NYPD COLLISIONS

Hourly Analysis

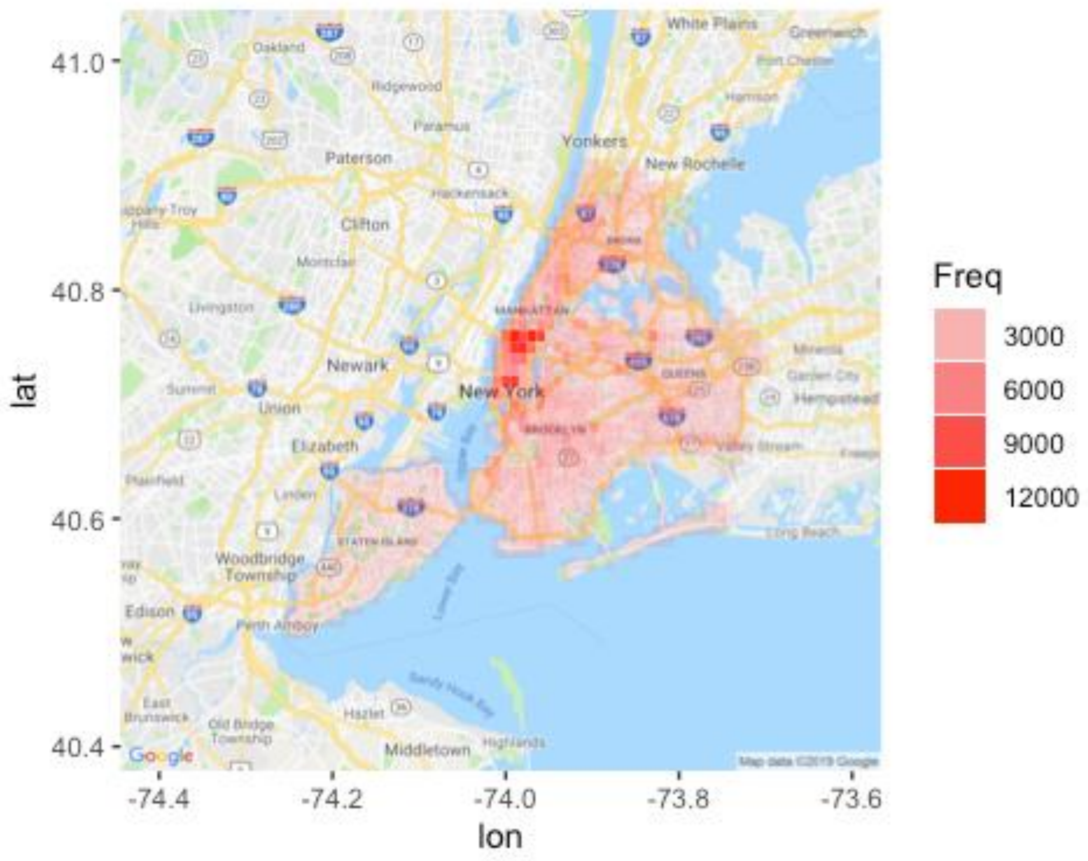
Jiahuan Li Tushar Sharma

1 March 2019

```
NYPD <- read.csv("~/Downloads/NYPD_Motor_Vehicle_Collisions.csv", stringsAsFa
ctors=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+Yor
k+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 = Fr
eq), fill = "red")

## Warning: Removed 18 rows containing missing values (geom_tile).
```



create time plot

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

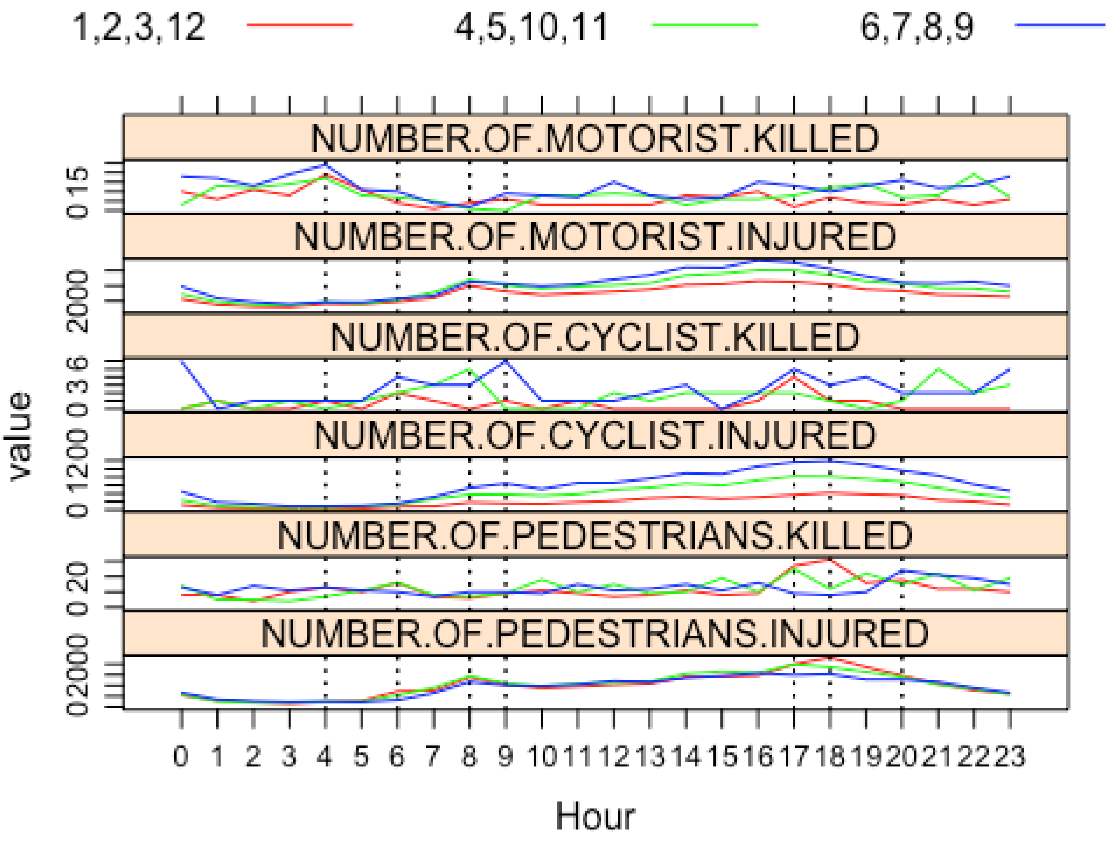
split the date and time column

```
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

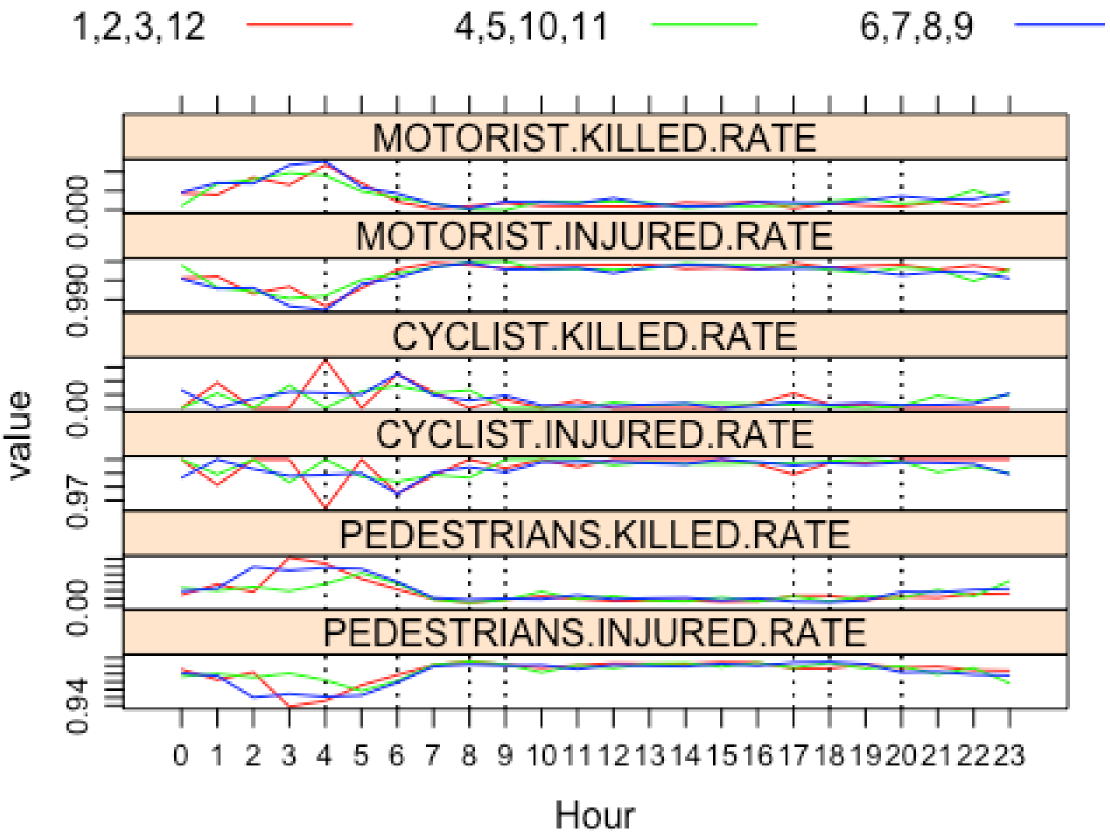
```
p1 <- c(1,2,3,12)
p2 <- c(4,5,10,11)
p3 <- c(6,7,8,9)
test_df$Part[test_df$Month %in% p1] <- 1
test_df$Part[test_df$Month %in% p2] <- 2
```

```
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

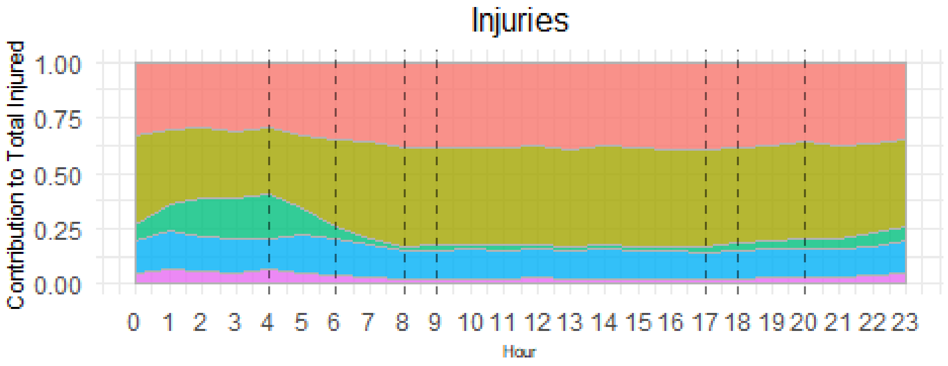
```r
lmi<-list(c(0.94,1),c(0,0.06),c(0.965,1),c(0,0.035),c(0.9875,1),c(0,0.0125))
for(i in 1:3){
  # create different frames
  P <- splitlist[[i]]
  # sum by hour
  P_df <- P[c(3:8,12)]
  P_df <- ddply(P_df, "Hour", numcolwise(sum))
  # create injured/accident + killed/accident rates
  P_df$NUMBER.OF.PEDESTRIANS <- P_df$NUMBER.OF.PEDESTRIANS.INJURED + P_df$NUM
BER.OF.PEDESTRIANS.KILLED
  P_df$NUMBER.OF.CYCLIST <- P_df$NUMBER.OF.CYCLIST.INJURED + P_df$NUMBER.OF.C
YCLIST.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$NU
MBER.OF.PEDESTRIANS
  P_df$CYCLIST.INJURED.RATE <- P_df$NUMBER.OF.CYCLIST.INJURED / P_df$NUMBER.O
F.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.O
F.MOTORIST
  #plot
  mm <- melt(subset(P_df,select=c(
    Hour,PEDESTRIANS.INJURED.RATE,PEDESTRIANS.KILLED.RATE,CYCLIST.INJURED.RAT
E,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
```

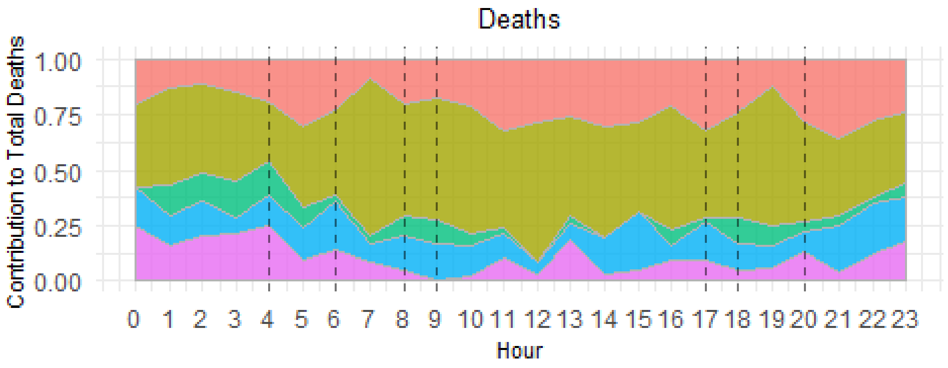


```
library(scales)
library(ggplot2)
library(stringr)
library(ggplot2)
library(changepoint)
library(scales)
library(dplyr)
library(tidyr)
library(grid)
library(gridExtra)

#importing data
nydata<-read.csv("NYPD_Motor_Vehicle_Collisions.csv", stringsAsFactors = FALS
E, 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","Thur
sday","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 = TRU
E),

                             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.Cat
egory, "\\.",4)[,3]
nysummary_killed$Fatal.Category <- str_split_fixed(nysummary_killed$Fatal.Cat
egory, "\\.",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$CONTRIBUT
ING.FACTOR.VEHICLE.1, "Others")

# || (nysummary_final$cont!="Others") && (is.na(nysummary_final$cont)!=TRUE))
nysummary_final_sub <- nysummary_final[(nysummary_final$CONTRIBUTING.FACTOR.V
EHICLE.1!="Unspecified"),]
nysummary_final_sub <- nysummary_final_sub[(is.na(nysummary_final$cont)!=TRUE
),]


contmap<- read.csv("cont.csv", stringsAsFactors = FALSE)


nysummary_final_red <- merge(x=nysummary_final_sub, y= contmap, by=("CONTRIBU
TING.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$TOTAL.injured/(nysummar
y_final_red$TOTAL.injured+
                                                                nysummary_fina
l_red$TOTAL.killed)

nysummary_final_red$killedrate<-nysummary_final_red$TOTAL.killed/nysummary_fi
nal_red$Accident.count.x
nysummary_final_red$killedratio<-nysummary_final_red$TOTAL.killed/(nysummary_
final_red$TOTAL.killed+
                                                                nysummary_fina
l_red$TOTAL.injured)

nysummary_final_red<-nysummary_final_red[is.na(nysummary_final_red$Cont)==FAL
SE,]

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.r
m = TRUE),
                                          TOTAL.Incidents=sum(Accident.count.
x, na.rm = TRUE))
```

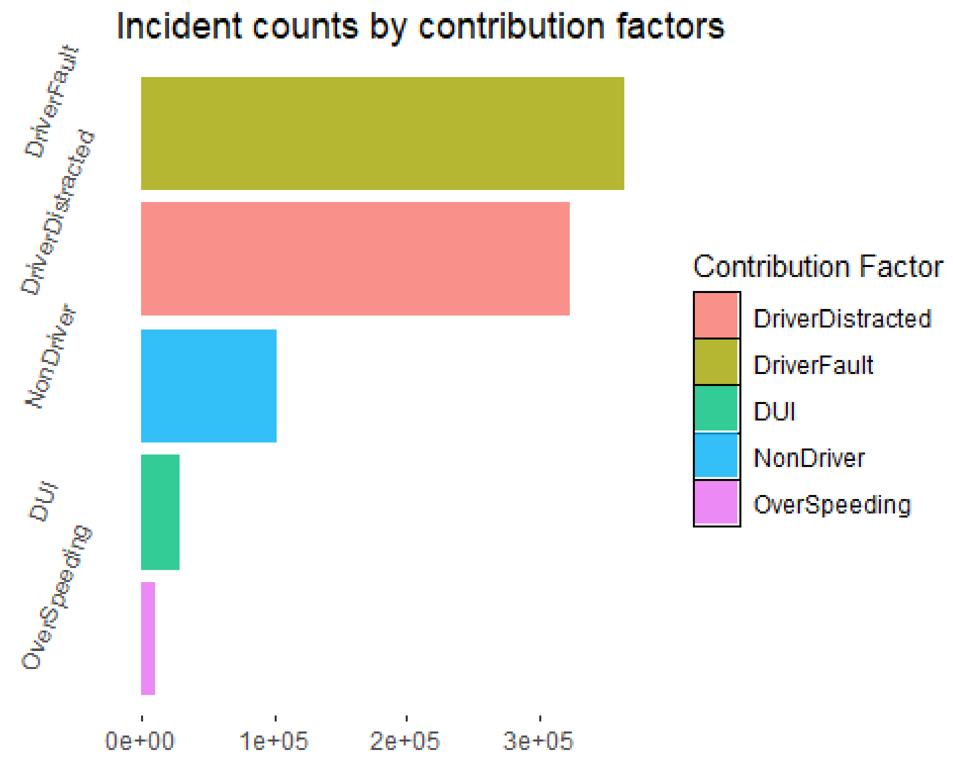
```
p1<- ggplot(nysummary_area_Plot, aes(x = hour, y= TOTAL.Incidents, fill=Cont)
)+
  scale_x_continuous(name="Hour", breaks = seq(0,23,1))+scale_y_continuous(na
me = "Contribution to Total Incidents",labels = percent_format())+geom_area(s
tat="Identity",position="fill",

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.posit
ion="none")+theme(plot.title = element_text(hjust = 0.5, size = 12))+theme(ax
is.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(na
me = "Contribution to Total Injured")+geom_area(stat="Identity",position="fil
l",

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.posit
ion="none")+theme(plot.title = element_text(hjust = 0.5, size =12))+theme(axi
s.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(na
me = "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.posit
ion="none")+theme(plot.title = element_text(hjust = 0.5, size = 12))+theme(ax
is.title.y=element_text(size=8), axis.title.x =element_text(size=8), plot.tit
le = element_text(size = 10))


grid.arrange(p2,p3)
```





Including Plots

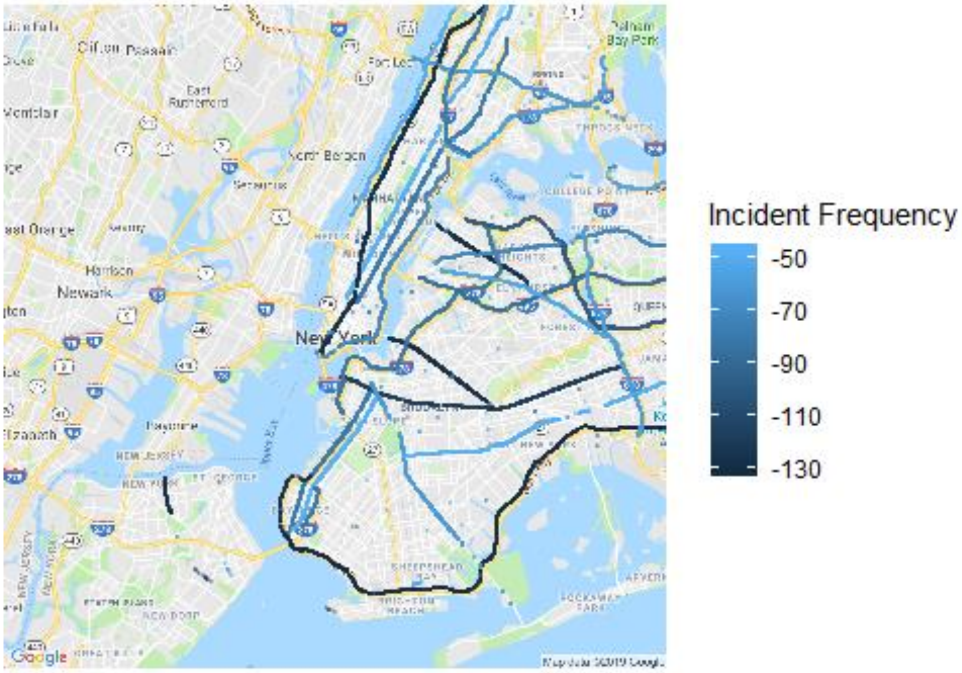
```
nysummary_incidents<- summarise(group_by(nysummary_area_Plot, Cont), Total.Incidents = sum(TOTAL.Incidents))

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"))
```



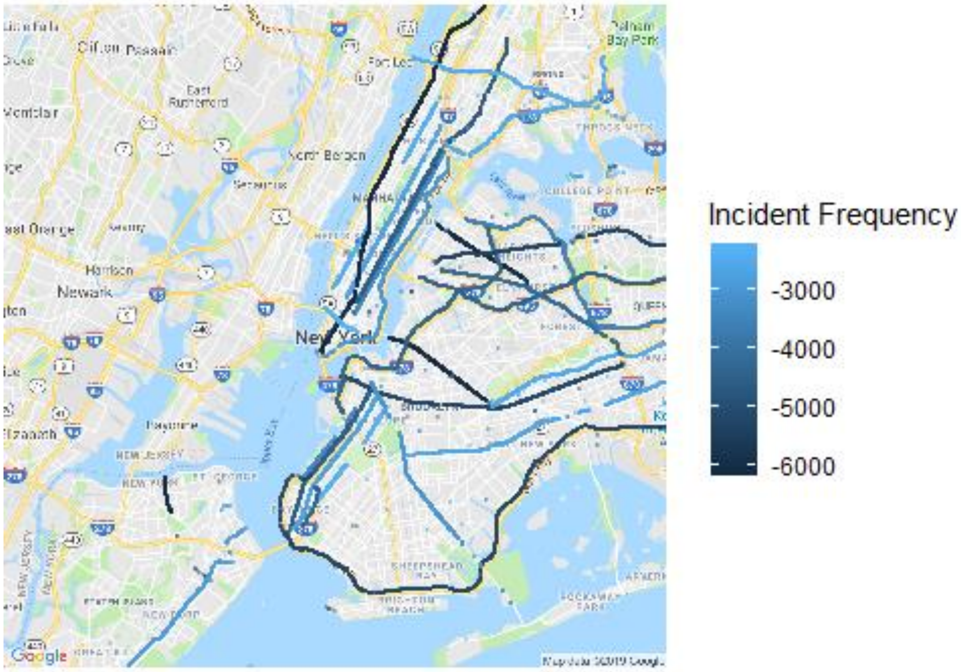
```
library(ggmap)

## 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(TOT
AL.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 = TRU
E),
                                 TOTAL.injured = sum(TOTAL.injured, na.rm = T
RUE),
                                 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(Tot
al))

day <- arrange(data.frame(summarise(group_by(daymap, ON.STREET.NAME),
                                     Total = sum(TOTAL.incidents))), desc(Tot
al))

#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+yor
k&key=xxx
```

```
NYMap +
  geom_point(aes(x = LONGITUDE, y = LATITUDE,
                 colour = desc(Total)),
             data = night, size = 0.2, alpha = 0.3)+labs(colour = "Incident F
requency")

## 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 Fre
quency")

## Warning: Removed 28848 rows containing missing values (geom_point).
```


Incident Frequency
-3000
-4000
-5000
-6000
New York
Newark
North Bergen
Bayonne
Google
Map data ©2019 Google