

lab7

chaithra

14/10/2021

```
## Intro to Data Science - Lab 7

# IST687 Section M002

# Professor Anderson

# Enter your name here: Chaithra Kopparam Cheluvaiyah

#Select one of the below and add needed information

# 2. I did this homework with help from the book and the professor and
# these Internet sources:
#https://www.rdocumentation.org/packages/maps/versions/3.4.0
#https://www.dominodatalab.com/data-science-dictionary/ggmap
#https://www.rdocumentation.org/packages/mapproj/versions/1.2.7/topics/mapproject
#https://stackoverflow.com/questions/37912418/how-does-geom-map-map-id-function-work
```

```
# package required for geographical maps
#install.packages("maps")

# package required to retrieve raster maps from Stamen Maps
#install.packages("ggmap")

# required for converting lat and long into projected coordinates
#install.packages("mapproj")

# importing all the installed packages
library(ggplot2);
library(maps);
library(ggmap);
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(mapproj)
```

```
# converting US states build-in data from the maps package in to a data frame
us <- map_data("state")
```

```

# converting region names to lowercase
us$state_name <- tolower(us$region)

# defining the column that hold the identifier for other layers
map <- ggplot(us, aes(map_id= state_name))

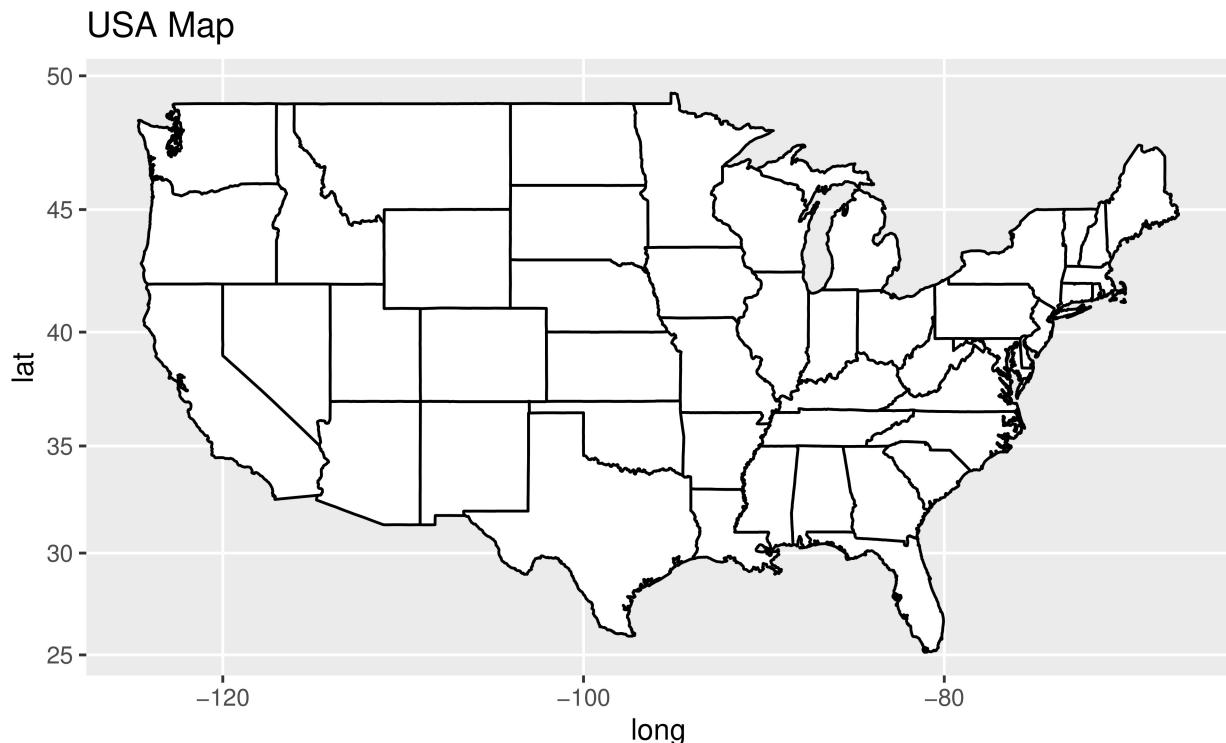
# plot the map using lines instead of dots and fill the polygon with white
# and outline with black colors
map <- map + aes(x=long, y=lat, group=group) +
  geom_polygon(fill = "white", color = "black")

# expanding plot limits based on data
map <- map + expand_limits(x=us$long, y=us$lat)

# project states onto flat 2D plane using any of the projection defined
# by mpproj package
# and adding title to the plot
map <- map + coord_map() + ggtitle("USA Map")

#invoking map
map

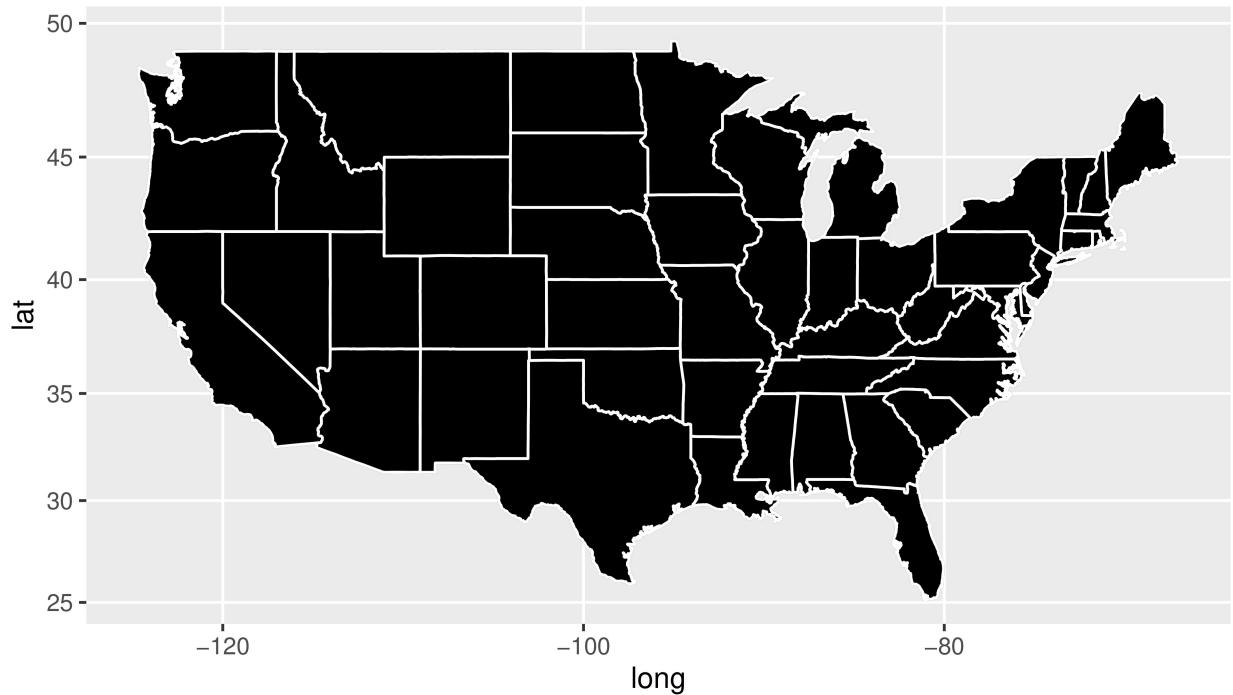
```



1. Add a comment for each line of code, explaining what that line of code does.
2. The map you just created fills in the area of each state in white while outlining it with a thin black line. Use the `fill=` and `color=` commands inside the call to `geom_polygon()` to reverse the color scheme.

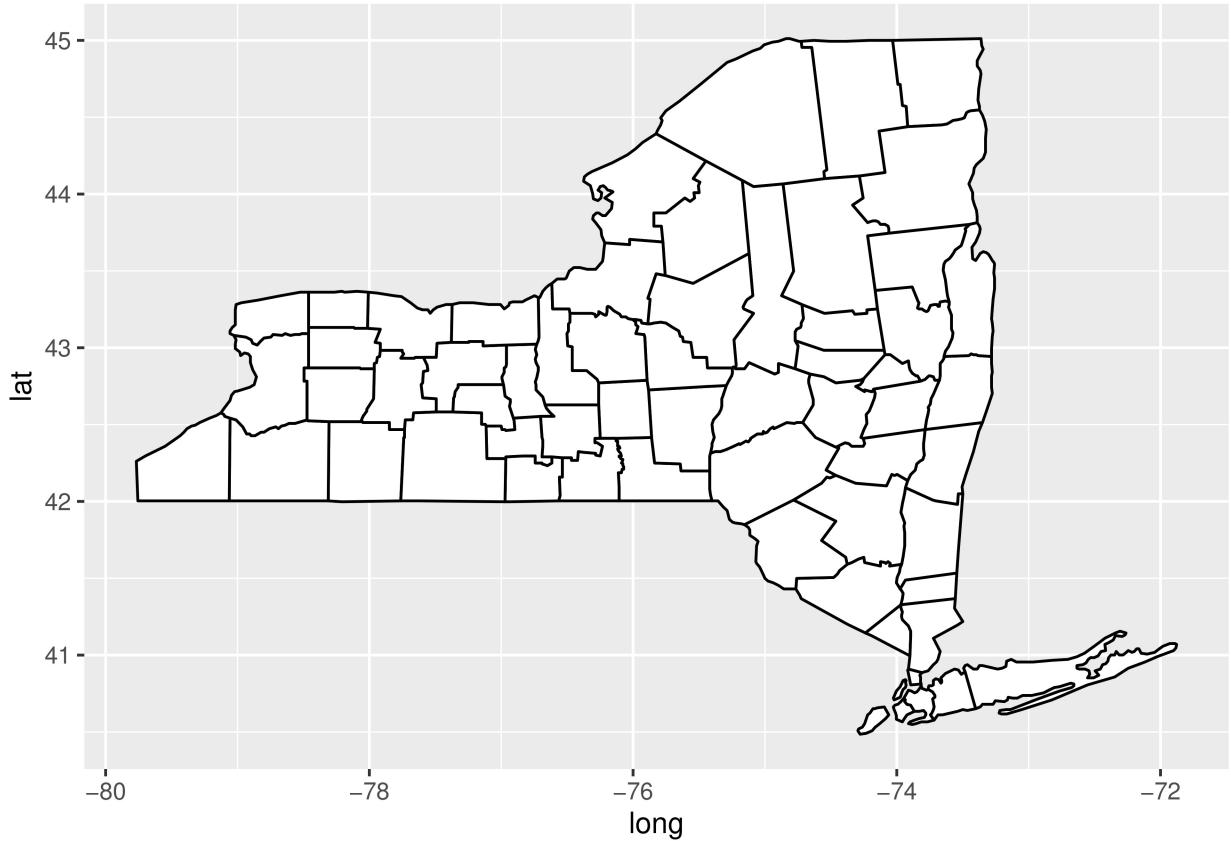
```
# reversing the color scheme  
map <- map + geom_polygon(fill = "black", color = "white")  
map
```

USA Map



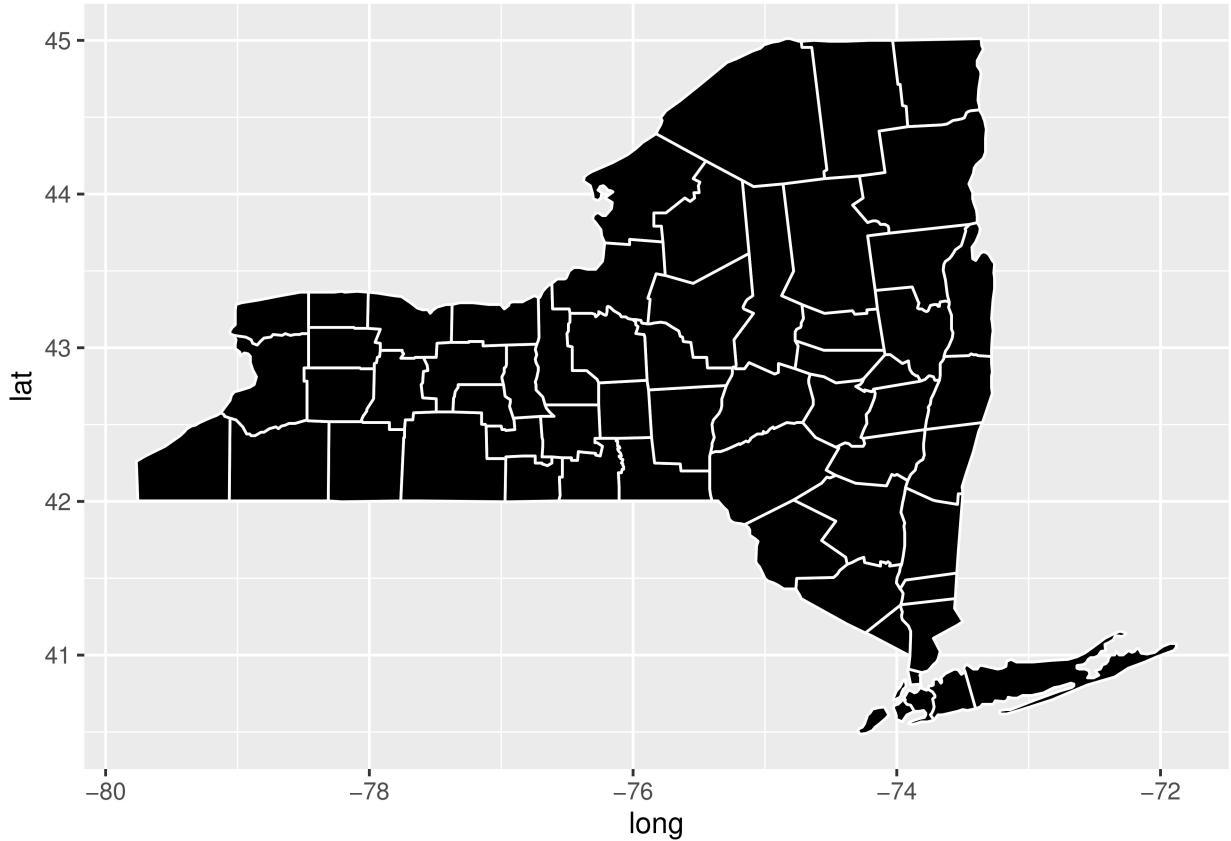
Now run the following code:

```
# zooming on New York state counties  
ny_counties <- map_data("county", "new york")  
  
# plotting the counties  
ggplot(ny_counties) + aes(long, lat, group=group) +  
  geom_polygon(fill = "white", color = "black")
```



- Just as in step 2, the map you just created fills in the area of each county in black while outlining it with a thin white lines. Use the `fill=` and `color=` commands inside the call to `geom_polygon()` to reverse the color scheme.

```
# reversing the color scheme
ggplot(ny_counties) + aes(long,lat, group=group) +
  geom_polygon(fill = "black", color = "white")
```



4. Run head(ny_counties) to verify how the county outline data looks.

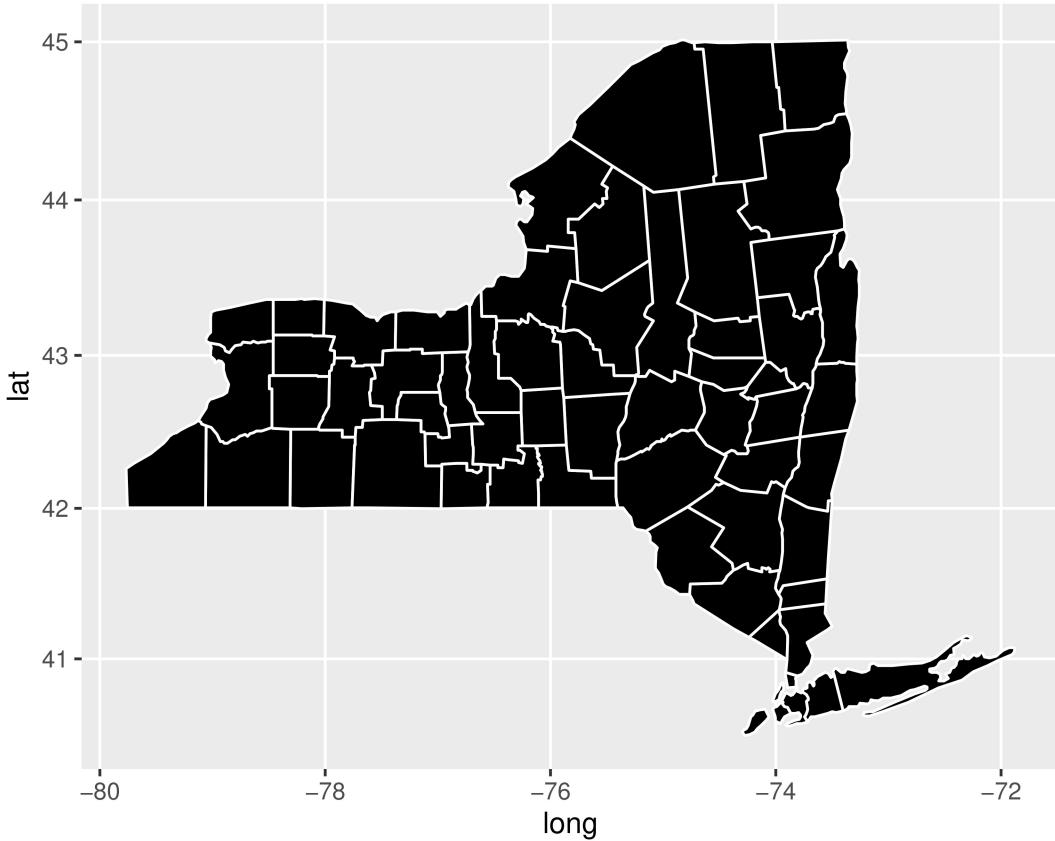
```
head(ny_counties) # exploring first few lines of NY counties
```

```
##      long      lat group order   region subregion
## 1 -73.78550 42.46763     1     1 new york    albany
## 2 -74.25533 42.41034     1     2 new york    albany
## 3 -74.25533 42.41034     1     3 new york    albany
## 4 -74.27252 42.41607     1     4 new york    albany
## 5 -74.24960 42.46763     1     5 new york    albany
## 6 -74.22668 42.50774     1     6 new york    albany
```

5. Make a copy of your code from step 3 and add the following subcommand to your ggplot() call (don't forget to put a plus sign after the geom_polygon() statement to tell R that you are continuing to build the command): coord_map(projection = "mercator") In what way is the map different from the previous map. Be prepared to explain what a Mercator projection is.

```
# plot(ny_counties$long, ny_counties$lat) # map created with lat and long points

# Mercator projection makes earth's surface which is spherical into a 2D plane
# to make a map
ggplot(ny_counties) + aes(long,lat, group=group) +
  geom_polygon(fill = "black", color = "white") +
  coord_map(projection="mercator")
```



6. Grab a copy of the nyData.csv data set from: <https://intro-datasience.s3.us-east-2.amazonaws.com/nyData.csv>

Read that data set into R with `read_csv()`. This will require you have installed and libraries the `readr` package. The next step assumes that you have named the resulting data frame “nyData.”

```
library(readr) # loading library

# reading data from URL
nyData <- read_csv("https://intro-datasience.s3.us-east-2.amazonaws.com/nyData.csv")

## Rows: 62 Columns: 5

## -- Column specification -----
## Delimiter: ","
## chr (1): county

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

head(nyData)
```

```

## # A tibble: 6 x 5
##   county      pop2010 pop2000 sqMiles popDen
##   <chr>       <dbl>    <dbl>    <dbl>    <dbl>
## 1 albany     304204  294565    523.    582.
## 2 allegany    48946   49927   1029.    47.6
## 3 bronx      1385108 1332650    42.1  32900.
## 4 broome     200600   200536    706.    284.
## 5 cattaraugus 80317   83955   1308.    61.4
## 6 cayuga      80026   81963    692.    116.

```

7. Next, merge your ny_counties data from the first set of questions with your new nyData data frame, with this code:

```

# joining data frames based on county names
mergeNY <- merge(ny_counties,nyData, all.x=TRUE, by.x="subregion",by.y="county")

```

8. Run head(mergeNY) to verify how the merged data looks.

```
head(mergeNY)
```

```

##   subregion      long      lat group order   region pop2010 pop2000 sqMiles
## 1 albany -73.78550 42.46763     1     1 new york  304204  294565  522.8
## 2 albany -74.25533 42.41034     1     2 new york  304204  294565  522.8
## 3 albany -74.25533 42.41034     1     3 new york  304204  294565  522.8
## 4 albany -74.27252 42.41607     1     4 new york  304204  294565  522.8
## 5 albany -74.24960 42.46763     1     5 new york  304204  294565  522.8
## 6 albany -74.22668 42.50774     1     6 new york  304204  294565  522.8
##   popDen
## 1 581.87
## 2 581.87
## 3 581.87
## 4 581.87
## 5 581.87
## 6 581.87

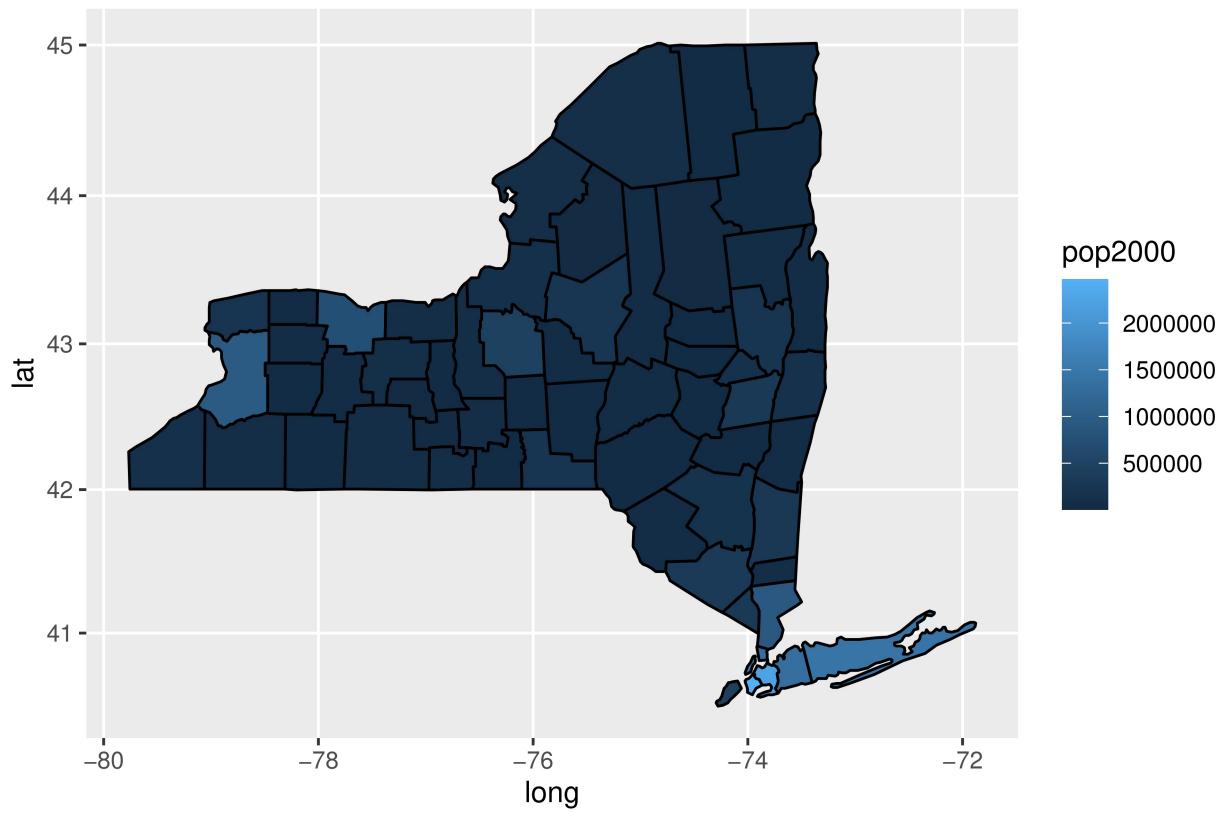
```

9. Now drive the fill color inside each county by adding the fill aesthetic inside of your geom_polygon() subcommand (fill based on the pop2000).

```

# plotting the merged data with color shade showing population of counties
# in 2000
ggplot(mergeNY) +
  geom_polygon(color = "black", aes(long,lat, group=group, fill=pop2000)) +
  coord_map()

```



10. Extra (not required):

- a. Read in the following JSON datasets: 'https://gbfs.citibikenyc.com/gbfs/en/station_information.json' 'https://gbfs.citibikenyc.com/gbfs/en/station_status.json'

```
# loading the RCurl package which was installed in the previous step
# used for accessing internet data
library(RCurl)

# loading the jsonlite package which was installed in the previous step
# required for decoding json
library(jsonlite)

# getting json station information and parsing to R object
stationInfoDF <- fromJSON(getURL('https://gbfs.citibikenyc.com/gbfs/en/station_information.json'))
stationInfoDF <- stationInfoDF$data$stations
head(stationInfoDF)

##   station_id region_id rental_methods eightd_station_services
## 1          72        71 CREDITCARD, KEY                  NULL
## 2          79        71 CREDITCARD, KEY                  NULL
## 3          82        71 CREDITCARD, KEY                  NULL
## 4          83        71 CREDITCARD, KEY                  NULL
## 5         116        71 CREDITCARD, KEY                  NULL
## 6         119        71 CREDITCARD, KEY                  NULL
```

```

##   electric_bike_surcharge_waiver capacity eightd_has_key_dispenser
## 1                         FALSE      55                         FALSE
## 2                         FALSE      33                         FALSE
## 3                         FALSE      27                         FALSE
## 4                         FALSE      62                         FALSE
## 5                         FALSE      50                         FALSE
## 6                         FALSE      53                         FALSE
##                               external_id      lon station_type      lat
## 1 66db237e-0aca-11e7-82f6-3863bb44ef7c -73.99393    classic 40.76727
## 2 66db269c-0aca-11e7-82f6-3863bb44ef7c -74.00667    classic 40.71912
## 3 66db277a-0aca-11e7-82f6-3863bb44ef7c -74.00017    classic 40.71117
## 4 66db281e-0aca-11e7-82f6-3863bb44ef7c -73.97632    classic 40.68383
## 5 66db28b5-0aca-11e7-82f6-3863bb44ef7c -74.00150    classic 40.74178
## 6 66db2953-0aca-11e7-82f6-3863bb44ef7c -73.97803    classic 40.69609
##                               rental_uris.android      rental_uris.ios
## 1 https://bkn.lft.to/lastmile_qr_scan https://bkn.lft.to/lastmile_qr_scan
## 2 https://bkn.lft.to/lastmile_qr_scan https://bkn.lft.to/lastmile_qr_scan
## 3 https://bkn.lft.to/lastmile_qr_scan https://bkn.lft.to/lastmile_qr_scan
## 4 https://bkn.lft.to/lastmile_qr_scan https://bkn.lft.to/lastmile_qr_scan
## 5 https://bkn.lft.to/lastmile_qr_scan https://bkn.lft.to/lastmile_qr_scan
## 6 https://bkn.lft.to/lastmile_qr_scan https://bkn.lft.to/lastmile_qr_scan
##   short_name legacy_id                           name has_kiosk
## 1 6926.01          72      W 52 St & 11 Ave      TRUE
## 2 5430.08          79      Franklin St & W Broadway      TRUE
## 3 5167.06          82      St James Pl & Pearl St      TRUE
## 4 4354.07          83      Atlantic Ave & Fort Greene Pl      TRUE
## 5 6148.02          116     W 17 St & 8 Ave      TRUE
## 6 4700.06          119     Park Ave & St Edwards St      TRUE

# getting json station status and parsing to R object
stationStatusDF <- fromJSON(getURL('https://gbfs.citibikenyc.com/gbfs/en/station_status.json'))
stationStatusDF <- stationStatusDF$data$stations
head(stationStatusDF)

##   num_docks_disabled station_id eightd_has_available_keys is_installed
## 1                  0        72                         FALSE      1
## 2                  0        79                         FALSE      1
## 3                  0        82                         FALSE      1
## 4                  0        83                         FALSE      1
## 5                  0       116                         FALSE      1
## 6                  0       119                         FALSE      1
##   num_bikes_disabled last_reported is_returning num_bikes_available
## 1                  2 1634256733            1                 13
## 2                  2 1634256223            1                 27
## 3                  0 1634256436            1                 21
## 4                  5 1634256608            1                 36
## 5                  0 1634256684            1                 25
## 6                  2 1634250938            1                 45
##   num_docks_available is_renting legacy_id station_status num_ebikes_available
## 1                  40           1        72      active             0
## 2                   4           1        79      active             0
## 3                   6           1        82      active             0
## 4                  21           1        83      active             0
## 5                  25           1       116      active             0

```

```

## 6          6      1     119      active      0
##   eightd_active_station_services valet.off_dock_capacity valet.station_id
## 1             NULL        NA    <NA>
## 2             NULL        NA    <NA>
## 3             NULL        NA    <NA>
## 4             NULL        NA    <NA>
## 5             NULL        NA    <NA>
## 6             NULL        NA    <NA>
##   valet.region valet.valet_revision valet.off_dock_count valet.active
## 1      <NA>        NA        NA      NA
## 2      <NA>        NA        NA      NA
## 3      <NA>        NA        NA      NA
## 4      <NA>        NA        NA      NA
## 5      <NA>        NA        NA      NA
## 6      <NA>        NA        NA      NA
##   valet.dock_blocked_count
## 1           NA
## 2           NA
## 3           NA
## 4           NA
## 5           NA
## 6           NA

```

- b. Merge the datasets, based on ‘station_id’

```

staion_bikes <- merge(stationInfoDF,stationStatusDF, all.x=TRUE,
                      by.x="station_id",by.y="station_id")

```

- c. Clean the merged dataset to only include useful information For this work, you only need lat, lon and the number of bikes available

```

cols <- c('lat', 'lon', 'num_bikes_available')
staion_bikes <- staion_bikes[,cols]
str(staion_bikes)

```

```

## 'data.frame': 1584 obs. of 3 variables:
## $ lat : num 40.7 40.7 40.7 40.7 40.7 ...
## $ lon : num -74 -74 -74 -74 -74 ...
## $ num_bikes_available: int 25 45 7 24 54 23 55 51 9 18 ...

```

- d. Create a stamen map using ‘get_stamenmap()’ Have the limits of the map be defined by the lat and lon of the stations

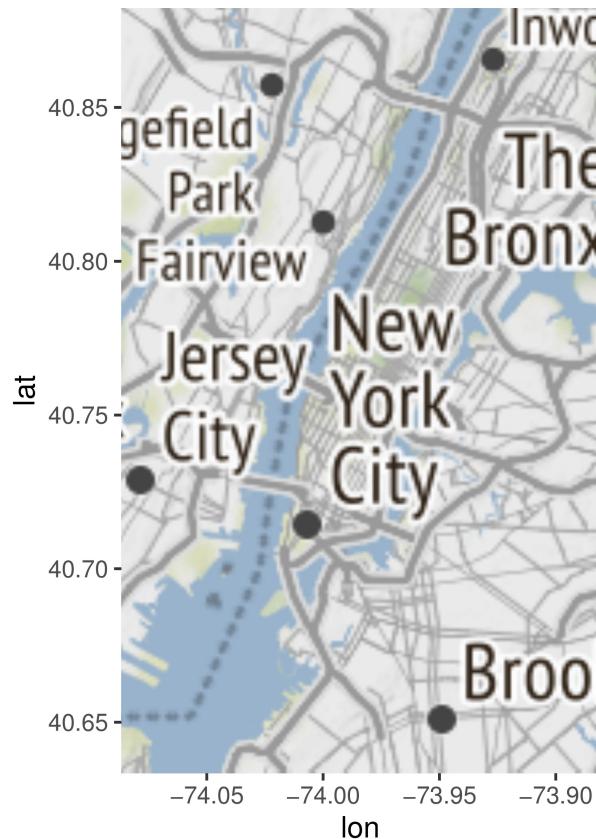
```

#defining limits of the map
bb <- c(left = min(staion_bikes$lon),
       bottom = min(staion_bikes$lat),
       right = max(staion_bikes$lon),
       top = max(staion_bikes$lat))

library(ggmap)
map <- get_stamenmap(bbox = bb)

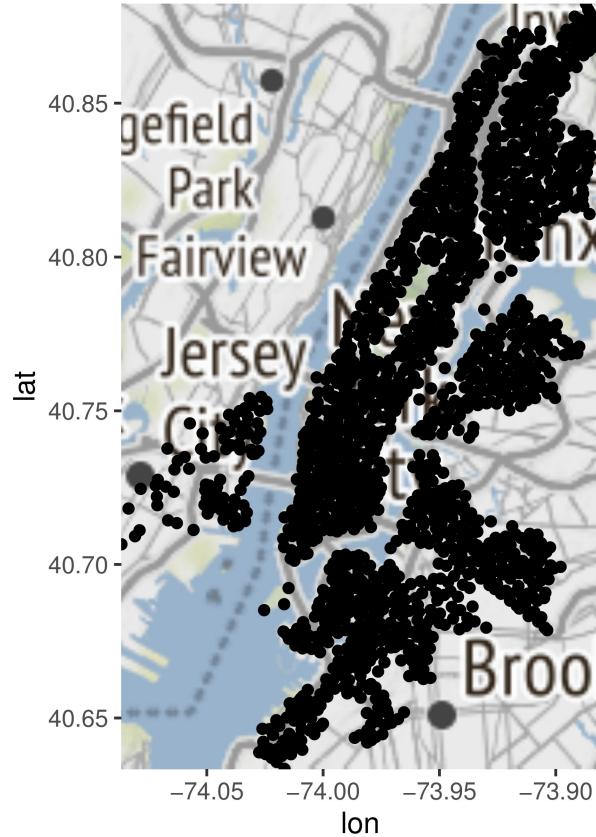
```

```
## Source : http://tile.stamen.com/terrain/10/301/384.png  
## Source : http://tile.stamen.com/terrain/10/301/385.png  
  
#invoking the map using ggmap  
ggmap(map)
```



- e. Show the stations, as points on the map.

```
ggmap(map) + geom_point(data= staion_bikes,aes(x=lon,y=lat))
```



f. Show the number of bikes available as a color

```
ggmap(map) + geom_point(data=staion_bikes,  
                         aes(x=lon,y=lat, color=num_bikes_available))
```

