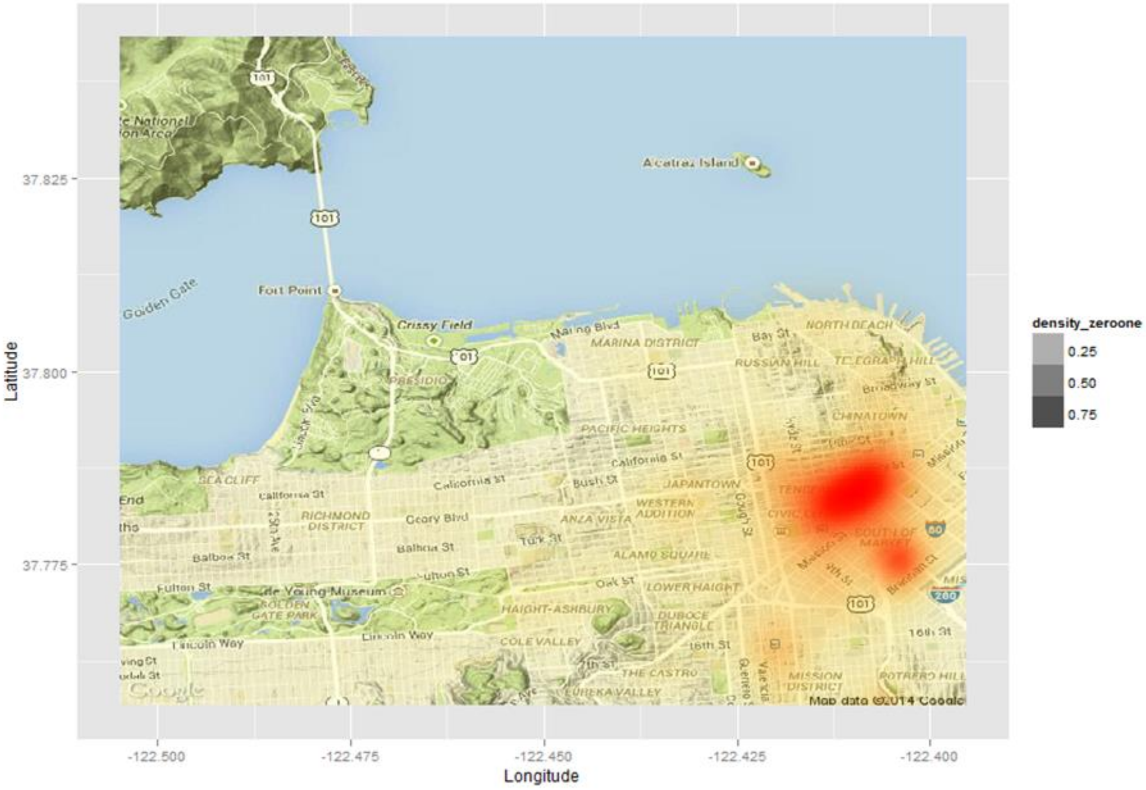
RgoogleMaps



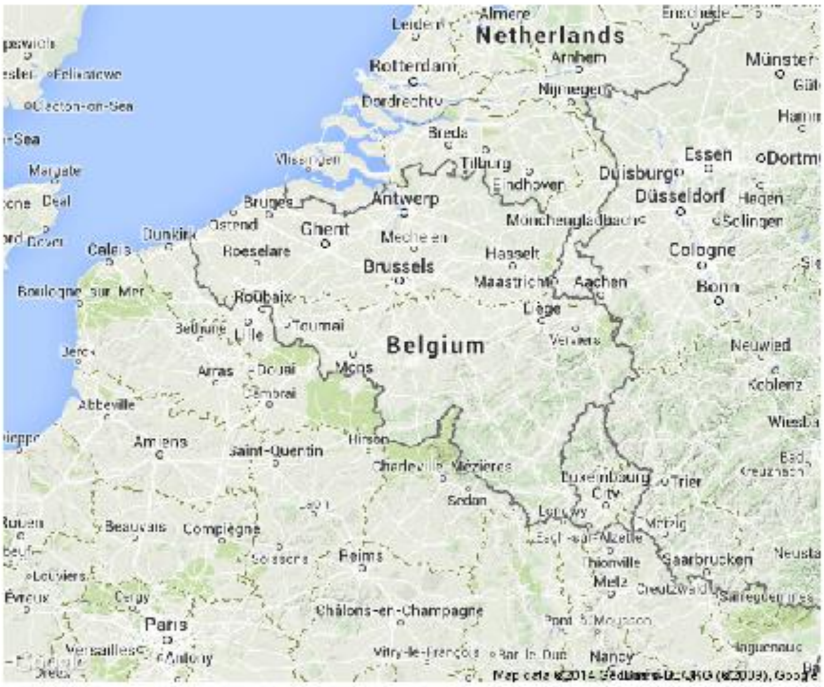
Johan Blomme

2016

www.johanblomme.net

① Retrieve base maps from Google with gmap function in package dismo

```
library(dismo)
mymap <- gmap("Belgium")
```

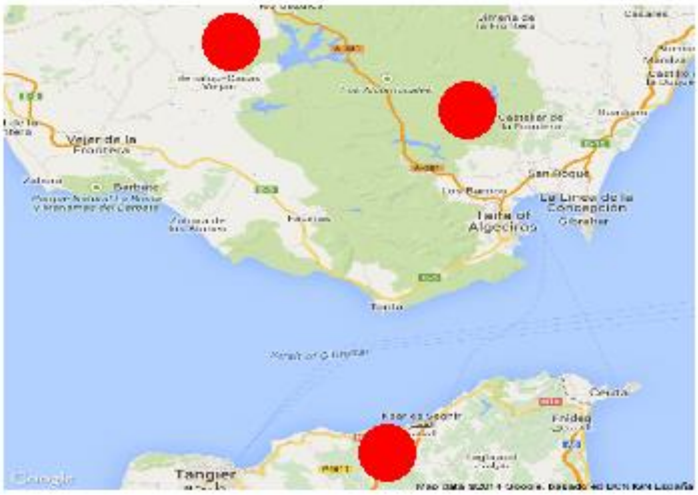


```
mymap <- gmap("Belgium",type="satellite")
plot(mymap)
```

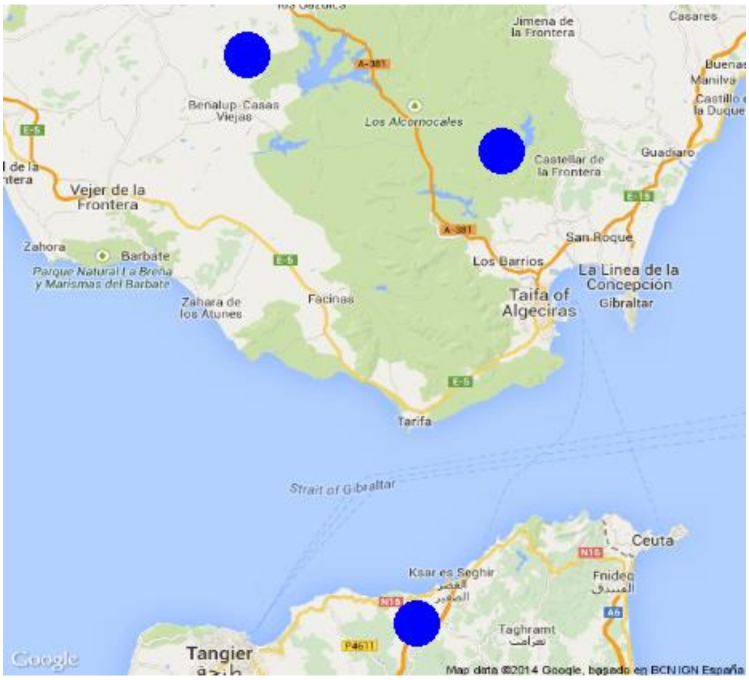
Google
Imagery ©2014 TerraMetrics

② RgoogleMaps : Map data on Google map tiles

```
library(RgoogleMaps)
# with center coordinates
newmap <- GetMap(center = c(36.7, -5.9), zoom = 10, destfile = "newmap.png",
                 maptype = "satellite")
PlotOnStaticMap(mymap=newmap,lat = c(36.3, 35.8, 36.4), lon = c(-5.5, -5.6, -5.8),
                zoom = 10,cex = 4, pch = 19, col = "red")
```

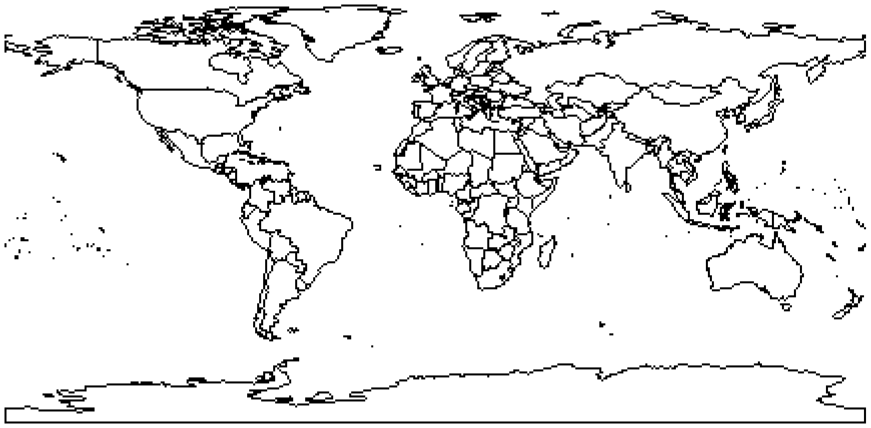


```
# with bounding box
newmap2 <- GetMap.bbox(lonR = c(-5, -6), latR = c(36, 37), destfile =
                       "newmap2.png", maptype = "terrain")
PlotOnStaticMap(mymap=newmap2,lat = c(36.3, 35.8, 36.4), lon = c(-5.5, -5.6, -5.8),
                zoom = 10,cex = 4, pch = 19, col = "blue")
```



③ Mapping global data

```
library(rworldmap)
mymap <- getMap(resolution="coarse")
plot(mymap)
```



```
# retrieve point occurrence data from GBIF
# example dataset: retrieve occurrence data for the laurel tree (Laurus nobilis)
# from the Global Biodiversity Information Facility (GBIF)

library(dismo)
laurus <- gbif("Laurus", "nobilis")
```

```
Laurus nobilis : 2120 occurrences found
1-1000-2000-2120
```

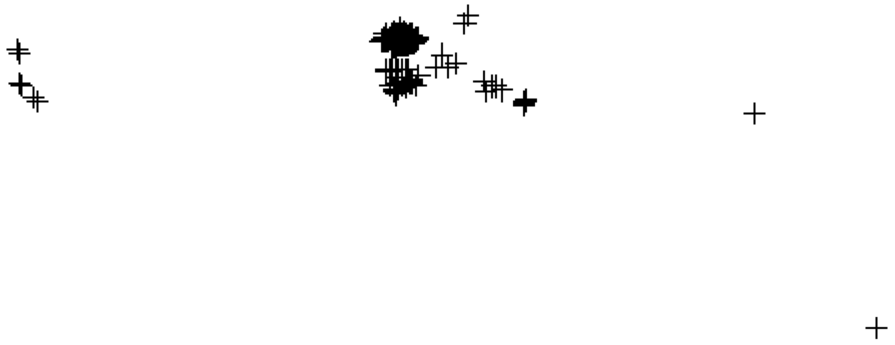
```
# get data frame with spatial coordinates (points)
locs <- subset(laurus, select = c("country", "lat", "lon"))
head(locs)
```

```
> head(locs)
        country      lat         lon
1 United States       NA          NA
2 United States 34.00000 -118.000000
3 United States 33.00000 -117.000000
4         Spain 36.22924   -5.716103
5         Spain 37.30082   -1.918051
6         Spain 36.09781   -5.544987
```

```
# discard data with errors in coordinates:
locs <- subset(locs, locs$lat < 90)
# set spatial coordinates
coordinates(locs) <- c("lon", "lat")
str(locs)
```

```
> str(locs)
Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
  ..@ data       :'data.frame':  2109 obs. of  1 variable:
  .. ..$ country: chr [1:2109] "United States" "United States" "Spain" "Spain" ...
  ..@ coords.nrs : int [1:2] 3 2
  ..@ coords     : num [1:2109, 1:2] -118 -117 -5.72 -1.92 -5.54 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : NULL
  .. .. ..$ : chr [1:2] "lon" "lat"
  ..@ bbox       : num [1:2, 1:2] -123.3 -37.8 145 59.8
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr [1:2] "lon" "lat"
  .. .. ..$ : chr [1:2] "min" "max"
  ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slots
  .. .. ..@ projargs: chr NA
```

```
plot(locs)
```



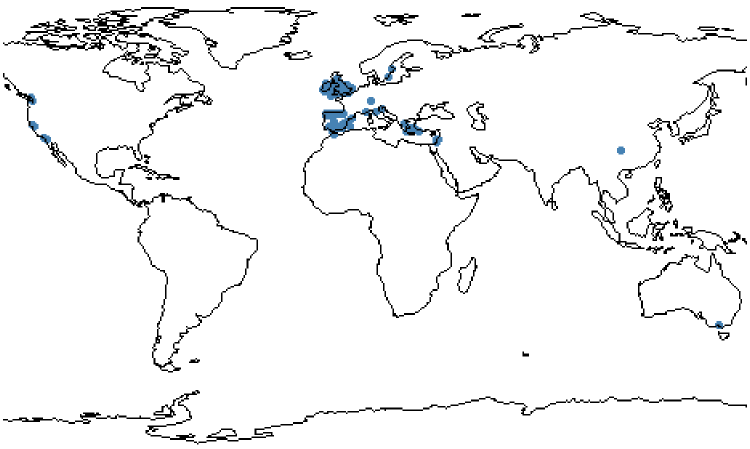
```
# define spatial projection
proj4string(locs)
```

```
> proj4string(locs)
[1] NA
```

```
crs.geo <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84")
proj4string(locs) <- crs.geo
summary(locs)
```

```
> summary(locs)
Object of class SpatialPointsDataFrame
Coordinates:
          min      max
lon -123.2508 145.0433
lat  -37.7786  59.8448
Is projected: FALSE
proj4string :
[+proj=longlat +ellps=WGS84 +datum=WGS84 +towgs84=0,0,0]
Number of points: 2109
Data attributes:
   Length     Class      Mode
     2109 character character
```

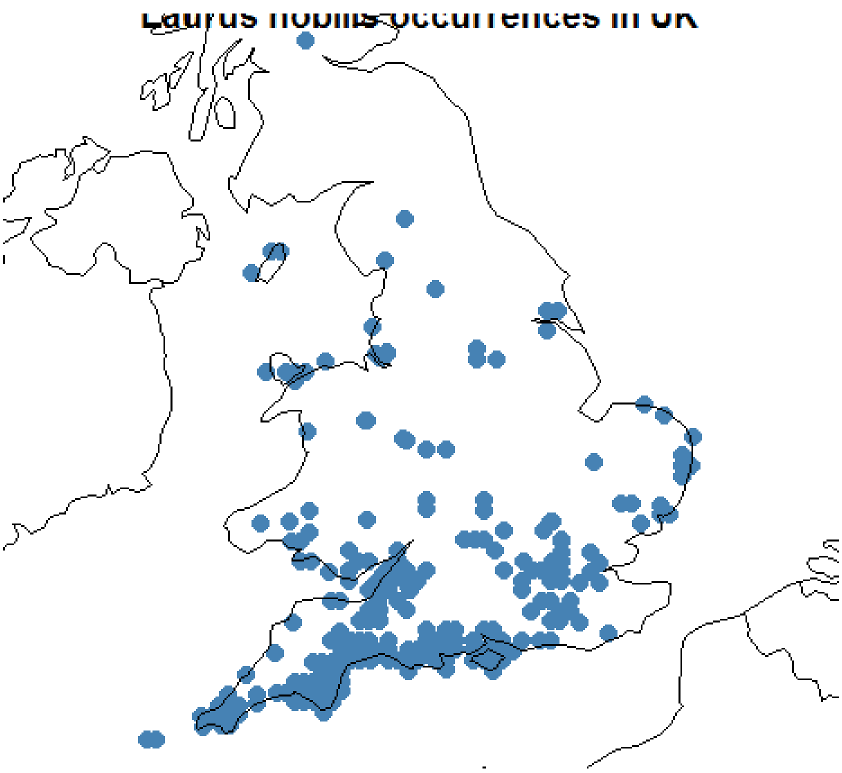
```
# library rworldmap provides different types of global maps, e.g:
plot(locs, pch = 20, col = "steelblue")
data(coastsCoarse)
data(countriesLow)
plot(coastsCoarse, add = T)
```



```
table(locs$country)
```

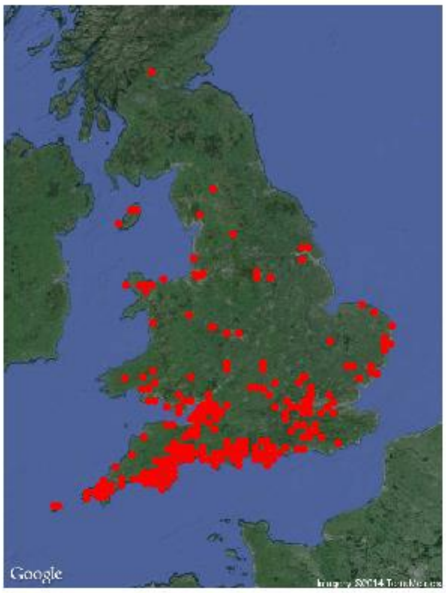
```
> table(locs$country)

     Australia         Canada        Croatia         France        Germany         Greece        Ireland
Israel
             2              1              1              1              1              5             69
1231
         Italy          Spain         Sweden United Kingdom  United States
             2            206              2            578             10
```

```
locs.gb <- subset(locs, locs$country == "United Kingdom")
plot(locs.gb, pch = 20, cex = 2, col = "steelblue")
title("Laurus nobilis occurrences in UK")
plot(countriesLow, add = T)
```



```
library(RgoogleMaps)
summary(locs.gb)
gbmap <- gmap(locs.gb, type = "satellite")
plot(gbmap)
```

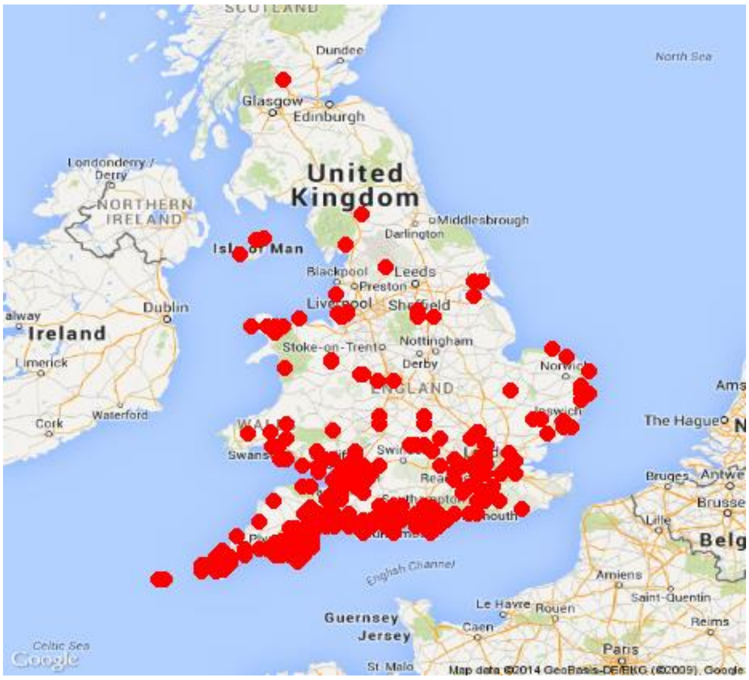
```
locs.gb.merc <- Mercator(locs.gb)
points(locs.gb.merc, pch = 20, col = "red")
```



```
# retrieve coordinates from locs.gb
locs.gb.coords <- as.data.frame(coordinates(locs.gb))
str(locs.gb.coords)
```

```
> str(locs.gb.coords)
'data.frame':        578 obs. of  2 variables:
 $ lon: num  -2.216 -0.364 -0.197 -2.856 -2.597 ...
 $ lat: num  51.5 51.9 51.7 51.4 51.4 ...
```

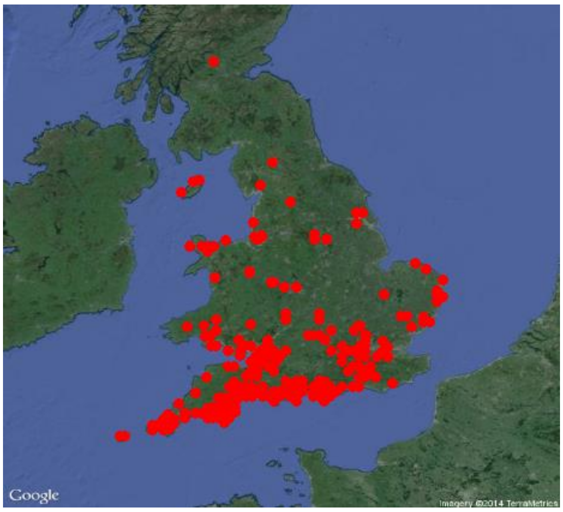
```
PlotOnStaticMap(lat = locs.gb.coords$lat, lon = locs.gb.coords$lon, zoom = 5,
                cex = 1.4, pch = 19, col = "red", FUN = points)
```



```
# download base map from Google maps and plot points on it
# define the region
map.lim <- qbbox(locs.gb.coords$lat, locs.gb.coords$lon, TYPE = "all")
# bounding box
mymap <- GetMap.bbox(map.lim$lonR, map.lim$latR, destfile = "gmap.png", maptype =
                     "satellite")
```

```
> mymap <- GetMap.bbox(map.lim$lonR, map.lim$latR, destfile = "gmap.png", maptype = "satellite")
[1] "http://maps.google.com/maps/api/staticmap?center=53.086237,-
2.30987445&zoom=6&size=640x640&maptype=satellite&format=png32&sensor=true"
```

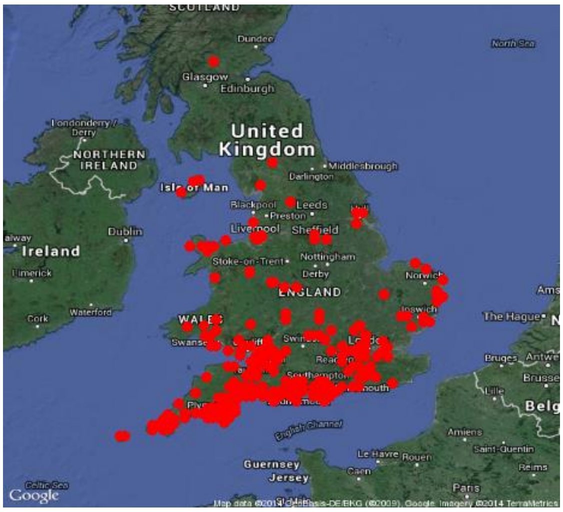
```
PlotOnStaticMap(mymap, lat = locs.gb.coords$lat, lon = locs.gb.coords$lon, zoom =
                NULL,cex = 1.3, pch = 19, col = "red")
```



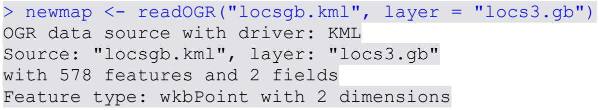
```
# using different background
mymap <- GetMap.bbox(map.lim$lonR, map.lim$latR, destfile = "gmap.png", maptype =
                     "hybrid")
```

```
> mymap <- GetMap.bbox(map.lim$lonR, map.lim$latR, destfile = "gmap.png", maptype = "hybrid")
[1] "http://maps.google.com/maps/api/staticmap?center=53.086237,-
2.30987445&zoom=6&size=640x640&maptype=hybrid&format=png32&sensor=true"
```

```
PlotOnStaticMap(mymap, lat = locs.gb.coords$lat, lon = locs.gb.coords$lon, zoom =
                NULL,cex = 1.3, pch = 19, col = "red")
```

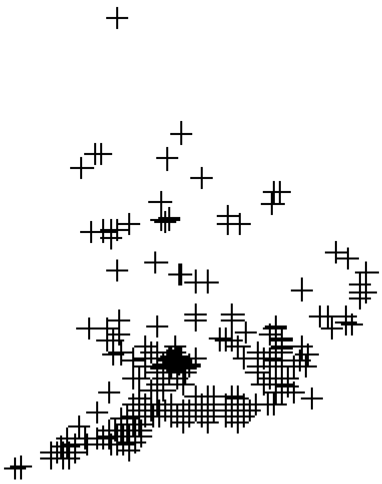


```
# saving and reading in spatial vector data
setwd("c:/R/Rdata")

# export KML
writeOGR(locs.gb, dsn = "locsgb.kml", layer = "locs3.gb", driver = "KML")
# read KML
newmap <- readOGR("locsgb.kml", layer = "locs3.gb")
```

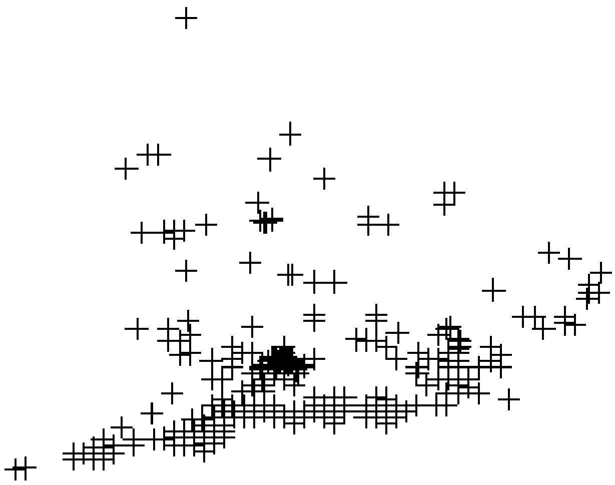


```
> newmap <- readOGR("locsgb.kml", layer = "locs3.gb")
OGR data source with driver: KML
Source: "locsgb.kml", layer: "locs3.gb"
with 578 features and 2 fields
Feature type: wkbPoint with 2 dimensions
```

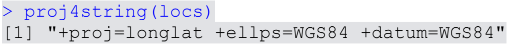
```
plot(newmap)
```



```
# save as shapefile
library(maptools)
writePointsShape(locs.gb, "locsgb")
# read shapefile
gb.shape <- readShapePoints("locsgb.shp")
plot(gb.shape)
```

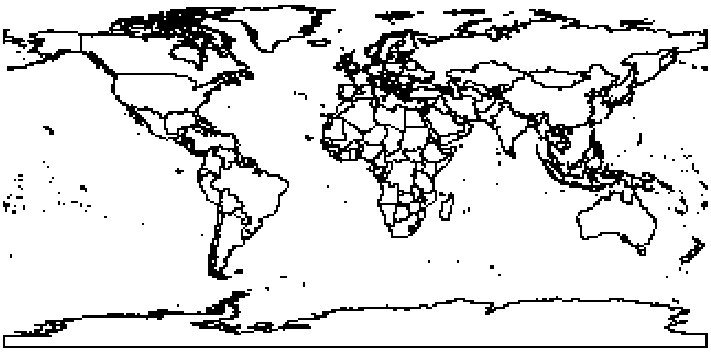


```
# changing the projection of spatial vector data

# library sp will do the projection
library(sp)
proj4string(locs)
```

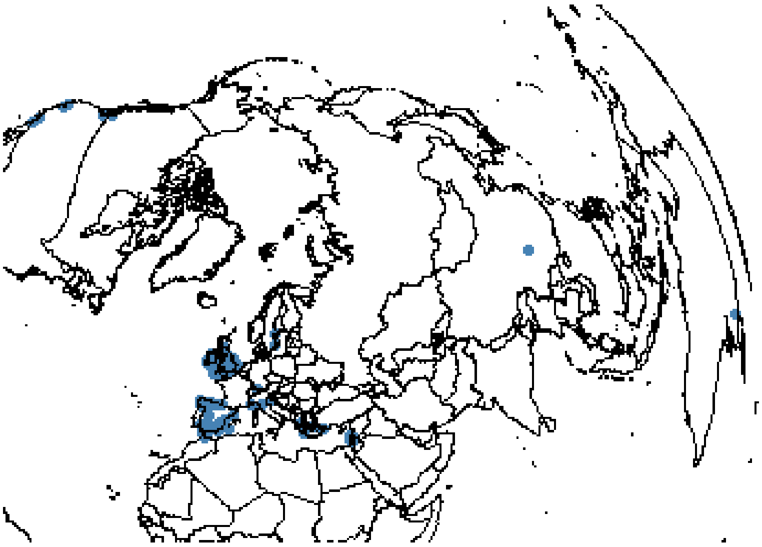


```
> proj4string(locs)
[1] "+proj=longlat +ellps=WGS84 +datum=WGS84"
```

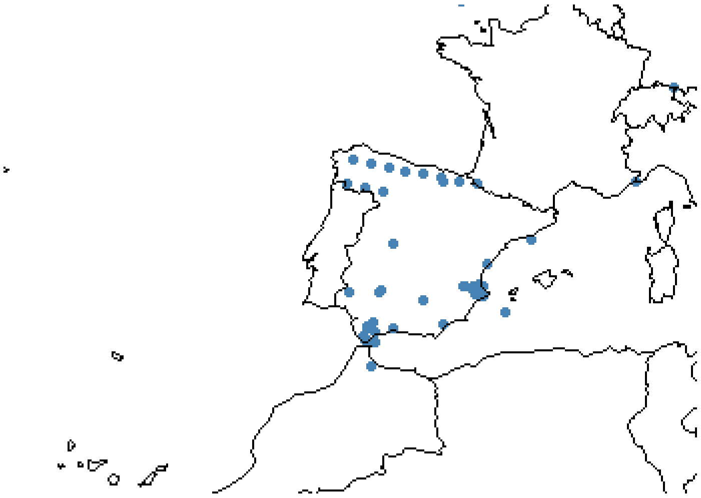
```
plot(countriesLow)
```



```
country.laea <- spTransform(countriesLow, crs.laea)
plot(locs.laea, pch = 20, col = "steelblue")
plot(country.laea, add = T)
```



```
# define spatial limits for plotting
plot(locs.laea, pch = 20, col = "steelblue", xlim = c(1800000, 3900000), ylim =
     c(1000000,3000000))
plot(country.laea, add = T)
```



④ RgoogleMaps and ggplot2

```
# example Italian schools
# data from the Italian ministry of education about the rate of school abandonment

setwd("c:/R/Rdata")
library(RgoogleMaps)

# schools data
schools <- read.csv("Anagrafica.csv",header=T,sep="|")
str(schools)
```

```
> str(schools)
'data.frame':          72356 obs. of  15 variables:
 $ codice_scuola          : Factor w/ 72356 levels "AG1A001006","AG1A002002",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ denominazione          : Factor w/ 54207 levels "","''A. ILVENTO''-GRASSANO",..: 48542 2862 22468 2876
2872 12583 12615 12611 12582 2967 ...
 $ des_tipo_scuola        : Factor w/ 4 levels "SCUOLA DELL'INFANZIA",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ indirizzo              : Factor w/ 48799 levels "","- VIA PASSANTI  N. 280 RIONE CASILLI",..: 19073
23036 43537 32585 29742 31795 1960 18319 20981 977 ...
 $ comune                 : Factor w/ 7242 levels "ABANO TERME",..: 3566 69 69 69 69 134 1109 1109 1109
1572 ...
 $ cap                    : Factor w/ 4903 levels "","-----","00010",..: 4692 4709 4709 4709 4709 4689 4703
4703 4703 4704 ...
 $ telefono               : Factor w/ 47701 levels "","+922816434",..: 42018 41398 41399 41390 41402 41588
41472 41466 41474 41541 ...
 $ fax                    : Factor w/ 27221 levels "","/6439","+922816434",..: 24413 24082 24083 24079
24027 24179 24117 24116 24095 24153 ...
 $ email                  : Factor w/ 19896 levels "","04Alepaderi@tiscali.it",..: 6252 6317 1 9057 17057
3811 3803 3831 17995 11669 ...
 $ pec                    : Factor w/ 10169 levels "","agee00100t@pec.istruzione.it",..: 1 1 1 1 1 1 1 1 1
1 ...
 $ sito_web               : Factor w/ 13651 levels "","-------","---------------------------------",..: 1 1
1 1 1 1 1 1 1 1 ...
 $ cod_istituto_principale: Factor w/ 25625 levels "AG1A001006","AG1A002002",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ statale                : Factor w/ 2 levels "PARITARIA","STATALE": 1 1 1 1 1 1 1 1 1 1 ...
 $ LATITUDINE             : num  NA NA NA NA NA NA NA NA NA NA ...
 $ LONGITUDINE            : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
# create a new dataframe with only the necessary information
schools_geo <- subset(schools,select=c(codice_scuola,LATITUDINE,LONGITUDINE))
# omit rows with missing values
schools_geo <- na.omit(schools_geo)
dim(schools_geo)
head(schools_geo)
tail(schools_geo)
```

```
> head(schools_geo)
   codice_scuola LATITUDINE LONGITUDINE
68    AGAA00100L      37.30       13.59
69    AGAA00101N      37.30       13.59
70    AGAA00200C      37.31       13.57
71    AGAA00201D      37.31       13.57
72    AGAA00203G      37.33       13.42
73    AGAA003008      37.30       13.58
> tail(schools_geo)
      codice_scuola LATITUDINE LONGITUDINE
72351    VVTF04000P      38.55       15.93
72352    VVTH01000A      38.74       16.17
72353    VVTL00601X      38.68       16.12
72354    VVTL00651T      38.68       16.12
72355    VVVC010001      38.67       16.10
72356    VVVC02000G      38.66       16.11
```

```
# abandonment data
abandonments <- read.csv("abbandoni.csv",header=T,sep="|")
str(abandonments)
```

```
> str(abandonments)
'data.frame':	47021 obs. of  5 variables:
 $ cod_scuola: Factor w/ 12469 levels "AGMM004008","AGMM01500P",..: 1 1 1 2 2 2 3 3 3 4 ...
 $ anno_corso: int  1 2 3 1 2 3 1 2 3 1 ...
 $ scuola    : num  0 0 0 0 0 0 1.3 0 0 0 ...
 $ regione   : num  0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 ...
 $ nazionale : num  0.2 0.1 0.2 0.2 0.1 0.2 0.2 0.1 0.2 0.2 ...
```

```
# exclude schools with missing value for abandonment rate
abandonments <- subset(abandonments,abandonments$scuola != "NA")
dim(abandonments)
```

```
> dim(abandonments)
[1] 45574     5
```

```
# rename unique code of schools
colnames(schools_geo)[1] <- "cod_scuola"
# merge the two datasets by the unique school key
data <- merge(abandonments,schools_geo,by="cod_scuola")
str(data)
```

```
> str(data)
'data.frame':	42417 obs. of  7 variables:
 $ cod_scuola : Factor w/ 12469 levels "AGMM004008","AGMM01500P",..: 1 1 1 2 2 2 3 3 3 4 ...
 $ anno_corso : int  3 2 1 2 1 3 1 2 3 1 ...
 $ scuola     : num  0 0 0 0 0 0 1.3 0 0 0 ...
 $ regione    : num  0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 ...
 $ nazionale  : num  0.2 0.1 0.2 0.1 0.2 0.2 0.2 0.1 0.2 0.2 ...
 $ LATITUDINE : num  37.3 37.3 37.3 37.3 37.3 ...
 $ LONGITUDINE: num  13.6 13.6 13.6 13.6 13.6 ...
```

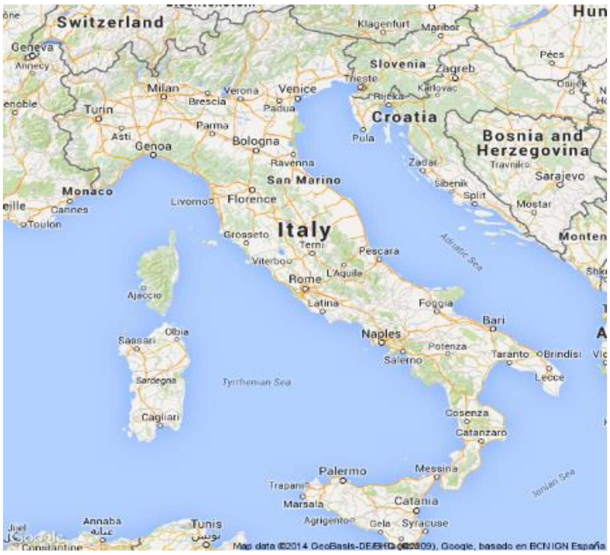
```
# delete records with abandonment rate= 0 (to make the visualization clearer)
data <- data[which(data$scuola > 0),]
dim(data)
```

```
> dim(data)
[1] 4545    7
```

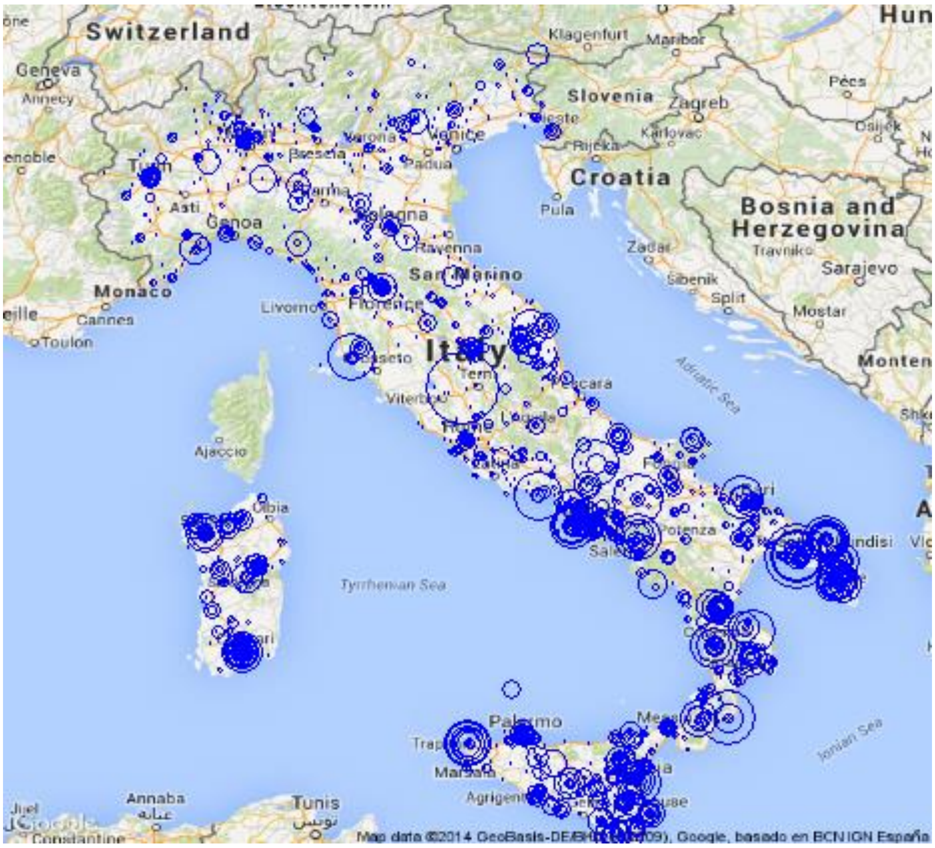
```
# map setup with RgoogleMaps
# setup of latitude and longitude centered on Italy
lat_c <- 42.1
lon_c <- 12.5
# create a rectangle of coordinates
rectangle <- qbbox(lat=c(lat_c[1]+5,lat_c[1]-5),lon=c(lon_c[1]+5,lon_c[1]-5))
str(rectangle)
```

```
> str(rectangle)
List of 2
 $ latR: num [1:2] 37.1 47.1
 $ lonR: num [1:2] 7.45 17.55
```

```
# retrieve a map from googlemaps
map <- GetMap.bbox(rectangle$lonR,rectangle$latR)
# plot the map
PlotOnStaticMap(MyMap=map)
```



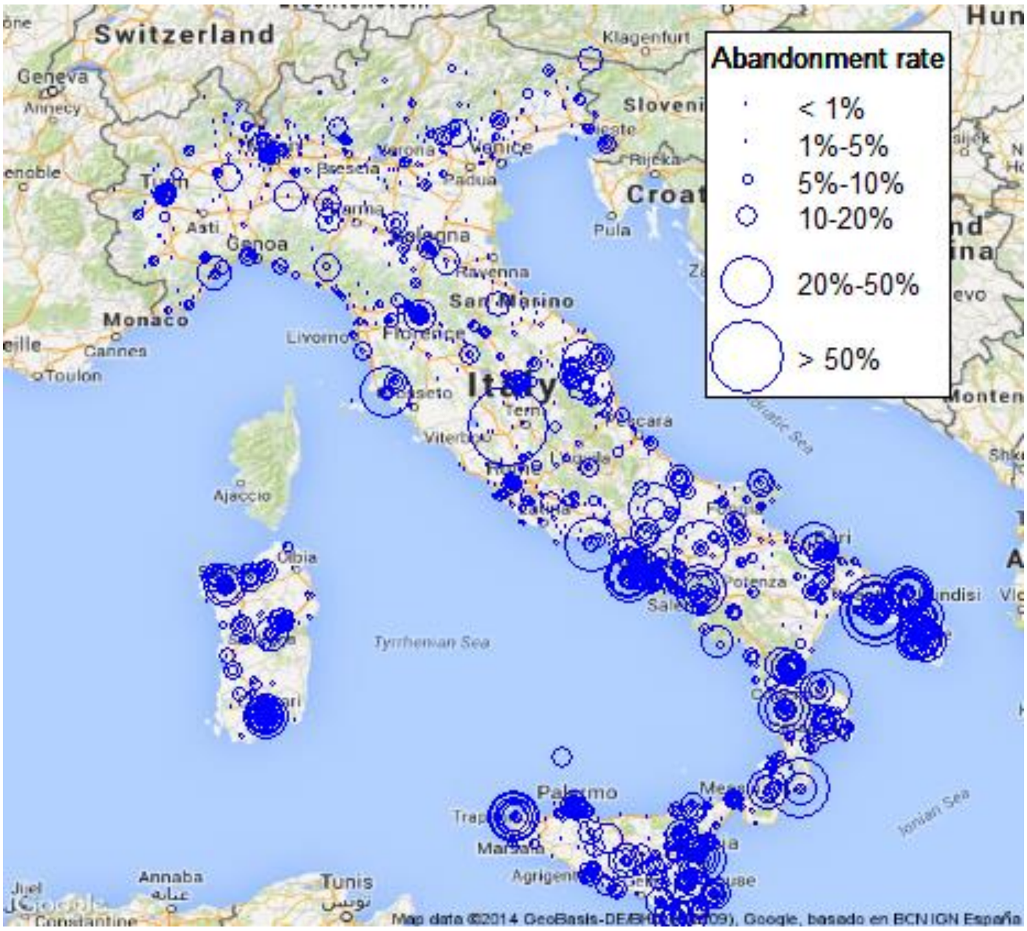
```
# plot the schools data on the map
PlotOnStaticMap(MyMap=map,lat=data$LATITUDINE,lon=data$LONGITUDINE,TrueProj=TRUE,
                cex=data$scuola/15,pch=1,col="blue")
```



```
min(data$scuola)
max(data$scuola)
```

```
> min(data$scuola)
[1] 0.2
> max(data$scuola)
[1] 76.9
```

```
# add legend to the map
legend(x=120,y=300,title="Abandonment rate",x.intersp=2,
       y.intersp=c(1.1,1.1,1.1,1.1,1.25,1.4),
       legend=c("< 1%","1%-5%","5%-10%","10-20%","20%-50%","> 50%"),
       col="blue",
       pch=1,cex=0.8,
       pt.cex=c(1/15,5/15,10/15,20/15,50/15,70/15))
```



```
# map crime incidents on a Google map
# http://www.win-vector.com/blog/2012/07/modeling-trick-impact-coding-of-
  categorical-variables-with-many-levels

setwd("c:/R/Rdata")
library(RgoogleMaps)

incidents = read.table("SFPD_Incidents_-_Previous_Three_Months.csv", header=T,
sep=",", as.is=T)
str(incidents)
table(incidents$Category)
```

```
> str(incidents)
'data.frame':         26404 obs. of  12 variables:
 $ IncidntNum: int  146205499 140900108 140859561 146234658 146232997 140879462 140757802 140817036
140821645 140859395 ...
 $ Category  : chr  "NON-CRIMINAL" "VEHICLE THEFT" "BURGLARY" "NON-CRIMINAL" ...
 $ Descript  : chr  "LOST PROPERTY" "STOLEN TRUCK" "BURGLARY, UNLAWFUL ENTRY" "LOST PROPERTY" ...
 $ DayOfWeek : chr  "Thursday" "Thursday" "Saturday" "Saturday" ...
 $ Date      : chr  "10/02/2014" "10/23/2014" "10/11/2014" "10/25/2014" ...
 $ Time      : chr  "13:00" "20:00" "11:24" "11:45" ...
 $ PdDistrict: chr  "SOUTHERN" "BAYVIEW" "SOUTHERN" "SOUTHERN" ...
 $ Resolution: chr  "NONE" "NONE" "ARREST, BOOKED" "NONE" ...
 $ Address   : chr  "900.0 Block of HARRISON ST" "2300.0 Block of SAN BRUNO AV" "800.0 Block of MARKET ST"
"800.0 Block of BRYANT ST" ...
 $ X         : num  -122 -122 -122 -122 -122 ...
 $ Y         : num  37.8 37.7 37.8 37.8 37.7 ...
 $ Location  : chr  "(37.7775833632336, -122.403876552366)" "(37.7335714048034, -122.406306152069)"
"(37.7847532835996, -122.407036790381)" "(37.7752316978411, -122.403742962696)" ...
> table(incidents$Category)
                      ARSON                     ASSAULT                  BAD CHECKS
BURGLARY          DISORDERLY CONDUCT
                         48                        2197                           8
1095                          68
DRIVING UNDER THE INFLUENCE               DRUG/NARCOTIC                 DRUNKENNESS
EMBEZZLEMENT                   EXTORTION
                         80                         997                         124
20                           9
            FAMILY OFFENSES      FORGERY/COUNTERFEITING                       FRAUD
GAMBLING                  KIDNAPPING
                          7                         115                         492
4                          95
              LARCENY/THEFT                 LIQUOR LAWS                   LOITERING              MISSING
PERSON                NON-CRIMINAL
                       7707                          28                          12
874                        3134
             OTHER OFFENSES     PORNOGRAPHY/OBSCENE MAT                PROSTITUTION
ROBBERY                     RUNAWAY
                       3228                           1                          14
723                          46
     SEX OFFENSES, FORCIBLE  SEX OFFENSES, NON FORCIBLE             STOLEN PROPERTY
SUICIDE               SUSPICIOUS OCC
                        121                           2                           4
17                         609
                   TRESPASS                   VANDALISM               VEHICLE THEFT
WARRANTS                 WEAPON LAWS
                        199                        1407                        1450
1248                         221
```

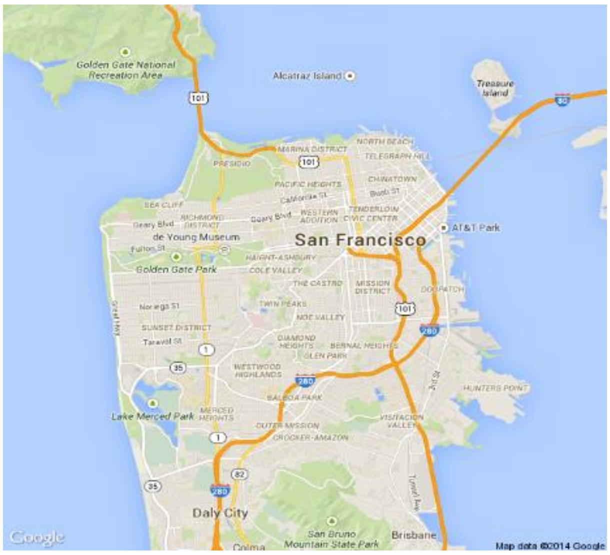
```
# create violent indicator
incidents$violent = with(incidents,
                         Category %in% c("ASSAULT", "ROBBERY",
                                         "SEX OFFENSES, FORCIBLE", "KIDNAPPING")
                         | Descript %in%
                           c("GRAND THEFT PURSESNATCH",
                             "ATTEMPTED GRAND THEFT PURSESNATCH"))
table(incidents$violent)/length(incidents$violent)
```

```
> table(incidents$violent)/length(incidents$violent)

    FALSE      TRUE 
0.8804727 0.1195273 
```

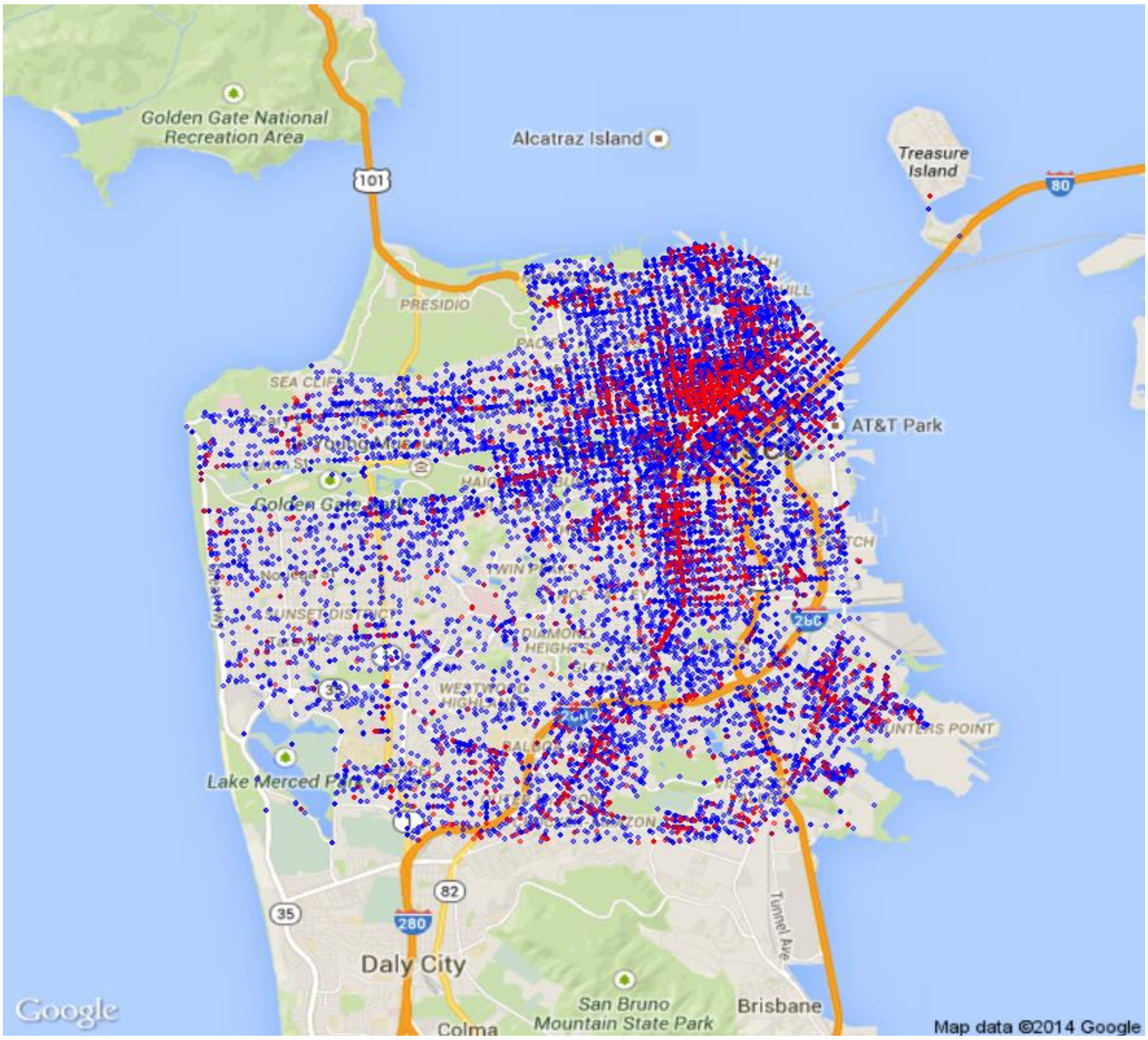
```
bb <- qbbox(lat = incidents$Y, lon = incidents$X)
str(bb)
#download the map:
map = GetMap.bbox(bb$lonR, bb$latR)
```

```
> map = GetMap.bbox(bb$lonR, bb$latR)
[1] "http://maps.google.com/maps/api/staticmap?center=37.7625999631566,-
122.439341185656&zoom=12&size=640x640&maptype=mobile&format=png32&sensor=true"
> PlotOnStaticMap(MyMap=map)
Warning message:
In (function ()  : Only one RStudio graphics device is permitted
```

```
PlotOnStaticMap(MyMap=map)
```



```
tmp = PlotOnStaticMap(map, lat = incidents$Y[!incidents$violent],
                      lon = incidents$X[!incidents$violent],
                      cex=0.75,pch=20,col=rgb(0,0,1,0.5), add=FALSE)
tmp =PlotOnStaticMap(map, lat = incidents$Y[incidents$violent],
                     lon = incidents$X[incidents$violent],
                     cex=0.75,pch=20,col=rgb(1,0,0,0.5), add=TRUE)
```



```
# geocoding locations of aid activities in Uganda

setwd("c:/R/Rdata")
library(RgoogleMaps)

activities <- read.csv("aiddata_UG_geocoding_2014-01-28.csv",header=T,sep=",")
activities <- subset(activities,select=c(status,latitude,longitude))
str(activities)
```

```
> str(activities)
'data.frame':	2458 obs. of  3 variables:
 $ status   : Factor w/ 4 levels "Closed","Committed",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ latitude : num  2.909 1.715 0.316 2.247 2.775 ...
 $ longitude: num  30.9 33.6 32.6 32.3 32.3 ...
```

```
head(activities)
```

```
> head(activities)
     status latitude longitude
1 Committed  2.90906  30.88248
2 Committed  1.71464  33.61113
3 Committed  0.31628  32.58219
4 Committed  2.24710  32.33762
5 Committed  2.77457  32.29899
6 Committed  3.02013  30.91105
```

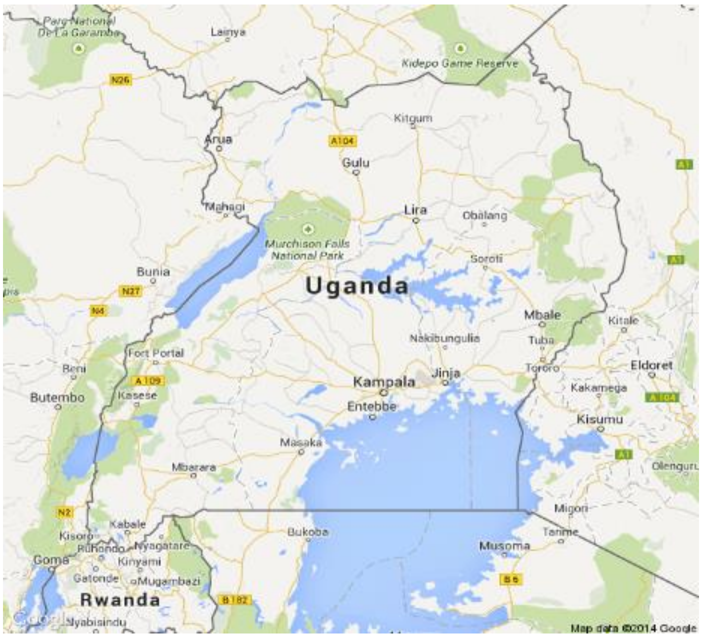
```
table(activities$status)
```

```
> table(activities$status)

   Closed Committed   Ongoing   Planned 
      874        30      1551         3 
```

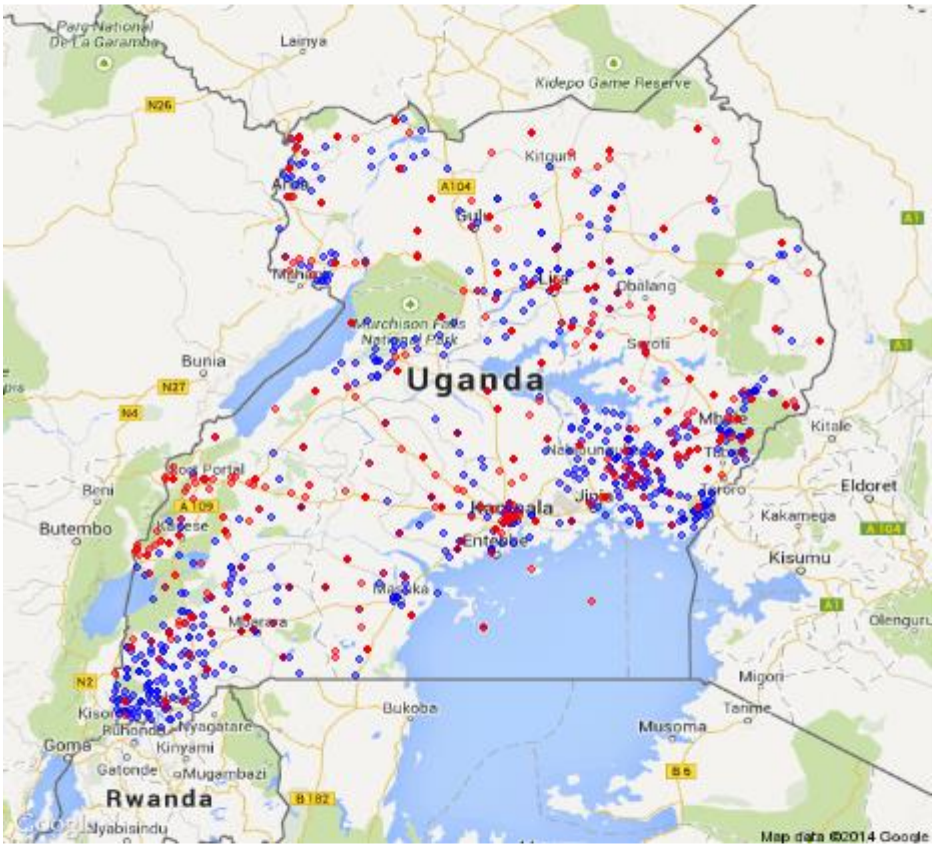
```
library("RgoogleMaps")
bb <- qbbox(lat = activities$latitude, lon = activities$longitude)
str(bb)
```

```
> str(bb)
List of 2
 $ latR: num [1:2] -1.44 3.7
 $ lonR: num [1:2] 29.6 35
```

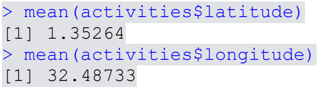
```
# download the map:
map = GetMap.bbox(bb$lonR, bb$latR)
PlotOnStaticMap(MyMap=map)
```



```
tmp = PlotOnStaticMap(map, lat = activities$latitude[activities$status=="Ongoing"],
                      lon = activities$longitude[activities$status=="Ongoing"],
                      cex=0.75,pch=20,col=rgb(0,0,1,0.5))
tmp = PlotOnStaticMap(map, lat = activities$latitude[activities$status=="Closed"],
                      lon = activities$longitude[activities$status=="Closed"],
                      cex=0.75,pch=20,col=rgb(1,0,0,0.5),add=TRUE)
```

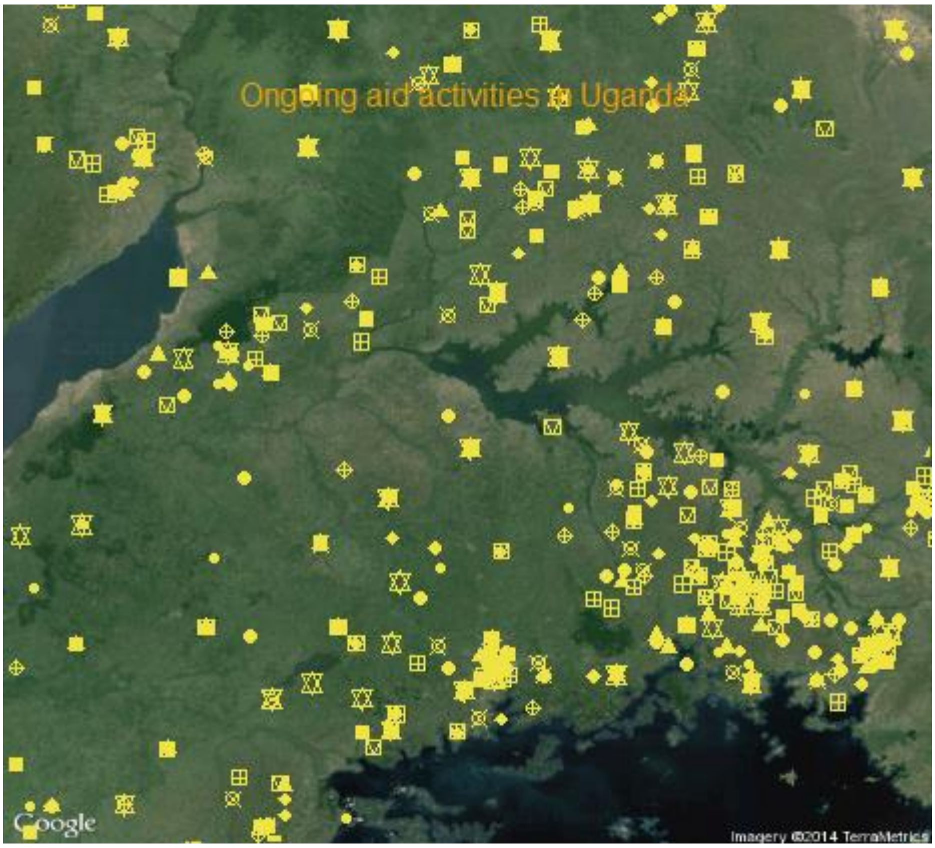


```
# an alternative way of plotting the map :
mean(activities$latitude)
mean(activities$longitude)
```



```
> mean(activities$latitude)
[1] 1.35264
> mean(activities$longitude)
[1] 32.48733
```

```
ongoing <- subset(activities,activities$status=="Ongoing")
map <- GetMap(center=c(lat=1.35264,lon=32.48733),size=c(640,640),
              destfile="uganda.png",zoom=8,maptype="satellite")
PlotOnStaticMap(map,ongoing$latitude,ongoing$longitude,pch=10:20,cex=1,
                col="#F0E442")
text(0,250,labels="Ongoing aid activities in Uganda",col="#E69F00")
```

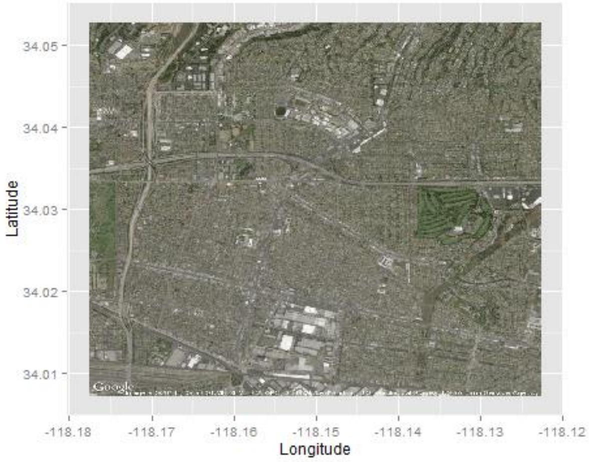


```
# load and plot a Google map in ggplot2

setwd("c:/R/Rdata")
library(RgoogleMaps)
library(jpeg)

# example : Los Angeles

LOCATION              <- 'los_angeles'
CoordinateCenter      <- c(lat = 34.030, lon = -118.150)
COLOR_TYPE            <- c('color','bw')[1]
RGBCoefficients       <- c(0, 1, 0)
ZOOM_LEVEL            <- 14
MAP_TYPE              <- 'satellite'
GOOGLE_MAP            <- 'GoogleMap.jpg'
NUMBER_OF_PIXELS      <- 640

GetMap(center   = CoordinateCenter[c('lat','lon')],
       size     = c(NUMBER_OF_PIXELS, NUMBER_OF_PIXELS),
       zoom     = ZOOM_LEVEL,
       format   = 'jpg',
       maptype  = MAP_TYPE,
       destfile = "test.jpg")
```

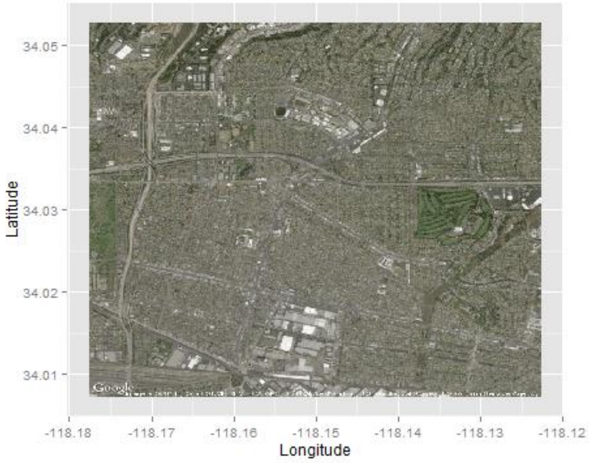
```
[1] "http://maps.google.com/maps/api/staticmap?center=34.03,-
118.15&zoom=14&size=640x640&maptype=satellite&format=jpg&sensor=true"
Error in readPNG(destfile, native = native) : file is not in PNG format
```

```
# load map
map <- readJPEG("test.jpg")
GoogleMapColorMatrix <- readJPEG("test.jpg")
GoogleMapColorMatrix <- apply(GoogleMapColorMatrix, 1:2, function(v) rgb(v[1],
v[2], v[3]))
GoogleMapInformationList <- list(lat= CoordinateCenter['lat'],
                                 lon= CoordinateCenter['lon'],
                                 zoom   = ZOOM_LEVEL,
                                 GoogleMapColorMatrix)
str(GoogleMapInformationList)
class(GoogleMapInformationList)
CoordinateIndex <- (-NUMBER_OF_PIXELS/2) : (NUMBER_OF_PIXELS/2 - 1)
CreateLatitudeValuesFromIndex <- function(x) XY2LatLon(GoogleMapInformationList, -
NUMBER_OF_PIXELS/2, x)[1]
CreateLongitudeValuesFromIndex <- function(y) XY2LatLon(GoogleMapInformationList,
y, -NUMBER_OF_PIXELS/2)[2]
Latitudes <- apply(data.frame(CoordinateIndex), 1, CreateLatitudeValuesFromIndex)
Longitudes <- apply(data.frame(CoordinateIndex), 1, CreateLongitudeValuesFromIndex)
Latitudes <- seq(range(Latitudes)[1], range(Latitudes)[2],
length.out=length(Latitudes))
Longitudes <- seq(range(Longitudes)[1], range(Longitudes)[2],
length.out=length(Longitudes))
library(reshape)
GoogleMapColorDataFrame        <- melt(GoogleMapColorMatrix)
names(GoogleMapColorDataFrame) <- c('x','y','fill')
GoogleMapColorDataFrame <- within(GoogleMapColorDataFrame,{
  x <- x - NUMBER_OF_PIXELS/2 - 1
  y <- y - NUMBER_OF_PIXELS/2 - 1
})
XYCoordinates                <- expand.grid(x = CoordinateIndex, y =
CoordinateIndex)
LatitudesAndLongitudes       <- expand.grid(lat = rev(Latitudes), lon =
Longitudes)
PlotData.Map  <- data.frame(XYCoordinates, LatitudesAndLongitudes)
PlotData.Map  <- suppressMessages(join(PlotData.Map, GoogleMapColorDataFrame, type
= 'right'))
PlotData.Map          <- PlotData.Map[,c('lon','lat','fill')]
LatitudeRange         <- range(PlotData.Map$lat)
LongitudeRange        <- range(PlotData.Map$lon)
str(PlotData.Map)
theme_nothing <- function (base_size = 12){
  structure(list(axis.line            = theme_blank(),
                 axis.text.x          = theme_blank(), axis.text.y = theme_blank(),
                 axis.ticks           = theme_blank(),
                 axis.title.x         = theme_blank(), axis.title.y =
theme_blank(),
                 axis.ticks.length    = unit(0, "lines"), axis.ticks.margin =
unit(0, "lines"),
                 legend.position      = "none",
                 panel.background     = theme_rect(fill = 'white'),
                 panel.border         = theme_blank(),
                 panel.grid.major     = theme_blank(), panel.grid.minor =
theme_blank(),
                 panel.margin         = unit(0, "lines"),
                 plot.background      = theme_rect(colour = 'white'),
                 plot.title           = theme_text(size = base_size * 1.2),
                 plot.margin          = unit(c(-1, -1, -1.5, -1.5), "lines")),
            class = "options")
}
```

```
vplayout <- function(x, y) {
  viewport(layout.pos.row = x, layout.pos.col = y)
}

# plot
library(ggplot2)
qplot(lon,lat,data=PlotData.Map,geom="tile",fill=fill) +
 scale_fill_identity() +
 scale_x_continuous("Longitude") +
 scale_y_continuous("Latitude") +
 coord_equal()
```



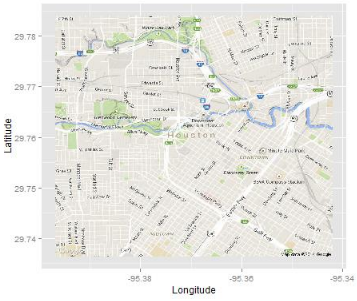
```
ggplot() +
 geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
 scale_fill_identity() +
 scale_x_continuous("Longitude") +
 scale_y_continuous("Latitude") +
 coord_equal()
```



```
# ggplot2 and Googlemaps

# Example : reproduce the Houston Crime Maps
# https://github.com/hadley/ggplot2/wiki/Crime-in-Downtown-Houston,-Texas-:-
  Combining-ggplot2-and-Google-Maps
# https://github.com/mtiernay/Houston_Crime/blob/master/houston_crime.R
# http://cs.smith.edu/classwiki/index.php/Geo_Tracking_Spring_2011_Betsy

setwd("c:/R/Rdata")
library(RgoogleMaps)
library(jpeg)

# input lat-lon coordinates Houston
LOCATION              <- 'Houston'
CoordinateCenter      <- c(lat = 29.760210, lon = -95.369318)
COLOR_TYPE            <- c('color','bw')[1]
RGBCoefficients       <- c(0, 1, 0)
ZOOM_LEVEL            <- 14
MAP_TYPE              <- 'terrain'
GOOGLE_MAP            <- 'GoogleMap.jpg'
NUMBER_OF_PIXELS      <- 640

GetMap(center   = CoordinateCenter[c('lat','lon')],
       size     = c(NUMBER_OF_PIXELS, NUMBER_OF_PIXELS),
       zoom     = ZOOM_LEVEL,
       format   = 'jpg',
       maptype  = MAP_TYPE,
       destfile = "test.jpg")
```

```
[1] "http://maps.google.com/maps/api/staticmap?center=29.76021,-
95.369318&zoom=14&size=640x640&maptype=terrain&format=jpg&sensor=true"
Error in readPNG(destfile, native = native) : file is not in PNG format
```

```
# load map
/ ... /

# plot
library(ggplot2)
qplot(lon,lat,data=PlotData.Map,geom="tile",fill=fill) +
  scale_fill_identity() +
  scale_x_continuous("Longitude") +
  scale_y_continuous("Latitude") +
  coord_equal()
ggplot() +
  geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
  scale_fill_identity() +
  scale_x_continuous("Longitude") +
  scale_y_continuous("Latitude") +
  coord_equal()
```



```
# load data
load("HoustonCrime.Rdata", ex <- new.env())
ls.str(ex)
```

```
> ls.str(ex)
crime_df : 'data.frame':	169042 obs. of  17 variables:
 $ time    : POSIXt, format: "2009-06-01 07:00:00" "2009-06-01 07:00:00" "2009-06-01 07:00:00" "2009-06-01
07:00:00" ...
 $ date    : chr  "6/1/2009" "6/1/2009" "6/1/2009" "6/1/2009" ...
 $ hour    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ premise : chr  "18O" "080" "210" "20A" ...
 $ offense : chr  "Aggravated Assault" "Burglary" "Burglary" "Burglary" ...
 $ beat    : chr  "20G80" "5F30" "1A10" "10H70" ...
 $ block   : chr  "1900-1999" "13200-13299" "1200-1299" "6300-6399" ...
 $ street  : chr  "HWY 6" "NORTHWEST" "FANNIN" "ALMEDA" ...
 $ type    : chr  "" "FWY" "" "RD" ...
 $ suffix  : chr  "S" "-" "-" "-" ...
 $ number  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ month   : Factor w/ 12 levels "January","February",..: 6 6 6 6 6 6 6 6 6 6 ...
 $ day     : Factor w/ 7 levels "Monday","Tuesday",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ location: chr  NA "Department / Discount Store" "Restaurant / Cafeteria" "Apartment" ...
 $ address : chr  "1950 hwy 6" "13250 northwest fwy" "1250 fannin" "6350 almeda rd" ...
 $ lon     : num  -95.6 -95.5 -95.4 -95.4 -95.4 ...
 $ lat     : num  29.7 29.8 29.8 29.7 29.7 ...
```

```
load("HoustonCrime.Rdata")
# restrict the analysis to violent crimes
table(crime_df$offense)
> table(crime_df$offense)
```

```
Aggravated Assault         Auto Theft           Burglary             Murder               Rape            Robbery 
Theft 
             14149              15786              35167                302                781              13200 
89657 
```

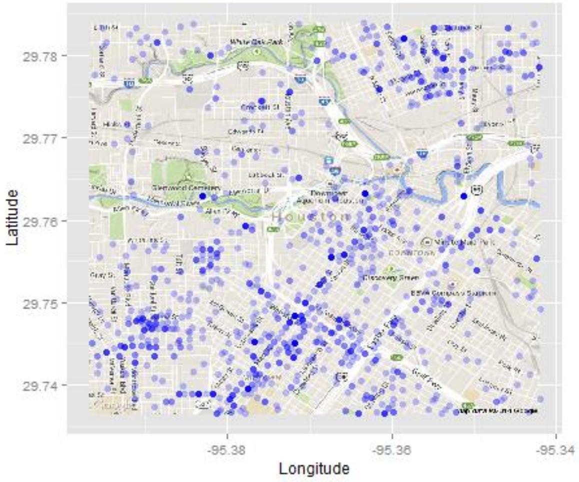
```
violent_crime <- subset(crime_df,offense !="Auto Theft" & offense !="Theft" &
                        offense !="Burglary")
dim(violent_crime)
```

```
> dim(violent_crime)
[1] 28432    17
```

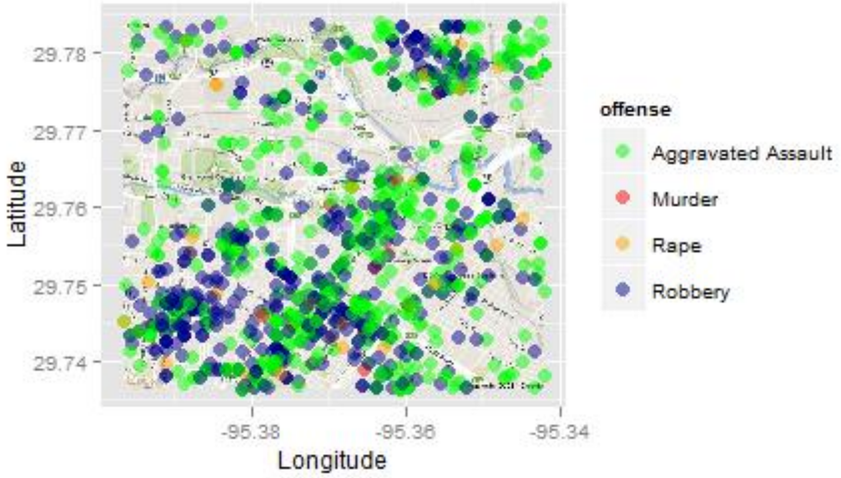
```
# restrict the analysis to records of year 2009
violent_crime <- subset(violent_crime,time >= ISOdatetime(2009,1,1,0,0,0))
# select only necessary variables from dataset violent_crime
# restrict the analysis to lat-lon coordinates of Houston
violent_crime <- subset(violent_crime,select=c(lon,lat,offense))
# restrict the analysis to lat-lon coordinates of Houston
lat_range <- range(PlotData.Map$lat)
lon_range <- range(PlotData.Map$lon)
violent_crime <- subset(violent_crime,lon_range[1] <= lon & lon <= lon_range[2] &
                                      lat_range[1] <= lat & lat <= lat_range[2])
str(violent_crime)
```

```
> str(violent_crime)
'data.frame':	1399 obs. of  3 variables:
 $ lon    : num  -95.4 -95.4 -95.4 -95.4 -95.4 ...
 $ lat    : num  29.7 29.8 29.7 29.8 29.8 ...
 $ offense: chr  "Aggravated Assault" "Rape" "Aggravated Assault" "Aggravated Assault" ...
```

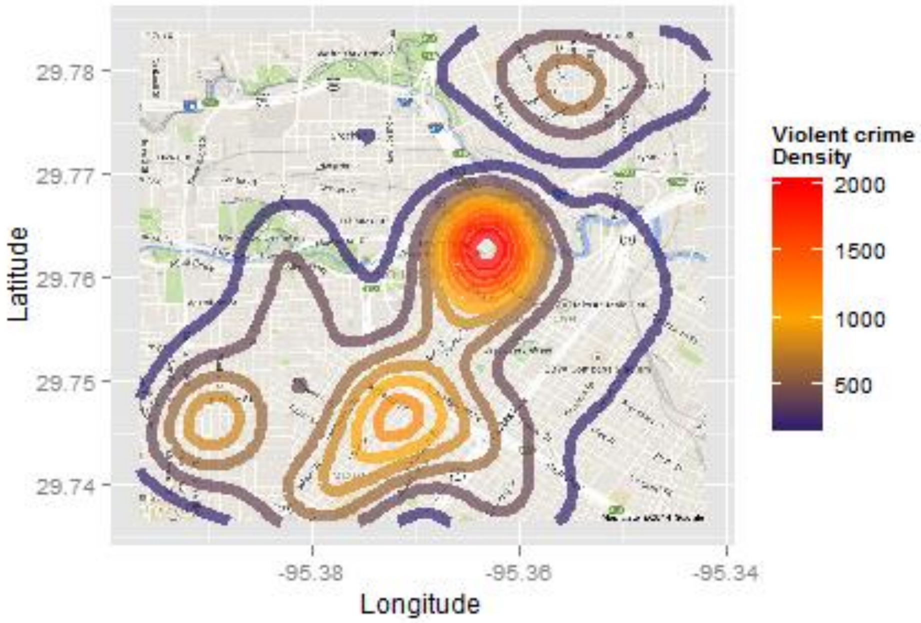
```
# point map
pointmap <- ggplot() +
            geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
           geom_point(aes(x=lon,y=lat),colour="blue",size=2.5,alpha=1/4,
                      data=violent_crime) +
            scale_x_continuous("Longitude",limits=lon_range) +
            scale_y_continuous("Latitude",limits=lat_range) +
            scale_fill_identity() +
            coord_equal()
pointmap
```



```
jitter <- ggplot() +
          geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
          scale_fill_identity() +
          geom_jitter(aes(x=lon,y=lat,colour=offense),size=3.5,alpha=0.5,
                      data=violent_crime) +
          scale_x_continuous("Longitude",limits=lon_range) +
          scale_y_continuous("Latitude",limits=lat_range) +
          scale_colour_manual(values=c("green","red","orange","darkblue")) +
          coord_equal()
jitter
```

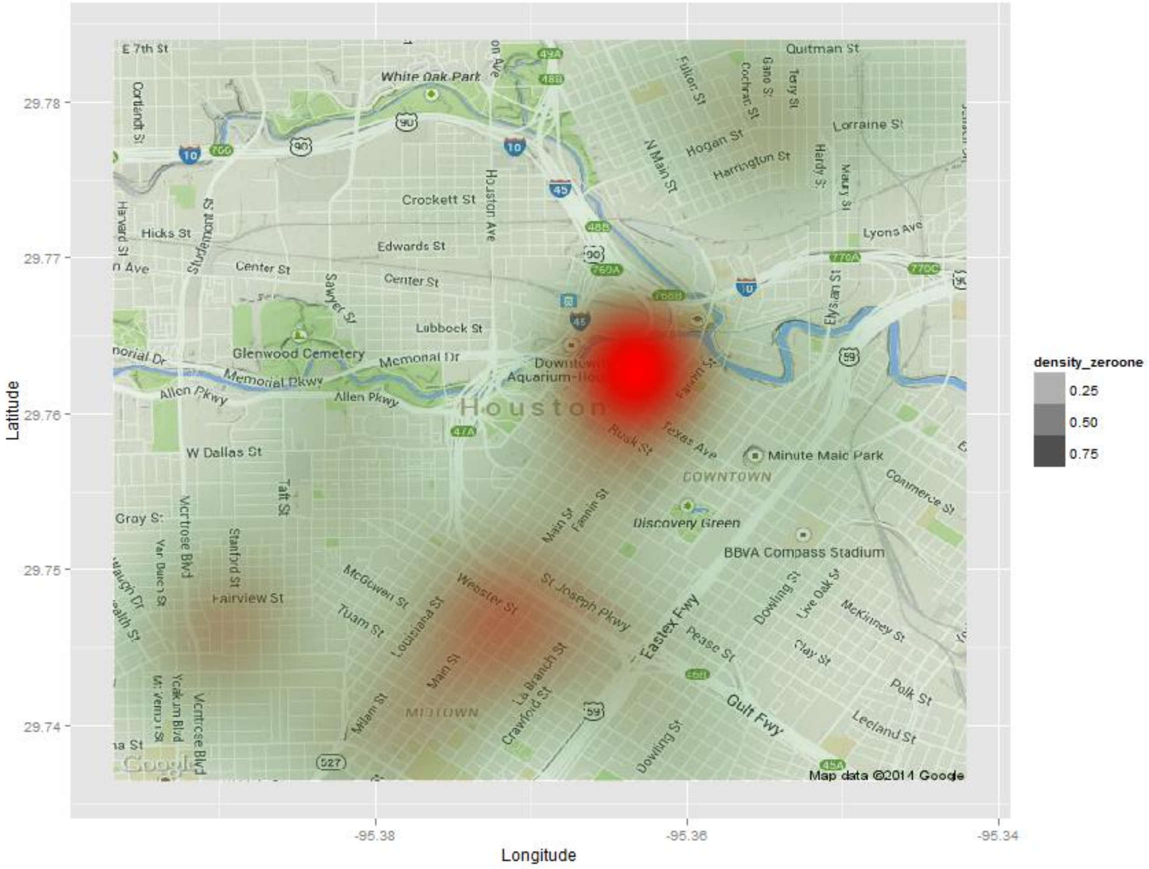


```
# contour plot
contour <- ggplot() +
           geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
           geom_density2d(aes(x=lon,y=lat,colour=..level..),size=2,alpha=0.7,
                          data=violent_crime) +
           scale_colour_gradient2("Violent Crime\nDensity",low="darkblue",
                                  mid="orange",high="red",midpoint=1000) +
           scale_x_continuous("Longitude",limits=lon_range) +
           scale_y_continuous("Latitude",limits=lat_range) +
           scale_fill_identity() +
           coord_equal()
contour
```



```
# calculate density (hotspots)
library(MASS)
density <-
kde2d(violent_crime$lon,violent_crime$lat,n=320,lims=c(lon_range,lat_range))
str(density)
kde_df <- expand.grid(lon=seq.int(lon_range[1],lon_range[2],length.out=320),
                      lat=seq.int(lat_range[1],lat_range[2],length.out=320))
kde_df$density <- melt(density$z)$value
summary(kde_df$density)
den_fill_scale <- scale_colour_gradient2(low="white",mid="darkgreen",high="red",
                                         midpoint=median(kde_df$density))
den_fill_scale$range$train(kde_df$density)
kde_df$density_s <- ggplot2:::scale_map(den_fill_scale,kde_df$density)
kde_df$density_zeroone <- pmin(kde_df$density / max(kde_df$density), .9)
```

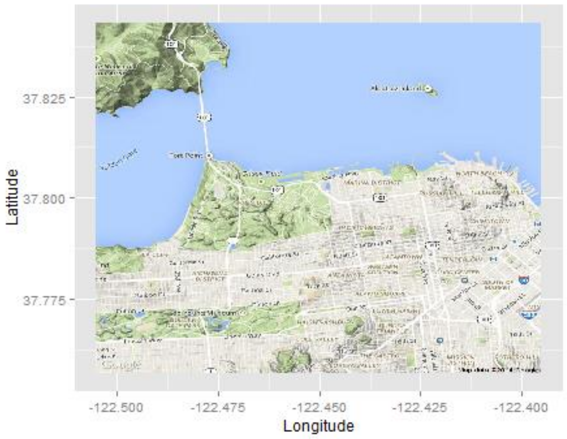
```
# heatmap (hotspots)
heatmap <- ggplot() +
           geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
           geom_tile(aes(x=lon,y=lat,fill=density_s,alpha=density_zeroone),
                     data=kde_df) +
           scale_x_continuous("Longitude",limits=lon_range) +
           scale_y_continuous("Latitude",limits=lat_range) +
           scale_fill_identity() +
           coord_equal()
heatmap
```



```
# create heatmap for crime incidents San Francisco

library(RgoogleMaps)
library(jpeg)

# input lat-lon coordinates San Fransisco
LOCATION              <- 'San Fransisco'
CoordinateCenter      <- c(lat = 37.8, lon = -122.45)
COLOR_TYPE            <- c('color','bw')[1]
RGBCoefficients       <- c(0, 1, 0)
ZOOM_LEVEL            <- 13
MAP_TYPE              <- 'terrain'
GOOGLE_MAP            <- 'GoogleMap.jpg'
NUMBER_OF_PIXELS      <- 640

GetMap(center   = CoordinateCenter[c('lat','lon')],
       size     = c(NUMBER_OF_PIXELS, NUMBER_OF_PIXELS),
       zoom     = ZOOM_LEVEL,
       format   = 'jpg',
       maptype  = MAP_TYPE,
       destfile = "test.jpg")
```

```
[1] "http://maps.google.com/maps/api/staticmap?center=37.8,-
122.45&zoom=13&size=640x640&maptype=terrain&format=jpg&sensor=true"
Error in readPNG(destfile, native = native) : file is not in PNG format
```

```
# load map
/ ... /

# plot
library(ggplot2)
qplot(lon,lat,data=PlotData.Map,geom="tile",fill=fill) +
  scale_fill_identity() +
  scale_x_continuous("Longitude") +
  scale_y_continuous("Latitude") +
  coord_equal()
ggplot() +
  geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
  scale_fill_identity() +
  scale_x_continuous("Longitude") +
  scale_y_continuous("Latitude") +
  coord_equal()
```



```
# http://www.win-vector.com/blog/2012/07/modeling-trick-impact-coding-of-
  categorical-variables-with-many-levels

setwd("c:/R/Rdata")
incidents = read.table("SFPD_Incidents_-_Previous_Three_Months.csv", header=T,
                       sep=",", as.is=T)

# create violent indicator
incidents$violent = with(incidents,
                         Category %in% c("ASSAULT", "ROBBERY",
                                         "SEX OFFENSES, FORCIBLE", "KIDNAPPING")
                         | Descript %in%
                           c("GRAND THEFT PURSESNATCH",
                             "ATTEMPTED GRAND THEFT PURSESNATCH"))
table(incidents$violent)/length(incidents$violent)
```

```
> table(incidents$violent)/length(incidents$violent)

    FALSE      TRUE 
0.8804727 0.1195273 
```

```
str(incidents)
```

```
> str(incidents)
'data.frame':	26404 obs. of  13 variables:
 $ IncidntNum: int  146205499 140900108 140859561 146234658 146232997 140879462 140757802 140817036 
140821645 140859395 ...
 $ Category  : chr  "NON-CRIMINAL" "VEHICLE THEFT" "BURGLARY" "NON-CRIMINAL" ...
 $ Descript  : chr  "LOST PROPERTY" "STOLEN TRUCK" "BURGLARY, UNLAWFUL ENTRY" "LOST PROPERTY" ...
 $ DayOfWeek : chr  "Thursday" "Thursday" "Saturday" "Saturday" ...
 $ Date      : chr  "10/02/2014" "10/23/2014" "10/11/2014" "10/25/2014" ...
 $ Time      : chr  "13:00" "20:00" "11:24" "11:45" ...
 $ PdDistrict: chr  "SOUTHERN" "BAYVIEW" "SOUTHERN" "SOUTHERN" ...
 $ Resolution: chr  "NONE" "NONE" "ARREST, BOOKED" "NONE" ...
 $ Address   : chr  "900.0 Block of HARRISON ST" "2300.0 Block of SAN BRUNO AV" "800.0 Block of MARKET ST" 
"800.0 Block of BRYANT ST" ...
 $ X         : num  -122 -122 -122 -122 -122 ...
 $ Y         : num  37.8 37.7 37.8 37.8 37.7 ...
 $ Location  : chr  "(37.7775833632336, -122.403876552366)" "(37.7335714048034, -122.406306152069)" 
"(37.7847532835996, -122.407036790381)" "(37.7752316978411, -122.403742962696)" ...
 $ violent   : logi  FALSE FALSE FALSE FALSE FALSE TRUE ...
```

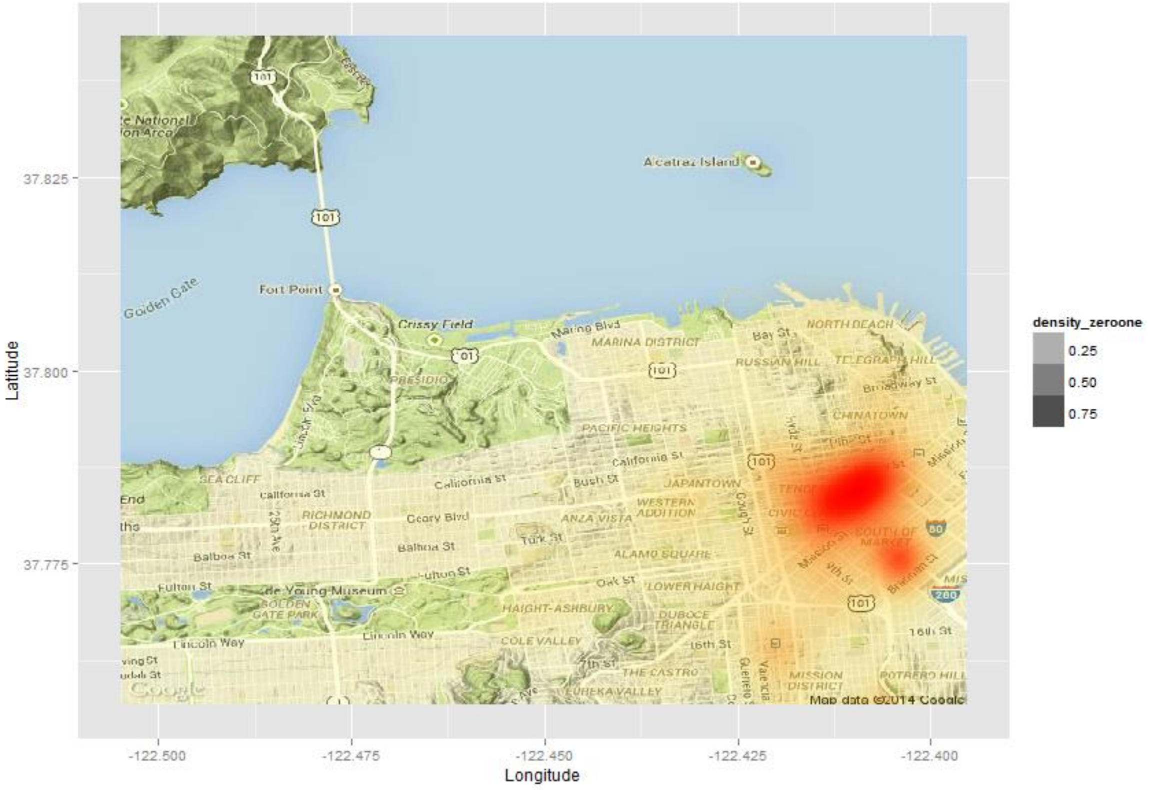
```
violent_incidents <- subset(incidents,violent=="TRUE",select=c(X,Y))
violent_incidents <- na.omit(violent_incidents)
colnames(violent_incidents) <- c("lon","lat")
str(violent_incidents)
```

```
> str(violent_incidents)
'data.frame':	3156 obs. of  2 variables:
 $ lon: num  -122 -122 -122 -122 -122 ...
 $ lat: num  37.8 37.8 37.7 37.7 37.8 ...
```

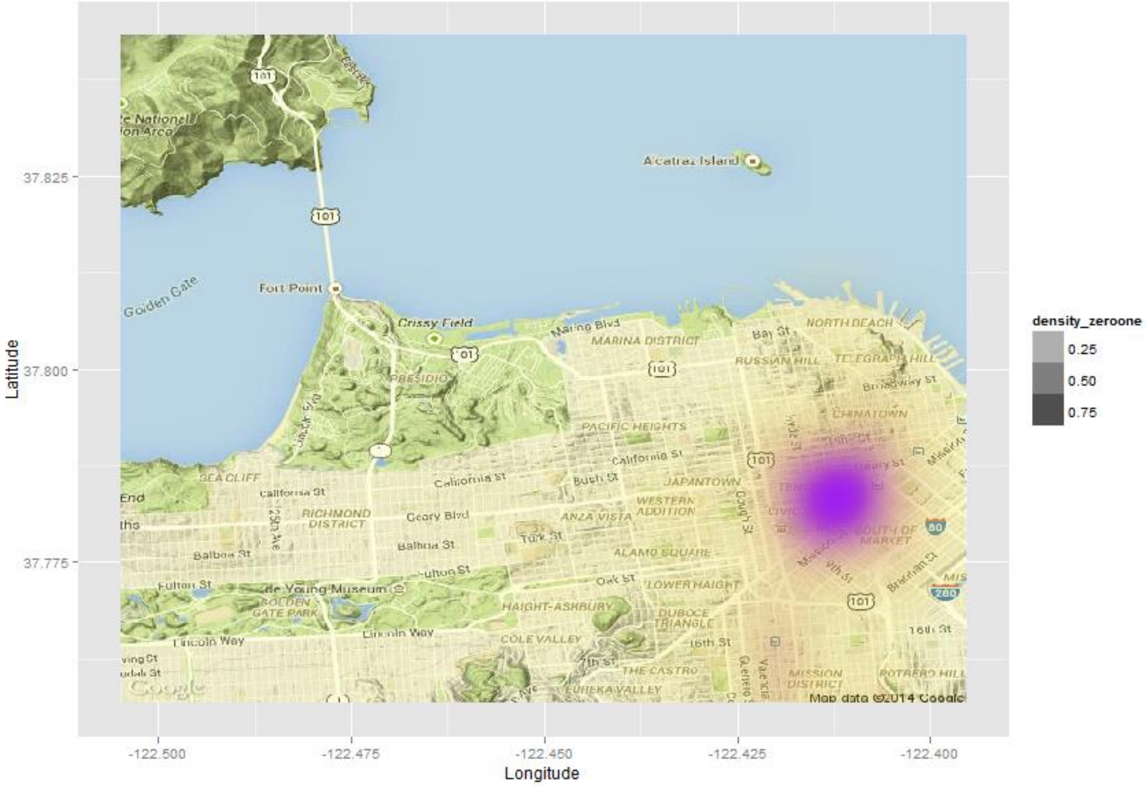
```
nonviolent_incidents <- subset(incidents,violent=="FALSE",select=c(X,Y))
nonviolent_incidents <- na.omit(nonviolent_incidents)
colnames(nonviolent_incidents) <- c("lon","lat")
str(nonviolent_incidents)
```

```
> str(nonviolent_incidents)
'data.frame':	23248 obs. of  2 variables:
 $ lon: num  -122 -122 -122 -122 -122 ...
 $ lat: num  37.8 37.7 37.8 37.8 37.7 ...
```

```
lat_range <- range(PlotData.Map$lat)
lon_range <- range(PlotData.Map$lon)

# calculate density (hotspots) for nonviolent incidents
library(MASS)
density <- kde2d(nonviolent_incidents$lon,nonviolent_incidents$lat,n=320,
                 lims=c(lon_range,lat_range))
str(density)
kde_df <- expand.grid(lon=seq.int(lon_range[1],lon_range[2],length.out=320),
                      lat=seq.int(lat_range[1],lat_range[2],length.out=320))
kde_df$density <- melt(density$z)$value
summary(kde_df$density)
den_fill_scale <- scale_colour_gradient2(low="white",mid="yellow",high="red",
                                         midpoint=median(kde_df$density))
den_fill_scale$range$train(kde_df$density)
kde_df$density_s <- ggplot2:::scale_map(den_fill_scale,kde_df$density)
kde_df$density_zeroone <- pmin(kde_df$density / max(kde_df$density), .9)

# heatmap (hotspots) for nonviolent incidents
library(ggplot2)
heatmap <- ggplot() +
  geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
  geom_tile(aes(x=lon,y=lat,fill=density_s,alpha=density_zeroone),data=kde_df) +
  scale_x_continuous("Longitude",limits=lon_range) +
  scale_y_continuous("Latitude",limits=lat_range) +
  scale_fill_identity() +
  coord_equal()
heatmap
```



```
# calculate density (hotspots) for violent incidents
library(MASS)
density <- kde2d(violent_incidents$lon,violent_incidents$lat,n=320,
                 lims=c(lon_range,lat_range))
str(density)
kde_df <- expand.grid(lon=seq.int(lon_range[1],lon_range[2],length.out=320),
                      lat=seq.int(lat_range[1],lat_range[2],length.out=320))
kde_df$density <- melt(density$z)$value
summary(kde_df$density)
den_fill_scale <- scale_colour_gradient2(low="white",mid="yellow",high="purple",
                                         midpoint=median(kde_df$density))
den_fill_scale$range$train(kde_df$density)
kde_df$density_s <- ggplot2:::scale_map(den_fill_scale,kde_df$density)
kde_df$density_zeroone <- pmin(kde_df$density / max(kde_df$density), .9)

# heatmap (hotspots) for violent incidents
heatmap <- ggplot() +
  geom_tile(aes(x=lon,y=lat,fill=fill),data=PlotData.Map) +
  geom_tile(aes(x=lon,y=lat,fill=density_s,alpha=density_zeroone),data=kde_df) +
  scale_x_continuous("Longitude",limits=lon_range) +
  scale_y_continuous("Latitude",limits=lat_range) +
  scale_fill_identity() +
  coord_equal()
heatmap
```


Latitude
37.825
37.800
37.775
Longitude
-122.500
-122.475
-122.450
-122.425
-122.400
density_zeroone
0.25
0.50
0.75
Alcatraz Island
Fort Point
Golden Gate
Crissy Field
Marina Blvd
MARINA DISTRICT
PRESIDIO
NORTH BEACH
RUSSIAN HILL
TELEGRAPH HILL
CHINATOWN
PACIFIC HEIGHTS
JAPANTOWN
WESTERN ADDITION
ANZA VISTA
SEA CLIFF
RICHMOND DISTRICT
Geary Blvd
California St
Balboa St
Fulton St
de Young Museum
GOLDEN GATE PARK
Lincoln Way
ALAMO SQUARE
LOWER HAIGHT
HAIGHT-ASHBURY
DUBOCE TRIANGLE
COLE VALLEY
THE CASTRO
EUREKA VALLEY
SOUTH OF MARKET
MISSION DISTRICT
POTRERO HILL
16th St
101
80
280
Map data ©2014 Google