MG/325MG TABS OXYCODONE HYDROCHLORIDE TAB 1.5
Actavis Pharma, Inc. Allergan, I^C
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$ zcat arcos_all_washpost.tsv.gz | wc
-l
178598027
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$
```

Figure 2: Total rows including the names of the columns printed using zcat and wc.

3. 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?

Ans:

The answer to the question is shown in the screenshot below.

```
samay@samay: ~/Desktop/MS/Fall-2022/CS-119/Assignment/2
samay@samay: ~/Desktop/MS/Fall-2022/CS-119/Assignment/2 187x46
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$ zcat arcoss_all_washpost.tsv.gz | shuf --head-count=5000 > ques3.csv
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$ ls -lh
total 6.5G
-rw-rw-r-- 1 samay samay 6.5G Sep 14 23:58 arcoss_all_washpost.tsv.gz
-rw-rw-r-- 1 samay samay 6.5M Sep 15 12:28 code.ipynb
-rw-rw-r-- 1 samay samay 904K Sep 13 16:27 cs-119-2022h2_quiz-2.pdf
-rw-rw-r-- 1 samay samay 2.2M Sep 15 14:24 ques3.csv
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$ code . &
[1] 32339
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$ clear

[1]+  Done                  code .
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$ ls -lh
total 6.5G
-rw-rw-r-- 1 samay samay 6.5G Sep 14 23:58 arcoss_all_washpost.tsv.gz
-rw-rw-r-- 1 samay samay 6.5M Sep 15 12:28 code.ipynb
-rw-rw-r-- 1 samay samay 904K Sep 13 16:27 cs-119-2022h2_quiz-2.pdf
-rw-rw-r-- 1 samay samay 2.2M Sep 15 14:24 ques3.csv
samay@samay:~/Desktop/MS/Fall-2022/CS-119/Assignment/2$ cat ques3.csv
```

Figure 3: Using zcat and shuf command to extract 5000 rows from the whole dataset and saving it in a separate csv file.

```

import pandas as pd
df = pd.read_csv('5000_rows.csv', sep='\t', error_bad_lines=False)
✓ 0.1s

year_dict = {}
for i in range(df.shape[0]):
    if str(df.iloc[i, 30])[-4:] not in year_dict:
        year_dict[str(df.iloc[i, 30])[-4:]] = 0
    else:
        year_dict[str(df.iloc[i, 30])[-4:]] += 1
year_dict
✓ 0.1s

{'2006': 541,
 '2007': 637,
 '2012': 808,
 '2011': 867,
 '2010': 761,
 '2009': 704,
 '2008': 674}

part_rows = 5000
total_rows = 178598826
rows_2006 = int((float(year_dict['2006'])/part_rows)*total_rows)
rows_2007 = int((float(year_dict['2007'])/part_rows)*total_rows)
rows_2008 = int((float(year_dict['2008'])/part_rows)*total_rows)
rows_2009 = int((float(year_dict['2009'])/part_rows)*total_rows)
rows_2010 = int((float(year_dict['2010'])/part_rows)*total_rows)
rows_2011 = int((float(year_dict['2011'])/part_rows)*total_rows)
rows_2012 = int((float(year_dict['2012'])/part_rows)*total_rows)
print("[CALCULATED]. Number of rows for 2006 : ", rows_2006)
print("[CALCULATED]. Number of rows for 2007 : ", rows_2007)
print("[CALCULATED]. Number of rows for 2008 : ", rows_2008)
print("[CALCULATED]. Number of rows for 2009 : ", rows_2009)
print("[CALCULATED]. Number of rows for 2010 : ", rows_2010)
print("[CALCULATED]. Number of rows for 2011 : ", rows_2011)
print("[CALCULATED]. Number of rows for 2012 : ", rows_2012)
print("\n[CALCULATED]. Total number of rows : ", rows_2006 + rows_2007 + rows_2008 + rows_2009 + rows_2010 + rows_2011 + rows_2012)
print("[ACTUAL]. Total number of rows : ", total_rows)
✓ 0.4s

[CALCULATED]. Number of rows for 2006 : 19324306
[CALCULATED]. Number of rows for 2007 : 22753388
[CALCULATED]. Number of rows for 2008 : 24075013
[CALCULATED]. Number of rows for 2009 : 25146602
[CALCULATED]. Number of rows for 2010 : 27182619
[CALCULATED]. Number of rows for 2011 : 30968897
[CALCULATED]. Number of rows for 2012 : 28861441

[CALCULATED]. Total number of rows : 178312266
[ACTUAL]. Total number of rows : 178598826

```

Figure 4: Code to count the number of rows for each year using pandas library.

NOTE: The 5000 rows were saved in the file named ques3.csv but the file was re-named later to 5000_rows.csv