Exploratory Time Analysis

Landon C

August 8, 2015

Mapping

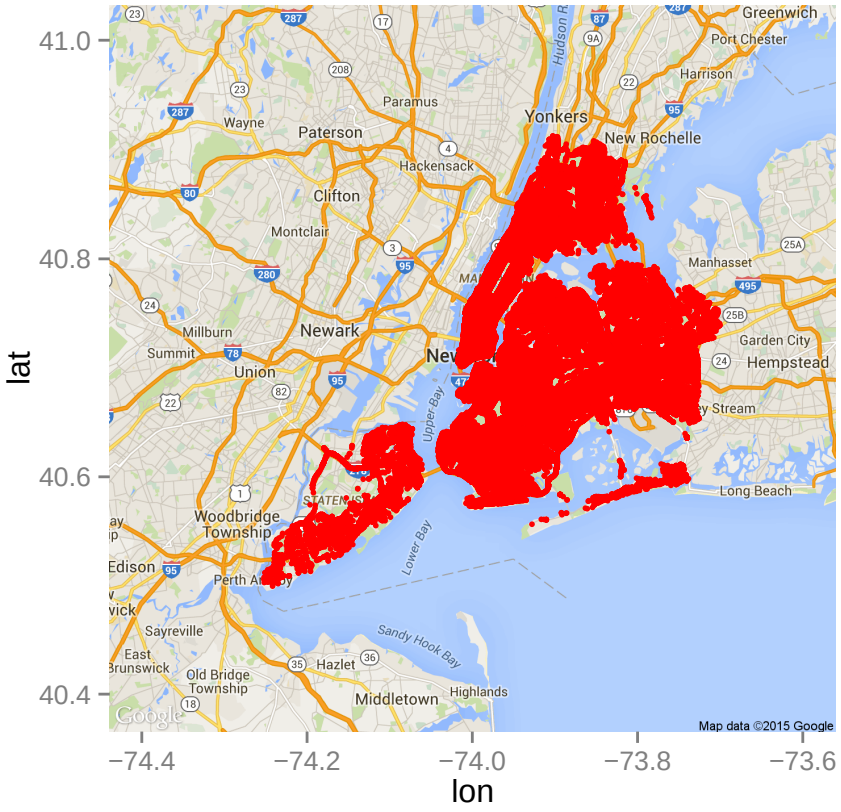
Getting a map with a location of all accidents

```
library(ggmap)
library(lubridate)

accident = read.csv("accident2015.csv", header=TRUE)
al1 = get_map(location = c(lon = -74., lat = 40.7), zoom = 10, maptype = 'roadmap')
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=40.7,-74&zoom=10&size=640x640&sca
```

```
al1MAP = ggmap(al1)

#al1MAP
al1MAP + geom_point(data = accident, aes(x = LONGITUDE, y = LATITUDE), colour = "red", size = 1)
```

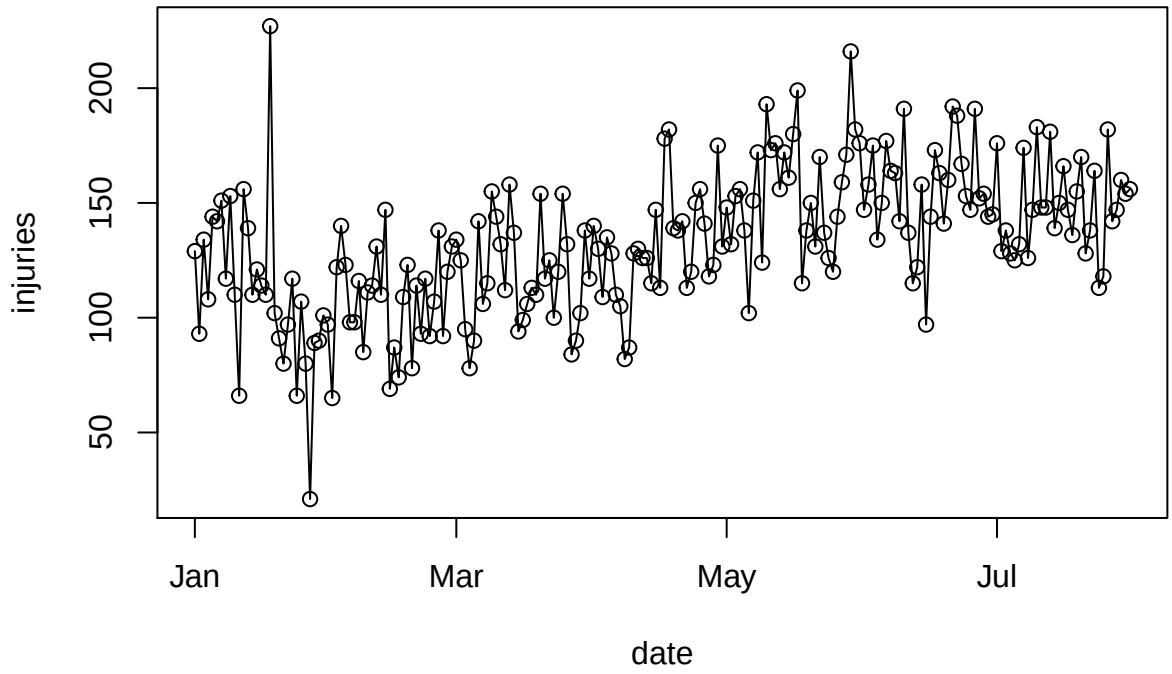
```
## Warning: Removed 20097 rows containing missing values (geom_point).
```



#Investigating Injuries Over time Investigating injuries over time for whole data set. Because the number of fatalities overall is not very high, we are not interested in this statistic. However, the number of injuries per day is high enough to possibly predict. If we have enough time, we will also look at the position (longitude and latitude) of each accident, organize accidents by clusters, and try to predict the number of accidents in each cluster.

```
#turn each date/time into a date/time object recognizable by computer
accident$DATE = as.Date(accident$DATE, "%m/%d/%Y")
accident$TIME = as.numeric(gsub('\\:','', accident$TIME))

#make month, day, time, day of week columns
accident$MONTH = month(accident$DATE)
accident$DAY = day(accident$DATE)
accident$WEEKDAY = weekdays(as.Date(accident$DATE))

#get vector of unique dates as well as months and days, and dayofweek and time of day. This will be use
dates = unique(accident$DATE)
months = unique(month(dates))
days = unique(day(dates))
weekdays = c("Monday", "Tuesday", "Wednesday","Thursday","Friday","Saturday","Sunday")
times = unique(accident$TIME)
times = sort(times)

#count the total injuries in each day
accident_injuries = 0*c(1:length(dates))
day_index = 1
for (date in dates){
```

```
  injuries_day = subset(accident, DATE == date)
  accident_injuries[day_index] =  sum(injuries_day$NUMBER.OF.PERSONS.INJURED)
  day_index = day_index + 1
}

#make dataframe of dates and injuries per day plot injuries over dates
date_injuries = data.frame(date = dates, injuries = accident_injuries)
plot(injuries ~ date, date_injuries)
lines(injuries ~ date, date_injuries, type = "l")
```

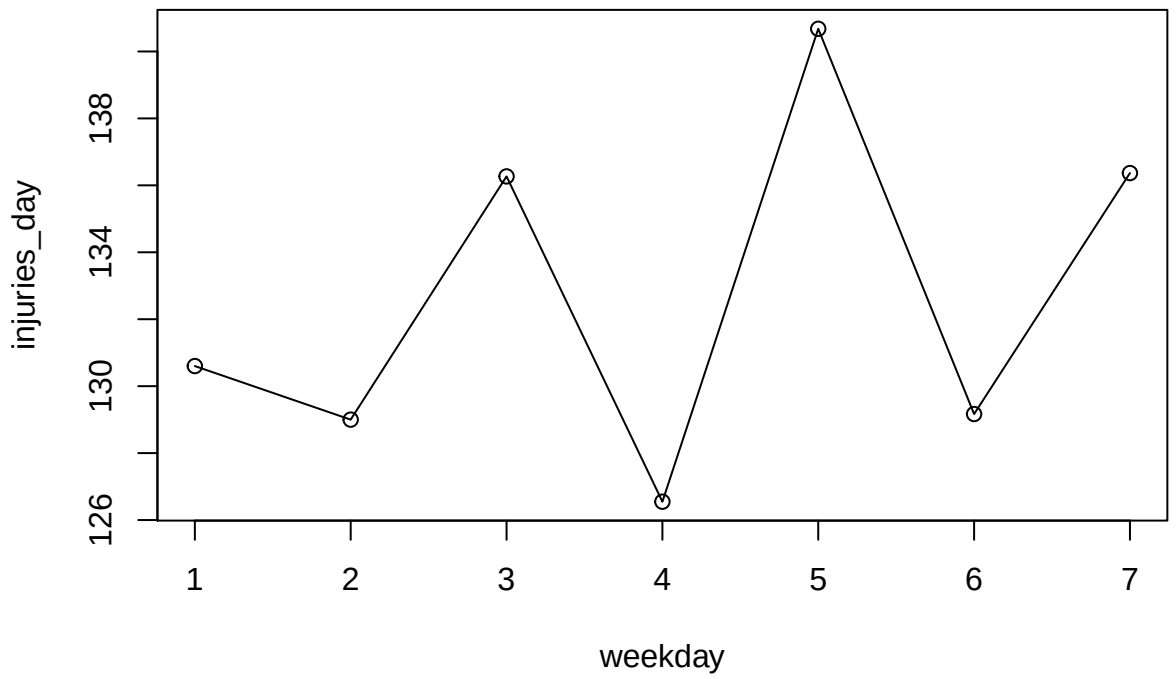



```
#outlier = identify(date_injuries$date, date_injuries$injuries)
#date_injuries[outlier,]
```

There are two clear groups: the winter to mid spring group, and the mid spring to summer group. They are seperated by the date April 15, 2015, where the dates before this have a lower average number of car crashes, and the dates after this have a higher number of car crashes. Lets discuss the two outliers in the winter, January 18 and January 27. According to New York news articles, there was a large freezing rain storm that caused a large number of accidents on January 18, causing the large spike that we see in our data. On January 27, a large winter storm hit New York, and in anticipation of that storm, many roads were closed off and schools were shut down meaning that there were few people on the roads that day.

Investigating Injuries Throughout the Week

```r
#count the total injuries for each day of the week
injuries_ct = 0*c(1:7)
day_of_week_ct = c(30,30,30,31,31,30,30) #cts the number of times that each day occurs in the data set
#count(count(accident, c("DATE","WEEKDAY"))[2],"WEEKDAY") returns occurences of each weekday
day_index = 1
for (weekday in weekdays){
  injuries_weekday = subset(date_injuries, weekdays(as.Date(date)) == weekday)
  injuries_ct[day_index] =  sum(injuries_weekday$injuries)
  day_index = day_index + 1
}

#plot the injuries over days
day_of_week = c(1:7) #Monday = 1, Tuesday = 2,.., Sunday = 7
daily_injuries = data.frame(weekday = day_of_week, injuries_day = injuries_ct/day_of_week_ct)
plot(injuries_day ~ weekday, daily_injuries)
lines(injuries_day ~ weekday, daily_injuries, type = "l")
```

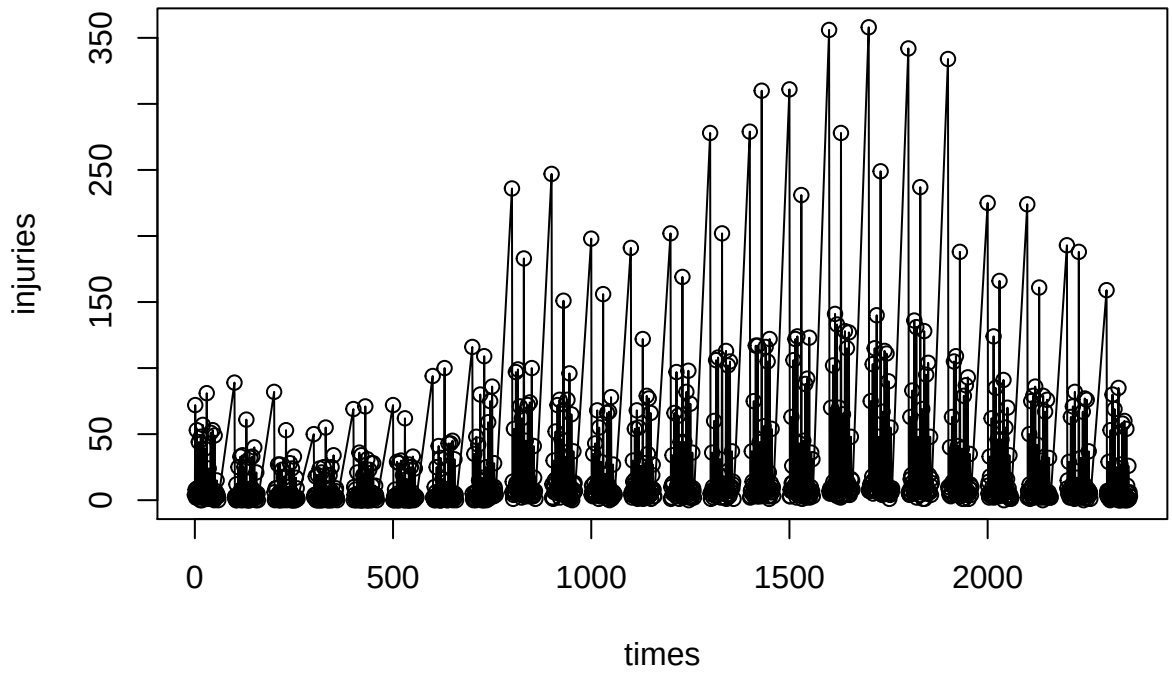



```r
#outlier = identify(daily_injuries$weekday, daily_injuries$injuries_day)
#daily_injuries[outlier,]
```

This plot shows how many accidents occured on certain days of the week. There isn't a huge variation in the number of injuries per weekday. However, if we were forced to make a distinction, one could say that the lower injury-frequency days are Monday, Tuesday, Thursday, and Saturday, while the higher frequency days are Wednesday, Friday, and Sunday. Because there is no logical distinction between Monday, Tuesday, wednesday and Thursday, we are not very keen on grouping the days this way.

Investigating Injuries at different times of the day

```
#count the total injuries over time of days
injuries_ct = 0*c(1:length(times))
time_index = 1
for (time in times){
  injuries_time = subset(accident, TIME == time)
  injuries_ct[time_index] =  sum(injuries_time$NUMBER.OF.PERSONS.INJURED)
  time_index = time_index + 1
}

#plot the injuries over days
time_injuries = data.frame(time = times, injuries = injuries_ct)
plot(injuries ~ times, time_injuries)
lines(injuries ~ times, time_injuries, type = "l")
```




```
#outlier = identify(time_injuries$time, time_injuries$injuries)
#time_injuries[outlier,]
```

Looking at the number of injuries at each hour, the first major patter we notice are the spikes at the top and halfway point of every hour. This is because when people report accidents, they usually round to the nearest hour, half-hour, or quarter-hour mark. We Also see regions with different average injuries. From 1am to 7am, we see a lower number of injuries, from 7am to 8am we see the number of injuries starting to increase, from 8am to 1pm, and, while from 1pm to 8pm, we see the most number of injuries, and from 8pm to 1am a dropping in injuries. If we were to make three major time categorizations, we would say early monring(1 am

to 7am) has a low number of injuries, morning to afternoon (8am to 1pm) as well as late evening (8pm to 1am) has a moderate number of injuries, and afternoon(1pm to 8pm) has the highest number of injuries. This makes sense because 1-8pm is when people rush home from work.