

Final Report

Group 6

Lucas Anderton, Hannah Brown, Lucas Gorak, Edward Mikkelsen

1/10/2020

1 Research Question

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.

2 Data Collection Procedure

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.

```
Details <- read.csv("https://opendata.arcgis.com/datasets/70248b73c20f46b0a5ee895fc91d6222_25.csv")
crashes <- read_csv("https://opendata.arcgis.com/datasets/70392a096a8e431381f1f692aaa06afd_24.csv")
```

```

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   CCN = col_character(),
##   REPORTDATE = col_datetime(format = ""),
##   ROUTEID = col_character(),
##   FROMDATE = col_datetime(format = ""),
##   TODATE = col_logical(),
##   ADDRESS = col_character(),
##   WARD = col_character(),
##   EVENTID = col_character(),
##   MAR_ADDRESS = col_character(),
##   NEARESTINTROUTEID = col_character(),
##   NEARESTINTSTREETNAME = col_character(),
##   INTAPPROACHDIRECTION = col_character(),
##   LOCATIONERROR = col_character(),
##   LASTUPDATEDATE = col_datetime(format = ""),
##   BLOCKKEY = col_character(),
##   SUBBLOCKKEY = col_character(),
##   FATALPASSENGER = col_logical(),
##   MAJORINJURIESPASSENGER = col_logical(),
##   MINORINJURIESPASSENGER = col_logical(),
##   UNKNOWNINJURIESPASSENGER = col_logical()
## )

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

## Warning: 11 parsing failures.
##   row      col      expected actual
## 1061 MINORINJURIESPASSENGER 1/0/T/F/TRUE/FALSE      3 'https://opendata.arcgis.com/datasets/70392a09
## 1098 MINORINJURIESPASSENGER 1/0/T/F/TRUE/FALSE      2 'https://opendata.arcgis.com/datasets/70392a09
## 1106 MINORINJURIESPASSENGER 1/0/T/F/TRUE/FALSE      4 'https://opendata.arcgis.com/datasets/70392a09
## 1201 MINORINJURIESPASSENGER 1/0/T/F/TRUE/FALSE      2 'https://opendata.arcgis.com/datasets/70392a09
## 1215 MAJORINJURIESPASSENGER 1/0/T/F/TRUE/FALSE      2 'https://opendata.arcgis.com/datasets/70392a09
## ....
## See problems(...) for more details.

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

## 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.
```

3 Important/Interesting Facets of Data Processing

```
# sorts out the three types of service codes applicable to traffic data
trafficrequests <- filter(threeneone, SERVICECODEDESCRIPTION == c("roadway signs","streetlight repair :
head(trafficrequests)
```

```
## # A tibble: 6 x 31
## XCOORD SERVICECODEDESC... DETAILS LONGITUDE STATE PRIORITY RESOLUTIONDATE
## <dbl> <chr> <chr> <dbl> <chr> <chr> <dtm>
## 1 3.97e5 streetlight rep... <NA> -77.0 DC URGENT 2012-07-05 11:43:00
## 2 3.98e5 streetlight rep... <NA> -77.0 DC URGENT 2012-06-26 05:34:44
## 3 3.95e5 streetlight rep... <NA> -77.1 DC URGENT NA
## 4 3.99e5 streetlight rep... <NA> -77.0 DC URGENT 2012-06-26 10:04:04
## 5 3.97e5 streetlight rep... <NA> -77.0 DC URGENT 2012-06-19 07:48:44
## 6 3.97e5 streetlight rep... <NA> -77.0 DC URGENT 2012-06-21 07:49:00
## # ... with 24 more variables: INSPECTIONDATE <dtm>, SERVICEDUE DATE <dtm>,
## # 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 <dtm>,
## # SERVICEORDERDATE <dtm>, CITY <chr>, ANC <chr>, STREETADDRESS <chr>,
## # location <chr>, LATITUDE <dbl>
```

```
crashes$WARD[crashes$WARD == "Null"] <- NA
summary(is.na(crashes))
```

```
## X Y OBJECTID CRIMEID
## Mode :logical Mode :logical Mode :logical Mode :logical
## FALSE:224403 FALSE:224403 FALSE:224403 FALSE:224403
##
## CCN REPORTDATE ROUTEID MEASURE
## Mode :logical Mode :logical Mode :logical Mode :logical
## FALSE:224403 FALSE:223322 FALSE:224403 FALSE:224403
## TRUE :1081
## OFFSET STREETSEGID ROADWAYSEGID FROMDATE TODATE
## Mode :logical Mode :logical Mode :logical Mode :logical Mode:logical
## FALSE:224403 FALSE:208184 FALSE:208184 FALSE:224396 TRUE:224403
## TRUE :16219 TRUE :16219 TRUE :7
## MARID ADDRESS LATITUDE LONGITUDE
## Mode :logical Mode :logical Mode :logical Mode :logical
## FALSE:224403 FALSE:224330 FALSE:224393 FALSE:224393
## TRUE :73 TRUE :10 TRUE :10
## XCOORD YCOORD WARD EVENTID
```

```

## Mode :logical      Mode :logical      Mode :logical      Mode :logical
## FALSE:224403      FALSE:224403      FALSE:223002      FALSE:224403
##
## MAR_ADDRESS      MAR_SCORE      MAJORINJURIES_BICYCLIST
## Mode :logical      Mode :logical      Mode :logical
## FALSE:214289      FALSE:224403      FALSE:224403
## TRUE :10114
## MINORINJURIES_BICYCLIST UNKNOWNINJURIES_BICYCLIST FATAL_BICYCLIST
## Mode :logical      Mode :logical      Mode :logical
## FALSE:224403      FALSE:224403      FALSE:224403
##
## MAJORINJURIES_DRIVER MINORINJURIES_DRIVER UNKNOWNINJURIES_DRIVER
## Mode :logical      Mode :logical      Mode :logical
## FALSE:224403      FALSE:224403      FALSE:224403
##
## FATAL_DRIVER      MAJORINJURIES_PEDESTRIAN MINORINJURIES_PEDESTRIAN
## Mode :logical      Mode :logical      Mode :logical
## FALSE:224403      FALSE:224403      FALSE:224403
##
## UNKNOWNINJURIES_PEDESTRIAN FATAL_PEDESTRIAN TOTAL_VEHICLES TOTAL_BICYCLES
## Mode :logical      Mode :logical      Mode :logical      Mode :logical
## FALSE:224403      FALSE:224403      FALSE:224403      FALSE:224403
##
## TOTAL_PEDESTRIANS PEDESTRIANSIMPAIRED BICYCLISTSIMPAIRED DRIVERSIMPAIRED
## Mode :logical      Mode :logical      Mode :logical      Mode :logical
## FALSE:224403      FALSE:224403      FALSE:224403      FALSE:224403
##
## TOTAL_TAXIS      TOTAL_GOVERNMENT SPEEDING_INVOLVED NEARESTINTROUTEID
## Mode :logical      Mode :logical      Mode :logical      Mode :logical
## FALSE:224403      FALSE:224403      FALSE:224403      FALSE:224403
##
## NEARESTINTSTREETNAME OFFINTERSECTION INTAPPROACHDIRECTION LOCATIONERROR
## Mode :logical      Mode :logical      Mode :logical      Mode :logical
## FALSE:224403      FALSE:224403      FALSE:224403      FALSE:9521
##
## TRUE :214882
## LASTUPDATEDATE MPDLATITUDE MPDLONGITUDE MPDGEOX
## Mode :logical      Mode :logical      Mode :logical      Mode :logical
## FALSE:74261      FALSE:157946      FALSE:157946      FALSE:105341
## TRUE :150142      TRUE :66457      TRUE :66457      TRUE :119062
## MPDGEOY      BLOCKKEY      SUBBLOCKKEY      FATALPASSENGER
## Mode :logical      Mode :logical      Mode :logical      Mode :logical
## FALSE:105341      FALSE:224403      FALSE:224403      FALSE:16660
## TRUE :119062      TRUE :207743
## MAJORINJURIESPASSENGER MINORINJURIESPASSENGER UNKNOWNINJURIESPASSENGER
## Mode :logical      Mode :logical      Mode :logical
## FALSE:16659      FALSE:16651      FALSE:16659
## TRUE :207744      TRUE :207752      TRUE :207744

```

```
#crashes2 <- na.omit(crashes)
```

```
# throw out columns
```

```
crashes$LOCATIONERROR <- crashes$LASTUPDATEDATE <- crashes$MPDLATITUDE <- crashes$MPDLONGITUDE <- crash
```

```
# sets the 2012 starting line for crashes data since it goes back early
```

```
crashes <- crashes %>% filter(as.Date(FROMDATE) >= "2012-01-01")
```

```
# gives a new variable, INDEX, and assigns every row a 1 for counting purposes.
```

```
crashes <- crashes %>%
  mutate(INDEX = 1)
```

```
#same as above
```

```
trafficrequests <- trafficrequests %>%
  mutate(INDEX = 1)
```

```
# fixes ward number problem. Some requests were labeled integers, but all the crashes data was 'Ward X'
```

```
trafficrequests %>%
  mutate(`WARD` = recode(WARD, "1" = "Ward 1",
                           "2" = "Ward 2",
                           "3" = "Ward 3",
                           "4" = "Ward 4",
                           "5" = "Ward 5",
                           "6" = "Ward 6",
                           "7" = "Ward 7",
                           "8" = "Ward 8")) ->
```

```
  trafficrequests
head(trafficrequests$WARD)
```

```
## [1] "Ward 1" "Ward 4" "Ward 4" "Ward 5" "Ward 1" "Ward 4"
```

```
na.omit(trafficrequests)
```

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

```
na.omit(crashes) # how do we omit all NULL and NA values?
```

```
## # A tibble: 15,228 x 48
##       X      Y OBJECTID CRIMEID CCN   REPORTDATE      ROUTEID MEASURE OFFSET
##   <dbl> <dbl>   <dbl>   <dbl> <chr> <dtm>         <chr>     <dbl> <dbl>
## 1 -77.0  38.9   1.23e8  2.83e7 2000... 2020-01-07 18:13:25 110013...   318.   20.4
## 2 -77.0  38.9   1.23e8  2.83e7 2000... 2020-01-07 18:37:42 110508...  1115.    4.75
## 3 -77.0  38.9   1.23e8  2.83e7 2000... 2020-01-07 19:00:01 130008...  1059.   22.4
## 4 -77.0  38.9   1.23e8  2.83e7 2000... 2020-01-07 19:18:53 110477...  1041.    8.4
## 5 -77.0  38.9   1.23e8  2.83e7 2000... 2020-01-07 19:36:36 150484...  3701.    0.02
## 6 -77.0  38.9   1.23e8  2.83e7 2000... 2020-01-08 20:31:32 120646...   57.3  10.8
## 7 -77.1  38.9   1.23e8  2.83e7 2000... 2020-01-08 20:45:18 110851...   219.  10.4
```

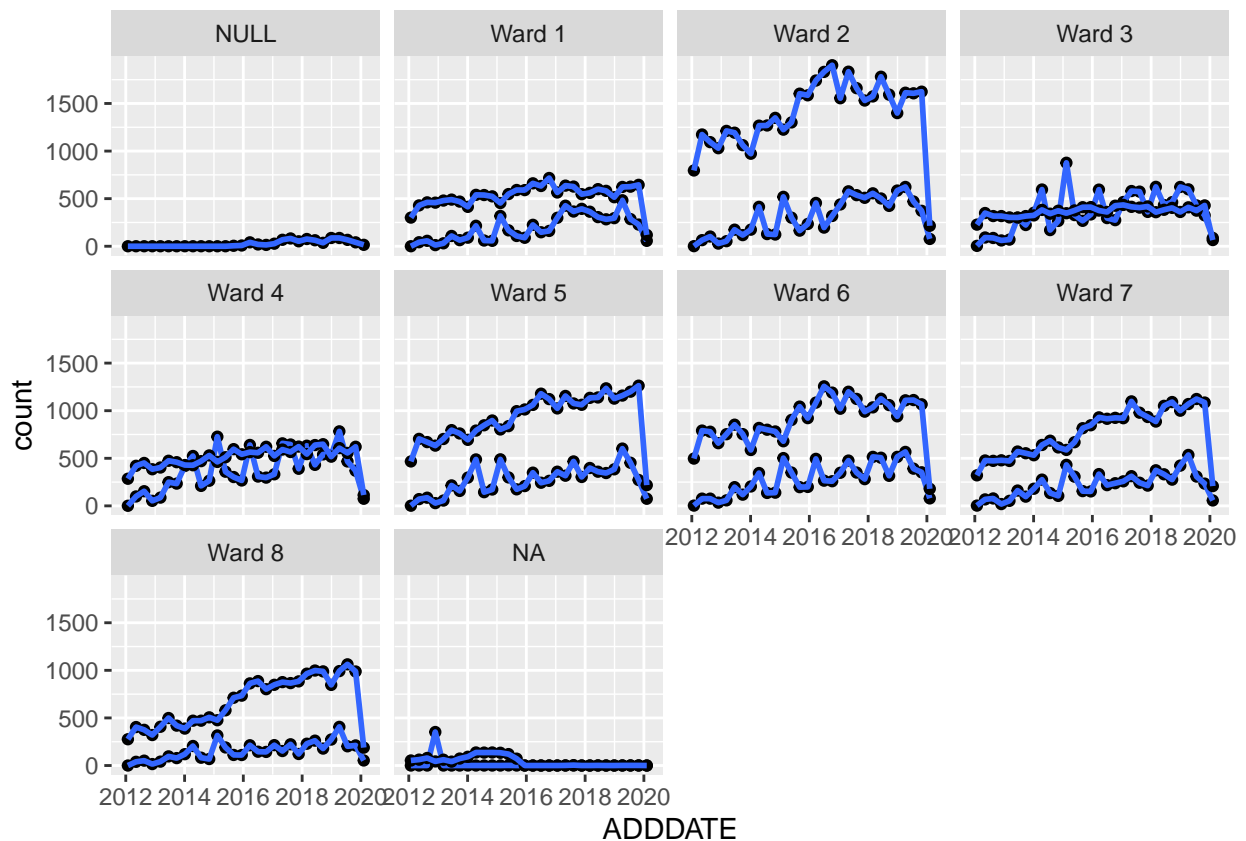
```
## 8 -77.0 38.9 1.23e8 2.83e7 2000... 2020-01-08 20:39:31 120319... 87.2 27.3
## 9 -77.0 38.9 1.23e8 2.83e7 2000... 2020-01-08 11:36:27 130407... 1942. 3.85
## 10 -77.0 38.9 1.23e8 2.83e7 2000... 2020-01-08 20:52:49 130698... 2861. 47.9
## # ... with 15,218 more rows, and 39 more variables: FROMDATE <dtm>, MARID <dbl>,
## # ADDRESS <chr>, LATITUDE <dbl>, LONGITUDE <dbl>, XCOORD <dbl>, YCOORD <dbl>,
## # WARD <chr>, EVENTID <chr>, MAR_SCORE <dbl>, MAJORINJURIES_BICYCLIST <dbl>,
## # MINORINJURIES_BICYCLIST <dbl>, UNKNOWNINJURIES_BICYCLIST <dbl>,
## # FATAL_BICYCLIST <dbl>, MAJORINJURIES_DRIVER <dbl>,
## # MINORINJURIES_DRIVER <dbl>, UNKNOWNINJURIES_DRIVER <dbl>,
## # FATAL_DRIVER <dbl>, MAJORINJURIES_PEDESTRIAN <dbl>,
## # MINORINJURIES_PEDESTRIAN <dbl>, UNKNOWNINJURIES_PEDESTRIAN <dbl>,
## # FATAL_PEDESTRIAN <dbl>, TOTAL_VEHICLES <dbl>, TOTAL_BICYCLES <dbl>,
## # TOTAL_PEDESTRIANS <dbl>, PEDESTRIANSIMPAIRED <dbl>,
## # BICYCLISTSIMPAIRED <dbl>, DRIVERSIMPAIRED <dbl>, TOTAL_TAXIS <dbl>,
## # TOTAL_GOVERNMENT <dbl>, SPEEDING_INVOLVED <dbl>, NEARESTINTRROUTEID <chr>,
## # NEARESTINTSTREETNAME <chr>, OFFINTERSECTION <dbl>,
## # INTAPPROACHDIRECTION <chr>, BLOCKKEY <chr>, SUBBLOCKKEY <chr>,
## # MINORINJURIESPASSENGER <lgl>, INDEX <dbl>
```

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

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

```
# plots the frequency of traffic requests and crashes by date, faceted with wards using bins (it default
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)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# 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
#by_date_with_time <- inner_join(by_date, elapsed_by_date, by = c('NEWDATE', 'WARD'))

## ^^ origin of this code

# joined the two tibbles seeking out matching values in BOTH 'NEWDATE' and 'WARD'.
grouped_by_date <- inner_join(crashes_by_date, requests_by_date, by = c('NEWDATE', 'WARD'))

# looks for NA. If TRUE, sets to zero.
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 groupi
without_ward <- by_date %>%
  group_by(NEWDATE) %>%
  summarize(dailycrashes = sum(dailycrashes), dailyrequests = sum(dailyrequests))

# this just makes sure the rows joined properly.
#greater_than_one <- by_date %>% filter(dailycrashes !=0 & dailyrequests != 0)
#View(greater_than_one)
```

4 Statistical Methods

```
library(moderndive)
library(GGally)

# do we want this?
#ggpairs(data=by_date, columns=c(2,3,4,6), title="Running Models on Each Pair of Variables")
pdf(file= "/Users/eddie/Desktop/Plot.pdf")
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      4.04      0.223     18.1     0      3.60    4.48
## 2 dailycrashes   0.096     0.001    119.     0      0.095   0.098
```

```
get_regression_points(first_model)
```

```
## # A tibble: 2,733 x 5
##   ID dailyrequests dailycrashes dailyrequests_hat residual
##   <int>      <dbl>      <dbl>          <dbl>      <dbl>
## 1     1          1          7          4.71     -3.71
## 2     2          5         37          7.60     -2.60
## 3     3          1          5          4.52     -3.52
## 4     4          2         18          5.77     -3.77
## 5     5          4         25          6.44     -2.44
## 6     6          2         24          6.35     -4.35
## 7     7          2         10          5.00     -3.00
## 8     8          2          6          4.62     -2.62
## 9     9          2         11          5.10     -3.10
## 10    10          2         16          5.58     -3.58
## # ... with 2,723 more rows
```

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



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

```
fit <- lm(dailyrequests ~ dailycrashes, data = without_ward)
without_ward$predicted <- predict(fit) # Save the predicted values
without_ward$residuals <- residuals(fit)
summary(fit)
```

```
##
## Call:
## lm(formula = dailyrequests ~ dailycrashes, data = without_ward)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -49.350  -3.195  -0.848   2.442 117.239
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.0415054  0.2234351   18.09  <2e-16 ***
## dailycrashes  0.0961383  0.0008081  118.97  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.027 on 2731 degrees of freedom
## Multiple R-squared:  0.8383, Adjusted R-squared:  0.8382
## F-statistic: 1.415e+04 on 1 and 2731 DF, p-value: < 2.2e-16
```

```
# Quick look at the actual, predicted, and residual values
without_ward %>% select(dailyrequests, predicted, residuals) %>% head()
```

```
## # A tibble: 6 x 3
##   dailyrequests predicted residuals
##         <dbl>     <dbl>     <dbl>
## 1             1       4.71      -3.71
## 2             5       7.60      -2.60
## 3             1       4.52      -3.52
## 4             2       5.77      -3.77
## 5             4       6.44      -2.44
## 6             2       6.35      -4.35
```

```
ggplot(without_ward, aes(x = dailyrequests, y = dailycrashes)) +
  geom_smooth(method = "lm", se = FALSE, color = "lightgrey") +
  geom_segment(aes(xend = dailyrequests, yend = predicted), alpha = .2) +

  # > Color AND size adjustments made here...
  geom_point(aes(color = abs(residuals), size = abs(residuals))) + # size also mapped
  scale_color_continuous(low = "black", high = "red") +
  guides(color = FALSE, size = FALSE) + # Size legend also removed
  # <

  geom_point(aes(y = predicted), shape = 1) +
  theme_bw()
dev.off()
```

pdf
2

5 Analysis of Results

6 Implications