Data Analysis II Term Paper – NYC Traffic Collision

Ancheng Deng

May 13, 2018

Abstract

Traffic is among the biggest causes of injuries or deaths in today's mega cities. With Google map, people are feeling more convenient travelling but not more secured, for that there hasn't been one product that can alert people about potential traffic accidents on streets they are driving through. Now I'd like to propose a Traffic Warning System to help predict on the potential risk of traffic accident in streets of Brooklyn, New York.

I collected datasets from Google Bigquery Public Dataset and NYC Open Data Website. Generalized Linear Mixed Effect Model is considered for predicting the probability of traffic accidents based on **time**, **traffic speed**, **weather**, **road construction** and **311 service report**. After evaluating the result of preliminary GLM model, I find **speed** to be insignificant to the model, while **weather** condition is of much significance in predicting occurrence of traffic accidents. With the final model I obtain 21.3% power with about 10% type I error rate on predicting occurrence of traffic accidents on 1023 streets of Brooklyn in testing set.

Introduction

Domain Problem

New York City Council passed a local law in 2011 that allows the collection and publication of NYC traffic collision data, which enables researchers to study the temperal and spatial information of the traffic accidents. Based upon this data, I'd like to consider using GLMM models I learnt from class to perform on such datasets and see whether it can be used for real-life purpose.

The collision datset itself is not adequate for building the model. After searching throughout the internet for other available metrices, I found the other datasets: **NYC road construction record**, **NYC 311 service log** and **historical weather in Brooklyn**. These datasets are useful because they directly relate to traffic conditions and they contains time and location information, i.e. can be matched to the traffic collision dataset.

Given the time and range of this paper, I filter the collision dataset from 2017-03-01 to 2018-01-21. The reason for the end date of dataset is due to both the incomplete dataset from Google Bigquery (they claim that the data is up-to-date but in fact it is not) and the intersection of multiple datasets (I want most of observations contained all the metrices). Finally the cleaned dataset contain 25796 observations with 51 variables.

Variable Selection & Tranformation

First I'd like to take a look at cleaned dataset.

```
## [1] "Variables in the Cleaned Dataset:"
```

```
##  [1] "date"                           "unique_key"
##  [3] "timestamp"                      "dayofweek"
##  [5] "dayofyear"                      "time"
##  [7] "year"                           "quarter"
##  [9] "month"                          "day"
## [11] "hour"                          "latitude"
## [13] "longitude"                     "location"
## [15] "cross_street_name"             "off_street_name"
## [17] "on_street_name"                "zip_code"
## [19] "contributing_factor_vehicle_1" "contributing_factor_vehicle_2"
## [21] "contributing_factor_vehicle_3" "vehicle_type_code1"
## [23] "vehicle_type_code2"            "vehicle_type_code_3"
## [25] "number_of_cyclist_injured"     "number_of_cyclist_killed"
## [27] "number_of_motorist_injured"    "number_of_motorist_killed"
## [29] "number_of_pedestrians_injured" "number_of_pedestrians_killed"
## [31] "number_of_persons_injured"     "number_of_persons_killed"
## [33] "Construction"                  "RainFall"
## [35] "SnowDepth"                     "Average_Temp"
## [37] "Max_temp"                      "Min_temp"
## [39] "Fog"                           "Heavy Fog"
## [41] "Thumder"                       "Ice_Pellets"
## [43] "Hail"                          "Rime"
## [45] "Smoke"                         "Borrowing"
## [47] "Demage_Win"                    "Date"
## [49] "Call"                          "Complaint_Type"
## [51] "Location_Type"
```

There are many time-related variables in the cleaned dataset, such as **date**, **timestamp** and **hour** (hour in a day), followed by longitude and latitude information. There are also street names which are standardized by the NYC Geographic Online Address Translator (GOAT). **on__street__name**, **cross__street__name** and **off__street__name** correspond to where the accident happened (from cross_street to off_street). **contributing__factor__vechicle__n** represent the cause of the accident for the main vehicle involved, such as Driver Inattention, Failure to Yield Right of Way. Those terms are standardized. **Construction** is a logical variables indicating whether there is road construction undergoing at the same street during the time of accident. Variables 34-47 stand for weather information, including temperature, rain fall (in mm), show depth, and extreme weather occurrence (logical). The last 3 variables **Call**, **Complaint__Type** and **Location__Type** is the 311 service call made by NYC citizens about municipal concerns (including traffic concerns).

It's flexible to choose our response variable. First I consider using bivariate response variable indicating whether people are injured in the accident, but find data to be very unbalanced (very few cases have casuality). Then I switch to using **contributing__factor__vehicle__1** and create a bivariate response variable **y** which is 1 when the main cause of the accident is NOT direcly driver behavior or car condition related. On the contrary, **y** = 0 when there are direct contribution of the accident by driver's behavior or bad car conditions (for example, DUI, brakes defective or steering failure). Eventually, I create the response variable to have 9529 TRUE cases compare to 16267 FALSE.

Exploratory Analysis

First I'd like to explore where does those traffic collision happened. Both point map and density map are performed to visulize the accident in Brooklyn borough. In order the draw the density map, I use cumulative dataset to better show the trend of the collision cumulation as well as better pinpoint the hot zone of traffic collision. *Sample Collision Information Map on 2017-3-22* indicates the accident importance and number of people injured in the traffic accidents happen on a single day of Mar22 2017. Note that the weather condition is snow, which I'll show later to be a significant factor to traffic accidents.

The *Sample Density Map of Cumulative Collisions till 2017-12-01* shows the cumulative collision density from 2017-03-01 to 2017-12-01. Note that from top to bottom, Williamsburg, Altantic Avenue and Flatbush Avenue are among the highest frequency of collision accidents. Dynamic gif can be found here.

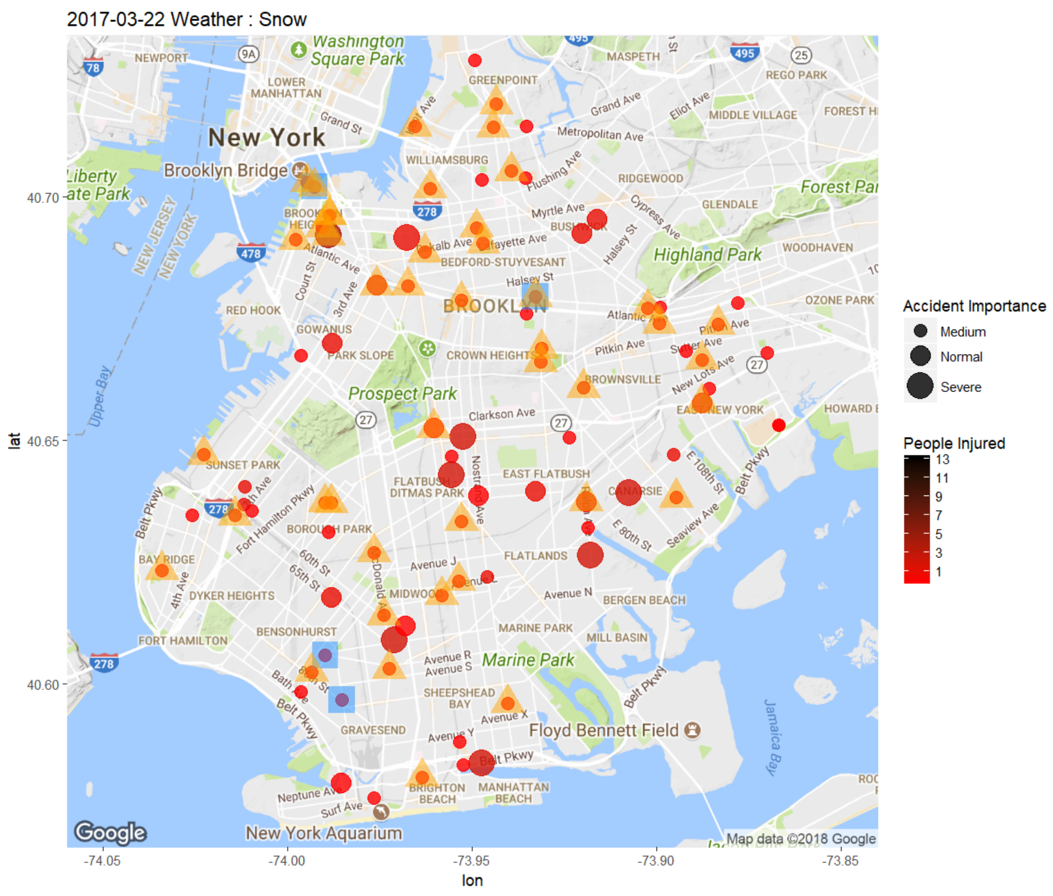
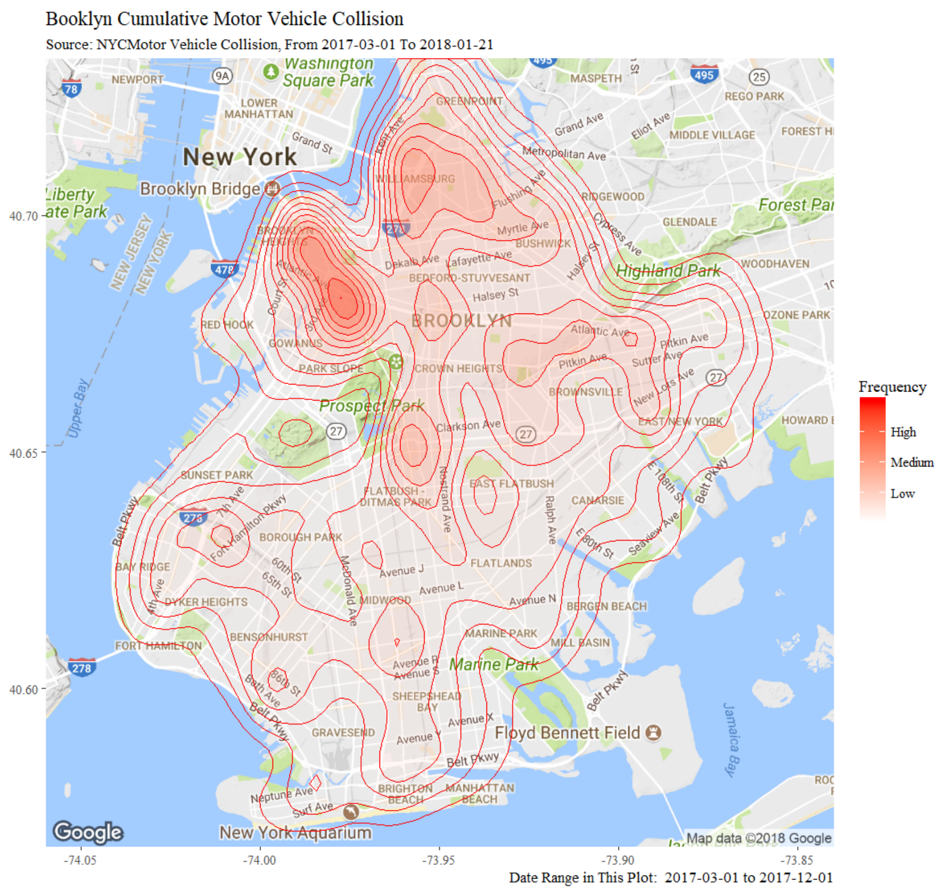


Figure 1: Sample Collision Information Map on 2017-3-22



Booklyn Cumulative Motor Vehicle Collision
Source: NYCMotor Vehicle Collision, From 2017-03-01 To 2018-01-21
Frequency
High
Medium
Low
40.70
40.65
40.60
-74.05
-74.00
-73.95
-73.90
-73.85
Date Range in This Plot: 2017-03-01 to 2017-12-01
New York
BROOKLYN
Prospect Park
Map data ©2018 Google

Figure 2: Sample Density Map of Cumulative Collisions till 2017-12-01

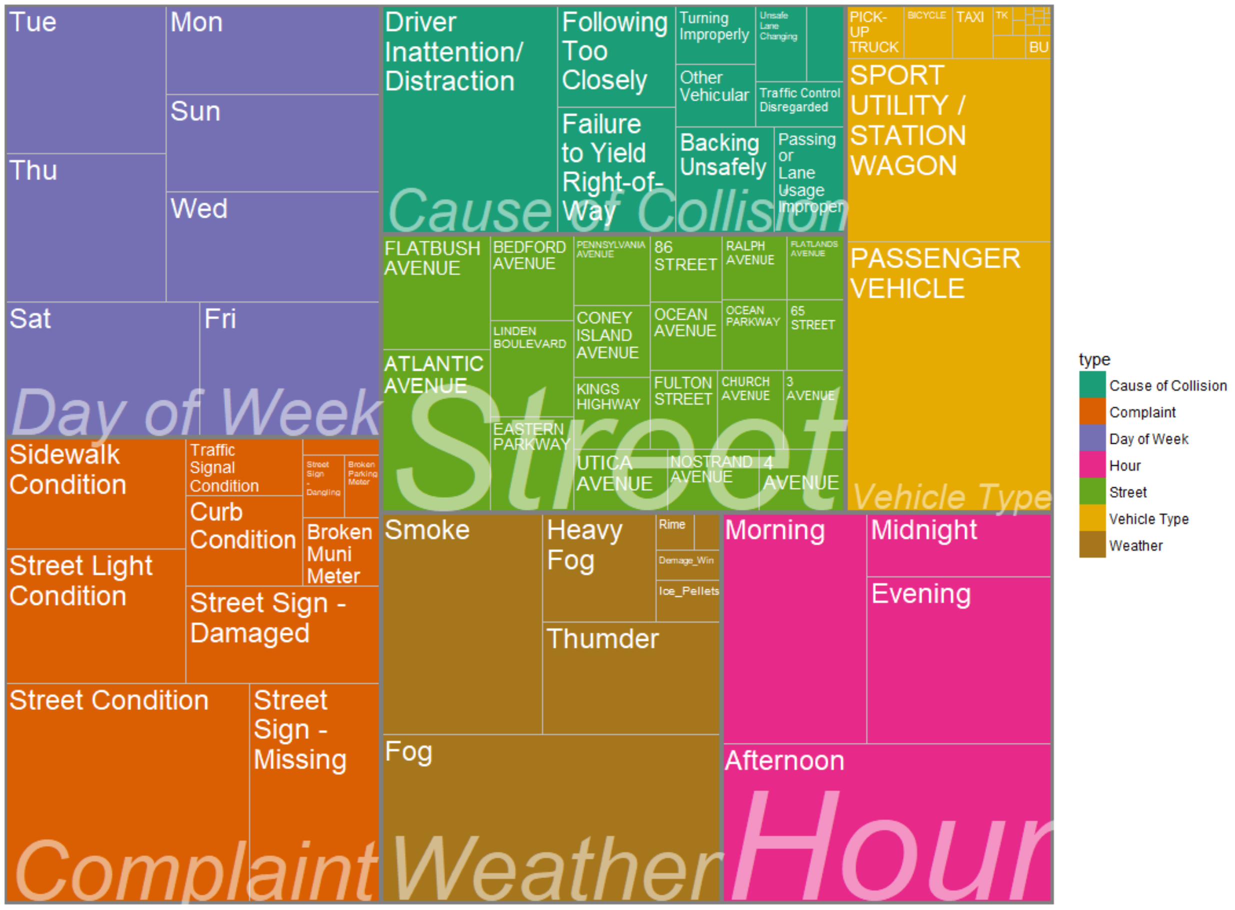


Figure 3: Treemap of Contributing Factors

Next I want to visualize the contributing factors by their appearing frequency in the collision. I use Treemaps and select contributing factors of interests: Day of Week, Hour, 311 Service Complaint Type, etc.. From the treemap we can see that the previously mentioned Altantic Avenue and Flatbush Avenue are truly the most dangerous streets in Brooklyn. We can also notice that the main reason for accidents is Driver Inattention/Distraction, and the government should consider putting more emphasis on noticing people about its danger. In terms of weather condition, most accident happened when the weather is Fog or Smoke.

Methods

Data Collection & Manipulation

In order to perform the analysis, I have to merge the collision, weather, construction information, 311 service log and speed dataset together as a single dataframe. I first downloaded the whole dataset of all metrices, each in different excel sheets. I uniform the time variable in each sheet to POSIXct (timezone = EST) and filter the dataset by time range from 2017-03-01 to 2018-01-22. Next is merging the events of collision with the metrices at the same location. I utilize the geological information (longitude and latitude) for spatial matching with precision of about 300 meters in radius. The speed datasets consists of speed information from all traffic cameras with speed detector in NYC and is recorded every 5 minutes. The speed data is stored in large excel sheets and I manage to filter them by Brooklyn borough and merge together as a single excel sheet. The ultimate speed dataset contains speed data from 18 speed detector in Brooklyn. I first match the street of collision with the camera unique linkid, then match the time within 10 minutes of the accident (use average speed). The other metrices are matched to collision events in the same way. Given the relative small amount of available speed camera in Brooklyn, the ultimate model dataset with speed information narrows down to 304 observations with 7 linkid (i.e. 7 street levels).

Preliminary Model

The model to my instinct is the one-level Generalized Linear Mixed Effect Model (GLMM):

$$\begin{aligned} logit(E[y_{ij}|\mu_i]) = \beta_0 + u_i + \beta_1 * Speed_{ij} + \beta_2 * 1\{Time_{ij} = Morning\} + \beta_3 * 1\{Time_{ij} = Afternoon\} \\ + \beta_4 * 1\{Time_{ij} = Night\} + \beta_5 * 1\{Time_{ij} = Midnight\} + \beta_6 * Rainfall_{ij} \\ + \beta_7 * Snowdepth_{ij} + \beta_8 * 1\{Weather\ Type_{ij} = Fog\} + \beta_9 * 1\{Weather\ Type_{ij} = Thunder\} \\ + \beta_{10} * 1\{Weather\ Type_{ij} = Smoke\} + \beta_{11} * 1\{Weather\ Type_{ij} = IcePellet\} \\ + \beta_{12} * 1\{Roadwork_{ij} = True\} + \beta_{13} * 1\{311Service_{ij} = True\} \end{aligned}$$

$$i: \ street, \ \ j: \ accident \ at \ street \ i, \ u_i \sim N(0, \sigma^2)$$

$$\begin{aligned} y_{ij} &= 1, \ if \ accident \ j \ caused \ by \ nondriver \ factors \ happens \ at \ street \ i \\ y_{ij} &= 0, \ otherwise \end{aligned}$$

Then I stratified-split the dataset by ratio of 8:2, write the formula into glmer() to fit on training set and see the result:

```r
load("df_model_withspeed.RData")

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 3.4.4

## Loading required package: Matrix
```

```r
# Stratified Sampling, each street 8:2 train-test split
set.seed(1)
df_test <- df_model %>% group_by(linkid) %>% sample_frac(0.2)
test_key <- df_test$unique_key
df_train <- df_model[!(df_model$unique_key %in% test_key),]
# Use training set for model building
mod.1 <- glmer(y ~ speed + factor(hour_div)  + RainFall + SnowDepth + Fog + Thumder + Smoke +
                 Ice_Pellets + (Construction=="Yes")  +
                 call4road + (1|linkid), control = glmerControl(optimizer = "bobyqa"), df_train,
               family = "binomial", nAGQ = 50)
summary(mod.1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 50) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## y ~ speed + factor(hour_div) + RainFall + SnowDepth + Fog + Thumder +
##     Smoke + Ice_Pellets + (Construction == "Yes") + call4road +
##     (1 | linkid)
##    Data: df_train
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##    352.7    401.6   -162.3    324.7      229
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9268 -0.9206 -0.6488  1.0132  1.4762
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  linkid (Intercept) 0        0
## Number of obs: 243, groups:  linkid, 7
##
## Fixed effects:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.098199   0.411154  -0.239   0.8112
## speed                      0.007087   0.008555   0.828   0.4075
## factor(hour_div)Midnight   0.246679   0.539779   0.457   0.6477
## factor(hour_div)Morning   -0.062978   0.302321  -0.208   0.8350
## factor(hour_div)Night      0.047951   0.448992   0.107   0.9149
## RainFall                   1.985542   0.957264   2.074   0.0381 *
## SnowDepth                  0.187680   0.172223   1.090   0.2758
## FogTRUE                   -0.230907   0.309801  -0.745   0.4561
## ThumderTRUE               -0.481696   0.385351  -1.250   0.2113
## SmokeTRUE                 -0.078852   0.298247  -0.264   0.7915
## Ice_PelletsTRUE            0.246364   0.772278   0.319   0.7497
## Construction == "Yes"TRUE  0.964580   1.235334   0.781   0.4349
## call4roadTRUE             -0.305558   0.280056  -1.091   0.2752
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 13 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

Note from the p-value of the fixed effect that except for **RainFall**, none of the factors are significant. Especially, the estimated coefficient for speed is 0.007, almost no effect on the model. In addition, the estimated variance for random effect is 0, which means that there are no random street level effect, and that's clearly not what I expect.

One thing worthy of further discussion is the insignificant effect of speed. Our instinct tells us that the higher speed will leave shorter reaction time to change of traffic condition, hence will be a major effect in predicting collision. However, the model tells the opposite, of which the weather condition (such as RainFall in mm unit) plays a much more important part. Possible reasons for this is:

- There are only 7 speed camera data in our training model, and they are distributed mostly in the

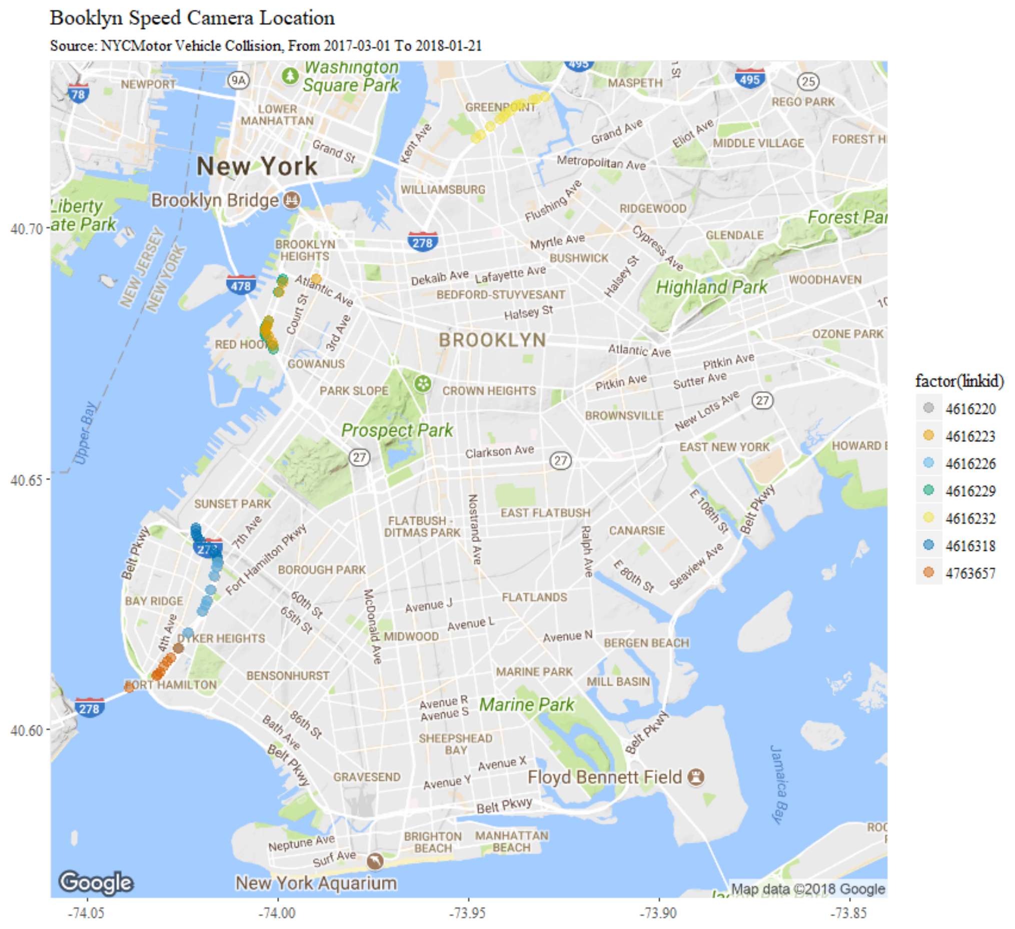


Figure 4: Brooklyn Speed Camera Location

278 express way and around the Red Hook and Greenpoint region (please see the camera distribution map below). Given the limitation of the camera location, speed on the road is reasonably not very significant. For example, in the express way, traffic speed is indeed high but also large gap between vehicles, hence drivers could have equal amount of time to respond to sudden change of traffic and will not be significantly more dangerous with a higher speed.

- The speed information is the average speed recorded by the speed camera on the road, with no direct relationship with the speed of the vehicles involved in the accident. And we know the accidents only happen in a fairly small amount of vehicles compare to the large volumn of traffic.
- There may be multicollinarity in our predictors, and we should take a look.

```
# VIF for Predictors
# First create dummy variables for Time
library(mctest)
time_dum = factor(df_model$hour_div)
dummies = model.matrix(~time_dum)
df_vif = cbind(dummies, as.matrix(df_model[,c("speed", "RainFall", "SnowDepth",
                                              "Fog", "Thumder", "Smoke", "Ice_Pellets",
                                              "Construction", "call4road")]))
df_vif[,8] <- ifelse(df_vif[,8] == "FALSE", 0, 1)
df_vif[,9] <- ifelse(df_vif[,9] == "FALSE", 0, 1)
df_vif[,10] <- ifelse(df_vif[,10] == "FALSE", 0, 1)
```

```r
df_vif[,11] <- ifelse(df_vif[,11] == "FALSE", 0, 1)
df_vif[,12] <- ifelse(df_vif[,12] == "NO", 0, 1)
df_vif[,13] <- ifelse(df_vif[,13] == "FALSE", 0, 1)
df_vif <- df_vif[,-1]

df_vif<- apply(df_vif, 2, as.numeric)

omcdiag(df_vif, as.numeric(df_model$y))
```

```
##
## Call:
## omcdiag(x = df_vif, y = as.numeric(df_model$y))
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.3511         0
## Farrar Chi-Square:       314.6253         1
## Red Indicator:             0.1253         0
## Sum of Lambda Inverse:    14.2751         0
## Theil's Method:            1.4489         1
## Condition Number:          8.3896         0
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

Modified Model

I then decide to remove the **speed** variable from the dataset. By removing it I can then make use of a way larger datset (of 25796 observations) with other factors such as **weather** and **construction**. Then we also do the stratified-split of the dataset by **on_street_name** and further assign random intercept to each street in our one-level generalized linear mixed effect model.

$$
\begin{aligned}
logit(E[y_{ij}|\mu_i]) = \beta_0 + u_i + \beta_1 * 1\{Time_{ij} = Morning\} + \beta_2 * 1\{Time_{ij} = Afternoon\} + \beta_3 * 1\{Time_{ij} = Night\} \\
+ \beta_4 * 1\{Time_{ij} = Midnight\} + \beta_5 * Rainfall_{ij} + \beta_6 * Snowdepth_{ij} + \beta_7 * 1\{Weather\ Type_{ij} = Fog\} \\
+ \beta_8 * 1\{Weather\ Type_{ij} = Thunder\} + \beta_9 * 1\{Weather\ Type_{ij} = Smoke\} \\
+ \beta_{10} * 1\{Weather\ Type_{ij} = IcePellet\} + \beta_{11} * 1\{Roadwork_{ij} = True\} \\
+ \beta_{12} * 1\{311Service_{ij} = True\}
\end{aligned}
$$

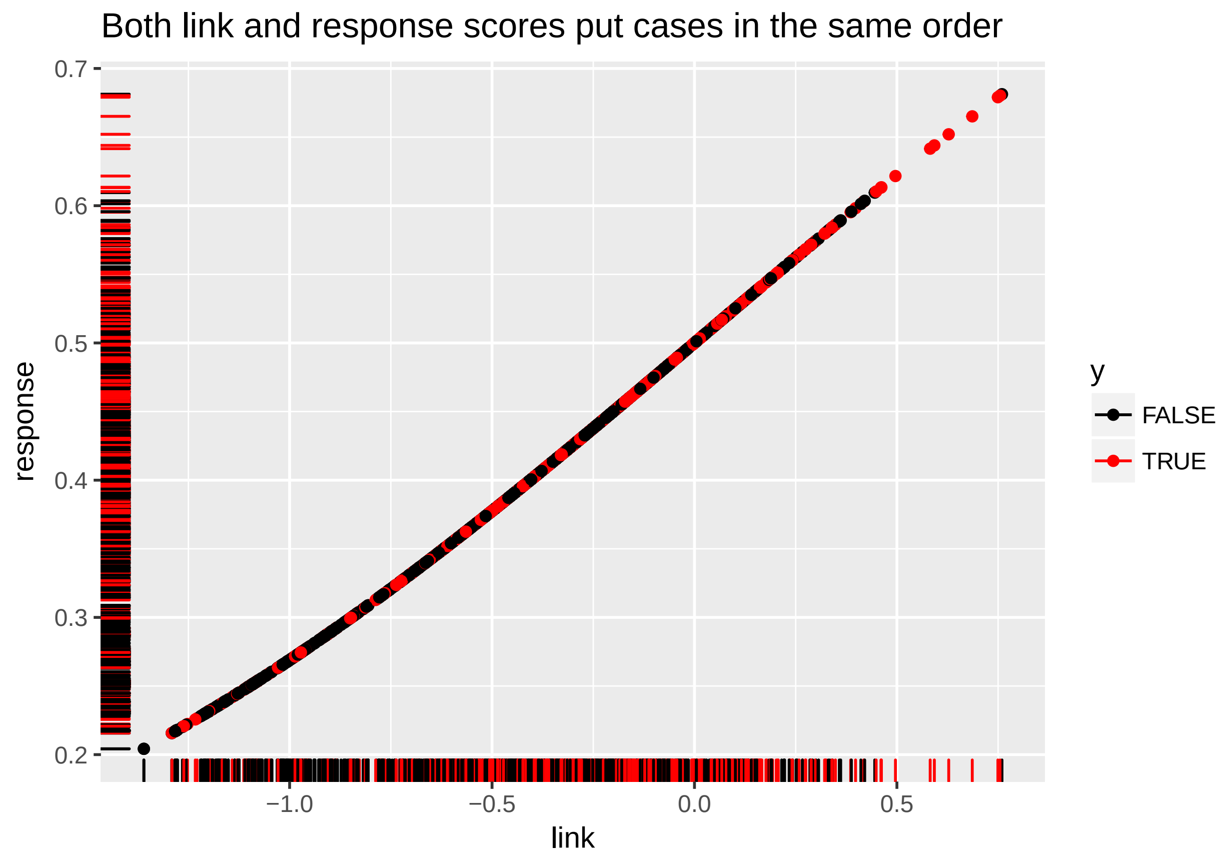
```r
load("df_model_correct.RData")
# Stratified Sampling
set.seed(1)
df_test <- df_model %>% group_by(on_street_name) %>% sample_frac(0.1)
test_key <- df_test$unique_key
df_train <- df_model[!(df_model$unique_key %in% test_key),]

mod.2 <- glmer(y ~ factor(hour_div)  + RainFall + SnowDepth + Fog + Thumder + Smoke +
                 Ice_Pellets  + (Construction=="Yes")  +
                 call4road +   (1|on_street_name), control = glmerControl(optimizer = "bobyqa"), df_tra
```

```
               family = "binomial", nAGQ = 50)
summary(mod.2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 50) [glmerMod]
##  Family: binomial  ( logit )
## Formula: y ~ factor(hour_div) + RainFall + SnowDepth + Fog + Thumder +
##     Smoke + Ice_Pellets + (Construction == "Yes") + call4road +
##     (1 | on_street_name)
##    Data: df_train
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  29871.5  29976.3 -14922.8  29845.5    23294
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4546 -0.7894 -0.5834  1.1018  2.0419
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  on_street_name (Intercept) 0.1437   0.3791
## Number of obs: 23307, groups:  on_street_name, 1023
##
## Fixed effects:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.352620   0.039628  -8.898  < 2e-16 ***
## factor(hour_div)Midnight  -0.189438   0.048542  -3.903 9.52e-05 ***
## factor(hour_div)Morning   -0.107811   0.033678  -3.201 0.001368 **
## factor(hour_div)Night      0.012758   0.039409   0.324 0.746140
## RainFall                  -0.018646   0.050831  -0.367 0.713747
## SnowDepth                  0.066617   0.013212   5.042 4.60e-07 ***
## FogTRUE                    0.121616   0.032065   3.793 0.000149 ***
## ThumderTRUE               -0.010667   0.038143  -0.280 0.779737
## SmokeTRUE                  0.073334   0.031372   2.338 0.019410 *
## Ice_PelletsTRUE           -0.283226   0.099585  -2.844 0.004454 **
## Construction == "Yes"TRUE -0.185226   0.092260  -2.008 0.044680 *
## call4roadTRUE              0.002982   0.029691   0.100 0.920002
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) fctr(hr_dv)Md fctr(hr_dv)Mr fc(_)N RanFll SnwDpt
## fctr(hr_dv)Md -0.237
## fctr(hr_dv)Mr -0.356  0.286
## fctr(hr_d)N   -0.297  0.247         0.351
## RainFall       0.020 -0.009         0.004        -0.003
## SnowDepth     -0.146  0.012         0.003         0.009 -0.048
## FogTRUE       -0.279 -0.003        -0.005        -0.003 -0.327  0.072
## ThumderTRUE   -0.053  0.011         0.006        -0.002 -0.189  0.053
## SmokeTRUE     -0.127  0.003        -0.002        -0.001  0.028 -0.093
## Ic_PlltTRUE    0.026 -0.007        -0.004        -0.009 -0.051 -0.380
## C=="Ys"TRUE   -0.094  0.007         0.011         0.017 -0.005  0.016
## call4rdTRUE   -0.461 -0.004        -0.001        -0.003 -0.017  0.102
##                  FgTRUE ThTRUE SmTRUE I_PTRU C=="Y"
## fctr(hr_dv)Md
## fctr(hr_dv)Mr
## fctr(hr_d)N
## RainFall
## SnowDepth
## FogTRUE
## ThumderTRUE   -0.159
## SmokeTRUE     -0.260 -0.030
## Ic_PlltTRUE   -0.070  0.008 -0.068
## C=="Ys"TRUE   -0.008 -0.011  0.002 -0.001
## call4rdTRUE   -0.006 -0.046 -0.007  0.014 -0.013
```

```r
library(ggplot2)
# Visualization on the link-response relation
score_data <- data.frame(link = predict(mod.2, newdata = df_test, type = "link"),
                         response = predict(mod.2, newdata = df_test, type = "response"),
                         y = df_test$y,
                         stringsAsFactors = FALSE)
score_data %>%
  ggplot(aes(x=link, y=response, col=y)) +
  scale_color_manual(values=c("black", "red")) +
  geom_point() +
  geom_rug() +
  ggtitle("Both link and response scores put cases in the same order")
```



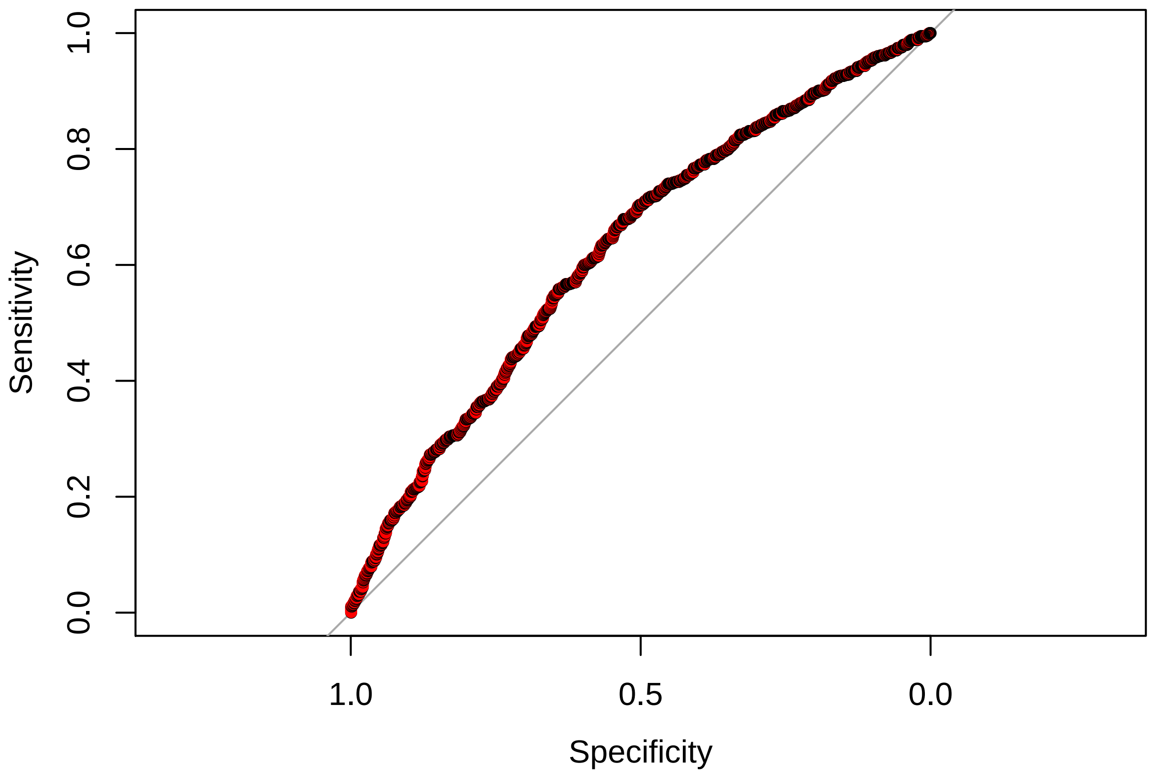
```r
library(pROC)

simple_roc <- function(labels, scores){
  labels <- labels[order(scores, decreasing=TRUE)]
  data.frame(TPR=cumsum(labels)/sum(labels), FPR=cumsum(!labels)/sum(!labels), labels)
}

# ROC for Testing set
glm_simple_roc <- simple_roc(df_test$y==TRUE, predict(mod.2, newdata = df_test, type = "link"))
(roc_result <- roc(df_test$y, predict(mod.2, newdata = df_test, type = "response"), direction="<"))
```

```
##
## Call:
## roc.default(response = df_test$y, predictor = predict(mod.2,     newdata = df_test, type = "response
##
## Data: predict(mod.2, newdata = df_test, type = "response") in 1554 controls (df_test$y FALSE) < 935
## Area under the curve: 0.6268
```

```r
plot(roc_result, col="yellow", lwd=3)
with(glm_simple_roc, points(1 - FPR, TPR, col=1 + labels, cex = 0.7))
```



```r
# prediction on df_test
y_pred = predict(mod.2, newdata = df_test, type = "response") > 0.47 # decision criteria
(tab_eval = table(y_pred, df_test$y))
```

```
##
## y_pred  FALSE TRUE
##   FALSE  1371  731
##   TRUE    183  204
```

```
(alpha = tab_eval[2]/(tab_eval[1] + tab_eval[2]))
```

```
## [1] 0.1177606
```

```
(power = tab_eval[4]/(tab_eval[3] + tab_eval[4]))
```

```
## [1] 0.2181818
```

Do a *Wald's Test* to know if we are safe to remove the non-significant factors in the model. If OK, then we further simplify the model:

```
mod.3 <- glmer(y ~ factor(hour_div)  + SnowDepth + Fog + Smoke +
                 Ice_Pellets   + (Construction=="Yes")  +
                 (1|on_street_name), control = glmerControl(optimizer = "bobyqa"), df_train,
               family = "binomial", nAGQ = 50)
summary(mod.3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 50) [glmerMod]
##  Family: binomial  ( logit )
## Formula: y ~ factor(hour_div) + SnowDepth + Fog + Smoke + Ice_Pellets +
##     (Construction == "Yes") + (1 | on_street_name)
##    Data: df_train
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  29865.8  29946.4 -14922.9  29845.8    23297
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4560 -0.7903 -0.5828  1.1011  2.0478
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  on_street_name (Intercept) 0.1435   0.3788
## Number of obs: 23307, groups:  on_street_name, 1023
##
## Fixed effects:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -0.35163    0.03504 -10.035  < 2e-16 ***
## factor(hour_div)Midnight  -0.18943    0.04854  -3.903 9.51e-05 ***
## factor(hour_div)Morning   -0.10768    0.03368  -3.198  0.00139 **
## factor(hour_div)Night      0.01269    0.03941   0.322  0.74738
## SnowDepth                  0.06649    0.01311   5.070 3.97e-07 ***
## FogTRUE                    0.11527    0.02943   3.916 8.98e-05 ***
## SmokeTRUE                  0.07339    0.03135   2.341  0.01923 *
## Ice_PelletsTRUE           -0.28526    0.09949  -2.867  0.00414 **
## Construction == "Yes"TRUE -0.18569    0.09223  -2.013  0.04409 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                 (Intr) fctr(hr_dv)Md fctr(hr_dv)Mr fc(_)N SnwDpt FgTRUE
## fctr(hr_dv)Md -0.269
## fctr(hr_dv)Mr -0.403  0.286
## fctr(hr_d)N   -0.337  0.247         0.351
## SnowDepth     -0.108  0.012         0.003         0.010
## FogTRUE       -0.363 -0.004        -0.002        -0.005  0.075
## SmokeTRUE     -0.149  0.004        -0.001        -0.001 -0.090 -0.280
## Ic_PlltTRUE    0.037 -0.008        -0.004        -0.009 -0.388 -0.094
## C=="Ys"TRUE   -0.114  0.007         0.011         0.017  0.018 -0.014
##               SmTRUE I_PTRU
## fctr(hr_dv)Md
## fctr(hr_dv)Mr
## fctr(hr_d)N
## SnowDepth
## FogTRUE
## SmokeTRUE
## Ic_PlltTRUE   -0.067
## C=="Ys"TRUE    0.002 -0.001
```

Now we predict on the testing set

```
y_pred = predict(mod.2, newdata = df_test, type = "response") > 0.47 # decision criteria
(tab_eval = table(y_pred, df_test$y))
```

```
##
## y_pred  FALSE TRUE
##   FALSE  1371  731
##   TRUE    183  204
```

```
(alpha = tab_eval[2]/(tab_eval[1] + tab_eval[2]))
```

```
## [1] 0.1177606
```

```
(power = tab_eval[4]/(tab_eval[3] + tab_eval[4]))
```

```
## [1] 0.2181818
```

Interpretation of the Result