Problem set 5 - CSCI E-88b Spring 2020

Due date: Friday, May 15 2020, 10pm EST.

Please write down the last 5 digits of your Harvard ID: 70196

Please write down the URL of your Renku project: https://renkulab.io/projects/stephen.t.knapp14/problem5 (https://renkulab.io/projects/stephen.t.knapp14/problem5)

Please indicate who you may have worked with on this problem and which section you worked with someone on. If you did not work with anyone then you can leave this blank: xxxxx

This problem set corresponds to 1/6 of your final grade. It is graded out of 10.

Question 5.0: 0pt

Question 5.1: 2pts

Question 5.2: 2pts

Question 5.3: 2pts

Question 5.4: 2pts

Question 5.5: 2pts

Question 5.0: Keep it private!

If your problem set is not private, your grade will be 0/10.

Question: Please confirm that your project is private. You can edit the markdown cell from `[ ]` to `[x]` to check a box.

Answer (5.0):

- [X] Yes, my project is private.
- [ ] No, my project is not private.

Objective

CMS (https://www.cms.gov/ (https://www.cms.gov/)) provides publicly available data (https://www.cms.gov/Research-Statistics-Data-and-Systems/ (https://www.cms.gov/Research-Statistics-Data-and-Systems/) Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/index.html):

> CMS has released a series of publicly available data files that summarize the utilization and payments for procedures, services, and prescription drugs provided to Medicare beneficiaries by specific inpatient and outpatient hospitals, physicians, and other suppliers. These Medicare Provider Utilization and Payment Data files include information for common inpatient and outpatient services, all physician and other supplier procedures and services, and all Part D prescriptions. Providers determine what they will charge for items, services, and procedures provided to patients and these charges are the amount that providers bill for an item, service, or procedure.

We use the 2015 detailed data: http://download.cms.gov/Research-Statistics-Data-and-Systems/ (http://download.cms.gov/Research-Statistics-Data-and-Systems/) Statistics-Trends-and-Reports/Medicare-Provider-Charge-Data/Downloads/PartD_Prescriber*PUF* NPI_DRUG_15.zip. It unzips to a tab-delimited format file `PartD_Prescriber_PUF_NPI_Drug_15.txt` that is about 3GB and that we also added as a table in a sqlite database under git lsf.

In the problem set, we want to get some basic information about this Medicare data without loading it in memory (as we would typically do in R or Python).

R or Python?

You are free to use R or Python. To use Python, change the kernel you are using (top right button in this Jupyter notebook, it is set to `R` now).

Question 5.1

We extracted a random sample of observations from the Medicare dataset to get a better sense of the data. It is available at `data/medicare/sample_medicare.csv` . We want to know how many observations from the full dataset were sampled, but without loading the file in memory (say, with R or Python). Use `awk` to find the number of observations we sampled. (Hint: you can use a pager such as `less` to view data page by page, in particular whether there is a header with row names).

Answer (5.1):

INSERT YOUR CODE AND COMMENTS HERE

```
1) Check for header - run in the terminal:
    $ less ./data/medicare/sample_medicare.csv
        yes there is a header, so subtract 1 from the next command to get the numbe
r of samples

2) Find number of samples - run in the terminal:
    $ awk 'END { print NR -1 }' ./data/medicare/sample_medicare.csv
        output: 100000
        Breakdown of the command:
            NR -1: is the total number of rows "scanned" by the awk command and 1 i
s deducted to account for the header row
            END { print: is telling the command to print the above once it finishes
scanning the file
            ./data/medicare/sample_medicare.csv: is the location of the file to be
 scanned
```

There are 100,000 samples in the .csv file

Question 5.2

Use R or Python to calculate the mean drug cost (from variable `total_drug_cost`) per state (from variable `nppes_provider_state`) on the sample Medicare dataset `data/medicare/sample_medicare.csv`. You can remove missing values if any. Sort your answer by alphabetical order of `nppes_provider_state` and show the first 15 entries in your answer.

Answer (5.2):

In [20]:
```
#load R libaries for working with databases
library(dplyr)
library(readr)
library(DBI)
library(RSQLite)

#read in csv file
df <- read.csv("../data/medicare/sample_medicare.csv")

#apply the following commands to the dataframe
df %>%

    # group the data by state
    group_by(nppes_provider_state) %>%

    # calculate summary statistic by group
    summarize(

        # take the mean and remove NA values
        Mean_Total_Drug_Cost = mean(total_drug_cost, na.rm = TRUE)
      ) %>%

        # now only display the first 15 values
        head(15)
```

A tibble: 15 × 2

nppes_provider_state	Mean_Total_Drug_Cost
<fct>	<dbl>
AE	142.565
AK	2084.695
AL	3308.722
AR	3886.480
AZ	3756.158
CA	3996.927
CO	4358.985
CT	4877.835
DC	3457.887
DE	3249.646
FL	3783.652
GA	3484.670
GU	1537.748
HI	3897.899
IA	4739.704

Question 5.3

The full Medicare dataset is available as a `medicare` table in a sqlite database in file `data/medicare/medicare.sqlite` under git lfs. Use SQL or R or Python to connect to the sqlite database and provide the list of tables in the database.

Answer (5.3):

In [23]:
```
## INSERT YOUR CODE AND COMMENTS HERE

# using the terminal enter $ git lfs fetch
# this gets the large file system parameters for the project from github

# using the terminal enter $ git lfs checkout
# this allows me to access and make changes to the large file system objects

#read the database
conn <- dbConnect(RSQLite::SQLite(), "../data/medicare/medicare.sqlite")

#list tables in the
dbListTables(conn)
```

'drug_categories' 'medicare' 'sqlite_stat1' 'sqlite_stat4'

Question 5.4

Use a SQL query (in SQL or via R or Python SQL tools) to obtain the number of observations in the `medicare` table.

Answer (5.4):

In [24]:
```
#dbGetQuery(db, "FROM medicare")

# create table reference
medicare.table <- tbl(conn, "medicare")

# count number of observations
tally(medicare.table)
```

```
# Source:   lazy query [?? x 1]
# Database: sqlite 3.29.0 [/work/problem5/data/medicare/medicare.sqlite]
         n
     <int>
1 24524894
```

As seen from the output above, there are 24,524,894 observations. It makes sense that loading this as a data frame in R would take a lot of memory and time to process.

Question 5.5

Use a SQL query (in SQL or via R or Python SQL tools) to answer Question 5.2 again, but on the whole dataset.

Answer (5.5):

In [28]:
```
dbGetQuery(
    conn,
    "SELECT nppes_provider_state AS 'State',
    AVG(total_drug_cost) AS 'Mean_Total_Drug_Cost' FROM medicare
                GROUP BY nppes_provider_state
                ORDER BY nppes_provider_state
                LIMIT 15")

# SELECT nppes_provider_state AS 'State': tells the function to use the data i
n column nppes_provider_state and create a column called 'State'
# AVG(total_drug_cost) AS 'Mean_Total_Drug_Cost': tells the function to take t
he means of the total_drug_cost column
# FROM medicare: specifies which table to find the data in inside the database
# GROUP BY nppes_provider_state: take means by groups in the 'State' column sp
ecified earlier
# ORDER BY nppes_provider_state: order the 'State' column alphabetically
# LIMIT 15: only show the first 15 lines of the query
```

A data.frame: 15 × 2

State	Mean_Total_Drug_Cost
<chr>	<dbl>
AA	498.4405
AE	2785.6377
AK	2876.4162
AL	4721.6955
AP	679.9736
AR	3629.1984
AS	133.1600
AZ	4073.9198
CA	4636.7208
CO	3666.7932
CT	4833.8620
DC	5480.7810
DE	4971.0013
FL	4770.0165
GA	4617.4674

The means vary significantly from 5.2 because the sample size in the .csv file was less than 0.4% of the population.

Self-assessment (not graded)

Questions

1. What do you think I was hoping for you to learn through this homework?
2. Did you find anything particularly challenging?

Your answer

1. Basic SQL commands and how to use them with R and Python.
2. I have never worked with a SQL database before so learning and applying some of the basics took some time.

Submitting your work

To submit the problem set, export the notebook to HTML and upload the file to Canvas.

`[ File -> Export Notebook As... -> Export Notebook to HTML ]`