R Notebook

NYPD <- read.csv("~/Downloads/NYPD\_Motor\_Vehicle\_Collisions.csv", stringsAsFactors=FALSE)

create map

require(ggmap)

## Loading required package: ggmap

## Warning: package 'ggmap' was built under R version 3.5.2

## Loading required package: ggplot2

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.

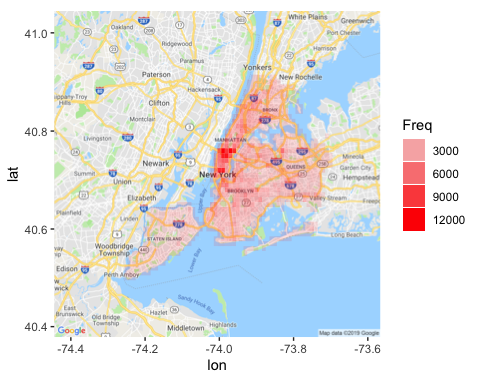
locs <- NYPD[c(5,6)]  
register\_google(key = "AIzaSyCLFqGoa-g\_cytqBGovpVtr-yuTPf031yM", account\_type = "standard")  
nyc\_locs <- get\_map(location = "New York City", maptype = 'roadmap')

## Source : https://maps.googleapis.com/maps/api/staticmap?center=New%20York%20City&zoom=10&size=640x640&scale=2&maptype=roadmap&language=en-EN&key=xxx-g\_cytqBGovpVtr-yuTPf031yM

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=New+York+City&key=xxx-g\_cytqBGovpVtr-yuTPf031yM

counts <- as.data.frame(table(round(locs$LONGITUDE,2), round(locs$LATITUDE,2)))  
counts$Long <- as.numeric(as.character(counts$Var1))  
counts$Lat <- as.numeric(as.character(counts$Var2))  
counts2 <- subset(counts, Freq > 0)  
ggmap(nyc\_locs) + geom\_tile(data = counts2, aes(x = Long, y = Lat, alpha = Freq), fill = "red")

## Warning: Removed 18 rows containing missing values (geom\_tile).



create time plot

df <- NYPD[c(1,2,13:18)]

split the date and time column

test\_df <- df  
test\_df$DATE <- strptime(as.character(test\_df$DATE), "%m/%d/%Y")  
test\_df$Year <- as.numeric(format(test\_df$DATE, format = "%Y"))  
test\_df$Month <- as.numeric(format(test\_df$DATE, format = "%m"))  
test\_df$Day <- as.numeric(format(test\_df$DATE, format = "%d"))  
test\_df$Time <- strptime(as.character(test\_df$TIME), "%H:%M")   
test\_df$Hour <- as.numeric(format(test\_df$Time, format = "%H"))  
test\_df$Minute <- as.numeric(format(test\_df$Time, format = "%M"))  
test\_df$Time <- NULL

solution1: convert to categorical data divide months according to the tempature, daylight and snowfall of NYC p1:1,2,3,12 p2:4,5,10,11 p3:6,7,8,9

p1 <- c(1,2,3,12)  
p2 <- c(4,5,10,11)  
p3 <- c(6,7,8,9)  
test\_df$Part[test\_df$Month %in% p1] <- 1  
test\_df$Part[test\_df$Month %in% p2] <- 2  
test\_df$Part[test\_df$Month %in% p3] <- 3  
# create different frames  
splitlist <- split(test\_df, test\_df$Part)  
# loop  
require(plyr)

## Loading required package: plyr

require(reshape2)

## Loading required package: reshape2

require(lattice)

## Loading required package: lattice

col <- c("red","green","blue")  
#vertical line  
divide <- c(4,6,8,9,17,18,20)  
#set legend  
#plot.new()  
#legend(x = "top",inset = 0,  
# legend =c("1,2,3,12","4,5,10,11","6,7,8,9"),   
# col=col, lwd=1, cex=.5, horiz = TRUE)  
  
# plot by number  
# avoid y axis changes  
lmi<-list(c(20,2250),c(0,31),c(40,1200),c(0,6),c(860,7100),c(0,25))  
for(i in 1:3){  
 # create different frames  
 P <- splitlist[[i]]  
 # sum by hour  
 P\_df <- P[c(3:8,12)]  
 P\_df <- ddply(P\_df, "Hour", numcolwise(sum))  
 #plot  
 mm <- melt(subset(P\_df,select=c(  
 Hour,NUMBER.OF.PEDESTRIANS.INJURED,NUMBER.OF.PEDESTRIANS.KILLED,NUMBER.OF.CYCLIST.INJURED,NUMBER.OF.CYCLIST.KILLED,  
 NUMBER.OF.MOTORIST.INJURED,NUMBER.OF.MOTORIST.KILLED)),id.var="Hour")  
 plot <- xyplot(value~Hour|variable,data=mm,type="l",col=col[i],  
 scales=list(y=list(relation="free",limits=lmi), x=list(at=c(0:23))),  
 par.settings = list(superpose.line = list(lwd=20)),  
 layout=c(1,6),  
 panel = function( x,y,...) {  
 panel.abline( v=x[ which(x %in% divide) ], lty = "dotted", col = "black")  
 panel.xyplot( x,y,...)  
 },  
 key=list(space="top",columns=3,text=list(lab=c("1,2,3,12","4,5,10,11","6,7,8,9")),  
 lines=list(lwt=2,col=col))  
 )  
 var\_name <- paste("plot", i, sep="\_")  
 assign(var\_name, plot, env=.GlobalEnv)  
}  
require(RColorBrewer)

## Loading required package: RColorBrewer

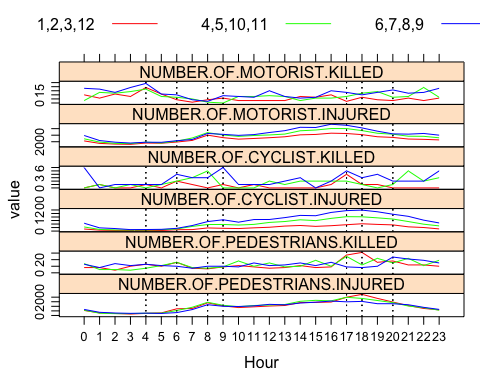
require(latticeExtra)

## Loading required package: latticeExtra

##   
## Attaching package: 'latticeExtra'

## The following object is masked from 'package:ggplot2':  
##   
## layer

plot\_1+plot\_2+plot\_3

 plot by rate

lmi<-list(c(0.94,1),c(0,0.06),c(0.965,1),c(0,0.035),c(0.9875,1),c(0,0.0125))  
for(i in 1:3){  
 # create different frames  
 P <- splitlist[[i]]  
 # sum by hour  
 P\_df <- P[c(3:8,12)]  
 P\_df <- ddply(P\_df, "Hour", numcolwise(sum))  
 # create injured/accident + killed/accident rates  
 P\_df$NUMBER.OF.PEDESTRIANS <- P\_df$NUMBER.OF.PEDESTRIANS.INJURED + P\_df$NUMBER.OF.PEDESTRIANS.KILLED  
 P\_df$NUMBER.OF.CYCLIST <- P\_df$NUMBER.OF.CYCLIST.INJURED + P\_df$NUMBER.OF.CYCLIST.KILLED  
 P\_df$NUMBER.OF.MOTORIST <- P\_df$NUMBER.OF.MOTORIST.INJURED + P\_df$NUMBER.OF.MOTORIST.KILLED  
 P\_df$PEDESTRIANS.INJURED.RATE <- P\_df$NUMBER.OF.PEDESTRIANS.INJURED / P\_df$NUMBER.OF.PEDESTRIANS  
 P\_df$PEDESTRIANS.KILLED.RATE <- P\_df$NUMBER.OF.PEDESTRIANS.KILLED / P\_df$NUMBER.OF.PEDESTRIANS  
 P\_df$CYCLIST.INJURED.RATE <- P\_df$NUMBER.OF.CYCLIST.INJURED / P\_df$NUMBER.OF.CYCLIST  
 P\_df$CYCLIST.KILLED.RATE <- P\_df$NUMBER.OF.CYCLIST.KILLED / P\_df$NUMBER.OF.CYCLIST  
 P\_df$MOTORIST.INJURED.RATE <- P\_df$NUMBER.OF.MOTORIST.INJURED / P\_df$NUMBER.OF.MOTORIST  
 P\_df$MOTORIST.KILLED.RATE <- P\_df$NUMBER.OF.MOTORIST.KILLED / P\_df$NUMBER.OF.MOTORIST  
 #plot  
 mm <- melt(subset(P\_df,select=c(  
 Hour,PEDESTRIANS.INJURED.RATE,PEDESTRIANS.KILLED.RATE,CYCLIST.INJURED.RATE,CYCLIST.KILLED.RATE,  
 MOTORIST.INJURED.RATE,MOTORIST.KILLED.RATE)),id.var="Hour")  
 plot <- xyplot(value~Hour|variable,data=mm,type="l",col=col[i],  
 scales=list(y=list(relation="free",limits=lmi), x=list(at=c(0:23))),  
 par.settings = list(superpose.line = list(lwd=20)),  
 layout=c(1,6),  
 panel = function( x,y,...) {  
 panel.abline( v=x[ which(x %in% divide) ], lty = "dotted", col = "black")  
 panel.xyplot( x,y,...)},  
 key=list(space="top",columns=3,text=list(lab=c("1,2,3,12","4,5,10,11","6,7,8,9")),  
 lines=list(lwt=2,col=col))  
 )  
 var\_name <- paste("plot", i, sep="\_")  
 assign(var\_name, plot, env=.GlobalEnv)  
}  
require(RColorBrewer)  
require(latticeExtra)  
plot\_1+plot\_2+plot\_3

