Data Visualization through the use of the Ggplot and Ggmap Packages to Assist in Analyzing Spacial Locations

Samantha Nguyen
10/26/2017

Introduction

R is a programming language used for statistical computing and data analysis. To assist in creating a more thorough analysis, the use of diagrams such as graphs and charts can be very helpful with data visualization. Harding University provides an easy to follow tutorial of simple graphs that can be made using R's programming language directly. The tutorial can be found at https://www.harding.edu/fmccown/r/. However, sometimes more complex visualizations can be much more beneficial and clearer to an analysis. The simplicity of R's graphs do not suffice, specifically when dealing with geographic content. Thus, the **ggmap** package is a tool which allows users to visualize data through combining spcial information of static maps taken from sources such as Google Maps and the OpenStreetMap project.

This post's main purpose is to guide readers toward a deeper understanding of the methods in **ggplot2** and **ggmap** to teach them how to use the packages to create comprehensive visuals. It will include a step-by-step tutorial that goes through how to install the package into RStudio and then how to actually utilize the package with your data. Examples will be used to illustrate a visual approach on how to apply the package on actual data sets.

A concise description on how to use the **ggmap** package is summed up in a Ggmap cheatsheet found here: https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/ggmap/ggmapCheatsheet.pdf.

This post will follow the order of:

1. A tutorial on how to create a map from the **maps** package.

2. Examples made by fellow R programmers who have used the **ggplot2** and **ggmap** packages to analyze data.

3. A summary of what can be learned from this post and an analysis of how this information can be used.

4. A reference list of what sources were used in this post.

Follow the tutorial below for a more comprehensive step-by-step tutorial with a separate example for increased ease in learning how to use the Ggmap package.

Setting up a Map in R Tutorial

Step 1: How to download the packages necessary to use ggmap

Downloading **ggmap** consists of first installing the packages **ggplot2** and **ggmap** into your RStudio. Ggmap builds off of the methods contained in the **ggplot2** package. Installing can be done with the following command:

```
# these are the basic packages that you will use to create your map/data visualizations
install.packages(c("ggplot2", "devtools", "dplyr", "stringr"))

# these are map packages that include locations you can use in your analysis
install.packages(c("maps", "mapdata"))

# this is just an update on ggmap that may be needed in case your ggmap does not load correctly
# even if your ggmap did load successfully, there is no harm in running this line
devtools::install_github("dkahle/ggmap")
```

You then need to load the actual packages into your R Markdown file, using the following commands in your Rmd code chunk:

```
library(ggplot2)
library(ggmap)
```

```
## Google Maps API Terms of Service: http://developers.google.com/maps/terms.

## Please cite ggmap if you use it: see citation("ggmap") for details.
```

```
library(maps)
library(mapdata)
```

Step 2: How to load your data

For this example, we will start with the bigger picture of the entirety of the USA. It just gives us a broad idea of the area we are working with, but soon we will zoom in closer to a specific state so we can do a more in-depth analysis.

We will use the data set given by http://eriqande.github.io/rep-res-web/lectures/making-maps-with-R.html, which gives a thorough example of different functionalities of the **ggmap** package in regards to the USA, such as unique visualizations you can create. For the purpose of this post, we will focus on the analysis of one specific area in the US.

```
# sets the map of the USA to the variable "usa"
usa <- map_data("usa")
```

Step 3: How to display your data

```
# display a map of the USA
ggplot() + geom_polygon(data = usa, aes(x=long, y = lat, group = group)) +
  coord_quickmap() + ggtitle("Map of the USA") + labs(x = "longitude", y = "latitude")
```
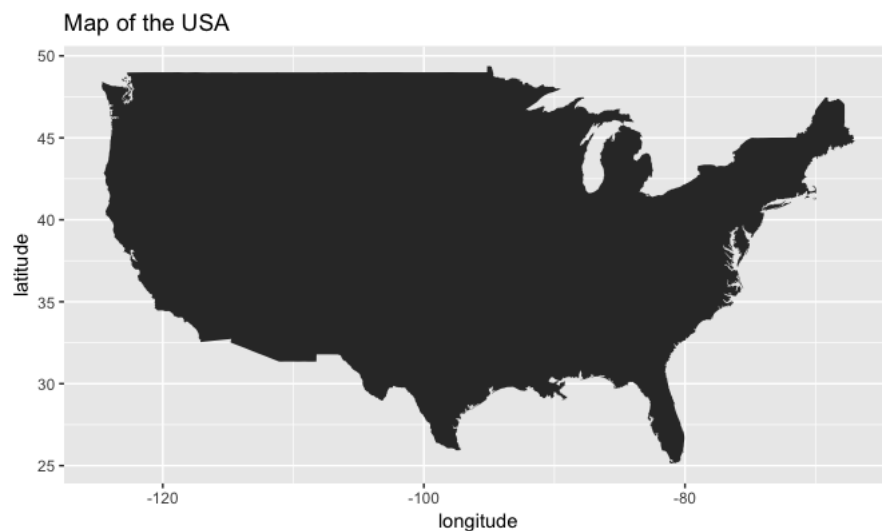

Coord_quickmap() explanation

The code we used to plot the map is mainly straightforward code that can be found on the ggplot cheat sheet: https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf. However, one function that you might not have seen before is **coord_quickmap()**. Coord_quickmap() is a function that will keep the aspect ratio of the x and y units so that if the size of the window/file is ever changed, the map will still come out looking right with the correct ratio.

Maps can also be displayed using the area's longitude and latitude coordinates. We will demonstrate this using the coordinates of Dwinelle Hall in the UC Berkeley campus.

```
# Dwinelle Hall
# input the longitude and latitude of the location
dwinelle <- c(lon = -122.261917, lat = 37.871611)

# retrieve the map from this specific location
dwinelleMap <- get_map(location = dwinelle, crop = FALSE)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=37.871611,-122.261917&zoom=10&size=640x64
```

```
# display the map using ggplot2 functions
ggmap(dwinelleMap) + ggtitle("Map of the Berkeley Area") + labs(x = "longitude", y = "latitude")
```
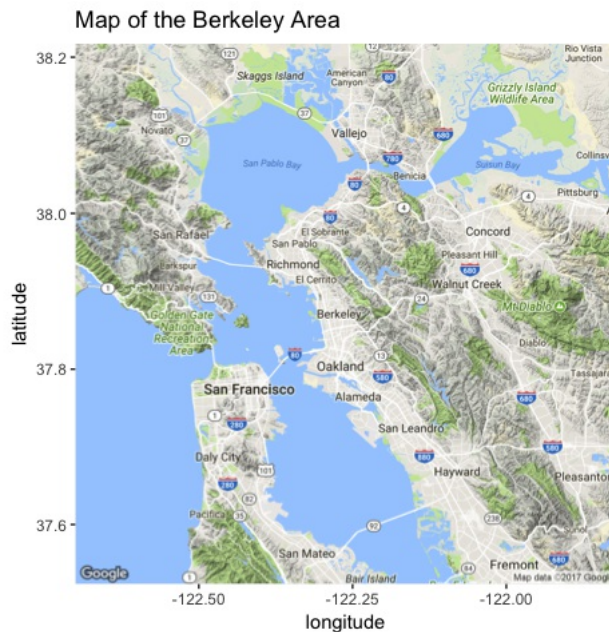

One more way a map can be displayed is through the actual address of the location you wish to focus on. Using the geocode() function that is part of **ggmap**, we can input the address. In this example, we will locate the Taco Bell on Durant Avenue in Berkeley, CA.

```
# use the geocode function to input your address
tacoBellGeocode <- geocode("2528 Durant Ave, Berkeley, CA 94704", output = "more")
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=2528%20Durant%20Ave%2C%20Berkeley%2C%2C9
```

```
tacoBellGeocode
```

```
##          lon      lat         type         loctype
## 1 -122.2576 37.86774 street_address range_interpolated
##                                  address    north    south      east
## 1 2528 durant ave, berkeley, ca 94704, usa 37.86909 37.86639 -122.2563
##       west street_number        route neighborhood locality
## 1 -122.259        2528 Durant Avenue    Southside Berkeley
##   administrative_area_level_2 administrative_area_level_1      country
## 1            Alameda County              California United States
##   postal_code
## 1      94704
```

```
# use the ggmap function gmap() to create your map
qmap(location = c(lon = tacoBellGeocode$lon, lat = tacoBellGeocode$lat), zoom = 17, maptype = "satellite")
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=37.86774,-122.257607&zoom=17&size=640x640
```

provides another example of using geocode() to output a map from a specific address. This website offers step by step images of the process as well, so is a great tool to use as well to follow.

Step 4: How to select a specfic area

Here we will look at California specifically.

```r
# stores all of the states into a variable called "states"
states <- map_data("state")

# retrieve California out of the states variable and save it as its own variable
california <- subset(states, region == "california")

# store the counties
counties <- map_data("county")

# save the counties of California into the "ca_counties" variable
ca_counties <- subset(counties, region == "california")

# create the map of California using ggplot2 functions and then save it into the variable "ca"
ca <- ggplot(data = california, mapping = aes(x = long, y = lat, group = group)) +
  coord_quickmap() + ggtitle("CA") + geom_polygon(color = "black", fill = "blue") +
  geom_polygon(data = ca_counties, fill = NA, color = "white") +
  geom_polygon(color = "black", fill = NA)
ca
```

CA
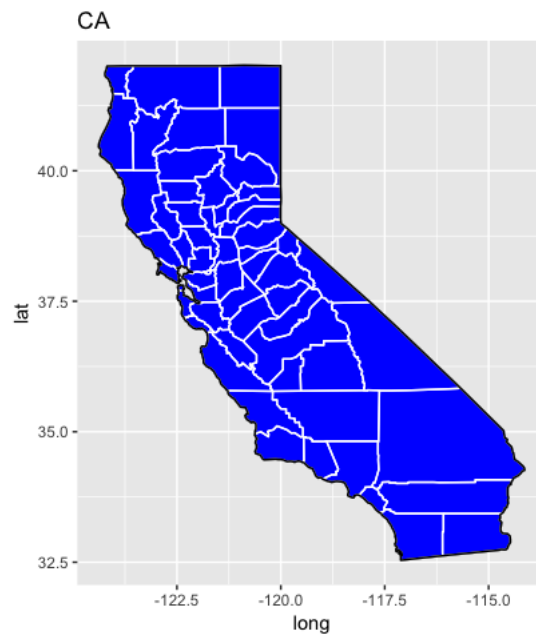
Again, these functions of ggplot2 can be found on the ggplot2 cheat sheet: https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf. We had to use geom_polygon() for the final time so that the outline of the state itself could be shown, as it was covered by the geom_polygon() function for the counties.

This was just one example of creating a map with area locations. With more data, we can do a visual analysis on information such as population number or terrain characteristics.

Ggmap Examples

Example 1:

Eric C. Anderson, the author of the page http://eriqande.github.io/rep-res-web/lectures/making-maps-with-R.html that we based this tutorial off of, showcased a beautiful map of California based on its population here:

**California based off of County Populations**
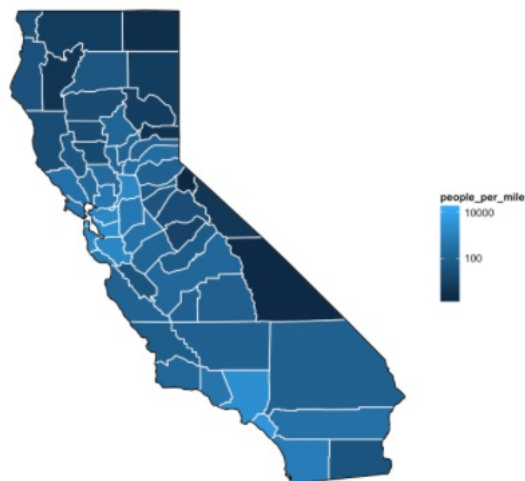
This map analysis can be created through the retrieval of a data set that contains the population information, and combining it into a data frame that has the county subregion names. The numbers can then be organized accordingly into colors that are set to each individual county.

Example 2:

https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf used **ggmap** and **ggplot** to create a map visualization of the density of crimes that took place in the Houston, Texas area.

Countoured plot of the violent crimes that took place in Houston, Texas
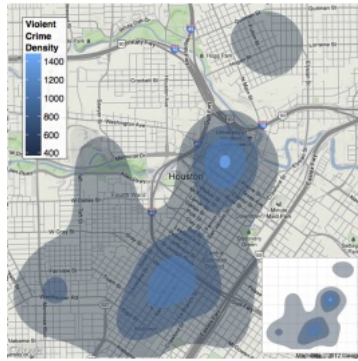
Example 3:

Crime in an area can be split into individual days of the week of the crime occurences, so an analysis of which days had the most crime occurences can be made. This is seen with the work of http://stat405.had.co.nz/ggmap.pdf which took the data crimes in Houston and created this faceted diagram:

Faceted contour plot of crimes in Houston organized by day
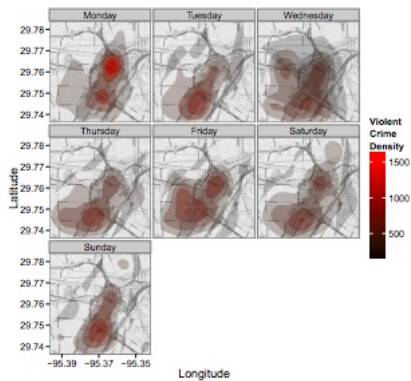
Example 4:

Companies can also hire data analysts to find trends in where their consumers are using their products. http://www.kevjohnson.org/making-maps-in-r-part-2/ showcases a broad approach to this when he finds the number of people who don't have health insurance in Georgia and demonstrates it with a map created through **ggmap**. This is helpful in seeing the trends between counties and what locations should be targeted more/why they may have a lower number of consumers than other areas.

The map created by Kevin Johnson is shown here:

Percentage of Population Without
Health Insurance
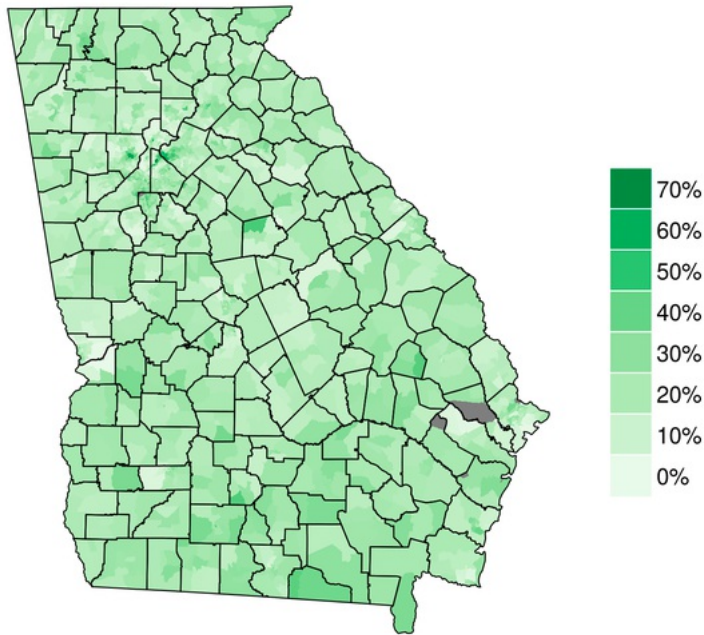
Summary

The overarching message that should be taken from this post is that a much more thorough analysis can be made at times through the use of packages such as **ggplot2** and **ggmap**. When looking at spacial locations, it can be easier to recognize trends through a visual identification of common colors/designs present in the diagram. Thus, this is most easily shown with the use of the packages introduced with this post.

We saw examples of the **ggmap** package in action, such as using it to distinguish specific locations that had the highest crime rates in an area. **Ggmap** can be used for a whole range of purposes, such as demonstrating the crime rate, the population, company locations, etc. Utilizing the packages from this post will allow analytic programmers to discover more from their data. Companies can use the data to see areas they need to target more with their product, consumers can use it to find the nearest restaurant of their preference, etc.

Reference List

https://www.harding.edu/fmccown/r/

https://journal.r-project.org/archive/2013-1/kahle-wickham.pdf

https://www.nceas.ucsb.edu/~frazier/RSpatialGuides/ggmap/ggmapCheatsheet.pdf

http://stat405.had.co.nz/ggmap.pdf

http://eriqande.github.io/rep-res-web/lectures/making-maps-with-R.html

http://zevross.com/blog/2014/03/19/geocoding-with-rs-ggmap-package/

http://www.kevjohnson.org/making-maps-in-r-part-2/