

# Final Analysis

```
save_dir<-getwd()
setwd("../R")
source("configuration.R")

## Warning: package 'dplyr' was built under R version 3.4.2
##
## 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
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.4.2
## Google Maps API Terms of Service: http://developers.google.com/maps/terms.
## Please cite ggmap if you use it: see citation("ggmap") for details.
setwd(save_dir)
```

The following function plot\_year allows one to plot the top five break-in regions for a given year in Vancouver. It returns a plot with the top five regions, with the break-in locations marked as well the number of break-ins.

```
setwd("../Data")

d <- load_filtered_dataset()

## Warning: package 'bindrcpp' was built under R version 3.4.2
setwd(save_dir)

plot_year <- function(d,year)
{
max_lat <- 49.29723
min_lat <- 49.20091
max_long <- -123.0236
min_long <- -123.2199
center <- c((max_long + min_long)/2, (max_lat + min_lat)/2)

if(!exists("d")) {
  print("loading dataset filtered for breakins and year 2003")
  d <- load_filtered_dataset()
} else {
  print("dataset already loaded!")
}

d <- dplyr::filter(d, YEAR==year)

events <- dplyr::select(d, Longitude, Latitude)
```

```

mt <- ManyTiles(min_lat, max_lat, min_long, max_long, 10, 5, events)

number_tiles <- get_number_tiles.ManyTiles(mt)
num_events <- rep(NA, number_tiles)
for (i in 1:number_tiles) {
  current_tile <- get_tile.ManyTiles(mt, i)
  current_events <- get_events.Tile(current_tile)
  num_events[i] <- nrow(current_events)
}

maxN <- function(x, N){
  len <- length(x)
  if(N>len){
    warning('N greater than length(x). Setting N=length(x)')
    N <- length(x)
  }
  num <- sort(x,partial=len-N+1)[len-N+1]
  log <- x==num
  w.num <- which(log)
  return(w.num)
}

tile1 <- get_tile.ManyTiles(mt,maxN(num_events,N=1))
tile2 <- get_tile.ManyTiles(mt,maxN(num_events,N=2))
tile3 <- get_tile.ManyTiles(mt,maxN(num_events,N=3))
tile4 <- get_tile.ManyTiles(mt,maxN(num_events,N=4))
tile5 <- get_tile.ManyTiles(mt,maxN(num_events,N=5))

Tile_to_rectangle_df <- function(tile) {
  lat <- get_latitudes.Tile(tile)
  long <- get_longitudes.Tile(tile)
  return (data.frame(min_long=long[1], max_long=long[2],
                     min_lat=lat[1], max_lat=lat[2],
                     stringsAsFactors = F))
}

tile_list <- list(tile1, tile2,tile3,tile4,tile5)
df_list <- as.list(rep(NA, 5))
for (i in 1:5) {
  df_list[[i]] <- Tile_to_rectangle_df(tile_list[[i]])
}
df <- do.call(rbind, df_list)

df_events_list <- as.list(rep(NA,5))
for(i in 1:5){
  df_events_list[[i]] <- tile_list[[i]]$events
}
df_events <- do.call(rbind, df_events_list)

long_loc <- rep(NA,length(tile_list))
lat_loc <- rep(NA,length(tile_list))
events <- rep(NA,length(tile_list))
for(i in 1:length(tile_list)){

```

```

long_loc[i] <- tile_list[[i]]$min_long
lat_loc[i] <- tile_list[[i]]$max_lat
events[i] <- nrow(tile_list[[i]]$events)
}
df_label <- data.frame(long_loc=long_loc, lat_loc=lat_loc, events=events, stringsAsFactors=F)

if (!exists("map")) { # loading the map takes a while, so do it only once
  map <- get_ggmap("vancouver ca", zoom = 11)
}

d_year <- unique(d$YEAR)

p <- ggmap(map)
p <- p + geom_rect(mapping=aes(xmin=min_long, xmax=max_long,
                                 ymin=min_lat, ymax=max_lat),
                     data=df, fill=NA, size=1, color="red",
                     inherit.aes=FALSE)
p <- p + geom_point(mapping=aes(x=Longitude, y=Latitude), data=df_events, size=.05)
p <- p + geom_label(mapping=aes(x=long_loc, y=lat_loc, label=events), data=df_label, size=5)
p <- p + geom_label(x=-123.0, y=49.4, label=year, size=7)
print(p)
}

```

The following lines produce plots for all years in the Vancouver data set.

\*The total number of events label is too large. Reducing the size in the geom\_label function did not help.

```

years<-unique(d$YEAR)
for(i in 1:length(years)){
  plot_year(d,years[i])
}

## [1] "dataset already loaded!"

## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1&format=png&key=AIzaSyCwDyfXzJLcOOGQHnVjPmIwzGKUWZBk

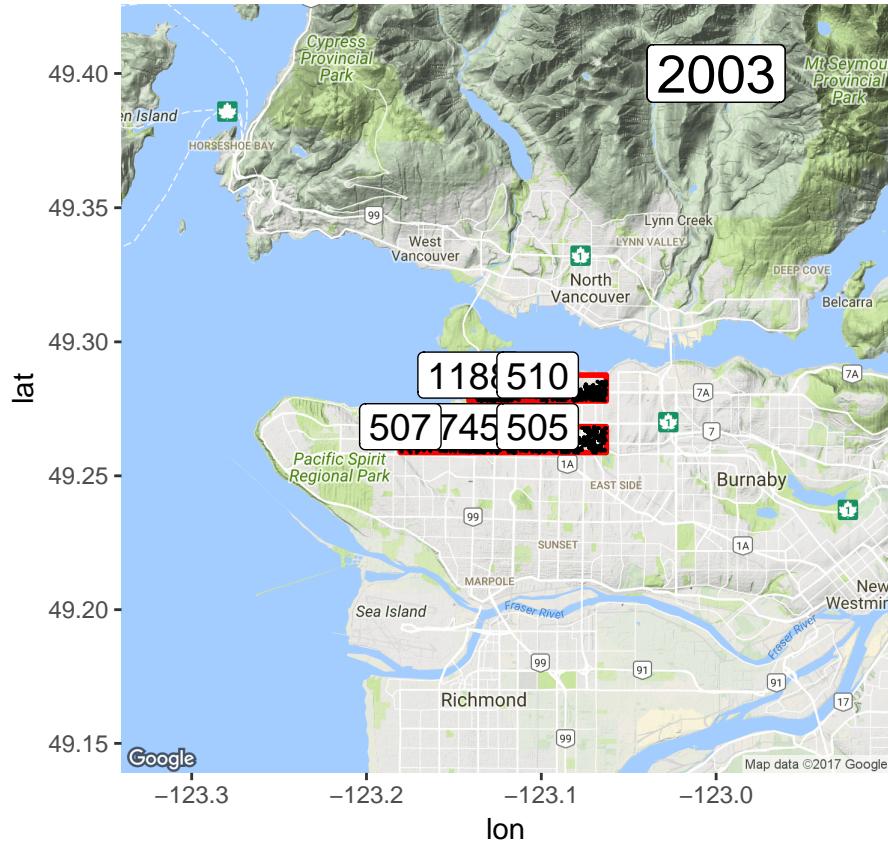
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca

## [1] "dataset already loaded!"

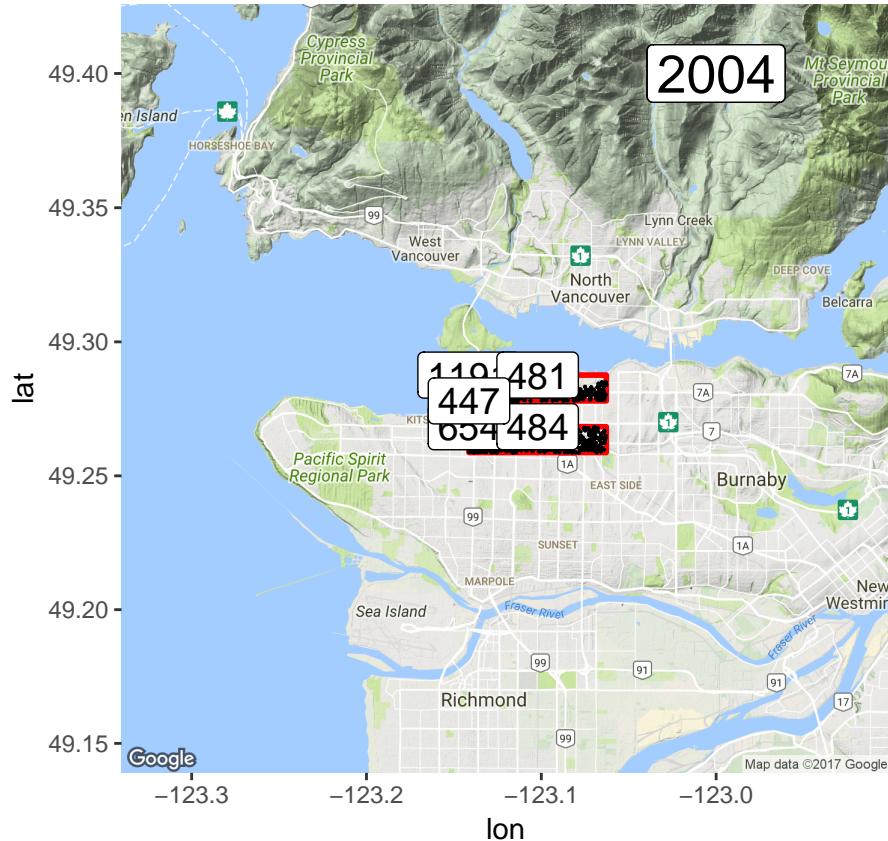
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1&format=png&key=AIzaSyCwDyfXzJLcOOGQHnVjPmIwzGKUWZBk

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca

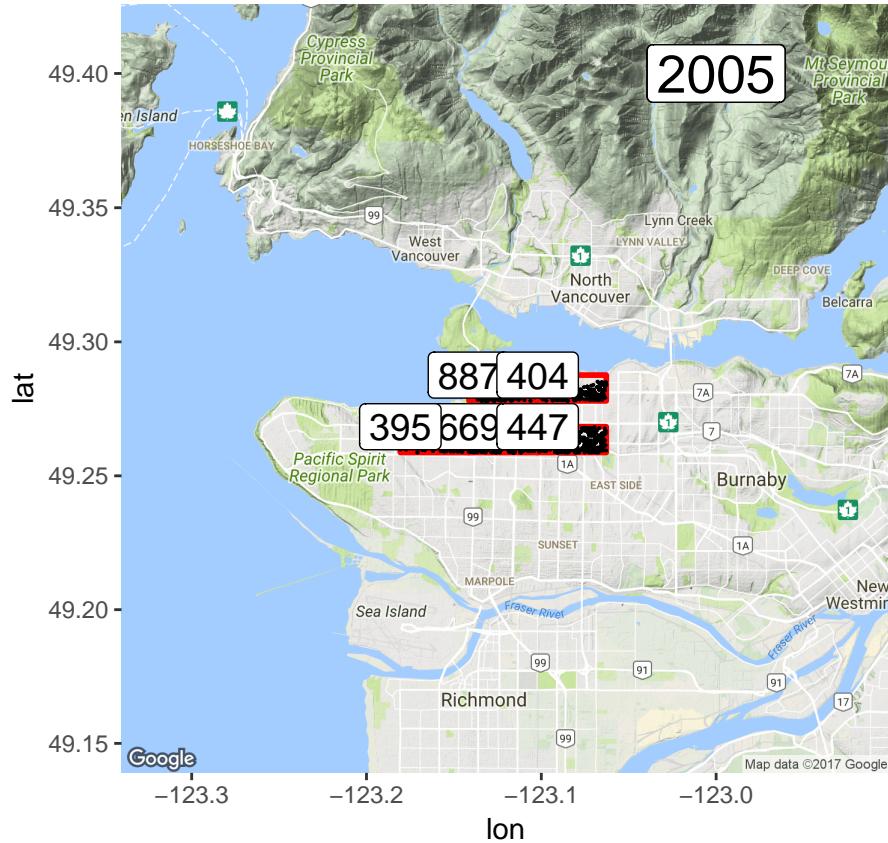
```



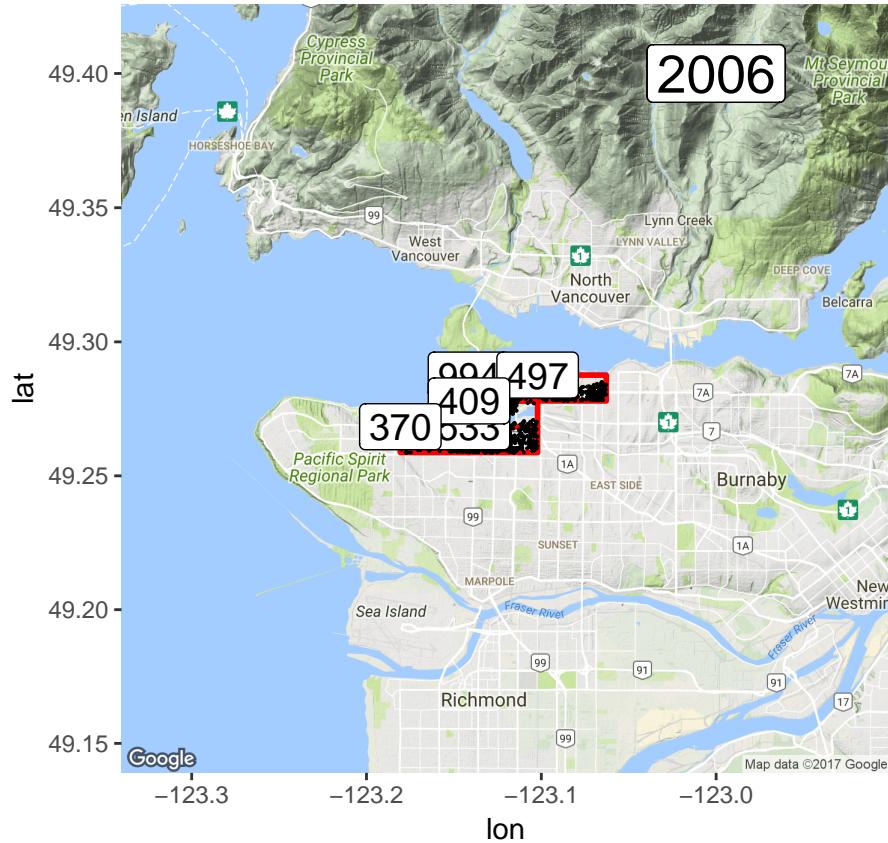
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



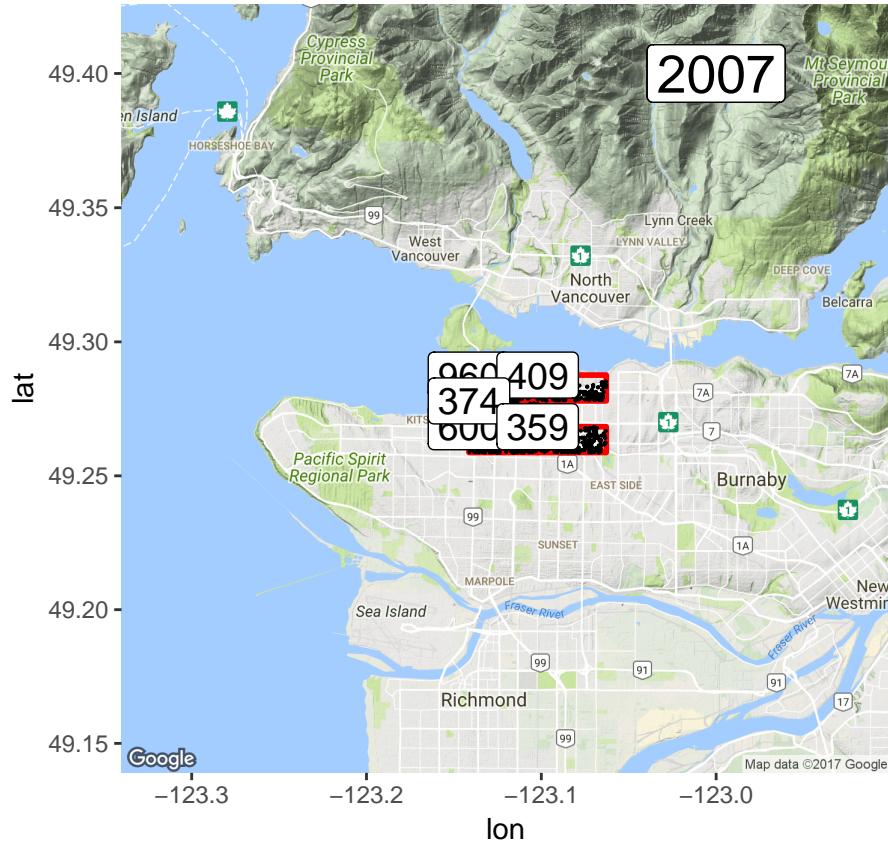
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



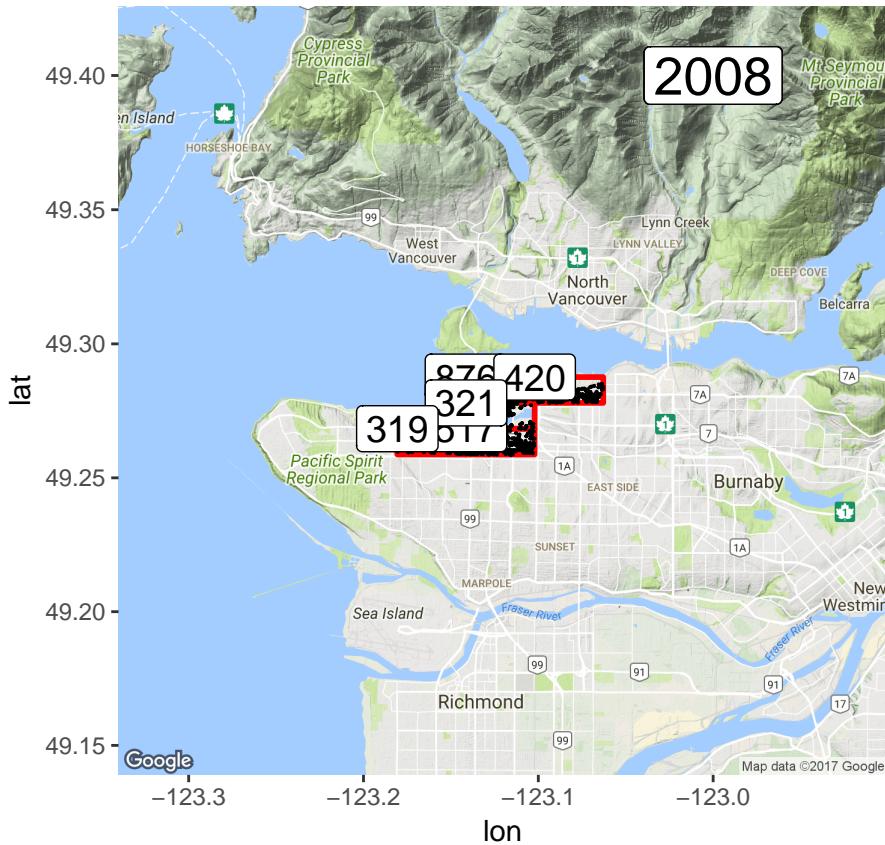
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



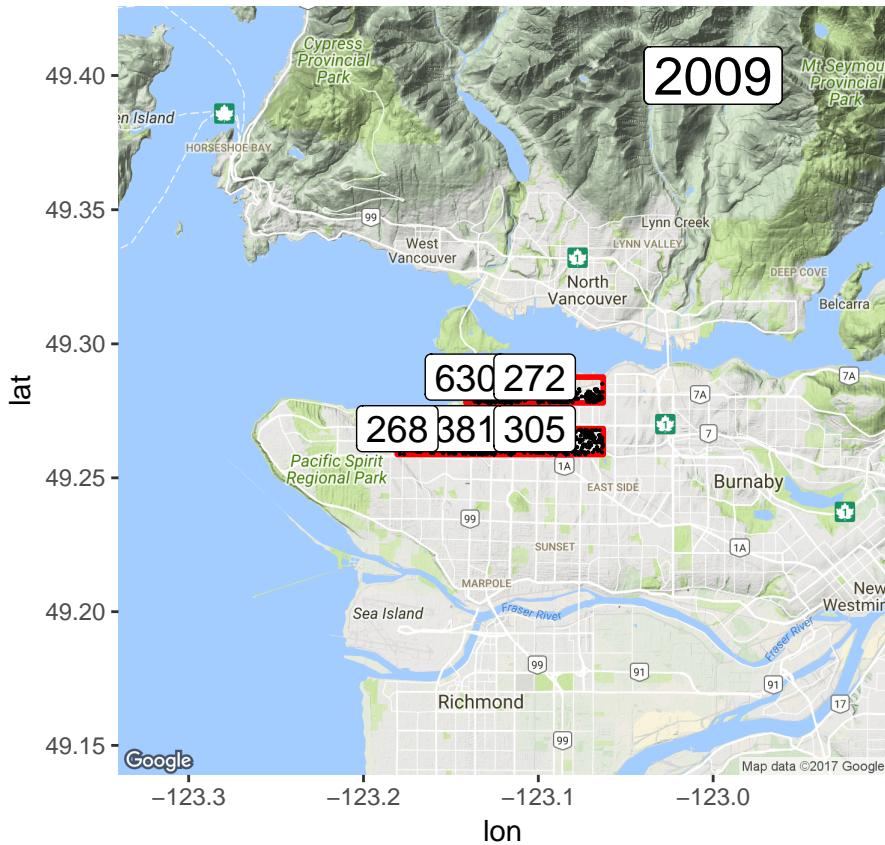
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



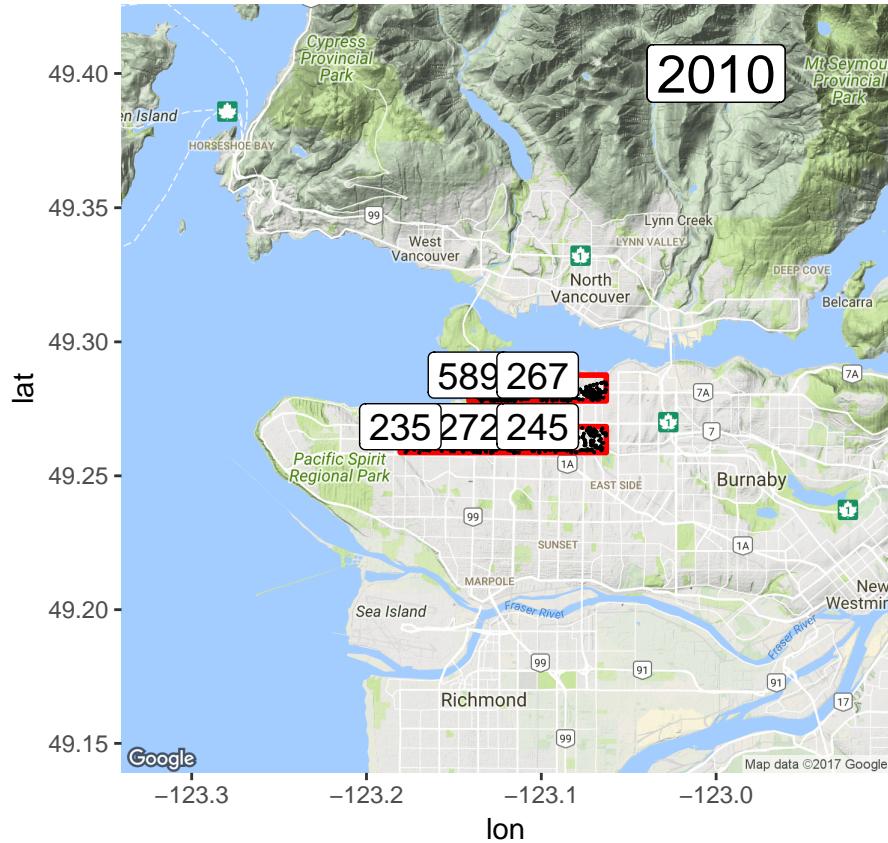
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



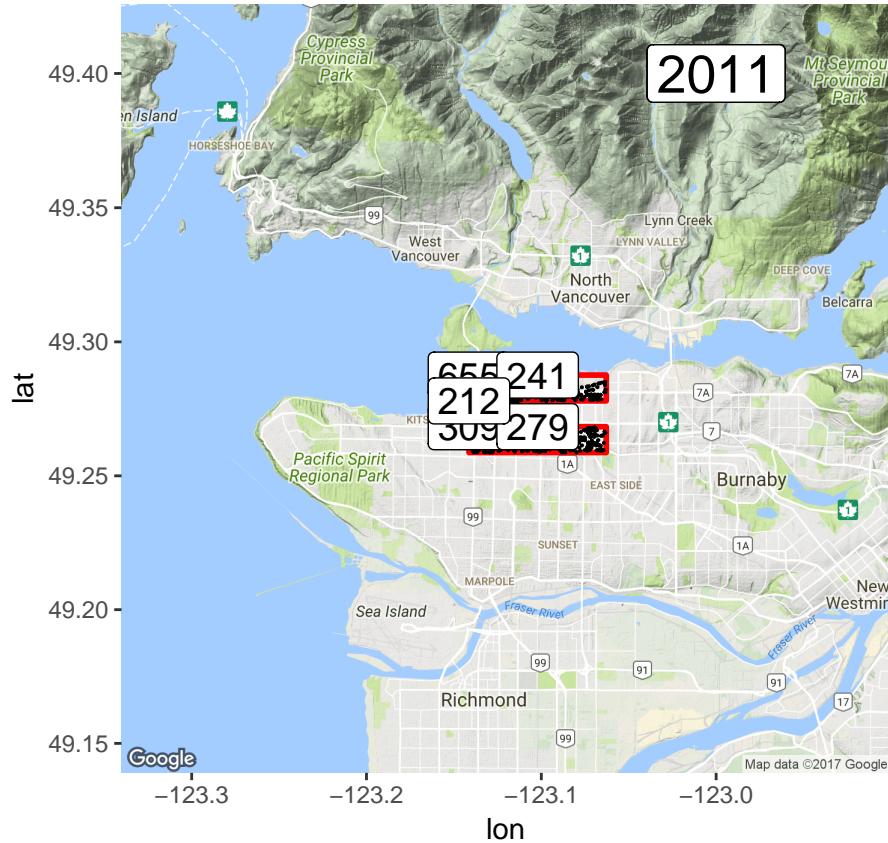
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



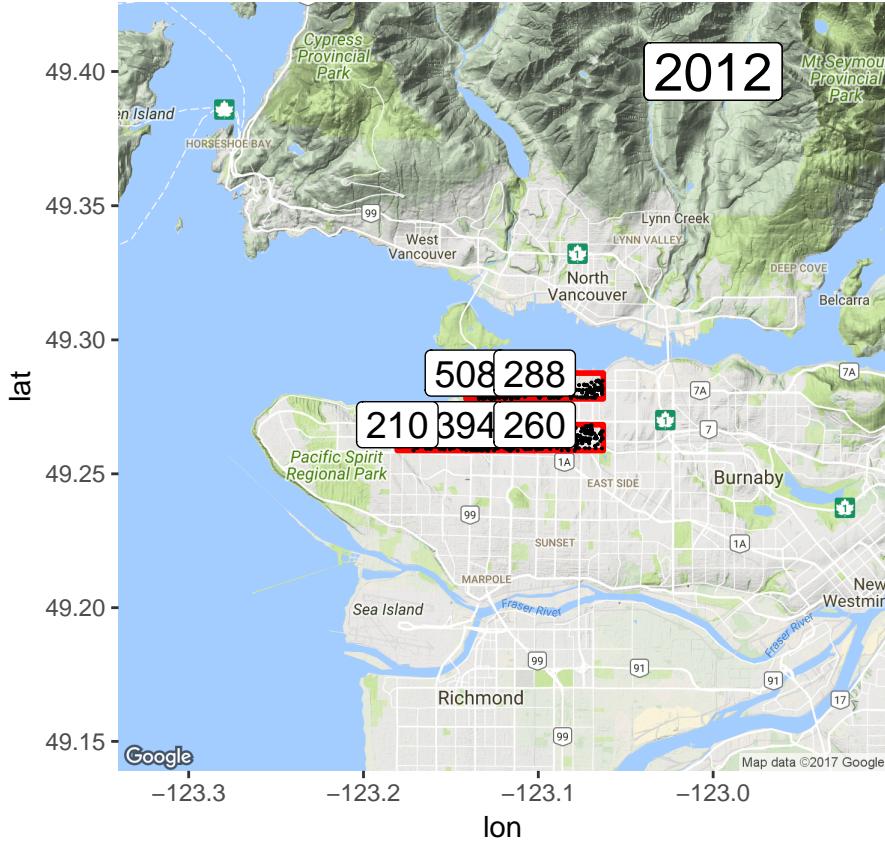
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



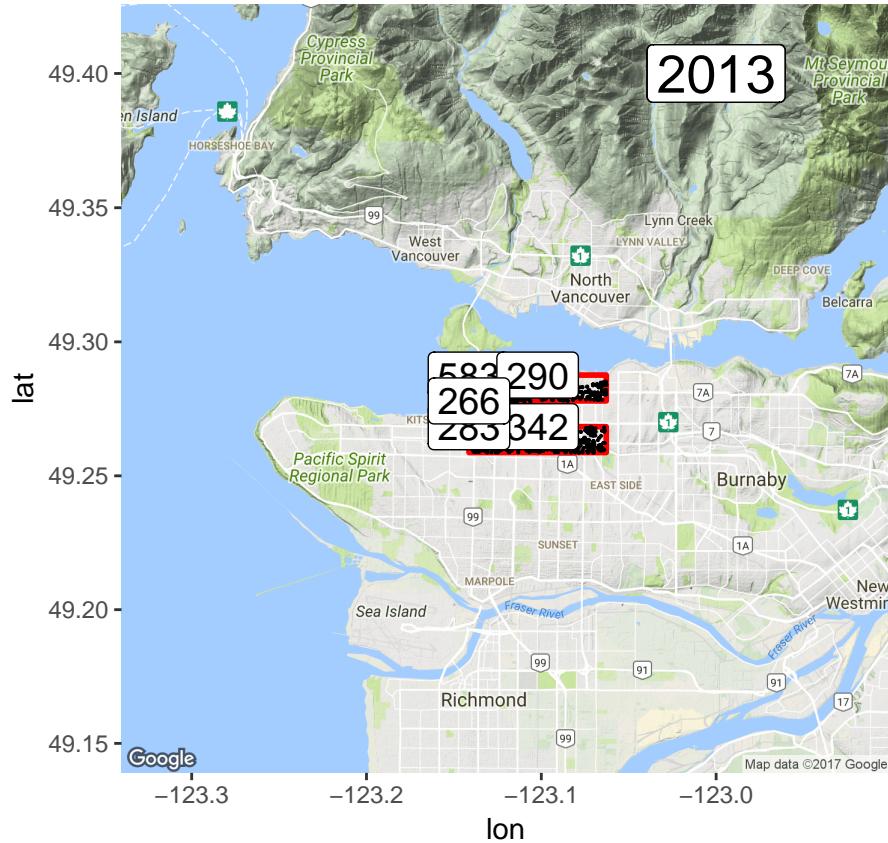
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



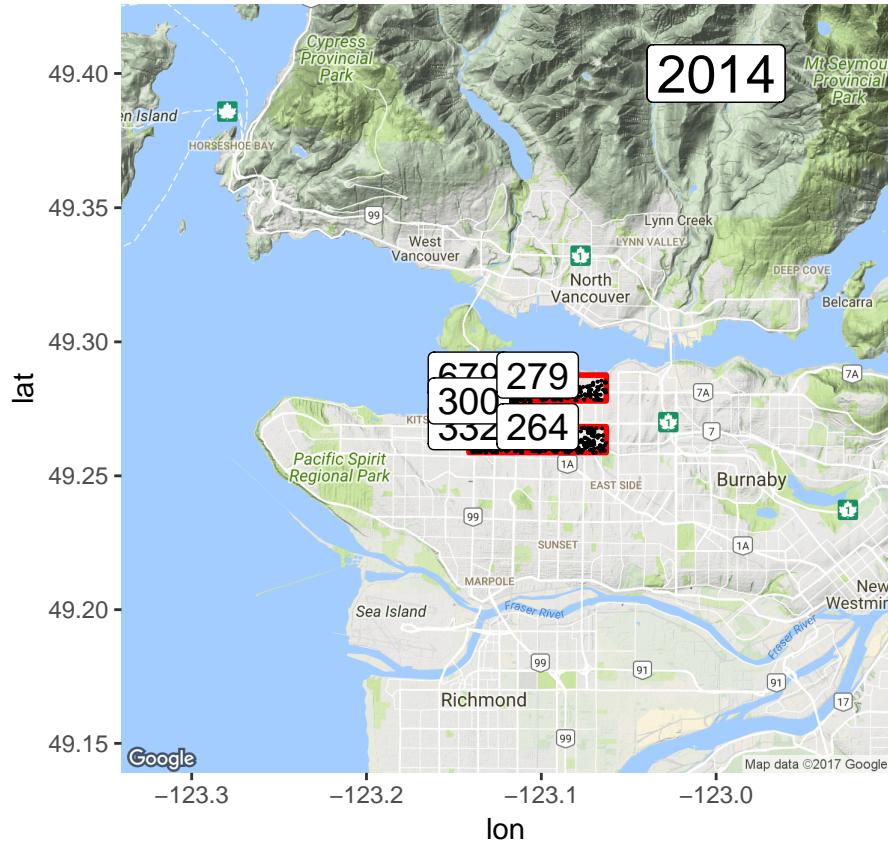
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



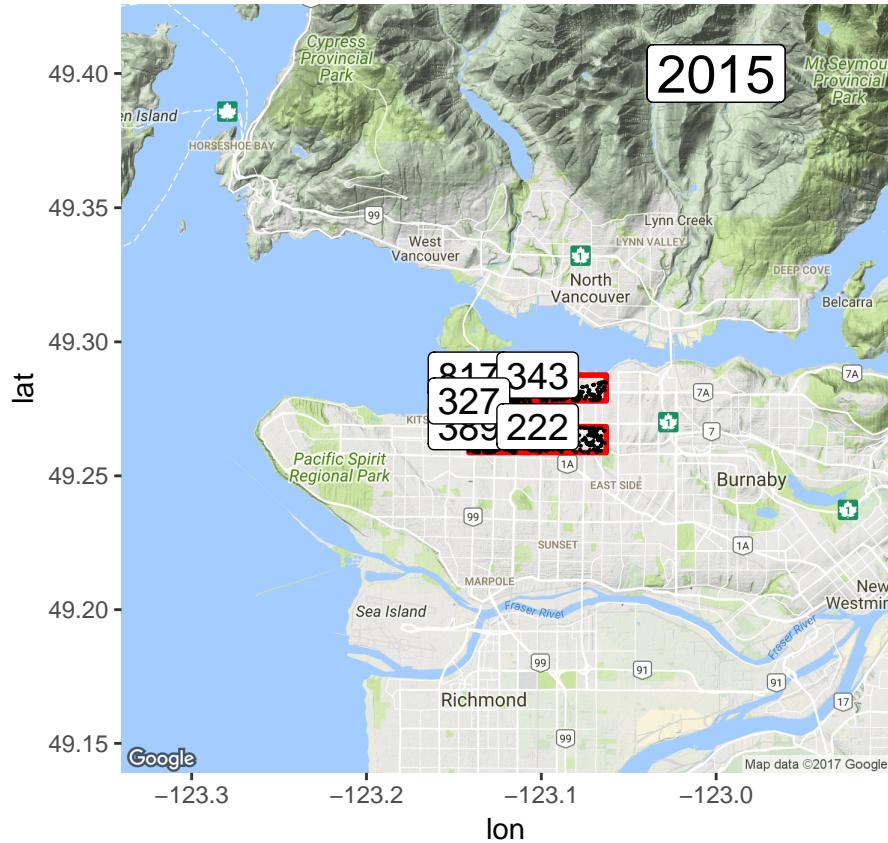
```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```



```
## [1] "dataset already loaded!"  
## Source : https://maps.googleapis.com/maps/api/staticmap?center=vancouver+ca&zoom=11&size=640x640&scale=1  
## Source : https://maps.googleapis.com/maps/api/geocode/json?address=vancouver%20ca
```

