Shaoqian_Chen_599Final

Shaoqian Chen 8831737894 Wed

4/25/2020

```r
library(dplyr)       # for data manipulation
```

```
##
## 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
```

```r
library(ggplot2)     # for data visualization
library(stringr)     # for string functionality
library(cluster)     # for general clustering algorithms
library(factoextra)  # for visualizing cluster results
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WB
a
```

```r
library(yardstick)
```

```
## For binary classification, the first factor level is assumed to be the event.
## Set the global option `yardstick.event_first` to `FALSE` to change this.
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:dplyr':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
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(dplyr)
library(readr)
```

```
##
## Attaching package: 'readr'
```

```
## The following object is masked from 'package:yardstick':
##
##     spec
```

```
ggmap::register_google(key = "AIzaSyDjdKw8npxWHoH5eL85_OrfxTUSyf8Wxcg")
```

Question 1

a)

```
mnist <- dslabs::read_mnist()
```

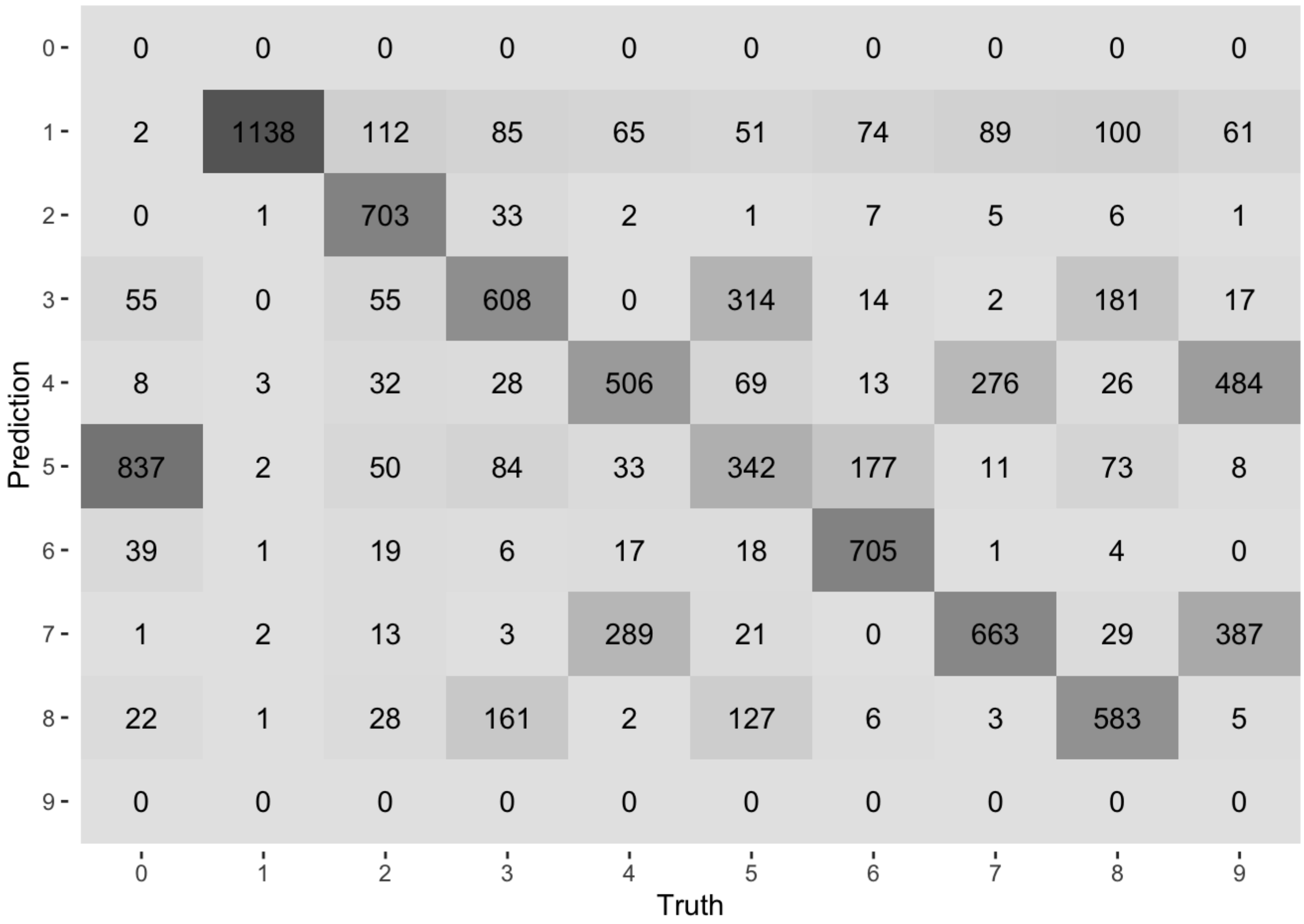
```
set.seed(1)
sample_1 = sample(60000,10000)
features <- mnist$train$images[sample_1,]

# Use k-means model with 10 centers and 10 random starts
mnist_clustering <- kmeans(features, centers = 10, nstart = 10)
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

```
# Print contents of the model output
str(mnist_clustering)
```

```
## List of 9
##  $ cluster     : int [1:10000] 7 7 6 9 1 10 5 9 5 10 ...
##  $ centers     : num [1:10, 1:784] 0 0 0 0 0 0 0 0 0 0 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:10] "1" "2" "3" "4" ...
##   .. ..$ : NULL
##  $ totss       : num 3.42e+10
##  $ withinss    : num [1:10] 1.02e+09 2.25e+09 2.42e+09 2.59e+09 3.70e+09 ...
##  $ tot.withinss: num 2.54e+10
##  $ betweenss   : num 8.78e+09
##  $ size        : int [1:10] 734 810 759 914 1246 1043 1408 703 938 1445
##  $ iter        : int 11
##  $ ifault      : int 2
##  - attr(*, "class")= chr "kmeans"
```

```
# Extract cluster centers
mnist_centers <- mnist_clustering$centers
# Create mode function
mode_fun <- function(x){ which.max(tabulate(x))
}
mnist_comparison <- data.frame(
  cluster = mnist_clustering$cluster,
  actual = mnist$train$labels[sample_1]
) %>%
  group_by(cluster) %>%
  mutate(mode = mode_fun(actual)) %>%
  ungroup() %>%
  mutate_all(factor, levels = 0:9)
# Create confusion matrix and plot results
yardstick::conf_mat(
  mnist_comparison,
  truth = actual,
  estimate = mode
) %>%
  autoplot(type = 'heatmap')
```

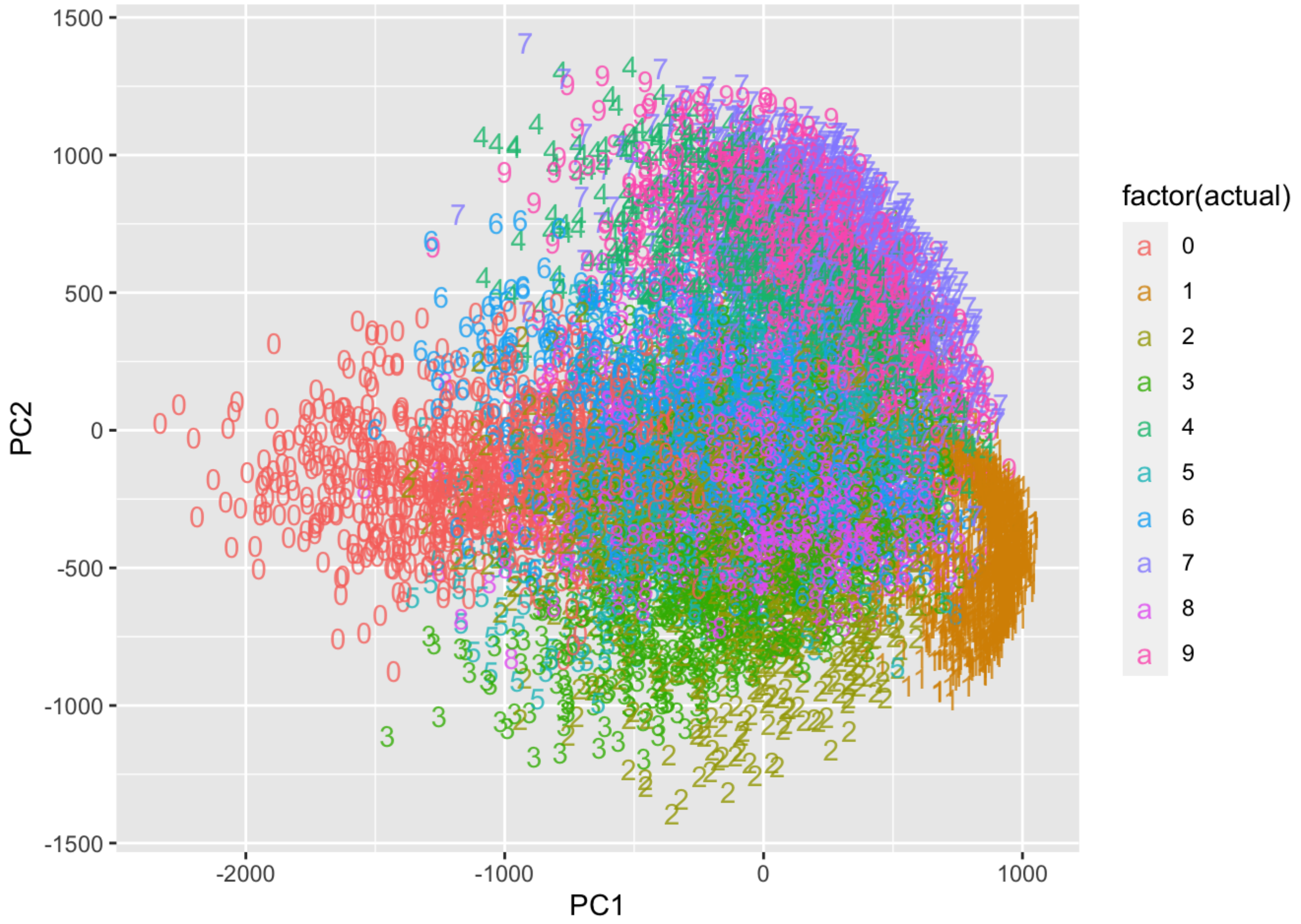


Prediction \ Truth	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	2	1138	112	85	65	51	74	89	100	61
2	0	1	703	33	2	1	7	5	6	1
3	55	0	55	608	0	314	14	2	181	17
4	8	3	32	28	506	69	13	276	26	484
5	837	2	50	84	33	342	177	11	73	8
6	39	1	19	6	17	18	705	1	4	0
7	1	2	13	3	289	21	0	663	29	387
8	22	1	28	161	2	127	6	3	583	5
9	0	0	0	0	0	0	0	0	0	0

b)

```
m1 = prcomp(features, scale = F)
```

```
newdata = m1$x
actual = mnist$train$labels[sample_1]
ggplot(data.frame(newdata),aes(x=PC1,y=PC2))+geom_text(aes(label=actual,color = factor(a
ctual)),alpha=0.8)
```



```
##
```

From the plot above we can identidy that digit 0 are well seperated and digit 1 are not

Questioin 2

a)

```
df = crime
#head(df)
```

```
dfselected <- df$offense=="robbery"|df$offense=="aggravated assault"|df$offense=="rape"|
df$offense=="murder"
df2 <- df[dfselected,]
```

```
geocode("Houston")
```

```
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=Houston&key=xxx
```

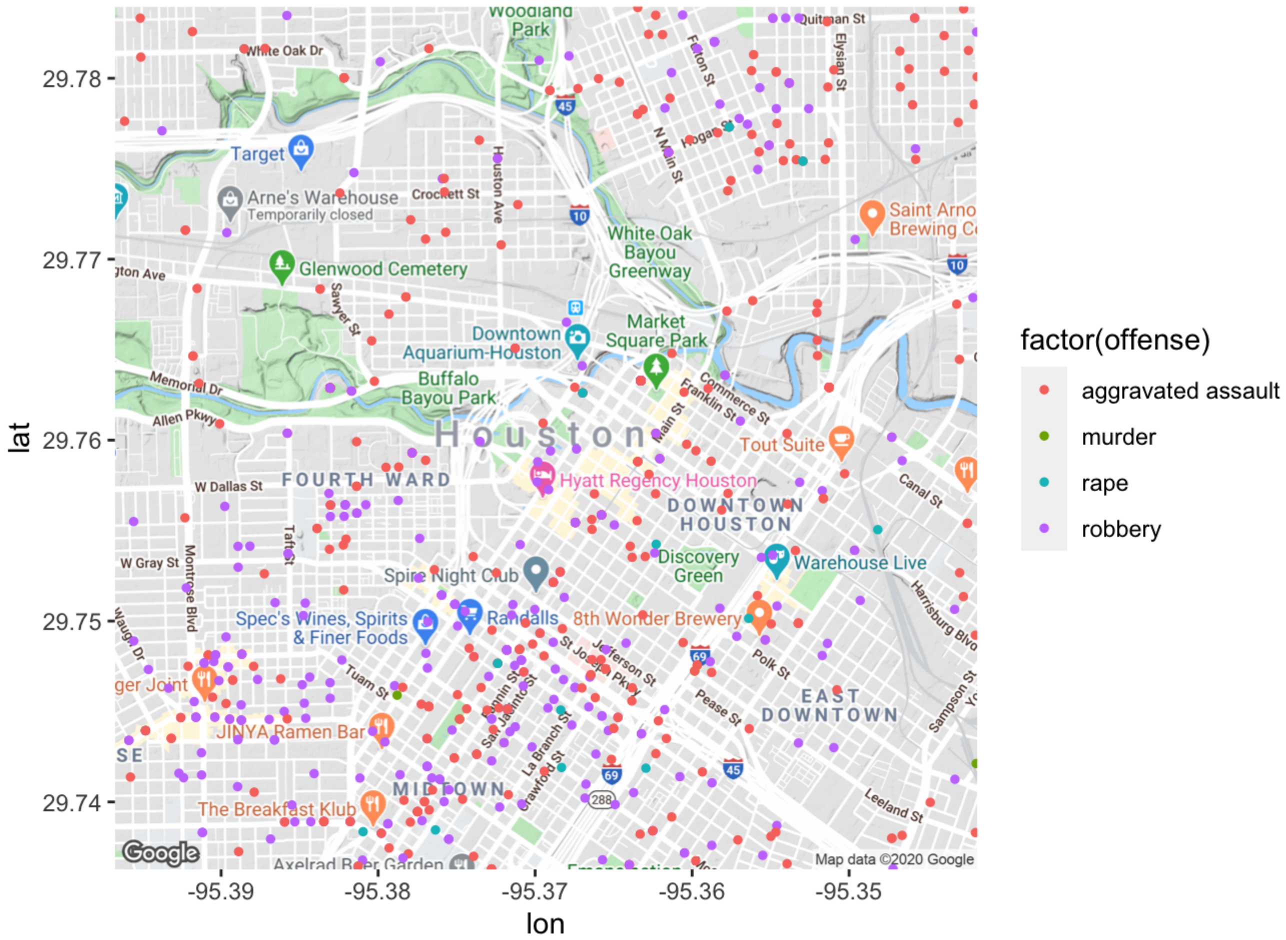
```
## # A tibble: 1 x 2
##     lon   lat
##   <dbl> <dbl>
## 1 -95.4  29.8
```

```
Houston.map = get_googlemap(center=c(lon = -95.369345,lat = 29.760155),lon=c(-95.39681,-
95.34188),lat = c(29.73631,29.78400),zoom = calc_zoom(c(-95.39681,-95.34188),c(29.73631,
29.78400)), scale = 2)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=29.760155,-95.369345&z
oom=14&size=640x640&scale=2&maptype=terrain&key=xxx
```

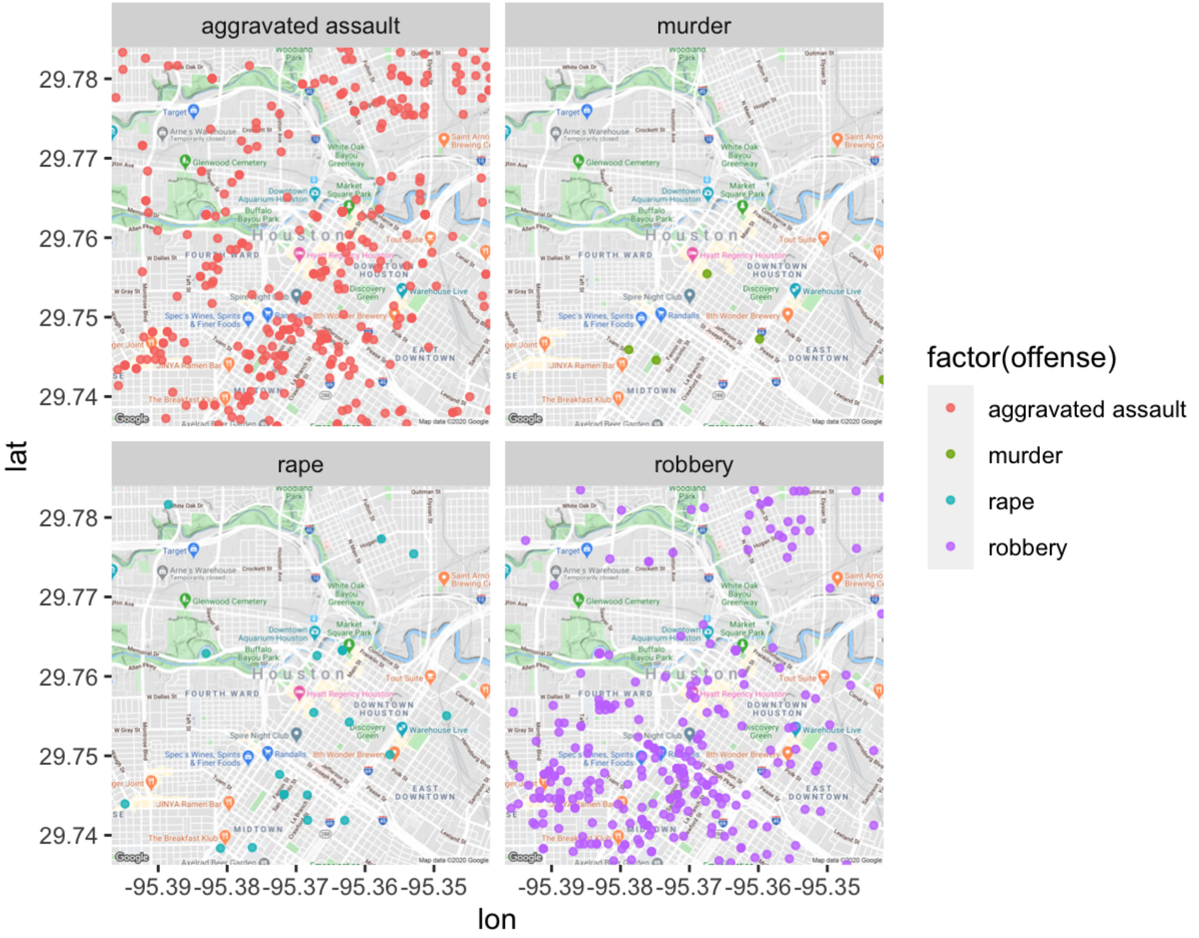
```
p = ggmap(Houston.map)
```

```
p+ geom_point(data = df2, aes(x=lon,y=lat,colour = factor(offense)),size = 1,alpha=1,na.
rm = TRUE)
```



b)

```
p+ geom_point(data = df2, aes(x=lon,y=lat,colour = factor(offense)),size = 1,alpha=0.8,n
a.rm = TRUE)+facet_wrap(~offense,nrow = 2)
```



Question 3

```
data = read_csv("question3.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   FIPS = col_double(),
##   County = col_character(),
##   population = col_double(),
##   cases = col_double(),
##   lat = col_double(),
##   lon = col_double()
## )
```

```
data$log <- log10(data$cases)
head(data)
```

```
## # A tibble: 6 x 8
##      X1  FIPS County             population cases   lat   lon   log
##   <dbl> <dbl> <chr>                   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1 10001 Kent County            180786   503  39.1 -75.5  2.70
## 2     2 10003 New Castle County      558753  1352  39.6 -75.6  3.13
## 3     3 10005 Sussex County          234225  1317  38.7 -75.3  3.12
## 4     4  1001 Autauga County          55869    32  32.5 -86.6  1.51
## 5     5  1003 Baldwin County         223234   132  30.7 -87.7  2.12
## 6     6  1005 Barbour County          24686    29  31.9 -85.4  1.46
```

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

```
## Source : http://tile.stamen.com/toner-lite/5/4/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/5/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/6/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/7/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/8/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/9/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/10/10.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/4/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/5/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/6/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/7/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/8/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/9/11.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/10/11.png
## Source : http://tile.stamen.com/toner-lite/5/4/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/5/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/6/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/7/12.png
```

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

```
## Source : http://tile.stamen.com/toner-lite/5/9/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/10/12.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/4/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/5/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/6/13.png
```

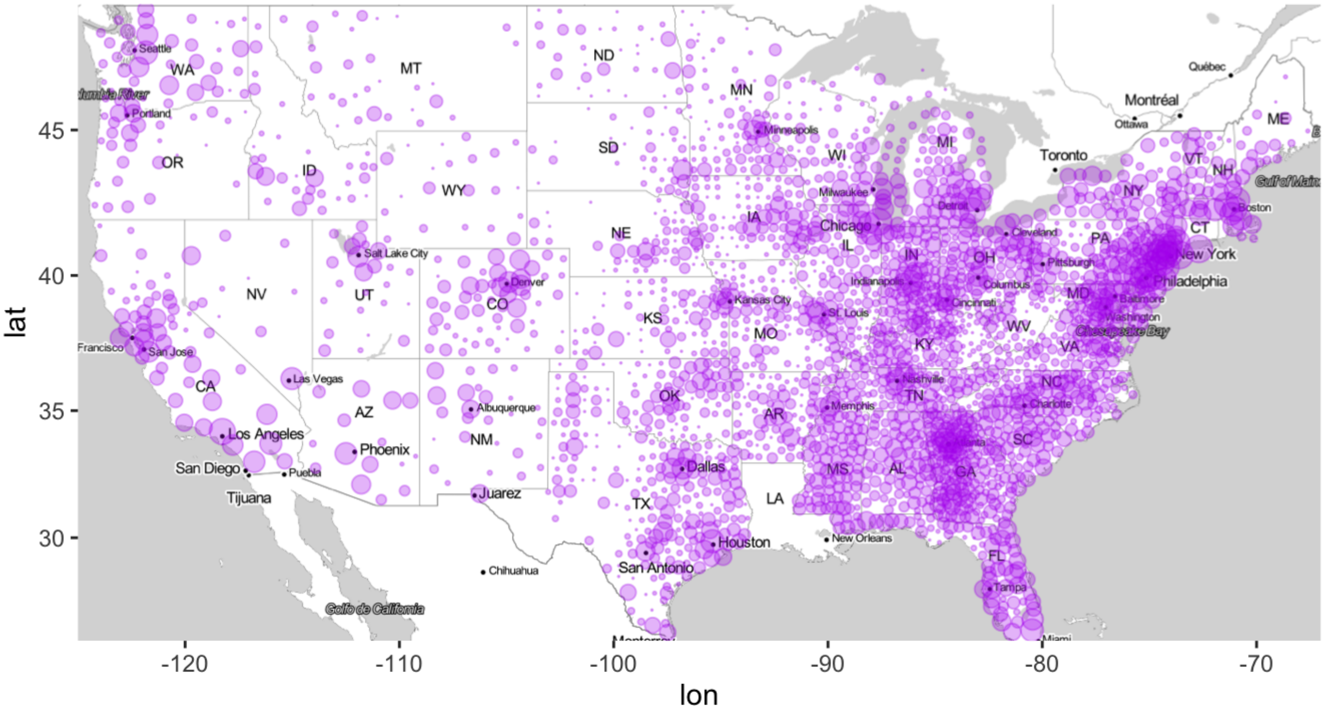
```
## Source : http://tile.stamen.com/toner-lite/5/7/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/8/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/9/13.png
```

```
## Source : http://tile.stamen.com/toner-lite/5/10/13.png
```

```
p = ggmap(US.map)
p+ geom_point(data=data,aes(x=lon,y=lat),colour = "#b207f0",size = data$log,alpha=0.3,na.rm = TRUE)
```


lat
45
40
35
30
lon
-120
-110
-100
-90
-80
-70
Seattle
WA
Columbia River
Portland
OR
ID
MT
ND
MN
SD
WY
Minneapolis
WI
MI
NE
IA
Milwaukee
Chicago
IL
Detroit
Cleveland
Toronto
Ottawa
Montréal
Québec
ME
VT
NH
NY
Gulf of Maine
Boston
CT
PA
New York
Philadelphia
Pittsburgh
OH
IN
Indianapolis
Columbus
Cincinnati
MD
Baltimore
Washington
Chesapeake Bay
WV
VA
KY
Salt Lake City
UT
NV
Denver
CO
Francisco
San Jose
KS
Kansas City
MO
St. Louis
CA
Las Vegas
AZ
Albuquerque
NM
OK
Nashville
TN
NC
Charlotte
AR
Memphis
SC
Atlanta
Los Angeles
Phoenix
Dallas
MS
AL
GA
San Diego
Puebla
Tijuana
Juarez
TX
LA
Houston
New Orleans
San Antonio
FL
Chihuahua
Tampa
Golfo de California
Monterrey
Miami