IST687 - Viz Map HW: Median Income

John Fields

5/14/2019

Download the dataset from the LMS that has median income by zip code (an excel file).

Step 1: Load the Data

1) Read the data – using the gdata package we have previously used.
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)

```
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
```

```
##
```

```
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
## Attaching package: 'gdata'
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```
rawdata <- read.xls("/Users/johnfields/Library/Mobile Documents/com~apple~CloudDocs/Syracuse/IST687/Hom
#Rename the columns
namesOfColumns<-c("zip","median","mean","population")
cleandata<-function(rawdata,namesOfColumns)
{colnames(rawdata)<-namesOfColumns
return(rawdata)
}
results<-cleandata(rawdata,namesOfColumns)
head(results)
```

```
##    zip median   mean population
## 1 1001 56,663 66,688     16,445
## 2 1002 49,853 75,063     28,069
## 3 1003 28,462 35,121      8,491
## 4 1005 75,423 82,442      4,798
## 5 1007 79,076 85,802     12,962
## 6 1008 63,980 78,391      1,244
```

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)

Note: This blog post was very helpful for this section -> https://blog.exploratory.io/geocoding-us-zip-code-data-with-dplyr-and-zipcode-package-7f539c3702b0

```
library(zipcode)
library(stringr)
#Add leading zero's to the results$zip field
results$zip <- str_pad(results$zip, pad="0", side="left", width=5)
data(zipcode)

#Merge the Median data with zip code data
newresults <- merge(results, zipcode, by="zip")

#Remove Alaska and Hawaii
newresults <- newresults[newresults$state!="AK",]
newresults <- newresults[newresults$state != "HI",]
newresults$median <- gsub(",","",newresults$median)
newresults$median <- as.numeric(newresults$median)
View(newresults)
```

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)

```
#use tapply to calculate the average median income and population
statemean <- tapply(newresults$median,newresults$state,mean)
statepopsum <- tapply(as.numeric(newresults$population),newresults$state,sum)
simpleresults <- data.frame(statemean,statepopsum)
simpleresults$stateabb <- row.names(simpleresults)
str(simpleresults)
```

```
## 'data.frame':    49 obs. of  3 variables:
##  $ statemean  : num [1:49(1d)] 40550 36961 48132 62629 56303 ...
##  $ statepopsum: num [1:49(1d)] 4625200 4480222 3153863 14341540 3915597 ...
##  $ stateabb   : chr  "AL" "AR" "AZ" "CA" ...
```

```
#The function abbr2state in the openintro package was used to convert the
#state abbreviations to names
#https://rdrr.io/cran/openintro/man/abbr2state.html
library(openintro)
```

```
## Please visit openintro.org for free statistics materials
```

```
##
## Attaching package: 'openintro'
```

```
## The following objects are masked from 'package:datasets':
##
##     cars, trees
```

```
simpleresults$states <- abbr2state(simpleresults$stateabb)
simpleresults$states <- tolower(simpleresults$states)
```

```
simpleresults <- simpleresults[simpleresults$stateabb !="DC",]
simpleresults$states
```

```
##  [1] "alabama"        "arkansas"       "arizona"        "california"
##  [5] "colorado"       "connecticut"    "delaware"       "florida"
##  [9] "georgia"        "iowa"           "idaho"          "illinois"
## [13] "indiana"        "kansas"         "kentucky"       "louisiana"
## [17] "massachusetts"  "maryland"       "maine"          "michigan"
## [21] "minnesota"      "missouri"       "mississippi"    "montana"
## [25] "north carolina" "north dakota"   "nebraska"       "new hampshire"
## [29] "new jersey"     "new mexico"     "nevada"         "new york"
## [33] "ohio"           "oklahoma"       "oregon"         "pennsylvania"
## [37] "rhode island"   "south carolina" "south dakota"   "tennessee"
## [41] "texas"          "utah"           "virginia"       "vermont"
## [45] "washington"     "wisconsin"      "west virginia"  "wyoming"
```

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

```
library("ggplot2")
```

```
##
## Attaching package: 'ggplot2'
```

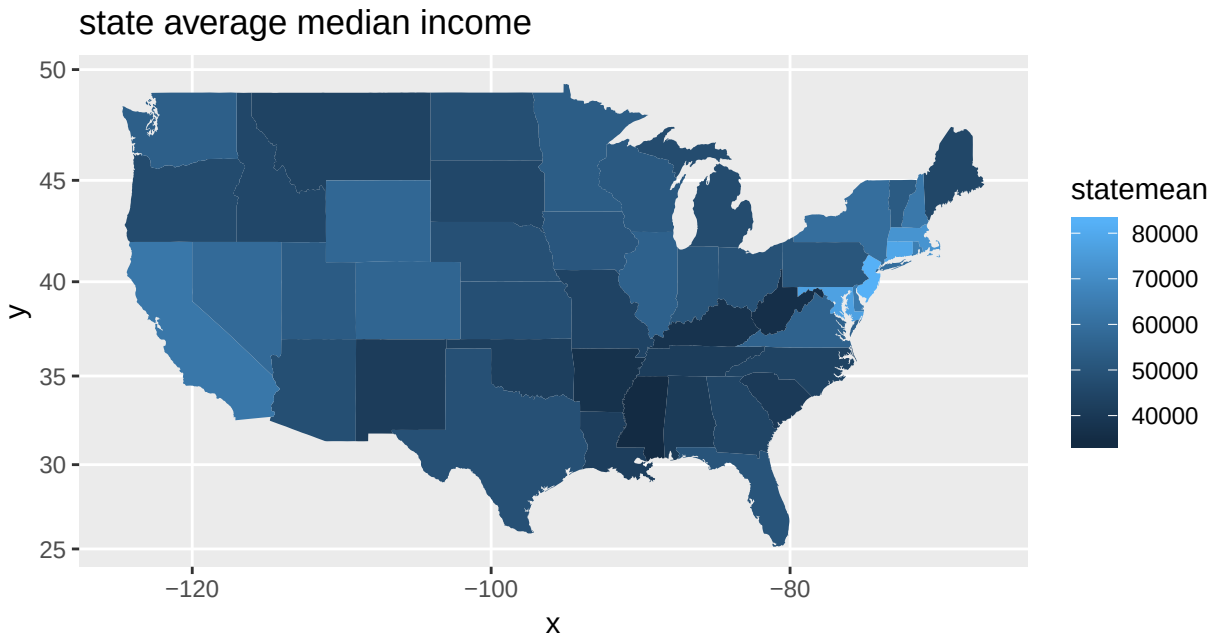
```
## The following object is masked from 'package:openintro':
##
##     diamonds
```

```
# I signed up for Google's API server so I can use the geocode function
library("ggmap")
```

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

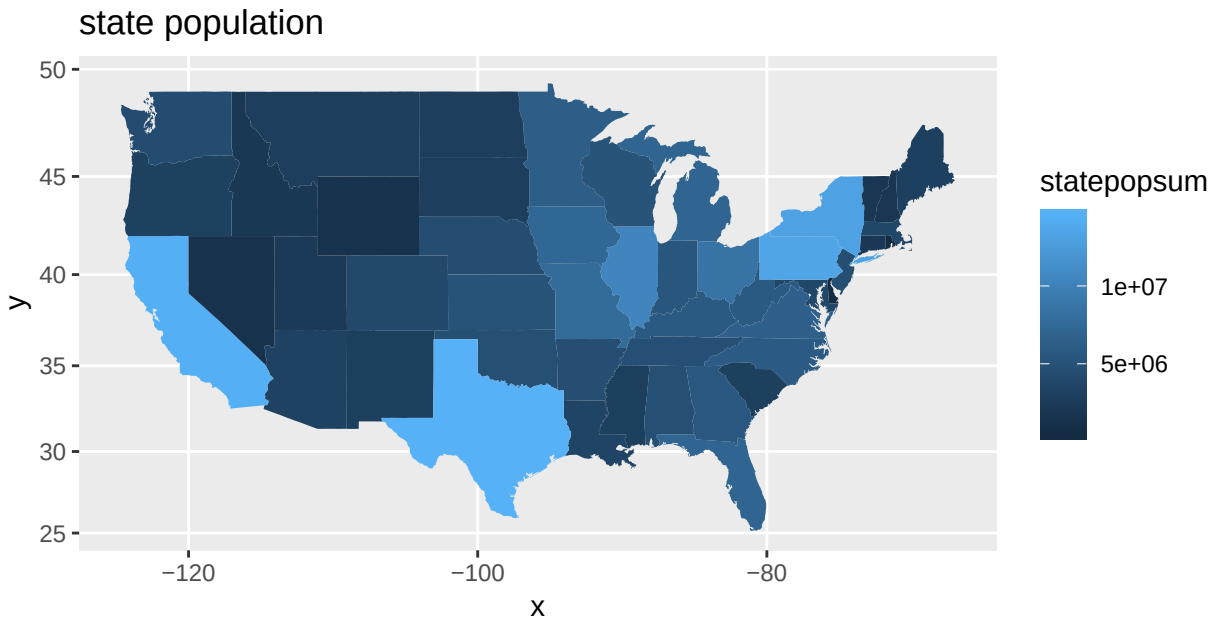
```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
simpleresults$states <- as.character(simpleresults$states)
us <- map_data("state")
#map_data("state")
map.meanColor <- ggplot(simpleresults,aes(map_id=states))
map.meanColor <- map.meanColor + geom_map(map=us,aes(fill=statemean))
map.meanColor <- map.meanColor + expand_limits(x=us$long,y=us$lat)
map.meanColor <- map.meanColor + coord_map() + ggtitle("state average median income")
map.meanColor
```



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

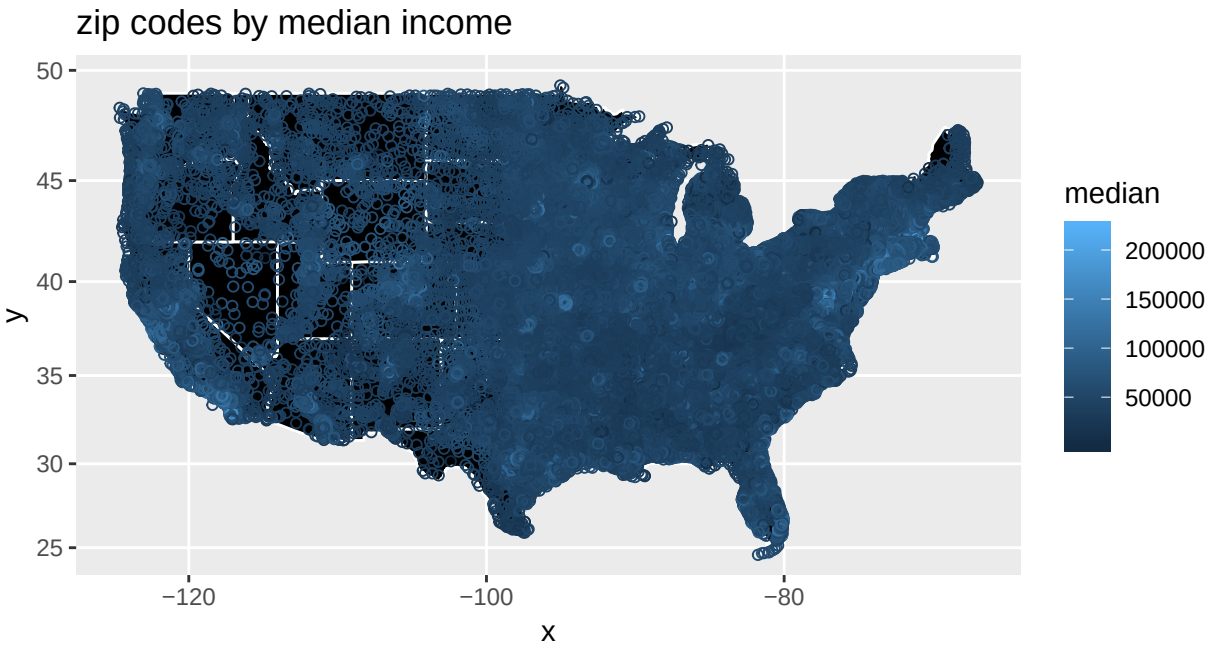
```
map.popColor <- ggplot(simpleresults,aes(map_id=states))
map.popColor <- map.popColor + geom_map(map=us,aes(fill=statepopsum))
map.popColor <- map.popColor + expand_limits(x=us$long,y=us$lat)
map.popColor <- map.popColor + coord_map() + ggtitle("state population")
map.popColor
```



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.

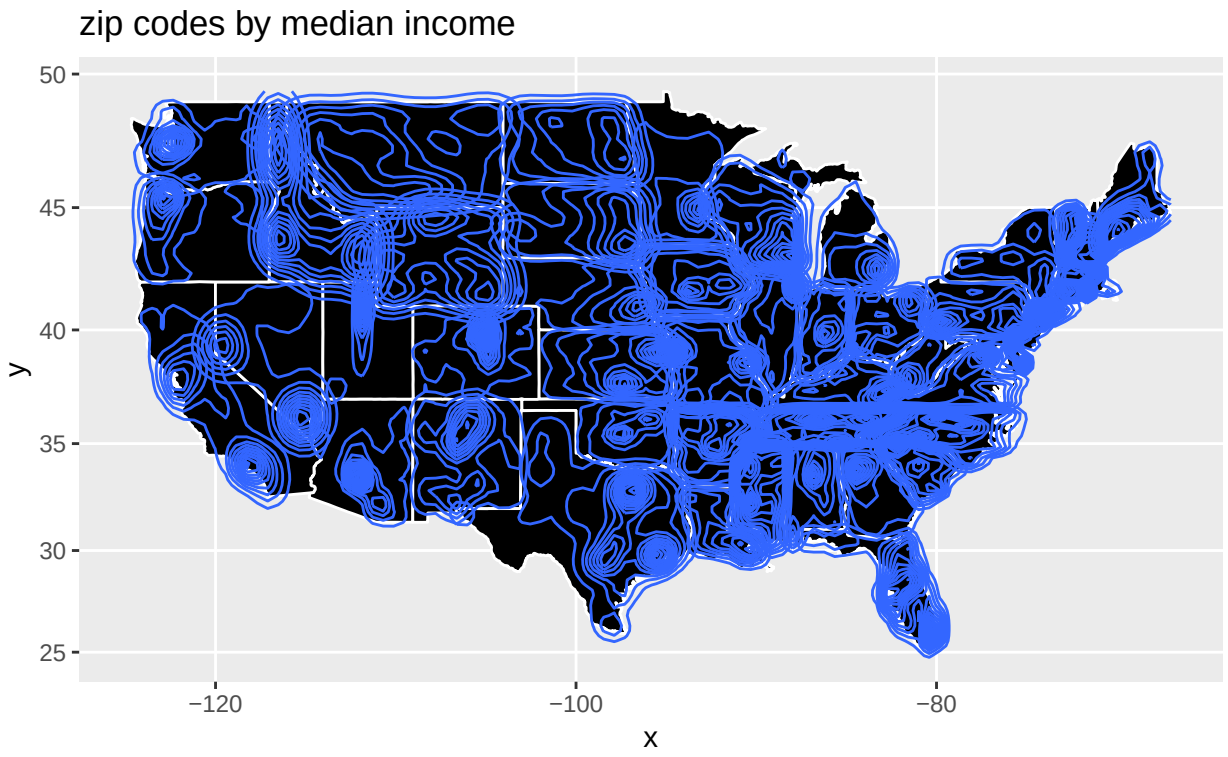
```
#Show the location of zip codes by median income
library(ggplot2)
library(ggmap)
newresults$statename <- abbr2state(newresults$state)
newresults$statename <- tolower(newresults$statename)
dummyDF <- data.frame(newresults$statename,stringsAsFactors=FALSE)
us <- map_data("state")
map.simple <- ggplot(dummyDF,aes(map_id=newresults$statename))
map.simple <- map.simple + geom_map(map=us,fill="black",color="white")
map.simple <- map.simple + expand_limits(x=us$long,y=us$lat)
map.simple <- map.simple + coord_map() + ggtitle("zip codes by median income")
medianBYzip <- map.simple + geom_point(data=newresults,aes(x=newresults$longitude,y=newresults$latitude
medianBYzip
```



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).

```
#Show the location of zip codes by density
library(ggplot2)
library(ggmap)
newresults$statename <- abbr2state(newresults$state)
newresults$statename <- tolower(newresults$statename)
dummyDF <- data.frame(newresults$statename,stringsAsFactors=FALSE)
us <- map_data("state")
map.simple <- ggplot(dummyDF,aes(map_id=newresults$statename))
map.simple <- map.simple + geom_map(map=us,fill="black",color="white")
map.simple <- map.simple + expand_limits(x=us$long,y=us$lat)
map.simple <- map.simple + coord_map() + ggtitle("zip codes by median income")
medianBYzipDensity <- map.simple + stat_density_2d(data=newresults,aes(x=newresults$longitude,y=newresul
medianBYzipDensity
```



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).

```
#Part 5 - Zoom Part 3
#Show NYC location of zip codes by median income
library(ggplot2)
library(ggmap)
register_google(key = "AIzaSyAj2U-Rmg3e2zdymwc23KSHXM5qExT27Zs")
zoomGeo <- geocode("New York, ny")
```

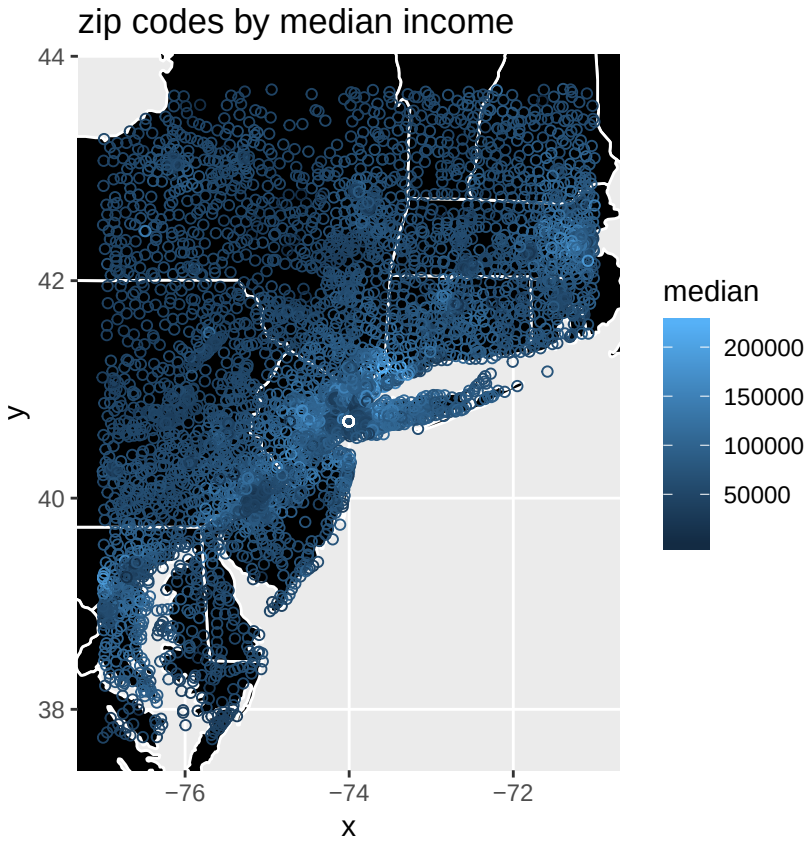
```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=New+York,+ny&key=xxx-Rmg3e2zdymwc
```

```
zoomAmount <- 3

centerx <- zoomGeo$lon
centery <- zoomGeo$lat
ylimit <- c(centery-zoomAmount, centery+zoomAmount)
xlimit <- c(centerx-zoomAmount, centerx+zoomAmount)

map.zoom1 <- medianBYzip
map.zoom1 + geom_point(aes(x=zoomGeo$lon,y=zoomGeo$lat),shape=1,color="white") + xlim(xlimit) + ylim(yl
```

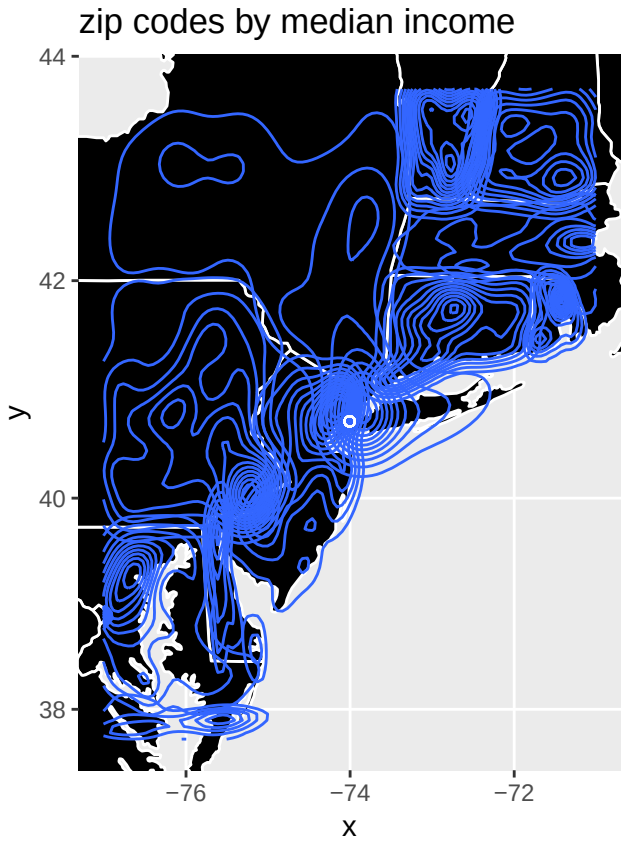
```
## Warning: Removed 28243 rows containing missing values (geom_point).
```




```
#Part 5 - Zoom Part 4
#Show NYC density of zip codes by median income

map.zoom2 <- medianBYzipDensity
map.zoom2 + geom_point(aes(x=zoomGeo$lon,y=zoomGeo$lat),shape=1,color="white") + xlim(xlimit) + ylim(yli
```

```
## Warning: Removed 28243 rows containing non-finite values (stat_density2d).
```


zip codes by median income
44
42
40
38
y
−76
−74
−72
x