Don't Break a Leg! Road Safety in the City of Toronto

STA2453 - Project II Draft

Sunwoo (Angela) Kang, Sergio E. Betancourt, Jing Li, and Jiahui (Eddy) Du

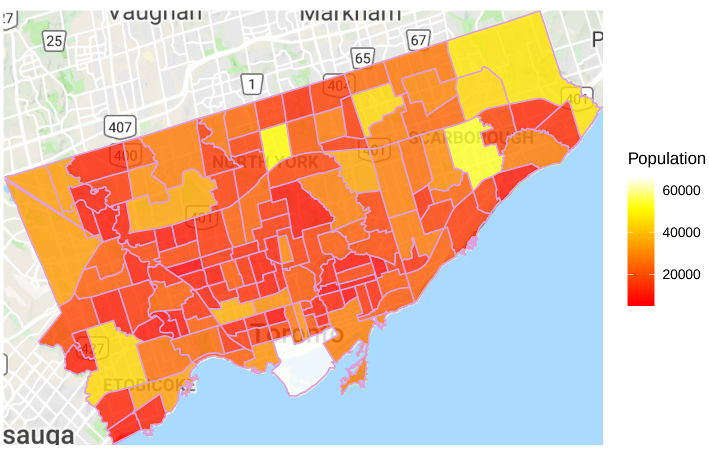
2019-03-02

Introduction

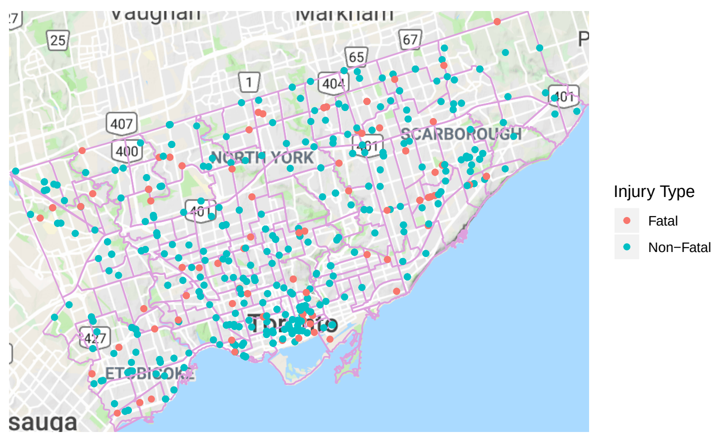
Road traffic safety is a crucial component of urban planning and development. Nowadays governments (and sometimes the private sector) dedicate significant resources to providing ample and sufficient infrastructure to accommodate diverse modes of transportation, thereby increasing the productivity of any given urban area. In this project we examine road safety in the City of Toronto from 2007 to 2017 and explore the areas with highest risk of a traffic incident, controlling for different factors.

Methods

We define the City of Toronto as per the these guidelines (https://www.toronto.ca/city-government/data-research-maps/neighbourhoods-communities/neighbourhood-profiles/). Below are the neighborhood limits and the 2016 population estimates:



(a) Population by neighborhood in the census year 2016



(b) Fatal and Non-Fatal Collisions in 2016

Figure 1: EDA with regards to the City of Toronto

Primary Questions

The analysis focuses on answering two main questions:

1. Given a collision occurred which areas in Toronto are the most deadly controlling for other factors?
2. Which factors are related to the collision safety of neighbourhoods?

Data Collection

For our analysis we employed data from the Toronto Police Service, the City of Toronto, and Environment Canada. Each of these datasets contains different levels of granularity and information, and were therefore combined to obtain the following variables of interest outlined in **Appendix: Dataset Variables and Definitions**.

Data Preparation

The following table provides an overview of the merged data.

Accident_ID	Fatal	Date	Neighborhood	Population	Max_Temp
5002235651	1	2015-12-30	Greenwood-Coxwell	7072	4.7
5000995174	1	2015-06-13	Annex	26703	22.3
5000995174	1	2015-06-13	Annex	26703	22.3
1249781	0	2011-08-04	Bay Street Corridor	19348	26.4

Traffic incident information provided by Toronto Police served as a base for the data used for this analysis. Each of the 12,557 entries represent a party involved in a traffic collision event where a person was either killed or seriously injured. Other features such as the location of the collision (intersection, neighborhood, ward), road condition (visibility, road precipitation), driver action (e.g. speeding, involved alcohol), and type of vehicles (e.g. automobile, pedestrian, cyclist) involved were also used.

Population counts for 2011 and 2016 are available through the national census for each neighborhood. The populations for the dates not provided by the census were extrapolated using a linear growth model.

Historical weather data collected from the station in University of Toronto was also merged based on the day the accident occurred.

Exploratory Analysis

By summing up counts from 2007 to 2017, West Humber-Clairville appears to be the deadliest intersection followed by South Parkdale, then Wexford/Maryvale. Thankfully, the fatalities appear to be quite low compared to the total number of collisions reported by the Toronto Police.

Neighbourhood	Total Fatalities	Total Collisions
West Humber-Clairville	22	426
South Parkdale	21	197
Wexford/Maryvale	15	225
Clairlea-Birchmount	14	193
Waterfront Communities-The Island	14	492
Glenfield-Jane Heights	11	105

West Humber-Clairville, and Wexford/Maryvale appear again as a dangerous neighborhood even when focussing on pedestrian or cyclist fatalities.

Neighbourhood	Total Pedestrian Fatalities	Total Pedestrian Collisions
Clairlea-Birchmount	11	39
Wexford/Maryvale	10	45
Moss Park	9	35
West Humber-Clairville	9	43
Newtonbrook West	8	27
Waterfront Communities-The Island	8	87

Neighbourhood	Total Cyclist Fatalities	Total Cyclist Collisions
South Parkdale	3	9
Dovercourt-Wallace Emerson-Junction	2	10
Kensington-Chinatown	2	19
Wexford/Maryvale	2	4
Annex	1	14
Bay Street Corridor	1	25

Neighbourhood	Total Other Fatalities	Total Other Collisions
South Parkdale	14	163
West Humber-Clairville	12	376
Islington-City Centre West	8	198
Glenfield-Jane Heights	6	82
Don Valley Village	5	73
Downsview-Roding-CFB	5	150

Modeling

We model our outcome of interest (fatal collision) using a generalized mixed effects model (to be expanded to spatial in the following iteration), clustered by neighborhood. We estimate the odds of experiencing a fatal accident with respect to experiencing a non-fatal one, accross neighborhoods in Toronto, controlling for each day's total precipitation and minimum temperature. We will continue exploring model complexity and structure for the next iteration.

Bayesian Mixed-Effects Semi-parametric Logit Model

Mixed effects logistic regression is used to model binary outcome variables, in which the log odds of the outcomes are modeled as a linear combination of the predictor variables when data are clustered or there are both fixed and random effects. A mixed effect model is used to describe the binomial probability of an auto accident resulting to fatality. Each neighbourhood has its own random intercept and we account for the progression of time through a semi-parametric term.

$$Y_{ijt} \sim \text{bernoulli}(\pi_{ijt}) \tag{1}$$

$$\text{logit}(\pi_{ijt}) = X_{ijt}\beta + U_i + f(W_{ijt}) \tag{2}$$

$$U_i \sim N(0, \sigma_U^2) \tag{3}$$

$$W_{ij(t+1)} - W_{ij(t)} \sim N(0, \sigma_W^2) \quad \text{(RW1)} \tag{4}$$

The fixed effects of this model contains are *visibility*, *types of road*, *traffic control* and *Precipitation.* Those covariantes used in the model are unrelated to the personel involved in the accidents, so factors such as condition of the drivers are not included.

- The covariante *visibility* was binarized to either "Clear" or "Not Clear", "Clear" was used as reference.
- For covariante *types of road*, "Major Arterial", "Major Arterial Ramp" and "Minor Arterial" were grouped into "Arterial"; "Expressway", "Expressway Ramp" were grouped into "expressway"; "Local", "Laneway" were grouped into "Local", where "Local" was used as reference.
- For covariante *traffic control*, "School Guard", "Police Control", "Traffic Controller" were grouped into "Human Control", and since there is not fatal accident in "Human Control", all records under "Human Control" were removed to avoid spiked estimate."Stop Sign", "Yield Sign", "Traffic Gate" were grouped into "Traffic Sign" and "Pedestrian Crossover", "Streetcar (Stop for)" were grouped into "Pedestrian Crossing". "No Traffic Control" was used as reference.

Below are the observations from table of estimates:

- The odds of having fatality are higher when driving on highway.
- Having traffic signage and traffic light leads to a lower odds, compared to no control.
- Accidents without pedestrian (vehicle to vehicle) involved has lower odds of having fatality
- The odds of fatality are slightly lower when there is more precipitation. -2% odds of fatality with 1mm of precipitation increasing. It may be due to drivers slow down their speed when they have difficulty seeing clear ahead or knowing road is slippery.

This time trend graph shows the odds of fatality rising until mid-2016 and then decreasing. This is consistent with press reports deeming 2015-2017 as bad years for Toronto in terms of fatality. The decrease post-2017 could be attributed to the Vision Zero municipal plan to address road fatalities.

Table 1: Posterior mean and 2.5 and 97.5 percentiles for the odds ratio of deadly accident by model coefficients

	mean	0.025quant	0.975quant
(Intercept)	0.115	0.078	0.168
visibilitybNot Clear	1.182	0.940	1.481
roadclassArterial	1.067	0.788	1.459
roadclassCollector	0.987	0.661	1.474
roadclassExpressway	1.737	1.023	2.934
trafficctrlPedestrian Crossing	0.871	0.447	1.619
trafficctrlTraffic Sign	0.557	0.422	0.730
trafficctrlTraffic Signal	0.538	0.462	0.628
persontypePedestrian not involved	0.638	0.542	0.752
totprecipmm	0.980	0.965	0.995
SD for weeknum	0.046	0.020	0.096
SD for weekiid	1.491	1.349	1.622
SD for hoodid	0.917	0.772	1.069

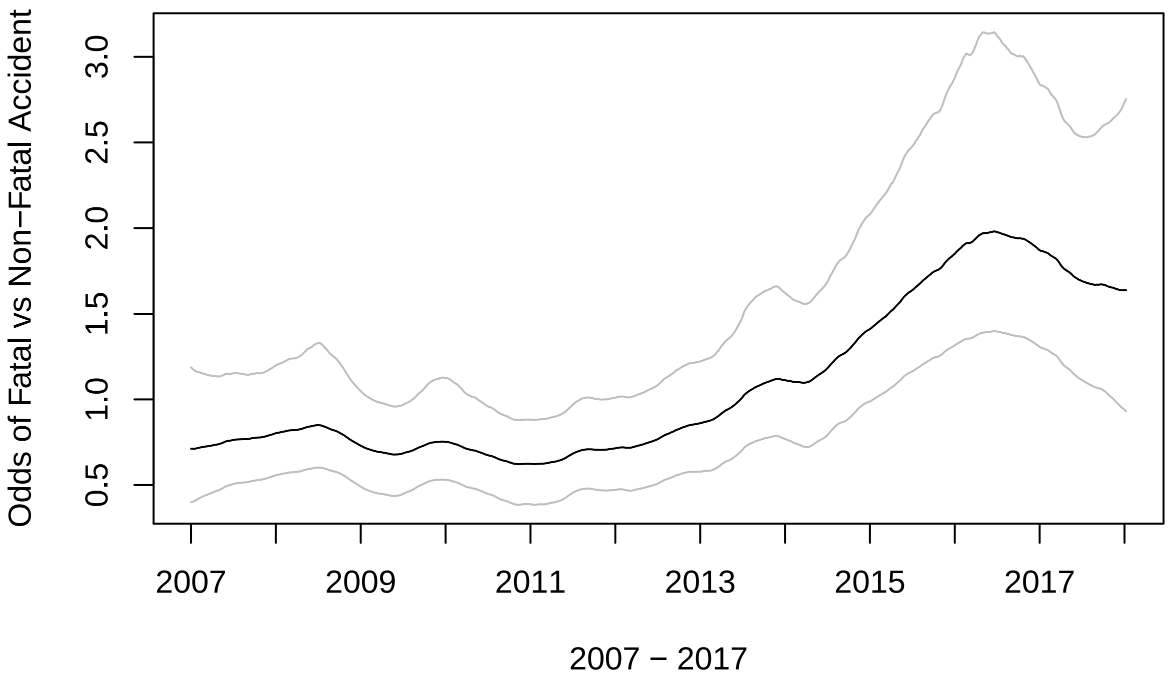
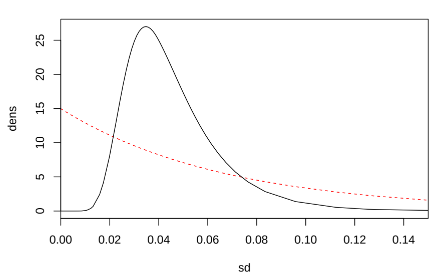
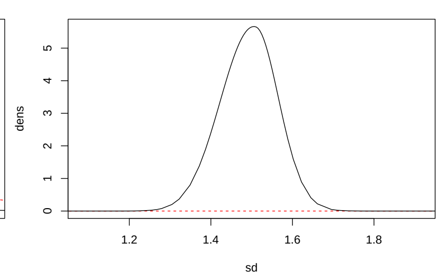


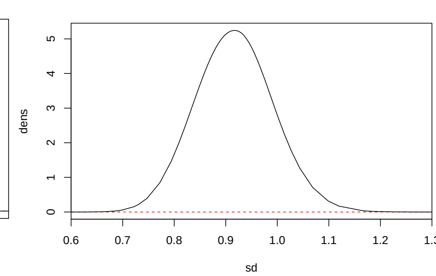
Figure 2: Plot of time trend effect of Odds of Fatality for the City of Toronto



(a) SD of time trend



(b) SD of time component



(c) SD of neighborhood effect

Figure 3: Plot of posteriors for distributions on random intercept (neighborhood) and random time components

GAM model

A semi-parametric temporal model is used to fit the total accident counts with months as factors, number of days from

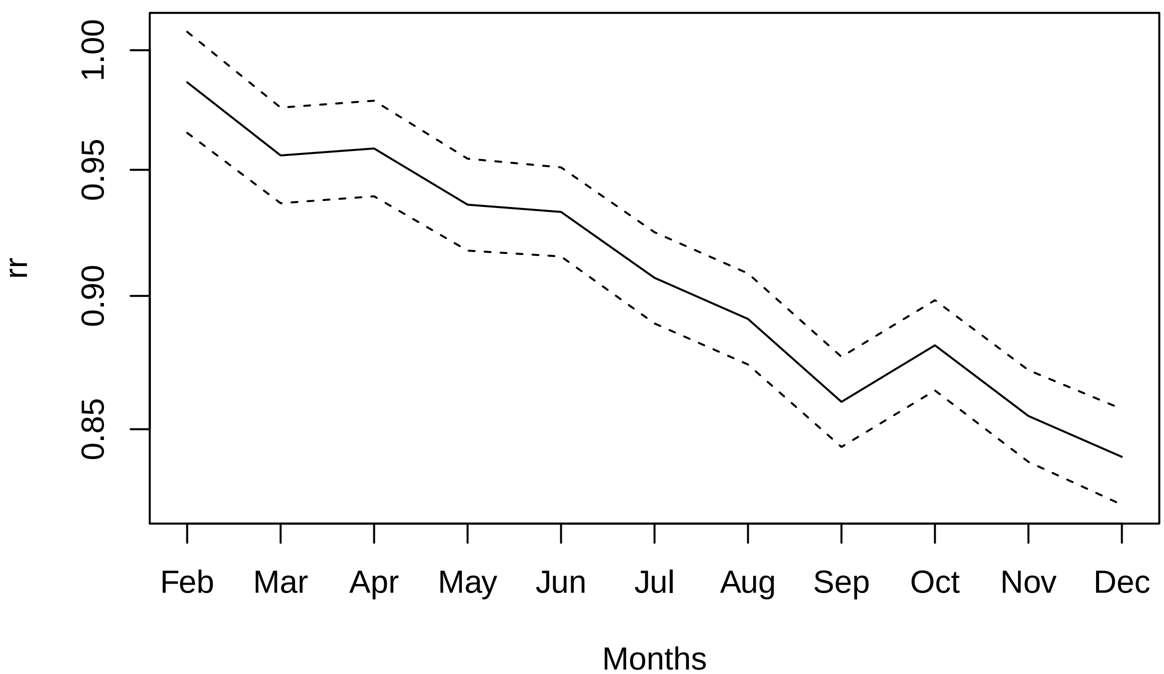
2007 as non-parametric term and neighbourhoods as random effects. Toronto population is estimated by using linear function. (linear function is estimated by using population at 2,503,281 in 2006 and at 2,731,571 in 2016.) Offset term is log of population.

$$Y_i \sim Poisson(O_i \lambda_i) \tag{5}$$

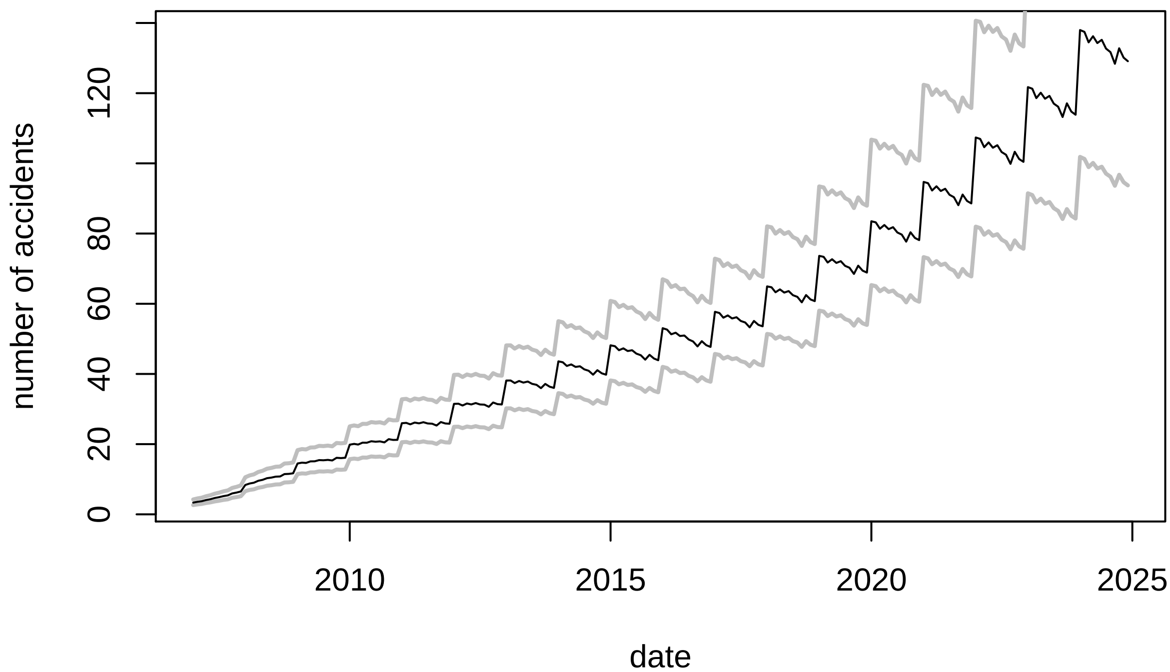
$$log(\lambda_i) = X_i \beta + f(day) + f(\mu_i) \tag{6}$$

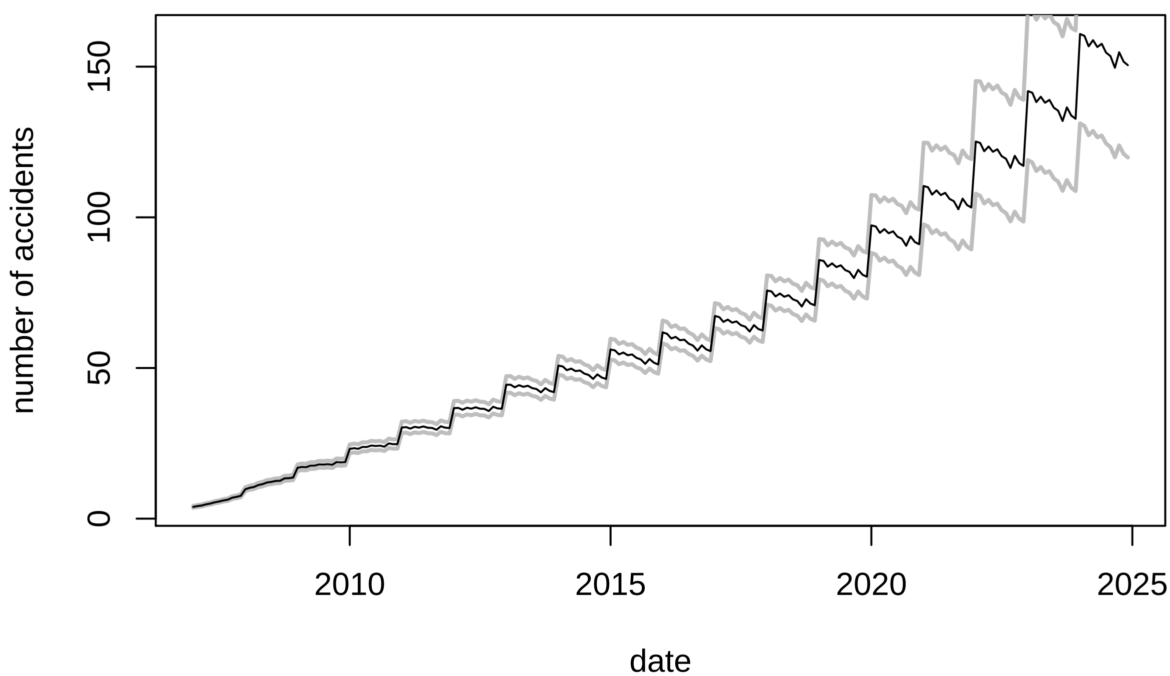
	Estimate	Std. Error
(Intercept)	-11.342	0.084
month_f02	-0.014	0.011
month_f03	-0.045	0.010
month_f04	-0.042	0.010
month_f05	-0.066	0.010
month_f06	-0.069	0.010
month_f07	-0.098	0.010
month_f08	-0.115	0.010
month_f09	-0.151	0.010
month_f10	-0.127	0.010
month_f11	-0.157	0.010
month_f12	-0.174	0.010

January has the highest rate of having accidents comparing to other months of the year. It is interesting to notice that within the winter period, only January and Feburary have such high amount of accidents. It may be due to the fact that drivers are more cautious when driving in snow days and November and Decemeber are holidays season so you may find less cars on the street.



Below is the prediction of accidents in neighbour 5 and 122





LGPC Model

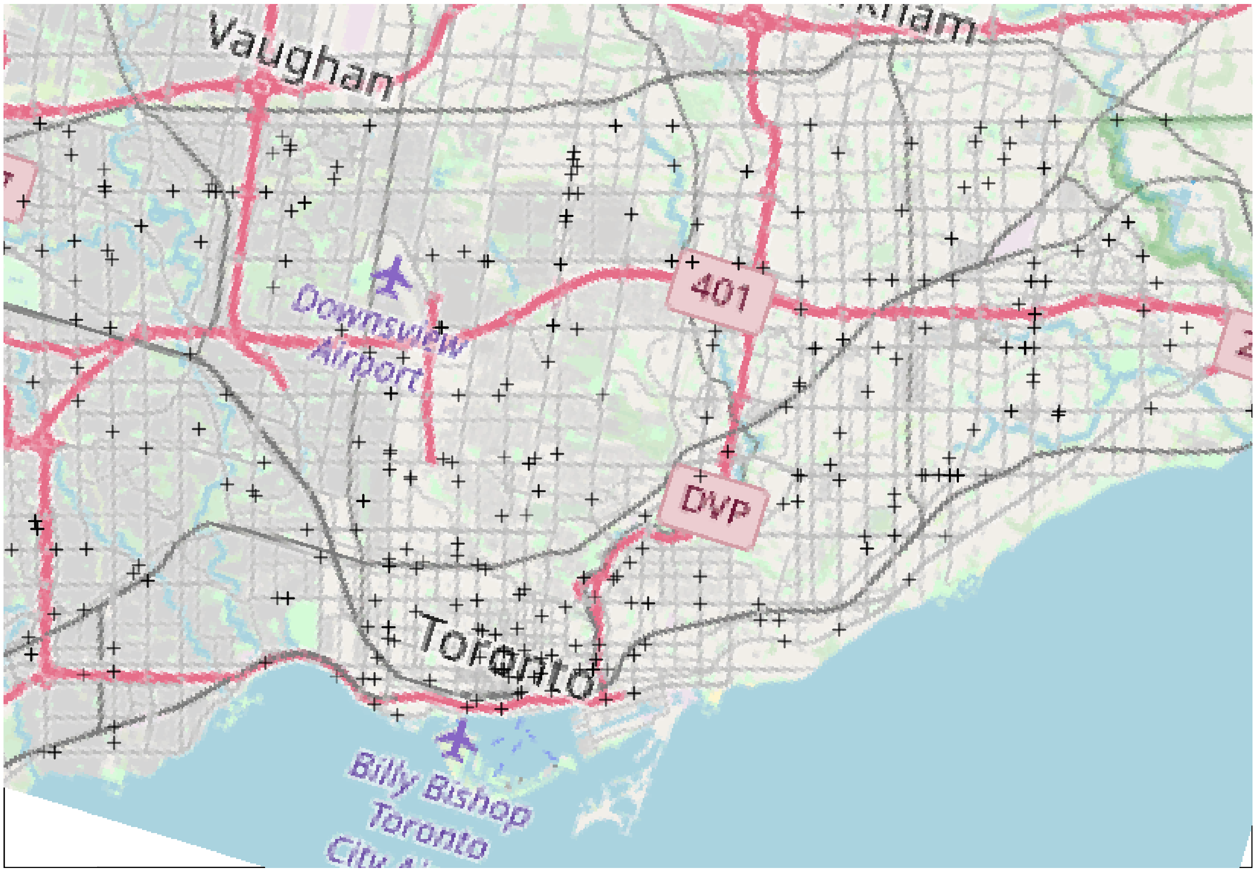
A spatial model Log Gaussian Cox Process LGCP is used to fit the accident counts in 2017 with intercept only.

$$Y_{ij} \sim N(\lambda(s_i), \tau^2) \tag{7}$$

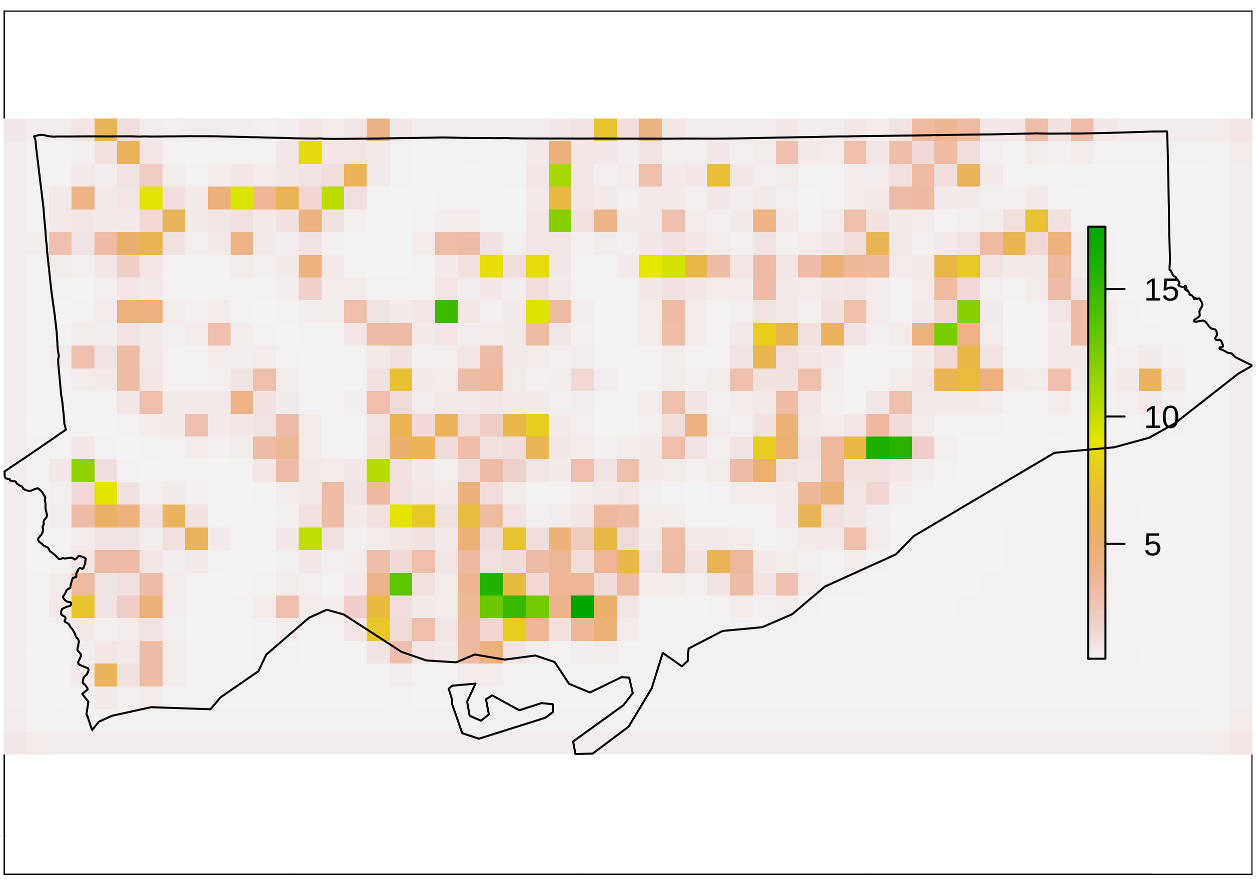
$$\lambda(s_i) = U(s) \tag{8}$$

$$cov[U(s+h), U(s)] = \sigma^2 \rho(h/\phi; v) \tag{9}$$

The plot below shows where the accidents happened in 2017 in Toronto.



The plot below is the expected value of the count (lambda). We could see there are many accidents downtown area and along the Yonge street. More traffic control may be required. We could try human control since it is the safest type of control among all.



Results

Our model indicates that a one milimeter increment of total precipitation for any neighborhood in the timeframe in question leads to an increment of 1.2% in the odds of suffering a fatal accident.

Conclusions and Discussion

One of the biggest limitations in our project has been data quality and granularity. The data made available by Geotab does not include large areas of the City of Toronto. Moreover, there are plenty missing observations. We also acknowledge the fact that the collision information we procured from the Toronto Police Service may not describe perfectly the actual number of incidents, as there are many of these that are non-fatal or go unreported.

Exploratory and Limitation

Getting spatial type of dataset is difficult. Most of the available dataset are outdated since collecting such data is expensive. At first, we tested with Geotab datasets since it seems to have enough information covering the whole Toronto. However, we failed to convert them into a proper and usable raster type data. However, this model would be useful once we have the information/covariates we want, we could use above plot to predict the expected number of accidents (and actually we can plug in many other responses into the model. Eg. Number of reported stolen cars) at places where there is no observation collected. And hence we could use this to suggest traffic control policy at certain location or to estimate insurance pricing.

Appendix: Dataset Variables and Definitions

Feature	Description	Source
YEAR	Year in range (2007-2017) inclusive	Automobile (Toronto Police)
MONTH	Month in range 1-12 inclusive	Automobile (Toronto Police)
Ward_ID	Ward in range (1-44) inclusive	Automobile (Toronto Police)
IncidentsTotal_TP	Total number of incidents	Automobile (Toronto Police)
Dark	Count accidents ocurred on dark conditions	Automobile (Toronto Police)
Dawn	Count accidents ocurred on dawn conditions	Automobile (Toronto Police)
Daylight	Count accidents ocurred on daylight conditions	Automobile (Toronto Police)
Dusk	Count accidents ocurred on dusk conditions	Automobile (Toronto Police)
Inv_PED	Count accidents involved pedstrains	Automobile (Toronto Police)
Inv_CYC	Count accidents involved cyclists	Automobile (Toronto Police)
Inv_AM	Count accidents involved automobiles	Automobile (Toronto Police)
Inv_MC	Count accidents involved motorcycles	Automobile (Toronto Police)
Inv_TC	Count accidents involved trucks	Automobile (Toronto Police)
Speeding	Count accidents ocurred on speeding condition	Automobile (Toronto Police)
Ag_Driv	Count accidents ocurred on angry driving condition	Automobile (Toronto Police)
Redlight	Count accidents ocurred with redlight	Automobile (Toronto Police)
Alcohol	Count accidents ocurred with driver with alcohol	Automobile (Toronto Police)
Disability	Count accidents ocurred with driver with disability	Automobile (Toronto Police)
SeverityScore	Average Score of Severitylevel (harsh brake)	HDA(Geotab)
IncidentsTotal_Geotab	Monthly average of total number of incidents	HDA(Geotab)
AvgAcceleration	Monthly average acceleration	RI(Geotab)
PercentOfVehicles	Monthly average on percentage of vehicles	RI(Geotab)
AvgMonthlyVolume	Monthly average on vehicle volumes	RI(Geotab)
PercentCar	Monthly average on car percentage	RI(Geotab)
PercentMPV	Monthly average on MPV percentage	RI(Geotab)
PercentLDT	Monthly average on LDT percentage	RI(Geotab)
PercentMDT	Monthly average on MDT percentage	RI(Geotab)
PercentHDT	Monthly average on HDT percentage	RI(Geotab)
PercentOther	Monthly average on other vehicle percentage	RI(Geotab)
Daily_dif	Monthly average on daily Weather change (in celsus)	Weather
Max_Temp	Monthly max on highest daily Weather degree (celsus)	Weather
Min_Temp	Monthly min on lowest daily Weather degree (celsus)	Weather
Ave_Temp	Monthly average on daily average Weather (in celsus)	Weather
Rain_vol	Monthly average on daily rain volumn	Weather
Snow_vol	Monthly average on daily snow volumn	Weather

Appendix: Neighborhoods of Toronto

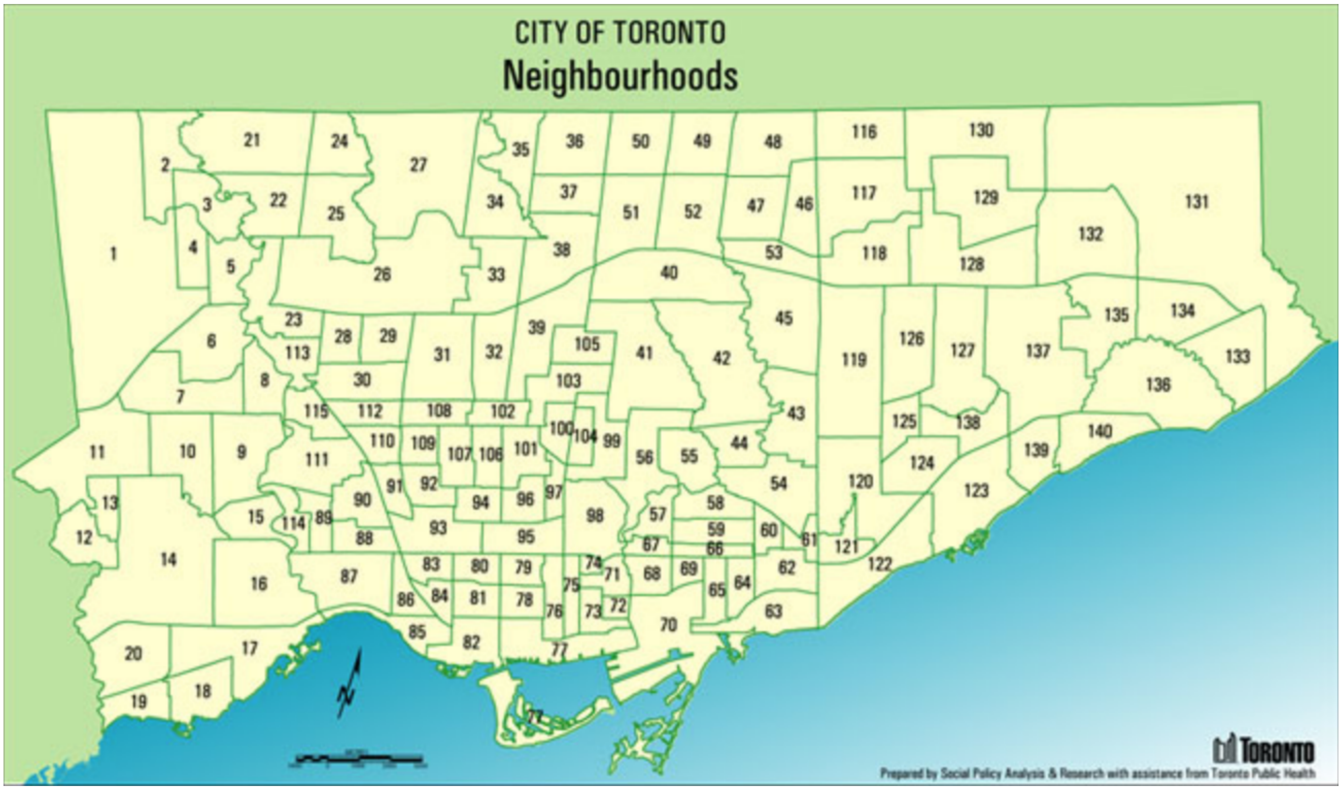


Figure 4: Official City of Toronto Neighborhoods

Refer to the **City of Toronto** for the neighborhood names matching the indeces above.

Appendix: Code

```
library(MASS); library(lmtest); library(knitr); library(kableExtra); library(nleqslv);
library(Pmisc); library(extrafont); library(VGAM); library(INLA); library(MEMSS);
library(nlme); library(ciTools); library(sf); library(tibble); library(sp); library(dplyr);
 library(lme4);  library(mgcv); library(data.table);
library(geostatsp, quietly = TRUE);library(mapmisc, quietly = TRUE);library(maptools);
library(raster);library(ggmap); library(rgdal); library(ggplot2);library(plyr)


knitr::opts_chunk$set(fig.pos = 'H');
options(tinytex.verbose = TRUE)
# Loading polygon and population data from the City of Toronto
population <- read.csv("https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/neighbourhoo

#require(sf)
shape <- read_sf(dsn = "~/Documents/Github/data_sci_geo/data/neighbourhoods_planning_areas_wgs84_SEB/", layer

neighborhoods <- shape

# Adding populaation info to neighborhood polygon
neighborhoods <- add_column(neighborhoods, '2016pop'=NA, 'x_coords' = NA, 'y_coords' = NA)

# Separating X and Y coordinates from polygon
for (hood in neighborhoods$AREA_NAME) {
  ## Adding population
  pop = as.numeric(neighborhoods[neighborhoods$AREA_NAME == hood,][["AREA_S_CD"]])
  neighborhoods[neighborhoods$AREA_NAME == hood,]$'2016pop' =
    population[population$HoodID == pop,]$Pop2016
  ## Adding x-y
  temp = unlist(subset(neighborhoods,AREA_NAME == hood)$geometry[[1]])
  ll = length(temp)
  x_coord = list(temp[1:(ll/2)])
  y_coord = list(temp[((ll/2)+1):ll])
  neighborhoods[neighborhoods$AREA_NAME == hood,]$x_coords = x_coord
  neighborhoods[neighborhoods$AREA_NAME == hood,]$y_coords = y_coord
}

st_write(neighborhoods,"~/Documents/Github/data_sci_geo/data/neighbourhoods_planning_areas_wgs84_SEB/NEIGHBOR
         , delete_layer = TRUE)

neighborhoods <- read_sf(dsn = "~/Documents/Github/data_sci_geo/data/neighbourhoods_planning_areas_wgs84_SEB/

###ALTERNATIVE VISUALIZATION
neighborhoods = rgdal::readOGR(dsn = "~/Documents/Github/data_sci_geo/data/neighbourhoods_planning_areas_wgs8
accidents <- read.csv("https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/accidents.csv

# Set up df
neighborhoods@data$id = rownames(neighborhoods@data)
neighborhoods.points = fortify(neighborhoods, region="id")
neighborhoods.df = join(neighborhoods.points, neighborhoods@data, by = "id")

# Plotting command - basic

#ggplot(neighborhoods.df) + aes(long,lat,group=group,fill=X2016pop)+ geom_polygon() +
#+   geom_path(color="black") + coord_equal()

# Adding points
```

```r
#sum_accidents <- accidents %>%
#  group_by(Neighbourhood, YEAR) %>%
#  summarize(`Total Fatalities` = sum(INJURY == "Fatal", na.rm = T),
 #           `Total Collisions` = n()) %>%
#  arrange(desc(`Total Fatalities`))

cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

#To use for fills, add
#scale_fill_manual(values=cbPalette)

# To use for line and point colors, add
#scale_colour_manual(values=cbPalette)


ggmap::register_google(key = "AIzaSyB13QyZy3PLnR5BYGtwezYWFaSq_pjrNjA")


#####
p0 <- ggmap(get_googlemap(center = c(lon = -79.384293, lat = 43.71),
                    zoom = 10, scale = 2,
                    maptype ='terrain',
                    color = 'color'), maprange=T,extent = "normal") +
    labs(x = "", y = "") +
    scale_x_continuous(limits = c(-79.63926, -79.11524), expand = c(0, 0)) +
scale_y_continuous(limits = c(43.581, 43.85546), expand = c(0, 0)) +
  theme(legend.position = "right",
        panel.background = element_blank(),
        axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        plot.margin = unit(c(0, 0, -1, -1), 'lines')) +
  xlab('') +
  ylab('')

p2 <- p0 + geom_polygon(aes(long,lat,group=group,fill=NA,color="white"),color="plum",fill=NA,data=neighborhoo
                         breaks=c("Fatal", "Non-Fatal Injury"),
                         labels=c("Fatal", "Non-Fatal"))

p1 <- p0 + geom_polygon(data=neighborhoods.df, aes(long,lat,group=group, fill=X2016pop),alpha = 0.8,color="pl

p1

p2

# Visualization of fatal vehicular incidents in the City of Toronto 2010-2016
collisiondat <- read.csv("https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/Fatal_Coll

coordinates(collisiondat) <- ~LONGITUDE+LATITUDE
#4326 - WGS84 std
proj4string(collisiondat) <- "+init=epsg:3034" #"+init=epsg:4326"
data_L93 <- spTransform(collisiondat, CRS("+proj=lcc +lat_1=44 +lat_2=49 +lat_0=46.5 +lon_0=3 +x_0=490000 +y_
#x_0/y_0 = 0.1060606


url1 <- "https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/reports/draft/STA2453-Toronto-20
download.file(url = url1,
          destfile = "toronto_incidents.png",
```

```r
         mode = 'wb')

knitr::include_graphics(path="Toronto-2016.png")

#spTransform() #Transform polygon or raster into Euclidian object - 3026 is Google std

data.frame(Accident_ID = c(5002235651, 5000995174, 5000995174, 1249781),
           Fatal = c(1, 1, 1, 0),
           Date = c("2015-12-30", "2015-06-13", "2015-06-13", "2011-08-04"),
           Neighborhood = c("Greenwood-Coxwell", "Annex", "Annex", "Bay Street Corridor"),
           Population = c(7072, 26703, 26703, 19348),
           `Max_Temp` = c(4.7, 22.3, 22.3, 26.4)) %>%
  kable() %>%
  kable_styling(bootstrap_options = c("striped"))
accidents <- read.csv("https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/accidents.csv
                      check.names = F)

accidents %>% group_by(Neighbourhood) %>%
  dplyr::summarize(`Total Fatalities` = sum(INJURY == "Fatal", na.rm = T),
            `Total Collisions` = n()) %>%
  arrange(desc(`Total Fatalities`)) %>%
  head() %>%
  kable()%>%
  kable_styling(bootstrap_options = c("striped"))


accidents %>% mutate(Pedestrian = INVTYPE == "Pedestrian",
                     Cyclist = INVTYPE == "Cyclist",
                     Other = INVTYPE != "Pedestrian" & INVTYPE != "Cyclist") %>%
  group_by(Neighbourhood) %>%
  dplyr::summarize(`Total Pedestrian Fatalities` = sum(INJURY == "Fatal" & Pedestrian == 1, na.rm = T),
            `Total Pedestrian Collisions` = sum(Pedestrian == 1, na.rm = T)) %>%
  arrange(desc(`Total Pedestrian Fatalities`)) %>%
  head() %>%
  kable()%>%
  kable_styling(bootstrap_options = c("striped"))

accidents %>% mutate(Pedestrian = INVTYPE == "Pedestrian",
                     Cyclist = INVTYPE == "Cyclist",
                     Other = INVTYPE != "Pedestrian" & INVTYPE != "Cyclist") %>%
  group_by(Neighbourhood) %>%
  dplyr::summarize(`Total Cyclist Fatalities` = sum(INJURY == "Fatal" & Cyclist == 1, na.rm = T),
            `Total Cyclist Collisions` = sum(Cyclist == 1, na.rm = T)) %>%
  arrange(desc(`Total Cyclist Fatalities`)) %>%
  head()%>%
  kable()%>%
  kable_styling(bootstrap_options = c("striped"))

accidents %>% mutate(Pedestrian = INVTYPE == "Pedestrian",
                     Cyclist = INVTYPE == "Cyclist",
                     Other = INVTYPE != "Pedestrian" & INVTYPE != "Cyclist") %>%
  group_by(Neighbourhood) %>%
  dplyr::summarize(`Total Other Fatalities` = sum(INJURY == "Fatal" & Other == 1, na.rm = T),
            `Total Other Collisions` = sum(Other == 1, na.rm =T)) %>%
  arrange(desc(`Total Other Fatalities`)) %>%
  head()%>%
  kable()%>%
  kable_styling(bootstrap_options = c("striped"))
```

```r
# Loading final monthly incident data, by neighborhood
incidentdata <- read.csv("https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/accidents.

#incidentdata$Population2 <- incidentdata$Population/1000
#incidentdata$Days_since_start2 <- incidentdata$Days_since_start/100
#incidentdata <- filter(incidentdata, ACCLASS != "Property Damage Only")

#population <- read.csv("https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/toronto_hoo

#Adding neighborhood area
#incidentdata_test <- incidentdata %>%
#  left_join(dplyr::select(population, HoodID, area_sqkm), by = c("Hood_ID" = "HoodID")) #%>% mutate(density

#write.csv(incidentdata_test, "~/Desktop/Grad_School/COURSEWORK/Spring 2019/Data Science/rough work/accidents

freqmod1 <- glmer(as.factor(ACCLASS) ~ Days_since_start2 + Tot_precip + Min_temp + (1 + Days_since_start2 |Ne
                  control=glmerControl(optimizer= "Nelder_Mead"))
accidents <- read.csv(file="https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/final/ac
accidents4 = accidents

accidents4$year = substr(as.character(accidents4$date),1,4)
accidents4$month = substr(as.character(accidents4$date),6,7)
accidents4$day = substr(as.character(accidents4$date),9,10)
accidents4$longitude = accidents4$long
accidents4$latitude = accidents4$lat
accidents4$hood_id = as.factor(accidents4$hood_num)


accidents4$date = paste(accidents4$year, accidents4$month, accidents4$day, sep = "-")

timeOrigin = ISOdate(2007,1,1,0,0,0, tz='UTC')
accidents4$daynum = as.integer(as.numeric(difftime(accidents4$date, timeOrigin, units='days')))
accidents4$weeknum = as.integer(as.numeric(difftime(accidents4$date, timeOrigin, units='weeks')))

accidents4 <- filter(accidents4, acc_class!="Property Damage Only")
accidents4$accclass <- ifelse(accidents4$acc_class=="Fatal",1,0)

accidents3 = accidents4
accidents3$visibilityb = as.character(accidents3$visibility)
accidents3$visibilityb = as.factor(ifelse(accidents3$visibilityb =="Clear", "Clear", "Not Clear"))

#factorize hood_id
accidents3$hoodid = as.factor(accidents3$hood_num)

#group road class
accidents3$roadclass = as.character(accidents3$road_class)
accidents3$roadclass = ifelse(accidents3$road_class %in% c("Major Arterial", "Major Arterial Ramp", "Minor Ar

accidents3$roadclass = as.factor(accidents3$roadclass)
accidents3$roadclass = relevel(accidents3$roadclass,ref='Local')

#traffic control class
accidents3$trafficctrl = as.character(accidents3$traffic_ctrl)
accidents3$trafficctrl = ifelse(accidents3$trafficctrl %in% c("", "No Control"), "No Control", ifelse(acciden

accidents3 =  subset(accidents3, trafficctrl != "Human Control")
accidents3$totprecipmm <- accidents3$tot_precip_mm
```

```r
accidents3$trafficctrl = as.factor(accidents3$trafficctrl)
accidents3$trafficctrl = relevel(accidents3$trafficctrl,ref='No Control')


#group invaded type - may be correlated to road class
accidents3$persontype = as.character(accidents3$person_type)
accidents3$persontype = as.factor(ifelse(accidents3$persontype %in% c("Pedestrian", "Pedestrian - Not Hit"),

accidents3$weekiid = accidents3$weeknum

fitS <- inla(accclass ~ visibilityb + roadclass + trafficctrl + persontype + totprecipmm +
               f(weeknum, model='rw1' , hyper = list(prec=list(prior='pc.prec', param=c(0.2, 0.05)))
) + f(weekiid, model='iid' , hyper = list(prec=list(prior='pc.prec', param=c(0.2, 0.05)))
)
  + f(hoodid, model='iid', hyper = list(prec=list(prior='pc.prec', param=c(0.25, 0.01)))
), data=accidents3, family='binomial',
control.mode = list(theta = c(2.2, 7.2, 5), restart=TRUE)
)

fitS$priorPost = Pmisc::priorPost(fitS)

resTable1 <- exp(fitS$summary.fixed[, c("mean", "0.025quant",
"0.975quant")]);
resTable2 <- Pmisc::priorPostSd(fitS)$summary[,
c("mean", "0.025quant", "0.975quant")]
restable <- rbind(resTable1,resTable2)

knitr::kable(restable, digits=3, escape=F, format="latex", booktab=T,linesep = "", caption="Posterior mean an
  kable_styling(latex_options = "hold_position")
# plotting
matplot(
as.numeric(fitS$summary.random$weeknum$ID),
exp(fitS$summary.random$weeknum[,
c('0.025quant','0.975quant', '0.5quant')]), xlab='2007 - 2017', lty=1, col=c('grey','grey','black'), type='l'
axis(1, at=seq(0,600,52), labels=c("2007","2008","2009","2010","2011","2012","2013","2014","2015","2016","201
par(mar = c(4,4,4,2) + 0.1);
#par(mgp=c(2,1,0));

for (Dparam in fitS$priorPost$parameters[2:4]) {
  do.call(matplot, fitS$priorPost[[Dparam]]$matplot)
}
fitS$priorPost$legend$x = "topleft"
#do.call(legend, fitS$priorPost$legend)

#GAM

# library(Hmisc)

accidents <- read.csv(file="https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/final/ac
accidents4 = accidents

accidents4$year = substr(as.character(accidents4$date),1,4)
accidents4$month = substr(as.character(accidents4$date),6,7)
accidents4$day = substr(as.character(accidents4$date),9,10)
accidents4$longitude = accidents4$long
accidents4$latitude = accidents4$lat
accidents4$hood_id = as.factor(accidents4$hood_num)
```

```
accidents_time <- accidents4 %>% group_by(hood_id, year,month,day) %>% dplyr::summarize(value_perd=n())
accidents_time_weather <- accidents4 %>% group_by(hood_id, year,month,day) %>% dplyr::summarize(avg_snow = me


accidents_ts <-  merge(accidents_time,accidents_time_weather, by.x=c("hood_id","year", "month", "day"), all.x
accidents_ts$date = paste(accidents_ts$year, accidents_ts$month, accidents_ts$day, sep = "-")

accidents_ts$month_f = as.factor(accidents_ts$month)

timeOrigin = ISOdate(2007,1,1,0,0,0, tz='UTC')
accidents_ts$day_num = as.numeric(difftime(accidents_ts$date, timeOrigin, units='days'))


#offset pop
  # pop = accidents4 %>%
  #   select(hood_id, year, Population) %>%
  #   group_by(hood_id, year) %>%
  #   arrange(hood_id, year) %>%
  #   slice(n())
  #
  # pop2 = pop %>%
  #   select(year,Population)%>%
  #   group_by(year) %>%
  #   summarise(Population_sum=sum(Population))
  #
  # accidents_ts <-  merge(accidents_ts, pop2, by=c("year"), all.x=TRUE)
  #estimate population
  A = (2731571-2503281)/10; B = 2503281 - 2006*A
  year = seq(2007, 2017, by=1)

  est_pop = as.data.frame(cbind(year, year*A + B))
  names(est_pop)[2] = "population_est"

  accidents_ts <-  merge(accidents_ts, est_pop, by=c("year"), all.x=TRUE)
  accidents_ts$log_pop = log(accidents_ts$population_est)

# accidents_ts$value = cumsum(accidents_ts$value_perd)
accidents_ts2 = c()
for (i in 1:length(levels(accidents_ts$hood_id)))
{ temp = accidents_ts
  temp$hood_num = as.numeric(accidents_ts$hood_id)

  current = subset(temp,temp$hood_num == i)
  current$value = cumsum(current$value_perd)

  accidents_ts2 = rbind(accidents_ts2, current) }




accident_ts_gam = gam(value ~ month_f + offset(log_pop) + s(day_num) + s(hood_id,bs="re"), data=accidents_ts2
# accident_ts_gam = gam(value ~ month_f + s(day_num,bs="re", by = hood_id), data=accidents_ts2, family='poiss

# rownames(accident_ts_gam) = c("Intercept", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "

# summary(accident_ts_gam)
knitr::kable(summary(accident_ts_gam)$p.table[,1:2],digits=3)%>%
kable_styling(bootstrap_options = c("striped"))
```

```
#plot rr by month
accident_ts_gam_pred_rr = exp(summary(accident_ts_gam)$p.table[2:12,1:2] %*% Pmisc::ciMat())

matplot( accident_ts_gam_pred_rr, log = "y", xaxt = "n", xlab = "Months", type = "l", lty = c(1, 2, 2), col =
axis(1, at = 1:11, labels = c("Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))
# check 1 hood
plot_predict_hoodid = function(hood_id){
  i = hood_id
  newX = data.frame(date = seq(from = timeOrigin, by = "months", length.out = 12 * 18))
  newX$day_num = as.numeric(difftime(newX$date, timeOrigin, units = "days"))
  newX$month_f = as.factor(substr(as.character(newX$date),6,7))
  newX$year = substr(as.character(newX$date),1,4)
  newX$hood_id = i
  newX_all = newX

  year = seq(min(newX$year), max(newX$year), by=1)
  est_pop = as.data.frame(cbind(year, year*A + B))
  names(est_pop)[2] = "population_est"

  newX_all <-  merge(newX_all, est_pop, by=c("year"), all.x=TRUE)
  newX_all$log_pop = log(newX_all$population_est)

  newX_all$hood_id = as.factor(newX_all$hood_id)

  accident_ts_gam_pred = predict(accident_ts_gam, newX_all, se.fit = TRUE)
  accident_ts_gam_pred = cbind(newX, accident_ts_gam_pred)

  accident_ts_gam_pred$lower = accident_ts_gam_pred$fit - 2 * accident_ts_gam_pred$se.fit
  accident_ts_gam_pred$upper = accident_ts_gam_pred$fit + 2 * accident_ts_gam_pred$se.fit
  for (D in c("fit", "lower", "upper")) {
  accident_ts_gam_pred[[paste(D, "exp", sep = "")]] = exp(accident_ts_gam_pred[[D]])
  ####################plot rr################
 # accident_ts_gam_pred_rr = as.matrix(as.data.frame(predict.gam(accident_ts_gam, newX_all, type = "terms", t
 # accident_ts_gam_pred_rr = exp(accident_ts_gam_pred_rr[,c(1,4)] %*% Pmisc::ciMat())
 #
 # matplot(newX_all$year, accident_ts_gam_pred_rr, log = "y", xaxt = "n", xlab = "date", type = "l", lty = c(
 # axis(1, at = difftime(newX_all$year, timeOrigin, units = "days"), labels = format(dSeq, "%Y"))

}


pred_hood = accident_ts_gam_pred
plot(pred_hood$date, pred_hood[, "fitexp"], type = "n", xlab = "date", ylab = "number of accidents")
matlines(pred_hood$date, pred_hood[, c("lowerexp", "upperexp", "fitexp")], lty = 1, col = c("grey","grey", "b
}

plot_predict_hoodid(5)
plot_predict_hoodid(122)

# neighborhoods = rgdal::readOGR(dsn = "C:/Users/ThinkPad/Desktop/Eddy/DS", layer = "NEIGHBORHOODS_WGS84")
# neighborhoods = rgdal::readOGR("C:/Users/ThinkPad/Desktop/Eddy/DS/NEIGHBORHOODS_WGS84.shp",layer="NEIGHBORH

# zoning = rgdal::readOGR("C:/Users/EDDY/Documents/UNIVERSITY/STA2453/Proj2/zoning/ZONING_ZONE_CATAGORIES_WGS
# traffic_signals <- read.csv(file="C:/Users/EDDY/Documents/UNIVERSITY/STA2453/Proj2/traffic_signals.csv", he

accidents <- read.csv(file="https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/data/final/ac
```

```r
accidents$YEAR = substr(as.character(accidents$date),1,4)
accidents$longitude = accidents$long
accidents$latitude = accidents$lat


#add new features
#day/night; logdensity
# accidents$day_night = as.factor(ifelse(accidents$Hour >21 | accidents$Hour <6, "Night", "Day")) #day time i
# accidents$logdensity = log(accidents$density) #day time is 1

#subset to 2017 for now
accidents = subset(accidents, accidents$YEAR==2017)
#####

accidents_lonlat = as.matrix(cbind(accidents$longitude, accidents$latitude),nrow=nrow(accidents))

accidents_spatial = SpatialPointsDataFrame(coords= accidents_lonlat, data = accidents, coords.nrs = numeric(0

# spRbind(accidents_spatial, zoning)

accidents2 = spTransform(accidents_spatial, mapmisc::omerc(accidents_spatial, angle=-17))
theMap = mapmisc::openmap(accidents2, maxTiles=4, fact=3)
mapmisc::map.new(accidents2)
plot(theMap, add=TRUE, maxpixels=10^7)
plot(accidents2, col=mapmisc::col2html("black", 0.4), cex=0.6, add=TRUE)
#testing

canada <- getData(name="GADM", country="CAN", level=2)
trt_border = subset(canada, NAME_2=="Toronto")
accidents_spatial_border = spTransform(trt_border, projection(accidents2))
# plot(accidents_spatial)

accidents_fit = lgcp(formula = ~ 1, data = accidents2, grid = 55, shape = 1, buffer = 2000,
                     prior = list(range = 6000, sd =0.5), border=accidents_spatial_border,
                                 control.inla = list(strategy='gaussian'), verbose=FALSE)

 mapmisc::map.new(accidents_spatial_border)
 plot(accidents_fit$raster[['predict.exp']]*10^6, add=TRUE)
 plot(accidents_spatial_border, add=TRUE)
var_def <- read.csv("https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/reports/draft/variab

knitr::kable(var_def, format="latex", booktab=T, linesep = "")%>%
#escape=F,
kable_styling(bootstrap_options = c("striped"))
## Visualizing neighborhoods of Toronto for reference
url7 <- "https://raw.githubusercontent.com/sergiosonline/data_sci_geo/master/reports/draft/toronto-hoods.png"
download.file(url = url7,
          destfile = "toronto-hoods.png",
          mode = 'wb')

knitr::include_graphics(path="toronto-hoods.png")
```