R Notebook

Title: "IST687 – Viz Map HW: Median Income"
Name: Sathish Kumar Rajendiran
Week: 7
Date: 05/21/2020

Exercise: Median Income

Install necessary packages

```
install.packages( pkgs=c("gdata","readxl","zipcode","ggplot2","reshape2","ggeasy","viridis"),repos = "h
```

```
## Warning: package 'zipcode' is not available (for R version 3.6.3)
```

```
##
## The downloaded binary packages are in
##  /var/folders/_z/ltmjkt4156b37rsk7cgvj7l80000gn/T//RtmpmayPq1/downloaded_packages
```

```
library(readxl)
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
```

```r
library(xlsx)
library(ggplot2)
library(ggcorrplot)
library(reshape2)
library(ggeasy)
library(zipcode)
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(viridisLite)
```

#printDataInfo function

```r
# function printDataInfo
  printDataInfo <- function(myData)
    {
      strinfo <- str(myData)
      cat("str:",strinfo,"\n")

      colnamesinfo <- colnames(myData)
      cat("colnames:",colnamesinfo,"\n")

      diminfo <- dim(myData)
      cat("dim:",diminfo,"\n")

      nrowinfo <- nrow(myData)
      cat("nrow:",nrowinfo,"\n")

      nrowsinfo <- myData[1:3,]
      return(nrowsinfo)
     }
```

Step 1: Load the Data

```r
# 1) Read the data - using the gdata package we have previously used.

    filepath <- "/Users/sathishrajendiran/Documents/MedianZIP_2_2.xlsx"
    data_types<-c("text","text","text","text")

    # function readFiles
    readFiles <- function(fpath)
      {
      dftemp <- read_excel(filepath,skip = 1,col_types = data_types)
      return(dftemp)
    }

    medianxls_dftemp <- readFiles(filepath)
    medianxls_dftemp[1:10,]
```

```
## # A tibble: 10 x 4
##    Zip   Median             Mean               Pop
##    <chr> <chr>              <chr>              <chr>
##  1 1001  56662.573499999999 66687.750899999999 16445
##  2 1002  49853.417699999998 75062.634300000005 28069
##  3 1003  28462              35121              8491
##  4 1005  75423              82442              4798
##  5 1007  79076.354000000007 85801.975000000006 12962
##  6 1008  63980              78391              1244
##  7 1009  51452              66737              889
##  8 1010  75625              80919              3340
##  9 1011  63476.383999999998 77443.486399999994 1323
## 10 1012  58750              74722              677
```

Clean 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)

#   Remove NAs
     medianxls <- na.omit(medianxls_dftemp)

#   Round and convert to integer
     medianxls$Median <- round(as.numeric( medianxls$Median),0)
     medianxls$Mean <- round(as.numeric( medianxls$Mean),0)
```

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

```
     medianxls$Pop <- round(as.numeric( medianxls$Pop),0)

     medianxlsDF <- data.frame( medianxls)

     medianxlsDF <- na.omit(medianxlsDF)

    #   Rename  column names as (zip, median, mean, population)
     colnames(medianxlsDF) <- c("zip","median","mean","population")

    #   Review the dataframe
     dim(medianxlsDF) # 32634 rows    4 columns
```

```
## [1] 32627     4
```

```
     str(medianxlsDF)
```

```
## 'data.frame':    32627 obs. of  4 variables:
##  $ zip       : chr  "1001" "1002" "1003" "1005" ...
##  $ median    : num  56663 49853 28462 75423 79076 ...
##  $ mean      : num  66688 75063 35121 82442 85802 ...
```

```
##  $ population: num  16445 28069 8491 4798 12962 ...
##  - attr(*, "na.action")= 'omit' Named int  7056 26132 26133 26134 26201 29645 29980
##   ..- attr(*, "names")= chr  "7056" "26132" "26133" "26134" ...
```

```
medianxlsDF[1:3,]
```

```
##    zip median  mean population
## 1 1001  56663 66688      16445
## 2 1002  49853 75063      28069
## 3 1003  28462 35121       8491
```

Load Zipcode Datafram

```
# 3) Load the 'zipcode' package

install.packages("zipcode", repos = "https://cran.r-project.org/src/contrib/Archive/zipcode/zipcode_
```

```
## Warning: unable to access index for repository https://cran.r-project.org/src/contrib/Archive/zipcod
##   cannot open URL 'https://cran.r-project.org/src/contrib/Archive/zipcode/zipcode_1.0.tar.gz/src/con
```

```
## Warning: package 'zipcode' is not available (for R version 3.6.3)
```

```
## Warning: unable to access index for repository https://cran.r-project.org/src/contrib/Archive/zipcod
##   cannot open URL 'https://cran.r-project.org/src/contrib/Archive/zipcode/zipcode_1.0.tar.gz/bin/mac
```

```
library(zipcode)
data(zipcode)

str(zipcode)
```

```
## 'data.frame':    44336 obs. of  5 variables:
##  $ zip      : chr  "00210" "00211" "00212" "00213" ...
##  $ city     : chr  "Portsmouth" "Portsmouth" "Portsmouth" "Portsmouth" ...
##  $ state    : chr  "NH" "NH" "NH" "NH" ...
##  $ latitude : num  43 43 43 43 43 ...
##  $ longitude: num  -71 -71 -71 -71 -71 ...
```

```
zipcodeDF <- zipcode
zipcodeDF <- na.omit(zipcodeDF)

length(unique(zipcodeDF$state))
```

```
## [1] 60
```

```
unique(zipcodeDF$state)
```

```
##  [1] "NH" "NY" "PR" "VI" "MA" "RI" "ME" "VT" "CT" "NJ" "AE" "PA" "DE" "DC" "VA"
## [16] "MD" "WV" "NC" "SC" "GA" "TN" "FL" "AL" "AR" "KY" "MS" "OH" "IN" "MI" "IA"
## [31] "IL" "WI" "MN" "SD" "ND" "MT" "MO" "KS" "NE" "LA" "OK" "TX" "NM" "CO" "WY"
## [46] "ID" "UT" "AZ" "NV" "CA" "HI" "AS" "GU" "PW" "FM" "MP" "MH" "OR" "WA" "AK"
```

```
    # zipcodeDF[1:5,]

    colnames(zipcodeDF)
```

```
## [1] "zip"       "city"      "state"     "latitude"  "longitude"
```

```
    # zipcodeDF[1:5,]

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

    mergeDF <- merge(medianxlsDF, zipcodeDF, by="zip")

    mergeDF[1:10,]
```

```
##      zip median   mean population     city state latitude longitude
## 1  10001  71245 123113      17678 New York    NY 40.75074 -73.99653
## 2  10002  30844  46259      70878 New York    NY 40.71704 -73.98700
## 3  10003  89999 139331      53609 New York    NY 40.73251 -73.98935
## 4  10004 110184 156683       1271 New York    NJ 40.69923 -74.04118
## 5  10005 115133 163763       1517 New York    NY 40.70602 -74.00858
## 6  10006 111220 156776        972 New York    NY 40.70790 -74.01342
## 7  10007 145459 256236       3520 New York    NY 40.71475 -74.00721
## 8  10009  56615  78138      56975 New York    NY 40.72709 -73.97864
## 9  10010  93702 137106      27322 New York    NY 40.73902 -73.98205
## 10 10011  92359 160937      45899 New York    NY 40.74101 -74.00012
```

```
    dim(mergeDF)
```

```
## [1] 30236     8
```

```
    dim(mergeDF)
```

```
## [1] 30236     8
```

```
    length(unique(mergeDF$state))
```

```
## [1] 45
```

```
#5) Remove Hawaii and Alaska (just focus on the 'lower 48' states)

    mergeDF <- mergeDF[which(mergeDF$state != "AK" & mergeDF$state != "HI"), ]
    mergeDF[1:10,]
```

```
##      zip median   mean population     city state latitude longitude
## 1  10001  71245 123113      17678 New York    NY 40.75074 -73.99653
## 2  10002  30844  46259      70878 New York    NY 40.71704 -73.98700
## 3  10003  89999 139331      53609 New York    NY 40.73251 -73.98935
## 4  10004 110184 156683       1271 New York    NJ 40.69923 -74.04118
## 5  10005 115133 163763       1517 New York    NY 40.70602 -74.00858
## 6  10006 111220 156776        972 New York    NY 40.70790 -74.01342
## 7  10007 145459 256236       3520 New York    NY 40.71475 -74.00721
## 8  10009  56615  78138      56975 New York    NY 40.72709 -73.97864
## 9  10010  93702 137106      27322 New York    NY 40.73902 -73.98205
## 10 10011  92359 160937      45899 New York    NY 40.74101 -74.00012
```

```
    length(unique(mergeDF$state))
```

```
## [1] 43
```

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 st

    #tapply to calculate avg median by state
    income <- tapply(mergeDF$median,mergeDF$state,mean)

    # prepare column calues for state
    state <- rownames(income)

    # prepare dataframe containing state names and average median values
    avgIncome <- data.frame(state,income)
    # avgIncome[1:3,]

    #reset row index
    row.names(avgIncome) <- 1:nrow(avgIncome)
    avgIncome[1:3,]
```

```
##   state   income
## 1    AL 40549.90
## 2    AR 36960.95
## 3    AZ 48132.07
```

```
    #tapply to calculate avg population by state
    population <- tapply(mergeDF$population,mergeDF$state,mean)

    # prepare column calues for state
    state <- rownames(population)

    # prepare dataframe containing state names and average population values
    avgPopulation <- data.frame(state,population)
    # avgpopulation[1:3,]

    #reset row index
```

```
    row.names(avgPopulation) <- 1:nrow(avgPopulation)
    avgPopulation[1:3,]
```

```
##   state population
## 1    AL   7441.875
## 2    AR   4943.938
## 3    AZ  15744.255
```

```
    #merge avgIncome and avgPopulation dataframes by state

    dfIncome <- merge(avgIncome,avgPopulation, by="state")
    dfIncome[1:3,]
```

```
##   state   income population
## 1    AL 40549.90   7441.875
## 2    AR 36960.95   4943.938
## 3    AZ 48132.07  15744.255
```

```
# 2) Add the state abbreviations and the state names as new columns (make sure the state names are all
    dfIncome$stateName <- tolower(state.name[match(dfIncome$state,state.abb)])
    dfIncome[1:3,]
```

```
##   state   income population stateName
## 1    AL 40549.90   7441.875   alabama
## 2    AR 36960.95   4943.938  arkansas
## 3    AZ 48132.07  15744.255   arizona
```

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

    # load US States dataset.
    us <- map_data("state")

    # Plot map using average median income

    dummyDF <- data.frame(state.name,stringsAsFactors = FALSE)
    dummyDF$state.name <- tolower(dummyDF$state.name)
    printDataInfo(dummyDF)
```

```
## 'data.frame':    50 obs. of  1 variable:
##  $ state.name: chr  "alabama" "alaska" "arizona" "arkansas" ...
## str:
## colnames: state.name
## dim: 50 1
## nrow: 50
```

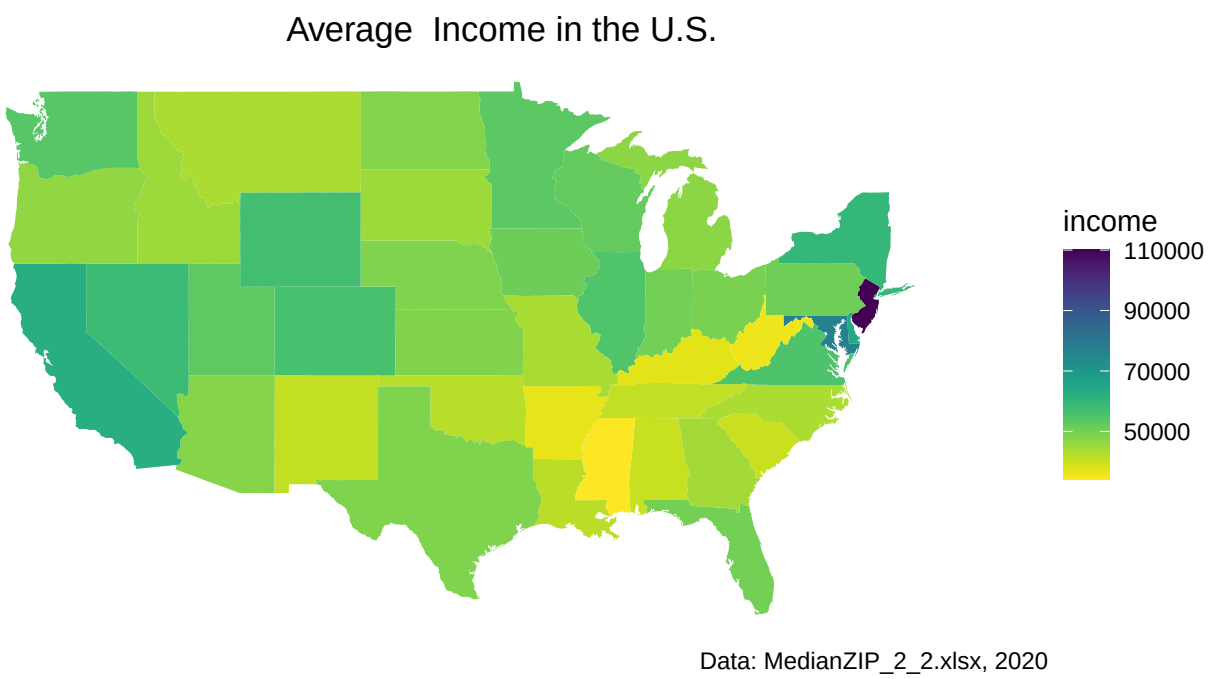
```
## [1] "alabama" "alaska"  "arizona"
```

```
    theme <-theme(plot.title = element_text(hjust = 0.5),axis.title = element_text())

    map.incomeColor <- ggplot(dfIncome,aes(map_id=stateName))
```

```
    map.incomeColor <- map.incomeColor + geom_map(map=us,aes(fill=income))
    map.incomeColor <- map.incomeColor + expand_limits(x=us$long,y=us$lat)
    map.incomeColor <- map.incomeColor + coord_map()
    map.incomeColor <- map.incomeColor +  scale_fill_viridis(option = "viridis", direction = -1) + theme
    labs(x = NULL,
         y = NULL,
         title = "Average  Income in the U.S.",
         caption = "Data: MedianZIP_2_2.xlsx, 2020") +theme

    map.incomeColor
```




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

    # Plot map using average population

    # dummyDF <- data.frame(state.name,stringsAsFactors = FALSE)
    # dummyDF$state.name <- tolower(dummyDF$state.name)
    printDataInfo(dummyDF)
```

```
## 'data.frame':    50 obs. of  1 variable:
##  $ state.name: chr  "alabama" "alaska" "arizona" "arkansas" ...
## str:
## colnames: state.name
## dim: 50 1
## nrow: 50
```

```
## [1] "alabama" "alaska"  "arizona"
```

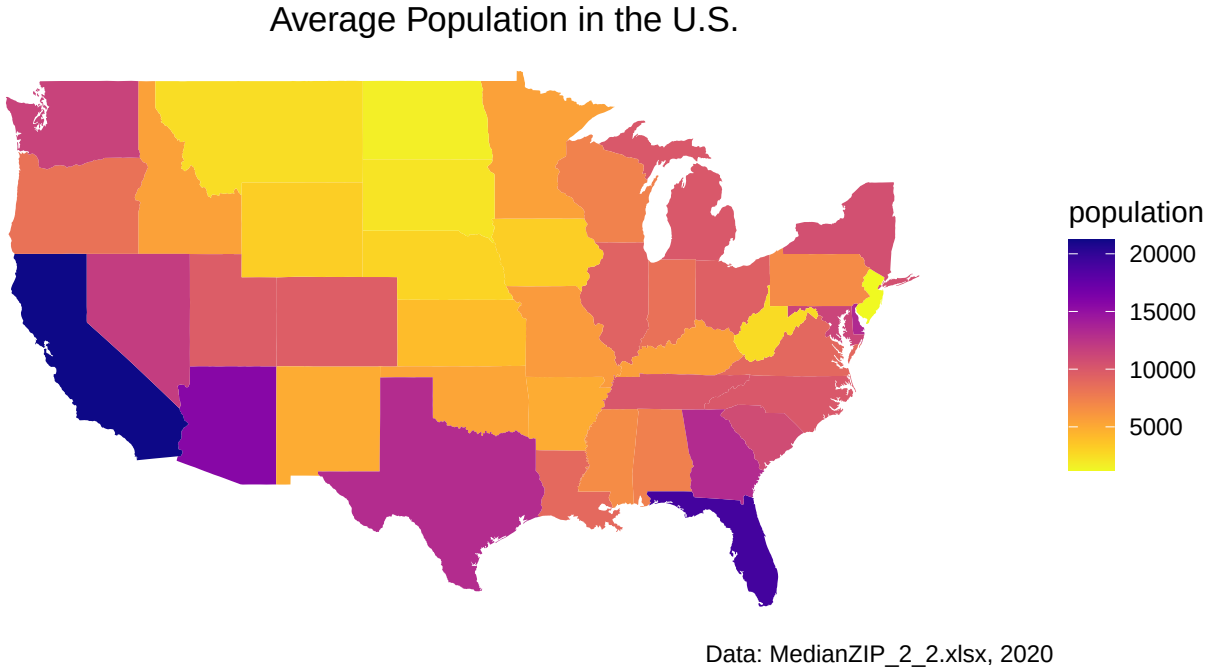
```
    printDataInfo(dfIncome)
```

```
## 'data.frame':    43 obs. of  4 variables:
##  $ state     : Factor w/ 43 levels "AL","AR","AZ",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ income    : num  40550 36961 48132 62626 56303 ...
##  $ population: num  7442 4944 15744 21174 9631 ...
##  $ stateName : chr  "alabama" "arkansas" "arizona" "california" ...
## str:
## colnames: state income population stateName
## dim: 43 4
## nrow: 43
```

```
##   state   income population stateName
## 1    AL 40549.90   7441.875   alabama
## 2    AR 36960.95   4943.938  arkansas
## 3    AZ 48132.07  15744.255   arizona
```

```
    map.popColor <- ggplot(dfIncome,aes(map_id=stateName))
    map.popColor <- map.popColor + geom_map(map=us,aes(fill=population))
    map.popColor <- map.popColor + expand_limits(x=us$long,y=us$lat)
    map.popColor <- map.popColor + coord_map()
    map.popColor <- map.popColor + scale_fill_viridis(option = "plasma", direction = -1) + theme_void()
    labs(x = NULL,
         y = NULL,
         title = "Average Population in the U.S.",
         caption = "Data: MedianZIP_2_2.xlsx, 2020") +theme

    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

    printDataInfo(mergeDF)
```

```
## 'data.frame':    29923 obs. of  8 variables:
##  $ zip       : chr  "10001" "10002" "10003" "10004" ...
##  $ median    : num  71245 30844 89999 110184 115133 ...
##  $ mean      : num  123113 46259 139331 156683 163763 ...
##  $ population: num  17678 70878 53609 1271 1517 ...
##  $ city      : chr  "New York" "New York" "New York" "New York" ...
##  $ state     : chr  "NY" "NY" "NY" "NJ" ...
##  $ latitude  : num  40.8 40.7 40.7 40.7 40.7 ...
##  $ longitude : num  -74 -74 -74 -74 -74 ...
## str:
## colnames: zip median mean population city state latitude longitude
## dim: 29923 8
## nrow: 29923
```

```
##     zip median   mean population     city state latitude longitude
## 1 10001  71245 123113      17678 New York    NY 40.75074 -73.99653
```

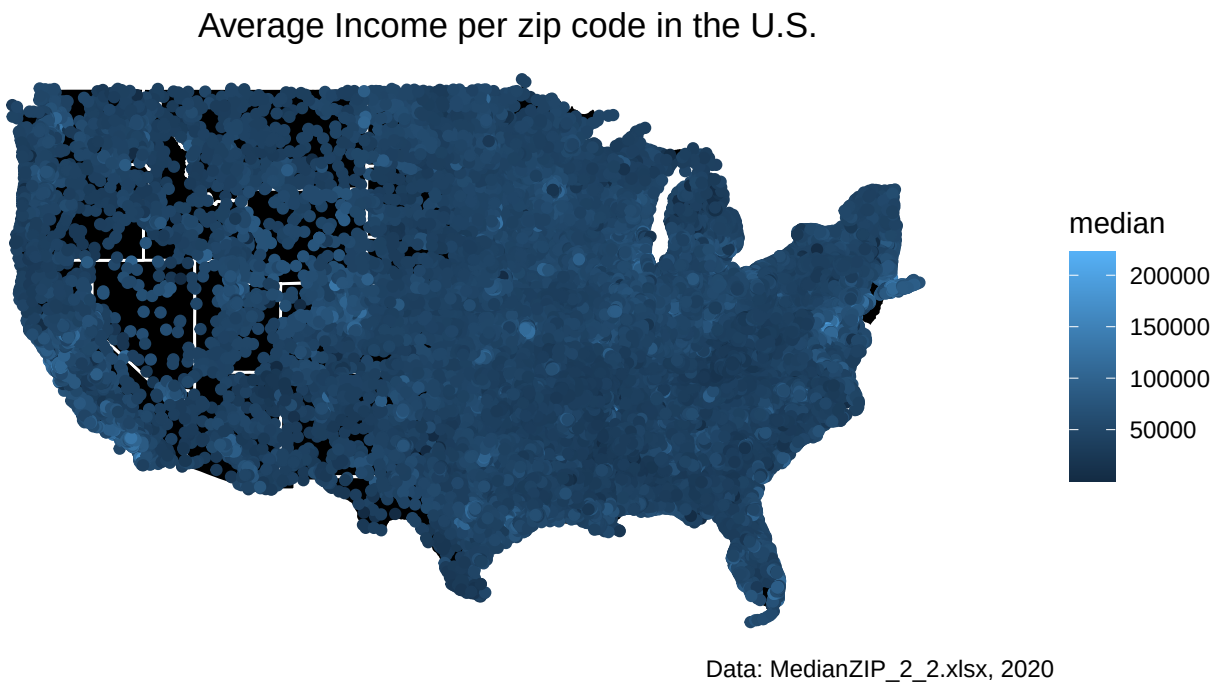
```
## 2 10002  30844  46259      70878 New York    NY 40.71704 -73.98700
## 3 10003  89999 139331      53609 New York    NY 40.73251 -73.98935
```

```
    mergeDF$stateName <- tolower(state.name[match(mergeDF$state,state.abb)])
    mergeDF[1:3,]
```

```
##     zip median   mean population     city state latitude longitude stateName
## 1 10001  71245 123113      17678 New York    NY 40.75074 -73.99653  new york
## 2 10002  30844  46259      70878 New York    NY 40.71704 -73.98700  new york
## 3 10003  89999 139331      53609 New York    NY 40.73251 -73.98935  new york
```

```
    map.zipColor <- ggplot(mergeDF,aes(map_id=stateName))
    map.zipColor <- map.zipColor + geom_map(map=us, fill="black", color="white")
    map.zipColor <- map.zipColor + expand_limits(x=us$long,y=us$lat)
    map.zipColor <- map.zipColor + coord_map()
    map.zipColor <- map.zipColor + geom_point(data = mergeDF,aes(x = longitude, y = latitude, color=medi
    map.zipColor <- map.zipColor + theme_void() +
      labs(x = NULL,
           y = NULL,
           title = "Average Income per zip code in the U.S.",
           caption = "Data: MedianZIP_2_2.xlsx, 2020") +theme

    map.zipColor
```



#Step 4: Show Zip Code Density

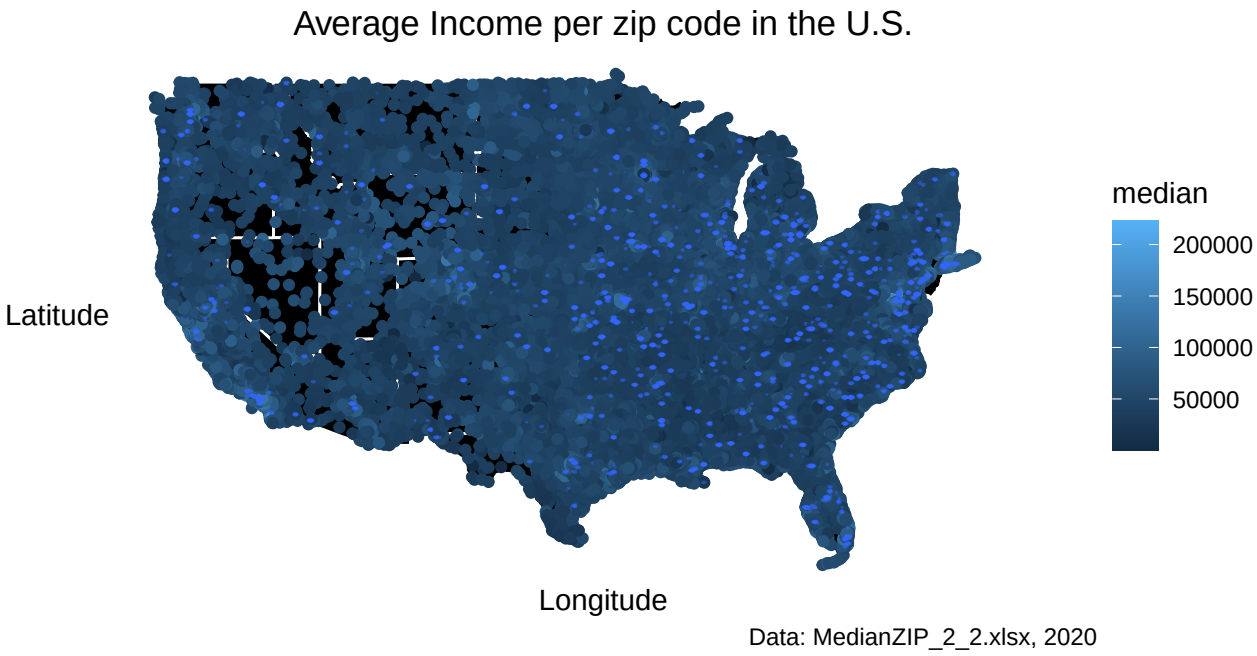
```
# 1) Now generate a different map, one where we can easily see where there are lots of zip codes, and w

    mergeDF[1:3,]
```

```
##     zip median   mean population     city state latitude longitude stateName
## 1 10001  71245 123113      17678 New York    NY 40.75074 -73.99653  new york
## 2 10002  30844  46259      70878 New York    NY 40.71704 -73.98700  new york
## 3 10003  89999 139331      53609 New York    NY 40.73251 -73.98935  new york
```

```
    map.density2D <- map.zipColor + stat_density2d(aes(x = longitude, y = latitude), alpha = .5, h = .0
                geom = "density_2d", data = mergeDF)

    map.density2D +  xlab('Longitude') + ylab('Latitude')
```



#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).

    #review NY Zipcodes
    # mergeDF[1:3,]
    # nyDF <- subset(mergeDF, state == "NY")
    # nyDF[,c("state","stateName","latitude","longitude")]
    #
    #import NY geocodes from library(ggmap)
    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.
```

```
    # ?register_google

    register_google("AIzaSyDWp5xj2USZ5eOYnL7V5sJMx0w76d1NCDY")

    latlon <- geocode("new york,ny")
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=new+york,ny&key=xxx
```

```
    latlon
```

```
## # A tibble: 1 x 2
##     lon   lat
##   <dbl> <dbl>
## 1 -74.0  40.7
```

```
     map.NY <- map.density2D+ geom_point(aes(x=latlon$lon,y=latlon$lat),color="yellow",size=5,na.rm = TR

     map.NY <-  map.NY + xlim(latlon$lon-10, latlon$lon+10) + ylim(latlon$lat-10,latlon$lat+10) + coord_
```

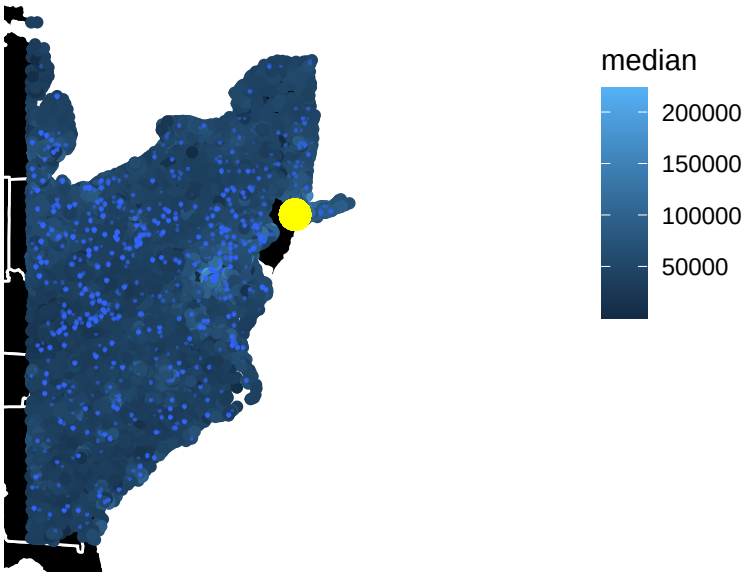
```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```

```
     map.NY +ggtitle("NYC Zoomed")
```

```
## Warning: Removed 20919 rows containing non-finite values (stat_density2d).

## Warning: Removed 20919 rows containing missing values (geom_point).
```

NYC Zoomed



Data: MedianZIP_2_2.xlsx, 2020