NYPD COLLISIONS

Hourly Analysis

Jiahuan Li Tushar Sharma

1 March 2019

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.

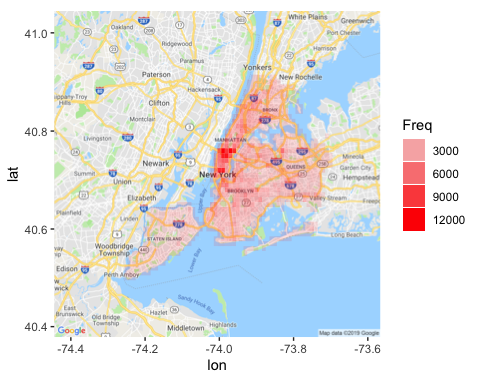
locs <- NYPD[c(5,6)]  
register\_google(key = "AIzaSyCLFqGoa-g\_cytqBGovpVtr-yuTPf031yM", account\_type = "standard")  
nyc\_locs <- get\_map(location = "New York City", maptype = 'roadmap')

## 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

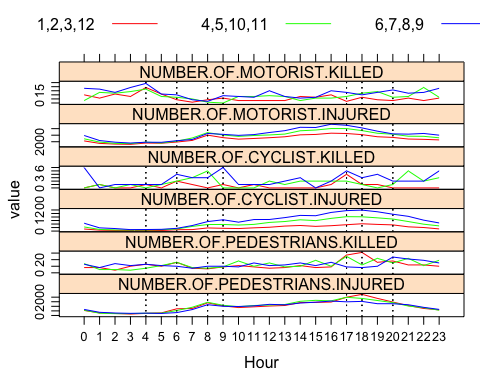
require(latticeExtra)

## Loading required package: latticeExtra

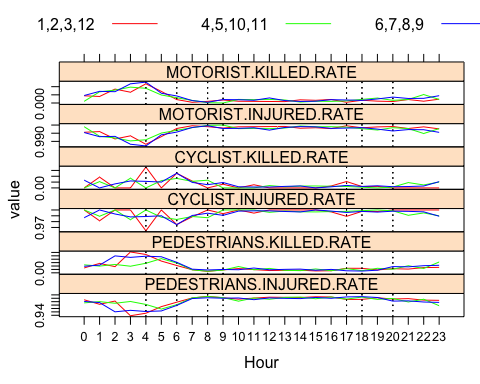
##   
## Attaching package: 'latticeExtra'

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

plot\_1+plot\_2+plot\_3

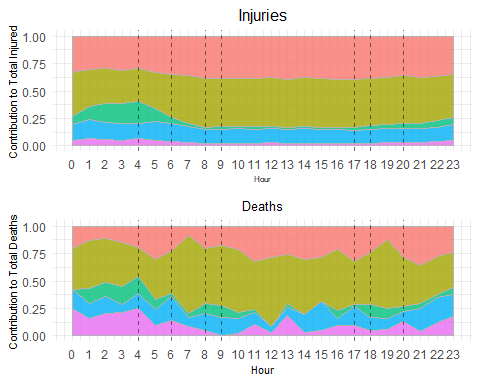
 plot by rate

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



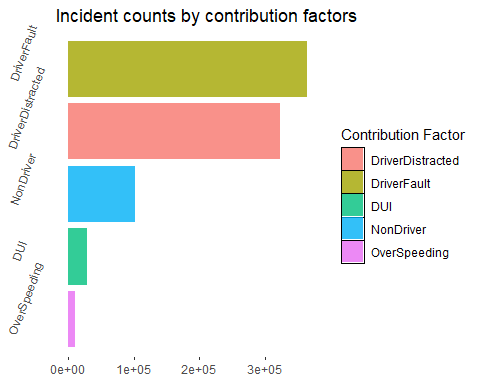
library(scales)  
library(ggplot2)  
library(stringr)  
library(ggplot2)  
library(changepoint)  
library(scales)  
library(dplyr)  
library(tidyr)  
library(grid)  
library(gridExtra)  
  
#importing data  
nydata<-read.csv("NYPD\_Motor\_Vehicle\_Collisions.csv", stringsAsFactors = FALSE, na.strings = '')  
  
#extracting hour and year in another column  
nydata$hour<- as.integer(str\_split\_fixed(nydata$TIME,":",2)[,1])  
nydata$year<- as.integer(str\_split\_fixed(nydata$DATE,"/",3)[,3])  
  
#Weekday weekend split with general count and sums  
nydata$Weekday<- weekdays(as.Date(nydata$DATE, "%m/%d/%Y"))  
  
nydata$tag <- ifelse(nydata$Weekday %in%   
 c("Monday","Tuesday","Wednesday","Thursday","Friday"),   
 "Weekday", "Weekend")  
  
  
  
#key-value to convert injured columns->rows  
nydata\_injury <- filter(gather(nydata[,c(1,19,33,30,13,15,17,24)] ,   
 key = "Fatal.Category",   
 value = "Injured",   
 NUMBER.OF.PEDESTRIANS.INJURED,  
 NUMBER.OF.CYCLIST.INJURED,   
 NUMBER.OF.MOTORIST.INJURED), is.na(Injured) == FALSE)  
  
#key-value to convert killed columns->rows  
nydata\_killed <- filter(gather(nydata[,c(1,19,33,30,14,16,18,24)] ,   
 key = "Fatal.Category",  
 value = "Killed",  
 NUMBER.OF.PEDESTRIANS.KILLED,  
 NUMBER.OF.CYCLIST.KILLED,  
 NUMBER.OF.MOTORIST.KILLED), is.na(Killed) == FALSE)  
  
#summarizing measures for required parameters  
nysummary\_injury <- nydata\_injury %>%  
 group\_by(CONTRIBUTING.FACTOR.VEHICLE.1,  
 hour, tag, Fatal.Category) %>%  
 summarise(TOTAL.injured = sum(Injured, na.rm = TRUE),  
 Day.Count = n\_distinct(DATE, na.rm = TRUE),  
 Accident.count = n\_distinct(UNIQUE.KEY, na.rm = TRUE))  
  
nysummary\_killed <- nydata\_killed %>%  
 group\_by(CONTRIBUTING.FACTOR.VEHICLE.1,  
 hour, tag, Fatal.Category) %>%  
 summarise(TOTAL.killed = sum(Killed, na.rm = TRUE),  
 Day.Count = n\_distinct(DATE, na.rm = TRUE),  
 Accident.count = n\_distinct(UNIQUE.KEY, na.rm = TRUE))  
  
  
nysummary\_injury\_final <- nydata\_injury %>%  
 group\_by(CONTRIBUTING.FACTOR.VEHICLE.1,  
 hour) %>%  
 summarise(TOTAL.injured = sum(Injured, na.rm = TRUE),  
 Day.Count = n\_distinct(DATE, na.rm = TRUE),  
 Accident.count = n\_distinct(UNIQUE.KEY, na.rm = TRUE))  
  
nysummary\_killed\_final <- nydata\_killed %>%  
 group\_by(CONTRIBUTING.FACTOR.VEHICLE.1,  
 hour) %>%  
 summarise(TOTAL.killed = sum(Killed, na.rm = TRUE),  
 Day.Count = n\_distinct(DATE, na.rm = TRUE),  
 Accident.count = n\_distinct(UNIQUE.KEY, na.rm = TRUE))  
  
#replacing fatal categories for consistency before merge  
nysummary\_injury$Fatal.Category <- str\_split\_fixed(nysummary\_injury$Fatal.Category, "\\.",4)[,3]  
nysummary\_killed$Fatal.Category <- str\_split\_fixed(nysummary\_killed$Fatal.Category, "\\.",4)[,3]  
  
#creating final dataset here  
nysummary<- merge(x= nysummary\_injury,   
 y= nysummary\_killed,   
 by = c("CONTRIBUTING.FACTOR.VEHICLE.1",  
 "hour", "tag", "Fatal.Category"),  
 all = TRUE)  
  
nysummary\_final<- merge(x= nysummary\_injury\_final,   
 y= nysummary\_killed\_final,   
 by = c("CONTRIBUTING.FACTOR.VEHICLE.1",  
 "hour"),  
 all = TRUE)  
  
nysummary\_final <- arrange(nysummary\_final, desc(TOTAL.injured))  
  
  
nysummary\_final$cont <- nysummary\_final$Accident.count.x/sum(nysummary\_final$Accident.count.x)  
  
nysummary\_final$cont <- ifelse(nysummary\_final$CONTRIBUTING.FACTOR.VEHICLE.1 %in%  
 c("Driver Inattention/Distraction",  
 "Failure to Yield Right-of-Way",  
 "Following Too Closely",  
 "Backing Unsafely",  
 "Fatigued/Drowsy",  
 "Other Vehicular",  
 "Turning Improperly",  
 "Passing or Lane Usage Improper",  
 "Passing Too Closely",  
 "Unsafe Lane Changing",  
 "Traffic Control Disregarded",  
 "Driver Inexperience",  
 "Lost Consciousness",  
 "Prescription Medication",  
 "Pavement Slippery",  
 "Alcohol Involvement",  
 "Outside Car Distraction",  
 "Reaction to Uninvolved Vehicle",  
 "Unsafe Speed"), nysummary\_final$CONTRIBUTING.FACTOR.VEHICLE.1, "Others")  
  
# || (nysummary\_final$cont!="Others") && (is.na(nysummary\_final$cont)!=TRUE))  
nysummary\_final\_sub <- nysummary\_final[(nysummary\_final$CONTRIBUTING.FACTOR.VEHICLE.1!="Unspecified"),]  
nysummary\_final\_sub <- nysummary\_final\_sub[(is.na(nysummary\_final$cont)!=TRUE),]  
  
  
contmap<- read.csv("cont.csv", stringsAsFactors = FALSE)  
  
  
nysummary\_final\_red <- merge(x=nysummary\_final\_sub, y= contmap, by=("CONTRIBUTING.FACTOR.VEHICLE.1"), all.x = TRUE)  
  
  
nysummary\_final\_red$injuredrate<-nysummary\_final\_red$TOTAL.injured/nysummary\_final\_red$Accident.count.x  
nysummary\_final\_red$injuredratio<-nysummary\_final\_red$TOTAL.injured/(nysummary\_final\_red$TOTAL.injured+  
 nysummary\_final\_red$TOTAL.killed)  
  
nysummary\_final\_red$killedrate<-nysummary\_final\_red$TOTAL.killed/nysummary\_final\_red$Accident.count.x  
nysummary\_final\_red$killedratio<-nysummary\_final\_red$TOTAL.killed/(nysummary\_final\_red$TOTAL.killed+  
 nysummary\_final\_red$TOTAL.injured)  
  
nysummary\_final\_red<-nysummary\_final\_red[is.na(nysummary\_final\_red$Cont)==FALSE,]   
  
nysummary\_area\_Plot <- nysummary\_final\_red %>%   
 group\_by(Cont, hour) %>%  
 summarise(Total.Injured=sum(TOTAL.injured, na.rm = TRUE),  
 TOTAL.killed=sum(TOTAL.killed, na.rm = TRUE),  
 TOTAL.Incidents=sum(Accident.count.x, na.rm = TRUE))

p1<- ggplot(nysummary\_area\_Plot, aes(x = hour, y= TOTAL.Incidents, fill=Cont))+  
 scale\_x\_continuous(name="Hour", breaks = seq(0,23,1))+scale\_y\_continuous(name = "Contribution to Total Incidents",labels = percent\_format())+geom\_area(stat="Identity",position="fill",  
 alpha = 0.8, color = "grey70")+ggtitle("Incidents")+theme\_minimal()+  
 geom\_vline(xintercept = c(4,6,8,9,17,18,20), linetype = "dashed", size=0.5, alpha = 0.5)+theme(legend.box.background = element\_rect())+theme(legend.position="none")+theme(plot.title = element\_text(hjust = 0.5, size = 12))+theme(axis.title.y=element\_text(size=8), axis.title.x =element\_text(size=6))  
  
p2<- ggplot(nysummary\_area\_Plot, aes(x = hour, y= Total.Injured, fill=Cont))+  
 scale\_x\_continuous(name="Hour", breaks = seq(0,23,1))+scale\_y\_continuous(name = "Contribution to Total Injured")+geom\_area(stat="Identity",position="fill",  
 alpha = 0.8, color = "grey70")+ggtitle("Injuries")+theme\_minimal()+  
 geom\_vline(xintercept = c(4,6,8,9,17,18,20), linetype = "dashed", size=0.5, alpha = 0.5)+theme(legend.box.background = element\_rect())+theme(legend.position="none")+theme(plot.title = element\_text(hjust = 0.5, size =12))+theme(axis.title.y=element\_text(size=8), axis.title.x =element\_text(size=6))  
  
p3<- ggplot(nysummary\_area\_Plot, aes(x = hour, y= TOTAL.killed, fill=Cont))+  
 scale\_x\_continuous(name="Hour", breaks = seq(0,23,1))+scale\_y\_continuous(name = "Contribution to Total Deaths")+geom\_area(stat="Identity",position="fill",alpha = 0.8, color = "grey70")+ggtitle("Deaths")+theme\_minimal()+  
 geom\_vline(xintercept = c(4,6,8,9,17,18,20), linetype = "dashed", size=0.5, alpha = 0.5)+theme(legend.box.background = element\_rect())+theme(legend.position="none")+theme(plot.title = element\_text(hjust = 0.5, size = 12))+theme(axis.title.y=element\_text(size=8), axis.title.x =element\_text(size=8), plot.title = element\_text(size = 10))  
  
  
grid.arrange(p2,p3)



## Including Plots

nysummary\_incidents<- summarise(group\_by(nysummary\_area\_Plot, Cont), Total.Incidents = sum(TOTAL.Incidents))  
  
ggplot(nysummary\_incidents, aes(x = reorder(Cont, Total.Incidents), y = Total.Incidents, fill = factor(Cont))) +scale\_y\_continuous(breaks = c(0,100000,200000,300000,400000))+scale\_x\_discrete()+  
 geom\_bar(stat = "identity", alpha = 0.8) +  
 coord\_flip() +   
 ggtitle("Incident counts by contribution factors") + theme(axis.text.y = element\_text(angle =70, hjust = 0.2), axis.title.x=element\_blank(), axis.ticks.y = element\_blank(),  
 axis.title.y = element\_blank(),legend.key = element\_rect(fill = "white", colour = "black"),panel.grid.major = element\_blank(), panel.grid.minor = element\_blank(), panel.background = element\_blank())+guides(fill=guide\_legend(title="Contribution Factor"))



library(ggmap)

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

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

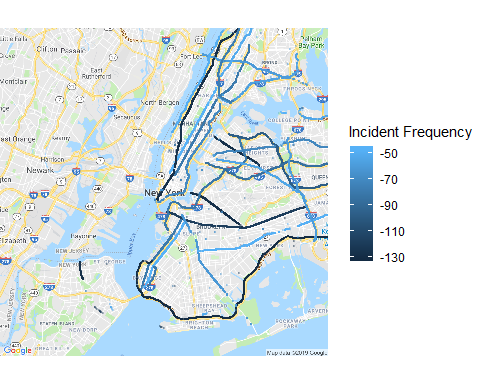
register\_google(key = "AIzaSyBtBKJv6Owt0yYRnj6VUZOci9gYh1B4\_bM", account\_type = "standard")  
  
  
nydata\_streets<- nydata[is.na(nydata$ON.STREET.NAME) == FALSE,]  
  
nydata\_streets\_contfact<- merge(x=nydata\_streets,   
 y= contmap, by=("CONTRIBUTING.FACTOR.VEHICLE.1"), all.x = TRUE)  
  
  
#library(dplyr)  
nystreets <- arrange(nydata\_streets\_contfact %>%  
 group\_by(ON.STREET.NAME, Cont, hour) %>%  
 summarise(TOTAL.killed = sum(NUMBER.OF.PERSONS.KILLED, na.rm = TRUE),  
 TOTAL.injured = sum(NUMBER.OF.PERSONS.INJURED, na.rm = TRUE),  
 TOTAL.incidents = n\_distinct(UNIQUE.KEY, na.rm = TRUE)), desc(TOTAL.incidents))  
  
nystreets <- nystreets[is.na(nystreets$Cont) == FALSE,]  
  
  
  
nystreets$daysplit <- ifelse(nystreets$hour<7, "Night", "Day")  
  
nystreets\_summary <- arrange(nystreets %>%  
 group\_by(ON.STREET.NAME, Cont, daysplit) %>%  
 summarise(TOTAL.killed = sum(TOTAL.killed, na.rm = TRUE),  
 TOTAL.injured = sum(TOTAL.injured, na.rm = TRUE),  
 TOTAL.incidents = sum(TOTAL.incidents, na.rm = TRUE)),  
 desc(TOTAL.incidents))  
  
  
daymap<-nystreets\_summary[nystreets\_summary$daysplit=="Day",]  
  
nightmap<-nystreets\_summary[nystreets\_summary$daysplit=="Night",]  
  
nightmap<-nightmap[(nightmap$Cont=="DUI" | nightmap$Cont=="OverSpeeding"),]  
daymap<-daymap[(daymap$Cont=="DriverFault" | daymap$Cont=="DriverDistracted"),]  
  
night<- arrange(data.frame(summarise(group\_by(nightmap, ON.STREET.NAME),  
 Total = sum(TOTAL.incidents))), desc(Total))  
  
day <- arrange(data.frame(summarise(group\_by(daymap, ON.STREET.NAME),  
 Total = sum(TOTAL.incidents))), desc(Total))  
  
#Selecting top 20 streets  
night<-night[1:20,]  
day<-day[1:20,]  
  
#getting all coordinates for top 20 streets  
night <- merge(x= night, y = nydata, by = c("ON.STREET.NAME"), all.x = TRUE)  
  
day <- merge(x= day, y = nydata, by = c("ON.STREET.NAME"), all.x = TRUE)  
  
#map settings  
theme\_set(theme\_dark())  
NYMap <- qmap("new york", zoom = 11, maptype = c("roadmap"))

## Source : https://maps.googleapis.com/maps/api/staticmap?center=new%20york&zoom=11&size=640x640&scale=2&maptype=roadmap&language=en-EN&key=xxx

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=new+york&key=xxx

NYMap +  
 geom\_point(aes(x = LONGITUDE, y = LATITUDE,  
 colour = desc(Total)),  
 data = night, size = 0.2, alpha = 0.3)+labs(colour = "Incident Frequency")

## Warning: Removed 29271 rows containing missing values (geom\_point).



NYMap +  
 geom\_point(aes(x = LONGITUDE, y = LATITUDE,  
 colour = desc(Total)),  
 data = day, size = 0.2, alpha = 0.3)+labs(colour = "Incident Frequency")

## Warning: Removed 28848 rows containing missing values (geom\_point).

