LewisDB\_HW7.R

dblewis

Fri Mar 01 10:16:09 2019

#############################################################################  
#Name: Daniel Lewis  
#Description: Homework Assignment 7  
#Date: 3/1/19 (JST)  
#input: MedianZIP-3.csv  
#output: report of code including charts  
#update history:  
#############################################################################  
#####################HW7: Viz Map HW: Median Income##########################  
#############################################################################  
#  
#############################################################################  
############################LOCAL FUNCTIONS##################################  
#############################################################################  
#  
EnsurePackage<-function(x){  
 x<-as.character(x)  
 if (!require(x,character.only=TRUE)){  
 install.packages(pkgs=x, repos="http://cran.r-project.org")  
 require(x, character.only=TRUE)  
 }  
}  
Numberize <- function(inputVector){  
 inputVector <-gsub(",","",inputVector)  
 inputVector <- gsub(" ", "", inputVector)  
 inputVector <- as.numeric(inputVector)  
}  
#  
#############################################################################  
#############################IMPORTS SECTION#################################  
#############################################################################  
#Packages#  
EnsurePackage("compare")

## Loading required package: compare

##   
## Attaching package: 'compare'

## The following object is masked from 'package:base':  
##   
## isTRUE

EnsurePackage("ggmap")

## Loading required package: ggmap

## Loading required package: ggplot2

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

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

EnsurePackage("ggplot2")  
EnsurePackage("mapproj")

## Loading required package: mapproj

## Loading required package: maps

EnsurePackage("sqldf")

## Loading required package: sqldf

## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite

EnsurePackage("stringr")

## Loading required package: stringr

EnsurePackage("reprex")

## Loading required package: reprex

EnsurePackage("zipcode")

## Loading required package: zipcode

#DataSets#  
###Works on my windows 10 and mac machines by hitting the source button#  
###[sets working directory to the script folder]#  
#this.dir <- dirname(parent.frame(2)$ofile) #Commented out to generate report, uncomment to grade  
#setwd(this.dir) #Commented out to generate report, uncomment to grade  
csv\_import<-read.csv("MedianZIP-3.csv", stringsAsFactors = FALSE)  
#  
#############################################################################  
#############################Problems Solved#################################  
#############################################################################  
#  
#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 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)  
#############################################################################  
#  
#remove the non numbers   
csv\_import<-csv\_import[-c(29980,29645,26201,26134,26133,26132,7056)]  
#turn the col names to lowercase#  
colnames(csv\_import)<-tolower(colnames(csv\_import))  
#  
#clean up the data frame#  
csv\_import<-data.frame(sapply(csv\_import, Numberize))

## Warning in FUN(X[[i]], ...): NAs introduced by coercion

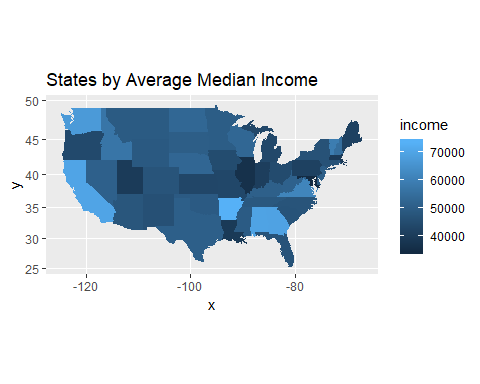
#  
#fix the zip column length#  
csv\_import$zip<-str\_pad(csv\_import$zip, 5, "left","0")  
#  
#generate zipcode data frame#  
data(zipcode)  
#  
#perform a merge based on zipcode col#  
zipcode\_joincsv<-merge(x=zipcode, y=csv\_import, by="zip")  
#  
#sort by abv  
colnames(zipcode\_joincsv)[3]<-"abv"  
zipcode\_joincsv$abv<-sort(zipcode\_joincsv$abv)  
#  
#Restrict to Lower 48 States (excluding all US territories)#  
zipcode\_joincsv<-data.frame(sqldf("SELECT \*  
 FROM zipcode\_joincsv  
 WHERE abv NOT IN ('AK', 'HI', 'DC')", row.names=TRUE))  
#check the data#  
print(head(zipcode\_joincsv))

## zip city abv latitude longitude median mean pop  
## 226 01749 Hudson AL 42.38981 -71.55791 76500 85689 18081  
## 227 01752 Marlborough AL 42.35091 -71.54753 71835 89002 36273  
## 228 01754 Maynard AL 42.43078 -71.45594 76228 82167 10414  
## 229 01756 Mendon AL 42.09201 -71.54474 102625 117692 5257  
## 230 01757 Milford AL 42.14918 -71.52149 68565 82206 26877  
## 231 01760 Natick AL 42.29076 -71.35368 90673 113933 31763

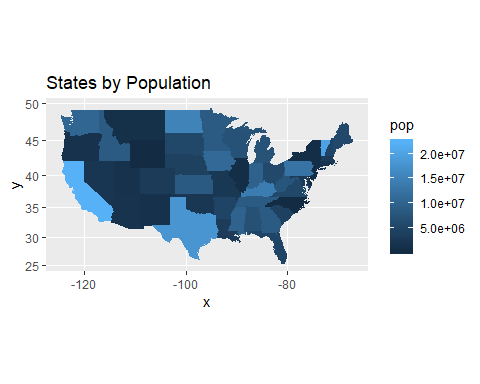
#  
#Step 2: Show the income & population per state  
##1) Create a simpler dataframe, with just the average median income and the the population for each state.  
##2) Add the state abbreviations and the state names as new columns (make sure the state names are all lower case)#  
#create a vector filled with the 50 states, than clip AK and HI  
states<-tolower(state.name)  
states<-states[-2]  
states<-states[-10]  
#generate a vector filled with the sates abreviations  
abv<-state.abb  
abv<-abv[-2]  
abv<-abv[-10]  
#places state and abv into a data frame#  
step2\_df<-data.frame(states, abv)  
#sums the population by states and puts them in the data frame#  
population<-sqldf("SELECT sum(pop) AS 'pop'  
 FROM zipcode\_joincsv   
 GROUP BY abv")  
#generates the medium income by state and stores it into a vector#  
income<-sqldf("SELECT AVG(median) AS 'income'  
 FROM zipcode\_joincsv   
 GROUP BY abv")  
#reformat data frame with all fields#  
step2\_df<-data.frame(states,abv,population,income)  
#check the data#  
print(head(step2\_df))

## states abv pop income  
## 1 alabama AL 6881143 69739.11  
## 2 arizona AZ 1725246 48014.01  
## 3 arkansas AR 4862986 73523.77  
## 4 california CA 22362184 70102.18  
## 5 colorado CO 3273024 47458.86  
## 6 connecticut CT 1990572 46798.04

##3) Show the U.S. map, representing the color with the average median income of that state  
#  
us<-map\_data("state")  
map.incomeColor<-ggplot(step2\_df,aes(map\_id=states))  
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()+ggtitle("States by Average Median Income")  
map.incomeColor



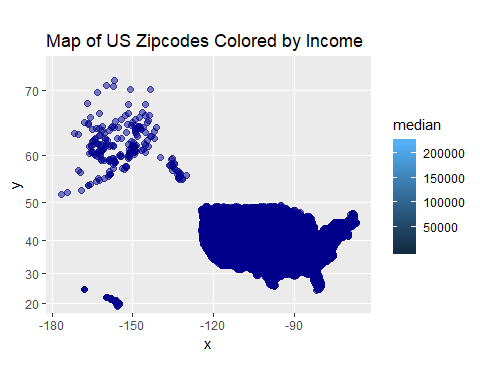
##4) Create a second map with color representing the population of the state  
#  
us<-map\_data("state")  
map.popColor<-ggplot(step2\_df,aes(map\_id=states))  
map.popColor<-map.popColor+ geom\_map(map=us, aes(fill=pop))  
map.popColor<-map.popColor+expand\_limits(x=us$long, y=us$lat)  
map.popColor<-map.popColor+coord\_map()+ggtitle("States by Population")  
map.popColor



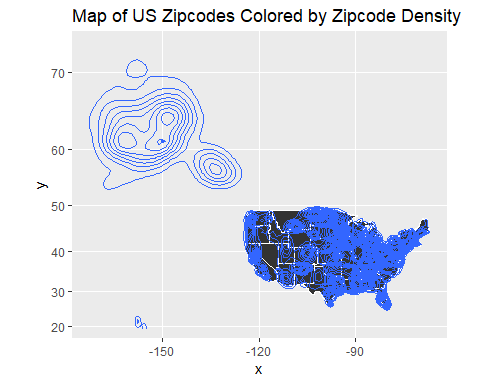
#  
#Step 3: Show the income per zip code  
##1) Have draw each zip code on the map, where the color of the dota is based on the median income. To make the map look appealing, have the background of the map be black.  
#  
#Create a frame of all the data from this hw#  
step3\_df<-data.frame(as.character(tolower(state.name)),as.character(state.abb))  
colnames(step3\_df)<-c("state", "abv")  
step3\_df<-data.frame(sqldf("SELECT \*  
 FROM step3\_df  
 WHERE abv NOT IN ('AK', 'HI', 'DC')", row.names=TRUE))  
step3\_df<-dplyr::left\_join(zipcode\_joincsv, step3\_df, by="abv")

## Warning: Column `abv` joining character vector and factor, coercing into  
## character vector

step3\_df$state<-sapply(step3\_df$state,as.character)  
step3\_df<-step3\_df[,c(9,3,2,1,4,5,6,7,8)]  
rownames(step3\_df)<-NULL  
#  
#show each zipcode on the map where color of dot is based on mean income#  
us<-map\_data("state")  
map.zipIncome<-ggplot(step3\_df, aes(map\_id=state))  
map.zipIncome<-map.zipIncome+geom\_map(map=us, color="white")  
map.zipIncome<-map.zipIncome+expand\_limits(x=us$long, y=us$lat)  
map.zipIncome<-map.zipIncome+coord\_map() + ggtitle("Map of US Zipcodes Colored by Income")  
map.zipIncome <- map.zipIncome + geom\_point(data=step3\_df, aes(x = longitude, y = latitude, fill=median), alpha=.5, size=2, colour="darkblue")  
map.zipIncome



#  
#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\_density2 function).  
#  
step4\_df<-step3\_df  
us<-map\_data("state")  
map.zipDensity<-ggplot(step4\_df, aes(map\_id=state))  
map.zipDensity<-map.zipDensity+geom\_map(map=us, color="white")  
map.zipDensity<-map.zipDensity+expand\_limits(x=us$long, y=us$lat)  
map.zipDensity<-map.zipDensity+coord\_map() + ggtitle("Map of US Zipcodes Colored by Zipcode Density")  
map.zipDensity<-map.zipDensity+stat\_density2d( data=step4\_df, aes(x=longitude, y=latitude, color=median))   
map.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).  
#  
#instructed not to do this during live session  
#  
#END OF SCRIPT