STA 141A
Fundamentals of Statistical Data Science

Fall 2016

Instructor: Debashis Paul

Lecture 11

Working with maps

- Maps are very useful for representing spatial information and data that are spatially referenced, such as population centers, important geographic landmarks, historical places of interest, locations and outcomes of natural calamities, international trade, political events, and many more types of data.
- There are many different packages in R that can be used to access and represent maps graphically. We cover two different packages: an older one named **maps** and a more recent package named **ggmap**. Both packages have interfaces with **ggplot2** package. Indeed, **ggmap** package is built using the same grammar of graphics as **ggplot2**.
- Other important packages for dealing with spatial data include **fields**, **gstat**, **rgdal**, **sp**, **spatia**l, etc. A nice overview of packages used for accessing (reading/writing), visualizing and analyzing different kinds of spatial data can be found in the following link:

https://cran.r-project.org/web/views/Spatial.html

maps package and interface with **ggplot2**

- **maps** package is useful for drawing simple line maps indicating boundaries of countries, states (within US) and counties (within US). It also contains data on various aspects of major US cities.

```
library(maps)data(us.cities) # data on major US cities
big.cities = subset(us.cities, pop > 500000)  # choose cities with population at least 0.5 million
city.loc = ggplot(big.cities,aes(long,lat)) # borders of states using borders( ) function in ggplot2
city.loc = city.loc + borders("state",size=0.5,col="green") # plot cities with size determined by population
city.loc + geom_point(aes(size=pop,color=pop),alpha=I(1/3)) # add the names of the cities as labels
city.loc + geom_point(aes(size=pop,color=pop),alpha=I(1/3)) + geom_text(aes(label=name),size=2)
ca.state = map_data("county","california")  # data on counties in California
# draw the map of each county from the points indicating the boundary of the counties
ggplot(ca.state,aes(long,lat)) + geom_polygon(aes(group=group), fill=NA, col="blue")
```

ggmap package

- **ggmap** uses the layered grammar of graphics in the same way as **ggplot2**, and can be used to display maps, together with contextual information.
- Maps in **ggmap** are accessed from online service providers such as Google Maps or OpenStreetMap.
- **Reference:** Kahle, D. and Wickham, H. (2013). ggmap : Spatial visualization with ggplot2. *R Journal*, Vol. 5, No. 1.

```
library(ggmap)
data(crime) # data frame on incidences of crime in Houston area (part of ggmap package)
murder = subset(crime, offense == "murder") # extract the part of data frame corresponding to incidences of murder
qmplot(lon, lat, data=murder, color = I("red"), size=I(2), darken=0.3, zoom = 11) # plot the locations of murders
# distinguish the locations by the weekday of occurrence of murder as well as number of murders
qmplot(lon, lat, data=murder, color = factor(day), size=factor(number), darken=0.3, zoom = 11)
```

Various operations on maps

- We can search the location of a landmark or place using geocode( )

```
geocode("Davis",output="all") # gives a list, we can use argument “more” instead, returning less information
```

- We can find physical address for a landmark or place using revgeocode( )

```
revgeocode(as.numeric(geocode("white house")))
```

- We can compute the map distance between pairs of locations using mapdist( )

```
from.loc = c("davis","davis","berkeley");  to.loc = c("berkeley","san francisco","santa cruz")
mapdist(from.loc, to.loc, mode = "driving")  # or "walking" or "bicycling“
```

- We finding the route between pairs of locations using route( )

```
from.loc = c("davis, california");  to.loc = c("berkeley, california")
route_df  = route(from.loc, to.loc, mode=“driving", structure = "route") # returns a data frame containing route
```

Use of qmap( ) and get_map( )

- qmap( ) function is analogous to ggplot( ) function in **ggplot2**

```
qmap("Davis", zoom =12) # plot map of Davis using Google Maps
qmap("Davis", zoom =12, source = "osm") # using OpenStreetMap
```

- We can use get_map( ) function to obtain the map of a city or state

```
SFO = get_map("san francisco",zoom=12, maptype="roadmap")  # default is “terrain”
# a ggmap object (a raster image) is created, but nothing plotted
ggmap(SFO)  # plot the map of San Francisco
SFO = get_map("san francisco",zoom=12,maptype="satellite") # creates a satellite map
ggmap(SFO)
```

Using layered graphics with **ggmap** : Houston crime data

- We use the crime data in Houston metropolitan area to demonstrate some features of ggmap. Especially, we can use density plots, subdivided by type of crime and day of the week when a crime is committed to get an idea about the pattern of crime by type, spatial location and seasonal (weekly) effects.

```
houston = get_map("houston, texas", zoom = 11, maptype = "terrain")
ggmap(houston)
# overlay with incidences of crime (together with number and type of offense)
ggmap(houston) + geom_point(aes(color=factor(day)),data=crime,size=0.2,alpha=I(1/5)) + facet_wrap(~offense)
# draw contour plot of the kernel density estimate of incidences of crime
ggmap(houston) + stat_density2d(aes(x=lon,y=lat),data=crime)
# contour plot of the density of crime subdivided by day of crime (notice the similarity among the contours)
ggmap(houston) + stat_density2d(aes(x=lon,y=lat,color=factor(day)),data=crime)
```

Analysis of crime data (cont.)

```
# draw separate contour plot of the density estimate of crime incidences subdivided by type of crime
ggmap(houston) + stat_density2d(aes(x=lon,y=lat),data=crime) + facet_wrap(~offense)
# instead of the contour plot, use surface plot
overlay = stat_density2d(aes(x=lon,y=lat, fill = ..level.., alpha = ..level..),bins = 5,geom="polygon",data =crime)
# overlay is a ggplot object to be used as a layer for depicting the density surface
ggmap(houston) + overlay + scale_fill_gradient(low="blue",high="red")
# surface plot of the density of crime subdivided by type of crime
ggmap(houston) + overlay + scale_fill_gradient(low="blue",high="red") + facet_wrap(~offense)
```

- Notice the two prominent zones (and one less prominent one) of criminal activities and the difference in spatial patterns across crime types.