Data Visualization assignment 2

Jiangshan Ni jn2587

3/7/2017

1.Overall Location

To explore the location of Airbnb places, I use the ggplot2 to graph the New York City and got the hotpot in the high density area to emphasize the popular places where airbnb are located.First, I transform the spatial data to make preparation for the ggmap. I make data wrangling by using ggplot2 package. Then, I graph the desity of popular airbnb to show the location in the map and annotate the name of places. The figure 1 shows the hot pot of airbnb listings. The figure 2 shows thedensity of the listings and highlight the hotspots for AirBnB locations. I provide a second and third map in which I summarize the density of these listings.

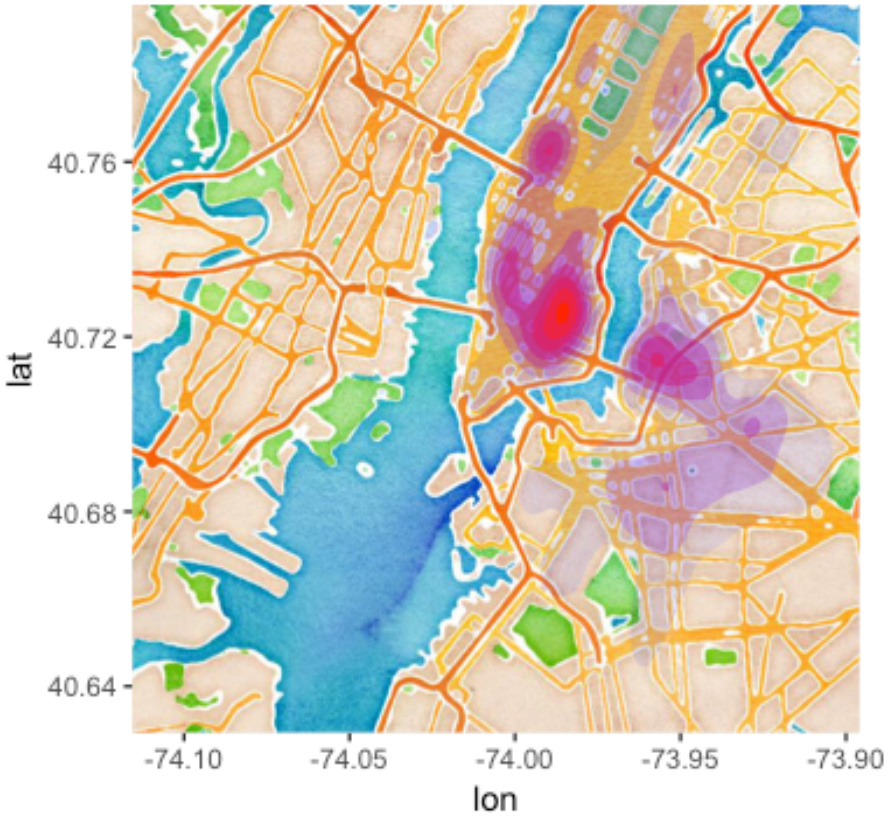
```
library(sp)
library(rgdal)

library(maptools)

## Checking rgeos availability: TRUE

library("RColorBrewer")
library("dplyr")

library("ggplot2")
library("ggmap")
library("sp")
library("rgeos")

library("rgdal")
library("geosphere")
nyc_map <- readOGR("./nybb_17","nybb")

## OGR data source with driver: ESRI Shapefile
## Source: "./nybb_17", layer: "nybb"
## with 5 features
## It has 4 fields

proj4string(nyc_map) <- CRS("+proj=tmerc +lat_0=0 +lon_0=9 +k=1 +x_0=3500000
+y_0=0 +datum=potsdam +units=m +no_defs")

g <- ggmap(m)
```

The figure 1 shows the hot pot of airbnb listings.

```
nyc_map <- readOGR("./nybb_17","nybb")

## OGR data source with driver: ESRI Shapefile
## Source: "./nybb_17", layer: "nybb"
## with 5 features
## It has 4 fields
```

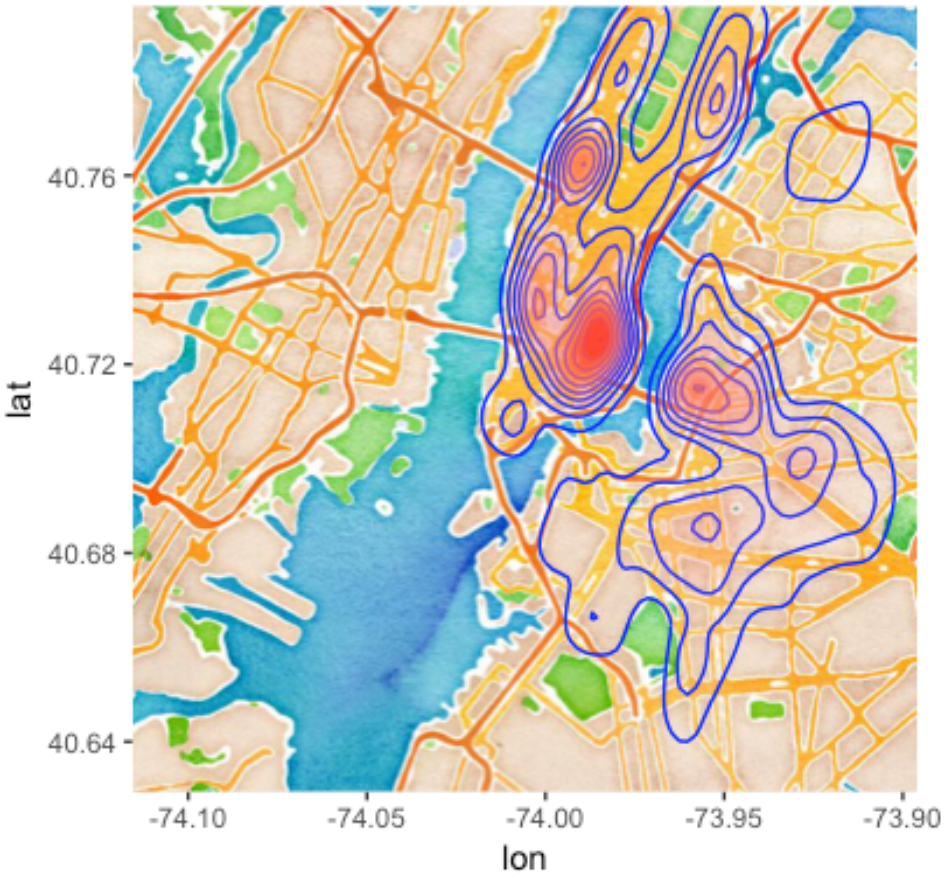
```
proj4string(nyc_map) <- CRS("+proj=tmerc +lat_0=0 +lon_0=9 +k=1 +x_0=3500000
+y_0=0 +datum=potsdam +units=m +no_defs")
nyc_map <- spTransform(nyc_map, CRS("+proj=longlat +datum=WGS84"))
nyc_map <- fortify(nyc_map)
library(ggmap)
m <- get_map("New York City",zoom=12,maptype="watercolor",source="stamen")
g <- ggmap(m)
g

library(readr)
airbnb1 <- read_csv("./airbnb_listings.csv")
hotpot <- g +
        stat_density2d(aes(x = longitude,y=latitude, fill = ..level.., alpha
=..level..),
                       data=airbnb1, geom="polygon") +
                       scale_fill_gradient(low = "blue", high = "red") +
                       theme(legend.position = "none")
hotpot
```

Figure 1.Overlooking of Airbnb Apartments



The figure 2 shows thedensity of the listings and highlight the hotspots for AirBnB locations. I provide a second and third map in which I summarize the density of these listings.

```
g <- ggmap(m)
g + geom_density2d(aes(x=longitude,y=latitude), data=airbnb1,
                   color="blue", size=0.5, bins=10) +
  stat_density2d(aes(x=longitude,y=latitude,fill = ..level.., alpha = ..leve
l..),
                 data=airbnb1, geom = 'polygon', bins=12) +
  scale_fill_gradient2(low = "green", mid="pink", high = "red") +
  scale_alpha(range = c(0.00, 0.5)) + theme(legend.position = "none")
```

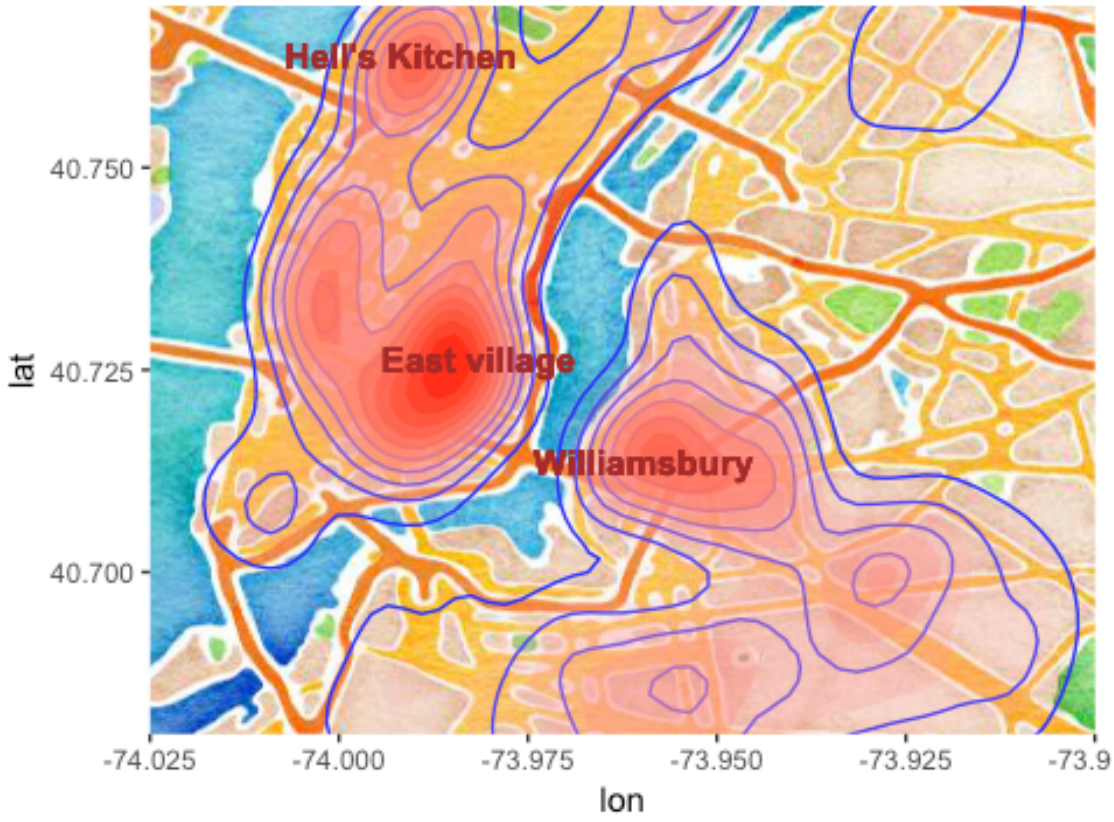


The figure 2 shows thedensity of the listings and highlight the hotspots for AirBnB locations.

```
G <- ggmap(m)
g + geom_density2d(aes(x=longitude,y=latitude), data=airbnb1,
                   color="blue", size=0.5, bins=10) +
  stat_density2d(aes(x=longitude,y=latitude,fill = ..level.., alpha = ..leve
l..),
                 data=airbnb1, geom = 'polygon', bins=12) +
  scale_fill_gradient2(low = "green", mid="pink", high = "red") +
  theme(legend.position = "none") +
                          coord_cartesian(xlim = c(-74.025, -73.90), ylim= c
(40.68, 40.77)) +
                          geom_text(x = -73.9815, y = 40.7265, label = "East
```

```
 village",
                                        color = "brown", fontface = 2, size = 4)
 +
                                geom_text(x = -73.9918, y = 40.7638, label = "Hel
l's Kitchen",
                                        color = "brown", fontface = 2, size = 4)
 +
                                geom_text(x = -73.9597, y = 40.7136, label = "Will
iamsbury",
                                        color = "brown", fontface = 2, size = 4)
 +
                                ggtitle("Fig 2: Hotpots of Airbnb Apartments") +
                                theme(plot.title = element_text(hjust = 0.5)) +
                                guides(alpha = FALSE)
```

Figure 2.Hotpots of Airbnb Apartments



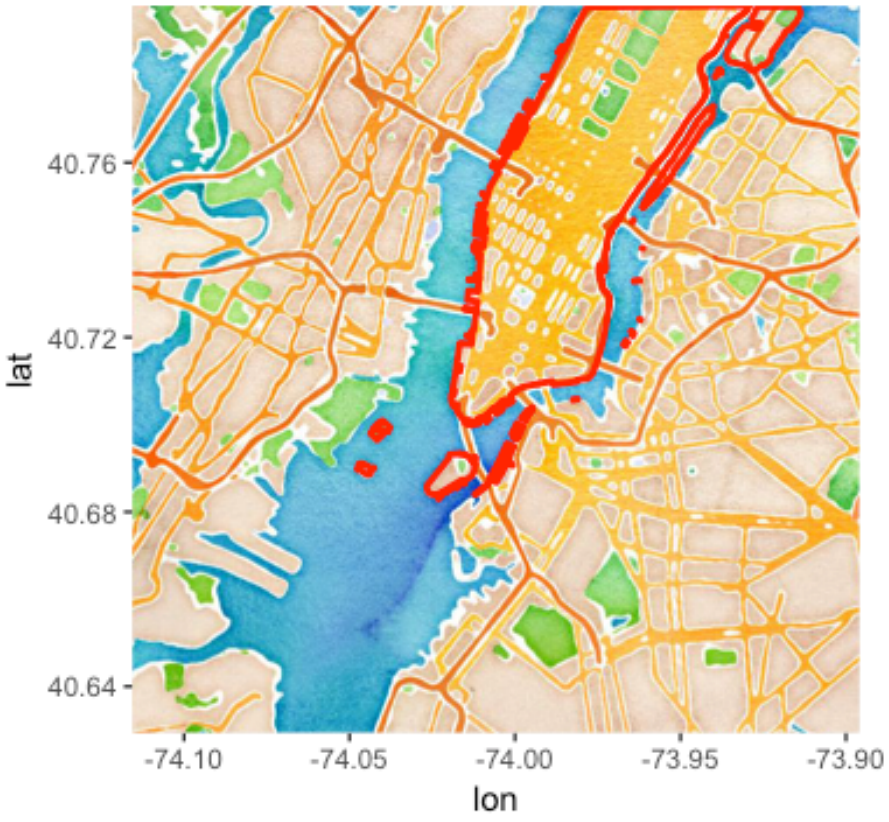
2.Renting out your partment vs. permanent rentals

I diveded the day of availability into 2 subsets. I assume that the permanent rental has 240 availability of days(10 month). The rest of them are semi-permanent rental(less than 10 month). I also make the polygon of New York city to ensure the polygon are displayed in the map. For the data wrangling, I select some key variables at first. And I use the monthly

price of each Airbnb multiplies by number of month. I get the mean price by using mean function and summarize the mean price of permanent rental Airbnb and semi-permanent Airbnb. Then we plot the price and airbnb listing(permanent and semi-permanent) in the map. I try to find more details about the permanent. I choose to find hotspots of permanent and semi-permanent. Based on both non-map and map graphs, I make the result that the price of permanent rental are much higher than semi-permanent rental Airbnb. I also highlight the neighborhoods were most listings appear to be permanent or semi-permanent rentals.

```
nyc_map <- readOGR("./nybb_17/.","nybb")

## OGR data source with driver: ESRI Shapefile
## Source: "./nybb_17/.", layer: "nybb"
## with 5 features
## It has 4 fields

nyc_map <- subset(nyc_map, BoroName=="Manhattan")
nyc_map <- spTransform(nyc_map, CRS("+proj=longlat +datum=WGS84"))
nyc_map <- fortify(nyc_map)
map_census <- ggmap(m) +    # our raster map from before
  geom_polygon(aes(x=long, y=lat, group=group),
  size=1, color='red', data=nyc_map, alpha=0)
map_census
```

Figure 3. Polygon of NYC



```
library(dplyr)
air <- read.csv('/Users/Anna/Downloads/airbnb_listings.csv', header= TRUE, st
ringsAsFactors= FALSE)
subair<- air[,c("availability_365","price","latitude","longitude","availabili
ty_30","review_scores_rating","room_type")]
group_airbnb2 <- group_by(air, room_type, as.factor(air$availability_365))
summarise(group_airbnb2,
          mean_price = mean(price),
          mean_monthly_income = mean(price * availability_30),
          mean_rating = mean(review_scores_rating))
```

```
## Source: local data frame [973 x 5]
## Groups: room_type [?]
##
##          room_type `as.factor(air$availability_365)` mean_price
##              <chr>                             <fctr>      <dbl>
## 1  Entire home/apt                                  0   198.6291
## 2  Entire home/apt                                  1   168.1667
## 3  Entire home/apt                                  2   171.2762
## 4  Entire home/apt                                  3   171.7311
## 5  Entire home/apt                                  4   197.7456
## 6  Entire home/apt                                  5   172.7190
## 7  Entire home/apt                                  6   177.6562
## 8  Entire home/apt                                  7   176.3125
## 9  Entire home/apt                                  8   200.4397
## 10 Entire home/apt                                  9   180.9238
## # ... with 963 more rows, and 2 more variables: mean_monthly_income <dbl>,
## #   mean_rating <dbl>
```

I diveded the day of availability into 2 subsets. I assume that the permanent rental has 240 availability of days(10 month). The rest of them are semi-permanent rental.

```
air <- read.csv('/Users/Anna/Downloads/airbnb_listings.csv', header= TRUE, st
ringsAsFactors= FALSE)

air$availability_365[air$availability_365 > 300 & air$availability_365 <= 36
5] <- 10
sub_airbnb <- air[, c( "availability_365","room_type", "price", "availability
_30",
                       "review_scores_rating", "longitude", "latitude" )]
sub_airbnb <- na.omit(sub_airbnb)
sub_airbnb$availability_365 <- ifelse(sub_airbnb$availability_365 == 10, 1,
0)
group3 <- group_by(sub_airbnb, as.factor(sub_airbnb$availability_365))
summarise(group3, mean_monthly_income = mean(price * availability_30),
          mean_price = mean(price),  mean_rating = mean(review_scores_ratin
g))
```

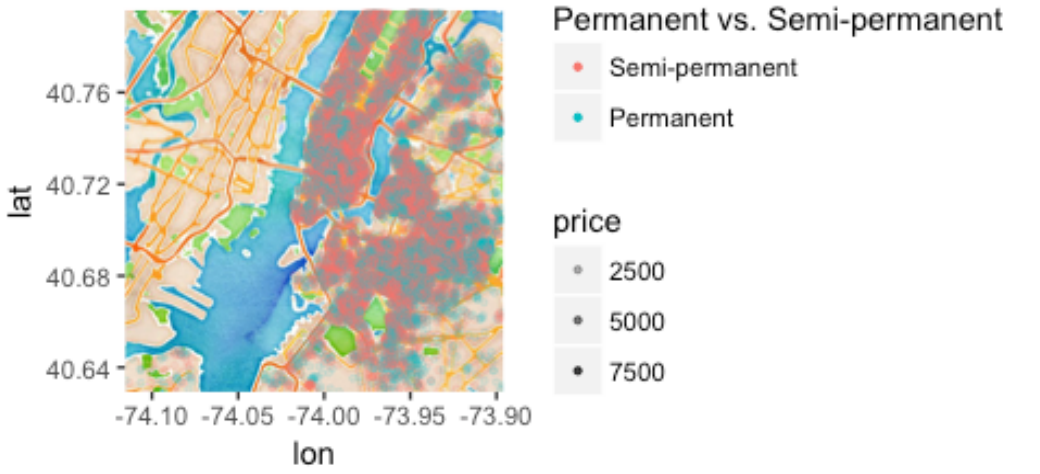
```
## # A tibble: 2 × 4
##   `as.factor(sub_airbnb$availability_365)` mean_monthly_income mean_price
##                                      <fctr>               <dbl>      <dbl>
```

```
## 1                                          0            883.3134   139.7786
## 2                                          1           2264.3776   148.5070
## # ... with 1 more variables: mean_rating <dbl>
```

Then we plot the price and airbnb listing(permanent and semi-permanent) in the map. I try to find more details about the permanent. I choose to find hotspots of permanent and semi-permanent.

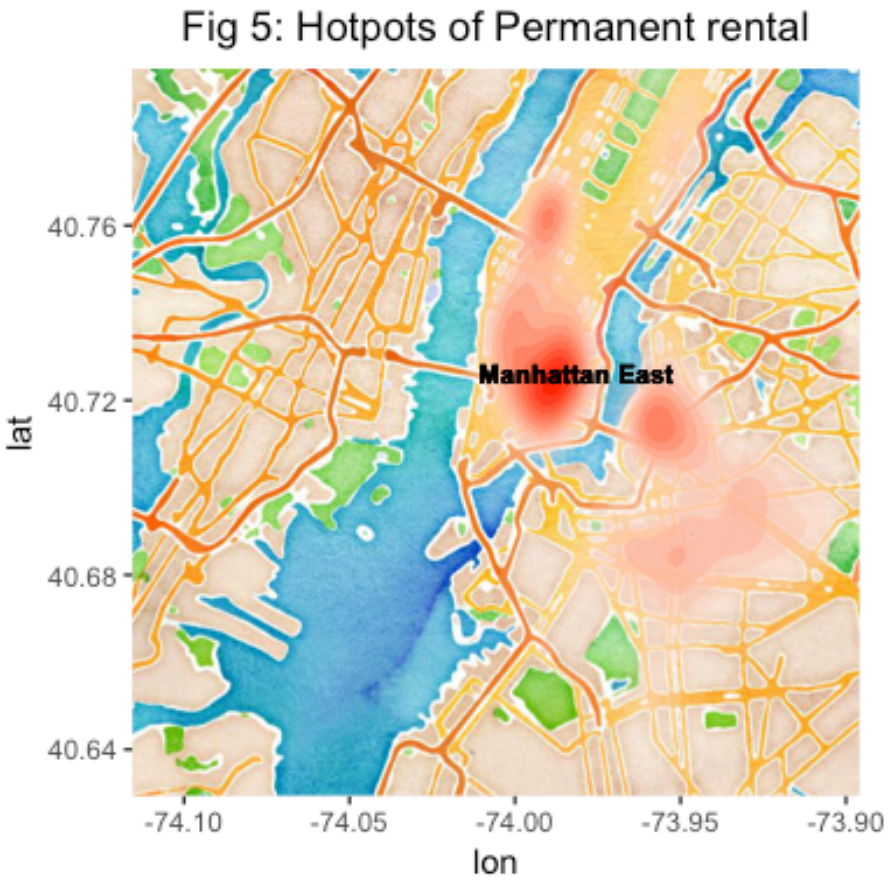
```
permanent <- g + geom_point(data = sub_airbnb, aes(x = longitude, y=latitude,

                                                   color = as.factor(availability_36
5),                                                                              a
lpha = price), size = 0.5) +
              scale_color_discrete(name="Permanent vs. Semi-permanent",
                                   breaks=c("0", "1"),
                                   labels=c("Semi-permanent", "Permanent"))

permanent
```

Figure 4. price and permanent/semi-permanent rental



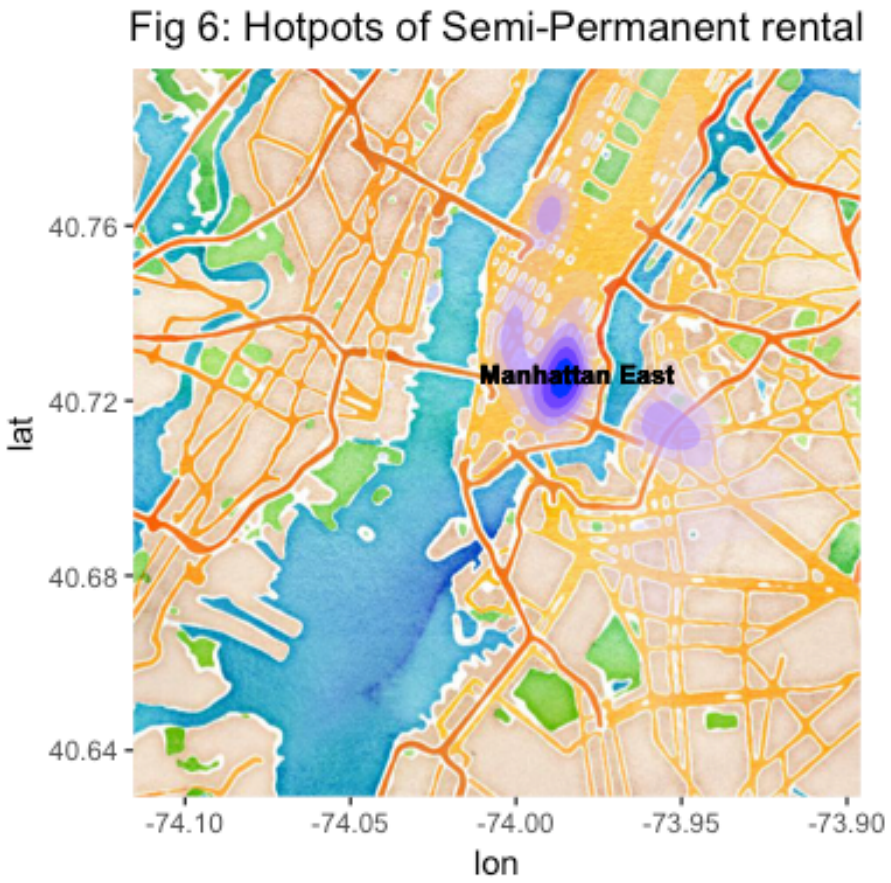

```
library(ggplot2)
perma <- filter(sub_airbnb, availability_365 == 1)
semi_perma <- filter(sub_airbnb, availability_365 == 0)
permanent_hotpot <- g + stat_density2d(aes(x = longitude,y=latitude, fill =
 ..level.., alpha=..level..),
                        data = perma, geom="polygon") +
```

```
                        scale_fill_gradient(low = "white", high = "red") +
                        geom_text(x = -73.9815, y = 40.7265, label = "Manhatt
an East",
                                  color = "black", fontface = 2, size = 3)
 +
                        theme(legend.position = "none") +
                        ggtitle("Fig 5: Hotpots of Permanent rental") +
                        theme(plot.title = element_text(hjust = 0.5))
permanent_hotpot
```




```
library(ggplot2)
perma <- filter(sub_airbnb, availability_365 == 1)
semi_perma <- filter(sub_airbnb, availability_365 == 0)
semi_perma_hotpot <- g +
                    stat_density2d(aes(x = longitude,y=latitude, fill = ..leve
l.., alpha=..level..),
                                   data = semi_perma, geom="polygon") +
                    scale_fill_gradient(low = "white", high = "blue") +
                    geom_text(x = -73.9815, y = 40.7265, label = "Manhattan Ea
st",
                                   color = "black", fontface = 2, size = 3)
 +
                    theme(legend.position = "none") +
```

```
                    ggtitle("Fig 6: Hotpots of Semi-Permanent rental") +
                    theme(plot.title = element_text(hjust = 0.5))

semi_perma_hotpot
```

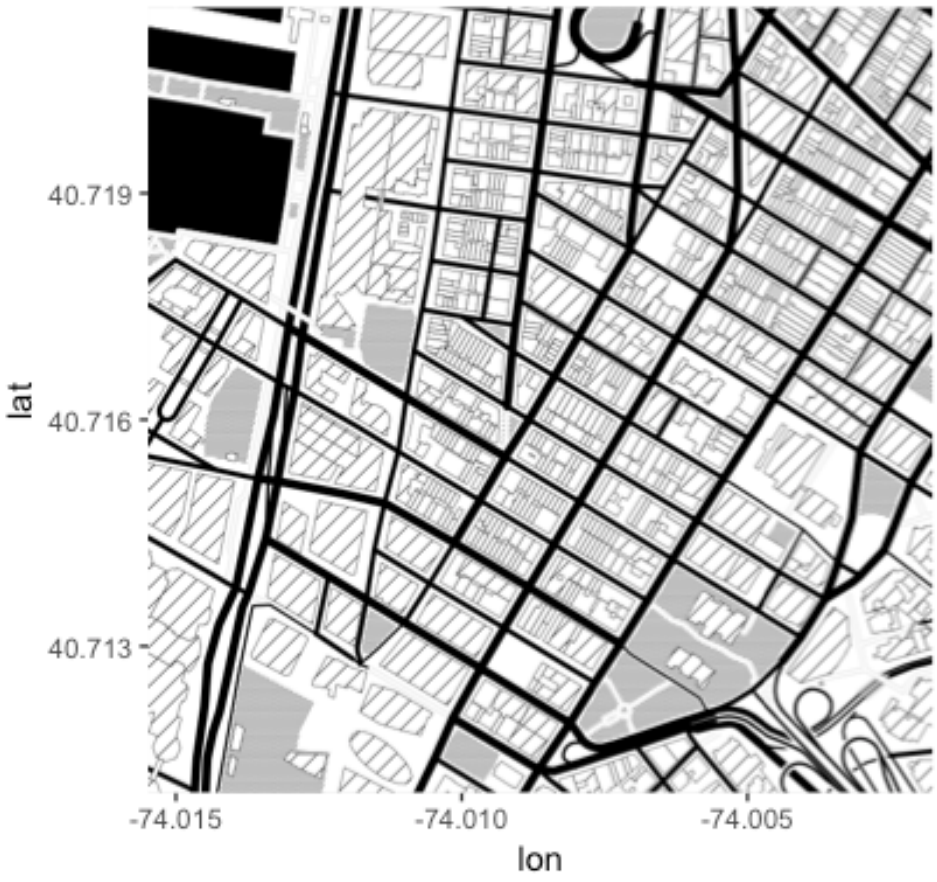


3. AirBnB and Subway Access

I choose Tribeca as the selected neighbourhood in Manhattan. I want to explore how the location, type, and features of AirBnB listings are related to subway access. First, I mapped the location of subway within the Tribeca neighbourhood. Then I added the Airbnb location to the graph to display the relationship between subway station and Airbnb locations. From the figure, we see the main style of Airbnb is the entire rentals. Also, the prices in this area are really high and the location are coverge the lower Manhattan area. The minimium distance between every Airbnb Listings and the nearest subway stations is 5.3887. By using different buffers both in the non-map calculation and map display, I could find that the number of Airbnb listings of smaller buffers are 7 and the number of Airbnb listings of larger buffers are 44. Also, I calculate the mean price of Airbnb which are located in these two different buffers.

```
library(rgeos)
library(rgdal)
nycsubway <- read.csv("nyc_subway.csv",header=TRUE)
airbnb_sub <- air[, c("name", "availability_365","availability_30","room_type
```

```
", "price",
                                "review_scores_rating", "longitude",
                                "latitude", "neighbourhood", "number_of_review
s", "accommodates")]
airbnb_sub$price <- as.numeric(substring(as.character(airbnb_sub$price),2))
airbnb_sub <- filter(airbnb_sub, neighbourhood == "Tribeca")
airbnb_sub <- na.omit(airbnb_sub)

nyc_subway <- read.csv("nyc_subway.csv", header = TRUE)
library(ggmap)
map_tribeca <- get_map("New York, Tribeca", zoom=16,
                   source="stamen",maptype="toner-background")

ggmap(map_tribeca)
```




```
nyc_subway_sel <- nyc_subway[, c("Line", "Station.Name", "Station.Latitude",
                                 "Station.Longitude", "Route1", "Route2")]
subway_tribeca <- filter(nyc_subway_sel, Station.Latitude >= 40.710 & Statio
n.Latitude <= 40.720,
                     Station.Longitude >= -74.015 & Station.Longitude <= -74.
005)
airbnb_sub2 <- air[, c("name", "availability_365","room_type", "price",
                                "availability_30", "review_scores_rating", "lo
ngitude",
```

```
                                "latitude", "neighbourhood", "number_of_review
s", "accommodates")]
airbnb_sub2$price <- as.numeric(substring(as.character(airbnb_sub2$price),2))

airbnb_sub2 <- filter(airbnb_sub2, neighbourhood == "Tribeca")
airbnb_sub2 <- na.omit(airbnb_sub2)

nyc_sub <- ggmap(map_tribeca) + geom_point(data = airbnb_sub2, aes(x = longit
ude, y = latitude,
                                           color = room_type, size = price,

                                           alpha = availability_365)) +
  geom_path(data = filter(subway_tribeca, Line == "Broadway"),aes(x = Statio
n.Longitude, y = Station.Latitude),
                  color = "light blue", linetype = 1, size = 1, alpha = 0.8)
+ geom_path(data = filter(subway_tribeca, Line == "8 Avenue"),
                  aes(x = Station.Longitude, y = Station.Latitude),
                  color = "mediumorchid1", linetype = 1, size = 1, alpha = 0.
8)+geom_path(data = filter(subway_tribeca, Line == "Lexington"),
                  aes(x = Station.Longitude, y = Station.Latitude),
                  color = "lightpink1", linetype = 1, size = 1, alpha = 0.8)
+ geom_point(data = subway_tribeca, aes(x = Station.Longitude, y = Station.La
titude),
                   size = 2, alpha = 0.8, shape = 6, color = "orange") +
        geom_text(data = subway_tribeca, aes(x = Station.Longitude, y = Stati
on.Latitude),
                  label = subway_tribeca$Station.Name, size = 2, alpha = 0.5)
 +
         ggtitle("Fig 7: AirBnB and Subway Access") +
                   theme(plot.title = element_text(hjust = 0.5))
nyc_sub
```


```
airbnb_sub3 <- air[, c("name", "availability_365","room_type", "price",
                                "availability_30", "review_scores_rating", "lo
ngitude",
                                "latitude", "neighbourhood", "number_of_review
s", "accommodates")]
airbnb_sub3$price <- as.numeric(substring(as.character(airbnb_sub3$price),2))

airbnb_sub3 <- filter(airbnb_sub3, neighbourhood == "Tribeca")
airbnb_sub3 <- na.omit(airbnb_sub3)
x  <- c(1:nrow(airbnb_sub3))
for(i in x){
        airbnb_sub3[i,"min_dist"] <- min(distVincentyEllipsoid(airbnb_sub3[i,
 c("longitude", "latitude")], subway_tribeca[, c( "Station.Longitude", "Stati
on.Latitude")]))
}
airbnb_sub3[,c("name", "min_dist")]

##                                    name min_dist
## 1   Charming 1BR Prospect Heights Apt 5388.732
## 4 Large Bedroom in Apt. Over Cute Bar 5588.879
## 5    Brooklyn, Sundrenched & Peaceful 6517.790
## 6       Sunny Duplex on Prospect Park 6868.607
```

```
## 8                  Private Room & Bath 6315.012
## 9      BEAUTIFUL LOFT!!! Williamsburg 3001.333
```

```
range(airbnb_sub3$min_dist)
```

```
## [1] 3001.333 6868.607
```

```
library(rgdal)
neighbor <- readOGR("neighbourhoods.geojson")
```

```
## OGR data source with driver: GeoJSON
## Source: "neighbourhoods.geojson", layer: "OGRGeoJSON"
## with 233 features
## It has 2 fields
```

```
t <- neighbor$neighbourhood == c("Tribeca")
neighbor_t <- neighbor[t, ]
```

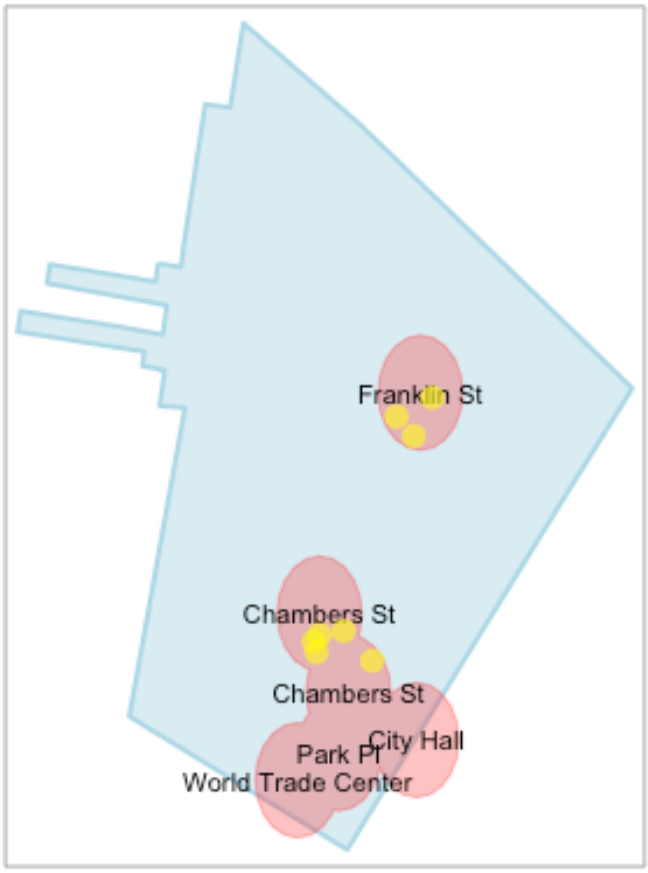
```
sub_stop <- readOGR("./nybb_17/", "stops_nyc_subway_jan2017")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "./nybb_17/", layer: "stops_nyc_subway_jan2017"
## with 493 features
## It has 8 fields
```

```
select_sub <- sub_stop$stop_lon >= -74.015& sub_stop$stop_lon <= -74.005 & su
b_stop$stop_lat >= 40.712 & sub_stop$stop_lat <= 40.720
sub_b <- sub_stop[select_sub, ]
sub_b <- spTransform(sub_b, CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=W
GS84 +towgs84=0,0,0"))
airbnb_sub3 <- air[, c("name", "availability_365","room_type", "price",
                               "availability_30", "review_scores_rating", "lo
ngitude",
                               "latitude", "neighbourhood", "number_of_review
s", "accommodates")]
airbnb_sub3$price <- as.numeric(substring(as.character(airbnb_sub3$price),2))

airbnb_sub3 <- filter(airbnb_sub3, neighbourhood == "Tribeca")
airbnb_sub3 <- na.omit(airbnb_sub3)
coordinates(airbnb_sub3) <- ~ longitude + latitude
geo_airbnb <- SpatialPointsDataFrame(coordinates(airbnb_sub3), as.data.frame
(airbnb_sub3),
                                     proj4string =CRS(proj4string(sub_b)))
buf1 <- gBuffer(sub_b, width = 0.001)
buf2 <- gBuffer(sub_b, width = 0.002)
buf3 <- gBuffer(sub_b, width = 0.003)

buffer1 <- gBuffer(sub_b, width = 0.001)
buffer2 <- gBuffer(sub_b, width = 0.002)
library(tmap)
library(rgeos)
neighbor <- readOGR("neighbourhoods.geojson")
```

```
## OGR data source with driver: GeoJSON
## Source: "neighbourhoods.geojson", layer: "OGRGeoJSON"
## with 233 features
## It has 2 fields
```

```
b <- neighbor$neighbourhood == c("Tribeca")
neighbor_b<- neighbor[b, ]
tm_shape(neighbor_b)+   tm_fill(col = "lightblue", alpha = 0.5)+tm_borders(co
l = "lightblue", lwd =2)+  tm_shape(buffer1) + tm_borders(col = "red", alpha=
0.3)+ tm_fill(col = "red", alpha = 0.3)+tm_shape(sub_b)+tm_text("stop_name",
size = 0.7, shadow = FALSE, bg.alpha= 0,bg.color="peachpuff3", remove.overlap
=FALSE)+  tm_shape(geo_airbnb[buf1,])+ tm_bubbles(size = 0.3, col = "yellow",
 alpha = 0.6)
```




```
library("raster")
```

```
##
## Attaching package: 'raster'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
list_buf1 <- na.omit(extract(buf1, geo_airbnb))
sel1 <- list_buf1$point.ID
group1 <- geo_airbnb@data[sel1, c("price", "room_type", "accommodates","numbe
r_of_reviews")]
mean1 <- group_by(group1 , room_type) %>%
        summarise(mean_price = mean(price))
mean2 <- group_by(group1 , accommodates) %>%
        summarise(mean_price = mean(price))
mean1
```

```
## # A tibble: 2 × 2
##         room_type mean_price
##            <fctr>      <dbl>
## 1 Entire home/apt   42.20000
## 2    Private room   20.66667
```

```
mean2
```

```
## # A tibble: 5 × 2
##   accommodates mean_price
##          <int>      <dbl>
## 1            1      15.00
## 2            2      35.75
## 3            4      90.00
## 4            6      25.00
## 5            7       0.00
```

```
buffer1 <- gBuffer(sub_b, width = 0.001)
```

```
## Warning in gBuffer(sub_b, width = 0.001): Spatial object is not projected;
## GEOS expects planar coordinates
```

```
buffer2 <- gBuffer(sub_b, width = 0.003)
```

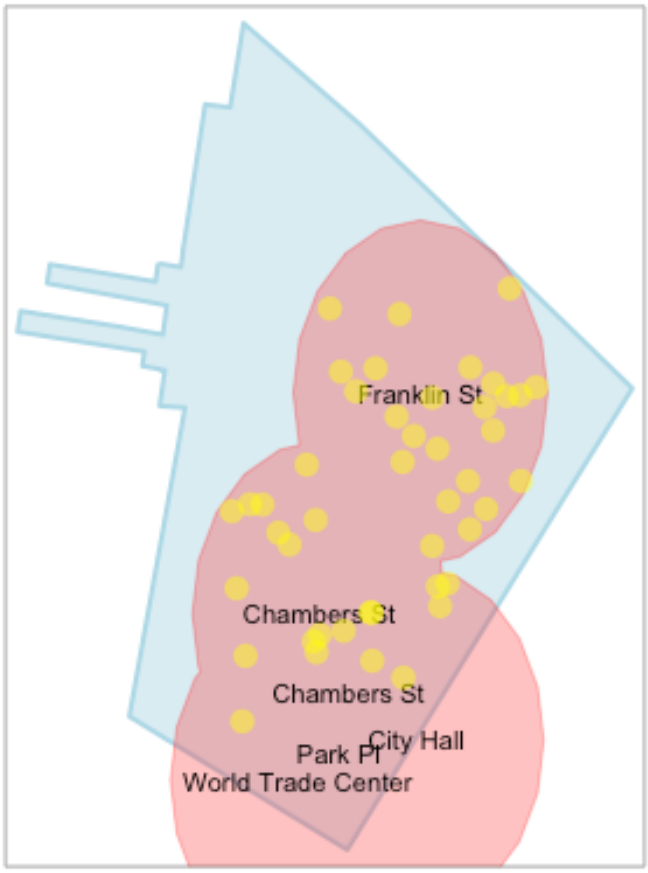
```
## Warning in gBuffer(sub_b, width = 0.003): Spatial object is not projected;
## GEOS expects planar coordinates
```

```
library(tmap)
library(rgeos)
neighbor <- readOGR("neighbourhoods.geojson")
```

```
## OGR data source with driver: GeoJSON
## Source: "neighbourhoods.geojson", layer: "OGRGeoJSON"
## with 233 features
## It has 2 fields
```

```
b <- neighbor$neighbourhood == c("Tribeca")
neighbor_b<- neighbor[b, ]
tm_shape(neighbor_b)+ tm_fill(col = "lightblue", alpha = 0.5) +tm_borders(col
 = "lightblue", lwd =2)+  tm_shape(buffer2) + tm_borders(col = "red", alpha=
0.3)+ tm_fill(col = "red", alpha = 0.3)+tm_shape(sub_b)+tm_text("stop_name",
size = 0.7, shadow = FALSE, bg.alpha= 0,bg.color="peachpuff3", remove.overlap
```

```
=FALSE)+  tm_shape(geo_airbnb[buffer2,])+ tm_bubbles(size = 0.3, col = "yello
w", alpha = 0.5)
```




```
list_buffer2 <- na.omit(extract(buffer2, geo_airbnb))
sel2 <- list_buffer2$point.ID
group2 <- geo_airbnb@data[sel2, c("room_type", "accommodates","number_of_revi
ews", "price")]
mean2_1 <- group_by(group2 , room_type) %>%
        summarise(mean_price = mean(price))
mean2_2 <- group_by(group2 , accommodates) %>%
        summarise(mean_price = mean(price))
mean2_1
```

```
## # A tibble: 2 × 2
##         room_type mean_price
##            <fctr>      <dbl>
## 1 Entire home/apt   39.73333
## 2    Private room   21.26667
```

```
mean2_2
```

```
## # A tibble: 8 × 2
##   accommodates mean_price
##          <int>      <dbl>
```

```
## 1             1   10.25000
## 2             2   29.38889
## 3             3   42.66667
## 4             4   53.62500
## 5             5   47.25000
## 6             6   24.16667
## 7             7    0.00000
## 8             8   50.00000
```