# --------------- HW 7: Viz Map ---------------

# install.packages("gdata")

library(gdata)

# ----------------------------------------------

# Step 1: Load the Data

# 1) Read the data

# read in the excel file using read.xls function in gdata package

mydata <- read.xls("MedianZIP.xlsx")

# 2) clean up the data

# change column names to "zip", "Median","Mean", and "Population"

colnames(mydata) <- c("zip", "Median", "Mean", "Population")

# delete the first row of the dataframe

mydata <- mydata[-1,]

head(mydata)

# gsub function is used to perform replacement of matches determined by regular expression matching

# in this case gsub replace all "," in column "Median" with nothing ("")

mydata$Median <- gsub(",", "", mydata$Median)

# delete the "," in column "Mean" (replace all "," in column "Mean" with nothing)

mydata$Mean <- gsub(",", "", mydata$Mean)

# delete the "," in column "Population" (replace all "," in column "Population" with nothing)

mydata$Population <- gsub(",","",mydata$Population)

head(mydata)

# 3) Load the "zipcodez" package

# install.packages("zipcode")

library(zipcode)

# load a dataframe that contains city, state, latitude, and longitude for U.S. ZIP codes

data(zipcode)

head(zipcode)

# clean up and standardize ZIP codes in mydata

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

# 4) Merge the zip code information

# merge mydata and zipcode by the common column "zip" and store the new dataframe into "dfNew"

dfNew <- merge(mydata, zipcode, by="zip")

head(dfNew)

# 5) Remove Hawaii and Alaska

# remove Hawaii and Alaska from dfNew

dfNew <- dfNew[which(dfNew$state != "AK" & dfNew$state != "HI"), ]

# ----------------------------------------------

# Step 2: Show the income & population per state

# 1) Create a simpler dataframe,with just the average median income and the population for each state

# convert column "Median" into numeric

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

# convert column "Population" into numeric

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

# use aggregate function to calculate average median income for each states and store the results in a new dataframe called "medianIncome"

# aggregate function is used to split the data into subsets and compute summary statistics for each

medianIncome <- aggregate(dfNew$Median, by=list(dfNew$state), FUN=mean)

head(medianIncome)

# use aggregate function to calculate total population for each states and store the results in a new dataframe called "statePop"

# aggregate function is used to split the data into subsets and compute summary statistics for each

statePop <- aggregate(dfNew$Population, by=list(dfNew$state), FUN=sum)

head(statePop)

# change column names of "medianIncome" dataframe

colnames(medianIncome) <- c("states", "MedianIncome")

# change column names of "statePop" dataframe

colnames(statePop) <- c("states", "Population")

# merge "medianIncome" and "statePop" by common column "states"

dfSimple <- merge(medianIncome, statePop, by="states")

head(dfSimple)

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

# install.packages("openintro")

library(openintro)

# Convert abbreviations to state names and store it to a new column in dfSimple

dfSimple$stateName <- abbr2state(dfSimple$states)

# convert "stateName" to lower case

dfSimple$stateName <- tolower(dfSimple$stateName)

head(dfSimple)

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

# install.packages("ggplot2")

library(ggplot2)

# install.packages("ggmap")

library(ggmap)

# get the data on the "state" to be mapped

us <- map\_data("state")

head(us)

# use "dfSimple" to create map and set "stateName" as map ID

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

# represent the color with average median income of each states

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

# change the limits of x and y axes to print the whole map

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

# make sure the map is not stretched

mapIncome <- mapIncome + coord\_map()

# add a tile for the map

mapIncome <- mapIncome + ggtitle("average median Income of the U.S")

# plot the income map

mapIncome

# 4) Show the U.S. map, with color representing the population of the state

# use "dfSimple" to create map and set "stateName" as map ID

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

# represent the color with population of each states

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

# change the limits of x and y axes to print the whole map

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

# make sure the map is not stretched and add a title for the map

mapPop <- mapPop + coord\_map() + ggtitle("Population of the U.S")

# plot the map

mapPop

# ----------------------------------------------

# Step 3: Show the income per zip code

# Convert abbreviations to state names and store it to a new column

dfNew$stateName <- abbr2state(dfNew$state)

# convert "stateName" to lower case

dfNew$stateName <- tolower(dfNew$stateName)

# use dfNew to create map and set "stateName" as map ID

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

# set the backgroud color to be black and line color to be white

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

# change the limits of x and y axes to print the whole map

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

# plot points on map each "dot" represent a zip code and the color of "dots" is based on median income

mapZip <- mapZip + geom\_point(data = dfNew,aes(x = dfNew$longitude, y = dfNew$latitude, color=dfNew$Median))

# make sure the map is not stretched and add a title for the map

mapZip <- mapZip + coord\_map() + ggtitle("Income per zip code")

# plot the map

mapZip

# ----------------------------------------------

# Step 4: Show Zip Code Density

# create the density map based on the existing "mapZip"

# use geom\_density\_2d function to show the density

mapD <- mapZip + geom\_density\_2d(data = dfNew, aes(x = dfNew$longitude, y = dfNew$latitude))

# plot the map

mapD

# ----------------------------------------------

# Step 5: Zoom in to the region around NYC

# use geocode function to get latitude and longtitude of NYC

latlon <- geocode("NYC, ny")

# create the first zoomed map based on "mapZip", and plot a point representing NYC

mapZipZoomed <- mapZip + geom\_point(aes(x = latlon$lon, y = latlon$lat), color="darkred", size = 3)

# zoom into the region arount NYC with 10 degrees latitude and longtitude fluctuation (+/- 10)

mapZipZoomed <- mapZipZoomed + xlim(latlon$lon-10, latlon$lon+10) + ylim(latlon$lat-10,latlon$lat+10) + coord\_map()

# plot the map

mapZipZoomed

# create the first zoomed map based on "mapD" and plot a point, which representing NYC

mapDZoomed <- mapD + geom\_point(aes(x = latlon$lon, y = latlon$lat), color="darkred", size = 3)

# zoom into the region arount NYC (latitude and longtitude +/- 10)

mapDZoomed <- mapDZoomed + xlim(latlon$lon-10,latlon$lon+10) + ylim(latlon$lat-10,latlon$lat+10) + coord\_map()

# plot the map

mapDZoomed