

Ryan Timbrook

Applied Data Science

IST687 Intro to Data Science, Spring 2019

Due Date: 05/21/2019

Homework: 7

NetID: RTIMBROO

SUID: 386792749

#R Code - unexecuted

Homework Week 7: Viz Map HW: Median Income

#--- Preprocess Steps:-----

Clear objects from Memory

```
rm(list=ls())
```

Clear Console:

```
cat("\014")
```

Set Working Directory

```
setwd("C:\\workspaces\\ms_datascience_su\\IST687-IntroDataScience\\R_workspace\\hw")
```

#---- Global Variable Assignments -----

```
incomeDataSetFileName <- 'MedianZIP_2_2.xlsx'
```

#---- Load Required Packages -----

```
if(!require("devtools")) {install.packages("devtools")}
```

```
devtools::install_github("dkahle/ggmap")
```

```
if(!require("readxl")) {install.packages("readxl")}
```

```
if(!require("gdata")) {install.packages("gdata")}
```

```
if(!require("ggplot2")){install.packages("ggplot2")}
```

```
if(!require("ggmap")){install.packages("ggmap")}
```

```
if(!require("mapproj")){install.packages("mapproj")}
```

```
if(!require("dplyr")) {install.packages("dplyr")}
```

```
if(!require("sqldf")) {install.packages("sqldf")}
```

```
if(!require("zipcode")) {install.packages("zipcode")}
```

```
if(!require("reshape2")) {install.packages("reshape2")}
```

Register Google API

```
register_google(key="AlzaSyDdjIKuumlpQunYJxtMYEdEq5o32QJgJ28")
```

#---- Step 1: Load the data -----

1.1: Read the data:

```
readDataSetsasXLSX <- function(fName){
```

```
  ds <- read_xlsx(fName)
```

```
  return(data.frame(ds))
```

```
}
```

```
income.df <- readDataSetsXLSX(incomeDataSetFileName)
str(income.df)
head(income.df)
```

1.2: Clean the dataframe:

```
cleanIncomeDf <- function(ds){
  ## Remove Columns
  colnames(ds) <- NULL
  ds <- ds[-1,]

  ## Rename Columns
  newColNames <- c('zip', 'median', 'mean', 'population')
  colnames(ds) <- newColNames

  ## Remove commas and make numeric
  ds$median <- as.numeric(gsub(",", "", ds$median))
  ds$mean <- as.numeric(gsub(",", "", ds$mean))
  ds$population <- as.numeric(gsub(",", "", ds$population))

  return(ds)
}
```

```
income.df <- cleanIncomeDf(income.df)
str(income.df)
head(income.df)
```

1.3: Load the 'zipcode' package:

```
data(zipcode)
head(zipcode)
```

Reformat zip codes

```
income.df$zip <- clean.zipcodes(income.df$zip)
head(income.df$zip)
```

1.4: Merge the zip code information from the two data frames (merge into one dataframe)

```
income.by.zipcode.df <- merge(income.df, zipcode, by='zip')
head(income.by.zipcode.df)
```

1.5: Remove the Hawaii and Alaska (just focus on the 'lower 48' states)

```
income.by.zipcode.df <- income.by.zipcode.df[income.by.zipcode.df$state != 'HI',]
income.by.zipcode.df <- income.by.zipcode.df[income.by.zipcode.df$state != 'AK',]
income.by.zipcode.df <- income.by.zipcode.df[income.by.zipcode.df$state != 'DC',]
```

#---- Step 2: Show the income & population per state -----

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

Average Median Income

```
income <- tapply(income.by.zipcode.df$median, income.by.zipcode.df$state, mean)
state <- rownames(income)
median.income <- data.frame(state, income)
```

```

# Population for each state
pop <- tapply(income.by.zipcode.df$population, income.by.zipcode.df$state, sum)
state <- rownames(pop)
state.pop <- data.frame(state,pop)

# Merge the above two data frames by 'state'
income.by.state.simp.df <- merge(median.income,state.pop,by="state")
head(income.by.state.simp.df)

# Alternative method of simplifying data frame
incomeByZipDf <- income.by.zipcode.df
incomeByStateSimpAltDf <- sqldf("select state, avg(median) as income, sum(population) as pop from incomeByZipDf
group by state")
#incomeByStateSimpAltDf <- sqldf("select state, (income/pop) as income, pop from incomeByStateSimpAltDf")

## 2.2: Add the state abbreviations and the state names as new columns (make sure the state names are all lower case)
income.by.state.simp.df$state_name <- tolower(state.name[match(income.by.state.simp.df$state,state.abb)])

## 2.3: Show the U.S. map, representing the color with the average median income of that state
removeThemeAxis <- theme(
  axis.text = element_blank(),
  axis.line = element_blank(),
  axis.ticks = element_blank(),
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.title = element_blank()
)

us.map <- map_data("state")
g.map.income <- ggplot(data=income.by.state.simp.df, mapping=aes(map_id=state_name))
g.map.income <- g.map.income + geom_map(map=us.map, mapping = aes(fill=income))
g.map.income <- g.map.income + expand_limits(x=us.map$long, y=us.map$lat)

g.map.income <- g.map.income + coord_map()
g.map.income <- g.map.income + ggtitle("Average Median Income by State") +
theme(plot.title=element_text(hjust=0.5))
g.map.income <- g.map.income + guides(fill=guide_legend(title="Income")) + removeThemeAxis
g.map.income
ggsave("U.S._Map_of_Average_Median_Income_by_State.jpg", width = 6, height = 6)

## 2.4: Create a second map with color representing the population of the state
g.map.pop <- ggplot(data=income.by.state.simp.df, mapping=aes(map_id=state_name))
g.map.pop <- g.map.pop + geom_map(map=us.map, mapping = aes(fill=pop))
g.map.pop <- g.map.pop + expand_limits(x=us.map$long, y=us.map$lat)

g.map.pop <- g.map.pop + coord_map()
g.map.pop <- g.map.pop + ggtitle("State Population") + theme(plot.title=element_text(hjust=0.5))
g.map.pop <- g.map.pop + guides(fill=guide_legend(title="Population")) + removeThemeAxis
g.map.pop
ggsave("U.S._Map_of_Population_by_State.jpg", width = 6, height = 6)

```

#---- Step 3: Show the income per zip code -----

3.1: Draw each zipcode 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.

```
income.by.zipcode.df$state_name <- tolower(state.name[match(income.by.zipcode.df$state,state.abb)])
head(income.by.zipcode.df)
```

```
g.map.zip <- ggplot(data=income.by.zipcode.df, mapping=aes(map_id=state_name))
g.map.zip <- g.map.zip + geom_map(map=us.map, fill="black", color="white")
g.map.zip <- g.map.zip + expand_limits(x=us.map$long, y=us.map$lat)
```

```
g.map.zip <- g.map.zip + geom_point(data=income.by.zipcode.df, mapping=aes(x=income.by.zipcode.df$longitude,
y=income.by.zipcode.df$latitude, color=income.by.zipcode.df$median))
g.map.zip <- g.map.zip + coord_map()
g.map.zip <- g.map.zip + ggtitle("Income per Zip Code") + theme(plot.title=element_text(hjust=0.5))
g.map.zip <- g.map.zip + guides(color=guide_legend(title="Median Income"))
g.map.zip <- g.map.zip + removeThemeAxis
g.map.zip
ggsave("U.S._Map_of_Median_Income_by_ZipCode.jpg", width = 6, height = 6)
```

#---- Step 4: Show Zip Code Density -----

4.1: 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)

```
g.map.zip.density <- ggplot(data=income.by.zipcode.df, mapping=aes(map_id=state_name))
g.map.zip.density <- g.map.zip.density + geom_map(map=us.map, fill="black", color="white")
g.map.zip.density <- g.map.zip.density + expand_limits(x=us.map$long, y=us.map$lat)
```

```
g.map.zip.density <- g.map.zip.density + stat_density_2d(data=income.by.zipcode.df,
mapping=aes(x=income.by.zipcode.df$longitude, y=income.by.zipcode.df$latitude))
g.map.zip.density <- g.map.zip.density + coord_map()
g.map.zip.density <- g.map.zip.density + ggtitle("Zip Code Density") + theme(plot.title=element_text(hjust=0.5))
g.map.zip.density <- g.map.zip.density + removeThemeAxis
g.map.zip.density
ggsave("U.S._Map_of_ZipCode_Density.jpg", width = 6, height = 6)
```

#---- Step 5: Zoom in to the region around NYC -----

5.1: Repeat stes 3 & 4, but have the image / map be of the northeast U.S. (Centered around New York)

```
nyc <- geocode("New York, NY", source = "dsk")
```

```
zoom <- 2
```

```
#ggmap(get_map(nyc, zoom=4))
```

```
center_x <- nyc$lon
```

```
center_y <- nyc$lat
```

```
y_limit <- c(center_y-zoom, center_y+zoom)
```

```
x_limit <- c(center_x-zoom, center_x+zoom)
```

5.1.1: Draw each zipcode 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.

```
g.map.nyc.zip.income <- g.map.zip + xlim(x_limit) + ylim(y_limit) + coord_map()
g.map.nyc.zip.income <- g.map.nyc.zip.income + geom_point(aes(x=center_x, y=center_y), color="darkred", size=3)
```

```

g.map.nyc.zip.income <- g.map.nyc.zip.income + ggtitle("Income by Zip around NYC") +
theme(plot.title=element_text(hjust=0.5))
g.map.nyc.zip.income
ggsave("U.S._Map_of_ZipCode_NYC.jpg", width = 6, height = 6)

```

```

## 5.1.2: 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)
g.map.nyc.zip.density <- g.map.zip.density + xlim(x_limit) + ylim(y_limit) + coord_map()
g.map.nyc.zip.density <- g.map.nyc.zip.density + stat_density_2d(aes(x=center_x, y=center_y), color="darkred", size=3)
g.map.nyc.zip.density <- g.map.nyc.zip.density + ggtitle("Zip Code Density around NYC") +
theme(plot.title=element_text(hjust=0.5))
g.map.nyc.zip.density

```

```

ggsave("U.S._Map_of_NYC_ZipCode_Density.jpg", width = 6, height = 6)

```

#R Code – executed

```

> ### Set Working Directory
> setwd("C:\\workspaces\\ms_datascience_su\\IST687-IntroDataScience\\R_workspace\\hw")
>
> #---- Global Variable Assignments -----
> incomeDataSetFileName <- 'MedianZIP_2_2.xlsx'
>
> #---- Load Required Packages -----
> if(!require("devtools")) {install.packages("devtools")}
> devtools::install_github("dkahle/ggmap")
Skipping install of 'ggmap' from a github remote, the SHA1 (a9455693) has not changed since last
  Use `force = TRUE` to force installation
>
> if(!require("readxl")) {install.packages("readxl")}
> if(!require("gdata")) {install.packages("gdata")}
> if(!require("ggplot2")){install.packages("ggplot2")}
> if(!require("ggmap")){install.packages("ggmap")}
> if(!require("mapproj")){install.packages("mapproj")}
> if(!require("dplyr")) {install.packages("dplyr")}
> if(!require("sqldf")) {install.packages("sqldf")}
> if(!require("zipcode")) {install.packages("zipcode")}
> if(!require("reshape2")) {install.packages("reshape2")}
>
> # Register Google API
> register_google(key="AIzaSyDdjikuUmlpQunYJxtMYEdEq5o32QJgJ28")
>
> #---- Step 1: Load the data -----
>
> ## 1.1: Read the data:
> readDataSetasXLSX <- function(fName){
+   ds <- read_xlsx(fName)
+   return(data.frame(ds))
+ }
>
> income.df <- readDataSetasXLSX(incomeDataSetFileName)
New names:
* ` ` -> ...2
* ` ` -> ...3
* ` ` -> ...4
> str(income.df)
'data.frame': 32635 obs. of 4 variables:
 $ Data.from..http...www.psc.isr.umich.edu.dis.census.Features.tract2zip.: chr "Zip" "1001" "1
 $ ...2 : chr "Median" "56662
 $ ...3 : chr "Mean" "66687.7
 $ ...4 : chr "Pop" "16445" "
> head(income.df)
Data.from..http...www.psc.isr.umich.edu.dis.census.Features.tract2zip. ...2

```

```

1
2
3
4
5
6
>
> ## 1.2: Clean the dataframe:
> cleanIncomeDf <- function(ds){
+   ## Remove Columns
+   colnames(ds) <- NULL
+   ds <- ds[-1,]
+
+   ## Rename Columns
+   newColnames <- c('zip', 'median', 'mean', 'population')
+   colnames(ds) <- newColnames
+
+   ## Remove commas and make numeric
+   ds$median <- as.numeric(gsub(",", "", ds$median))
+   ds$mean <- as.numeric(gsub(",", "", ds$mean))
+   ds$population <- as.numeric(gsub(",", "", ds$population))
+
+   return(ds)
+ }
>
> income.df <- cleanIncomeDf(income.df)
Warning message:
In cleanIncomeDf(income.df) : NAs introduced by coercion
> str(income.df)
'data.frame': 32634 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 ...
> head(income.df)
  zip median mean population
2 1001 56662.57 66687.75 16445
3 1002 49853.42 75062.63 28069
4 1003 28462.00 35121.00 8491
5 1005 75423.00 82442.00 4798
6 1007 79076.35 85801.98 12962
7 1008 63980.00 78391.00 1244
>
> ## 1.3: Load the 'zipcode' package:
> data(zipcode)
> head(zipcode)
  zip city state latitude longitude
1 00210 Portsmouth NH 43.0059 -71.0132
2 00211 Portsmouth NH 43.0059 -71.0132
3 00212 Portsmouth NH 43.0059 -71.0132
4 00213 Portsmouth NH 43.0059 -71.0132
5 00214 Portsmouth NH 43.0059 -71.0132
6 00215 Portsmouth NH 43.0059 -71.0132
>
> ## Reformat zip codes
> income.df$zip <- clean.zipcodes(income.df$zip)
> head(income.df$zip)
[1] "01001" "01002" "01003" "01005" "01007" "01008"
>
> ## 1.4: Merge the zip code information from the two data frames (merge into one dataframe)
> income.by.zipcode.df <- merge(income.df, zipcode, by='zip')
> head(income.by.zipcode.df)
  zip median mean population city state latitude longitude
1 01001 56662.57 66687.75 16445 Agawam MA 42.07061 -72.62029
2 01002 49853.42 75062.63 28069 Amherst MA 42.37765 -72.50323
3 01003 28462.00 35121.00 8491 Amherst MA 42.36956 -72.63599
4 01005 75423.00 82442.00 4798 Barre MA 42.41209 -72.10443
5 01007 79076.35 85801.98 12962 Belchertown MA 42.27842 -72.41100
6 01008 63980.00 78391.00 1244 Blandford MA 42.17431 -72.94828
>
> ## 1.5: Remove the Hawaii and Alaska (just focus on the 'lower 48' states)

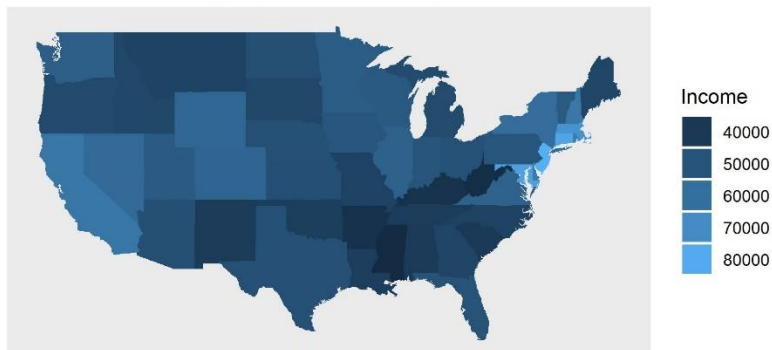
```

```

> income.by.zipcode.df <- income.by.zipcode.df[income.by.zipcode.df$state != 'HI',]
> income.by.zipcode.df <- income.by.zipcode.df[income.by.zipcode.df$state != 'AK',]
> income.by.zipcode.df <- income.by.zipcode.df[income.by.zipcode.df$state != 'DC',]
>
>
> #---- Step 2: Show the income & population per state -----
>
> ## 2.1: Create a simpler dataframe, with just the average median income and the population for each state
> # Average Median Income
> income <- tapply(income.by.zipcode.df$median, income.by.zipcode.df$state, mean)
> state <- rownames(income)
> median.income <- data.frame(state,income)
>
> # Population for each state
> pop <- tapply(income.by.zipcode.df$population, income.by.zipcode.df$state, sum)
> state <- rownames(pop)
> state.pop <- data.frame(state,pop)
>
> # Merge the above two data frames by 'state'
> income.by.state.simp.df <- merge(median.income,state.pop,by="state")
> head(income.by.state.simp.df)
  state  income      pop
1    AL 40549.90 4770242
2    AR 36960.95 2936699
3    AZ 48132.06 6360679
4    CA 62628.71 36927999
5    CO 56303.02 4979279
6    CT 78520.16 3548308
>
> # Alternative method of simplifying data frame
> incomeByZipDf <- income.by.zipcode.df
> incomeByStatesSimpAltDf <- sqldf("select state, avg(median) as income, sum(population) as pop from incomeByZipDf group by state")
> #incomeByStatesSimpAltDf <- sqldf("select state, (income/pop) as income, pop from incomeByStatesSimpAltDf")
>
> ## 2.2: Add the state abbreviations and the state names as new columns (make sure the state names are in lowercase)
> income.by.state.simp.df$state_name <- tolower(state.name[match(income.by.state.simp.df$state,state.name)])
>
> ## 2.3: Show the U.S. map, representing the color with the average median income of that state
> removeThemeAxis <- theme(
+   axis.text = element_blank(),
+   axis.line = element_blank(),
+   axis.ticks = element_blank(),
+   panel.border = element_blank(),
+   panel.grid = element_blank(),
+   axis.title = element_blank()
+ )
>
> us.map <- map_data("state")
> g.map.income <- ggplot(data=income.by.state.simp.df, mapping=aes(map_id=state_name))
> g.map.income <- g.map.income + geom_map(map=us.map, mapping = aes(fill=income))
> g.map.income <- g.map.income + expand_limits(x=us.map$long, y=us.map$lat)
>
> g.map.income <- g.map.income + coord_map()
> g.map.income <- g.map.income + ggtitle("Average Median Income by State") + theme(plot.title=element_text(size=14))
> g.map.income <- g.map.income + guides(fill=guide_legend(title="Income")) + removeThemeAxis
> g.map.income
> ggsave("U.S._Map_of_Average_Median_Income_by_State.jpg", width = 6, height = 6)

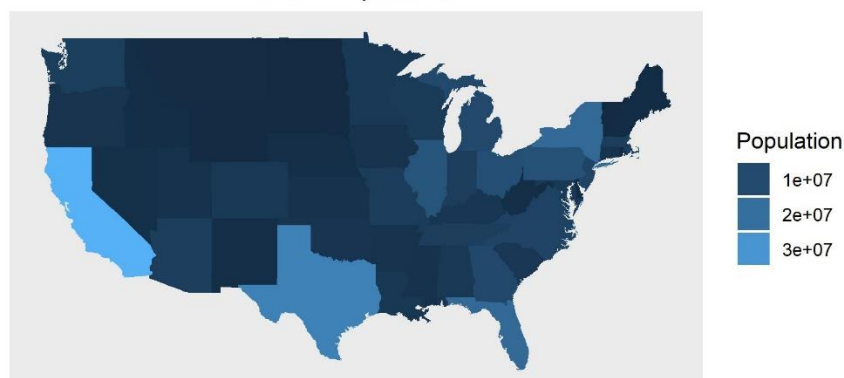
```

Average Median Income by State



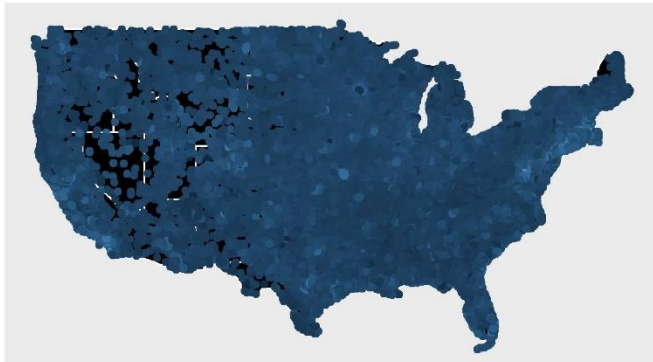
```
>
> ## 2.4: Create a second map with color representing the population of the state
> g.map.pop <- ggplot(data=income.by.state.simp.df, mapping=aes(map_id=state_name))
> g.map.pop <- g.map.pop + geom_map(map=us.map, mapping = aes(fill=pop))
> g.map.pop <- g.map.pop + expand_limits(x=us.map$long, y=us.map$lat)
>
> g.map.pop <- g.map.pop + coord_map()
> g.map.pop <- g.map.pop + ggtitle("State Population") + theme(plot.title=element_text(hjust=0.
> g.map.pop <- g.map.pop + guides(fill=guide_legend(title="Population")) + removeThemeAxis
> g.map.pop
> ggsave("U.S._Map_of_Population_by_State.jpg", width = 6, height = 6)
```


State Population

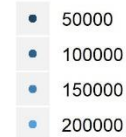


```
>
> #---- Step 3: Show the income per zip code -----
>
> ## 3.1: Draw each zipcode on the map, where the color of the 'dot' is based on the median income
> # To make the map look appealing, have the background of the map be black.
> income.by.zipcode.df$state_name <- tolower(state.name[match(income.by.zipcode.df$state,state.name)])
> head(income.by.zipcode.df)
  zip    median    mean population    city state latitude longitude state_name
1 01001 56662.57 66687.75     16445   Agawam    MA 42.07061 -72.62029 massachusetts
2 01002 49853.42 75062.63     28069   Amherst    MA 42.37765 -72.50323 massachusetts
3 01003 28462.00 35121.00       8491   Amherst    MA 42.36956 -72.63599 massachusetts
4 01005 75423.00 82442.00       4798    Barre    MA 42.41209 -72.10443 massachusetts
5 01007 79076.35 85801.98     12962 Belchertown MA 42.27842 -72.41100 massachusetts
6 01008 63980.00 78391.00      1244  Blandford    MA 42.17431 -72.94828 massachusetts
>
> g.map.zip <- ggplot(data=income.by.zipcode.df, mapping=aes(map_id=state_name))
> g.map.zip <- g.map.zip + geom_map(map=us.map, fill="black", color="white")
> g.map.zip <- g.map.zip + expand_limits(x=us.map$long, y=us.map$lat)
>
> g.map.zip <- g.map.zip + geom_point(data=income.by.zipcode.df, mapping=aes(x=income.by.zipcode.df$zip,
  y=income.by.zipcode.df$median))
> g.map.zip <- g.map.zip + coord_map()
> g.map.zip <- g.map.zip + ggtitle("Income per Zip Code") + theme(plot.title=element_text(hjust=0))
> g.map.zip <- g.map.zip + guides(color=guide_legend(title="Median Income"))
> g.map.zip <- g.map.zip + removeThemeAxis
> g.map.zip
> ggsave("U.S._Map_of_Median_Income_by_ZipCode.jpg", width = 6, height = 6)
```

Income per Zip Code

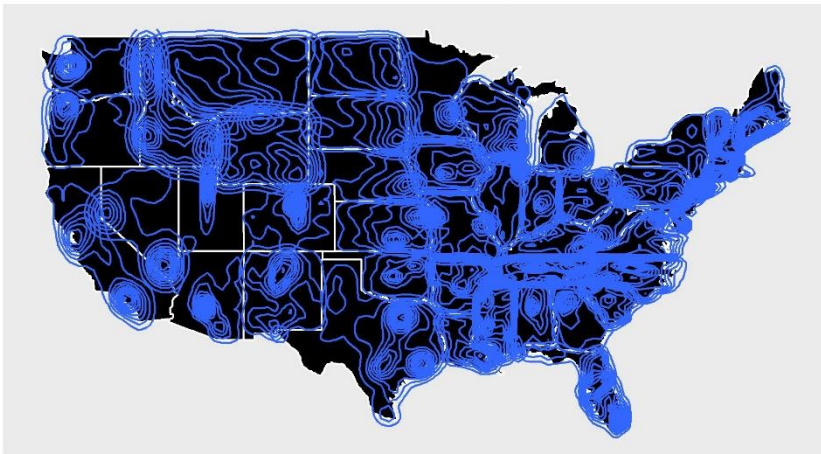


Median Income



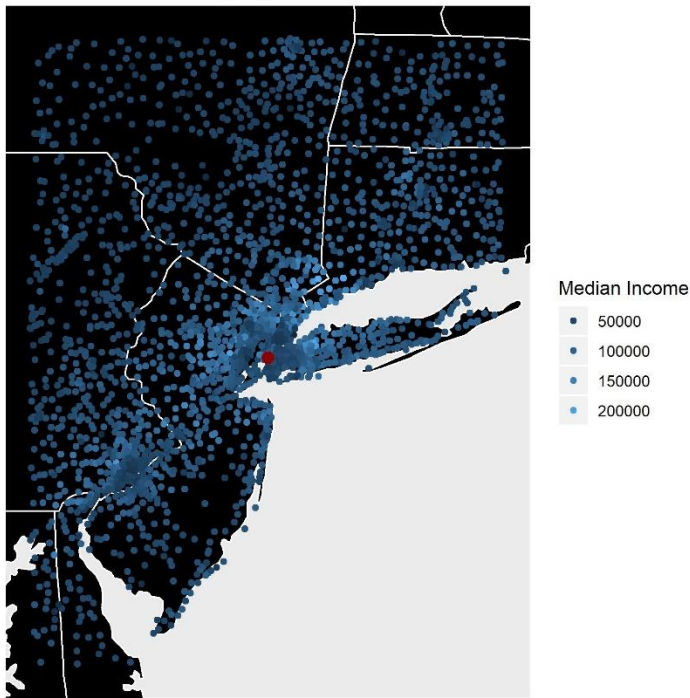
```
> #---- Step 4: Show Zip Code Density -----
>
> ## 4.1: Now generate a different map, one where we can easily see where there are lots of zip
> #       and where there are few (using the 'stat_density2d' function)
> g.map.zip.density <- ggplot(data=income.by.zipcode.df, mapping=aes(map_id=state_name))
> g.map.zip.density <- g.map.zip.density + geom_map(map=us.map, fill="black", color="white")
> g.map.zip.density <- g.map.zip.density + expand_limits(x=us.map$long, y=us.map$lat)
>
> g.map.zip.density <- g.map.zip.density + stat_density_2d(data=income.by.zipcode.df, mapping=aes(longitude, latitude))
> g.map.zip.density <- g.map.zip.density + coord_map()
> g.map.zip.density <- g.map.zip.density + ggtitle("Zip Code Density") + theme(plot.title=element_text(size=16))
> g.map.zip.density <- g.map.zip.density + removeThemeAxis()
> g.map.zip.density
> ggsave("U.S._Map_of_ZipCode_Density.jpg", width = 6, height = 6)
```

Zip Code Density



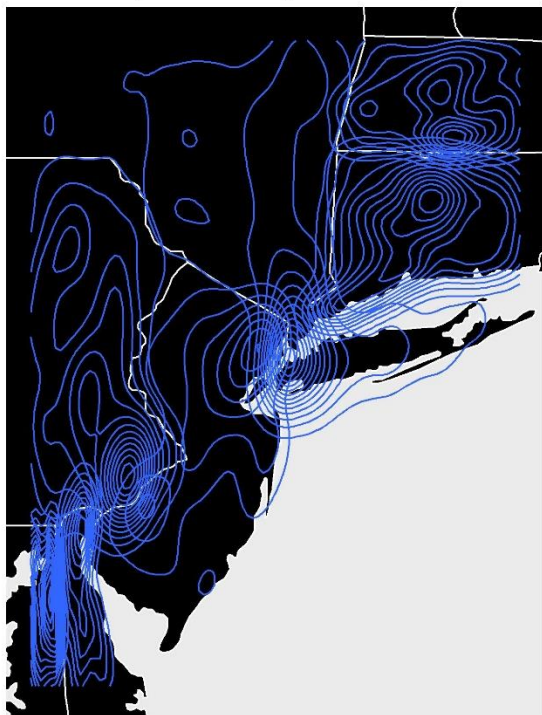
```
> #---- Step 5: Zoom in to the region around NYC -----
>
> ## 5.1: Repeat stes 3 & 4, but have the image / map be of the northeast U.S. (Centered around
> nyc <- geocode("New York, NY", source = "dsk")
> zoom <- 2
> #ggmap(get_map(nyc, zoom=4))
>
> center_x <- nyc$lon
> center_y <- nyc$lat
>
> y_limit <- c(center_y-zoom, center_y+zoom)
> x_limit <- c(center_x-zoom, center_x+zoom)
>
> ## 5.1.1: Draw each zipcode on the map, where the color of the 'dot' is based on the median i
> #       To make the map look appealing, have the background of the map be black.
> g.map.nyc.zip.income <- g.map.zip + xlim(x_limit) + ylim(y_limit) + coord_map()
Coordinate system already present. Adding new coordinate system, which will replace the existin
> g.map.nyc.zip.income <- g.map.nyc.zip.income + geom_point(aes(x=center_x, y=center_y), color=
> g.map.nyc.zip.income <- g.map.nyc.zip.income + ggtitle("Income by Zip around NYC") + theme(pl
> g.map.nyc.zip.income
Warning message:
Removed 29877 rows containing missing values (geom_point).
> ggsave("U.S._Map_of_ZipCode_NYC.jpg", width = 6, height = 6)
Warning message:
Removed 29877 rows containing missing values (geom_point).
```

Income by Zip around NYC



```
>
> ## 5.1.2: Now generate a different map, one where we can easily see where there are lots of z
> # and where there are few (using the 'stat_density2d' function)
> g.map.nyc.zip.density <- g.map.zip.density + xlim(x_limit) + ylim(y_limit) + coord_map()
Coordinate system already present. Adding new coordinate system, which will replace the existin
> g.map.nyc.zip.density <- g.map.nyc.zip.density + stat_density_2d(aes(x=center_x, y=center_y),
> g.map.nyc.zip.density <- g.map.nyc.zip.density + ggtitle("Zip Code Density around NYC") + the
> g.map.nyc.zip.density
Warning messages:
1: Removed 29877 rows containing non-finite values (stat_density2d).
2: Computation failed in `stat_density2d()`:
bandwidths must be strictly positive
>
> ggsave("U.S._Map_of_NYC_ZipCode_Density.jpg", width = 6, height = 6)
Warning messages:
1: Removed 29877 rows containing non-finite values (stat_density2d).
2: Computation failed in `stat_density2d()`:
bandwidths must be strictly positive
```

Zip Code Density around NYC



>