# Exc2B - Places around the world

*Ido Algom & Natalie Gilboa*

In this section we chose to explore the sites with rating with Google Maps API.

First, some library loads

```
library(RCurl)
library(jsonlite)
library(plyr)
library(ggplot2)
library(ggmap)
```

API key

```
my_api <- "AIzaSyC2YlDWnRRyDalApkny42181XQPmUhuTAU"
#my_api <- "AIzaSyApghFW9gHFboZlstE4opcBwmVpmC_odjc"
```

This function will compuse a URL based on a free text city name string coming as inpur parameter

```
mapsURL <- function(address, return.call = "json", sensor = "false") {
  root <- "http://maps.google.com/maps/api/geocode/"
  u <- paste(root, return.call, "?address=", address, "&sensor=", sensor, sep = "")
  return(URLencode(u))
}
```

The next function will access Google Maps API to retrive the city center geo code

```
geoCode <- function(address) {
  u <- mapsURL(address)
  doc <- getURL(u)
  x <- fromJSON(doc,simplifyVector = FALSE)
  if(x$status=="OK") {
    lat <- x$results[[1]]$geometry$location$lat
    lng <- x$results[[1]]$geometry$location$lng
    formatted_address <- x$results[[1]]$formatted_address
    return(c(lat, lng, formatted_address))
  } else {
    return(c(NA,NA, NA))
  }
}
```

getLocations will retrive all the resturants (200 is max Google provides) and return a JSON file with all data

```
getLocations <- function(type,geo,keyword ="",
                         key=my_api,radius = 2500) {
  root="https://maps.googleapis.com/maps/api/place/radarsearch/json?"
  if (keyword == "")
    keyword0 <- ""
  else
    keyword0 <- paste0("&keyword=",keyword)
  u <- paste0(root,"location=",geo[1],",",geo[2],"&radius=",radius,"&type=",type,ke
yword0,"&key=",key)
  jsonLink <- URLencode(u)
  jsonFile <- getURL(jsonLink)
  jsonTree <- fromJSON(jsonFile, simplifyVector = FALSE)
  if (jsonTree$status == "OK") {
    return(jsonTree)
  }
  else
    msg = paste("Error:",jsonTree$error_message)
    stop(msg)
}
```

parseLocations and analyzeLocations will create a table with all the nesscery information from the JSON file

```
parseLocations <- function(tree) {
  vec = c()
  for (i in 1:length(tree$results)) {
    vec[i] <-tree$results[[i]]$place_id
  }
  res <- analyzeLocations(vec)
  show(paste("Total results:",length(vec)))
  return(res)
}


analyzeLocations <- function(vec) {
  lat = c()
  lng = c()
  rating = c()
  base="https://maps.googleapis.com/maps/api/place/details/json?placeid="
  key <- paste0("&key=",my_api)
  j <- 1
  for (i in 1:length(vec)) {
    u <- paste0(base,vec[i],key)
    jsonLink <- URLencode(u)
    #jsonFile <- getURL(jsonLink,crlf = TRUE)
    jsonTree <- fromJSON(jsonLink, simplifyVector = FALSE)
    if (jsonTree$status == "OK") {
      if (!is.null(jsonTree$result$rating)) {
        rating[j] <-jsonTree$result$rating
        lat[j] <- jsonTree$result$geometry$location$lat
        lng[j] <- jsonTree$result$geometry$location$lng
        j <- j+1
      }
    }
    else
      next
  }
  locationsTable = data.frame(rating = rating, lat = lat, lon = lng)
}
```

showLocationsOnMap will draw the map using Google Maps API

```
showLocationsOnMap <- function(geo,locationsInfo) {
  map <- get_map(location = c( lon =as.numeric(geo[2]), lat = as.numeric(geo[1])),
maptype = "satellite", zoom = 14)
  ggmap(map) + geom_point(data = locationsInfo, aes(x=lon, y=lat, color=as.numeric(
rating))) + scale_colour_gradientn(colours=rainbow(4))
}
```

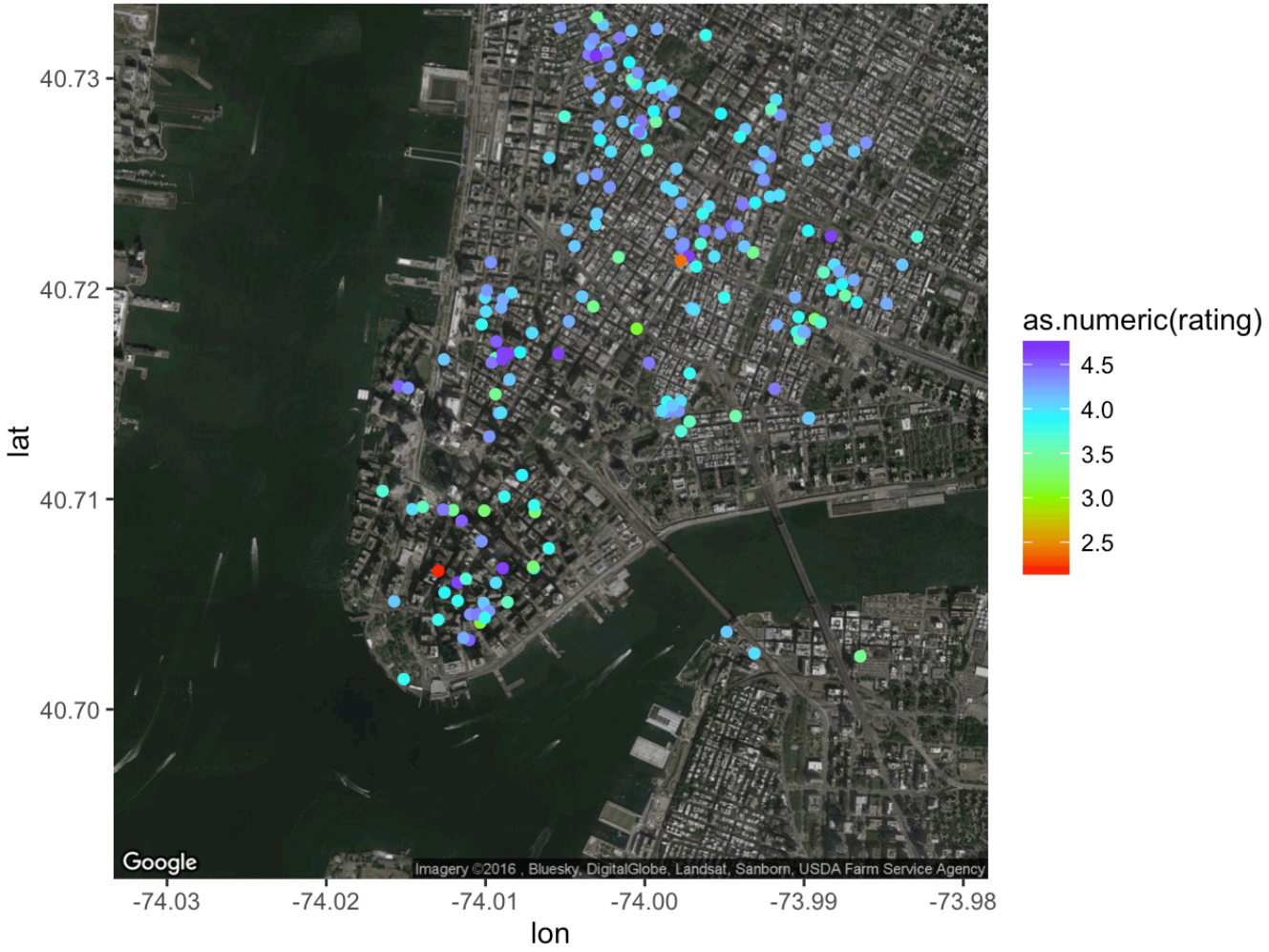
This function is the main function

```
lookFor <- function(type, location) {
  geo <- geoCode(location)
  locations <- getLocations(type,geo)
  locationsInfo <- parseLocations(locations)
  showLocationsOnMap(geo,locationsInfo)
}
```

Those are simples for using it

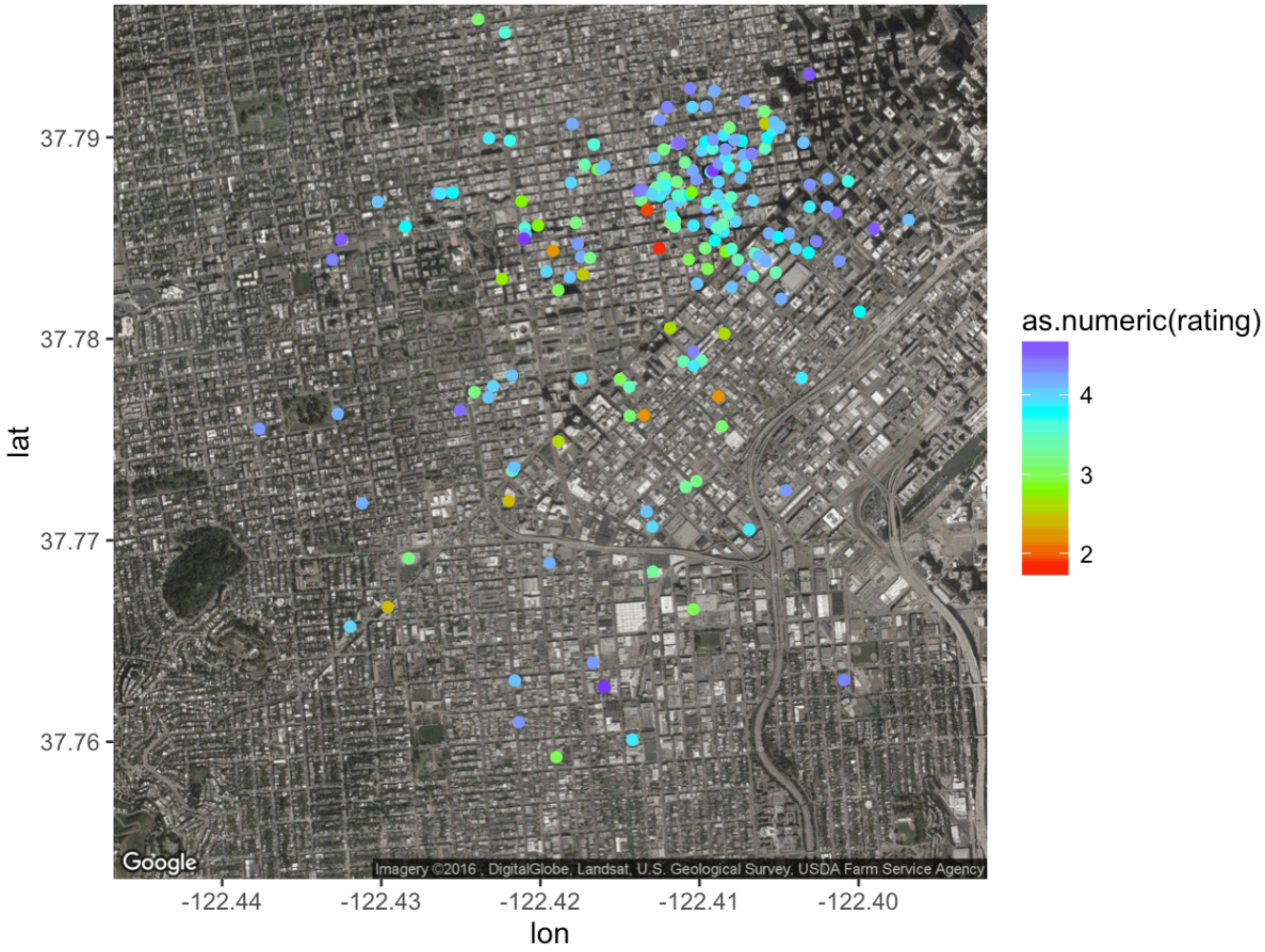
```
lookFor("restaurant","new york city")
```

```
## [1] "Total results: 200"
```




```
lookFor("hotel","san francisco")
```

```
## [1] "Total results: 200"
```


37.79
37.78
37.77
37.76
lat
Google
Imagery ©2016 , DigitalGlobe, Landsat, U.S. Geological Survey, USDA Farm Service Agency
-122.44
-122.43
-122.42
-122.41
-122.40
lon
as.numeric(rating)
4
3
2