

Geographic Visualization with ggmap

Sylvia Feng

Introduction

Have you ever encountered some data sets that involved spatial locations and geographic data, and wondering how to visualize them or transform them into statistically analyzable form? If you do, you might know how tedious and troubling it is to create spatial plots in languages like Matlab or applications like Excel. If you don't, then here is a new way to deal with spatial data without long, tiring process! With a powerful data analyzing tool R, we can use a number of new R libraries to make spatial data visualization a more enjoyable endeavor. Of these new options, one useful package is **ggmap**. ggmap is a new tool which enables geographic visualization by combining the spatial information of static maps like Google Maps, OpenStreetMap, Stamen Maps, and **ggplot2**'s plotting framework.

Getting Started

```
library(ggmap)

## Warning: package 'ggmap' was built under R version 3.4.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.4.2
```

Get a map of the United States first to warm-up:

```
US <- c(left = -125, bottom = 25.75, right = -67, top = 49)
map <- get_stamenmap(US, zoom = 5, maptype = "toner-lite")

## Map from URL : http://tile.stamen.com/toner-lite/5/4/10.png

## Map from URL : http://tile.stamen.com/toner-lite/5/5/10.png

## Map from URL : http://tile.stamen.com/toner-lite/5/6/10.png

## Map from URL : http://tile.stamen.com/toner-lite/5/7/10.png

## Map from URL : http://tile.stamen.com/toner-lite/5/8/10.png

## Map from URL : http://tile.stamen.com/toner-lite/5/9/10.png

## Map from URL : http://tile.stamen.com/toner-lite/5/10/10.png

## Map from URL : http://tile.stamen.com/toner-lite/5/4/11.png

## Map from URL : http://tile.stamen.com/toner-lite/5/5/11.png

## Map from URL : http://tile.stamen.com/toner-lite/5/6/11.png

## Map from URL : http://tile.stamen.com/toner-lite/5/7/11.png

## Map from URL : http://tile.stamen.com/toner-lite/5/8/11.png

## Map from URL : http://tile.stamen.com/toner-lite/5/9/11.png

## Map from URL : http://tile.stamen.com/toner-lite/5/10/11.png

## Map from URL : http://tile.stamen.com/toner-lite/5/4/12.png

## Map from URL : http://tile.stamen.com/toner-lite/5/5/12.png

## Map from URL : http://tile.stamen.com/toner-lite/5/6/12.png

## Map from URL : http://tile.stamen.com/toner-lite/5/7/12.png

## Map from URL : http://tile.stamen.com/toner-lite/5/8/12.png
```

```

## Map from URL : http://tile.stamen.com/toner-lite/5/9/12.png

## Map from URL : http://tile.stamen.com/toner-lite/5/10/12.png

## Map from URL : http://tile.stamen.com/toner-lite/5/4/13.png

## Map from URL : http://tile.stamen.com/toner-lite/5/5/13.png

## Map from URL : http://tile.stamen.com/toner-lite/5/6/13.png

## Map from URL : http://tile.stamen.com/toner-lite/5/7/13.png

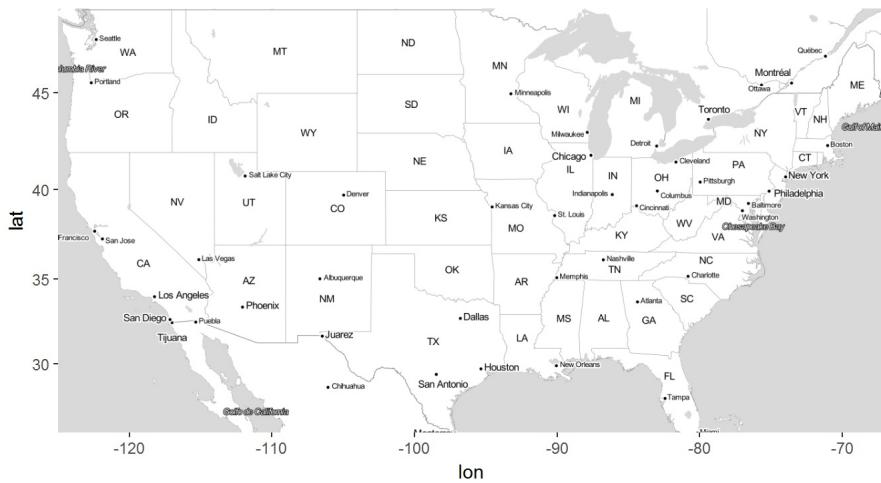
## Map from URL : http://tile.stamen.com/toner-lite/5/8/13.png

## Map from URL : http://tile.stamen.com/toner-lite/5/9/13.png

## Map from URL : http://tile.stamen.com/toner-lite/5/10/13.png

ggmap(map)

```



The argument “zoom” specified how close we want to look at the map, i.e. how large and specific the map is. “maptyle” determines what types of map we want to use.

For example, a different map type might be “watercolor”:

```

an <- get_stamenmap(US, zoom = 5, maptype = "watercolor")

## Map from URL : http://tile.stamen.com/watercolor/5/4/10.jpg

## Map from URL : http://tile.stamen.com/watercolor/5/5/10.jpg

## Map from URL : http://tile.stamen.com/watercolor/5/6/10.jpg

## Map from URL : http://tile.stamen.com/watercolor/5/7/10.jpg

## Map from URL : http://tile.stamen.com/watercolor/5/8/10.jpg

## Map from URL : http://tile.stamen.com/watercolor/5/9/10.jpg

## Map from URL : http://tile.stamen.com/watercolor/5/10/10.jpg

## Map from URL : http://tile.stamen.com/watercolor/5/4/11.jpg

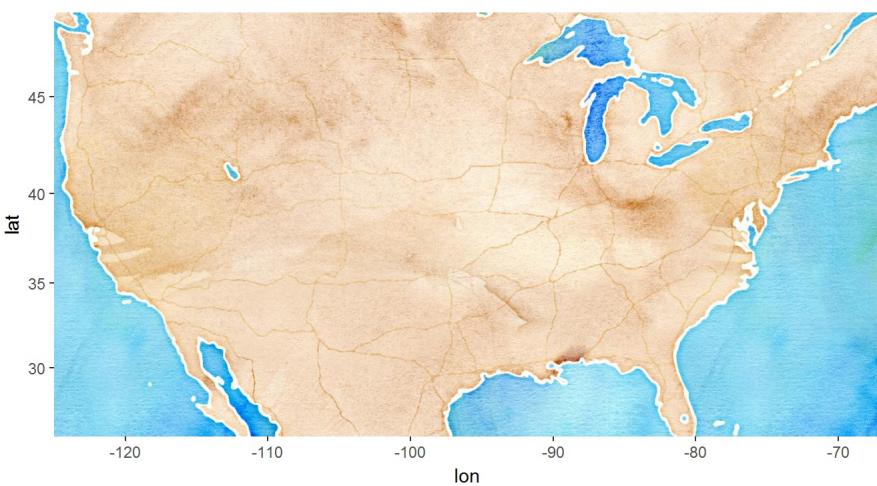
## Map from URL : http://tile.stamen.com/watercolor/5/5/11.jpg

```

```

## Map from URL : http://tile.stamen.com/watercolor/5/6/11.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/7/11.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/8/11.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/9/11.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/10/11.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/4/12.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/5/12.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/6/12.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/7/12.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/8/12.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/9/12.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/10/12.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/4/13.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/5/13.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/6/13.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/7/13.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/8/13.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/9/13.jpg
## Map from URL : http://tile.stamen.com/watercolor/5/10/13.jpg
ggmap(an)

```



An Example: Crime Distirbution

After we know how to extract basic maps from ggmap, let's see an example:

We can extract a data set about crime information in North Carolina. “[crime](#)”, from R’s built-in data sets, and we only want to plot the violent

crimes and disregard the smaller ones.

First, we need to load package **dplyr** for data preparation purpose.

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.2

## 
## 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

# only violent crimes
violent_crimes <- filter(crime,
  offense != "auto theft", offense != "theft", offense != "burglary"
)
```

Next, let's rearrange the order of factors to rank the data based on how serious the crime is:

```
# rank violent crimes
violent_crimes$offense <- factor(
  violent_crimes$offense,
  levels = c("robbery", "aggravated assault", "rape", "murder")
)
```

To narrow down the number of crimes we deal with, we only want to look at the crimes happened in a certain area:

```
# restrict to downtown
violent_crimes <- filter(violent_crimes,
  -95.39681 <= lon & lon <= -95.34188,
  29.73631 <= lat & lat <= 29.78400
)
```

Finally, we use a function in ggmap, qmplot(), to plot the crime distribution:

```
qmplot(lon, lat, data = violent_crimes, maptype = "toner-lite", color = I("red"))

## Using zoom = 14...

## Map from URL : http://tile.stamen.com/toner-lite/14/3850/6770.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3851/6770.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3852/6770.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3853/6770.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3850/6771.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3851/6771.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3852/6771.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3853/6771.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3850/6772.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3851/6772.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3852/6772.png

## Map from URL : http://tile.stamen.com/toner-lite/14/3853/6772.png

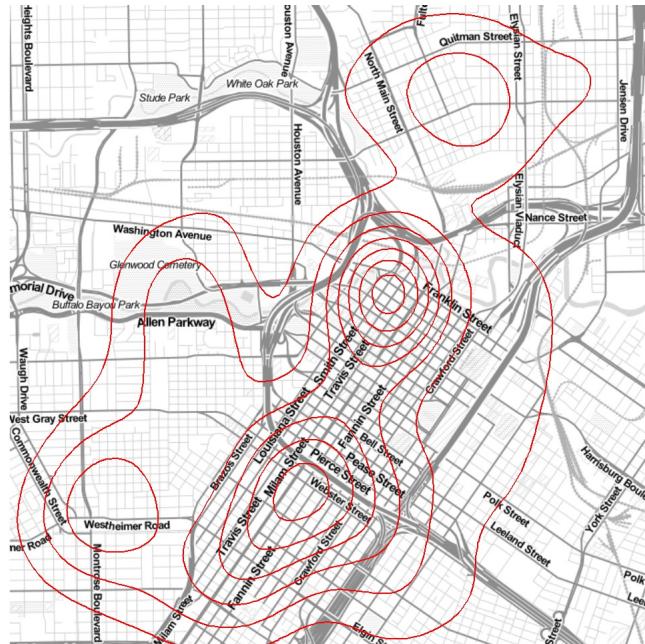
## Map from URL : http://tile.stamen.com/toner-lite/14/3850/6773.png
```

```
## Map from URL : http://tile.stamen.com/toner-lite/14/3851/6773.png
## Map from URL : http://tile.stamen.com/toner-lite/14/3852/6773.png
## Map from URL : http://tile.stamen.com/toner-lite/14/3853/6773.png
## Warning: `panel.margin` is deprecated. Please use `panel.spacing` property
## instead
```



Recall how we used to draw isoheight lines on geography class and see which areas are at the same height. Now we want to do the same thing with this plot; however, we are not interested in the geographic height, but the density of crimes:

```
qmapplot(lon, lat, data = violent_crimes, maptype = "toner-lite", geom = "density2d", color = I("red"))
## Using zoom = 14...
## Warning: `panel.margin` is deprecated. Please use `panel.spacing` property
## instead
```



Great! Now we can see where in downtown do crimes happen more often.

However, if the people are reading this plot don't understand isoheight lines, this plot might be still in a format that is harder to read. Since ggmap's built on top of ggplot2, all the usual ggplot2 stuff (geoms, polishing, etc.) will work, and there are some unique graphing perks ggmap brings to the table to make the plot more readable, too.

To start easily, let's use robberies as an example:

```
robberies <- violent_crimes %>% filter(offense == "robbery")

qmplot(lon, lat, data = violent_crimes, geom = "blank", zoom = 15, maptype = "toner-background", darken = .7, legend = "topleft") +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", alpha = .3, color = NA) +
  scale_fill_gradient2("Robbery\nnPropensity", low = "white", mid = "yellow", high = "red", midpoint = 650)

## 49 tiles needed, this may take a while (try a smaller zoom).

## Map from URL : http://tile.stamen.com/toner-background/15/7700/13541.png

## Map from URL : http://tile.stamen.com/toner-background/15/7701/13541.png

## Map from URL : http://tile.stamen.com/toner-background/15/7702/13541.png

## Map from URL : http://tile.stamen.com/toner-background/15/7703/13541.png

## Map from URL : http://tile.stamen.com/toner-background/15/7704/13541.png

## Map from URL : http://tile.stamen.com/toner-background/15/7705/13541.png

## Map from URL : http://tile.stamen.com/toner-background/15/7706/13541.png

## Map from URL : http://tile.stamen.com/toner-background/15/7700/13542.png

## Map from URL : http://tile.stamen.com/toner-background/15/7701/13542.png

## Map from URL : http://tile.stamen.com/toner-background/15/7702/13542.png

## Map from URL : http://tile.stamen.com/toner-background/15/7703/13542.png

## Map from URL : http://tile.stamen.com/toner-background/15/7704/13542.png

## Map from URL : http://tile.stamen.com/toner-background/15/7705/13542.png

## Map from URL : http://tile.stamen.com/toner-background/15/7706/13542.png

## Map from URL : http://tile.stamen.com/toner-background/15/7700/13543.png

## Map from URL : http://tile.stamen.com/toner-background/15/7701/13543.png

## Map from URL : http://tile.stamen.com/toner-background/15/7702/13543.png

## Map from URL : http://tile.stamen.com/toner-background/15/7703/13543.png

## Map from URL : http://tile.stamen.com/toner-background/15/7704/13543.png

## Map from URL : http://tile.stamen.com/toner-background/15/7705/13543.png

## Map from URL : http://tile.stamen.com/toner-background/15/7706/13543.png

## Map from URL : http://tile.stamen.com/toner-background/15/7700/13544.png

## Map from URL : http://tile.stamen.com/toner-background/15/7701/13544.png

## Map from URL : http://tile.stamen.com/toner-background/15/7702/13544.png

## Map from URL : http://tile.stamen.com/toner-background/15/7703/13544.png

## Map from URL : http://tile.stamen.com/toner-background/15/7704/13544.png

## Map from URL : http://tile.stamen.com/toner-background/15/7705/13544.png

## Map from URL : http://tile.stamen.com/toner-background/15/7706/13544.png
```

```
## Map from URL : http://tile.stamen.com/toner-background/15/7700/13545.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7701/13545.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7702/13545.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7703/13545.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7704/13545.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7705/13545.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7706/13545.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7700/13546.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7701/13546.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7702/13546.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7703/13546.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7704/13546.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7705/13546.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7706/13546.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7700/13547.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7701/13547.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7702/13547.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7703/13547.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7704/13547.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7705/13547.png  
## Map from URL : http://tile.stamen.com/toner-background/15/7706/13547.png  
  
## Warning: `panel.margin` is deprecated. Please use `panel.spacing` property  
## instead
```



We can use the faceting feature too:

```
qmpplot(lon, lat, data = violent_crimes, maptype = "toner-background", color = offense) +
  facet_wrap(~ offense)

## Using zoom = 14...

## Map from URL : http://tile.stamen.com/toner-background/14/3850/6770.png

## Map from URL : http://tile.stamen.com/toner-background/14/3851/6770.png

## Map from URL : http://tile.stamen.com/toner-background/14/3852/6770.png

## Map from URL : http://tile.stamen.com/toner-background/14/3853/6770.png

## Map from URL : http://tile.stamen.com/toner-background/14/3850/6771.png

## Map from URL : http://tile.stamen.com/toner-background/14/3851/6771.png

## Map from URL : http://tile.stamen.com/toner-background/14/3852/6771.png

## Map from URL : http://tile.stamen.com/toner-background/14/3853/6771.png

## Map from URL : http://tile.stamen.com/toner-background/14/3850/6772.png

## Map from URL : http://tile.stamen.com/toner-background/14/3851/6772.png

## Map from URL : http://tile.stamen.com/toner-background/14/3852/6772.png

## Map from URL : http://tile.stamen.com/toner-background/14/3853/6772.png

## Map from URL : http://tile.stamen.com/toner-background/14/3850/6773.png

## Map from URL : http://tile.stamen.com/toner-background/14/3851/6773.png

## Map from URL : http://tile.stamen.com/toner-background/14/3852/6773.png

## Map from URL : http://tile.stamen.com/toner-background/14/3853/6773.png

## Warning: `panel.margin` is deprecated. Please use `panel.spacing` property
## instead
```



Noted that this is only one way of plotting them. We can also have other methods. For instance, ggplot2 has some useful functions:

```
theme_set(theme_bw(16))
HoustonMap <- qmap("houston", zoom = 14, color = "bw", legend = "topleft")
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=houston&zoom=14&size=640x640&scale=2&maptype=terrain&language=en-EN&sensor=false
```

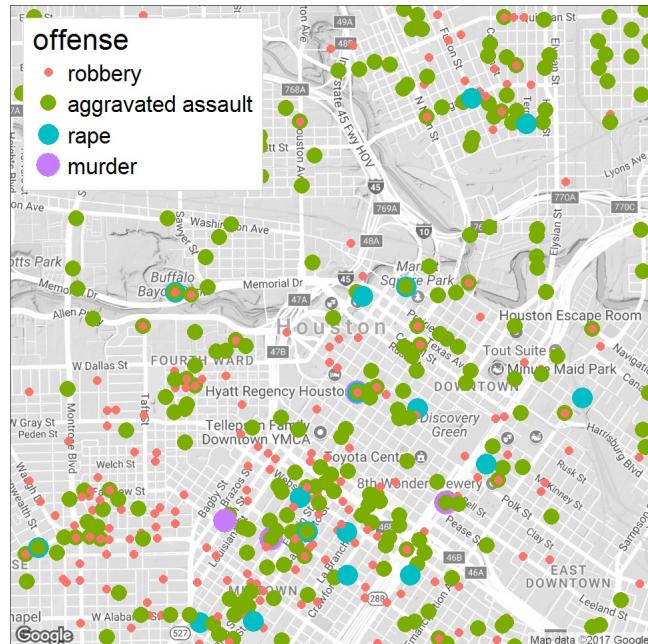
```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=houston&sensor=false
```

```
## Warning: `panel.margin` is deprecated. Please use `panel.spacing` property
## instead
```

```
HoustonMap +
  geom_point(aes(x = lon, y = lat, colour = offense, size = offense),
             data = violent_crimes)
```

```
## Warning: Using size for a discrete variable is not advised.
```

```
## Warning: Removed 11 rows containing missing values (geom_point).
```



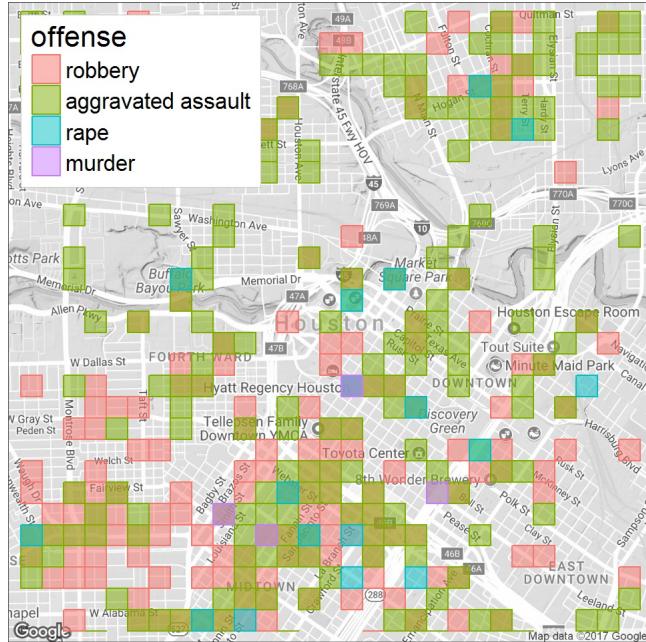
```

HoustonMap +
stat_bin2d(
  aes(x = lon, y = lat, colour = offense, fill = offense),
  size = .5, bins = 30, alpha = 1/2,
  data = violent_crimes
)

## Warning: Removed 11 rows containing non-finite values (stat_bin2d).

## Warning: Removed 8 rows containing missing values (geom_tile).

```



Instead of faceting, the two graphs above put all the data onto one plot. The first one use the size and color of circles to differentiate different crimes. The second one used squares instead of circles.

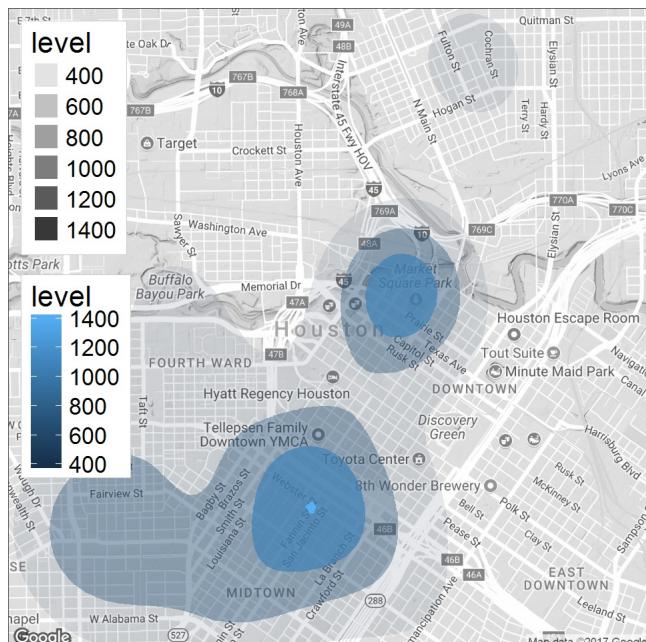
Likewise, we can use ggplot to produce density plots too:

```

HoustonMap +
stat_density2d(
  aes(x = lon, y = lat, fill = ..level.., alpha = ..level..),
  size = 2, bins = 4, data = robberies,
  geom = "polygon"
)

```

```
## Warning: Removed 2 rows containing non-finite values (stat_density2d).
```



```

overlay <- stat_density2d(
  aes(x = lon, y = lat, fill = ..level.., alpha = ..level..),
  bins = 4, geom = "polygon",
  data = robberies
)

```

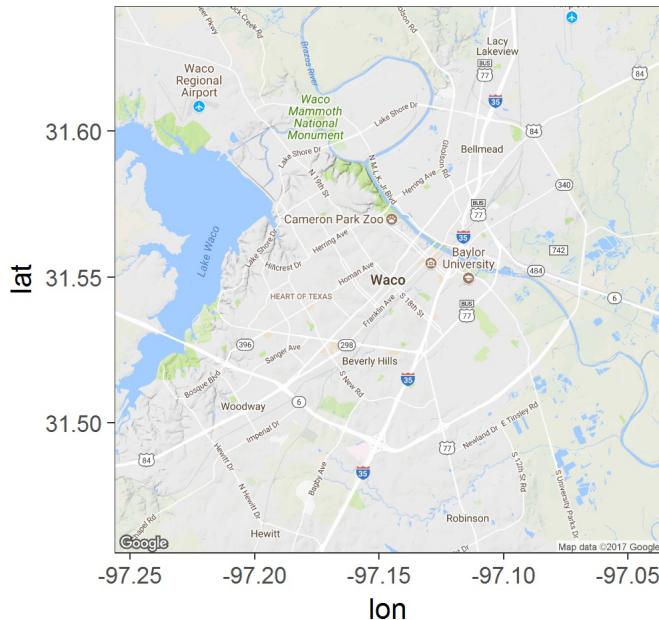
Using Another Map Source: Google Maps

In the above codes, we use Stamen Map for convenience. Google Maps can be used just as easily. However, since Google Maps use a center/zoom specification, their input is a bit different:

```
get_googlemap("waco texas", zoom = 12) %>% ggmap()
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=waco+texas&zoom=12&size=640x640&scale=2&maptype=terrain&sensor=false
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=waco%20texas&sensor=false
```

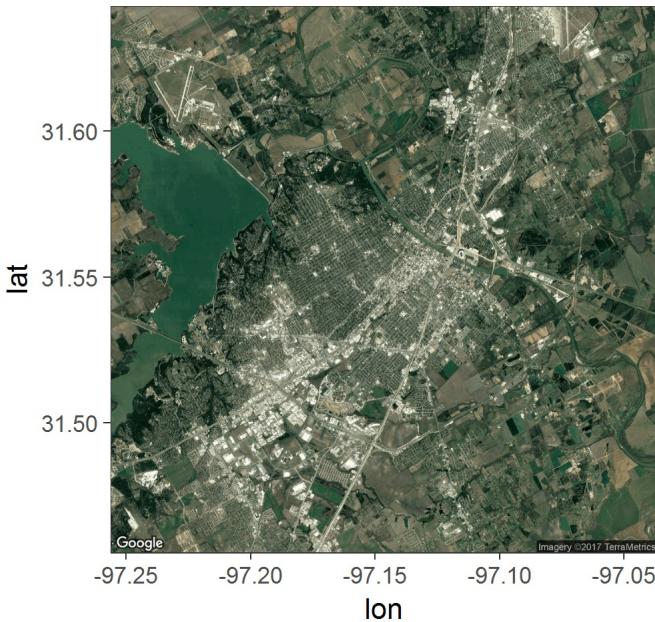


Just like Stamen Maps, we can get various different styles of Google Maps with ggmap. The above map is roadmap as default, but we can switch to satellite view:

```
get_googlemap("waco texas", zoom = 12, maptype = "satellite") %>% ggmap()
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=waco+texas&zoom=12&size=640x640&scale=2&maptype=satellite&sensor=false
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=waco%20texas&sensor=false
```

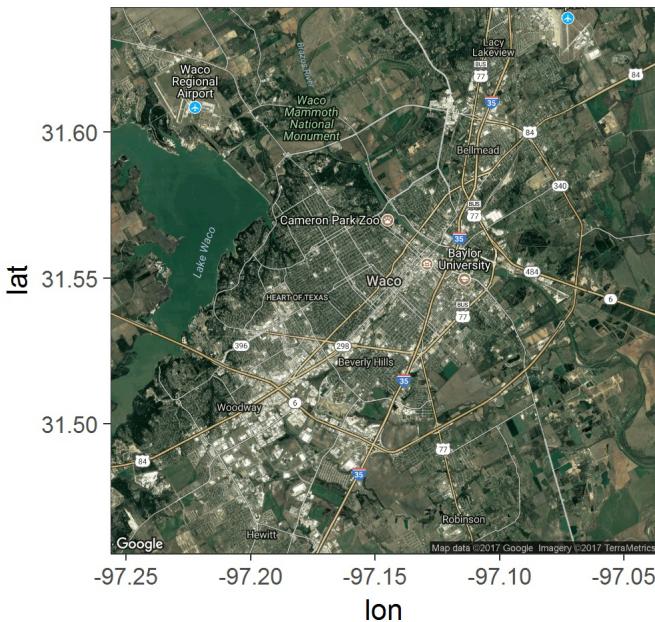


Combining them together:

```
get_googlemap("waco texas", zoom = 12, maptype = "hybrid") %>% ggmap()

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=waco+texas&zoom=12&size=640x640&scale=2&map
type=hybrid&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=waco%20texas&sensor=false
```



We can calculate the map distances, in both length and anticipated time, using mapdist().

```
mapdist(c("houston, texas", "dallas"), "waco, texas")

## by using this function you are agreeing to the terms at :

## http://code.google.com/apis/maps/documentation/distancematrix/

## Information from URL : http://maps.googleapis.com/maps/api/distancematrix/json?origins=dallas&destinations=waco
+texas&mode=driving&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/distancematrix/json?origins=houston+texas&destinatio
ns=waco+texas&mode=driving&sensor=false
```

```

##          from      to    m   km   miles seconds minutes
## 1 houston, texas waco, texas 299328 299.328 186.00242  10799 179.98333
## 2      dallas waco, texas 152484 152.484  94.75356   5348  89.13333
##   hours
## 1 2.999722
## 2 1.485556

```

We can also draw a random set of locations, and ask google map to find out the routes to get to these different points for us.

```

set.seed(500)
df <- round(data.frame(
x = jitter(rep(-95.36, 50), amount = .3),
y = jitter(rep( 29.76, 50), amount = .3)
), digits = 2)
map <- get_googlemap('houston', markers = df, path = df, scale = 2)

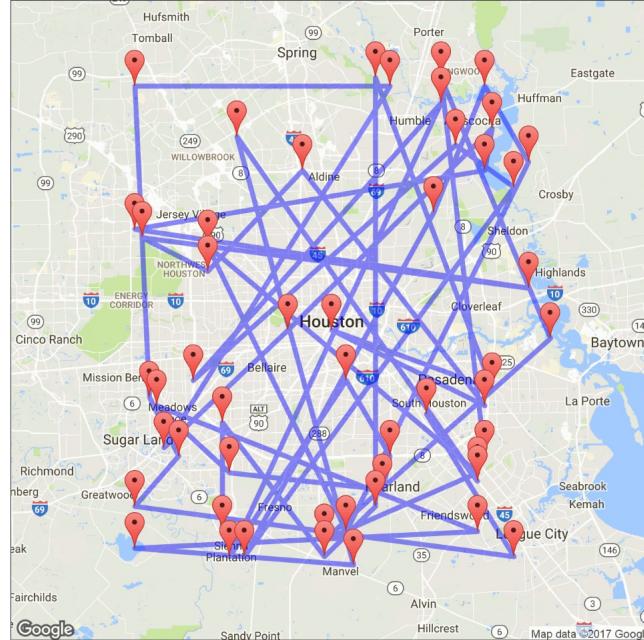
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=houston&zoom=10&size=640x640&scale=2&maptype=terrain&markers=29.6,-95.16%7C30.05,-95.22%7C29.74,-95.07%7C29.5,-95.38%7C29.58,-95.17%7C29.85,-95.54%7C29.51,-95.35%7C29.95,-95.1%7C30.04,-95.16%7C29.89,-95.23%7C29.48,-95.49%7C29.92,-95.12%7C29.97,-95.2%7C29.69,-95.56%7C30.02,-95.22%7C29.48,-95.51%7C29.69,-95.35%7C29.6,-95.29%7C29.56,-95.3%7C29.58,-95.51%7C29.66,-95.61%7C29.51,-95.17%7C29.49,-95.64%7C29.47,-95.34%7C29.98,-95.5%7C29.57,-95.17%7C29.65,-95.24%7C29.48,-95.12%7C29.54,-95.64%7C29.6,-95.5%7C29.67,-95.62%7C30.04,-95.64%7C30.04,-95.29%7C29.61,-95.6%7C29.75,-95.43%7C29.99,-95.15%7C29.94,-95.16%7C29.87,-95.64%7C29.8,-95.1%7C29.86,-95.63%7C29.82,-95.54%7C29.94,-95.41%7C29.66,-95.16%7C29.75,-95.37%7C29.68,-95.15%7C30.05,-95.31%7C29.54,-95.31%7C29.48,-95.38%7C29.64,-95.52%7C29.51,-95.52&path=29.6,-95.16%7C30.05,-95.22%7C29.74,-95.07%7C29.5,-95.38%7C29.58,-95.17%7C29.85,-95.54%7C29.51,-95.35%7C29.95,-95.1%7C30.04,-95.16%7C29.89,-95.23%7C29.48,-95.49%7C29.92,-95.12%7C29.97,-95.2%7C29.69,-95.56%7C30.02,-95.22%7C29.48,-95.51%7C29.69,-95.35%7C29.6,-95.29%7C29.56,-95.3%7C29.58,-95.51%7C29.66,-95.61%7C29.51,-95.17%7C29.49,-95.64%7C29.47,-95.34%7C29.98,-95.5%7C29.57,-95.17%7C29.65,-95.24%7C29.48,-95.12%7C29.54,-95.64%7C29.6,-95.5%7C29.67,-95.62%7C30.04,-95.64%7C30.04,-95.29%7C29.61,-95.6%7C29.75,-95.43%7C29.99,-95.15%7C29.94,-95.16%7C29.87,-95.64%7C29.8,-95.1%7C29.86,-95.63%7C29.82,-95.54%7C29.94,-95.41%7C29.66,-95.16%7C29.75,-95.37%7C29.68,-95.15%7C30.05,-95.31%7C29.54,-95.31%7C29.48,-95.38%7C29.64,-95.52%7C29.51,-95.52&path=29.6,-95.16%7C30.05,-95.22%7C29.74,-95.07%7C29.5,-95.38%7C29.58,-95.17%7C29.85,-95.54%7C29.51,-95.35%7C29.95,-95.1%7C30.04,-95.16%7C29.89,-95.23%7C29.48,-95.49%7C29.92,-95.12%7C29.97,-95.2%7C29.69,-95.56%7C30.02,-95.22%7C29.48,-95.51%7C29.69,-95.35%7C29.6,-95.29%7C29.56,-95.3%7C29.58,-95.51%7C29.66,-95.61%7C29.51,-95.17%7C29.49,-95.64%7C29.47,-95.34%7C29.98,-95.5%7C29.57,-95.17%7C29.65,-95.24%7C29.48,-95.12%7C29.54,-95.64%7C29.6,-95.5%7C29.67,-95.62%7C30.04,-95.64%7C30.04,-95.29%7C29.61,-95.6%7C29.75,-95.43%7C29.99,-95.15%7C29.94,-95.16%7C29.87,-95.64%7C29.8,-95.1%7C29.86,-95.63%7C29.82,-95.54%7C29.94,-95.41%7C29.66,-95.16%7C29.75,-95.37%7C29.68,-95.15%7C30.05,-95.31%7C29.54,-95.31%7C29.48,-95.38%7C29.64,-95.52%7C29.51,-95.52&sensor=false

## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=houston&sensor=false

ggmap(map, extent = 'device')

## Warning: `panel.margin` is deprecated. Please use `panel.spacing` property
## instead

```



Conclusion

Introduced by the lab materials in stat133, in this post, I have further discussed various geographic data visualization methods in the useful R package ggmap. By covering from the basic map extraction from sources like Stamen Maps and Google Maps, to some advanced features to incorporate more information onto the plot, I hope to introduce readers to this tool and encourage you to further explore other available features.

There are many more complex functions in ggmap that are widely used by data analysts and statisticians. If you are interested in learning beyond what have been covered in this post, here are some resources that you can look into:

- The [ggmap reference manual](#) by CRAN
- [Introduction to visualising spatial data in R](#) by researchers at the University of Leeds
- [ggmap: Spatial Visualization with ggplot2](#) by David Kahle and Hadley Wickham

References (linked in the post)

- [ggmap](#)
- [ggplot2](#)
- [dplyr](#)
- [datasets](#)
- [outside source 1](#)
- [outside source 2](#)
- [outside source 3!](#)