Alonso\_Week 7 Homework Assignment

Martin Alonso

8/30/2018

IST687 Introduction to Data Science: Week 7 Homework.

Map (S)mashing

### Part 1: The Code

# We're loading all the packages that will be needed for this exercise.  
require(ggplot2)  
require(ggmap)  
require(readxl)  
require(gdata)  
require(dplyr)  
require(zipcode)  
  
# Step 1: Load the Data.   
# Load the MedianZIP\_2\_2\_2.xlsx file.  
dat <- read\_excel('MedianZIP\_2\_2\_2.xlsx')  
  
# Let's take a quick survey of the data.   
str(dat)  
head(dat)  
summary(dat)  
  
# Looks like the column names will need to be updated and the second to fourth column will need to be converted to numerical type.  
names <- dat[1, ]  
colnames(dat) <- names  
dat <- dat[-1, ]  
  
numdat <- colnames(dat[, 2:4])  
for(i in numdat){  
 dat[[i]] <- as.numeric(dat[[i]])  
 dat[[i]][is.na(dat[[i]])] <- median(dat[[i]], na.rm = T)  
}  
  
# Let's check once again that the data is correct.  
str(dat)  
  
# Now let's upload the zipcode data, merge the data, and remove Alaska and Hawaii.  
data(zipcode)  
dat$zip <- clean.zipcodes(dat$Zip)  
  
zip\_comp <- merge(dat, zipcode, by = 'zip')  
zip\_comp <- zip\_comp %>%  
 filter(!state %in% c('AK', 'HI'))  
  
  
#Step 2: Show income & population per state.  
med\_dat <- zip\_comp %>%  
 group\_by(state) %>%   
 summarize(mean\_med\_income = mean(Median),   
 tot\_pop = sum(Pop))   
  
# Add the state names and convert to lower case.   
med\_dat$stateName <- state.name[match(med\_dat$state, state.abb)]  
med\_dat$stateName <- tolower(med\_dat$stateName)  
  
us <- map\_data("state")  
  
# Time to map average median income by state. We'll color code the map according to the median income.   
ggplot(med\_dat, aes(map\_id = stateName)) +   
 geom\_map(map = us, aes(fill = med\_dat$mean\_med\_income)) +   
 expand\_limits(x = us$long, y = us$lat) +  
 coord\_map() +  
 labs(x = 'Long', y = 'Lat', fill = 'Income') +   
 ggtitle('Avg median income by state')  
  
# And now we'll do the same by population.   
ggplot(med\_dat, aes(map\_id = stateName)) +   
 geom\_map(map = us, aes(fill = med\_dat$tot\_pop)) +   
 expand\_limits(x = us$long, y = us$lat) +  
 coord\_map() +  
 labs(x = 'Long', y = 'Lat', fill = 'Population') +   
 ggtitle('Population by state')  
  
  
# Step 3: Show income per zipcode.  
zip\_comp$stateName <- state.name[match(zip\_comp$state, state.abb)]  
zip\_comp$stateName <- tolower(zip\_comp$stateName)  
  
ggplot(zip\_comp, aes(map\_id = stateName)) +  
 geom\_map(map = us, fill = 'black', colour = 'white') +   
 expand\_limits(x = us$long, y = us$lat) +  
 geom\_point(data = zip\_comp, aes(zip\_comp$longitude, zip\_comp$latitude, color = zip\_comp$Median)) +  
 coord\_map() +  
 labs(x = 'Long', y = 'Lat', fill = 'Income') +   
 ggtitle('Median Income per Zipcode')  
  
# Step 4: Show zipcode density.  
ggplot(zip\_comp, aes(map\_id = stateName)) +  
 geom\_map(map = us, fill = 'black', colour = 'white') +   
 expand\_limits(x = us$long, y = us$lat) +  
 stat\_density2d(data = zip\_comp, aes(zip\_comp$longitude, zip\_comp$latitude)) +  
 coord\_map() +  
 labs(x = 'Long', y = 'Lat') +   
 ggtitle('Zipcode Density')  
  
# Step 5: Zoom in to the region around NYC.   
zoomnyc <- geocode('New York City, NY')  
  
# Average Median Income by Zipcode for Northeastern US.   
ggplot(zip\_comp, aes(map\_id = stateName)) +  
 geom\_map(map = us, fill = 'black', colour = 'white') +  
 xlim(zoomnyc$lon - 5, zoomnyc$lon + 5) +  
 ylim(zoomnyc$lat - 5, zoomnyc$lat + 5) +  
 coord\_map() +   
 geom\_point(data = zip\_comp, aes(zip\_comp$longitude, zip\_comp$latitude, color = zip\_comp$Median)) +  
 labs(x = 'Long', y = 'Lat', fill = 'Avg Median Income') +  
 ggtitle('Avg Median Income by Zipcode', subtitle = 'Northeastern US; center NYC')  
  
# Zipcode density for Northeastern US.   
ggplot(zip\_comp, aes(map\_id = stateName)) +  
 geom\_map(map = us, fill = 'black', colour = 'white') +  
 xlim(zoomnyc$lon - 5, zoomnyc$lon + 5) +  
 ylim(zoomnyc$lat - 5, zoomnyc$lat + 5) +  
 coord\_map() +   
 stat\_density2d(data = zip\_comp, aes(zip\_comp$longitude, zip\_comp$latitude)) +  
 labs(x = 'Long', y = 'Lat', fill = 'Avg Median Income') +  
 ggtitle('Zipcode density', subtitle = 'Northeastern US; center NYC')

### Part 2: Running the Code

# We're loading all the packages that will be needed for this exercise.  
require(ggplot2)

## Loading required package: ggplot2

require(ggmap)

## Loading required package: ggmap

require(readxl)

## Loading required package: readxl

require(gdata)

## Loading required package: gdata

## gdata: Unable to locate valid perl interpreter  
## gdata:   
## gdata: read.xls() will be unable to read Excel XLS and XLSX files  
## gdata: unless the 'perl=' argument is used to specify the location  
## gdata: of a valid perl intrpreter.  
## gdata:   
## gdata: (To avoid display of this message in the future, please  
## gdata: ensure perl is installed and available on the executable  
## gdata: search path.)

## gdata: Unable to load perl libaries needed by read.xls()  
## gdata: to support 'XLX' (Excel 97-2004) files.

##

## gdata: Unable to load perl libaries needed by read.xls()  
## gdata: to support 'XLSX' (Excel 2007+) files.

##

## gdata: Run the function 'installXLSXsupport()'  
## gdata: to automatically download and install the perl  
## gdata: libaries needed to support Excel XLS and XLSX formats.

##   
## 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

require(dplyr)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:gdata':  
##   
## combine, first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

require(zipcode)

## Loading required package: zipcode

# Step 1: Load the Data.   
# Load the MedianZIP\_2\_2\_2.xlsx file.  
dat <- read\_excel('MedianZIP\_2\_2\_2.xlsx')  
  
# Let's take a quick survey of the data.   
str(dat)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 32635 obs. of 4 variables:  
## $ Data from: http://www.psc.isr.umich.edu/dis/census/Features/tract2zip/: chr "Zip" "1001" "1002" "1003" ...  
## $ X\_\_1 : chr "Median" "56662.573499999999" "49853.417699999998" "28462" ...  
## $ X\_\_2 : chr "Mean" "66687.750899999999" "75062.634300000005" "35121" ...  
## $ X\_\_3 : chr "Pop" "16445" "28069" "8491" ...

head(dat)

## # A tibble: 6 x 4  
## `Data from: http://www.psc.isr.umich.edu/~ X\_\_1 X\_\_2 X\_\_3   
## <chr> <chr> <chr> <chr>  
## 1 Zip Median Mean Pop   
## 2 1001 56662.5734~ 66687.7508~ 16445  
## 3 1002 49853.4176~ 75062.6343~ 28069  
## 4 1003 28462 35121 8491   
## 5 1005 75423 82442 4798   
## 6 1007 79076.3540~ 85801.9750~ 12962

summary(dat)

## Data from: http://www.psc.isr.umich.edu/dis/census/Features/tract2zip/  
## Length:32635   
## Class :character   
## Mode :character   
## X\_\_1 X\_\_2 X\_\_3   
## Length:32635 Length:32635 Length:32635   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character

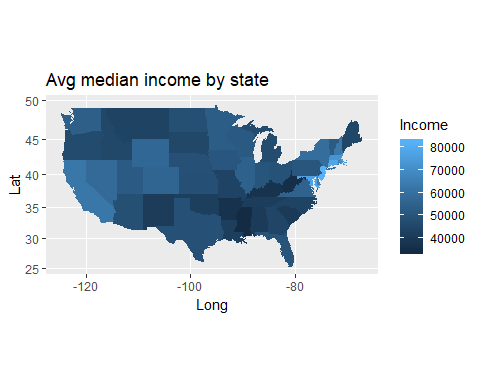
# Looks like the column names will need to be updated and the second to fourth column will need to be converted to numerical type.  
names <- dat[1, ]  
colnames(dat) <- names  
dat <- dat[-1, ]  
  
numdat <- colnames(dat[, 2:4])  
for(i in numdat){  
 dat[[i]] <- as.numeric(dat[[i]])  
 dat[[i]][is.na(dat[[i]])] <- median(dat[[i]], na.rm = T)  
}

## Warning: NAs introducidos por coerción

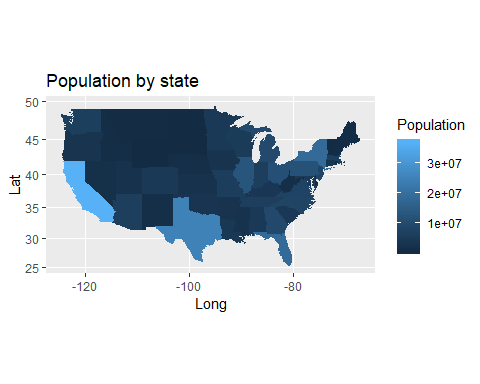
# Let's check once again that the data is correct.  
str(dat)

## Classes 'tbl\_df', 'tbl' and '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 ...  
## $ Pop : num 16445 28069 8491 4798 12962 ...

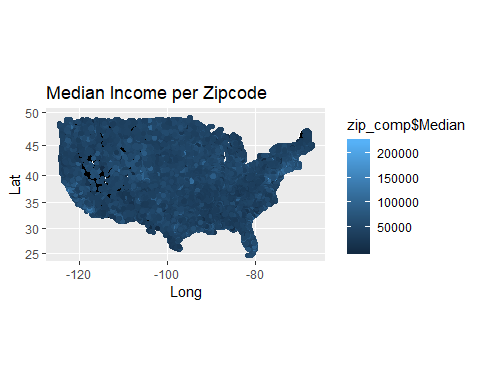
# Now let's upload the zipcode data, merge the data, and remove Alaska and Hawaii.  
data(zipcode)  
dat$zip <- clean.zipcodes(dat$Zip)  
  
zip\_comp <- merge(dat, zipcode, by = 'zip')  
zip\_comp <- zip\_comp %>%  
 filter(!state %in% c('AK', 'HI'))  
  
  
#Step 2: Show income & population per state.  
med\_dat <- zip\_comp %>%  
 group\_by(state) %>%   
 summarize(mean\_med\_income = mean(Median),   
 tot\_pop = sum(Pop))   
  
# Add the state names and convert to lower case.   
med\_dat$stateName <- state.name[match(med\_dat$state, state.abb)]  
med\_dat$stateName <- tolower(med\_dat$stateName)  
  
us <- map\_data("state")  
  
# Time to map average median income by state. We'll color code the map according to the median income.   
ggplot(med\_dat, aes(map\_id = stateName)) +   
 geom\_map(map = us, aes(fill = med\_dat$mean\_med\_income)) +   
 expand\_limits(x = us$long, y = us$lat) +  
 coord\_map() +  
 labs(x = 'Long', y = 'Lat', fill = 'Income') +   
 ggtitle('Avg median income by state')



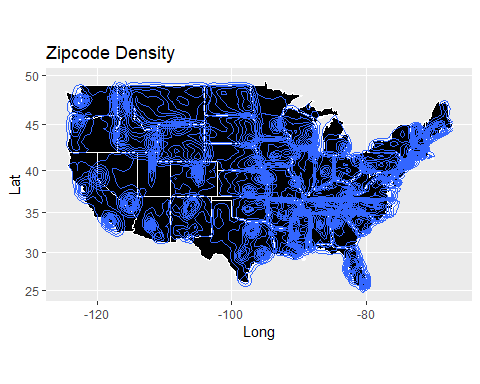
# And now we'll do the same by population.   
ggplot(med\_dat, aes(map\_id = stateName)) +   
 geom\_map(map = us, aes(fill = med\_dat$tot\_pop)) +   
 expand\_limits(x = us$long, y = us$lat) +  
 coord\_map() +  
 labs(x = 'Long', y = 'Lat', fill = 'Population') +   
 ggtitle('Population by state')



# Step 3: Show income per zipcode.  
zip\_comp$stateName <- state.name[match(zip\_comp$state, state.abb)]  
zip\_comp$stateName <- tolower(zip\_comp$stateName)  
  
ggplot(zip\_comp, aes(map\_id = stateName)) +  
 geom\_map(map = us, fill = 'black', colour = 'white') +   
 expand\_limits(x = us$long, y = us$lat) +  
 geom\_point(data = zip\_comp, aes(zip\_comp$longitude, zip\_comp$latitude, color = zip\_comp$Median)) +  
 coord\_map() +  
 labs(x = 'Long', y = 'Lat', fill = 'Income') +   
 ggtitle('Median Income per Zipcode')



# Step 4: Show zipcode density.  
ggplot(zip\_comp, aes(map\_id = stateName)) +  
 geom\_map(map = us, fill = 'black', colour = 'white') +   
 expand\_limits(x = us$long, y = us$lat) +  
 stat\_density2d(data = zip\_comp, aes(zip\_comp$longitude, zip\_comp$latitude)) +  
 coord\_map() +  
 labs(x = 'Long', y = 'Lat') +   
 ggtitle('Zipcode Density')

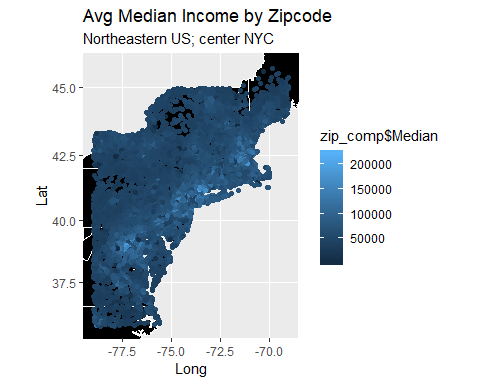


# Step 5: Zoom in to the region around NYC.   
zoomnyc <- geocode('New York City, NY')

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=New%20York%20City,%20NY&sensor=false

# Average Median Income by Zipcode for Northeastern US.   
ggplot(zip\_comp, aes(map\_id = stateName)) +  
 geom\_map(map = us, fill = 'black', colour = 'white') +  
 xlim(zoomnyc$lon - 5, zoomnyc$lon + 5) +  
 ylim(zoomnyc$lat - 5, zoomnyc$lat + 5) +  
 coord\_map() +   
 geom\_point(data = zip\_comp, aes(zip\_comp$longitude, zip\_comp$latitude, color = zip\_comp$Median)) +  
 labs(x = 'Long', y = 'Lat', fill = 'Avg Median Income') +  
 ggtitle('Avg Median Income by Zipcode', subtitle = 'Northeastern US; center NYC')

## Warning: Removed 25809 rows containing missing values (geom\_point).



# Zipcode density for Northeastern US.   
ggplot(zip\_comp, aes(map\_id = stateName)) +  
 geom\_map(map = us, fill = 'black', colour = 'white') +  
 xlim(zoomnyc$lon - 5, zoomnyc$lon + 5) +  
 ylim(zoomnyc$lat - 5, zoomnyc$lat + 5) +  
 coord\_map() +   
 stat\_density2d(data = zip\_comp, aes(zip\_comp$longitude, zip\_comp$latitude)) +  
 labs(x = 'Long', y = 'Lat', fill = 'Avg Median Income') +  
 ggtitle('Zipcode density', subtitle = 'Northeastern US; center NYC')

## Warning: Removed 25809 rows containing non-finite values (stat\_density2d).

