IST687 – Viz Map HW: Median Income

Dan Burke

8/30/2021

Assignment Due: 8/23/2021

Submitted: 9/11/2021

Note:

As the 'zipcode package' is no longer supported, I chose to utilize the 'usa' package. This package did possess some downsides, one of which is that it does not have a complete up to date list of all zipcodes.

When matching the CSV data and zipcode lat/long position data, it caused some regions be excluded, this in turn has caused my plots to be absent some states.

Download the dataset from the LMS that has median income
by zip code (an excel file). # Step 1: Load the Data

2) Clean up the dataframe

a. Remove any info at the front of the file that's not needed
b. Update the column names (zip, median, mean, population)

3) Load the 'zipcode' package
4) Merge the zip code information from the two data frames (merge into one dataframe)
5) Remove Hawaii and Alaska (just focus on the 'lower 48' states)

```
#install.packages("ggplot2")
#install.packages("ggmap")
#install.packages("usa")
if(!require("ggplot2") || !require("ggmap") || !require("usa") || !require("openintro")){
  library(ggplot2)
  library(ggmap)
  library(usa)
  library(openintro)
}
```

```
## Loading required package: ggplot2
```

```
## Loading required package: ggmap
```

```
## Warning: package 'ggmap' was built under R version 4.1.1
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
## Loading required package: usa
```

```
## Warning: package 'usa' was built under R version 4.1.1
```

```
## The 'usa' package masks the state datasets included in base R:
## * state.abb
## * state.area
## * state.center
## * state.division
## * state.name
## * state.region
## Objects are similar in class and content but updated and expanded.
```

```
##
## Attaching package: 'usa'

## The following objects are masked from 'package:datasets':
##
##     state.abb, state.area, state.center, state.division, state.name,
##     state.region

## Loading required package: openintro

## Warning: package 'openintro' was built under R version 4.1.1

## Loading required package: airports

## Warning: package 'airports' was built under R version 4.1.1

## Loading required package: cherryblossom

## Warning: package 'cherryblossom' was built under R version 4.1.1

## Loading required package: usdata

## Warning: package 'usdata' was built under R version 4.1.1
```

```
#1) Read    the data    - using the gdata   package we  have    previously  used.
incomeFrame <- read.csv("C:\\Users\\danbu\\Desktop\\IntroToDataScience\\MedianZIP_2_2.csv")


#2) Clean   up  the dataframe
  #Check for NA's
  naQuant <- paste("Amount of NA's ", sum(is.na(incomeFrame)))
  print(naQuant)
```

```
## [1] "Amount of NA's  0"
```

```
  #Check the Column Names
  incomeCols <- colnames(incomeFrame)

  #Here we rename the columns
  newColnames <- as.character(incomeFrame[1,])
  colnames(incomeFrame) <- newColnames

  #Now we remove the first column and have clean data
  incomeFrame <- incomeFrame[2:nrow(incomeFrame),]
  incomeFrame$Pop <- gsub(",", "",incomeFrame$Pop)
  incomeFrame$Mean <- gsub(",", "",incomeFrame$Mean)
  incomeFrame$Median <- gsub(",", "",incomeFrame$Median)
  incomeFrame$Pop <- as.numeric(incomeFrame$Pop)
  incomeFrame$Mean <- as.numeric(incomeFrame$Mean)
```

```
## Warning: NAs introduced by coercion
```

```
  incomeFrame$Median <- as.numeric(incomeFrame$Median)

#3) Load    the 'zipcode'   package

  #I had to use the following link to manually install
  #https://cran.r-project.org/src/contrib/Archive/zipcode/

  #library(dplyr)
  library(zipcodeR)
```

```
## Warning: package 'zipcodeR' was built under R version 4.1.1
```

```
#4) Merge   the zip code    information from    the two data    frames
#(merge into    one dataframe)

  #Grab all of the Zips clean and place into a DataFrame
  zips <- data.frame(usa::zipcodes)
  zips <- zips[c(1,3:5)]
  zips <-na.omit(zips)
  matchedZips <- match(zips$zip,incomeFrame$Zip)
  cleanZips <- incomeFrame[matchedZips,]

  cleanZips <- incomeFrame[matchedZips,]
  cleanZips <- na.omit(cleanZips)

  lat <- vector()
  lon <- vector()
  state<- vector()

  for(i in 1:nrow(cleanZips)){

      state[i] <- zips[which(zips$zip == cleanZips[i,1]),2]
      lat[i] <- zips[which(zips$zip == cleanZips[i,1]),3]
      lon[i] <- zips[which(zips$zip == cleanZips[i,1]),4]

    }

  locationsFrame <- data.frame(cleanZips$Zip, cleanZips$Median, cleanZips$Pop, cleanZips$Mean, lat, lon
  colnames(locationsFrame) <- c("zip", "median", "pop", "mean", "lat", "lon", "state")
  locationsFrame <- na.omit(locationsFrame)
```

Step 2: Show the income & population per state

1) Create a simpler dataframe, with just the average median income and the the population for each state.

2) Add the state abbreviations and the state names as new columns (make sure the state names are all lower case)

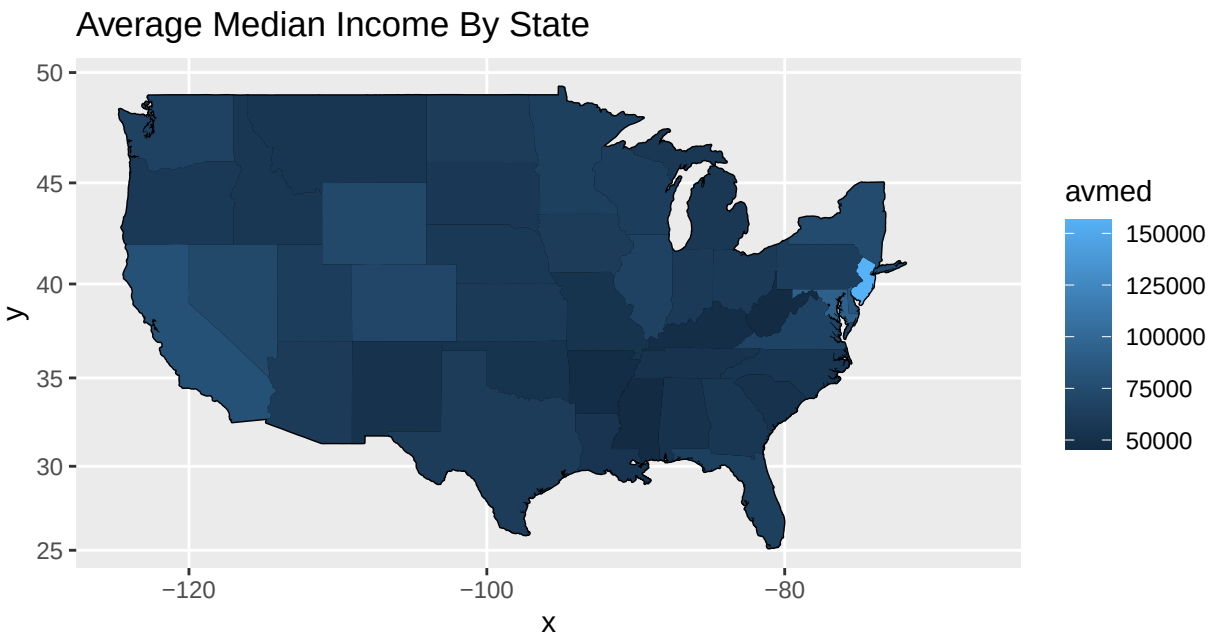
3) Show the U.S. map, representing the color with the average median income of that state

4) Create a second map with color representing the population of the state

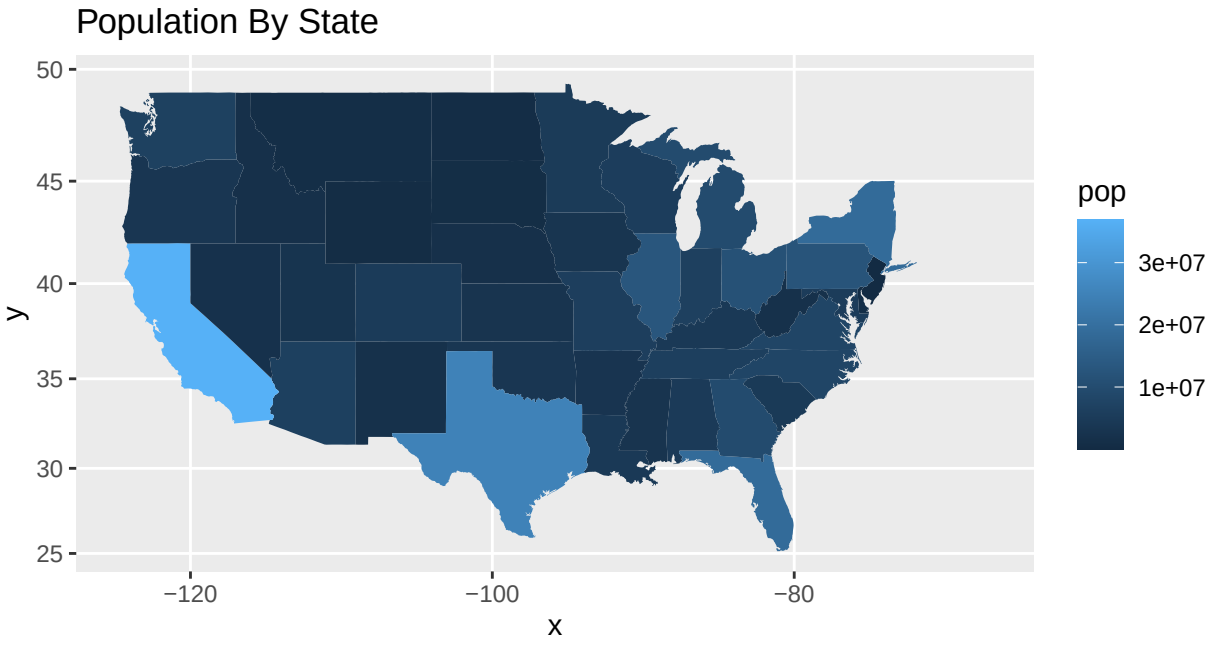
```
states <- usa::states
pop <- vector()
avMed <- vector()

for(i in 1:nrow(states)){
  pop[i]<- sum(as.numeric(locationsFrame[which(locationsFrame$state == states$abb[i]), 3]))
  avMed[i] <- mean(as.numeric(locationsFrame[which(locationsFrame$state == states$abb[i]), 4]))
}

states$name<- tolower(states$name)
states$abb <- tolower(states$abb)

simplerData <- data.frame(states$name, states$abb, pop, avMed)
colnames (simplerData) <- c("state", "abb", "pop", "avmed")
simplerData <- na.omit(simplerData)
simplerData <- simplerData[which(simplerData$abb != "ak" & simplerData$abb != "dc" & simplerData$abb !=

str(simplerData)
```

```
## 'data.frame':    42 obs. of  4 variables:
##  $ state: chr  "alabama" "arizona" "arkansas" "california" ...
##  $ abb  : chr  "al" "az" "ar" "ca" ...
##  $ pop  : num  4770242 6360679 2936699 36926889 4979279 ...
##  $ avmed: num  51963 60153 47749 79870 70361 ...
##  - attr(*, "na.action")= 'omit' Named int [1:7] 7 20 22 30 40 41 47
##   ..- attr(*, "names")= chr [1:7] "7" "20" "22" "30" ...
```

```
us <- map_data("state")

map.incomePopMap <- ggplot(simplerData, aes(map_id=state))
map.incomePopMap <- map.incomePopMap + geom_map(map=us, fill="White", color="black")
map.incomePopMap <- map.incomePopMap + expand_limits(x=us$long, y=us$lat)
map.incomePopMap <- map.incomePopMap + geom_map(map=us, aes(fill=avmed))
map.incomePopMap <- map.incomePopMap + expand_limits(x=us$long, y = us$lat)
map.incomePopMap <- map.incomePopMap + coord_map() + ggtitle("Average Median Income By State")
map.incomePopMap
```



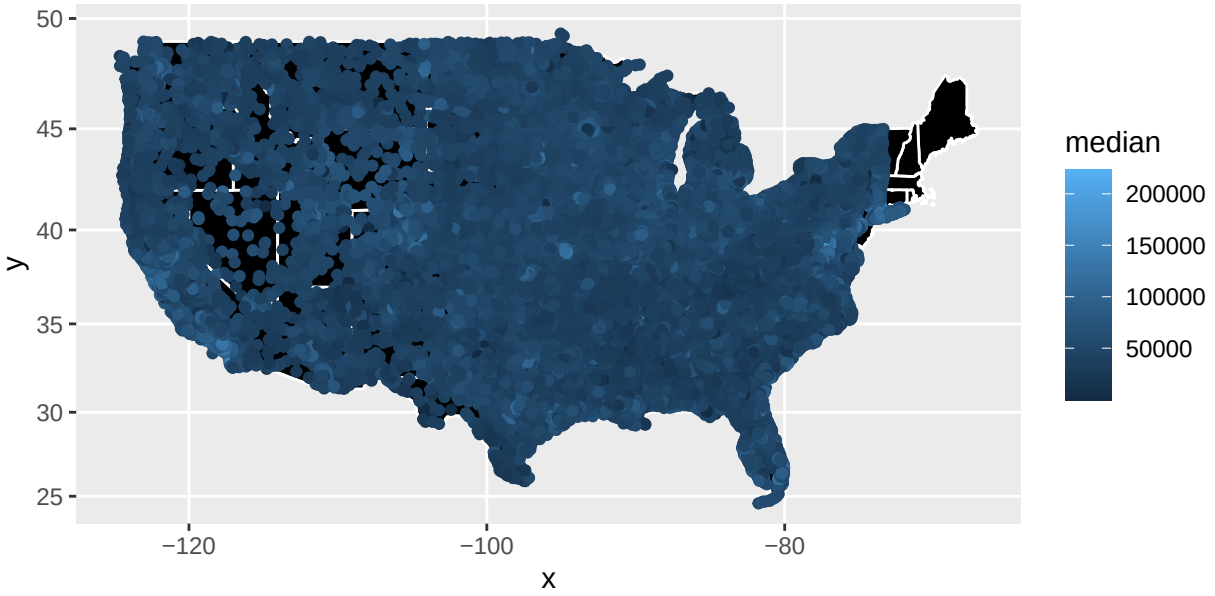

```
map.popMap <- ggplot(simplerData, aes(map_id = state))

map.popMap <- map.popMap + geom_map(map=us, aes(fill=pop))
map.popMap <- map.popMap + expand_limits(x=us$long, y = us$lat)
map.popMap <- map.popMap + coord_map() + ggtitle("Population By State")
map.popMap
```



Step 3: Show the income per zip code

1) Have draw each zip code on the map, where the color of the 'dot' is based on the median income. To make the map look appealing, have the background of the map be black.

```
locationsFrame$state <- tolower(locationsFrame$state)

zipmapdata <- data.frame(tolower(states$name), tolower(states$abb))
colnames(zipmapdata) <- c("state", "abb")
zipmapdata <- zipmapdata[which(zipmapdata$abb != "ak" & zipmapdata$abb != "dc" & zipmapdata$abb != "hi"
locationsFrame <- locationsFrame[which(locationsFrame$state != "ak" & locationsFrame$state != "dc" & lo

zipMap <- ggplot(zipmapdata, aes(map_id=state))
zipMap <- zipMap + geom_map(map=us, fill="black", color="white")
zipMap <- zipMap + expand_limits(x=us$long, y=us$lat)
zipMap <- zipMap + coord_map()


zipMap <- zipMap +  geom_point(data=locationsFrame, aes(x=lon, y=lat, color=median))
zipMap
```



Step 4: Show Zip Code Density

1) Now generate a different map, one where we can easily see where there are lots of zip codes, and where there are few (using the 'stat_density2d' function).

```
#I encountered a lot of difficulty attempting to get this code block to run.
#I've followed and searched through documentation and stack overflow looking
#for a remedy. The below code does run, with the exception of the final
#output being an error of: "Error Aesthetics must be either
#lenth of 1 or the same as the data"


dummyDf <- data.frame(locationsFrame$state, stringsAsFactors = FALSE)
us<- map_data("state")

locationsFrame <- locationsFrame[complete.cases(locationsFrame),]
statenames <- abbr2state(locationsFrame$state)
statenames <- tolower(statenames)
tempDf <- data.frame(locationsFrame$state,statenames, locationsFrame$lat, locationsFrame$lon, locationsF
colnames(tempDf) <- c("abbrv", "statename", "lat", "lon", "median")

zipdensity <- ggplot(dummyDf, aes(map_id=state))
zipdensity <- zipdensity + geom_map(map=us, fill="black", color="white")
zipdensity <- zipdensity + expand_limits(x=us$long, y=us$lat)
```

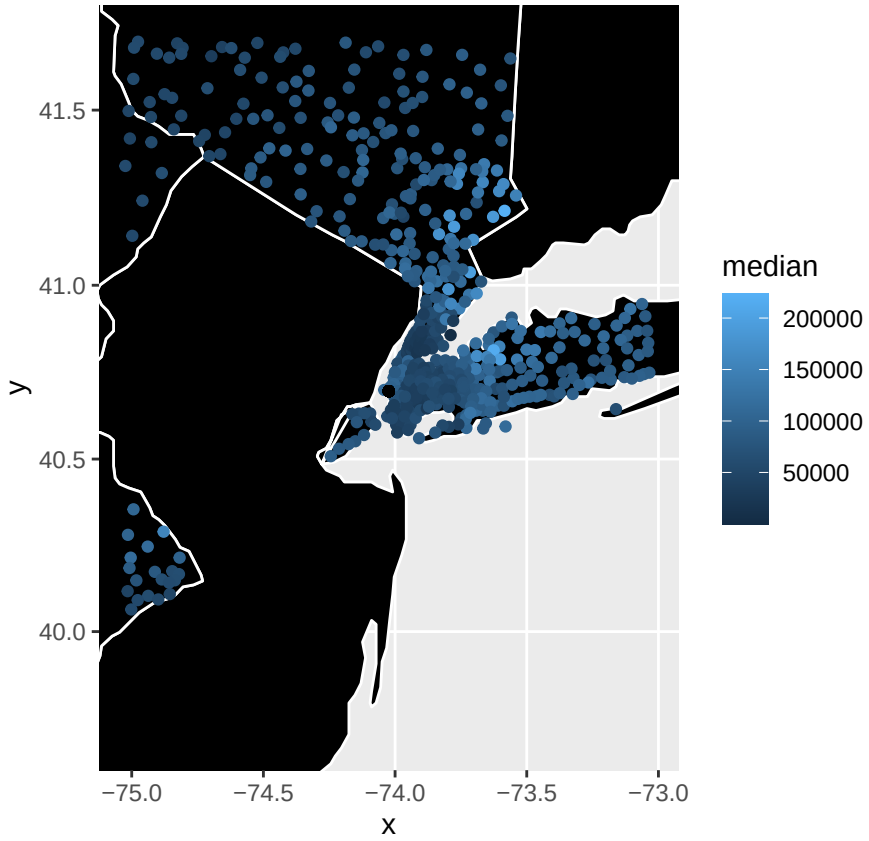
```
zipdensity <- zipdensity + coord_map() + ggtitle("Zipcode Density")
zipdensity <- zipdensity + stat_density2d(data=tempDf, aes(x=lon, y=lat, color=median))
#zipdensity
```

Step 5: Zoom in to the region around NYC

1) Repeat steps 3 & 4, but have the image / map be of the northeast U.S. (centered around New York).

```
zoomLat <- 40.695760
zoomLon <- -74.024070

zoomLevel <- 1
lonLimit <- c(zoomLon - zoomLevel,zoomLon + zoomLevel)
latLimit <- c(zoomLat - zoomLevel,zoomLat + zoomLevel)

nycZoom <- zipMap + geom_point(aes(x=zoomLon, y=zoomLat)) + xlim(lonLimit) + ylim(latLimit)
nycZoom
```

```
## Warning: Removed 29361 rows containing missing values (geom_point).
```



```
#Because I had difficulty in getting the stat_density2d() to run properly;
#I've included the code where if it it was able to run It will have created a zoomed density map.

zoomedZipDensity <- zipdensity
zoomedZipDensity <- zoomedZipDensity + geom_point(aes(x=zoomLon, y=zoomLat)) + xlim(lonLimit) + ylim(la
#zoomedZipDensity
```