# Homework #7 – Viz Map HW: Median Income

## R Script (Code)

#

# Course: IST687

# Name: Joyce Woznica

# Homework 7 - Viz Map HW: Median Income

# Due Date: 2/26/2019

# Date Submitted:

#

# Step 1: Load the data

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

# I elected to use the "readxl" package

install.packages("readxl")

library(readxl)

medIncomeDF <- read\_excel("C:/Users/Joyce/Desktop/Syracuse/IST687/Submissions/Week7/MedianZIP\_2\_2\_2\_2\_2\_2\_2.xlsx")

# 2) Clean up dataframe

# a) remove information at the front of the file not needed

# b) update column names

newColNames <- c("zip","Median","Mean","Population")

colnames(medIncomeDF)<-newColNames

medIncomeDF <- medIncomeDF[-1,]

# 3) Load the zipcode package

install.packages("zipcode")

library(zipcode)

data(zipcode)

medIncomeDF$zip <- clean.zipcodes(medIncomeDF$zip)

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

NewDF <- merge(medIncomeDF, zipcode, by="zip")

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

NewDF <- subset(NewDF, NewDF$state!="HI")

NewDF <- subset(NewDF, NewDF$state!="AK")

str(NewDF)

# now make the numerics as they should be

NewDF$Mean<-as.numeric(NewDF$Mean)

NewDF$Median<-as.numeric(NewDF$Median)

NewDF$Population<-as.numeric(NewDF$Population)

# Step 2: Show the income and popultation per state

# 1) Create a simpler dataframe with just the average median income and population

# by state

# calc mean of median by state

incomeDF <- tapply(NewDF$Median, NewDF$state, mean)

# place rownames from income into state variable

state <- rownames(incomeDF)

# mean Median Income by State

# create a df with state variable & income variable

medianIncome <- data.frame(state, incomeDF)

# sum up population for each state

pop <- tapply(NewDF$Population, NewDF$state, sum )

# create new df statePop

statePopDF <- data.frame(state, pop)

# create new df by merging df's medianIncome, staeIncome

dfSimple <- merge(medianIncome, statePopDF, by="state")

# 2) Add the state abbreviations and the state names as new columns

# (make sure the state names are lower case)

dfSimple$stateName <- state.name[match(dfSimple$state,state.abb)]

dfSimple$stateName <- tolower(dfSimple$stateName)

# 3) Show the US map, representing the color with the average median income by state

us <- map\_data("state")

mapIncome <- ggplot(dfSimple, aes(map\_id = stateName))

mapIncome <- mapIncome + geom\_map(map = us, aes(fill = dfSimple$income))

mapIncome <- mapIncome + expand\_limits(x = us$long, y = us$lat)

mapIncome <- mapIncome + coord\_map()

mapIncome <- mapIncome + ggtitle("Average Median Income by State in the US")

mapIncome

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

mapPop <- ggplot(dfSimple, aes(map\_id = stateName))

mapPop <- mapPop + geom\_map(map = us, aes(fill = dfSimple$pop))

mapPop <- mapPop + expand\_limits(x = us$long, y = us$lat)

mapPop <- mapPop + coord\_map()

mapPop <- mapPop + ggtitle("Population by State in the US")

mapPop

# Step 3: Show the income per zipcode

# 1) Have 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.

# first need to add the state name to our NewDF dataframe

NewDF$stateName <- state.name[match(NewDF$state,state.abb)]

NewDF$stateName <- tolower(NewDF$stateName)

NewDF <- subset(NewDF, NewDF$state!="hawaii")

NewDF <- subset(NewDF, NewDF$state!="alaska")

# now complete the zip map

mapZip <- ggplot(NewDF, aes(map\_id = stateName))

mapZip <- mapZip + geom\_map(map=us, fill="black", color="white")

mapZip <- mapZip + expand\_limits(x =us$long, y = us$lat)

mapZip <- mapZip + geom\_point(data = NewDF,aes(x = NewDF$longitude,

y = NewDF$latitude, color=NewDF$Median))

mapZip <- mapZip + coord\_map()

mapZip <- mapZip + ggtitle("Income by Zipcode")

mapZip

# 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 just a few

# (using the stat\_density2d function)

mapDensity <- mapZip + geom\_density\_2d(data = NewDF,

aes(x = NewDF$longitude, y = NewDF$latitude))

mapDensity

# Step 5: Zoom into the region around New York City

# 1) Repeat steps 3 & 4, but have have the image/map only be of the

# the northeast US (centered around NYC)

# this section assumes that I have loaded the "NewLatLon" function (provided later)

latlon <- NewLatLon("NYC, ny")

# now complete the zooming

mapZipZoomed <- mapZip + geom\_point(aes(x = latlon$lon, y = latlon$lat),

color="darkred", size = 3)

mapZipZoomed <- mapZipZoomed + xlim(latlon$lon-10, latlon$lon+10) +

ylim(latlon$lat-10,latlon$lat+10)

mapZipZoomed <- mapZipZoomed + coord\_map()

mapZipZoomed <- mapZipZoomed + ggtitle("Income by Zipcode in the Northeast US")

mapZipZoomed

# repeating from Step 4

mapDensityZoomed <- mapDensity + geom\_point(aes(x = latlon$lon, y=latlon$lat),

color="darkred", size = 3)

mapDensityZoomed <- mapDensityZoomed + xlim(latlon$lon-10, latlon$lon+10) +

ylim(latlon$lat-10, latlon$lat+10)

mapDensityZoomed <- mapDensityZoomed + coord\_map()

mapDensityZoomed <- mapDensityZoomed + ggtitle("Density Map - Income by Zipcode in the Northeast US")

mapDensityZoomed

## Console Log (Executed Code)

> #

> # Course: IST687

> # Name: Joyce Woznica

> # Homework 7 - Viz Map HW: Median Income

> # Due Date: 2/26/2019

> # Date Submitted:

> #

> # Step 1: Load the data

> # 1) Read the data - using the gdata package we have previously used

> # I elected to use the "readxl" package

> install.packages("readxl")

Error in install.packages : Updating loaded packages

> library(readxl)

> install.packages("readxl")

Installing package into ‘C:/Users/Joyce/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

Warning in install.packages :

package ‘readxl’ is in use and will not be installed

> medIncomeDF <- read\_excel("C:/Users/Joyce/Desktop/Syracuse/IST687/Submissions/Week7/MedianZIP\_2\_2\_2\_2\_2\_2\_2.xlsx")

New names:

\* `` -> `..2`

\* `` -> `..3`

\* `` -> `..4`

>

> # 2) Clean up dataframe

> # a) remove information at the front of the file not needed

> # b) update column names

> newColNames <- c("zip","Median","Mean","Population")

> colnames(medIncomeDF)<-newColNames

> medIncomeDF <- medIncomeDF[-1,]

> # 3) Load the zipcode package

> install.packages("zipcode")

Error in install.packages : Updating loaded packages

> library(zipcode)

> install.packages("zipcode")

Installing package into ‘C:/Users/Joyce/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

Warning in install.packages :

package ‘zipcode’ is in use and will not be installed

> data(zipcode)

> medIncomeDF$zip <- clean.zipcodes(medIncomeDF$zip)

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

> NewDF <- merge(medIncomeDF, zipcode, by="zip")

>

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

> NewDF <- subset(NewDF, NewDF$state!="HI")

> NewDF <- subset(NewDF, NewDF$state!="AK")

> # now make the numerics as they should be

> NewDF$Mean<-as.numeric(NewDF$Mean)

Warning message:

NAs introduced by coercion

> NewDF$Median<-as.numeric(NewDF$Median)

> NewDF$Population<-as.numeric(NewDF$Population)

> # Step 2: Show the income and popultation per state

> # 1) Create a simpler dataframe with just the average median income and population

> # by state

> # calc mean of median by state

> incomeDF <- tapply(NewDF$Median, NewDF$state, mean)

> # place rownames from income into state variable

> state <- rownames(incomeDF)

> # mean Median Income by State

> # create a df with state variable & income variable

> medianIncome <- data.frame(state, incomeDF)

> # sum up population for each state

> pop <- tapply(NewDF$Population, NewDF$state, sum )

>

> # create new df statePop

> statePopDF <- data.frame(state, pop)

> # create new df by merging df's medianIncome, staeIncome

> dfSimple <- merge(medianIncome, statePopDF, by="state")

> # 2) Add the state abbreviations and the state names as new columns

> # (make sure the state names are lower case)

> dfSimple$stateName <- state.name[match(dfSimple$state,state.abb)]

> dfSimple$stateName <- tolower(dfSimple$stateName)

> # 3) Show the US map, representing the color with the average median income by state

> us <- map\_data("state")

> mapIncome <- ggplot(dfSimple, aes(map\_id = stateName))

> mapIncome <- mapIncome + geom\_map(map = us, aes(fill = dfSimple$income))

> mapIncome <- mapIncome + expand\_limits(x = us$long, y = us$lat)

> mapIncome <- mapIncome + coord\_map()

> mapIncome <- mapIncome + ggtitle("Average Median Income by State in the US")

> mapIncome

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

> mapPop <- ggplot(dfSimple, aes(map\_id = stateName))

> mapPop <- mapPop + geom\_map(map = us, aes(fill = dfSimple$pop))

> mapPop <- mapPop + expand\_limits(x = us$long, y = us$lat)

> mapPop <- mapPop + coord\_map()

> mapPop <- mapPop + ggtitle("Population by State in the US")

> mapPop

> # Step 3: Show the income per zipcode

> # 1) Have 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.

>

> # first need to add the state name to our NewDF dataframe

> NewDF$stateName <- state.name[match(NewDF$state,state.abb)]

> NewDF$stateName <- tolower(NewDF$stateName)

> NewDF <- subset(NewDF, NewDF$state!="hawaii")

> NewDF <- subset(NewDF, NewDF$state!="alaska")

> # now complete the zip map

> mapZip <- ggplot(NewDF, aes(map\_id = stateName))

> mapZip <- mapZip + geom\_map(map=us, fill="black", color="white")

> mapZip <- mapZip + expand\_limits(x =us$long, y = us$lat)

> mapZip <- mapZip + geom\_point(data = NewDF,aes(x = NewDF$longitude,

+ y = NewDF$latitude, color=NewDF$Median))

> mapZip <- mapZip + coord\_map()

> mapZip <- mapZip + ggtitle("Income by Zipcode")

> mapZip

> # 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 just a few

> # (using the stat\_density2d function)

> mapDensity <- mapZip + geom\_density\_2d(data = NewDF,

+ aes(x = NewDF$longitude, y = NewDF$latitude))

> mapDensity

> # Step 5: Zoom into the region around New York City

> # 1) Repeat steps 3 & 4, but have have the image/map only be of the

> # the northeast US (centered around NYC)

> # this section assumes that I have loaded the "NewLatLon" function (provided later)

> ## geocoding function using OSM Nominatim API

> ## details: http://wiki.openstreetmap.org/wiki/Nominatim

> ## made by: D.Kisler

>

> install.packages("jsonlite")

Error in install.packages : Updating loaded packages

> install.packages("tidyverse")

Error in install.packages : Updating loaded packages

>

> library(jsonlite)

> library(tidyverse)

>

> nominatim\_osm <- function(address = NULL)

+ {

+ if(suppressWarnings(is.null(address)))

+ return(data.frame())

+ tryCatch(

+ d <- jsonlite::fromJSON(

+ gsub('\\@addr\\@', gsub('\\s+', '\\%20', address),

+ 'http://nominatim.openstreetmap.org/search/@addr@?format=json&addressdetails=0&limit=1')

+ ), error = function(c) return(data.frame())

+ )

+ if(length(d) == 0) return(data.frame())

+ return(data.frame(lon = as.numeric(d$lon), lat = as.numeric(d$lat)))

+ }

>

> #dplyr will be used to stack lists together into a data.frame and to get the pipe operator '%>%'

> suppressPackageStartupMessages(library(dplyr))

>

> NewLatLon<-function(addresses){

+ d <- suppressWarnings(lapply(addresses, function(address) {

+ #set the elapsed time counter to 0

+ t <- Sys.time()

+ #calling the nominatim OSM API

+ api\_output <- nominatim\_osm(address)

+ #get the elapsed time

+ t <- difftime(Sys.time(), t, 'secs')

+ #return data.frame with the input address, output of the nominatim\_osm function and elapsed time

+ return(data.frame(address = address, api\_output, elapsed\_time = t))

+ }) %>%

+ #stack the list output into data.frame

+ bind\_rows() %>% data.frame())

+ #output the data.frame content into console

+ return(d)

+ }

> install.packages("jsonlite")

Installing package into ‘C:/Users/Joyce/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

Warning in install.packages :

package ‘jsonlite’ is in use and will not be installed

> install.packages("tidyverse")

Installing package into ‘C:/Users/Joyce/Documents/R/win-library/3.5’

(as ‘lib’ is unspecified)

Warning in install.packages :

package ‘tidyverse’ is in use and will not be installed

> latlon <- NewLatLon("NYC, ny")

> # now complete the zooming

> mapZipZoomed <- mapZip + geom\_point(aes(x = latlon$lon, y = latlon$lat),

+ color="darkred", size = 3)

> mapZipZoomed <- mapZipZoomed + xlim(latlon$lon-10, latlon$lon+10) +

+ ylim(latlon$lat-10,latlon$lat+10)

> mapZipZoomed <- mapZipZoomed + coord\_map()

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

> mapZipZoomed <- mapZipZoomed + ggtitle("Income by Zipcode in the Northeast US")

> mapZipZoomed

Warning message:

Removed 20941 rows containing missing values (geom\_point).

> # repeating from Step 4

> mapDensityZoomed <- mapDensity + geom\_point(aes(x = latlon$lon, y=latlon$lat),

+ color="darkred", size = 3)

> mapDensityZoomed <- mapDensityZoomed + xlim(latlon$lon-10, latlon$lon+10) +

+ ylim(latlon$lat-10, latlon$lat+10)

> mapDensityZoomed <- mapDensityZoomed + coord\_map()

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

> mapDensityZoomed <- mapDensityZoomed + ggtitle("Density Map - Income by Zipcode in the Northeast US")

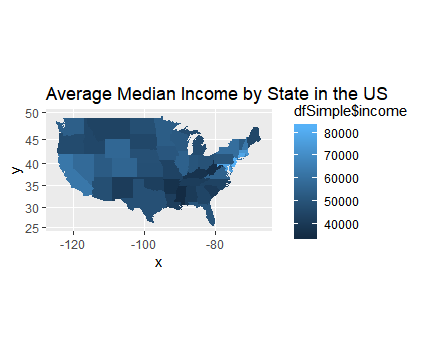
> mapDensityZoomed

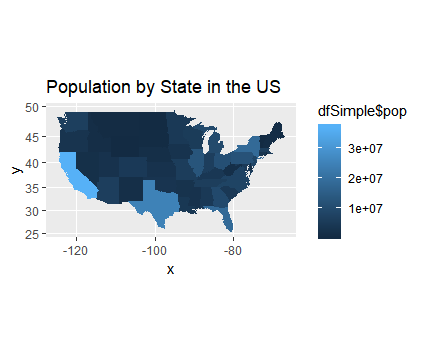
Warning messages:

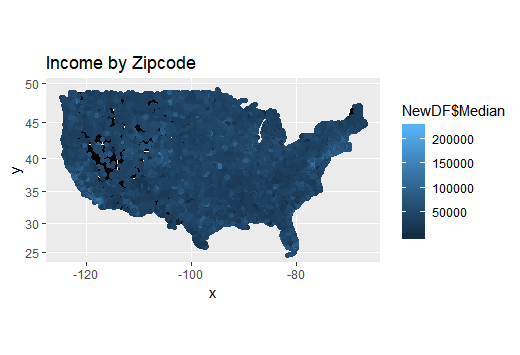
1: Removed 20941 rows containing non-finite values (stat\_density2d).

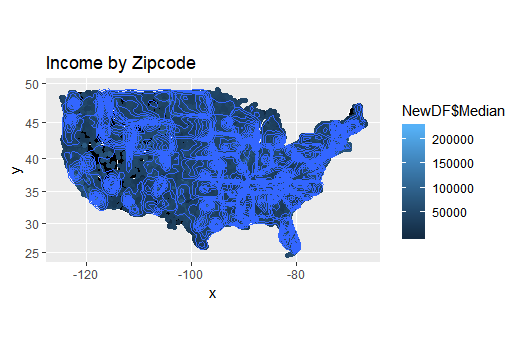
2: Removed 20941 rows containing missing values (geom\_point).

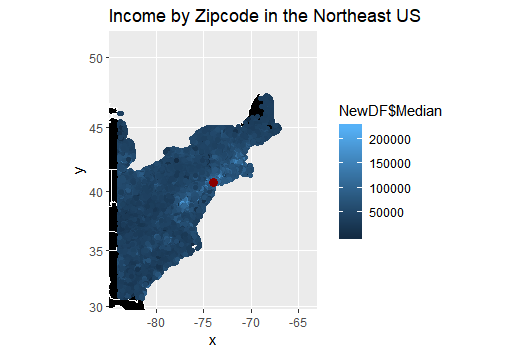
## Visualizations

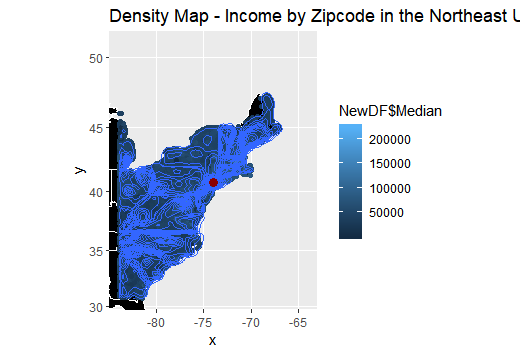












## New Lat Lon Function to replace geocode

## geocoding function using OSM Nominatim API

## details: http://wiki.openstreetmap.org/wiki/Nominatim

## made by: D.Kisler

install.packages("jsonlite")

install.packages("tidyverse")

library(jsonlite)

library(tidyverse)

nominatim\_osm <- function(address = NULL)

{

if(suppressWarnings(is.null(address)))

return(data.frame())

tryCatch(

d <- jsonlite::fromJSON(

gsub('\\@addr\\@', gsub('\\s+', '\\%20', address),

'http://nominatim.openstreetmap.org/search/@addr@?format=json&addressdetails=0&limit=1')

), error = function(c) return(data.frame())

)

if(length(d) == 0) return(data.frame())

return(data.frame(lon = as.numeric(d$lon), lat = as.numeric(d$lat)))

}

#dplyr will be used to stack lists together into a data.frame and to get the pipe operator '%>%'

suppressPackageStartupMessages(library(dplyr))

NewLatLon<-function(addresses){

d <- suppressWarnings(lapply(addresses, function(address) {

#set the elapsed time counter to 0

t <- Sys.time()

#calling the nominatim OSM API

api\_output <- nominatim\_osm(address)

#get the elapsed time

t <- difftime(Sys.time(), t, 'secs')

#return data.frame with the input address, output of the nominatim\_osm function and elapsed time

return(data.frame(address = address, api\_output, elapsed\_time = t))

}) %>%

#stack the list output into data.frame

bind\_rows() %>% data.frame())

#output the data.frame content into console

return(d)

}