Estimation of Fatal Crashes

Lucas Anderton, Hannah Brown, Lucas Gorak, Edward Mikkelson

1/11/2020

Question

1. What is the relationship between 311 requests and frequency of crashes?
2. How does the average latency between requested date and resolution date affect frequency of crashes?
3. Is there a geographic relationship between crashes and 311 requests?
4. What factors best estimate fatalities?

In our analysis, we were interested in understanding the relationship, if any, between various factors, like population increase and road conditions, and the crashes that occurred in the District of Columbia. Previous research has indicated that D.C. has the third worst traffic congestion in the United States. However, their ranking has improved since 2011, which D.C. had the worst congestion of all 50 states. In that same period of time, the number of crashes in D.C. has increased significantly. Intuitively, we credited the increase to more traffic from the constantly growing population in the Washington metro area. D.C.'s public datasets on traffic are convoluted and difficult to manipulate for analysis. It is recorded in 15,000+ rows from various traffic-volume meters throughout the city on an annual basis. However, every year, the number of meters and other variables in the data change. Alternately, we chose to compare the number of crashes to the 311 traffic service requests. 311 traffic service requests offer information regarding the number of commuters out on the roads and the state of the city's vehicle infrastructure. In our research, we wanted to understand what the relationship between these 311 traffic requests and the frequency of crashes in D.C. Additionally, these data include timestamp information, like the date they were submitted, the date they were due (most likely based on a city algorithm for how long certain requests should take), and the date they were resolved. We wanted to see if a relationship existed between rising latency in resolution times and the frequency of crashes. The reasoning behind our hypothesis was that if the city was taking longer to respond to poor road and navigation conditions, more incidents may occur. We also wanted to investigate if there was a geographical relationship between crashes and 311 requests. And lastly, we were interested in analyzing what factors best estimate risk of faility in a crash.

Social Context

- Public health and safety
- Government responsiveness
- Manifestations of demographic disparities

In our preliminary research, we found three comprehensive datasets from D.C.'s open data site, each offering salient variables pertaining to vehicle crashes in D.C. over time. The first set we explored was mostly categorical and qualitative, which left much to be desired in terms of quantitative analysis. The remaining sets include a combined 63 variables, both qualitative and quantitative in nature.

Next, we happened upon the city's 311 service request data portal, on the same OpenData library. The portal offered custom data downloaded, allowing the user to select a date range or data for one specific type of request. The full data set included more than 1.5 million rows with approximately ten types of requests. We were able to

import our data directly from D.C.'s OpenData portal.

Data Collection Procedure

- Started with OpenDataDC's `crashes` dataset.
- Explored related `crashdetails` in a relational table.
- Using that for additional data on a subset of the DC crashes.
- Summarized 311 Requests `threeoneonerequestts` and `crashes` by count per day.
- Merged the `crashes` with `crashdetails` by `CRIMEID` to include `WARD`

Interesting Features and Data Processing

- `crashes` has 223990 observations and 60 columns
- `crashdetails` has 443867 observations and 15 variables
- `threeoneonerequests` 1.5 million observations >>> compressed to 65,000 rows
- Ages beginning at -7990 and ranged up to 237
- License plate `None` produced a high number of crashes in the data set
- The full dataset reaches back until 1975

In order to compare the number of service requests to the number of accidents on a given day, we had to combine the `crashes` and `details` datasets. This proved challenging, as the datasets are remarkably large, and have dates that do not align. We remedied the former issue by selecting 311 requests that pertained to traffic related requests only. This included "roadway signs," "streetlight repair," and "potholes."

Methods

- Recoding data from categorical to binary for use in regression models
- Ran a series of exploratory models with `ggplot`
- Estimate Bayesian logistic regression

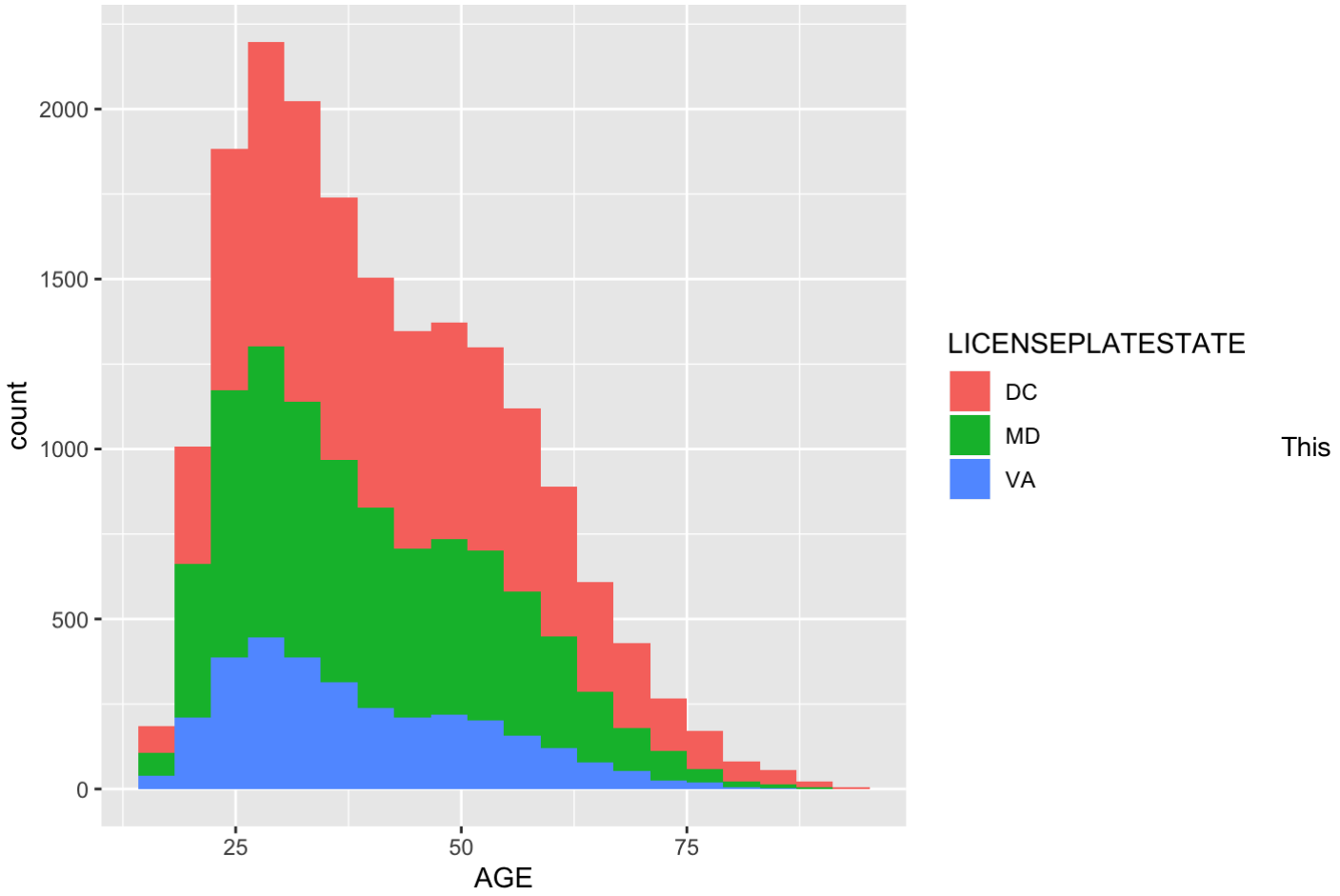
For estimation of hte varoius regression models, the first order of operation was to recode the data from character types to binary so variables such as `SPEEDING` and `IMPAIRED` functioned as factors. Other data wrangling such as limiting the `AGE` variable and restricting the `FROMDATE` begging at 2012 was completed to complete better analysis. `ggplot` was used primarily to visualize data for exploratory data analysis and mapping.

We began by creating a regression model for the daily requests and daily crashes. Our model shows a clear correlation between the number of service requests on a given day and the number of crashes. We then plotted the service request number and crash number for each day with the regression line. We also mapped both service requests and crashes on a geographical map using `ggmap` and a Google API. We then used a K-means clustering algorithm on geospatial data to find groupings.

Analysis

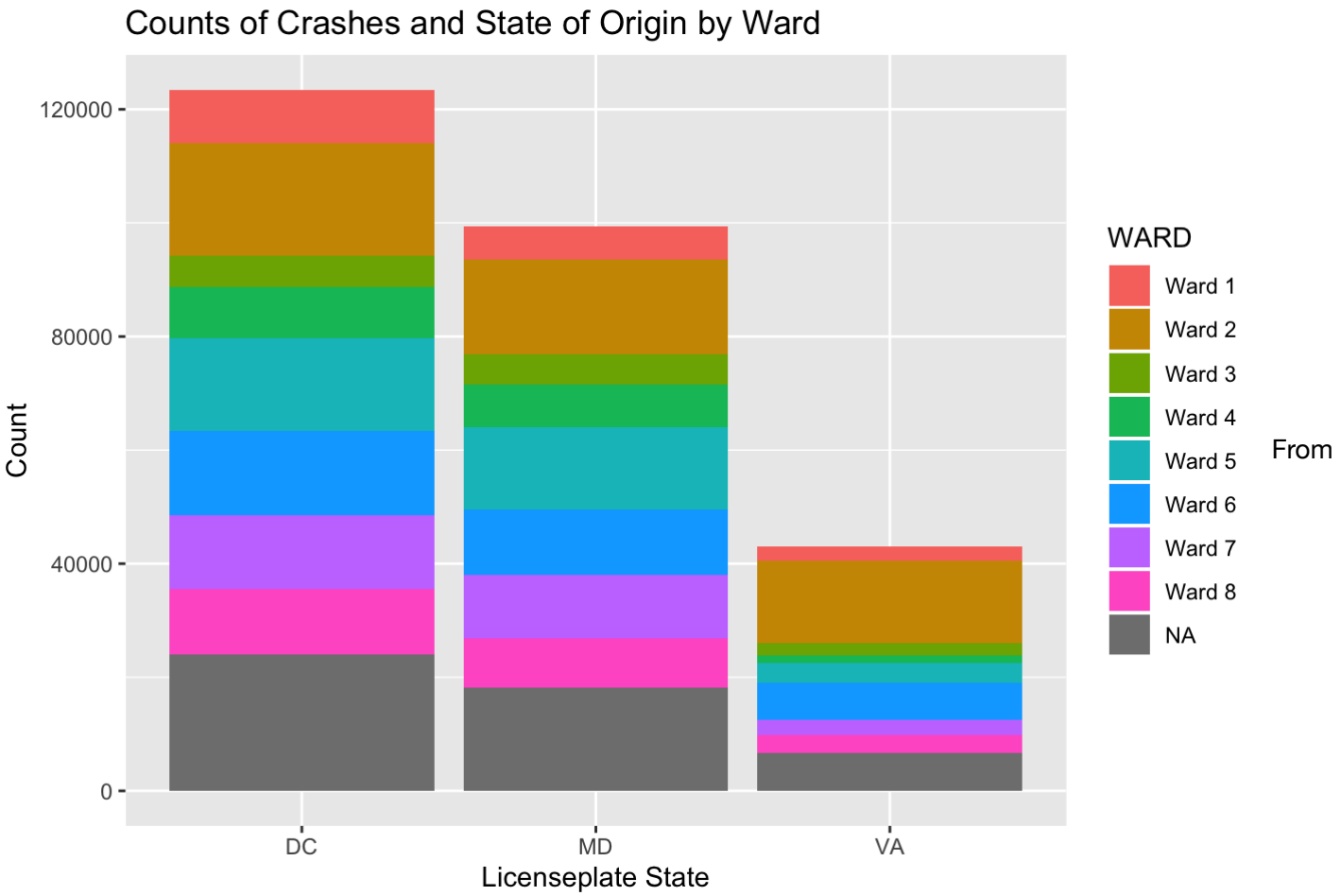
This summary provides additional information about the distribution of each of the variables of interest, primarily that the data is heavily skewed to the right, with very few instnaces of ticketing, speeding, impairment, major and minory injuries, and fatalities recorded during vehicle crashes observed in DC.

```
#plotting relevant ages
 ggplot(data = Details2, mapping = aes(x = AGE, fill = LICENSEPLATESTATE)) +
  geom_histogram( bins = 20)
```



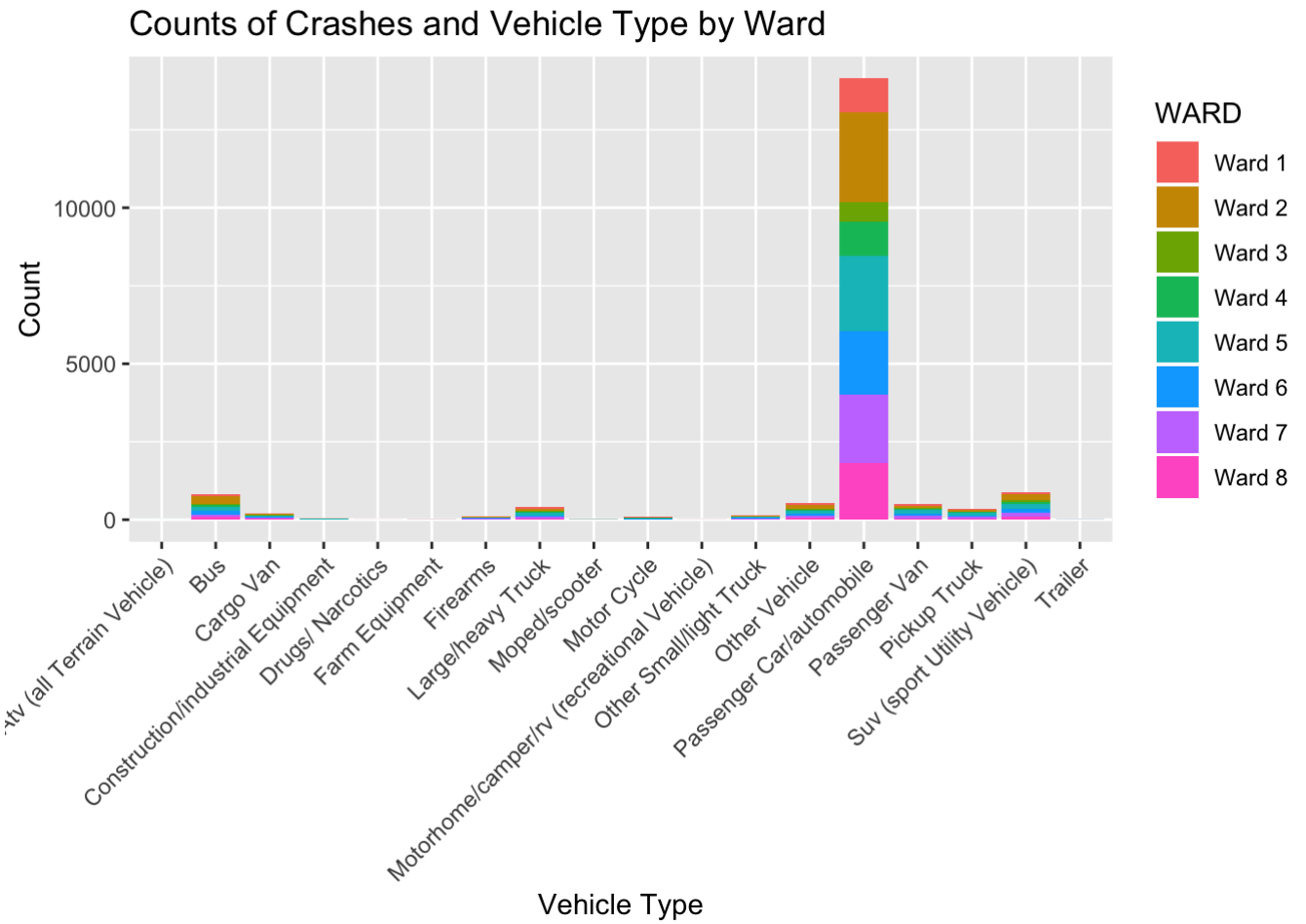
This graph provides EDA for the relationship between age, where the vehicle involved in the crash is from by age. Notably the distribution is skewed to the right with the center approximately at the 25-30 age range.

```
Details2 %>%
  ggplot(data = Details1, mapping = aes(x = LICENSEPLATESTATE, fill = WARD)) +
  geom_bar() +
  labs(title = "Counts of Crashes and State of Origin by Ward", x = "Licenseplate State"
, y = "Count")
```



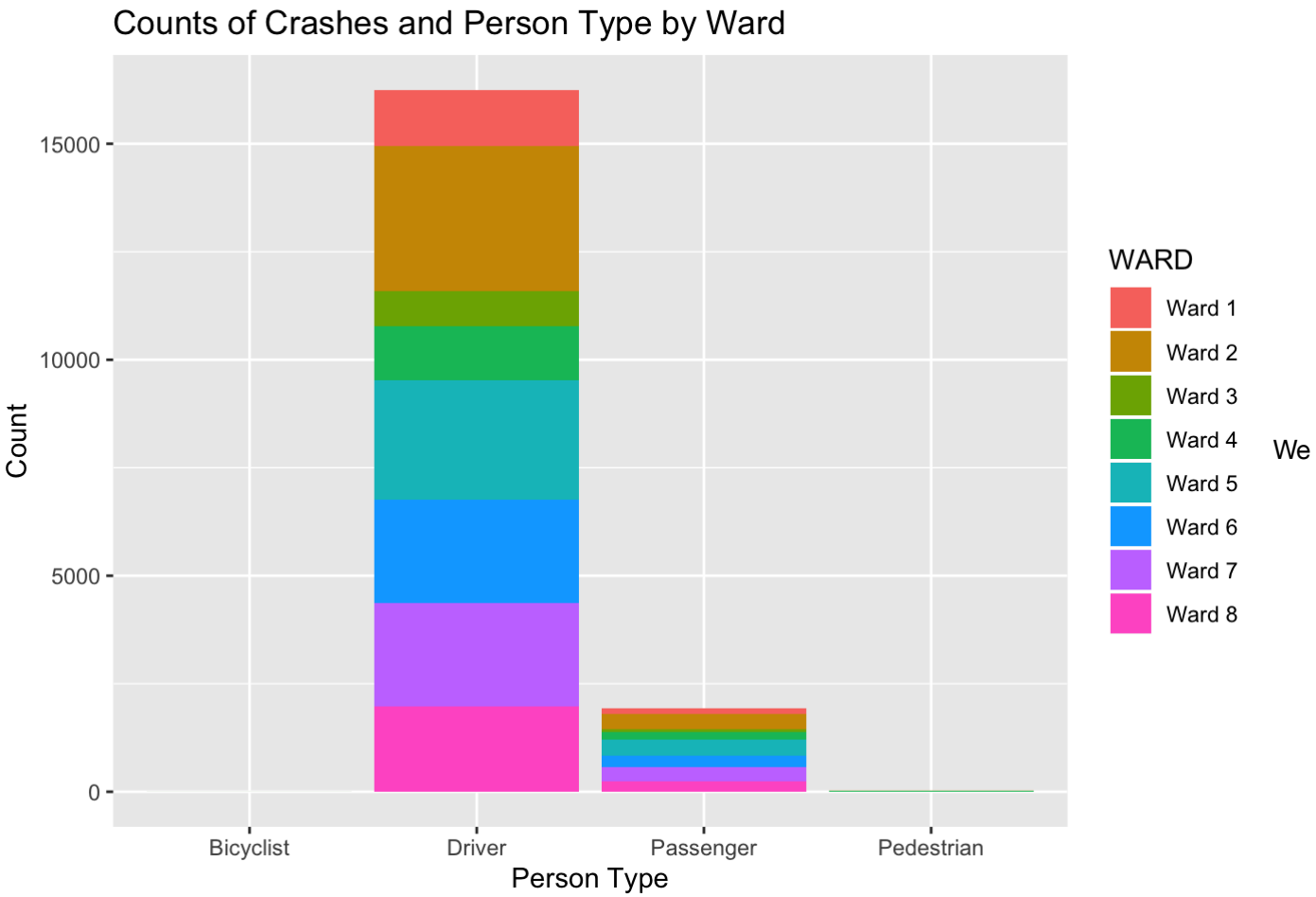
this plot it appears that the most crashes across the 3 most prevalent car state license plates is Ward 2. However looking at the frequency of this data is actually contigent on the time of day and day of the week that the crash occured considering that a number of people in the city are commuters who live outside the city so population density is not actually based on permanent address, but rather is quite complicated to measure because it includes commuters.

```
Details2 %>%
  ggplot(data = Details2, mapping = aes(INVEHICLETYPE, fill = WARD)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Counts of Crashes and Vehicle Type by Ward",  x = "Vehicle Type", y = "C
ount")
```



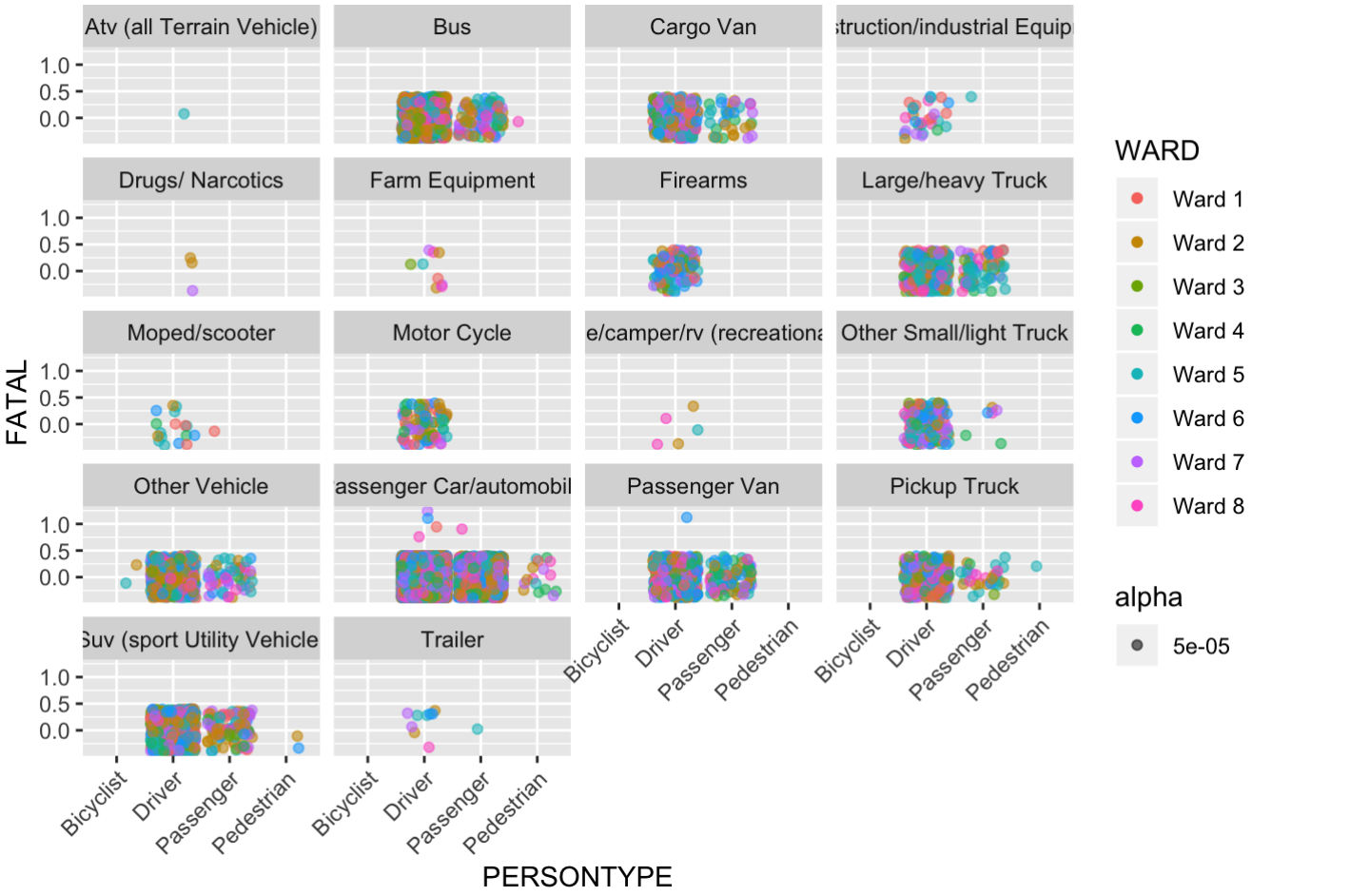
Passenger car is clearly the most likely vehicle to experience a crash in, followed by bus and SUV. Interesting features: farm equipment, firearms, and motorhome.

```
Details2 %>%
  ggplot(data = Details2, mapping = aes(PERSONTYPE, fill = WARD)) +
  geom_bar() +
  labs(title = "Counts of Crashes and Person Type by Ward",  x = "Person Type", y = "Cou
nt")
```

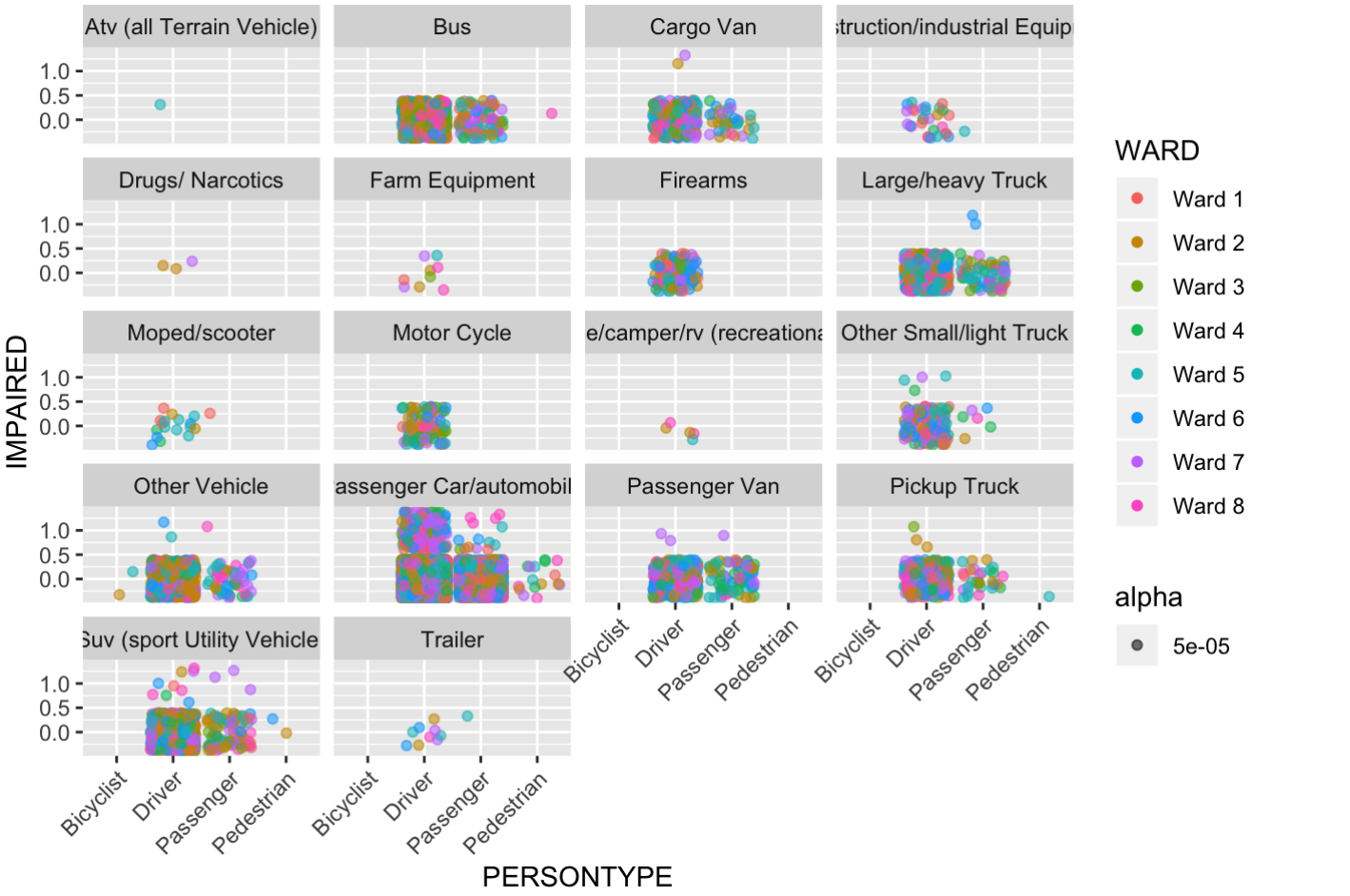


see the same pattern of prevalence of crashes by ward in person type as well, again knowing that population density is difficult to measure and that Ward 2 contains the downtown region which would expectedly have the most activity and therefore the most instances of crash.

```
Details2 %>%
  ggplot(data = Details2, mapping = aes(PERSONTYPE, FATAL, color = WARD, alpha = 0.00005
)) +
  geom_jitter() +
  facet_wrap(~INVEHICLETYPE, ncol = 4 ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

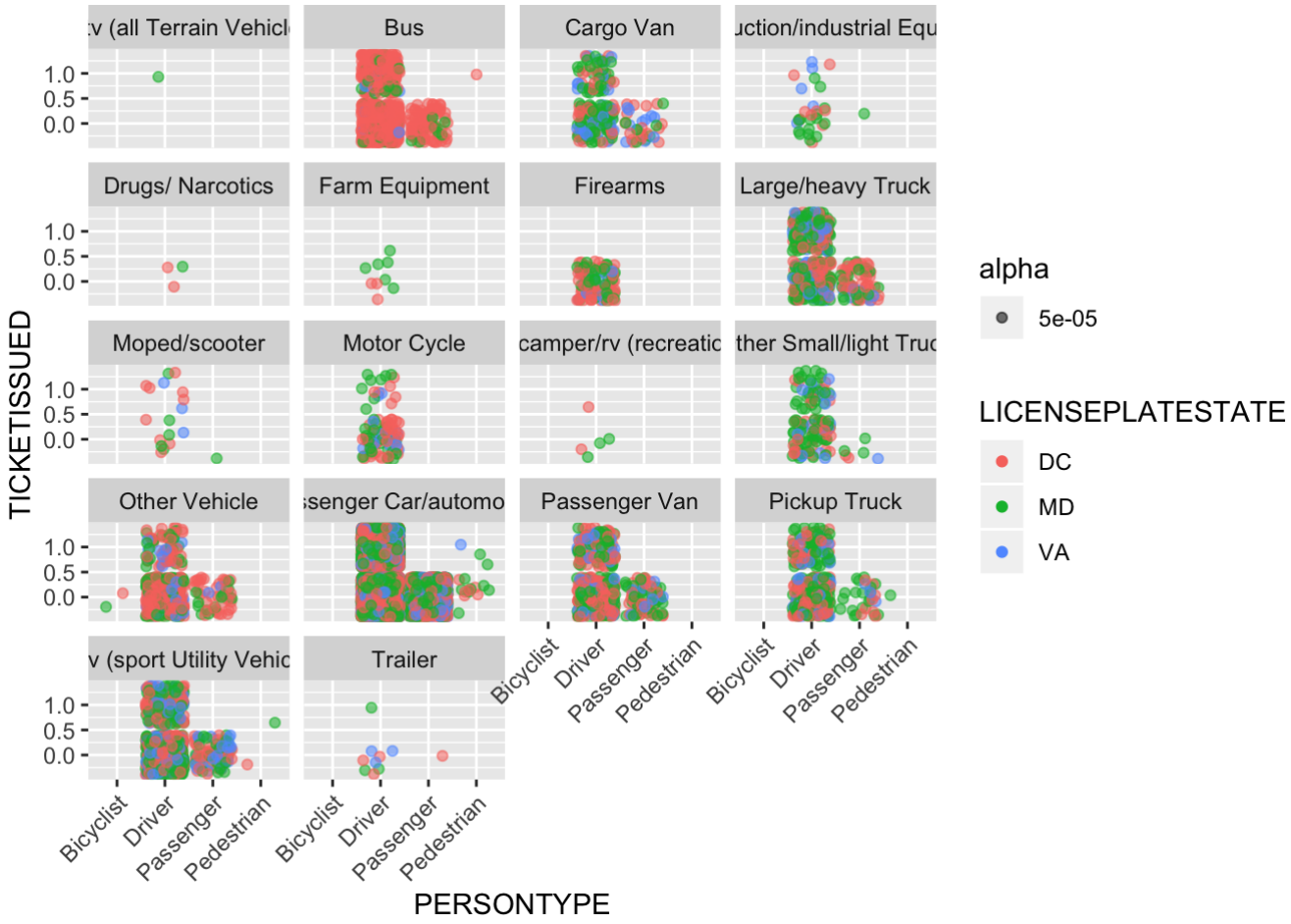



```
Details2 %>%
  ggplot(data = Details2, mapping = aes(PERSONTYPE, IMPAIRED, color = WARD, alpha = 0.00
005)) +
  geom_jitter() +
  facet_wrap(~INVEHICLETYPE, ncol = 4 ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

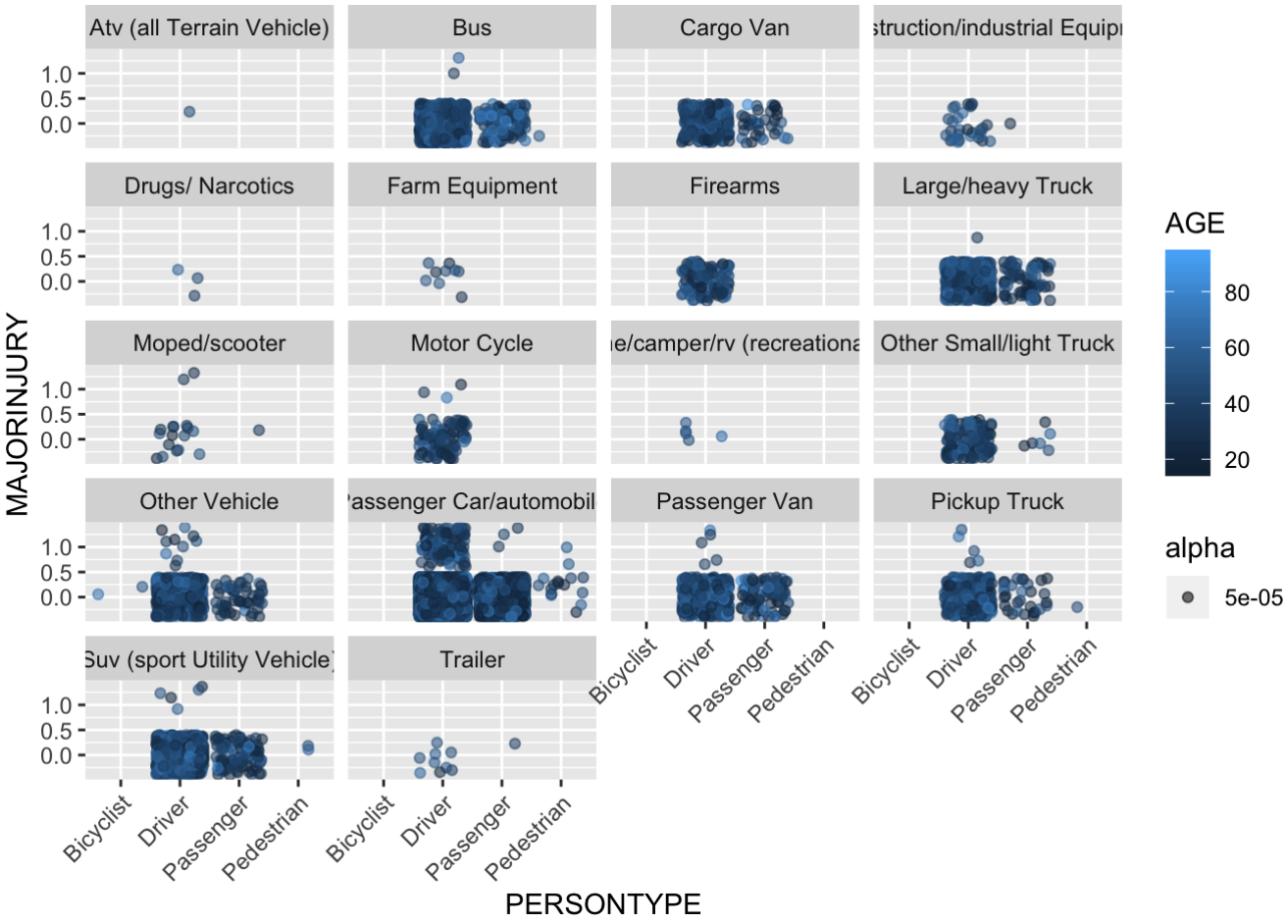



```
Details2 %>%
  ggplot(data = Details2, mapping = aes(PERSONTYPE, SPEEDING, color = LICENSEPLATESTATE,
alpha = 0.00005)) +
  geom_jitter() +
  facet_wrap(~INVEHICLETYPE, ncol = 4 ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

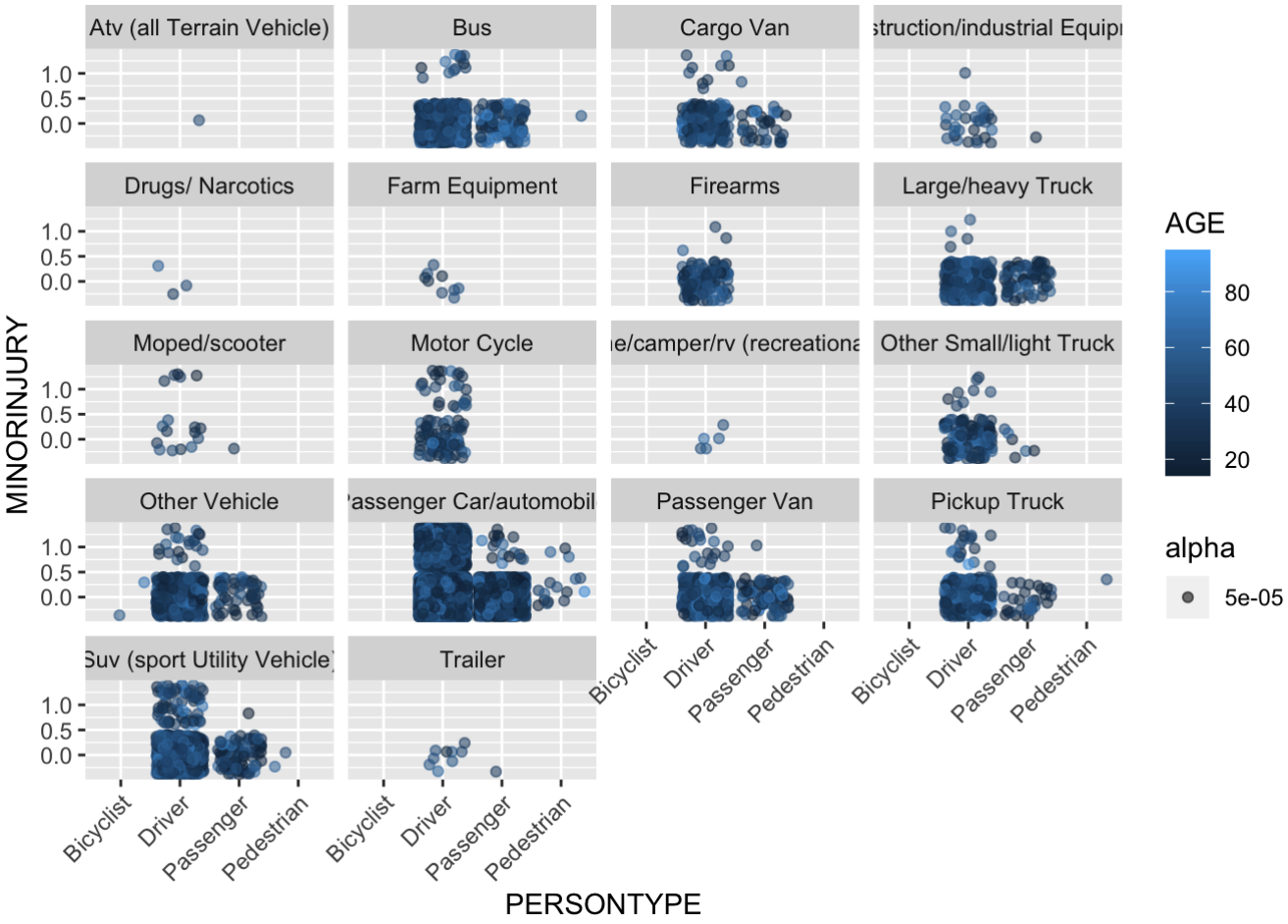

```
Details2 %>%
  ggplot(data = Details2, mapping = aes(PERSONTYPE, TICKETISSUED, color = LICENSEPLATEST
ATE, alpha = 0.00005)) +
  geom_jitter() +
  facet_wrap(~INVEHICLETYPE, ncol = 4 ) +
  theme(axis.text.x = element_text(angle = 45, hjust= 1))
```




```
Details2 %>%
  ggplot(data = Details2, mapping = aes(PERSONTYPE, MAJORINJURY, color = AGE, alpha = 0.
00005)) +
  geom_jitter() +
  facet_wrap(~INVEHICLETYPE, ncol = 4 ) +
  theme(axis.text.x = element_text(angle = 45, hjust= 1))
```




```
Details2 %>%
  ggplot(data = Details2, mapping = aes(PERSONTYPE, MINORINJURY, color = AGE, alpha = 0.
00005)) +
  geom_jitter() +
  facet_wrap(~INVEHICLETYPE, ncol = 4 ) +
  theme(axis.text.x = element_text(angle = 45, hjust= 1))
```

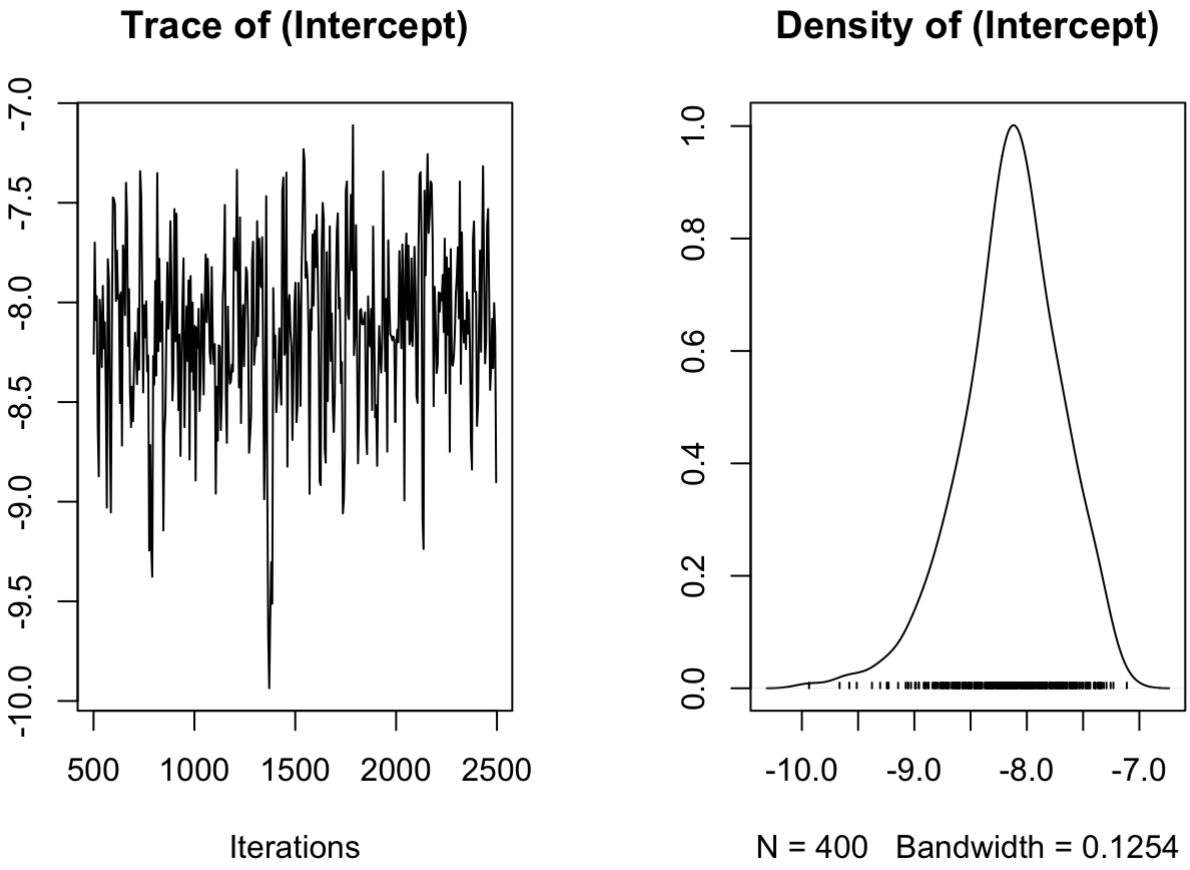


These faceted graphs provide more information abou the interactions between the people involved in the accident, the outcome (minor injury, major injury, ticketed, speeding, impaired, or fatal), in context of age, origin of car, or ward.

```
mc_post <- MCMClogit(FATAL ~ 1, data = Details2, , burnin = 500, mcmc = 2000, thin = 5)
summary(mc_post)
```

```
##
## Iterations = 501:2496
## Thinning interval = 5
## Number of chains = 1
## Sample size per chain = 400
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean             SD       Naive SE Time-series SE
##       -8.14222        0.44187         0.02209        0.03152
##
## 2. Quantiles for each variable:
##
##   2.5%    25%    50%    75%  97.5%
## -9.079 -8.380 -8.119 -7.854 -7.351
```

```
plot(mc_post)
```



Modeling fuller Bayesian model

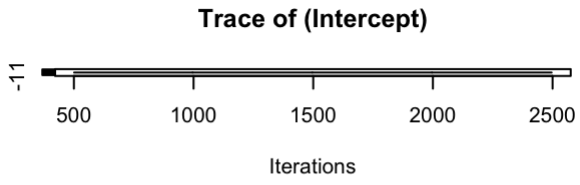
```
mc_post_full <- MCMClogit(FATAL ~ MAJORINJURY + MINORINJURY + TICKETISSUED + IMPAIRED +
SPEEDING + AGE, data = Details2, burnin = 500, mcmc = 2000, thin = 5)
```

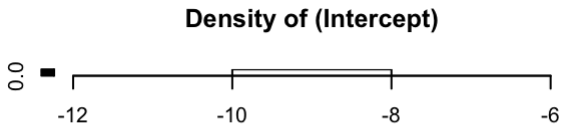
```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

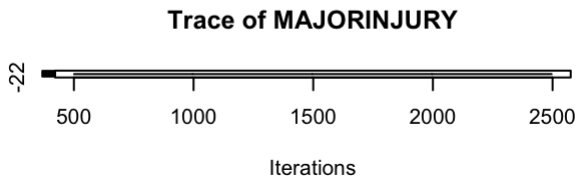
```
summary(mc_post_full)
```

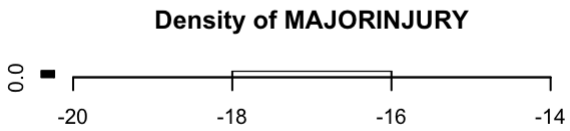
```
##
## Iterations = 501:2496
## Thinning interval = 5
## Number of chains = 1
## Sample size per chain = 400
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                   Mean SD Naive SE Time-series SE
## (Intercept)   -8.29067  0        0              0
## MAJORINJURY  -16.74073  0        0              0
## MINORINJURY  -16.46553  0        0              0
## TICKETISSUED -16.75844  0        0              0
## IMPAIRED     -15.73648  0        0              0
## SPEEDING     -16.19857  0        0              0
## AGE            0.01511  0        0              0
##
## 2. Quantiles for each variable:
##
##                   2.5%       25%       50%       75%     97.5%
## (Intercept)   -8.29067  -8.29067  -8.29067  -8.29067  -8.29067
## MAJORINJURY  -16.74073 -16.74073 -16.74073 -16.74073 -16.74073
## MINORINJURY  -16.46553 -16.46553 -16.46553 -16.46553 -16.46553
## TICKETISSUED -16.75844 -16.75844 -16.75844 -16.75844 -16.75844
## IMPAIRED     -15.73648 -15.73648 -15.73648 -15.73648 -15.73648
## SPEEDING     -16.19857 -16.19857 -16.19857 -16.19857 -16.19857
## AGE            0.01511   0.01511   0.01511   0.01511   0.01511
```

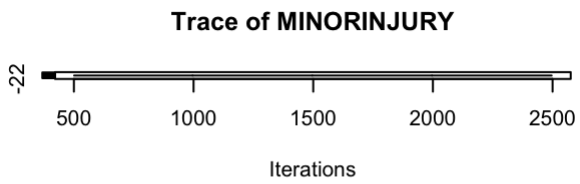
```
plot(mc_post_full)
```

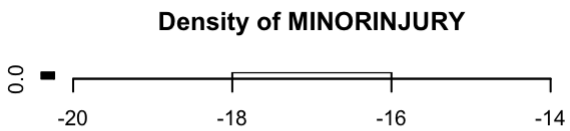

Trace of (Intercept)
-11
500
1000
1500
2000
2500
Iterations

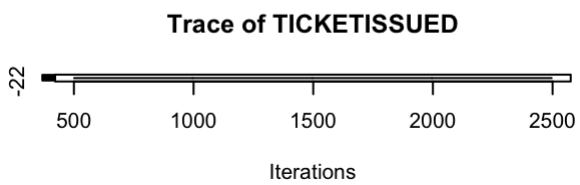

Density of (Intercept)
0.0
-12
-10
-8
-6

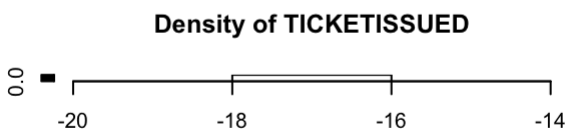

Trace of MAJORINJURY
-22
500
1000
1500
2000
2500
Iterations

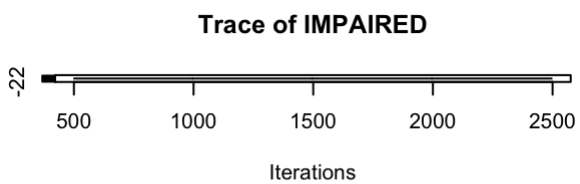

Density of MAJORINJURY
0.0
-20
-18
-16
-14

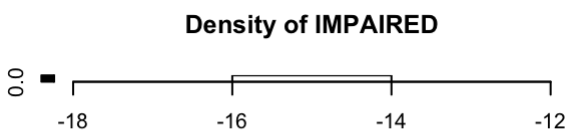

Trace of MINORINJURY
-22
500
1000
1500
2000
2500
Iterations

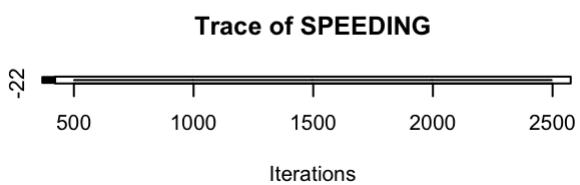

Density of MINORINJURY
0.0
-20
-18
-16
-14

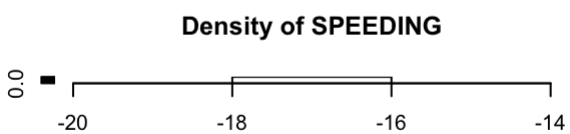

Trace of TICKETISSUED
-22
500
1000
1500
2000
2500
Iterations


Density of TICKETISSUED
0.0
-20
-18
-16
-14

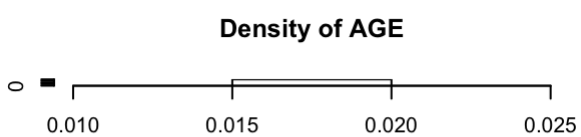

Trace of IMPAIRED
-22
500
1000
1500
2000
2500
Iterations


Density of IMPAIRED
0.0
-18
-16
-14
-12


Trace of SPEEDING
-22
500
1000
1500
2000
2500
Iterations


Density of SPEEDING
0.0
-20
-18
-16
-14


Trace of AGE
0.010
500
1000
1500
2000
2500
Iterations


Density of AGE
0
0.010
0.015
0.020
0.025

These MCMClogit show the posterior coefficients do not appear to converge to one specific value, so it is unlikely that the plots created in the previous slides represent what were to actually occur in the data due to few iterations.

Estimate Machine Learning Models with LS and LASSO

```
#Est LS model
lm_out <- lm(FATAL ~ MAJORINJURY + MINORINJURY + TICKETISSUED + IMPAIRED + SPEEDING + AG
E,
             data = Details1)
coefs_lm <- coef(lm_out)
summary(lm_out)
```

```
##
## Call:
## lm(formula = FATAL ~ MAJORINJURY + MINORINJURY + TICKETISSUED +
##     IMPAIRED + SPEEDING + AGE, data = Details1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.01382 -0.00065 -0.00054 -0.00045  0.99968
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.919e-04  1.486e-04   1.964  0.04952 *
## MAJORINJURY  -9.524e-04  3.221e-04  -2.957  0.00311 **
## MINORINJURY  -7.683e-04  1.535e-04  -5.006 5.57e-07 ***
## TICKETISSUED -1.503e-04  1.121e-04  -1.340  0.18014
## IMPAIRED     -5.658e-05  4.682e-04  -0.121  0.90381
## SPEEDING      1.284e-02  5.020e-04  25.586  < 2e-16 ***
## AGE           7.611e-06  3.263e-06   2.333  0.01965 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02419 on 265686 degrees of freedom
## Multiple R-squared:  0.002561,   Adjusted R-squared:  0.002538
## F-statistic: 113.7 on 6 and 265686 DF,  p-value: < 2.2e-16
```

```
#Estimate with LASSO
lasso_out <- glmnet(X, y, alpha = 1)
```

##Threeoneone Data Import

```
threeoneone_2 = read_csv("https://datagate.dc.gov/search/open/311requests?daterange=8yea
rs&details=true&format=csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   XCOORD = col_double(),
##   LONGITUDE = col_double(),
##   RESOLUTIONDATE = col_datetime(format = ""),
##   INSPECTIONDATE = col_datetime(format = ""),
##   SERVICEDUEDATE = col_datetime(format = ""),
##   YEAR = col_double(),
##   SERVICECALLCOUNT = col_double(),
##   MARADDRESSREPOSITORYID = col_double(),
##   ZIPCODE = col_double(),
##   YCOORD = col_double(),
##   ADDDATE = col_datetime(format = ""),
##   SERVICEORDERDATE = col_datetime(format = ""),
##   LATITUDE = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
threeoneone_2 <- filter(threeoneone_2, SERVICECODEDESCRIPTION == c("roadway signs","stre
etlight repair investigation","pothole"))

head(threeoneone_2)
```

```
## # A tibble: 6 x 31
##   XCOORD SERVICECODEDESC… DETAILS LONGITUDE STATE PRIORITY
##    <dbl> <chr>            <chr>       <dbl> <chr> <chr>
## 1 3.97e5 streetlight rep… <NA>        -77.0 DC    URGENT
## 2 3.98e5 streetlight rep… <NA>        -77.0 DC    URGENT
## 3 3.95e5 streetlight rep… <NA>        -77.1 DC    URGENT
## 4 3.99e5 streetlight rep… <NA>        -77.0 DC    URGENT
## 5 3.97e5 streetlight rep… <NA>        -77.0 DC    URGENT
## 6 3.97e5 streetlight rep… <NA>        -77.0 DC    URGENT
## # … with 25 more variables: RESOLUTIONDATE <dttm>, INSPECTIONDATE <dttm>,
## #   SERVICEDUEDATE <dttm>, YEAR <dbl>, WARD <chr>, INSPECTIONFLAG <chr>,
## #   SERVICEREQUESTID <chr>, INSPECTORNAME <chr>, SERVICECALLCOUNT <dbl>,
## #   MARADDRESSREPOSITORYID <dbl>, ZIPCODE <dbl>, YCOORD <dbl>,
## #   SERVICETYPECODEDESCRIPTION <chr>, STATUS_CODE <chr>,
## #   SERVICECODE <chr>, SERVICEORDERSTATUS <chr>,
## #   ORGANIZATIONACRONYM <chr>, `service-text` <chr>, ADDDATE <dttm>,
## #   SERVICEORDERDATE <dttm>, CITY <chr>, ANC <chr>, STREETADDRESS <chr>,
## #   location <chr>, LATITUDE <dbl>
```

```
attach(threeoneone_2)
threeoneone_2long <- threeoneone_2$LONGITUDE
threeoneone_2lat <- threeoneone_2$LATITUDE
```

##Crash Data Import

```
crashdata <- read.csv("https://opendata.arcgis.com/datasets/70392a096a8e431381f1f692aaa0
6afd_24.csv")
head(crashdata)
```

```
##           X        Y  OBJECTID  CRIMEID      CCN               REPORTDATE
## 1 -76.99084 38.92293 122658274 27425626 17155492 2017-09-08T16:11:58.000Z
## 2 -77.04491 38.89731 122658275 27425632 17155477 2017-09-08T16:27:07.000Z
## 3 -77.02599 38.91701 122658276 27425633 17155485 2017-09-08T16:29:50.000Z
## 4 -76.95123 38.89235 122658277 27425641 17155482 2017-09-08T16:52:55.000Z
## 5 -77.00906 38.90860 122658278 27425644 17151390 2017-09-08T17:25:07.000Z
## 6 -76.97901 38.86988 122658279 27949933 19025311 2019-02-12T22:29:32.000Z
##    ROUTEID MEASURE OFFSET STREETSEGID ROADWAYSEGID
## 1 12075462 1778.59  10.64        9161         9846
## 2 11002002  477.73   0.12        3099        20113
## 3 11001002 2421.74  39.43          -9        30902
## 4 12061162  322.15  41.75        9262        18023
## 5 15065322 1562.50  23.75        7841         2562
## 6 13001802 1155.39  20.67          -9        28374
##                    FROMDATE TODATE  MARID                      ADDRESS
## 1 2017-09-08T04:00:00.000Z     NA  75717     1000 RHODE ISLAND AVE NE
## 2 2017-09-08T04:00:00.000Z     NA 242821         F ST NW & 20TH ST NW
## 3 2017-09-08T04:00:00.000Z     NA 300313             1000 U STREET NW
## 4 2017-09-08T04:00:00.000Z     NA  30064     3820 MINNESOTA AVENUE NE
## 5 2017-09-01T04:00:00.000Z     NA 236905 1300 NORTH CAPITOL STREET NW
## 6 2019-02-12T05:00:00.000Z     NA 286349          1617 18TH STREET SE
##   LATITUDE LONGITUDE   XCOORD   YCOORD   WARD
## 1 38.92293 -76.99084 400794.3 139450.3 Ward 5
## 2 38.89730 -77.04491 396104.2 136606.6 Ward 2
## 3 38.91692 -77.02649 397702.4 138783.8 Ward 1
## 4 38.89245 -76.95169 404196.9 136065.7 Ward 7
## 5 38.90857 -77.00933 399191.0 137856.4 Ward 5
## 6 38.86987 -76.97946 401782.8 133560.6 Ward 8
##                                    EVENTID                   MAR_ADDRESS
## 1 {3090DBE7-311B-415B-BE38-ED6E483E8EF4}  1013 RHODE ISLAND AVENUE NE
## 2 {686BC596-3373-4577-BB88-22DC127AEA95}             1922 F STREET NW
## 3 {4D5AED1D-20BD-4511-BE2B-7E420DA49471}             1020 U STREET NW
## 4 {2FFDD224-2941-4C4C-A91F-339D37AB8F4C}     3820 MINNESOTA AVENUE NE
## 5 {9E820E70-BC37-4E32-80B3-651E7A1C8A91} 1400 NORTH CAPITOL STREET NW
## 6 {0C57B266-E5E6-4F53-BFA5-B177B19F2EFB}          1617 18TH STREET SE
##   MAR_SCORE MAJORINJURIES_BICYCLIST MINORINJURIES_BICYCLIST
## 1       200                       0                       0
## 2       200                       0                       0
## 3       200                       0                       0
## 4       200                       0                       0
## 5       200                       0                       0
## 6       200                       0                       0
##   UNKNOWNINJURIES_BICYCLIST FATAL_BICYCLIST MAJORINJURIES_DRIVER
## 1                         0               0                    0
## 2                         0               0                    0
## 3                         0               0                    0
## 4                         0               0                    0
## 5                         0               0                    0
## 6                         0               0                    0
##   MINORINJURIES_DRIVER UNKNOWNINJURIES_DRIVER FATAL_DRIVER
## 1                    0                       0            0
## 2                    0                       0            0
## 3                    0                       0            0
```

```
## 4                    0                     0            0
## 5                    0                     0            0
## 6                    0                     0            0
##   MAJORINJURIES_PEDESTRIAN MINORINJURIES_PEDESTRIAN
## 1                        0                        0
## 2                        0                        0
## 3                        0                        0
## 4                        0                        0
## 5                        0                        0
## 6                        0                        0
##   UNKNOWNINJURIES_PEDESTRIAN FATAL_PEDESTRIAN TOTAL_VEHICLES
## 1                          0                0              2
## 2                          0                0              2
## 3                          0                0              2
## 4                          0                0              2
## 5                          0                0              2
## 6                          0                0              2
##   TOTAL_BICYCLES TOTAL_PEDESTRIANS PEDESTRIANSIMPAIRED BICYCLISTSIMPAIRED
## 1              0                 0                   0                   0
## 2              0                 0                   0                   0
## 3              0                 0                   0                   0
## 4              0                 0                   0                   0
## 5              0                 0                   0                   0
## 6              0                 0                   0                   0
##   DRIVERSIMPAIRED TOTAL_TAXIS TOTAL_GOVERNMENT SPEEDING_INVOLVED
## 1               0           0                0                 0
## 2               0           0                0                 0
## 3               0           0                0                 0
## 4               0           0                1                 0
## 5               0           0                0                 0
## 6               0           0                1                 0
##   NEARESTINTROUTEID NEARESTINTSTREETNAME OFFINTERSECTION
## 1          12001202           12TH ST NE           60.02
## 2          11034382              F ST NW            5.85
## 3          11087232              U ST NW            0.00
## 4          12016252         BLAINE ST NE            6.44
## 5          25065321  NORTH CAPITOL ST NW           65.25
## 6          13074162              R ST SE           44.06
##   INTAPPROACHDIRECTION LOCATIONERROR           LASTUPDATEDATE MPDLATITUDE
## 1            Southwest                                            38.92284
## 2                South                                            38.89730
## 3                 East                                            38.91692
## 4                North               2018-05-19T14:33:02.000Z    38.89253
## 5                South                                            38.90859
## 6                North               2019-02-14T15:35:37.000Z    38.86987
##   MPDLONGITUDE  MPDGEOX  MPDGEOY                          BLOCKKEY
## 1    -76.99080 402454.8 139769.2 e707902cdd21f2d9e6306bb695b42abd
## 2    -77.04491 396104.2 136612.5 ecbada6bea27549f54bdd065374d2a35
## 3    -77.02643 397719.0 138758.1 0076d81baa4ba918d197f570e5ea9607
## 4    -76.95164 404209.5 136090.5 98f9b5e9a572f39bced9400a8b63729c
## 5    -77.00933 399192.0 137735.8 f9a06c7c49f480d7b6105a9fd7a71db5
## 6    -76.97925 401800.9 133560.7 494de19ebd13376397c7d715e2ba8369
##                       SUBBLOCKKEY FATALPASSENGER MAJORINJURIESPASSENGER
## 1 e707902cdd21f2d9e6306bb695b42abd            NA                    NA
```

```
## 2 5bd2628a60143421f02b35bbffbf6879             NA                      NA
## 3 0076d81baa4ba918d197f570e5ea9607             NA                      NA
## 4 98f9b5e9a572f39bced9400a8b63729c             NA                      NA
## 5 ba505ddbbcdb1957aa8e320b66045e99             NA                      NA
## 6 494de19ebd13376397c7d715e2ba8369             NA                      NA
##   MINORINJURIESPASSENGER UNKNOWNINJURIESPASSENGER
## 1                     NA                       NA
## 2                     NA                       NA
## 3                     NA                       NA
## 4                     NA                       NA
## 5                     NA                       NA
## 6                     NA                       NA
```

```
attach(crashdata)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     X
```

```
## The following objects are masked from threeoneone_2:
##
##     LATITUDE, LONGITUDE, WARD, XCOORD, YCOORD
```

##Data Wrangling

```
threeoneone_2$ymd <- ymd_hms(ADDDATE)
threeoneone_2[complete.cases(threeoneone_2), ]
```

```
## # A tibble: 0 x 32
## # … with 32 variables: XCOORD <dbl>, SERVICECODEDESCRIPTION <chr>,
## #   DETAILS <chr>, LONGITUDE <dbl>, STATE <chr>, PRIORITY <chr>,
## #   RESOLUTIONDATE <dttm>, INSPECTIONDATE <dttm>, SERVICEDUEDATE <dttm>,
## #   YEAR <dbl>, WARD <chr>, INSPECTIONFLAG <chr>, SERVICEREQUESTID <chr>,
## #   INSPECTORNAME <chr>, SERVICECALLCOUNT <dbl>,
## #   MARADDRESSREPOSITORYID <dbl>, ZIPCODE <dbl>, YCOORD <dbl>,
## #   SERVICETYPECODEDESCRIPTION <chr>, STATUS_CODE <chr>,
## #   SERVICECODE <chr>, SERVICEORDERSTATUS <chr>,
## #   ORGANIZATIONACRONYM <chr>, `service-text` <chr>, ADDDATE <dttm>,
## #   SERVICEORDERDATE <dttm>, CITY <chr>, ANC <chr>, STREETADDRESS <chr>,
## #   location <chr>, LATITUDE <dbl>, ymd <dttm>
```

```
crashdata_clean_date <- gsub("T.*", "", crashdata$FROMDATE)
crashdata$crashdate <- ymd(crashdata_clean_date)
crashdata[complete.cases(crashdata), ]
```

```
##  [1] X                          Y
##  [3] OBJECTID                   CRIMEID
##  [5] CCN                        REPORTDATE
##  [7] ROUTEID                    MEASURE
##  [9] OFFSET                     STREETSEGID
## [11] ROADWAYSEGID               FROMDATE
## [13] TODATE                     MARID
## [15] ADDRESS                    LATITUDE
## [17] LONGITUDE                  XCOORD
## [19] YCOORD                     WARD
## [21] EVENTID                    MAR_ADDRESS
## [23] MAR_SCORE                  MAJORINJURIES_BICYCLIST
## [25] MINORINJURIES_BICYCLIST    UNKNOWNINJURIES_BICYCLIST
## [27] FATAL_BICYCLIST            MAJORINJURIES_DRIVER
## [29] MINORINJURIES_DRIVER       UNKNOWNINJURIES_DRIVER
## [31] FATAL_DRIVER               MAJORINJURIES_PEDESTRIAN
## [33] MINORINJURIES_PEDESTRIAN   UNKNOWNINJURIES_PEDESTRIAN
## [35] FATAL_PEDESTRIAN           TOTAL_VEHICLES
## [37] TOTAL_BICYCLES             TOTAL_PEDESTRIANS
## [39] PEDESTRIANSIMPAIRED        BICYCLISTSIMPAIRED
## [41] DRIVERSIMPAIRED            TOTAL_TAXIS
## [43] TOTAL_GOVERNMENT           SPEEDING_INVOLVED
## [45] NEARESTINTROUTEID          NEARESTINTSTREETNAME
## [47] OFFINTERSECTION            INTAPPROACHDIRECTION
## [49] LOCATIONERROR              LASTUPDATEDATE
## [51] MPDLATITUDE                MPDLONGITUDE
## [53] MPDGEOX                    MPDGEOY
## [55] BLOCKKEY                   SUBBLOCKKEY
## [57] FATALPASSENGER             MAJORINJURIESPASSENGER
## [59] MINORINJURIESPASSENGER     UNKNOWNINJURIESPASSENGER
## [61] crashdate
## <0 rows> (or 0-length row.names)
```

```
crashdatalong <- crashdata$LONGITUDE
crashdatalat <- crashdata$LATITUDE
```

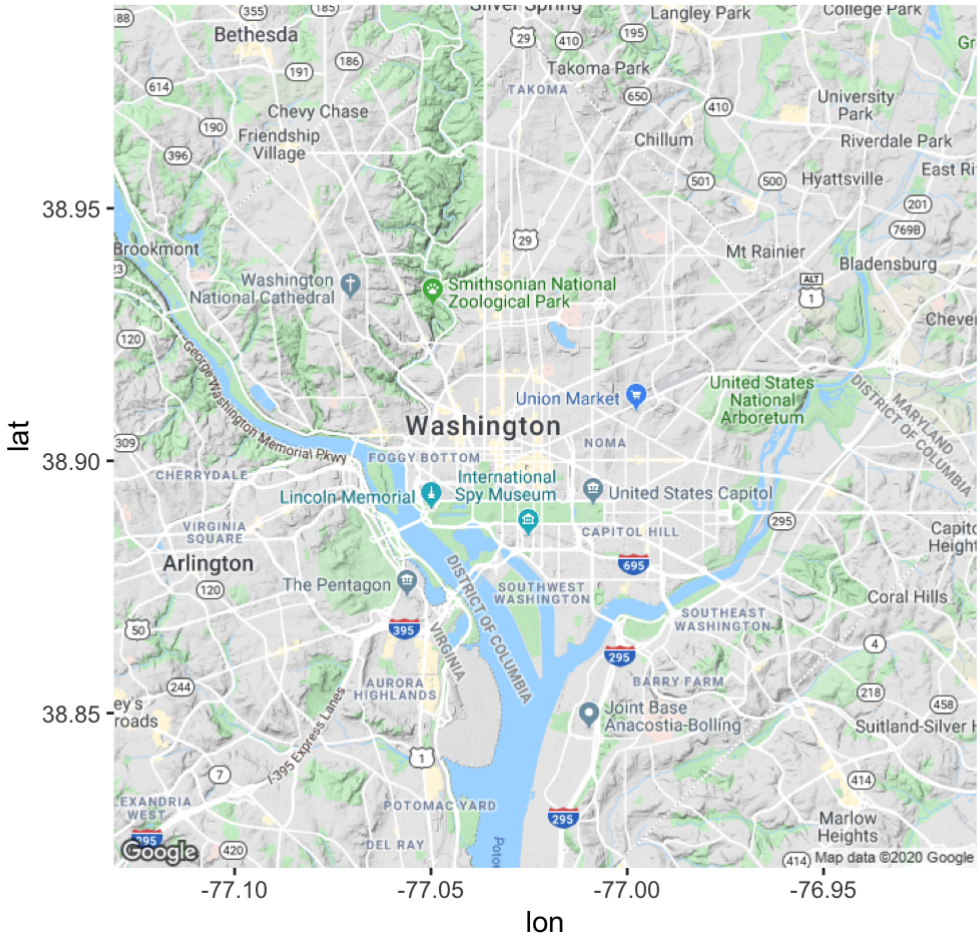
##Create DC Map with Google API

```
ggmap::register_google(key = "AIzaSyA3nim7i8YYK3_Bkz9oqFW1Vcg0rXDG7lY")
dcmap <- ggmap(get_map(location = c(lon = -77.02106, lat = 38.905), zoom = 12, scale = 1
, maptype = 'terrain', color = 'color'))
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=38.905,-77.02106&zoom=
12&size=640x640&scale=1&maptype=terrain&language=en-EN&key=xxx
```

##DC Map

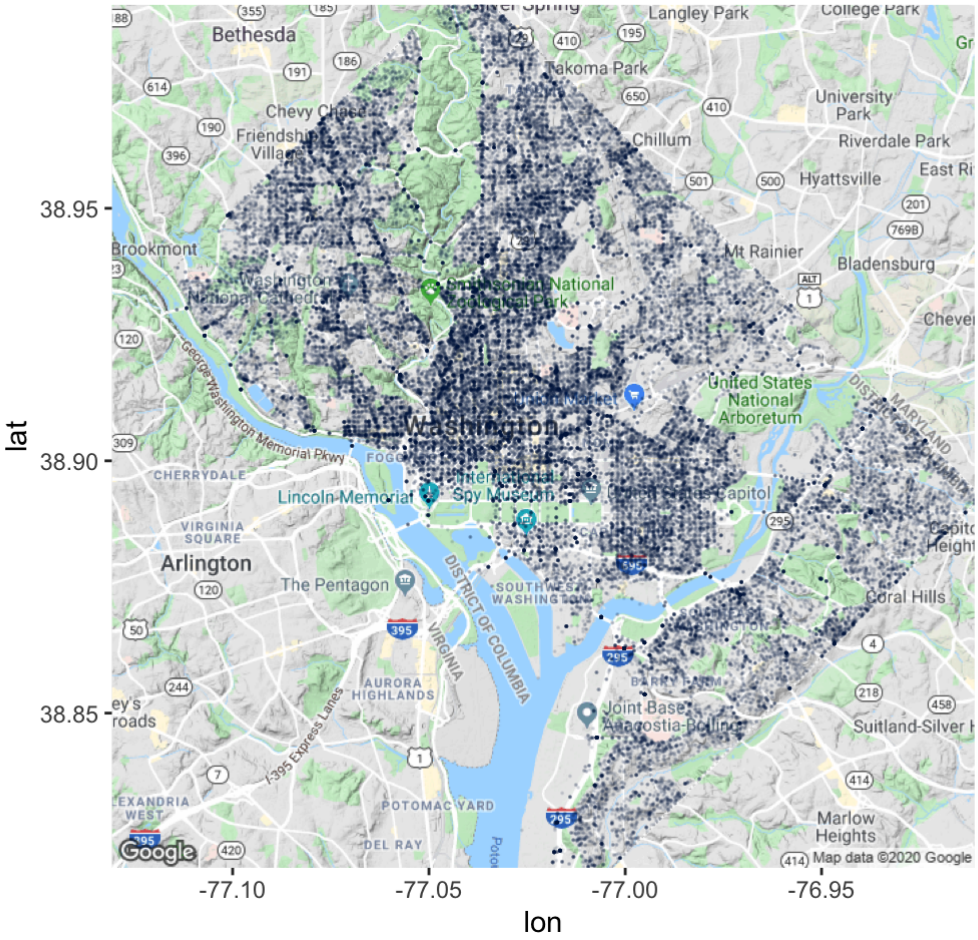
```
dcmap
```



##Threeoneone Requests in DC

```
dcmap_311 <- dcmap + geom_point(data = threeoneone_2, aes(x = threeoneone_2long, y = thr
eeoneone_2lat), color = col1, alpha = 0.1, size = 0.05) + theme(legend.position = "none"
)
dcmap_311
```

```
## Warning: Removed 593 rows containing missing values (geom_point).
```

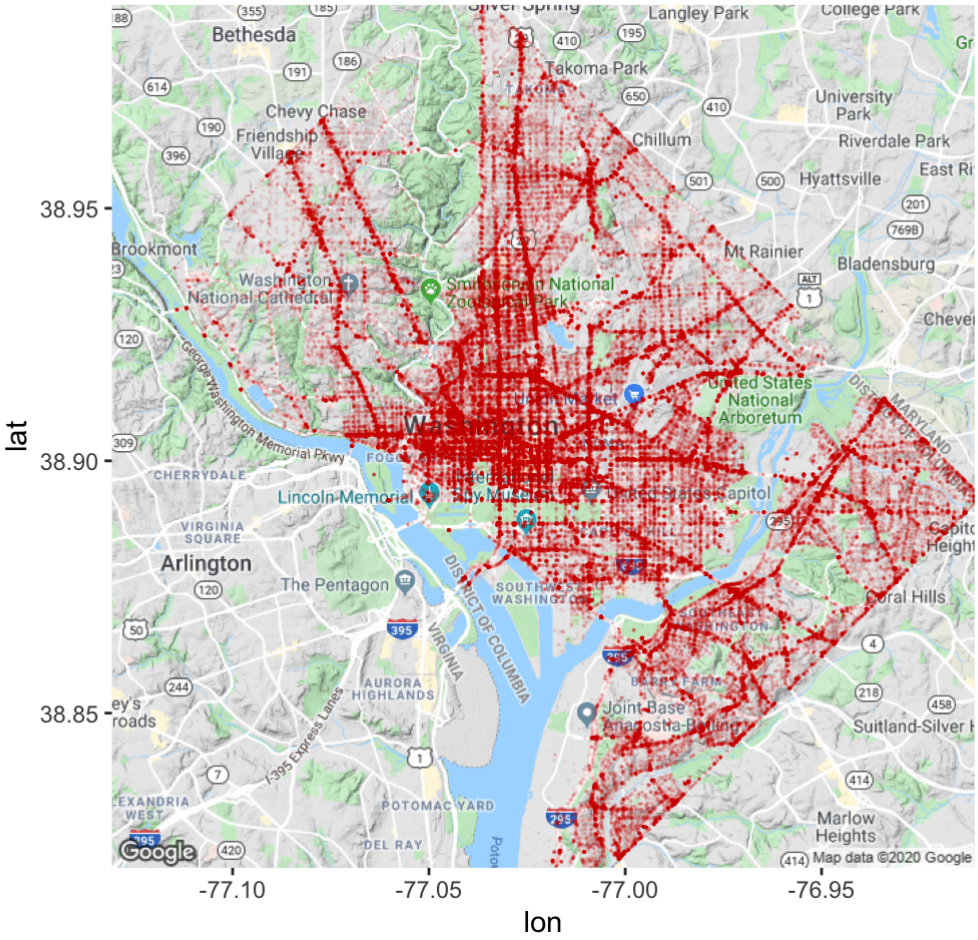


This map shows 311 requests in DC. Requests are concentrated in more urban, densely populated areas.

##Crashes in DC

```
dcmap_crashes <- dcmap + geom_point(data = crashdata, aes(x = crashdatalong, y = crashda
talat), color = col4, alpha = 0.05, size = 0.05) + theme(legend.position = "none")
dcmap_crashes
```

```
## Warning: Removed 557 rows containing missing values (geom_point).
```



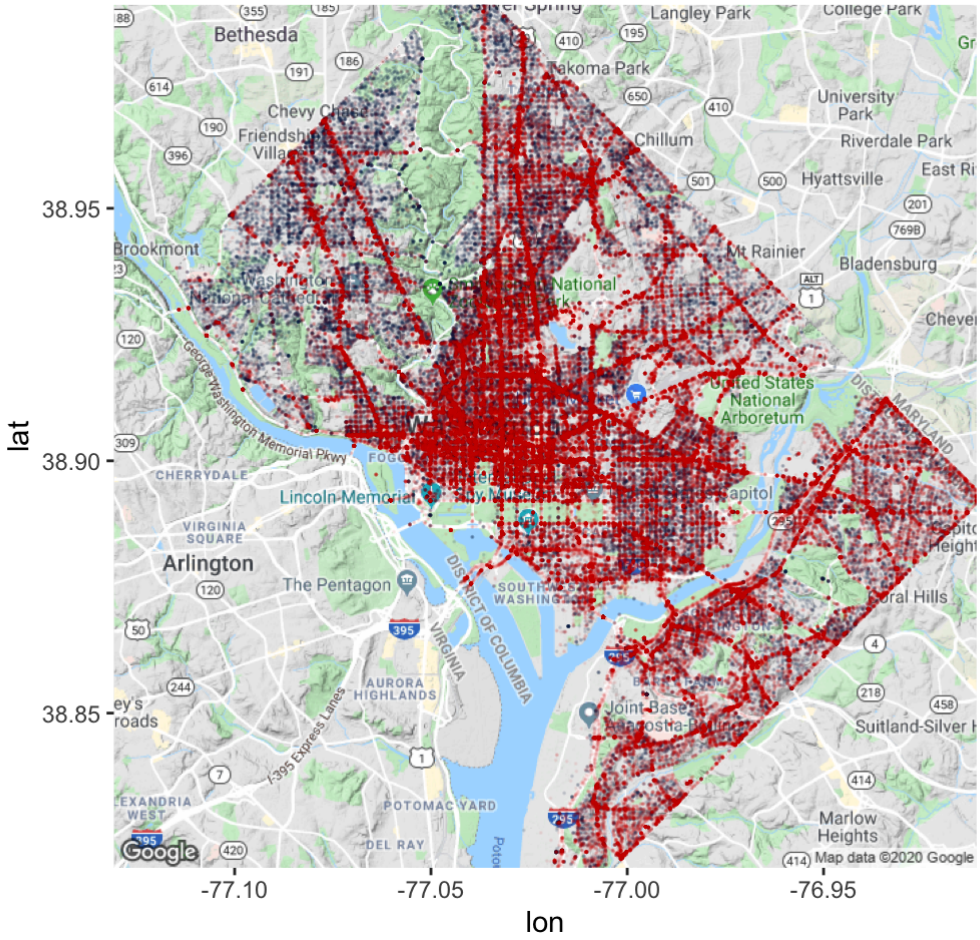
This map shows the recorded crashes in DC. Crashes are concentrated on busier roads and in the center of DC.

##Threeoneone and Crashes in DC

```
dcmap_311_crashes <- dcmap + geom_point(data = threeoneone_2, aes(x = threeoneone_2long, y = threeoneone_2lat), color = col1, alpha = 0.1, size = 0.05) + geom_point(data = crash data, aes(x = crashdatalong, y = crashdatalat), color = col4, alpha = 0.05, size = 0.05) + theme(legend.position = "none")
dcmap_311_crashes
```

```
## Warning: Removed 593 rows containing missing values (geom_point).
```

```
## Warning: Removed 557 rows containing missing values (geom_point).
```



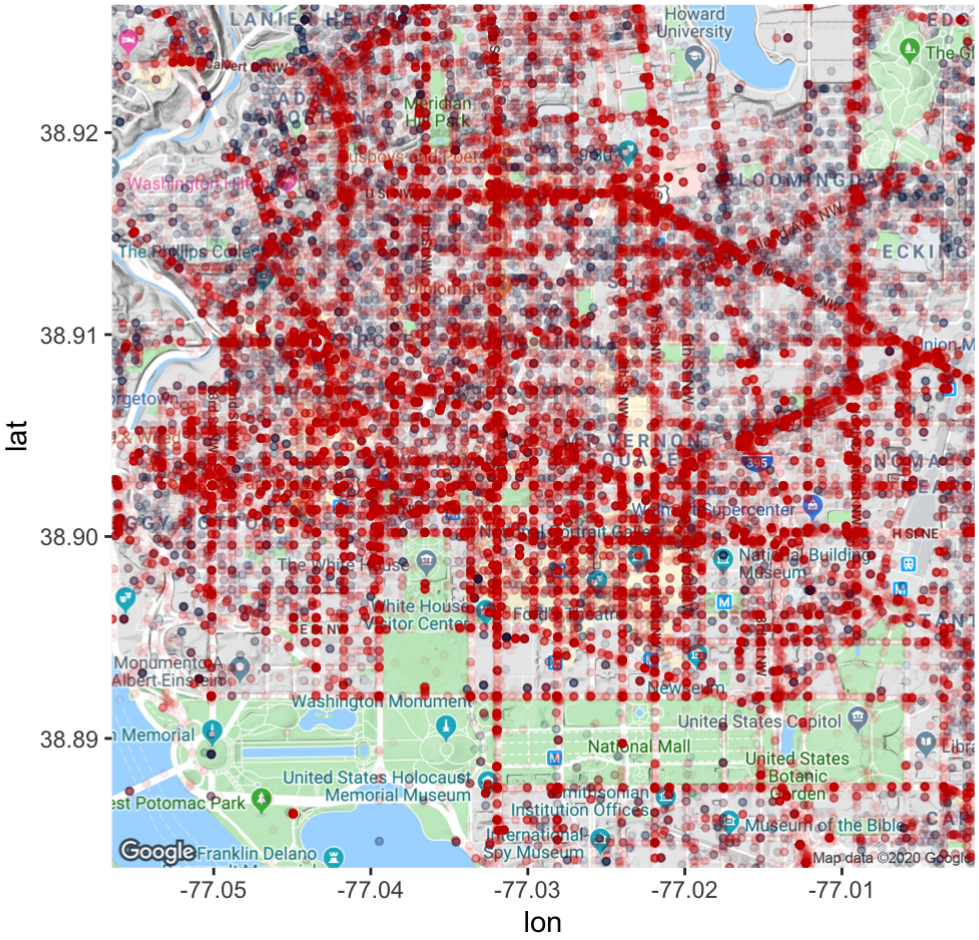
This map overlays crashes with service requests.

##Detail of Threeoneone and Crashes in Downtown DC

```
dcmap_dt <- ggmap(get_map(location = c(lon = -77.02906, lat = 38.905), zoom = 14, scale
 = 2, maptype = 'terrain', color = 'color'))
dcmap_dt_311_crashes <- dcmap_dt + geom_point(data = threeoneone_2, aes(x = threeoneone_
2long, y = threeoneone_2lat), color = col1, alpha = 0.1, size = 1) + geom_point(data = c
rashdata, aes(x = crashdatalong, y = crashdatalat), color = col4, alpha = 0.05, size = 1
) + theme(legend.position = "none")
dcmap_dt_311_crashes
```

```
## Warning: Removed 50151 rows containing missing values (geom_point).
```

```
## Warning: Removed 147935 rows containing missing values (geom_point).
```



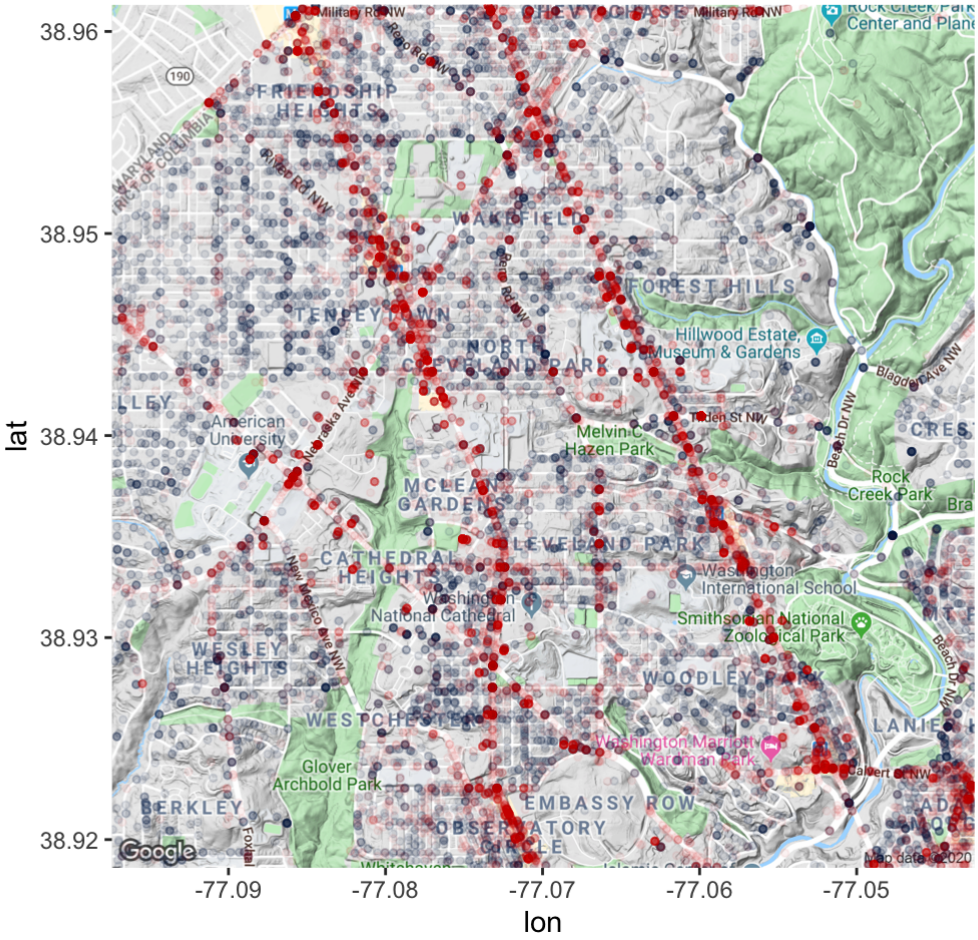
This map details the 311 requests and crashes in Downtown DC. In this area, the concentration of crashes is high, and the concentration of 311 requests is high (especially north of N St).

##Detail of Threeoneone and Crashes in NW DC

```
dcmap_nw <- ggmap(get_map(location = c(lon = -77.07, lat = 38.94), zoom = 14, scale = 2,
maptype = 'terrain', color = 'color'))
dcmap_nw_311_crashes <- dcmap_nw + geom_point(data = threeoneone_2, aes(x = threeoneone_
2long, y = threeoneone_2lat), color = col1, alpha = 0.1, size = 1) + geom_point(data = c
rashdata, aes(x = crashdatalong, y = crashdatalat), color = col4, alpha = 0.05, size = 1
) + theme(legend.position = "none")
dcmap_nw_311_crashes
```

```
## Warning: Removed 55925 rows containing missing values (geom_point).
```

```
## Warning: Removed 211350 rows containing missing values (geom_point).
```



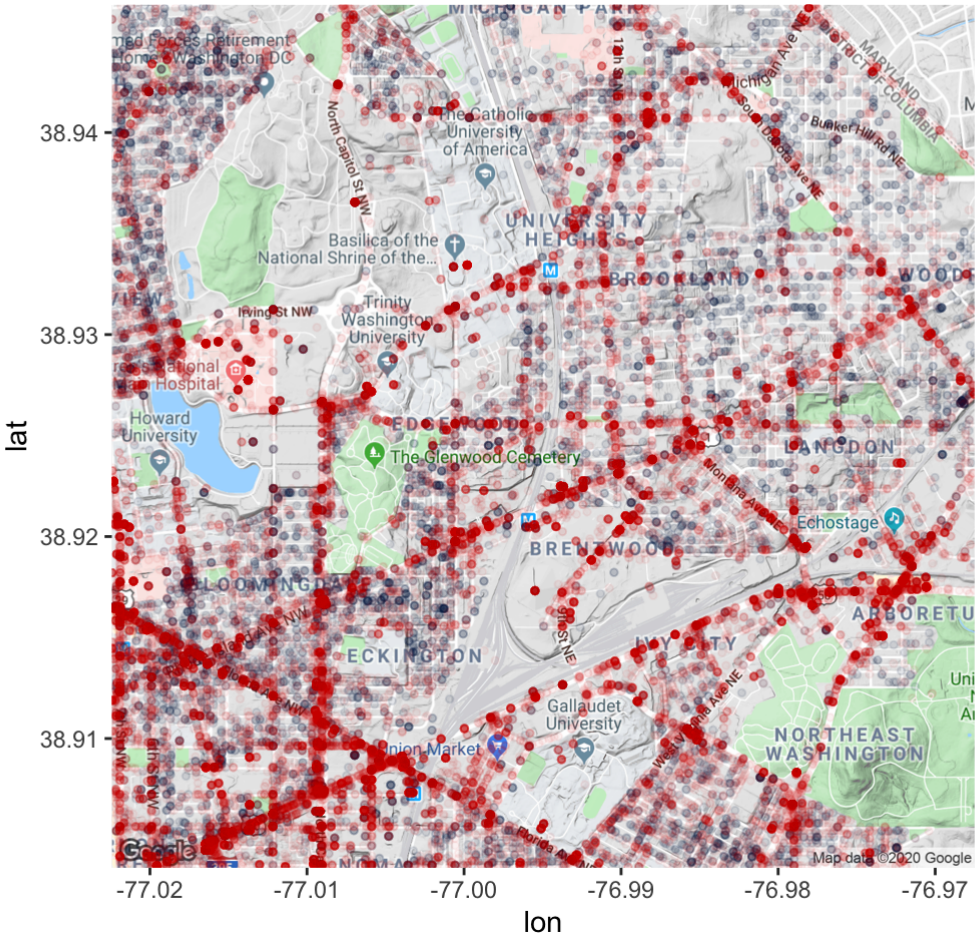
This map details the 311 requests and crashes in Northwest DC. In this area, crashes tend to be concentrated on busier roads and rarely occur within the suburbs. However, there is still a high concentration of 311 requests in this area.

##Detail of Threeoneone and Crashes in NE DC

```
dcmap_ne <- ggmap(get_map(location = c(lon = -76.995, lat = 38.925), zoom = 14, scale =
2, maptype = 'terrain', color = 'color'))
dcmap_ne_311_crashes <- dcmap_ne + geom_point(data = threeoneone_2, aes(x = threeoneone_
2long, y = threeoneone_2lat), color = col1, alpha = 0.1, size = 1) + geom_point(data = c
rashdata, aes(x = crashdatalong, y = crashdatalat), color = col4, alpha = 0.05, size = 1
) + theme(legend.position = "none")
dcmap_ne_311_crashes
```

```
## Warning: Removed 56338 rows containing missing values (geom_point).
```

```
## Warning: Removed 187003 rows containing missing values (geom_point).
```



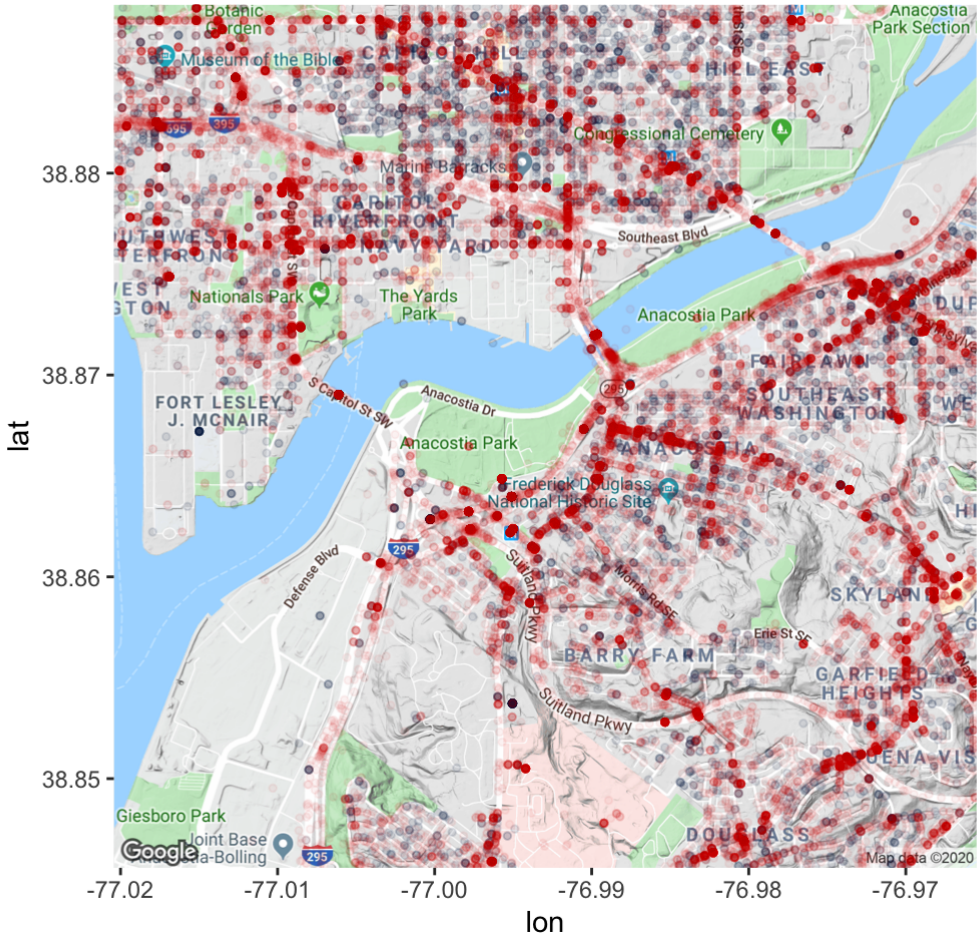
This map details the 311 requests and crashes in Northeast DC. While crashes are concentrated on busier roads, crashes still occur off of these roads. There are fewer 311 requests in this area compared to Northwest DC.

##Detail of Threeoneone and Crashes in SW/SE DC

```
dcmap_swse <- ggmap(get_map(location = c(lon = -76.993, lat = 38.867), zoom = 14, scale
 = 2, maptype = 'terrain', color = 'color'))
dcmap_swse_311_crashes <- dcmap_swse + geom_point(data = threeoneone_2, aes(x = threeone
one_2long, y = threeoneone_2lat), color = col1, alpha = 0.1, size = 1) + geom_point(data
= crashdata, aes(x = crashdatalong, y = crashdatalat), color = col4, alpha = 0.05, size
 = 1) + theme(legend.position = "none")
dcmap_swse_311_crashes
```

```
## Warning: Removed 59205 rows containing missing values (geom_point).
```

```
## Warning: Removed 193663 rows containing missing values (geom_point).
```



This map details the 311 requests and crashes in Southwest/Southeast DC. In this area there are high concentrations of crashes on the busier roads and highways and 311 requests are concentrated in batches.

##Clustering Crash Coordinates (Pre-Cluster Wrangling)

```
crashdata <- crashdata %>% filter( X > -77.11, X < -76.89)
crashdata <- crashdata %>% filter( Y < 39, Y > 38.7)
crashx <- crashdata$X
crashy <- crashdata$Y
crashxy <- matrix(c(crashx,crashy), ncol = 2)
```

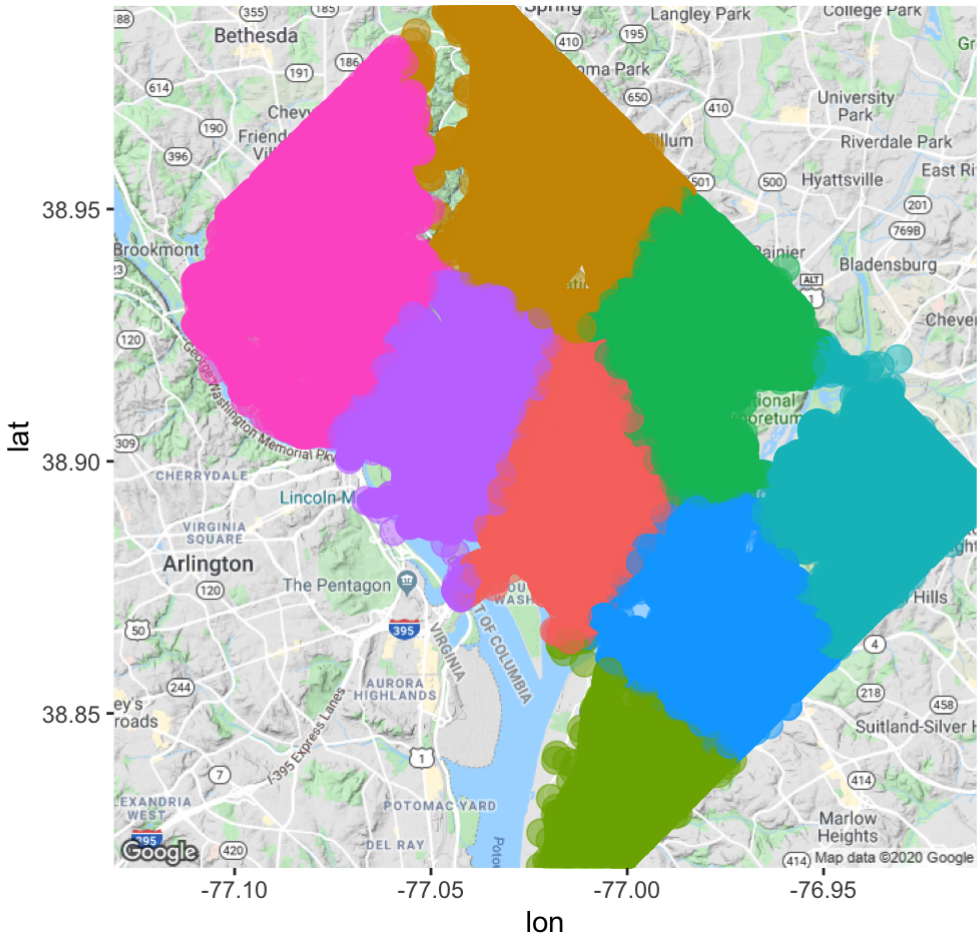
#Making Two Sets of Clusters

```
crash_clusterinfo_8 <- kmeans(crashxy, 8)
crash_clusterinfo_15 <- kmeans(crashxy, 15)
crashxy_8 <- augment(crash_clusterinfo_8, crashxy)
crashxy_15 <- augment(crash_clusterinfo_15, crashxy)
```

#Mapping the Clusters

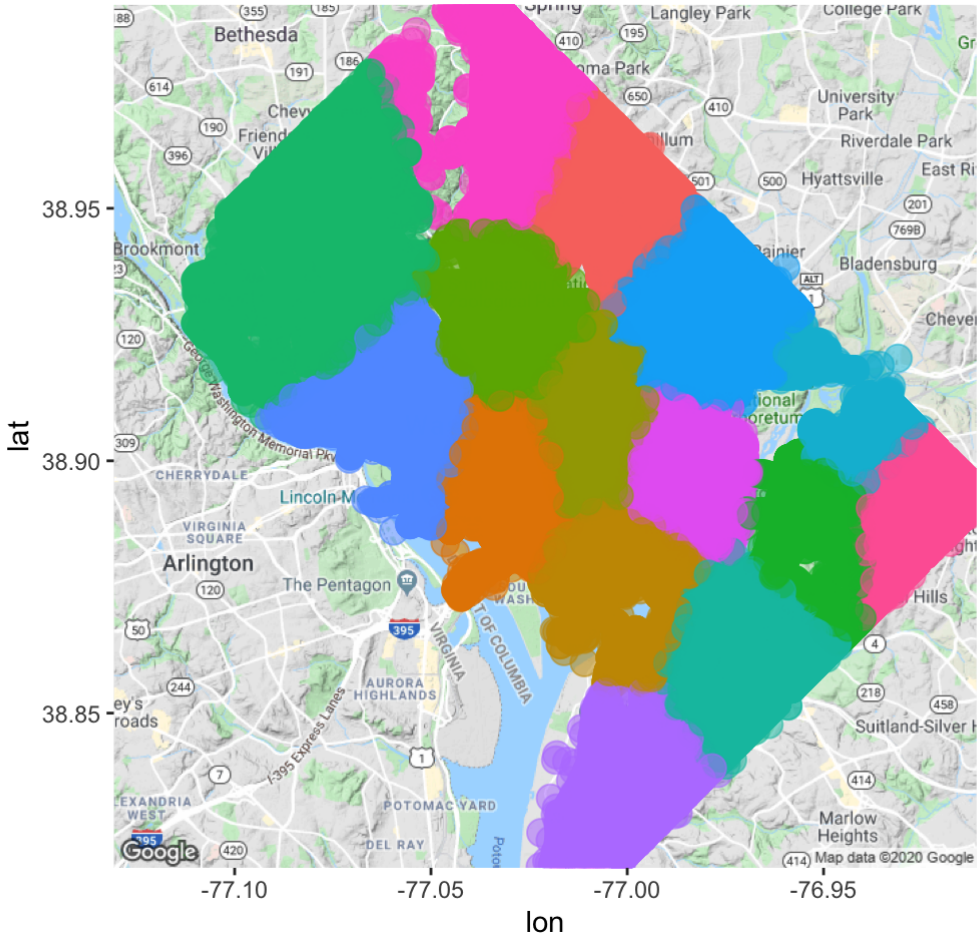
```
dcmap_crash_kcluster_8 <- dcmap + geom_point(data = crashxy_8, aes(x = crashxy_8$X1, y = crashxy_8$X2, color = crashxy_8$.cluster, alpha = 0.01, size = 0.001)) + theme(legend.position = "none")
dcmap_crash_kcluster_15 <- dcmap + geom_point(data = crashxy_15, aes(x = crashxy_15$X1,  y = crashxy_15$X2, color = crashxy_15$.cluster, alpha = 0.01, size = 0.001)) + theme(legend.position = "none")
dcmap_crash_kcluster_8
```

```
## Warning: Removed 662 rows containing missing values (geom_point).
```




```
dcmap_crash_kcluster_15
```

```
## Warning: Removed 662 rows containing missing values (geom_point).
```



These maps shows clusters of crashes created by a K-means clustering. These can be compared to clustering of service requests to find areas of DC that have a high concentration of both service requests and crashes. However, the K-means clustering algorithm is imperfect, so a better algorithm such as DBSCAN on a machine with more computing power could be used to produce better results.

```
threeoneone <- read_csv("https://datagate.dc.gov/search/open/311requests?daterange=8years&details=true&format=csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   XCOORD = col_double(),
##   LONGITUDE = col_double(),
##   RESOLUTIONDATE = col_datetime(format = ""),
##   INSPECTIONDATE = col_datetime(format = ""),
##   SERVICEDUEDATE = col_datetime(format = ""),
##   YEAR = col_double(),
##   SERVICECALLCOUNT = col_double(),
##   MARADDRESSREPOSITORYID = col_double(),
##   ZIPCODE = col_double(),
##   YCOORD = col_double(),
##   ADDDATE = col_datetime(format = ""),
##   SERVICEORDERDATE = col_datetime(format = ""),
##   LATITUDE = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

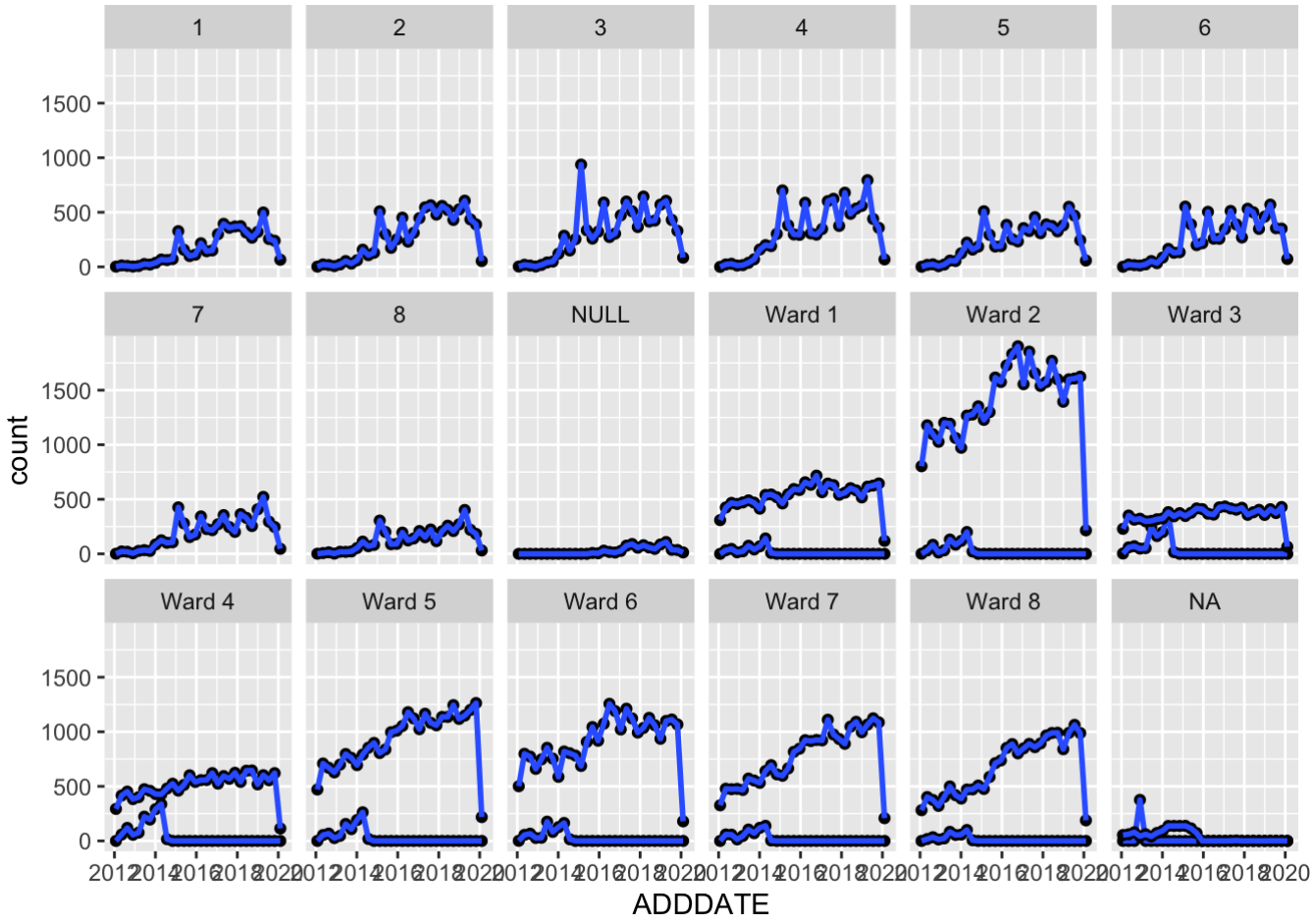
```
# sorts out the three types of service codes applicable to traffic data
trafficrequests <- filter(threeoneone, SERVICECODEDESCRIPTION == c("roadway signs","stre
etlight repair investigation","pothole"))
#head(trafficrequests)
```

```
# gives a new variable, INDEX, and assigns every row a 1 for counting purposes.
crashes <- crashes %>%
  mutate(INDEX = 1)
#View(crashes)
#same as above
trafficrequests <- trafficrequests %>%
  mutate(INDEX = 1)
#View(trafficrequests)
```

```
trafficrequests$time_elapsed <- as.Date(as.character(trafficrequests$SERVICEDUEDATE), fo
rmat= "%Y-%m-%d") -
  as.Date(as.character(trafficrequests$RESOLUTIONDATE), format = "%Y-%m-%d")
#View(trafficrequests)
```

```
trafficrequests$days_to_solve <- as.Date(as.character(trafficrequests$RESOLUTIONDATE), f
ormat= "%Y-%m-%d") -
  as.Date(as.character(trafficrequests$SERVICEORDERDATE), format = "%Y-%m-%d")
#View(trafficrequests)
```

```
# plots the frequency of traffic requests and crashes by date, faceted with wards using
 bins (it defaults to like 30 values binned into a single point).
ggplot() +
  geom_point(data = trafficrequests, aes(x = ADDDATE), stat = "bin") +
  geom_smooth(trafficrequests, mapping = aes(x = ADDDATE), stat = "bin") +
  geom_point(data = crashes, aes(x = FROMDATE), stat = "bin") +
  geom_smooth(crashes, mapping = aes(x = FROMDATE), stat = "bin") +
  facet_wrap(~ WARD, nrow=3)
```




```
# creates frequency tibble for crashes on each day
crashes_by_date <- crashes %>% group_by(FROMDATE,WARD) %>%
  summarise(dailycrashes = sum(INDEX))
# creates frequency tibble for 311 requests on each day
requests_by_date <- trafficrequests %>% group_by(ADDDATE,WARD) %>%
  summarise(dailyrequests = sum(INDEX))
# changes the head of FROMDATE in crashes and ADDDATE in requests to NEWDATE
requests_by_date <- requests_by_date %>% mutate(NEWDATE = as.Date(ADDDATE))
crashes_by_date <- crashes_by_date %>% mutate(NEWDATE = as.Date(FROMDATE))
```

```
by_date <- inner_join(crashes_by_date, requests_by_date, by = c('NEWDATE', 'WARD'))

by_date$dailycrashes[is.na(by_date$dailycrashes)] <- 0
by_date$dailyrequests[is.na(by_date$dailyrequests)] <- 0
```

```
# for the first regression, we want to do it without that extra ward variable. This does that by grouping by date and merging all wards.
without_ward <- by_date %>%
  group_by(NEWDATE) %>%
  summarize(dailycrashes = sum(dailycrashes), dailyrequests = sum(dailyrequests))
```

```
#ggpairs(data=by_date, columns=c(2,3,4,6), title="Running Models on Each Pair of Variables")
first_model <- lm(dailyrequests ~ dailycrashes, data = without_ward)
get_regression_table(first_model)
```

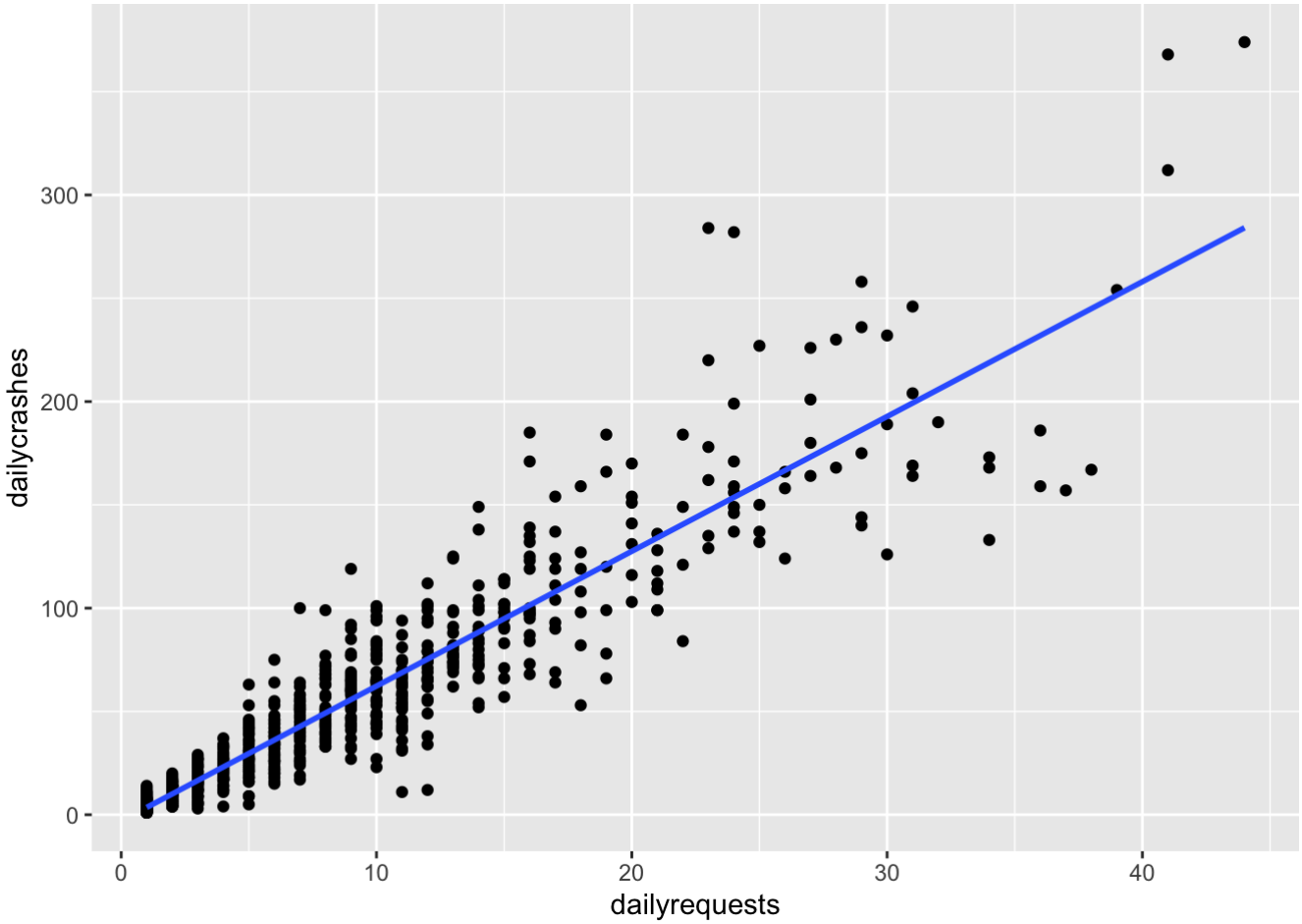
```
## # A tibble: 2 x 7
##   term         estimate std_error statistic p_value lower_ci upper_ci
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept       1.84      0.168      10.9       0    1.51     2.17
## 2 dailycrashes    0.129     0.002      60.7       0    0.125    0.134
```

```
#View(first_model)
get_regression_points(first_model)
```

```
## # A tibble: 681 x 5
##       ID dailyrequests dailycrashes dailyrequests_hat residual
##    <int>         <dbl>        <dbl>              <dbl>    <dbl>
##  1     1             1            4               2.36   -1.36
##  2     2             1            5               2.48   -1.48
##  3     3             1            2               2.10   -1.10
##  4     4             1            4               2.36   -1.36
##  5     5             1            9               3.00   -2.00
##  6     6             1            8               2.87   -1.87
##  7     7             2           16               3.91   -1.91
##  8     8             2            4               2.36   -0.356
##  9     9             1           13               3.52   -2.52
## 10    10             1            6               2.62   -1.62
## # … with 671 more rows
```

```
ggplot(without_ward, aes(x = dailyrequests, y = dailycrashes)) +
  geom_point() +
  geom_parallel_slopes(se = FALSE)
```

```
## Warning: `geom_parallel_slopes()` didn't recieve a grouping variable
## with more than one unique value. Make sure you supply one. Basic model is
## fitted.
```



Conventional wisdom holds that greater population densities correspond with greater volume of traffic incidents. However, the results from our linear regression model are generally consistent with our initial intuitions. Population does not capture the salient variables that can be used to predict traffic incidents in a given area. There is a clear and strong correlation between the 311 requests by day and the accident reports on the same day. This not only suggests that volume of traffic activity in a given area tends to correlate with a greater volume of traffic incidents, but also that 311 requests are a decent predictor of such volume. Future research should continue to explore how the nature of the 311 requests (such as their location, type of request, etc.) can offer information that would otherwise be unavailable in raw population data. Ward 2 posed a limitation to our model, as the density of traffic in this area is remarkably higher than any of the other wards during the day. Therefore, it is expected that this Ward would have far more traffic incidents even when controlling for the variables that may be contributing to them.

6 Implications

- Opportunity to analyze particularly dangerous intersections
- Increase usage of 311 complaints to address traffic andtransportation concerns
- Importance of responding to 311 requests
- Increase pedestrian safety infrastructure in crash-dense areas

D.C. may be one of the most traffic congested cities in the United States, but the causes of such traffic cannot be explained by population increase alone. 311 requests are instructive in terms of their volume, nature, and location. These can be used to develop a picture as to what different parts of D.C. are like, and which ones are being neglected. The state of a Ward's infrastructure, for instance, can be estimated by taking time elapsed from the time a service request was made and when it was resolved. We recommend that future research uses more variables within the 311 data to predict where accidents are more likely to occur. The 311 requests also provide a window into the interaction between civil society and state, which is suprisingly higher than anticipated.