STATS506 Final project

NYC data visualization

Yung-Chun Lee

We will attempt to construct a public access index using three data sets: NYC Free WiFi Hotspot Locations, NYC facilities data base and NYC subway station.To evaluate how many public access each neighborhood has in NYC.

Data Desciption

Data set	Description	Entries	Source	URL
Subway station	Subway entrances in NYC with GPS coordinates, station name	473	NYC Gov	https://goo.gl/wcQZCW
Free WiFi Hotspot Locations	Public wifi in NYC with provider name, GPS coordinates, neighbor and borough	2566	NYC Gov	https://goo.gl/gDiri4
Facilities data base	Public/ Non-Public facilities in NYC with GPS coordinates, facility type, neighbor and borough	36112	NYC Gov	https://goo.gl/5hqxuT
Poverty data	Each entry present individual information, borough, poverty unit type, race,...	69103	NYC Gov	https://goo.gl/gAnndw

- We use the data with seleceted variables and information to visualize the distribution of those facilities, wifi hotspots and subway stations, we count the points in each neighborhood and perform the visualisation as shown in *Figure 1*.
- Define the public access index as below using equal weight (*Figure 2*):

$$Public\ Access\ Index = Z_{Subway} + Z_{WiFi} + Z_{Public\ Facilities}$$

where Z is the standardized count number per neighborhood.

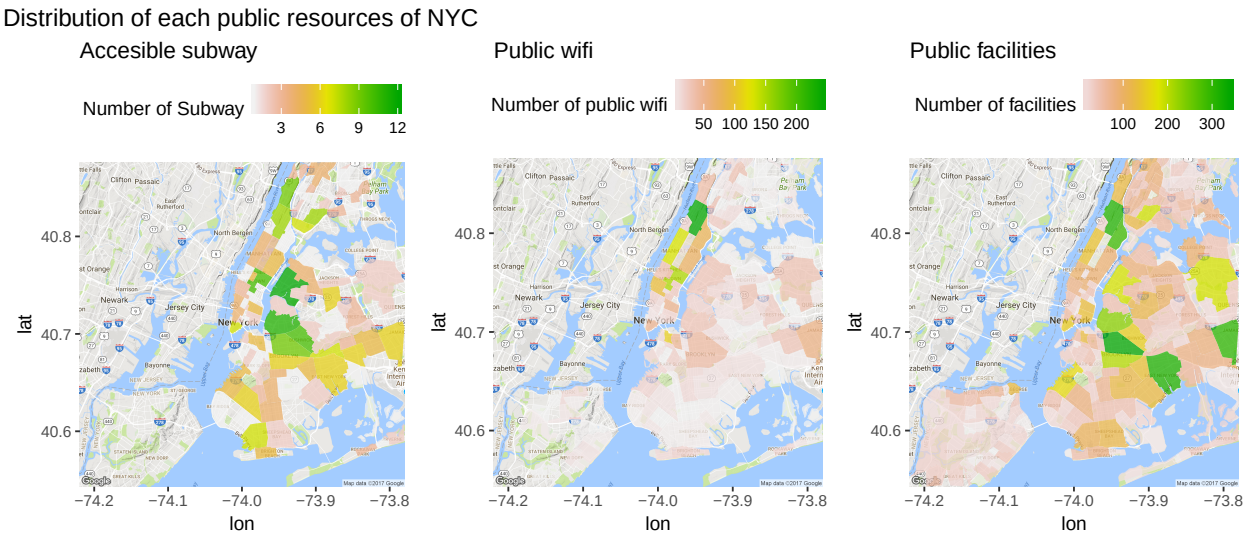

Figure 1: Distribution of each public resources of NYC
We agggregated the points from each neighborhood to produce a cleareer and more informative graphic.
*Notice that in subway dataset, Staten island Railway is not included as part of the NYC subway system.

We further discuss if there is exist any correlation between the poverty level and public access index we defineded in NYC using the 4th data set: Poverty data

- First we have to process the data to fit the format as in previous analysis format.
- Calaulate number of entries for each neighbor.
- Using **geocode** function to find the latitude and longtitude to fit in ggmap.
- Using the latitude and longtitude to fit the corresponding neighborhood in shapefile.(Since the defintion for neighbor is different for this data set and some of them are combined into a region)
- Then we visualise the output using the same shapefile and format.

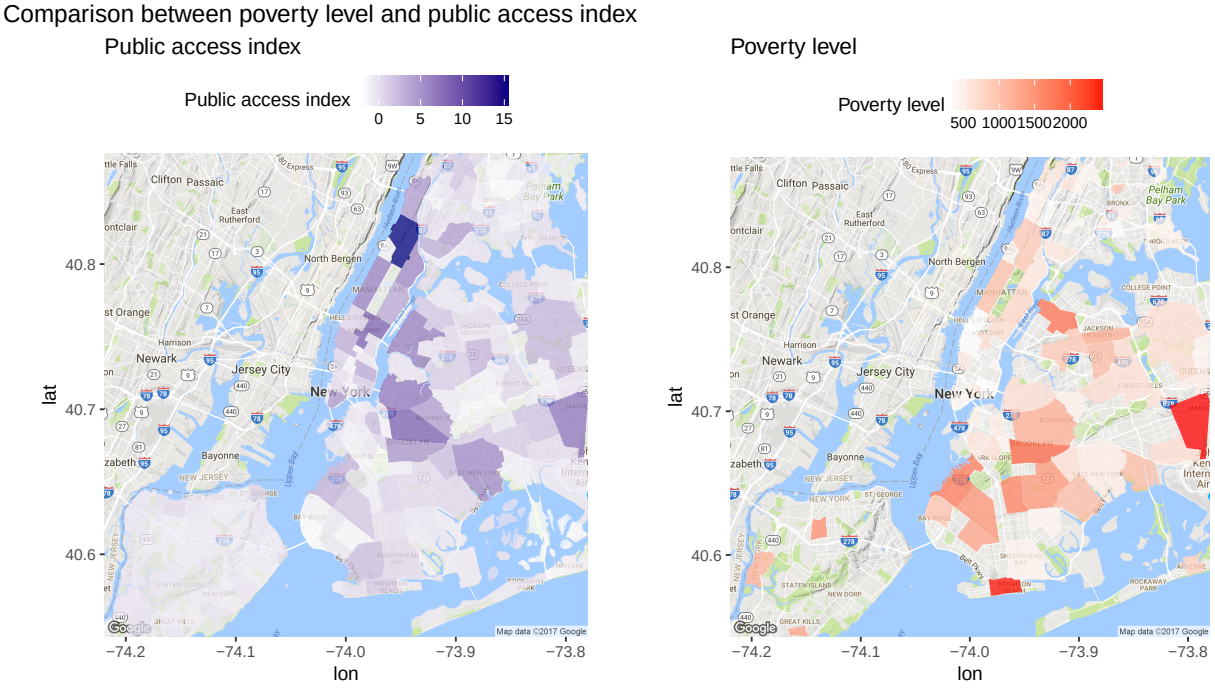

Figure 2: Comparison between poverty level and public access index
As we can observe in the plots, they show some degree of similarity, the correlation between two data set is 0.484

```
##
##  Pearson's product-moment correlation
##
## data:  pcindex[, 2] and povindex[, 2]
## t = 9.0044, df = 264, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3870164 0.5716694
## sample estimates:
##       cor
## 0.4847254
```

Conslusion

- From the correlation test we can know that the correlation between public access index and poverty level are not independent, the 95% confidence interval is (0.38,0.57)
- We can conclude that there's some certain relationship between poverty and the amount of public services in the area.

Appendix

```
#########################################################
# Course : STAT 506
# File: Fianl Project
# Author: Yung-Chun LEE
#########################################################
#Reference
#NYC maps: https://rstudio-pubs-static.s3.amazonaws.com/195412_aeb61836e07042d0afac8a1b022e54b2.html
############
# settings #
############
library(latex2exp)
library(tidyverse)
library(gridExtra)
library(ggplot2)
library(grid)
library(knitr)
library(MASS)
library(dplyr)
library(data.table)
library(stringr)
library(sp)
library(rgdal)
library(rgeos)
library(tigris)
library(leaflet)
library(sp)
library(ggmap)
library(maptools)
library(broom)
library(httr)
#setting working directory
setwd("C:/Users/user/Desktop/Umich Stat/2017 Fall/STATS_506/Final Project/")
# setting global echo = FALSE , codes in appendix
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)
##############
# data setup #
##############
subway_ori = fread("./Data/DOITT_SUBWAY_STATION_01_13SEPT2010.csv")
wifi_ori = fread("./Data/NYC_Free_Public_WiFi_03292017.csv")
facility_ori =fread("./Data/Facilities_Database.csv")
poverty_ori = fread("./Data/2015_NYC_Web_Dataset.csv", showProgress =  FALSE)

########################################
# Data Cleanup / preapre / exploratory #
########################################
f.subway_long =function(gps){
  tmp = unlist(str_split(gps," "))
  long = str_extract(tmp[2],"-?\\d.+")
  return(as.numeric(long))
}
f.subway_lat =function(gps){
```

```
  tmp = unlist(str_split(gps," "))
  lat = gsub("\\)","",tmp[3])  ###fix this
  return(as.numeric(lat))
}

subway_work = subway_ori[,.(long=f.subway_long(the_geom),
                            lat =f.subway_lat(the_geom)),by=.(NAME)]

wifi_work = wifi_ori[,.(name= OBJECTID,
                        long=LON,
                        lat =LAT),by=.(PROVIDER)]

facility_work = facility_ori[,.(name=facname,
                                type=facdomain,
                                public=ifelse(optype=="Public",1,0),
                                long= longitude,
                                lat = latitude)] [public==1,]
poverty_work = poverty_ori[,.(boro= as.factor(Boro),
                              povunit= Povunit,
                              income= NYCgov_Income,
                              CD=as.factor(CD))]

##Variable coding :
##URL: http://www1.nyc.gov/assets/opportunity/pdf/variable-descriptions.pdf

levels(poverty_work$CD) =c("Riverdale/Kingsbridge",
                           "Williamsbridge/Baychester",
                           "Throgs Neck/Co-op City",
                           "Pelham Parkway",
                           "Morrisania/East Tremont",
                           "Kingsbridge Heights/Moshulu",
                           "University Heights/Fordham",
                           "Highbridge/South Concourse",
                           "Soundview/Parkchester",
                           "Mott Haven/Hunts Point",
                           "Williamsburg/Greenpoint",
                           "Bushwick",
                           "Bedford Stuyvesant",
                           "Brooklyn Heights/Fort Greene",
                           "Park Slope/Carroll Gardens",
                           "North Crown Heights/Prospect Heights",
                           "Brownsville/Ocean Hill",
                           "East New York/Starrett City",
                           "Flatlands/Canarsie",
                           "East Flatbush",
                           "South Crown Heights",
                           "Sunset Park",
                           "Bay Ridge",
                           "Borough Park",
                           "Flatbush",
                           "Sheepshead Bay/Gravesend",
                           "Bensonhurst",
                           "Coney Island",
```

```
                            "Washington Heights/Inwood",
                            "Morningside Heights/Hamilton Heights",
                            "Central Harlem",
                            "East Harlem",
                            "Upper East Side",
                            "Upper West Side",
                            "Chelsea/Clinton/Midtown",
                            "Stuyvesant Town/Turtle Bay",
                            "Lower East Side/Chinatown",
                            "Greenwich Village/Financial District",
                            "Astoria",
                            "Jackson Heights",
                            "Flushing/Whitestone",
                            "Bayside/Little Neck",
                            "Bellerose/Rosedale",
                            "Hillcrest/Fresh Meadows",
                            "Elmhurst/Corona",
                            "Forest Hills/Rego Park",
                            "Sunnyside/Woodside",
                            "Middle Village/Ridgewood",
                            "Kew Gardens/Woodhaven",
                            "Jamaica",
                            "Howard Beach/South Ozone Park",
                            "Rockaways",
                            "South Shore",
                            "Mid-Island",
                            "North Shore")
levels(poverty_work$boro) = c("Bronx",
                              "Brooklyn",
                              "Manhattan",
                              "Queens",
                              "Staten Island")

poverty_work = poverty_work[,.(num_points=.N),by=.(boro,CD)]
### identify Cd with two neighborhoods by using grep targeting " / "
double_index = grep("./.",poverty_work$CD)
##preserve data with no double neighbor in one_poverty
one_poverty = poverty_work[-double_index,]
#create temp poverty file with double neighbor names in CD
tmp_poverty = poverty_work[double_index,]
#splitting names of double CD
double_cd = unlist(strsplit(as.character(tmp_poverty$CD), split = "/"))
#counts are euqally divided into two CD to fit the nyc_neighborhood object
double_num_points= rep(tmp_poverty$num_points, each=2)/2
double_boro = rep(tmp_poverty$boro, each=2)
double_poverty= data.table(boro=double_boro,
                           CD=double_cd,
                           num_points=double_num_points)

complete_poverty = rbind(one_poverty,double_poverty)
#getting longtitude and latitude using geocode

coord_complete_poverty = lapply(paste(complete_poverty$CD, complete_poverty$boro, sep=","),
```

```
                                function(x) geocode(as.character(x)))
##reformatting list of lists into one data frame
v_coord_complete_poverty = do.call(rbind.data.frame,coord_complete_poverty)

poverty_work = cbind(complete_poverty,v_coord_complete_poverty)

colnames(poverty_work)[4]="long"

###exclude NA columns
poverty_work = poverty_work[CD!="Gravesend"]

#Since it takes around 5 minutes to get lat/lom using geocode,
#we save the organized Rdata for convenience
save(poverty_work, file="./Data/poverty_work.RData")
#NYC map
nyc_map <- get_map(location = c(lon = -74.00, lat = 40.71), maptype = "terrain",zoom=11) ##terrain-lines
#ggmap(nyc_map)

r <- GET('https://goo.gl/gVuJrV')
nyc_neighborhoods <- readOGR(content(r,'text'), 'OGRGeoJSON', verbose = F)
nyc_neighborhoods_df <- tidy(nyc_neighborhoods)

############################
###Scatter plot in ggmap
############################

subway_plot= ggmap(nyc_map) +
  geom_polygon(data=nyc_neighborhoods_df, aes(x=long, y=lat, group=group),
               size=1, color="dodgerblue3", fill=NA) +
  geom_point(data=subway_work, aes(x=long, y=lat), shape=18,
             color="deeppink2", size=.5)

wifi_plot = ggmap(nyc_map) +
  geom_polygon(data=nyc_neighborhoods_df, aes(x=long, y=lat, group=group),
               size=1, color="dodgerblue3", fill=NA) +
  geom_point(data=wifi_work, aes(x=long, y=lat), shape=18,
             color="deeppink2", size=.5)

facility_plot = ggmap(nyc_map) +
  geom_polygon(data=nyc_neighborhoods_df, aes(x=long, y=lat, group=group),
               size=1, color="dodgerblue3", fill=NA) +
  geom_point(data=facility_work, aes(x=long, y=lat), shape=18,
             color="deeppink2",size=.5)
##spatial matching function
f.match.spatial = function(x , mapping =nyc_neighborhoods){
  x_spdf = x
  coordinates(x_spdf) = with(x_spdf,~ long + lat)
  proj4string(x_spdf) = proj4string(mapping)
  x_matches = over(x_spdf, mapping)
  return(cbind(x, x_matches))
}

#Matching df with coordinates with f.match.spatial
```

```
subway_work = f.match.spatial(subway_work)
wifi_work = f.match.spatial(wifi_work)
facility_work = f.match.spatial(facility_work)

##function for calculating number of observations of each neighbor
f.count.obs = function(x){
  points_by_neighborhood = x %>%
    group_by(neighborhood) %>%
    summarize(num_points=n())
  return(points_by_neighborhood)
}

##function for transforming count.obs to ggplot2 accesible object.
f.plot.data = function(count.obs){
  #count.obs = obj to be transformed into plot data
  plot_data = tidy(nyc_neighborhoods, region="neighborhood") %>%
    left_join(., count.obs, by=c("id"="neighborhood")) %>%
    filter(!is.na(num_points))
  return(plot_data)
}

###create plot_data
subway_plot_data =f.plot.data(f.count.obs(subway_work))
wifi_plot_data = f.plot.data(f.count.obs(wifi_work))
facility_plot_data = f.plot.data(f.count.obs(facility_work))

##function scaling numbers
f.scaling = function(x){
  neighborhood = x$neighborhood
  num_point = as.vector(scale(x$num_points))
  return(tibble(neighborhood=neighborhood, num_points=num_point))
}

###Overall plot_data
all.neighbor = tibble(neighborhood = unique(nyc_neighborhoods$neighborhood))
prosp = cbind(merge(all.neighbor, f.scaling(f.count.obs(subway_work)),all.x = T)[,2],
  merge(all.neighbor, f.scaling(f.count.obs(wifi_work)),all.x = T)[,2],
  merge(all.neighbor, f.scaling(f.count.obs(facility_work)),all.x = T)[,2]) %>%
  rowSums(.,na.rm=T) %>%
  cbind(all.neighbor,num_points=.)

prosp_plot_data = f.plot.data(prosp)

#####
# Plots for each dataset
#####


### adjust weighting on the wifi, might have negative correlation instead of positive
### correlation! i.e. as in north of central park.

#subway distribution
subway_spplot =
```

```
  ggmap(nyc_map) +
  geom_polygon(data=subway_plot_data,
               aes(x=long, y=lat, group=group, fill=num_points), alpha=0.75) +
  scale_fill_gradientn(colours = rev(terrain.colors(50)) ,
                       name= "Number of Subway") +
  ggtitle("Accesible subway") +
  theme(legend.position="top")

#wifi distribution
wifi_spplot =
  ggmap(nyc_map) +
  geom_polygon(data=wifi_plot_data,
               aes(x=long, y=lat, group=group, fill=num_points), alpha=0.75) +
  scale_fill_gradientn(colours = rev(terrain.colors(50)) ,
                       name= "Number of public wifi") +
  ggtitle("Public wifi")+
  theme(legend.position="top")

#facility distribution
facility_spplot =
  ggmap(nyc_map) +
  geom_polygon(data=facility_plot_data,
               aes(x=long, y=lat, group=group, fill=num_points), alpha=0.75) +
  scale_fill_gradientn(colours = rev(terrain.colors(50)) ,
                       name= "Number of facilities") +
  ggtitle("Public facilities")+
  theme(legend.position="top")


# Public access level
publicaccess_spplot =
  ggmap(nyc_map) +
  geom_polygon(data=prosp_plot_data,
               aes(x=long, y=lat, group=group, fill=num_points), alpha=0.75) +
  scale_fill_gradient(low="white",high="navyblue",
                       name= "Public access index") +
  ggtitle("Public access index")+
  theme(legend.position="top")
##Arrange output ,Figure 1
gs <- list(subway_spplot,wifi_spplot,facility_spplot)
lay <- rbind(c(1,2,3))
grid.arrange(grobs = gs, layout_matrix = lay,
             top = textGrob("Distribution of each public resources of NYC",
                            x = 0, # starts far left
                            y = 0.5, # experiment with
                                     # vertical placement
                            just = "left", # left-aligned,
                            gp = gpar(fontsize = 14) # bigger font
             ),
             bottom = textGrob("Figure 1: Distribution of each public resources of NYC\n We agggregated
                           x = 0.1,
                           y = 0.5,
                           just = "left"))
```

```
##reload pre-saved data
load("./Data/poverty_work.RData")
poverty_plot_data =f.plot.data(f.match.spatial(poverty_work))

# Poverty level
poverty_spplot =
  ggmap(nyc_map) +
  geom_polygon(data=poverty_plot_data,
               aes(x=long.x, y=lat.x, group=group, fill=num_points), alpha=0.75) +
  scale_fill_gradient(low="white",high="red",
                      name= "Poverty level") +
  ggtitle("Poverty level")+
  theme(legend.position="top")

gs2 <- list(publicaccess_spplot,poverty_spplot)
lay2 <- rbind(c(1,2))
grid.arrange(grobs = gs2, layout_matrix = lay2,
             top = textGrob("Comparison between poverty level and public access index",
                            x = 0, # starts far left
                            y = 0.5, # experiment with
                                     # vertical placement
                            just = "left", # left-aligned,
                            gp = gpar(fontsize = 14) # bigger font
             ),
             bottom = textGrob("Figure 2: Comparison between poverty level and public access index\n As
                           x = 0.1,
                           y = 0.5,
                           just = "left"))
pcindex = prosp

povindex =
  f.match.spatial(poverty_work) %>%
  group_by(neighborhood) %>%
  summarise(num_points=sum(num_points))%>%
  merge(x=all.neighbor, y =. , all.x=T)

##fill 0 with NA values to calculate correlation
povindex[is.na(povindex)] =0

#cor(pcindex[,2],povindex[,2])
cor.test(pcindex[,2],povindex[,2])
```