Data Visualization using ggplot2 and ggmaps

Min Ju Chung

10/29/2017

> ## Content:
>
> - Introduction
> - Scatterplots
> - Maps
> - Population Density Plot
> - State Map
> - Contour Heat Map
> - Summary
> - List of References

```
# packages used in this post
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
library(maps)
```

```
## Warning: package 'maps' was built under R version 3.3.2
```

```
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.
```

Introduction

As we had learned in lecture and lab, the package `"ggplot2"` is a popular package used to create complex multi-layered graphics. Created by Hadley Wichkham in 2005, this data visualization package is a plotting system for R. Although there may be cases when using R's base plotting tools is superior, `"ggplot2"` helps users easily make aesthetically pleasing and elaborate graphics!

The **motivation** of this post is to highlight the additional features and components (outside of what was taught in lecture and completed in lab) of both `"ggplot2"` and `"ggmap"`. One of the key features of `"ggplot2"` is the ability to superpose multiple layers, such as lines, maps, and points, from various data sources. This will be shown in the first example of the post. Furthermore, `"ggplot2"` follows grammar of graphics, which is a tool that helps users concisely describe the needed elements of a graphic. The importance of the implementation of the grammar of graphics can be found in this article. Similarly, the package `"ggmap"` is useful for creating visualizations of spatial data and models on maps from online sources such as Google Maps and Stamen Maps. One of its advantages is being able to apply text search inputs with the `location` parameter rather than specifying the latitude and longitude.

Scatterplot

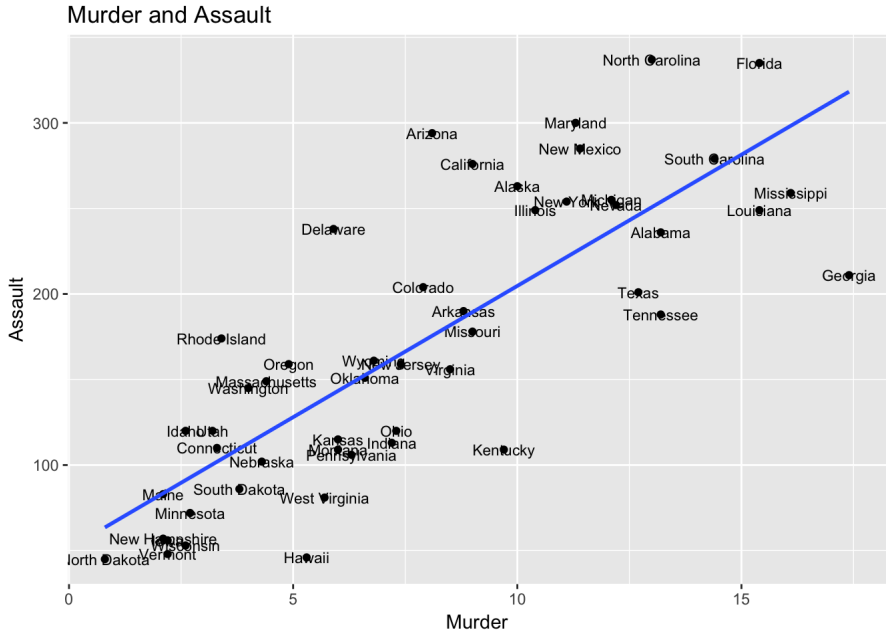
For reviewing purposes, I have included this scatterplot, which uses the main function `ggplot()` and other geometries, or layers. The following example utilizes the content of the R built-in data set, USArrests. This data set contains statistics of arrests for assault, murder, and rape in each US state in 1973.

Here's an example of a simple scatterplot of `Murder` and `Assault` created by using the `"ggplot2"` package.

```
# loading the data set
data("USArrests")

# Converting row names into the first column
df <- tibble::rownames_to_column(USArrests, "States")

ggplot(df, aes(x = Murder, y = Assault)) +
  geom_point() +
  geom_text(aes(label = df$States), size = 3) +
  ggtitle("Murder and Assault") +
  geom_smooth(method = lm, se = FALSE)
```



From this graph, we can further understand the somewhat positive correlation between the two variables, assault and murder, as shown in the linear regression line. The `geom_text()` function allows us to see, specifically, the labels of the states for each point on the graphics.

Maps

The ability to integrate `"ggmap"` and geometric objects `"ggplot2"` is a large advantage for geographic visualization, such as through heat maps, contour maps, and other possible spatial plot types. The package `"ggmap"` creates spatial data visualization and is built on top of `"ggplot2"`. This allows users to retrieve map tiles from online mapping platforms, such as Google Maps, and plot them by using the `"ggplot2"` framework.

The following describes more about the usage of `"ggmap"` and arguments of `get_map()` that are used in the example below.

- The main function in `"ggmap"` is `ggmap()`.
- The function `get_map()` retrieves a specified map from Google Maps API.
- The argument `center`, or the center of the map, uses the string address from geocode with source = “google”.
- The `scale` is the multiplicative factor for the number of pixels.
- The `zoom` parameter takes in an integer from 3 (continent) to 21 (building).

Population Density Plot

This example was adapted to reflect the population of California cities on a map from this source.

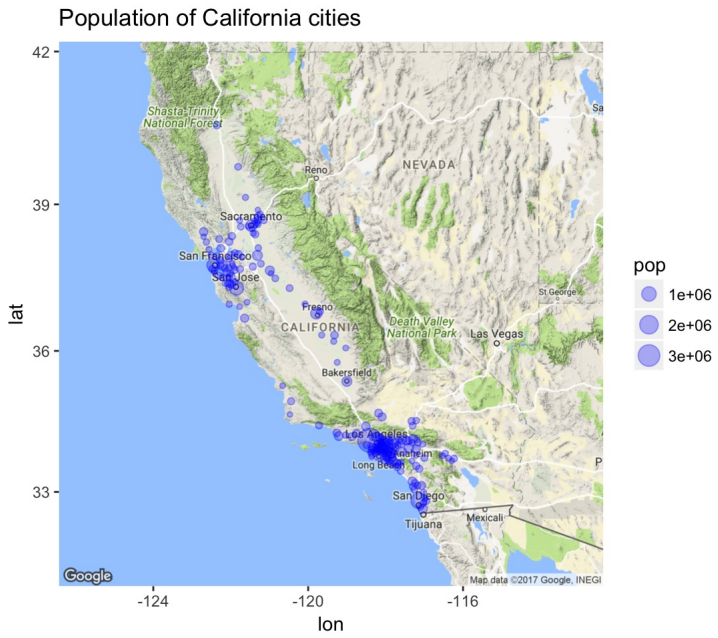
```
# install.packages('devtools')
# devtools::install_github("dkahle/ggmap")

# built-in data set from package, maps
data(us.cities)
CAmap <- ggmap(get_map("California", scale = 2, zoom = 6), extent = "panel")
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=California&zoom=6&size=640x640&scale=2&maptype=terrain&language=en-EN
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=California
```

```
CA.cities <- subset(us.cities, country.etc == "CA")
map_of_CA <- CAmap + geom_point(aes(x = long, y = lat, size = pop), col = "blue", data = CA.cities, alpha = 0.3) +
  ggtitle("Population of California cities")
map_of_CA
```




```
ggsave(filename = '../post01/images/map_of_CA.png',
       plot = map_of_CA,
       width = 5, height = 4)
```

With the use of `"ggmap"`, we can visualize the population density of the cities in California! We can note that the cities, San Francisco, San Jose, Sacramento, Los Angeles, and San Diego (just to name a few), have higher human population density than that of other cities.

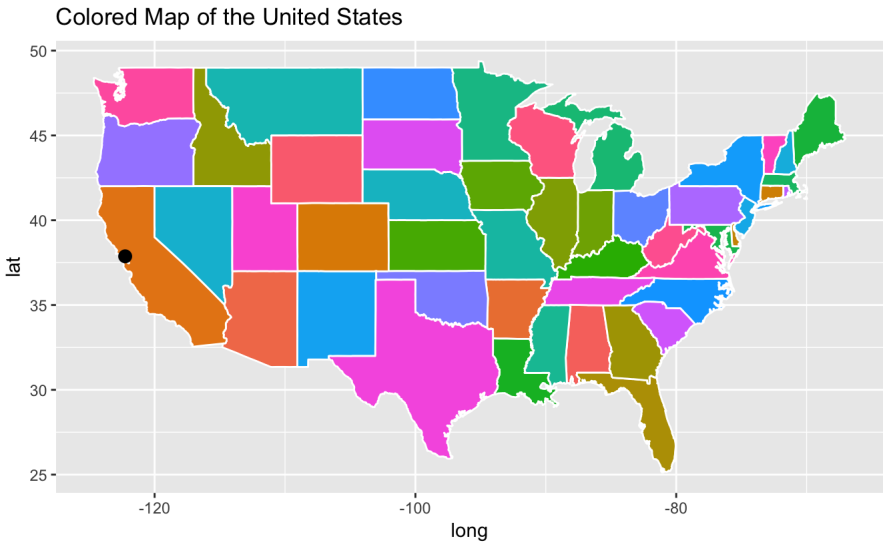
State Map

Maps in the this format can be created by using the `geom_polygon()`.

- `geom_polygon()` creates a line between the start and end points. The `group` aesthetic controls which cases are connected to form a polygon.
- `coord_fixed()` is used to change the relationship between one unit in the x direction and one unit in the y direction. Specifically, in the next example, `coord_fixed(1.3)` indicates that every y unit is 1.3 times longer than every x unit.
- Points can be added to the map by using `geom_point`.
- By setting fill to `region`, the colors of the states are all colored differently.
- `guides` creates the legend.

This **next example** was taken from this source.

```
states <- map_data("state")
UC.Berkeley <- data.frame(long = -122.258423,
                          lat = 37.871853,
                          stringsAsFactors = FALSE)
ggplot(data = states) +
  geom_polygon(aes(x = long, y = lat, fill = region, group = group), color = "white") +
  coord_fixed(1.3) +
  guides(fill = FALSE) +
  geom_point(data = UC.Berkeley, aes(x = long, y = lat), color = "black", size = 3) +
  ggtitle("Colored Map of the United States")
```



This is a colorful map of the United States! The black dot indicates where we are right now (UC Berkeley)!

Contour Heat Map

Another plot type that can be created with `"ggmap"` is contour heat maps! The contour heat map is produced by using a two-dimensional kernel density estimation. This can be interpreted as a histogram in two dimensions! The polygonal density layer is added on top of the map by using the `stat_density2d()` function. The density layers are then "filled," or shaded, respective to their density levels, which is created by using the `..level..` parameter for the argument `alpha`.

The contour heat map example was duplicated from this source. The example maps the crime density of Houston.

```
data(crime)
houston <- get_map(location = "houston", zoom = 13)
```

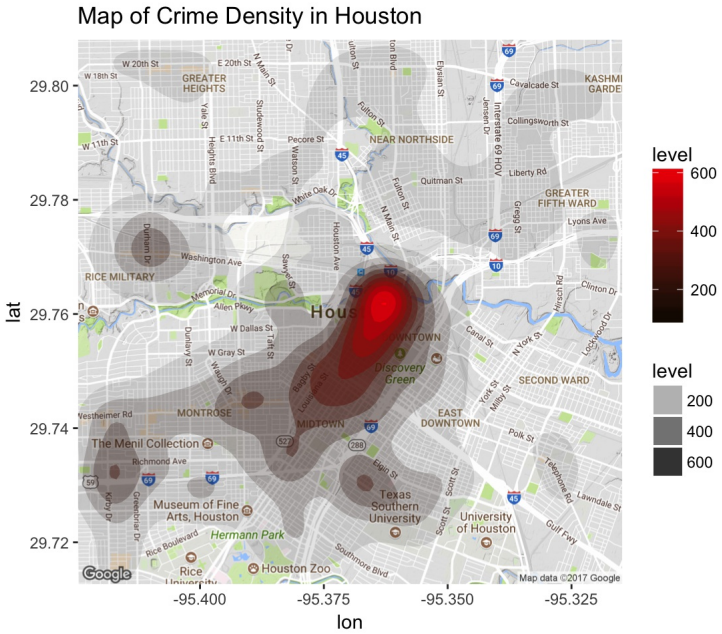
```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=houston&zoom=13&size=640x640&scale=2&maptype=ter
rain&language=en-EN
```

```
## Warning: package 'bindrcpp' was built under R version 3.3.2
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=houston
```

```
houstonMap <- ggmap(houston)
houstonMap +
  stat_density2d(aes(x = lon, y = lat, fill = ..level.., alpha = ..level..), bins = 10, geom = "polygon", data = c
rime) +
  scale_fill_gradient(low = "black", high = "red") +
  ggtitle("Map of Crime Density in Houston")
```

```
## Warning: Removed 72407 rows containing non-finite values (stat_density2d).
```



From the contour map, we can observe that the crime rates are more densely located in one region of Houston compared to other areas. Contour maps are pretty cool and informative!

Summary

The take-home message of this post is to understand the advantages of using the `"ggplot"` approach to data and spatial visualization. The previously mentioned examples of both `"ggplot2"` and `"ggmaps"` illustrate the numerous types of plots and maps, such as the state and contour maps, that can be created by using the functions provided in the packages. Furthermore, the packages `"ggplot2"` and `"ggmaps"` allow users to construct complex plots in layers, such as with the functions, `geom_point()` and `stat_density2d()` and are consistent (follows the grammar of graphics).

Reference

- Background on the package: ggplot2
- Information on Grammar of Graphics
- More about ggmaps
- Example 2
- Example 3
- Geographic visualization with R's ggmap
- More on Contour Heat Maps