# Spatial maps and geocoding in R

Install packages

install.packages("mapproj")  
install.packages("ggmap")  
install.packages("DeducerSpatial")

load packages

require(maps)  
require(ggmap)

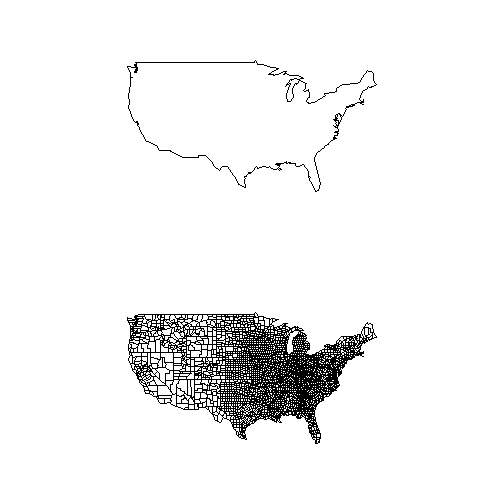
# Using the maps package

par(mfrow = c(2, 1))  
map("usa")



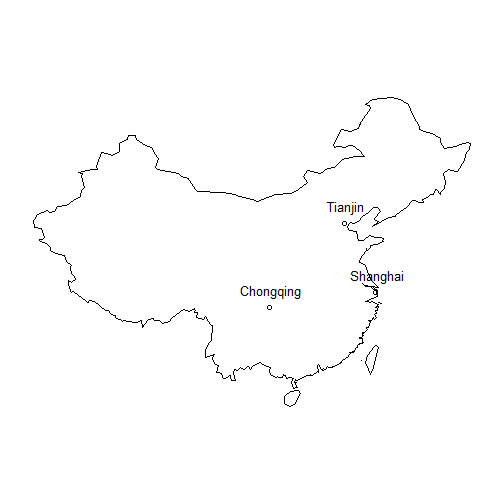
plot of chunk USA

map("county")



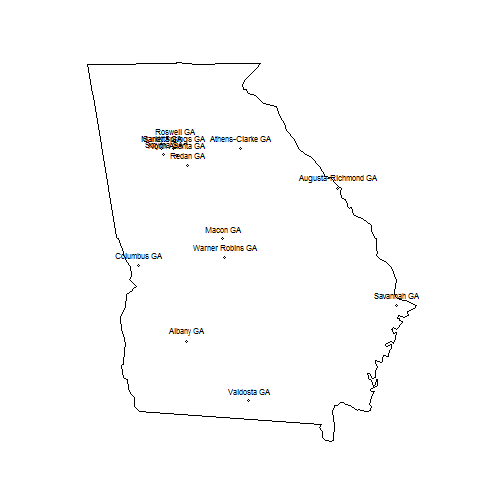
plot of chunk USA

map("world", "China")  
map.cities(country = "China", capitals = 2)



plot of chunk china

map("state", "GEORGIA")  
data(us.cities)  
map.cities(us.cities, country = "GA")



plot of chunk states

Plot the unemployment in each county

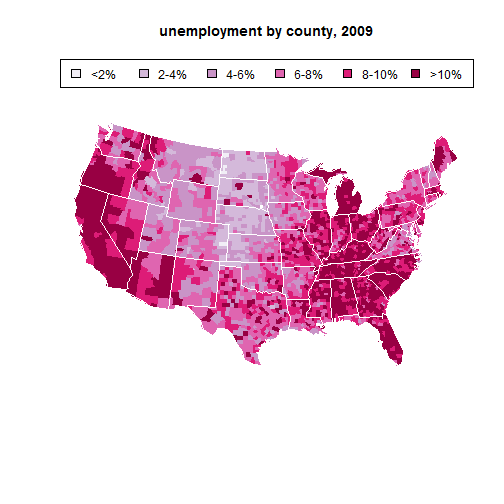
data(unemp)  
data(county.fips)  
  
# Plot unemployment by country  
colors = c("#F1EEF6", "#D4B9DA", "#C994C7", "#DF65B0", "#DD1C77",   
 "#980043")  
head(unemp)

## fips pop unemp  
## 1 1001 23288 9.7  
## 2 1003 81706 9.1  
## 3 1005 9703 13.4  
## 4 1007 8475 12.1  
## 5 1009 25306 9.9  
## 6 1011 3527 16.4

head(county.fips)

## fips polyname  
## 1 1001 alabama,autauga  
## 2 1003 alabama,baldwin  
## 3 1005 alabama,barbour  
## 4 1007 alabama,bibb  
## 5 1009 alabama,blount  
## 6 1011 alabama,bullock

unemp$colorBuckets <- as.numeric(cut(unemp$unemp, c(0, 2, 4, 6, 8,   
 10, 100)))  
colorsmatched <- unemp$colorBuckets[match(county.fips$fips, unemp$fips)]  
  
map("county", col = colors[colorsmatched], fill = TRUE, resolution = 0,   
 lty = 0, projection = "polyconic")  
  
# Add border around each State  
map("state", col = "white", fill = FALSE, add = TRUE, lty = 1, lwd = 0.2,   
 projection = "polyconic")  
title("unemployment by county, 2009")  
  
leg.txt <- c("<2%", "2-4%", "4-6%", "6-8%", "8-10%", ">10%")  
legend("topright", leg.txt, horiz = TRUE, fill = colors)



plot of chunk jobs

# Using OpenStreetMaps

OpenStreetMap is a new package that accesses raster open street maps from Mapnik, and satellite imagery from Bing.

Some features: - Uses multiple map tiles stitched together to create high quality images. - No files are created or stored on the hard drive. - Tiles are cached, so downloads occur only when necessary. - ggplot 0.9.0 integration

More information is available at: http://blog.fellstat.com/?p=130

# Deducer

DeducerSpatial is a package for spatial data analysis which includes the ability to plot and explore open street map and Bing satellite images.

```{Deducer} library(UScensus2000)

lat <- c(43.834526782236814,30.334953881988564) lon <- c(-131.0888671875 ,-107.8857421875) southwest <- openmap(c(lat[1],lon[1]),c(lat[2],lon[2]),5,'bing') data(california.tract) california.tract <- spTransform(california.tract,osm())

plot(southwest,removeMargin=TRUE) choro\_plot(california.tract,dem = slot(california.tract,"data")[,'med.age'], legend.title = 'Median Age',alpha=1)

Using the ggmap package  
===================================  
Maps are extracted from Google Maps, OpenStreetMap, or Stamen Maps server for a map. You can query the Google Maps, OpenStreetMap, or Stamen Maps server for a map at a certain location at a certain spatial zoom.   
  
geocode  
=========  
The geocode function will extract the position (latitude and longtitude) of a location using Google Maps  
  
```{geocode}  
> geocode('CDC')  
 lon lat  
1 -84.3258 33.7988  
> geocode('Baylor University')  
 lon lat  
1 -97.11332 31.54641  
> geocode('the white house', messaging = TRUE)  
contacting http://maps.googleapis.com/maps/api/geocode/json?address=the+white+house&sensor=false... done.  
 lon lat  
1 -77.03282 38.89521  
> geocode(c('baylor university', 'CDC'), output = 'latlona')  
 lon lat  
1 -97.11332 31.54641  
2 -84.32580 33.79880  
 address  
1 baylor university, 1311 s 5th st, waco, tx 76706, usa  
2 centers for disease control and prevention, 1600 clifton rd ne, atlanta, ga 30329, usa  
> geocode(c('harvard university', 'the vatican'), output = 'more')  
 lon lat type loctype  
1 -71.11847 42.37315 point\_of\_interest approximate  
2 12.45813 41.90226 locality approximate  
 address  
1 harvard university housing office, 1350 massachusetts ave, cambridge, ma 02138, usa  
2 00120 vatican city  
 north south east west postal\_code country  
1 42.38139 42.36490 -71.10246 -71.13447 02138 united states  
2 41.90747 41.73199 12.66542 12.44584 00120 vatican city  
 administrative\_area\_level\_2 administrative\_area\_level\_1 locality  
1 middlesex massachusetts cambridge  
2 <NA> <NA> vatican city  
 street streetNo point\_of\_interest  
1 massachusetts ave 1350 harvard university housing office  
2 <NA> NA <NA>

# Exercise

```{Exercise} Get the geocode for the eiffel tower. Is there a unique map?

geocode('the eiffel tower', output = 'all') ```

# mapdist

{mapdist} mapdist(from, to, mode = c("driving", "walking", "bicycling"), output = c("simple", "all"), messaging = FALSE, sensor = FALSE, language = "en-EN", override\_limit = FALSE)

Example, how far is it to walk from the CDC to the white house and map the route.

```in{mapdistCDC} > mapdist('CDC', 'the white house', mode = 'walking')

from to m km miles seconds minutes

1 the white house CDC 1019454 1019.454 633.4887 731359 12189.32 hours 1 203.1553 ```

Google Geocoding API is subject to a query limit of 2,500 geolocation requests per day

{GoogleCheck, eval=FALSE} geocodeQueryCheck()

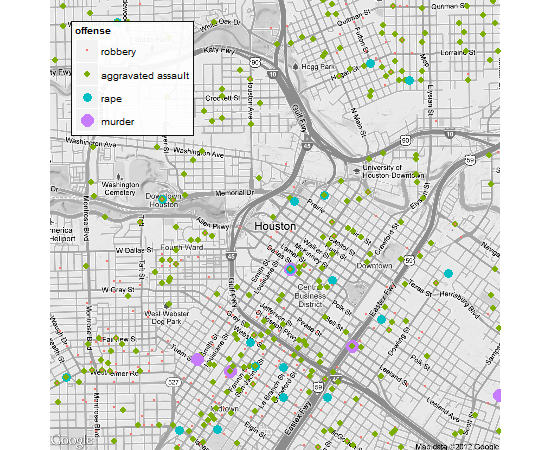
# Study of crimes in Houston

Extract location of crimes in houston

violent\_crimes <- subset(crime, offense != "auto theft" & offense !=   
 "theft" & offense != "burglary")  
  
# rank violent crimes  
violent\_crimes$offense <- factor(violent\_crimes$offense, levels = c("robbery",   
 "aggravated assault", "rape", "murder"))  
  
# restrict to downtown  
violent\_crimes <- subset(violent\_crimes, -95.39681 <= lon & lon <=   
 -95.34188 & 29.73631 <= lat & lat <= 29.784)

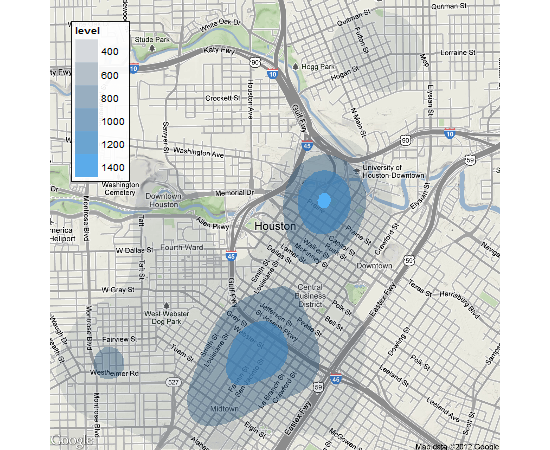
Map these crimes on the map of the city

{crimes, eval=FALSE} HoustonMap <- qmap('houston', zoom = 14,color = 'bw', legend = 'topleft') HoustonMap +geom\_point(aes(x = lon, y = lat, size = offense,colour = offense), data = violent\_crimes )

 Results of qmap using ggmap of crimes in houston

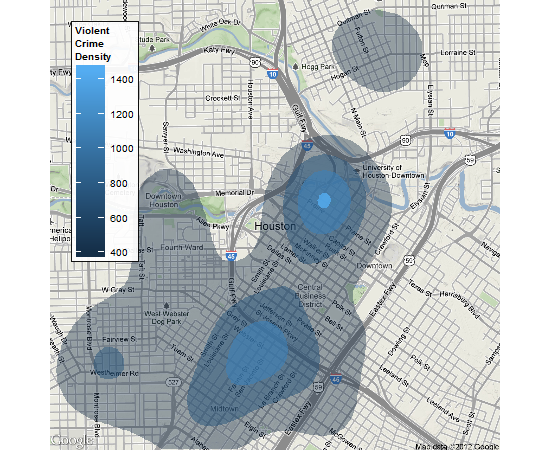
Plot again but use stats$\_$denisty layer ```{crimes2, eval=FALSE} houston <- get\_map('houston', zoom = 14) HoustonMap <- ggmap(houston, extent = 'device', legend = 'topleft')

HoustonMap + stat\_density2d(aes(x = lon, y = lat, fill = ..level.. , alpha = ..level..),size = 2, bins = 4, data = violent\_crimes, geom = 'polygon') scale\_fill\_gradient('Violent') + scale\_alpha(range = c(.4, .75), guide = FALSE) + guides(fill = guide\_colorbar(barwidth = 1.5, barheight = 10)) ```

 Results of qmap using ggmap of crimes in houston

Add the colorbar guide to the key

{crimes3,eval=FALSE} HoustonMap + stat\_density2d(aes(x = lon, y = lat, fill = ..level.., alpha = ..level..), size = 2, bins = 4, data = violent\_crimes, geom = 'polygon') +scale\_fill\_gradient('Violent\nCrime\nDensity') +scale\_alpha(range = c(.4, .75), guide = FALSE) +guides(fill = guide\_colorbar(barwidth = 1.5, barheight = 10))

 Results of qmap using ggmap of crimes in houston

sessionInfo()

## R version 2.15.1 (2012-06-22)  
## Platform: i386-pc-mingw32/i386 (32-bit)  
##   
## locale:  
## [1] LC\_COLLATE=English\_United States.1252   
## [2] LC\_CTYPE=English\_United States.1252   
## [3] LC\_MONETARY=English\_United States.1252  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United States.1252   
##   
## attached base packages:  
## [1] grid splines stats graphics grDevices utils datasets   
## [8] methods base   
##   
## other attached packages:  
## [1] knitr\_0.6.3 DeducerSpatial\_0.4 OpenStreetMap\_0.2   
## [4] raster\_2.0-05 maptools\_0.8-14 sp\_0.9-99   
## [7] Deducer\_0.6-3 plyr\_1.7.1 foreign\_0.8-50   
## [10] effects\_2.1-1 colorspace\_1.1-1 lattice\_0.20-6   
## [13] multcomp\_1.2-12 survival\_2.36-14 mvtnorm\_0.9-9992   
## [16] car\_2.0-12 nnet\_7.3-1 MASS\_7.3-18   
## [19] JGR\_1.7-9 iplots\_1.1-4 JavaGD\_0.5-5   
## [22] scales\_0.2.1 rJava\_0.9-3 mapproj\_1.1-8.3   
## [25] maps\_2.2-6 ggmap\_2.1 ggplot2\_0.9.1   
##   
## loaded via a namespace (and not attached):  
## [1] dichromat\_1.2-4 digest\_0.5.2 evaluate\_0.4.2   
## [4] formatR\_0.5 labeling\_0.1 memoise\_0.1   
## [7] munsell\_0.3 parser\_0.0-15 png\_0.1-4   
## [10] proto\_0.3-9.2 RColorBrewer\_1.0-5 Rcpp\_0.9.11   
## [13] reshape2\_1.2.1 RgoogleMaps\_1.2.0 rjson\_0.2.8   
## [16] stringr\_0.6 tools\_2.15.1