Week 4: Lab - Map Data and Visualization

Koener

2/7/21

# Instructions

**Warning: Do not copy and paste any code I give out here. You MUST write them up.** **If you copy and paste code from here to your RStudio, you will receive errors.**

In Chapter 13 we explored how to use “mashup” data to create a map and put additional data “on top of” the map. In this assignment, we will do that, but with one additional requirement to “zoom” into the region of the United States where the data is of interest.

First, read in the dataset crimeInSYR.csv (in the project’s data folder). The column show the type of crime, the address of the crime, the city of the crime, and the date of the crime. As you can guess from the title of the data file, the focus of the dataset is crimes around the Syracuse, NY area.

For this assignment, you need to create two different, but related, maps: A map with “points” for each of the crimes, and a “density map” showing the same information.

Install the “ggplot2” and “ggmap” packages as needed, and load the two packages in the setup code chunk below.

# Add your library below.  
  
library(ggplot2) # For graphing  
library(ggmap) # For mapping  
library(tidyverse) # For data munging  
library(dplyr)

# Step 1 - A map with “points” for each of the crimes

## Step 1.1 - Read the data

1. Read in crimeInSYR.csv and save it as an object.
2. Investigate the data structure.

# Write your code below.  
  
mydata <- read.csv("data/crimeInSYR.csv")  
workingDF <- mydata

## Step 1.2 - Clean the data

1. Change the column names to (“type”, “address”, “city”, “date”)
2. Combine the address column and city column together and store the values into a new column called address\_complete

## Step 1.3 - Get your Google API Key

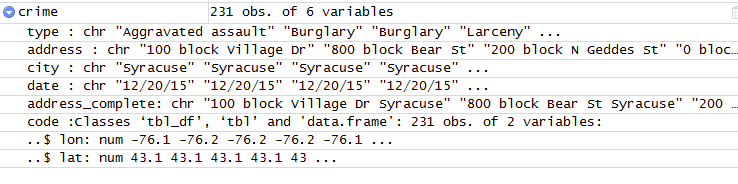
1. You need a Google API key for the next step, which uses the geocode() function. Google API requires you to register your credit card information.
   * To get the API key, follow the directions: <https://developers.google.com/maps/documentation/geocoding/get-api-key>
   * If you already have Google API key, go to <https://console.cloud.google.com/google/maps-apis/api-list?project=uber-kmeans-1549390594743>
     + Then click “Geocoding API” and enable it.
     + Go back a page, click “CREDENTIAL” in the left navigation bar
     + Retrieve your google API key.
     + Enter your API key in the register\_google() function below.

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=tampa,+fl&key=xxx

## Step 1.4 - Get Latitudes and Longitudes

1. Get latitudes and longitudes for all address in the crime dataset and store them in a new column using the geocode() function.
2. Make sure the new dataframe includes lon and lat columns.
3. Investigate the structure.

It should look like this at this point:



Step 1.4 Environment

# Write your code below.  
  
# Request longitudes and latitudes of addresses from google maps.  
lonlats <- geocode(workingDF$address\_complete)

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=100+block+Village+Dr+Syracuse&key=xxx

# New columns with longitudes and latitudes.  
workingDF$lon <- lonlats$lon  
workingDF$lat <- lonlats$lat

## Step 1.5 - Create mapSimple object

Use the following code to create an object called mapSimple:

#get latitude and longtitude for syracuse university  
syr <-geocode ("syracuse university, syracuse, ny")  
syr  
  
# obtain maps from multiple sources and zoom into the region around syracuse university  
syr.map <-get\_map(location=syr, zoom=11)  
  
# generate map and sore it in "mapSimple"  
mapSimple <- ggmap(syr.map)  
# plot the map  
mapSimple

# Write your code below.  
  
# Get latitude and longitude for syracuse university.  
syr <-geocode ("syracuse university, syracuse, ny")

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=syracuse+university,+syracuse,+ny&key=xxx

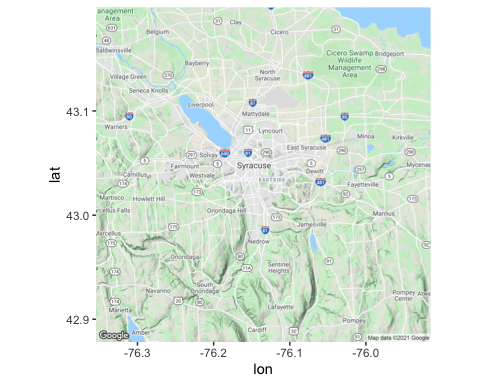
syr

## # A tibble: 1 x 2  
## lon lat  
## <dbl> <dbl>  
## 1 -76.1 43.0

# Obtain maps from multiple sources and zoom into the region around syracuse university.  
syr.map <-get\_map(syr, zoom=11)

## Source : https://maps.googleapis.com/maps/api/staticmap?center=43.039153,-76.135116&zoom=11&size=640x640&scale=2&maptype=terrain&language=en-EN&key=xxx

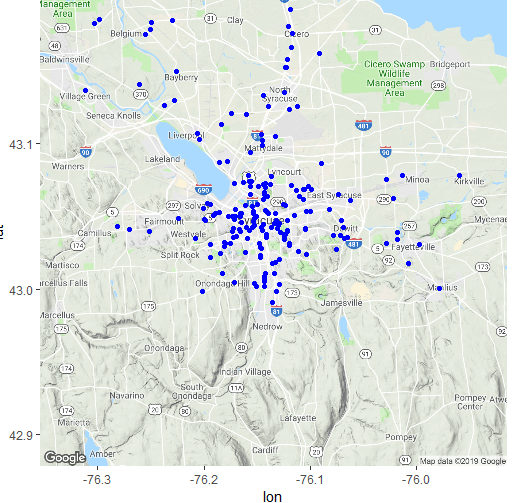
# Generate map and store it in "mapSimple".  
mapSimple <- ggmap(syr.map)  
  
# Plot the map.  
mapSimple



## Step 1.6 - Create crimemape

Create an object called crimemap based on mapSimple, where each point represents one crime.

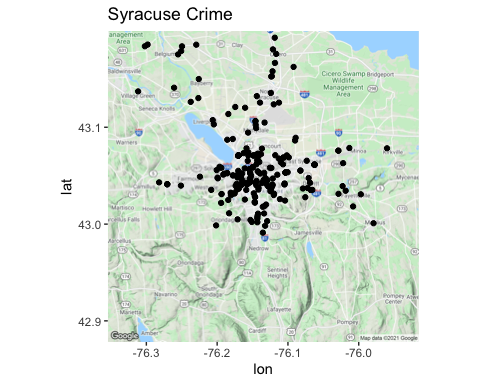
It should look like this:



Step 1.6 Map

# Write your code below.  
  
crimemap <- mapSimple  
crimemap <- crimemap + geom\_point(data=workingDF, aes(x=lon, y=lat))  
crimemap <- crimemap + ggtitle("Syracuse Crime")  
crimemap

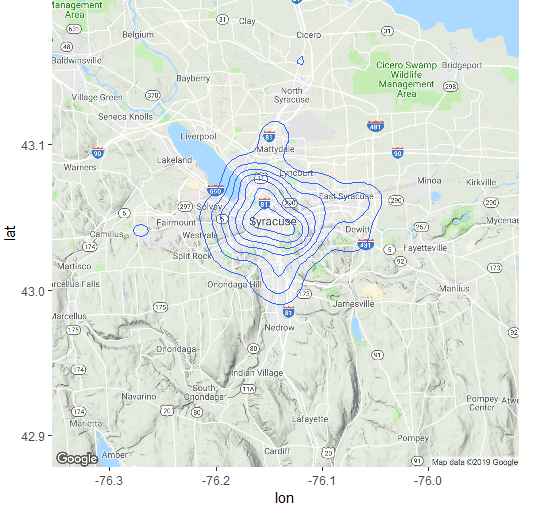
## Warning: Removed 2 rows containing missing values (geom\_point).



# Step 2 - A “density map” showing the same information.

Use the geom\_density2d() function. See the documentation as well as the ggplot cheatsheet to acquire more information on this.

The plot should look like this:



Step 2 Map

# Write your code below.  
  
densityMap <- mapSimple  
densityMap <- densityMap + geom\_density2d(mapping=aes(x=lon, y=lat), data=workingDF)  
densityMap <- densityMap + ggtitle("Syracuse Crime")  
densityMap

## Warning: Removed 2 rows containing non-finite values (stat\_density2d).

