Physician Supplement Payments for all CMS Program Years

CUNY MSDA - DATA607 - Project 2_b

Completed by: Duubar Villalobos Jimenez mydvtech@gmail.com

March 12, 2017

Figure 1:

The goal of this assignment is to give you practice in preparing different datasets for downstream analysis work.

Your task is to:

(1) Choose any **three** of the **"wide" datasets** identified in the Week 5 Discussion items. (You may use your own dataset; please don't use my Sample Post dataset, since that was used in your Week 5 assignment!)

For each of the three chosen datasets:

- Create a **.CSV** file (or optionally, a **MySQL** database!) that includes all of the information included in the dataset. You're encouraged to use a "wide" structure similar to how the information appears in the discussion item, so that you can practice tidying and transformations as described below.
- Read the information from your **.CSV** file into **R**, and use **tidyr** and **dplyr** as needed to tidy and transform your data. [Most of your grade will be based on this step!]
- Perform the analysis requested in the discussion item.
- Your code should be in an R Markdown file, posted to rpubs.com, and should include narrative descriptions of your data cleanup work, analysis, and conclusions.

(2) Please include in your homework submission, for each of the three chosen datasets:

The **URL** to the **.Rmd** file in your **GitHub** repository, and The URL for your **rpubs.com** web page.

PROCEDURE

Library definitions

```r
library(knitr)
library(stringr)
library(tidyr)
library(dplyr)
library(RMySQL)
library(zipcode)
library(ggplot2)
```

Centers for Medicare & Medicaid Services Open Payment Data.

Dataset url location: https://openpaymentsdata.cms.gov/

I will be exploring the Physician Supplement File for all Program Years.

This is a supplementary file that displays a list of physicians indicated as recipients of payments reported in Open Payments. Each record includes the physicians demographic information, specialties, and license information, as well as a unique identification number (Physician Profile ID) that can be used to search for a specific physician in the general, research, and physician ownership files.

This is a complete Open Payments Dataset from the program from The Centers for Medicare & Medicaid Services Open Payment Data Report site that includes data about a federal government website managed by the Centers for Medicare & Medicaid Services.

Last Updated:

This is a complete set of all data from the Program.

Version: 1.0

Date: January 2017.

Data Provided by:

Centers for Medicare & Medicaid Services (CMS).

Dataset Owner:

Centers for Medicare & Medicaid Services (CMS) and the Government of the United States of America.

Dictionary

This dataset does not seem to have a dictionary. The download link is for a **.zip** file containing the desired data. However in the zip file there's a **.txt** file describing the following:

Filename: OP_PH_PRFL_SPLMTL_README_P01172017.txt

1. Physician Profile Supplement File

The Physician Profile Supplement file contains information about physicians who were indicated as recipients of payments, other transfers of value, or ownership and investment interest in payment records as well as physicians identified as principal investigators associated with research payment records published by Open Payments.

However, this file contains only physicians who are included in at least one published payment record. The criteria used by the Centers for Medicare and Medicaid Services (CMS) to determine which payment records are eligible for publication is available in the Open Payments Data Dictionary and Methodology document. This document can be found on the Resources page of the Open Payments website (https://www.cms.gov/OpenPayments/About/Resources.html). The Data Dictionary and Methodology document also includes information on the data collection and reporting methodology, data fields included in the files, and any notes or special considerations that users should be aware of.

2. Considerations for using the CSV File

Microsoft Excel removes leading zeroes from data fields in comma-separated values (CSV) files. Certain fields in this data set may have leading zeroes. These zeroes will be missing when viewing the information within Microsoft Excel.

Additionally, the latest versions of Microsoft Excel cannot display data sets with more than 1,048,576 rows. This CSV file may exceed that limit. Displaying the data in its entirety may require the use of a spreadsheet program capable of handling very large numbers of records.

3. Details about the OP_PH_PRFL_SPLMTL_P01172017.zip File

This compressed (.zip) file contains one (1) comma-separated values (.csv) format file that uses commas as delimiters and one (1) README.txt file. A description of the CSV file is provided below.

- *OP_PH_PRFL_SPLMTL_P01172017.csv:*

This supplementary file displays information on all of the physicians indicated as covered recipients of payments and/or physician principal investigators associated with payments in records published by Open Payments. Each record includes the physician's demographic information, specialties, and license state, as well as the unique identification number (Physician Profile ID) assigned by Open Payments for each physician. The Physician Profile ID can be used to search Open Payments data to find payments made to or associated with that specific physician.

The physician profile information included in the data sets is submitted by the reporting entity. In contrast, the physician information included in the supplementary file can be derived from different sources including the National Plan and Provider Enumeration System (NPPES) and the Provider Enrollment, Chain and Ownership System (PECOS). As a result, the data in these sources may differ slightly. When searching for physicians using the Open Payments search tool on https://openpaymentsdata.cms.gov, use the physician profile information as listed in the supplementary file.

For simplicity reasons, I will read the raw data directly from the source.

URL and Raw data name and location definitions:

```r
url <- "http://download.cms.gov/openpayments/"
zipfile <- "PHPRFL_P011717.ZIP"
csvfile <- "OP_PH_PRFL_SPLMTL_P01172017.csv"
```

Local MySQL definitions:

```
# Need to change to correct local root password for the local database
myLocalPassword <- 'pswrd'
myLocalUser <- 'root'
myLocalHost <- 'localhost'
myLocalMySQLSchema <- 'cms_OpenPaymentData'
myLocalTableName <- 'tbl_OpenPaymentData'
```

(1) Read information from .CSV file into R.

From the above **.zip** file, I will choose **OP_PH_PRFL_SPLMTL_P01172017.csv** which includes the latest information recorded, I am just keeping in mind that this file contains all the records for all the physicians that received payments from CMS.

Read .csv from url by employing read.csv()

For this project I will experiment a few things as follows:

I will do a combined data management procedure by employing **MySQL**; that is:

a) I will create a procedure that will create a connection to **MySQL**.
b) Once the connection is **"ON"**, I will check to see if a database named **cms_OpenPaymentData** exist.

- If the database exist, do nothing.
- If the database does NOT exist, create one.

c) Once the previous step is performed, I will check to see if a table named **tbl_OpenPaymentData** exist.

- If the table exist, read the information from it.
- If the table does NOT exist, then do as follows:
  - Download the **.zip** file from url in a **temp** file.
  - Extract the **OP_PH_PRFL_SPLMTL_P01172017.csv** file containing the required information.
  - Write all the information contained in the **.csv** file into the **tbl_OpenPaymentData** table from the **cms_OpenPaymentData** MySQL scheme.
  - Delete the temp **.zip** file.

After that I will import all the data into a data frame.

This might take some time since the file contains a large number of records, but in a second or multiple iterations will save a ton of time. Also another advantage is that the information will stay local and it will be straight forward to update if needed.

Function to download .zip file, unzip and extract information from .csv

```
downloadZip <- function(myurl, myzipfile, mycsvfile){
  temp <- tempfile()
  url <- paste(myurl, myzipfile, sep="")
  download.file(url, temp)
  my.file <- unzip(myzipfile, files = mycsvfile)
  my.data <- read.csv(my.file, header=TRUE, sep=",", stringsAsFactors=FALSE)
# Deleting downloaded file
unlink(temp)
# Returning data
  return(my.data)
}
```

MySQL Procedure to Read or Write tables

```
# Establish MySQLconnection
mydbconnection <- dbConnect(MySQL(),
                  user = myLocalUser,
                  password = myLocalPassword,
                  host = myLocalHost)
# creating a database if it doesn't exist by employing RMySQL() in R
MySQLcode <- paste0("CREATE SCHEMA IF NOT EXISTS ",myLocalMySQLSchema,";", sep="")
dbSendQuery(mydbconnection, MySQLcode)

# Table exists?
mydbconnection <- dbConnect(MySQL(),
                  user = myLocalUser,
                  password = myLocalPassword,
                  host = myLocalHost,
                  dbname = myLocalMySQLSchema)
# Check to see if table data exist.
myLocalTableName <- tolower(myLocalTableName)
if (dbExistsTable(mydbconnection, name = myLocalTableName)  == FALSE){
# If the table does not exist, download .zip and write .csv file into MySQL
my.data <- downloadZip(myurl= url, myzipfile= zipfile,  mycsvfile= csvfile)
# Then Write the table in MySQL
dbWriteTable(mydbconnection, name= myLocalTableName , value= my.data)
} else {
# Read the data from the local table
 my.data <- dbReadTable(mydbconnection, name = myLocalTableName)
}

# Closing connection with local Schema
dbDisconnect(mydbconnection)
```

Imported file structure display

```
## 'data.frame':    814447 obs. of  27 variables:
##  $ Physician_Profile_ID                  : num  29708 29709 29719 29720 29721 ...
##  $ Physician_Profile_First_Name          : chr  "GREGORY" "JOHN" "GEORGE" "MARC" ...
##  $ Physician_Profile_Middle_Name         : chr  "" "D" "A" "S" ...
##  $ Physician_Profile_Last_Name           : chr  "SENSENICH" "BALUCH" "BLESSIOS" "FISK" ...
##  $ Physician_Profile_Suffix              : chr  "" "" "" "" ...
##  $ Physician_Profile_Alternate_First_Name : chr  "GREGORY" "" "" "MARC" ...
##  $ Physician_Profile_Alternate_Middle_Name: chr  "W" "" "" "SASLOW" ...
##  $ Physician_Profile_Alternate_Last_Name  : chr  "SENSENICH" "" "" "FISK" ...
##  $ Physician_Profile_Alternate_Suffix     : chr  "" "" "" "" ...
##  $ Physician_Profile_Address_Line_1       : chr  "861 FAIRWAY DR" "500 E MAIN ST" "550 ORCHARD PARK
##  $ Physician_Profile_Address_Line_2       : chr  "" "STE 220" "A103" "" ...
##  $ Physician_Profile_City                 : chr  "CHILLICOTHE" "COLUMBUS" "WEST SENECA" "WEST ORANGE
##  $ Physician_Profile_State                : chr  "MO" "OH" "NY" "NJ" ...
##  $ Physician_Profile_Zipcode              : chr  "64601-3673" "43215" "14224-2646" "07052-2724" ...
##  $ Physician_Profile_Country_Name         : chr  "UNITED STATES" "UNITED STATES" "UNITED STATES" "UNI
##  $ Physician_Profile_Province_Name        : chr  "" "" "" "" ...
##  $ Physician_Profile_Primary_Specialty    : chr  "Allopathic & Osteopathic Physicians|Family Medicin
##  $ Physician_Profile_OPS_Taxonomy_1       : chr  "207Q00000X" "208800000X" "208600000X" "207RC0000X"
##  $ Physician_Profile_OPS_Taxonomy_2       : chr  "" "" "204F00000X" "" ...
##  $ Physician_Profile_OPS_Taxonomy_3       : chr  "" "" "" "" ...
##  $ Physician_Profile_OPS_Taxonomy_4       : chr  "" "" "" "" ...
##  $ Physician_Profile_OPS_Taxonomy_5       : chr  "" "" "" "" ...
##  $ Physician_Profile_License_State_Code_1 : chr  "MO" "OH" "NY" "NJ" ...
##  $ Physician_Profile_License_State_Code_2 : chr  "" "" "PA" "" ...
##  $ Physician_Profile_License_State_Code_3 : chr  "" "" "" "" ...
##  $ Physician_Profile_License_State_Code_4 : chr  "" "" "" "" ...
##  $ Physician_Profile_License_State_Code_5 : chr  "" "" "" "" ...
```

In summary, this data frame contains 814447 independent observations with 27 recognizable variables.

Data transformation

Now that I have the data frame I will transform it in order to create some possible outcomes from the given information; for this, I will **subset()** by excluding small portion of it.

Excluding Information:

Excluding Alternate Name information:

This procedure will exclude the repeating of the alternate information columns for the physician's name by performing a subset with the required exclusion.

```
my.new.data <-  my.data %>% subset(select=-(Physician_Profile_Alternate_First_Name:Physician_Profile_Al
```

Physician_Profile_ID	Physician_Profile_First_Name	Physician_Profile_Middle_Name	Physician_Profile_Last_Na
29708	GREGORY		SENSENICH
29709	JOHN	D	BALUCH
29719	GEORGE	A	BLESSIOS
29720	MARC	S	FISK
29721	BENJAMIN	MICHAEL JULIAN	THOMPSON
29722	TAISSA	N	CHERRY

Excluding Street Address information:

This procedure will exclude the street address information columns **Physician_Profile_Address_Line_1** and **Physician_Profile_Address_Line_2** for the physician's name by performing a subset with the required exclusion.

```
my.new.data <-  my.new.data %>% subset(select=-(Physician_Profile_Address_Line_1:Physician_Profile_Addr
```

Physician_Profile_ID	Physician_Profile_First_Name	Physician_Profile_Middle_Name	Physician_Profile_Last_Na
29708	GREGORY		SENSENICH
29709	JOHN	D	BALUCH
29719	GEORGE	A	BLESSIOS
29720	MARC	S	FISK

Physician_Profile_ID	Physician_Profile_First_Name	Physician_Profile_Middle_Name	Physician_Profile_Last_Na
29721	BENJAMIN	MICHAEL JULIAN	THOMPSON
29722	TAISSA	N	CHERRY

Working physician's name:

For this, I will be combining the columns related to the physician's name information into one single column by employing the function **unite()** from the **tidyr** library.

This procedure will exclude the repeating of the alternate information columns for the physician's name.

```
my.new.data <-  my.new.data %>%
                unite(Physician, c(Physician_Profile_First_Name, Physician_Profile_Middle_Name, Physicia
```

Physician_Profile_ID	Physician	Physician_Profile_City	Physician_Profile_Sta
29708	GREGORY SENSENICH	CHILLICOTHE	MO
29709	JOHN D BALUCH	COLUMBUS	OH
29719	GEORGE A BLESSIOS	WEST SENECA	NY
29720	MARC S FISK	WEST ORANGE	NJ
29721	BENJAMIN MICHAEL JULIAN THOMPSON	EXETER	NH
29722	TAISSA N CHERRY	SAN FRANCISCO	CA

Zip Codes:

For this, I will be transforming the diverse representations for the zip codes into a 5 digit length zip code by employing **clean.zipcodes()** from the **zipcodes** library. That is, I will attempts to detect and clean up suspected ZIP codes. Will strip "ZIP+4" suffixes to match format of zipcode data.frame. Restores leading zeros, converts invalid entries to NAs, and returns character vector. Note that this function does not attempt to find a matching ZIP code in the database, but rather examines formatting alone.

```
my.new.data$Physician_Profile_Zipcode <-clean.zipcodes(my.new.data$Physician_Profile_Zipcode)
```

Physician_Profile_ID	Physician	Physician_Profile_City	Physician_Profile_Sta
29708	GREGORY SENSENICH	CHILLICOTHE	MO
29709	JOHN D BALUCH	COLUMBUS	OH
29719	GEORGE A BLESSIOS	WEST SENECA	NY
29720	MARC S FISK	WEST ORANGE	NJ
29721	BENJAMIN MICHAEL JULIAN THOMPSON	EXETER	NH
29722	TAISSA N CHERRY	SAN FRANCISCO	CA

Specialties:

For this I tried splitting the rows y employing **separate_rows()** from the **tidyr** library but the limited resources on my computer returned the following Error: *"Error: cannot allocate vector of size 332.9 Mb"*; hence I have to skip it and go look for more power.

```
# my.new.data$Physician_Profile_Primary_Specialty <- my.new.data %>%
#                                                    separate_rows(Physician_Profile_Primary_Specialty,
#View(my.new.data)
```

Taxonomies:

For this I will group all the Taxonomies under one **Taxonomy** Variable by employing the **unite()** function from the **tidyr** library.

```
my.new.data <-  my.new.data %>%
                unite(Taxonomy, c(Physician_Profile_OPS_Taxonomy_1, Physician_Profile_OPS_Taxonomy_2, Ph
```

Physician_Profile_ID	Physician	Physician_Profile_City	Physician_Profile_Sta
29708	GREGORY SENSENICH	CHILLICOTHE	MO
29709	JOHN D BALUCH	COLUMBUS	OH
29719	GEORGE A BLESSIOS	WEST SENECA	NY
29720	MARC S FISK	WEST ORANGE	NJ
29721	BENJAMIN MICHAEL JULIAN THOMPSON	EXETER	NH
29722	TAISSA N CHERRY	SAN FRANCISCO	CA

License: For this I will group all the Licenses under one **License** Variable by employing the **unite()** function from the **tidyr** library.

```
my.new.data <-  my.new.data %>%
                unite(License, c(Physician_Profile_License_State_Code_1, Physician_Profile_License_State
```

Physician_Profile_ID	Physician	Physician_Profile_City	Physician_Profile_Sta
29708	GREGORY SENSENICH	CHILLICOTHE	MO
29709	JOHN D BALUCH	COLUMBUS	OH
29719	GEORGE A BLESSIOS	WEST SENECA	NY
29720	MARC S FISK	WEST ORANGE	NJ
29721	BENJAMIN MICHAEL JULIAN THOMPSON	EXETER	NH
29722	TAISSA N CHERRY	SAN FRANCISCO	CA

Renaming Columns: I will provide friendly names for all remaining columns.

Original Structure:

```
## 'data.frame':    814447 obs. of  10 variables:
##  $ Physician_Profile_ID              : num  29708 29709 29719 29720 29721 ...
##  $ Physician                         : chr  "GREGORY  SENSENICH " "JOHN D BALUCH " "GEORGE A BLESSIO
##  $ Physician_Profile_City            : chr  "CHILLICOTHE" "COLUMBUS" "WEST SENECA" "WEST ORANGE" ..
##  $ Physician_Profile_State           : chr  "MO" "OH" "NY" "NJ" ...
##  $ Physician_Profile_Zipcode         : chr  "64601" "43215" "14224" "07052" ...
##  $ Physician_Profile_Country_Name    : chr  "UNITED STATES" "UNITED STATES" "UNITED STATES" "UNITED
##  $ Physician_Profile_Province_Name   : chr  "" "" "" "" ...
##  $ Physician_Profile_Primary_Specialty: chr  "Allopathic & Osteopathic Physicians|Family Medicine" "A
##  $ Taxonomy                          : chr  "207Q00000X    " "208800000X    " "208600000X 204F00000
##  $ License                           : chr  "MO    " "OH    " "NY PA   " "NJ    " ...
```

```
names(my.new.data) <- c("ID","Physician", "City", "State", "Zipcode", "Country", "Province", "Specialty
```

Structure after renaming columns:

```
## 'data.frame':    814447 obs. of  10 variables:
##  $ ID       : num  29708 29709 29719 29720 29721 ...
##  $ Physician: chr  "GREGORY  SENSENICH " "JOHN D BALUCH " "GEORGE A BLESSIOS " "MARC S FISK " ...
##  $ City     : chr  "CHILLICOTHE" "COLUMBUS" "WEST SENECA" "WEST ORANGE" ...
##  $ State    : chr  "MO" "OH" "NY" "NJ" ...
##  $ Zipcode  : chr  "64601" "43215" "14224" "07052" ...
##  $ Country  : chr  "UNITED STATES" "UNITED STATES" "UNITED STATES" "UNITED STATES" ...
##  $ Province : chr  "" "" "" "" ...
##  $ Specialty: chr  "Allopathic & Osteopathic Physicians|Family Medicine" "Allopathic & Osteopathic Ph
##  $ Taxonomy : chr  "207Q00000X    " "208800000X    " "208600000X 204F00000X   " "207RC0000X    " ...
##  $ License  : chr  "MO    " "OH    " "NY PA   " "NJ    " ...
```

Final Tidy Table

ID	Physician	City	State	Zipcode	Country	Pro
29708	GREGORY SENSENICH	CHILLICOTHE	MO	64601	UNITED STATES	
29709	JOHN D BALUCH	COLUMBUS	OH	43215	UNITED STATES	
29719	GEORGE A BLESSIOS	WEST SENECA	NY	14224	UNITED STATES	
29720	MARC S FISK	WEST ORANGE	NJ	07052	UNITED STATES	
29721	BENJAMIN MICHAEL JULIAN THOMPSON	EXETER	NH	03833	UNITED STATES	
29722	TAISSA N CHERRY	SAN FRANCISCO	CA	94115	UNITED STATES	

Data Exploration

From the above table we can explore a few things as follows:

Total number of physicians:

The grand total of physicians listed in this database is 814447 physicians.

Total number of physicians by Country:

Country list with respective number of physicians and matching percentages.

Country	n Physicians	Percentage
UNITED STATES	814153	99.96 %
UNITED STATES MINOR OUTLYING ISLANDS	65	0.01 %
GERMANY	42	0.01 %
CANADA	38	0 %
JAPAN	26	0 %
GREAT BRITAIN (UK)	16	0 %
KOREA (REPUBLIC OF)	16	0 %
ITALY	12	0 %
ISRAEL	10	0 %
TURKEY	6	0 %
SAUDI ARABIA	5	0 %
UNITED ARAB EMIRATES	5	0 %
INDIA	4	0 %
MEXICO	4	0 %
PAKISTAN	4	0 %
AUSTRALIA	3	0 %
THAILAND	3	0 %
BAHRAIN	2	0 %
BRAZIL	2	0 %
CHINA	2	0 %

Country	n Physicians	Percentage
EGYPT	2	0 %
FRANCE	2	0 %
LEBANON	2	0 %
SPAIN	2	0 %
SWITZERLAND	2	0 %
ANTIGUA AND BARBUDA	1	0 %
BERMUDA	1	0 %
CAMEROON	1	0 %
GABON	1	0 %
GREECE	1	0 %
GUATEMALA	1	0 %
ICELAND	1	0 %
IRELAND	1	0 %
KOREA (DEMOCRATIC PEOPLE'S REPUBLIC OF)	1	0 %
MADAGASCAR	1	0 %
NETHERLANDS ANTILLES	1	0 %
NEW ZEALAND	1	0 %
NULL	1	0 %
PAPUA NEW GUINEA	1	0 %
PHILIPPINES	1	0 %
SOUTH AFRICA	1	0 %
TRINIDAD AND TOBAGO	1	0 %
UGANDA	1	0 %
VENEZUELA	1	0 %

Interesting is to observe how CMS provided payment to physicians around the world, I was not expecting to see those indicators.

Total number of physicians by state:

Top 5 states with highest number of physicians and respective percentages.

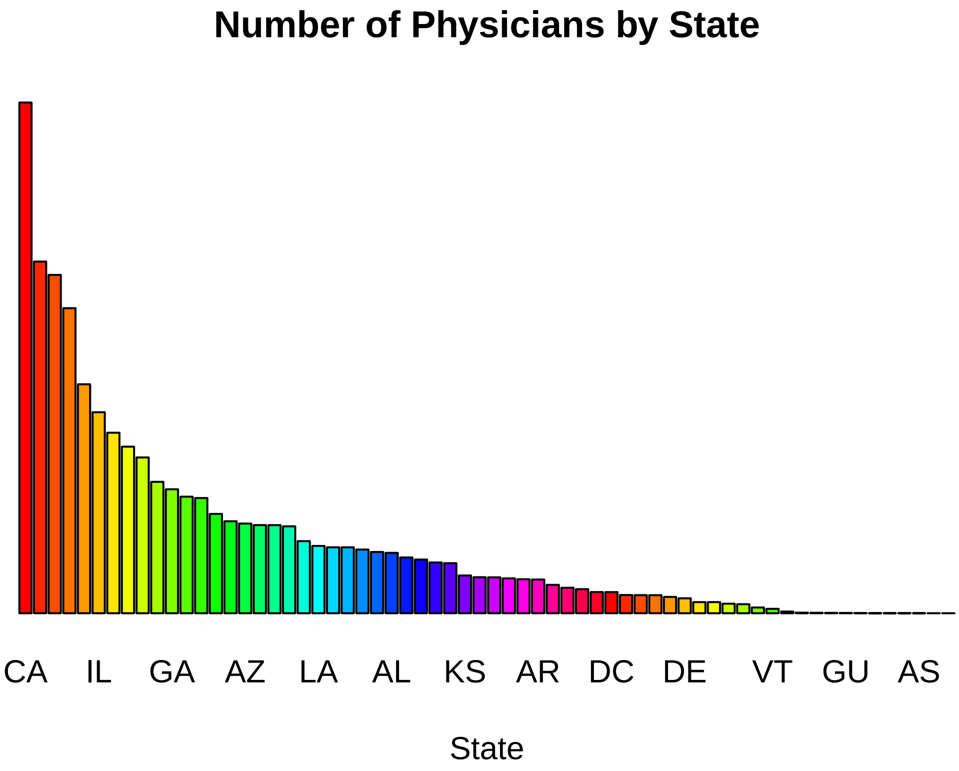
State	n Physicians	Percentage
CA	90242	11.08 %
NY	62140	7.63 %
TX	59782	7.34 %
FL	53902	6.62 %
PA	40456	4.97 %

Bottom 10 states with lowest number of physicians and respective percentages.

State	n Physicians	Percentage
AP	81	0.01 %
VI	65	0.01 %
GU	54	0.01 %
AA	29	0 %
MP	5	0 %
FM	4	0 %
PW	4	0 %

State	n Physicians	Percentage
AS	3	0 %
KO	1	0 %
MH	1	0 %

Bar plot: Total of physicians by state sorted by the number of physicians.



Something very interesting is that from the above table and the bar-plot we can spot a record that shows **MH** as a state when in reality this entry should be updated since the correct entry is **MD**. See the below table for the filtered entry from the original table. Once, an address verification is performed with the corresponding zip code, I found out that this entry should be **MD** and not **MH**.

Physician_Profile_ID	Physician_Profile_First_Name	Physician_Profile_Middle_Name	Physician_Profile_Last_Na
711782	SUSAN	G	SMIGOCKI

Similar analysis can be performed for **KO** where it stands for an APO.

Physician_Profile_ID	Physician_Profile_First_Name	Physician_Profile_Middle_Name	Physician_Profile_Last_Na
835220	BLAKE	CHARLES	STUART

Similar analysis can be performed for **PW** where it stands for different typos like **PA** or **PR** for example.

Physician_Profile_ID	Physician_Profile_First_Name	Physician_Profile_Middle_Name	Physician_Profile_Last_Na
512947	JEFFREY	ROBERT	WERT
725235	MIRIAM		KATZ
927562	ROSS	M	WEZMAR
1047744	FLOREN	E	PEREZ

The above typos can be analyzed by comparing them to our **zipcodes** library, and further analysis could be performed in order to find out how reliable is the raw data before some cleanup is performed for the physician's location.

Continuing with the data exploration I fond out that some data sanitation should be performed before any further state exploration and I should not take this data as 100% accurate as it is. If it is true the numbers so far are small these are not significant enough to create dramatic changes in our reports.

Total number of physicians by Zipcode:

Top 5 Zipcodes with highest number of physicians and respective percentages.

City	State	Zipcode	n Physicians	Percentage
Houston	TX	77030	4642	0.57 %
Chicago	IL	60612	2165	0.27 %
Philadelphia	PA	19104	2088	0.26 %
Saint Louis	MO	63110	1859	0.23 %
New York	NY	10021	1799	0.22 %

Bottom 10 Zipcodes with lowest number of physicians and respective percentages.

	City	State	Zipcode	n Physicians	Percentage
17955	Talkeetna	AK	99676	1	0 %
17956	Willow	AK	99688	1	0 %
17957	Dutch Harbor	AK	99692	1	0 %
17958	Funny River	AK	99699	1	0 %
17959	Fairbanks	AK	99707	1	0 %
17960	Central	AK	99730	1	0 %
17961	Fairbanks	AK	99775	1	0 %
17962	Tok	AK	99780	1	0 %
17963	Petersburg	AK	99833	1	0 %
17964	Klawock	AK	99925	1	0 %

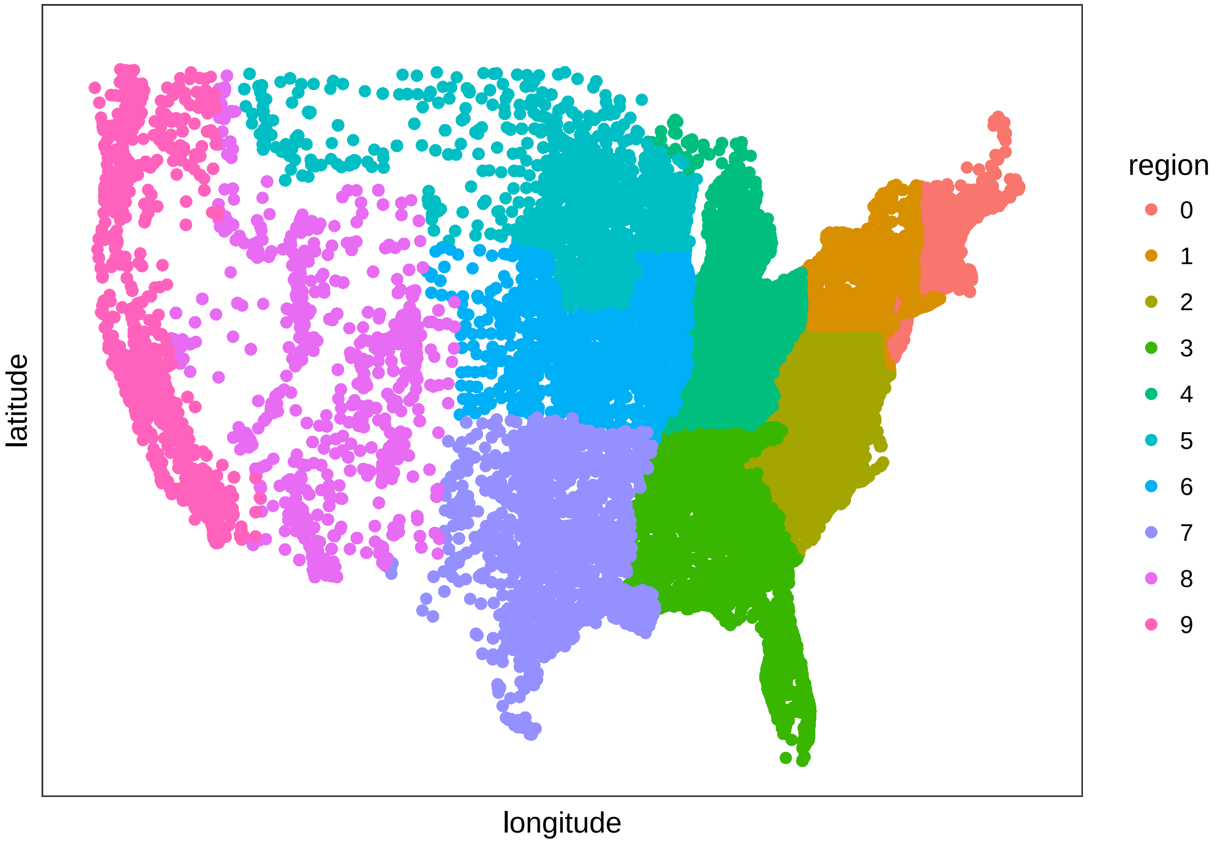
Distribution by Region:

Distribution of physicians by the first digit of the zipcode (Region).

```
# First I will subset the data for all the Zipcodes within "United States"
USzipCodes <- my.new.data %>% subset(Country == "UNITED STATES", select=c(Country, State, Zipcode))
# First I will Create a Region with the first value from the ZipCode.
USzipCodes$region = substr(USzipCodes$Zipcode, 1, 1)
# Merge Zipcodes with the zipcode library
data(zipcode)
USzipCodes <- merge(USzipCodes, zipcode, by.x='Zipcode', by.y='zip')
```

```r
# Creating ggplot of matches ZipCodes
g <- ggplot(data=USzipCodes) + geom_point(aes(x=longitude, y=latitude, colour=region))

# simplify display and limit to the "lower 48"
g <- g + theme_bw() + scale_x_continuous(limits = c(-125,-66), breaks = NULL)
g <- g + scale_y_continuous(limits = c(25,50), breaks = NULL)
g
```



Conclusions

Country:

Based on simple observation, is easy to spot how CMS has made payments to physicians from out of this country, this is something I was not expecting to find and it definitely caught my attention since I was under the impression that those kind of payments for services were performed in the US territories.

State:

If we look at the chart and results, we noticed that over all California and New York, presented the highest physician populations. This, I believe is correlated due to major concentration of humans in those states, once again in this comparison comes to show how less populated areas tend to have less physician populations registered with the open payment data from CMS.

Zipcode:

This is an interesting analysis and I believe it can play a great role in local discoveries related to quality of health care and competition, since it covers immediate surrounding areas.

For example, from the **Region** distribution we can visualize how the distribution is over the country having a lot of empty spaces on the mid west. Also, something interesting is to observe that the top 3 Zip codes present on the Zip code table, belong to Houston, Chicago and Philadelphia; this is remarkable since these states did not figured as the top states, yet these zip codes have a concentrated number of physicians.

Final conclusion:

Since there's more data available, it will be interesting to perform more comparisons in regards of the specialties, number of licenses or number of taxonomies that physicians have; this, to find out if some physicians are "better" prepared to deal with proper treatment for all the patients in the surrounding areas, the above conclusion is by assuming that patients tend to visit local health care practitioners in any given emergency situation.