Rashika Pramod Singh
IST 687

Lecture Notes-7

Mashup:

Mashup is anything that brings together disparate influences or elements

Is the visualization useful? If so, how?

The San Francisco gives a good overview of crime rates in different places at different itmes and there was filtering which allowed to select the information which is needed. The maps were interactive so made it clear, easy and linked which is useful for putting all the data needed in context.

Ggplot:

Ggplot is used for mapping.

Map_data- get the data on the region to be mapped

Geom_map- how to render the map (colors, heatmaps)

Coord_map-make sure the map is not stretched.

Map_id-logical way to describe data and part of aes (lower case)

Longitude and latitude- can add a specific physical location using geom_point; coordinates can be hard code and can be with a conversion from logical to physical

Code:

```
dummyDF<-data.frame(state.name,stringsAsFactors=FALSE) #create dataframe
dummyDF$state<-tolower(dummyDF$state.name)
#state.name is name of data set and lower case for gg plot
us<-map_data("state")
map.simple<-ggplot(dummyDF, aes(map_id=state)) #get the data from data
map.simple<-map.simple+geom_map(map=us, fill="white", + color="black")
#fill line and outline
map.simple<-map.simple+expand_limits(x=us$long, y=us$lat)
#size of map as per latitude and longitude
map.simple<-map.simple+coord_map() + ggtitle("basic map of usa") # to not stretch the map
```

map.simple

map.simple + geom_point(aes(x=-100, y=30)) #add points to map

Applications for maps:

Location data can be used to give information for relocation for rentals, real estate, for parking, for fire fighters or for restaurant ratings.

Getting a point to map

Latlon<-geocode("Syracuse university, Syracuse, ny")

Latlon

Code:

```
install.packages("ggplot2")
install.packages("ggmap")
library(ggplot2)
library(ggmap)

dummyDF<-data.frame(state.name,stringsAsFactors=FALSE) #create dataframe
dummyDF$state<-tolower(dummyDF$state.name)
#state.name is name of data set and lower case for gg plot
us<-map_data("state")
map.simple<-ggplot(dummyDF, aes(map_id=state))
#get the data from data
map.simple<-map.simple+geom_map(map=us, fill="white", + color="black")
#fill line and outline
map.simple<-map.simple+expand_limits(x=us$long, y=us$lat)
#size of map as per latitude and longitude
map.simple<-map.simple+coord_map() + ggtitle("basic map of usa")
# to not stretch the map
dfStates<-readStates() #read the data
dfStates$states<-gsub("\\.","", dfStates$state)
```

```
dfStates$states<-tolower(dfStates$states)
map.popColor<-ggplot(dfStates,aes(map_id=state))
map.popColor<-map.popColor+geom_map(map=us, aes(fill=base2010))
map.popColor<-map.popColor+expand_limits((x=us$long, y=us$lat)) #expanding limits
map.popColor<-map.popColor+coord_map() + ggtitle("state population") #adding a title
map.popColor

#show a point on the map
map.simple + geom_point(aes(x=-100, y=30))
map.simple + geom_point(aes(x=-100, y=30)), color="darkred", shape=1)
map.popColor + geom_point(aes(x=-100, y=30)), color="darkred", shape=1)



#show a logical location
latlon<-geocode("syracuse university, syracuse, ny")
latlon
map.popColor+geom_point(aes(x=latlon$lon, y=latlon$lat)), color="darkred", size=3)

#show second point
l<- data.frame(latlon)
latlon<-geocode("colorado")
l[2,]<-latlon
l[3,]<-geocode("denver,colorado")
map.simple + geom_point(data=1, aes(x=lon, y=lat))
l$state<-"?"
map.simple+geom_point(data=l,aes(x=lon, y=lat))
map.popColor+geom_point(data=l,aes(x=lon, y=lat), alpha=.5, color="darkred", size=3)
```

We need to clean data as it might have NAs, there might be some outliers and we can find that point. We can remove data point to make the data set clearer. We can reorder data points using continuous ranges.

Rashika Pramod Singh
IST 687

Zoom for maps:

```
#zoom in maps
zoomGeo<-geocode (“New York, ny”)
zoomAmount<-3
centerx<-zoomGeo$lon # to get longitude
centery<-zoomGeo$lay # to get latitude
ylimit<-c(centery-zoomAmount,centery+zoomAmount)
#add/subtract zoom amount for range of y
Xlimit<-c(centerX-zoomAmount,centerX+zoomAmount)
Map.zoom<-ggplot(allCnt, aes(map_id=state)
Map.zoom<-map.zoom + geom_map(map=us, aes(fill=count))
Map.zoom<-map.zoom + expand_limits(x=xlimit,y=ylimit)
Map.zoom<-map.zoom + coord_map()
```