Homework 6

Ed Huber

12/5/2019

In the following chunks, I will load and clean the datasets that will be used for this project. More information about the data used will follow when the data is prepared.

```
if(!require(data.table)) {
  install.packages('data.table', dependencies = TRUE)

library(data.table)}
```

```
## Loading required package: data.table
```

```
library(tidyverse)
```

```
## -- Attaching packages -------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ----------- tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday,
##     week, yday, year

## The following object is masked from 'package:base':
##
##     date
```

```
library(ggplot2)
theme_set(theme_classic())
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(forcats)
library(xtable)
library(gmodels)
library(utils)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
nyc311 <- fread('311.csv')
names(nyc311) <- names(nyc311) %>%
  stringr::str_replace_all('\\s', '.')
```

```r
ems <- fread('nyc_ems.csv')
names(ems) <- names(ems) %>%
  stringr::str_replace_all('_', '.')
```

```r
ems <- as_tibble(ems)
nyc311 <- as_tibble(nyc311)
```

```r
colnames(ems)
```

```
##  [1] "CAD.INCIDENT.ID"                "INCIDENT.DATETIME"
##  [3] "INITIAL.CALL.TYPE"              "INITIAL.SEVERITY.LEVEL.CODE"
##  [5] "FINAL.CALL.TYPE"                "FINAL.SEVERITY.LEVEL.CODE"
##  [7] "FIRST.ASSIGNMENT.DATETIME"      "VALID.DISPATCH.RSPNS.TIME.INDC"
##  [9] "DISPATCH.RESPONSE.SECONDS.QY"   "FIRST.ACTIVATION.DATETIME"
## [11] "FIRST.ON.SCENE.DATETIME"        "VALID.INCIDENT.RSPNS.TIME.INDC"
## [13] "INCIDENT.RESPONSE.SECONDS.QY"   "INCIDENT.TRAVEL.TM.SECONDS.QY"
## [15] "FIRST.TO.HOSP.DATETIME"         "FIRST.HOSP.ARRIVAL.DATETIME"
## [17] "INCIDENT.CLOSE.DATETIME"        "HELD.INDICATOR"
## [19] "INCIDENT.DISPOSITION.CODE"      "BOROUGH"
## [21] "INCIDENT.DISPATCH.AREA"         "ZIPCODE"
## [23] "POLICEPRECINCT"                 "CITYCOUNCILDISTRICT"
## [25] "COMMUNITYDISTRICT"              "COMMUNITYSCHOOLDISTRICT"
## [27] "CONGRESSIONALDISTRICT"          "REOPEN.INDICATOR"
## [29] "SPECIAL.EVENT.INDICATOR"        "STANDBY.INDICATOR"
## [31] "TRANSFER.INDICATOR"
```

```r
colnames(nyc311)
```

```
##  [1] "Unique.Key"                    "Created.Date"
##  [3] "Closed.Date"                   "Agency"
##  [5] "Agency.Name"                   "Complaint.Type"
##  [7] "Descriptor"                    "Location.Type"
##  [9] "Incident.Zip"                  "Incident.Address"
## [11] "Street.Name"                   "Cross.Street.1"
## [13] "Cross.Street.2"                "Intersection.Street.1"
## [15] "Intersection.Street.2"         "Address.Type"
## [17] "City"                          "Landmark"
## [19] "Facility.Type"                 "Status"
## [21] "Due.Date"                      "Resolution.Action.Updated.Date"
## [23] "Community.Board"               "Borough"
## [25] "X.Coordinate.(State.Plane)"    "Y.Coordinate.(State.Plane)"
## [27] "Park.Facility.Name"            "Park.Borough"
## [29] "School.Name"                   "School.Number"
## [31] "School.Region"                 "School.Code"
## [33] "School.Phone.Number"           "School.Address"
## [35] "School.City"                   "School.State"
## [37] "School.Zip"                    "School.Not.Found"
## [39] "School.or.Citywide.Complaint"  "Vehicle.Type"
## [41] "Taxi.Company.Borough"          "Taxi.Pick.Up.Location"
## [43] "Bridge.Highway.Name"           "Bridge.Highway.Direction"
## [45] "Road.Ramp"                     "Bridge.Highway.Segment"
## [47] "Garage.Lot.Name"               "Ferry.Direction"
```

```
## [49] "Ferry.Terminal.Name"            "Latitude"
## [51] "Longitude"                      "Location"
```

```
ems.2 <- ems %>%
  select(c(20,2,3,4,22))

nyc311.2 <- nyc311 %>%
  select(c(24,2,3,6,7,9, 20))
```

```
colnames(ems.2) <- tolower(colnames(ems.2))

colnames(nyc311.2) <- tolower(colnames(nyc311.2))
```

```
ems.2 <- na.omit(ems.2)
nyc311.2 <- na.omit(nyc311.2)
```

```
ems.2$incident.datetime <-
  lubridate::mdy_hms(ems.2$incident.datetime, tz = Sys.timezone())

nyc311.2$created.date <-
  lubridate::mdy_hms(nyc311.2$created.date, tz = Sys.timezone())
```

```
## Warning: 19 failed to parse.
```

```
ems.2$borough <- tolower(ems.2$borough)
ems.2$initial.call.type <- tolower(ems.2$initial.call.type)
nyc311.2$borough <- tolower(nyc311.2$borough)
nyc311.2$complaint.type <- tolower(nyc311.2$complaint.type)
nyc311.2$descriptor <- tolower(nyc311.2$descriptor)

ems.2$borough <- ems.2$borough %>%
  stringr::str_replace_all('richmond / staten island', 'staten island')

ems.2$zipcode <- gsub('-[[:digit:]]{4}$', '', ems.2$zipcode)
nyc311.2$incident.zip <- gsub('-[[:digit:]]{4}$', '', nyc311.2$incident.zip)

ems.3 <- ems.2 %>%
  subset(format(as.Date(incident.datetime), "%Y") >= 2013 &
  format(as.Date(incident.datetime), "%Y") <= 2015)

nyc311.3 <- nyc311.2 %>%
  subset(format(as.Date(created.date), "%Y") >= 2013 &
  format(as.Date(created.date), "%Y") <= 2015)

nyc311.3 <- nyc311.3 %>%
  dplyr::filter(borough != 'unspecified')

ems.3 <- ems.3 %>%
  dplyr::filter(borough != 'unknown')
```

Now, take a look at our primary data, the collection of 311 complaint calls in New York City's five boroughs.

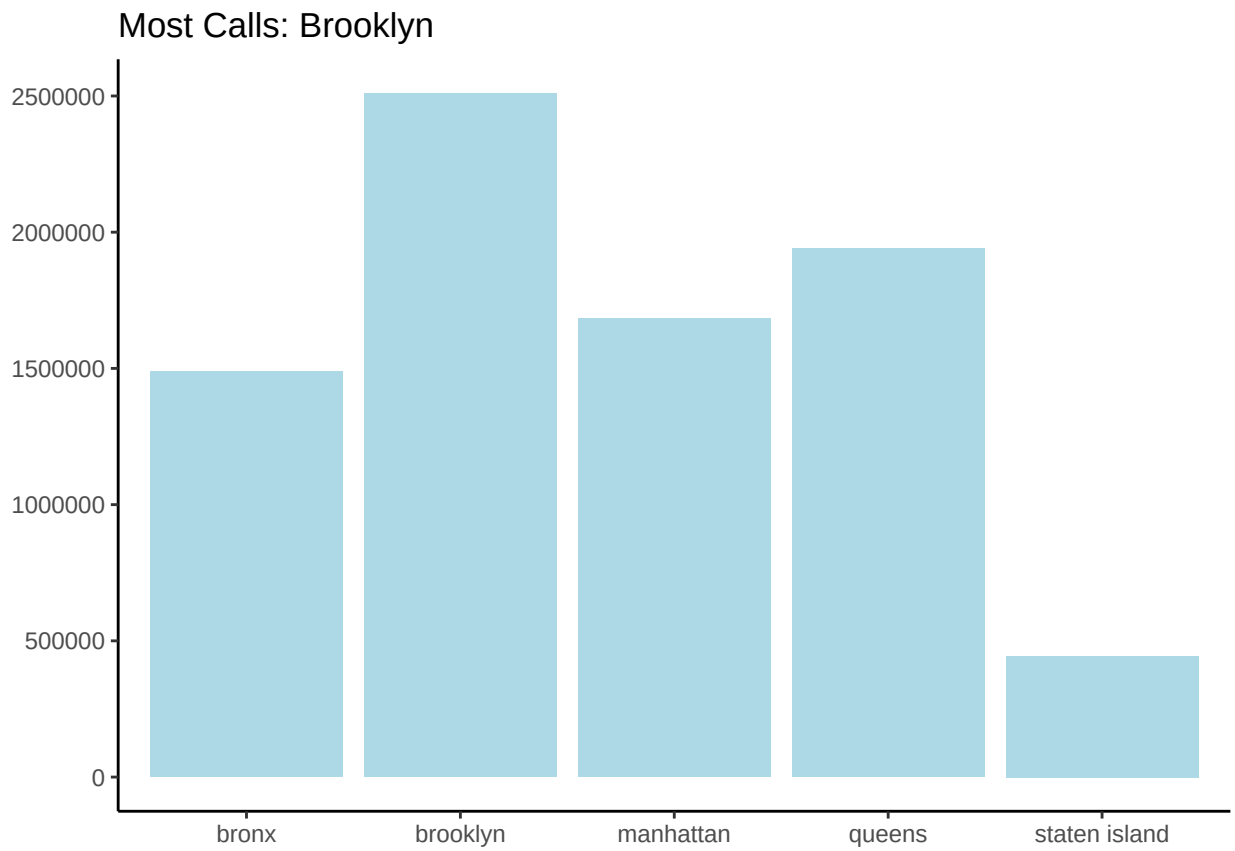
New York City's OpenData began publishing information about daily service requests in 2010 and updating it daily since. Being the massive city that it is, it's 311 line receives thousands of complaints and requests daily. These requests include but are not limited to heating and plumbing issues, illegally parked cars, rat sightings, and noise complaints. The data is provided in a great form for high or low level analysis. We are provided with open and close dates of the request, geo verified location information (latitude/longitude, cross streets, zip code, and borough), the agency (and sub-agency) to which the request was given, and of course the general complaint along with the specifics of the complaint/request. There are an endless number of ways to gain insight from this data, and for this project I will look at a few of them.

How many calls has each borough received in total?

```
total.borough <- ggplot(data = nyc311.2 %>% dplyr::filter(borough != 'unspecified'), aes(borough)) +
  geom_histogram(stat = 'count', fill = 'light blue') + ggtitle("Most Calls: Brooklyn") +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank())
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```

```
total.borough
```

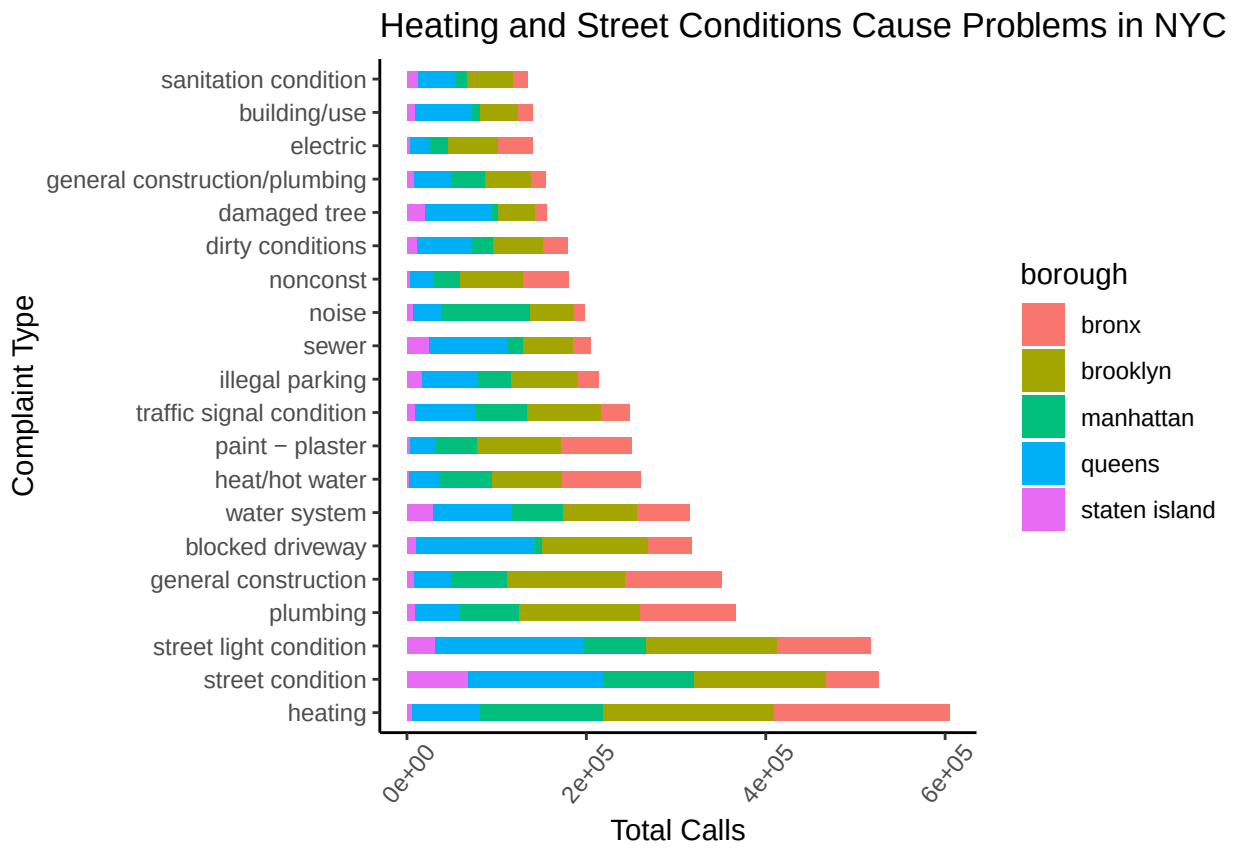


Per the plot above, Brooklyn received the most calls over the period, followed by (in order), queens, manhattan, bronx, and staten island - which receives a significantly lower aount of calls than the other four boroughs.

What are the top 20 most occuring complaints in this data and how are they distributed amongst boroughs?

As there are 226 complaint types listed in this data, I will filter this to the top 20 most occuring.

```
complaints <- ggplot(data = subset(nyc311.2, complaint.type %in%
              count(nyc311.2, complaint.type, sort = TRUE)[1:20,]$complaint.type) %>%
              dplyr::filter(borough != 'unspecified'),
              aes(complaint.type)) +

  geom_bar(aes(x = forcats::fct_infreq(complaint.type), fill = borough), width = 0.5) +
  theme(axis.text.x = element_text(angle = 50, vjust = 0.6)) +
  xlab('Complaint Type') + ylab('Total Calls') + coord_flip() +
  ggtitle('Heating and Street Conditions Cause Problems in NYC')

complaints
```



It looks like the most common comnplaints among all boroughs are heating, street/street light conditions, plumbing, and construction complaints. The total amount per descriptor per borough is not entirely clear here, so let's create a table of the relationship between the boroughs and the top three occuring complaints.

What percentage of the top complaints does each borough account for?

```
# First changing the complaint names to a shorter version to make sure the
# width of the table fits on the page
nyc311.2$complaint.type <- nyc311.2$complaint.type %>%
  stringr::str_replace_all('street condition', 'st cond') %>%
  stringr::str_replace_all('street light condition', 'st lt con')

chooseCRANmirror(graphics=FALSE, ind=1)
xtabA <-dplyr::filter(nyc311.2,
    complaint.type == 'heating' | complaint.type == 'st lt con' | complaint.type == 'st cond')
xtabB <-select(xtabA, borough, "complaint.type")
CrossTable(xtabA$borough, xtabB$'complaint.type')
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1938971
##
##
##                | xtabB$complaint.type
## xtabA$borough |   heating |   st cond | st lt con | Row Total |
## --------------|-----------|-----------|-----------|-----------|
##         bronx |    195246 |     58515 |    103931 |    357692 |
##               |  6056.319 | 15384.021 |   531.850 |           |
##               |     0.546 |     0.164 |     0.291 |     0.184 |
##               |     0.220 |     0.111 |     0.198 |           |
##               |     0.101 |     0.030 |     0.054 |           |
## --------------|-----------|-----------|-----------|-----------|
##      brooklyn |    190268 |    147546 |    145881 |    483695 |
##               |  4387.827 |  1980.294 |  1728.627 |           |
##               |     0.393 |     0.305 |     0.302 |     0.249 |
##               |     0.214 |     0.280 |     0.278 |           |
##               |     0.098 |     0.076 |     0.075 |           |
## --------------|-----------|-----------|-----------|-----------|
##     manhattan |    137458 |    101264 |     69511 |    308233 |
##               |    94.582 |  3665.720 |  2306.444 |           |
##               |     0.446 |     0.329 |     0.226 |     0.159 |
##               |     0.155 |     0.192 |     0.133 |           |
##               |     0.071 |     0.052 |     0.036 |           |
## --------------|-----------|-----------|-----------|-----------|
##        queens |     75776 |    150514 |    165684 |    391974 |
##               | 59894.586 | 18195.359 | 33560.963 |           |
##               |     0.193 |     0.384 |     0.423 |     0.202 |
```

```
##                |     0.085 |     0.286 |     0.316 |           |
##                |     0.039 |     0.078 |     0.085 |           |
## --------------|-----------|-----------|-----------|-----------|
## staten island |      6011 |     68456 |     31393 |    105860 |
##                | 37187.030 | 54786.297 |   265.509 |           |
##                |     0.057 |     0.647 |     0.297 |     0.055 |
##                |     0.007 |     0.130 |     0.060 |           |
##                |     0.003 |     0.035 |     0.016 |           |
## --------------|-----------|-----------|-----------|-----------|
##   unspecified |    282916 |       500 |      8101 |    291517 |
##                | 167374.114 | 78204.804 | 63486.974 |           |
##                |     0.970 |     0.002 |     0.028 |     0.150 |
##                |     0.319 |     0.001 |     0.015 |           |
##                |     0.146 |     0.000 |     0.004 |           |
## --------------|-----------|-----------|-----------|-----------|
##  Column Total |    887675 |    526795 |    524501 |   1938971 |
##                |     0.458 |     0.272 |     0.271 |           |
## --------------|-----------|-----------|-----------|-----------|
##
##
```

What can we take away from this table? First, notice that the amount of data in this set whose borough is labelled as unspecified accounts for much of the occurrence of the top complaints at 21.6% - this is more than enough data to give us inaccurate results, so for this table it was important to keep it in the data. Interestingly, most of this is seen in the heating compalint column, in which unspecified boroughs with heating issues accounts for 14.6% of our filtered data, and 97% of the data labelled as borough unspecified. With that in mind, and moving along, Brooklyn did come in highest of accurately labelled data with 24.9% of the top 3 complaints. Although that is true, Bronx has more heating complaints than Brooklyn, and Queens has more complaints about street light conditions and general street conditions. We do see consistency in that calls in Staten Island accounts for only 5.5% of our total data. The key takeaway from this is that although Brooklyn shows the most calls overall, the disparity between it and the other boroughs varies when looking at specific complaints as opposed to the entire aggregate.

Let's now dig a little deeper into specifics and take a look at the types of street condition complaints that are found. I would generally look into the number one complaint (heating), but taking a glance at the data shows only three unique types of heating complaints whereas street condition contains 34, so perhaps there is more insight there.

There are 34 unique types, how are these distributed across the data?

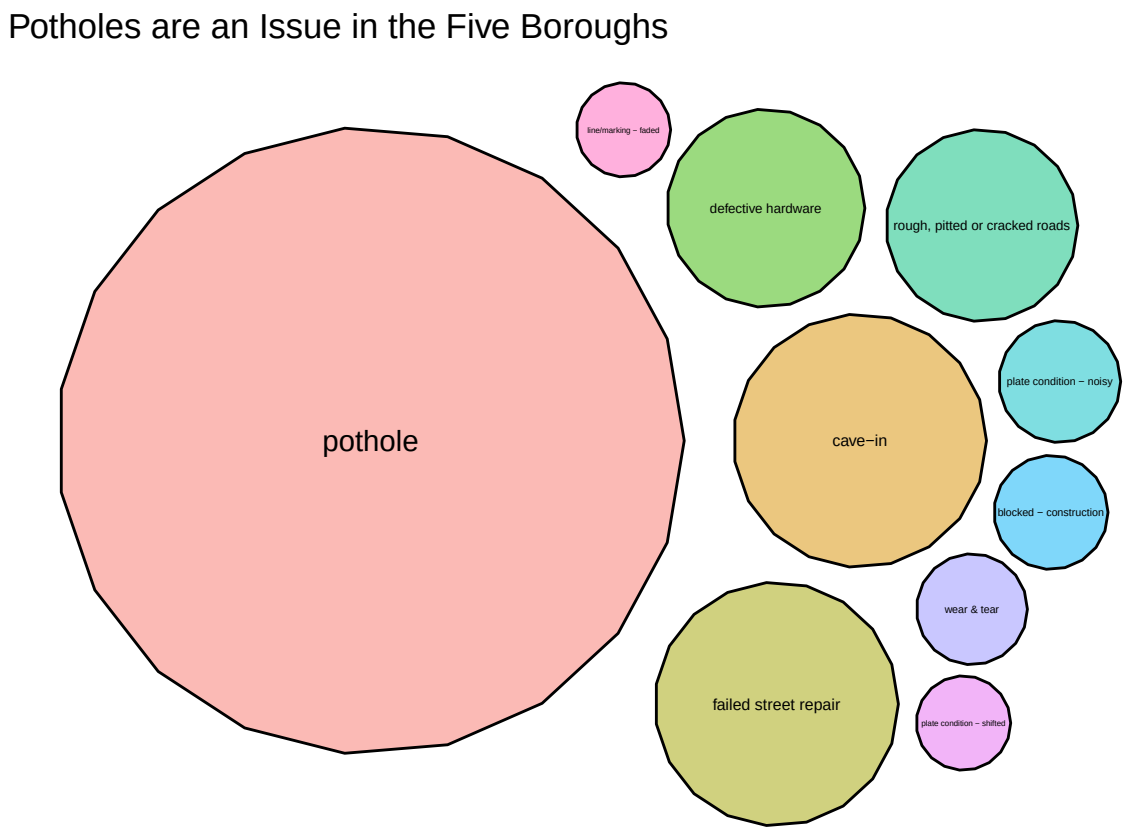
```
library(packcircles)
library(viridis)
```

```
## Loading required package: viridisLite
```

```
##
## Attaching package: 'viridis'
```

```
## The following object is masked from 'package:scales':
##
##     viridis_pal
```

```
street_cond <- nyc311.2 %>% filter(complaint.type == 'st cond') %>%
               group_by(descriptor) %>% tally(sort=TRUE, name="total")

street_cond <- street_cond[1:10,]

packing <- circleProgressiveLayout(street_cond$total, sizetype = 'area')
packing$radius <- 0.9 * packing$radius
street_cond <- cbind(street_cond, packing)
dat.gg <- circleLayoutVertices(packing, npoints = 19)
dat.gg$value <- (rep(street_cond$total, each = 20))

ggplot() +
  geom_polygon(data = dat.gg, aes(x, y, group = id,
  fill = as.factor(id)), colour = 'black', alpha = 0.5) +
  scale_color_viridis_d(option = 'plasma') +
  geom_text(data = street_cond, aes(x, y, size=total, label = descriptor),
  colour = 'black') +
  scale_size_continuous(range = c(1,4)) +

  # General theme:
  theme_void() +
  theme(legend.position="none") +
  coord_equal() + ggtitle('Potholes are an Issue in the Five Boroughs')
```

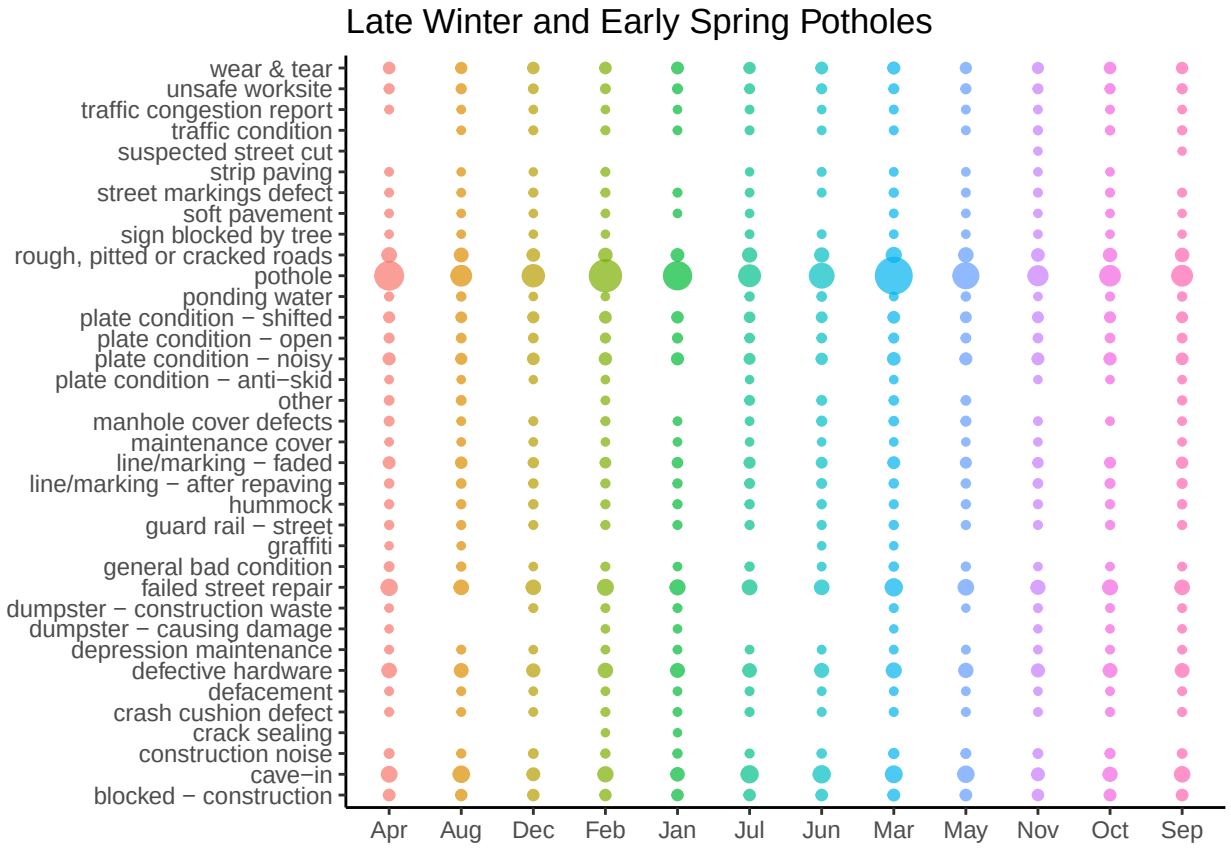


This is quite telling. Clearly, the most common type of street condition complained about is potholes, and from living in a city, I see this as no surprise. This is followed by a street cave-in, and then defective

hardware, rough/pitted/cracked road, and failed street repair - all with very similar totals. Let's take the look into street conditions one step further:

During what time of year do street condition complaints occur the most in NYC?

```
by_month <- ggplot(data = na.omit(nyc311.2) %>%
       filter(complaint.type == 'st cond'),
       aes(x = as.factor(month.abb[month(created.date)]), y = descriptor,
           colour = month.abb[month(created.date)])) +
       geom_count(alpha = 0.7) + ggtitle('Late Winter and Early Spring Potholes') +
       theme(axis.title.x = element_blank()) +
       theme(axis.title.y = element_blank()) +
       theme(legend.position = 'none')

by_month
```



Again, we see the fact that potholes account for most of the street condition data and heavily compared to the other 33 descriptions. It looks like, as no surprise, that potholes are most called about during the January - April time period.

What are the totals of calls received about street conditions?

```
total_cond <- nyc311.2 %>%
  filter(complaint.type == 'st cond') %>%
  group_by(Description = descriptor) %>%
  summarize(Total = length(Description)) %>%
  arrange(desc(Total))

knitr::kable(total_cond)
```

Description	Total
pothole	310017
cave-in	50584
failed street repair	46804
defective hardware	30939
rough, pitted or cracked roads	29046
plate condition - noisy	11715
blocked - construction	10258
wear & tear	9700
plate condition - shifted	7031
line/marking - faded	6996
unsafe worksite	2726
plate condition - open	2006
construction noise	1714
line/marking - after repaving	1459
other	953
manhole cover defects	877
hummock	679
ponding water	530
guard rail - street	518
general bad condition	505
street markings defect	305
crash cushion defect	294
traffic condition	285
dumpster - construction waste	167
depression maintenance	166
defacement	128
traffic congestion report	106
strip paving	92
sign blocked by tree	58
soft pavement	53
maintenance cover	40
plate condition - anti-skid	20
dumpster - causing damage	12
graffiti	8
crack sealing	2
suspected street cut	2

Now that we have a good idea of how the data is distributed across the boroughs and of what types of specifics it tells us, let's compare it to some different data.

Introducing the second dataset:

This data is found at https://data.cityofnewyork.us/resource/76xm-jjuj.csv and is a file of data generated by NYC's EMS computer aided dispatch system. This link can be substituted in the next chunk to download the data directly from the website. The data is very similar to the 311 data as the information included in the full file ranges from the creation of the incident to the closing of the incident and includes information about what type of incident it is, which city resources were used, and if the Fire Dept. responded to the emergency. It does not include such specific locations as the 311 data, but it is easily linkable by Borough or zip code, as that information is given. The information in the file regards emergency medical services calls throughout the five boroughs. Let's take a preliminary look at this data.

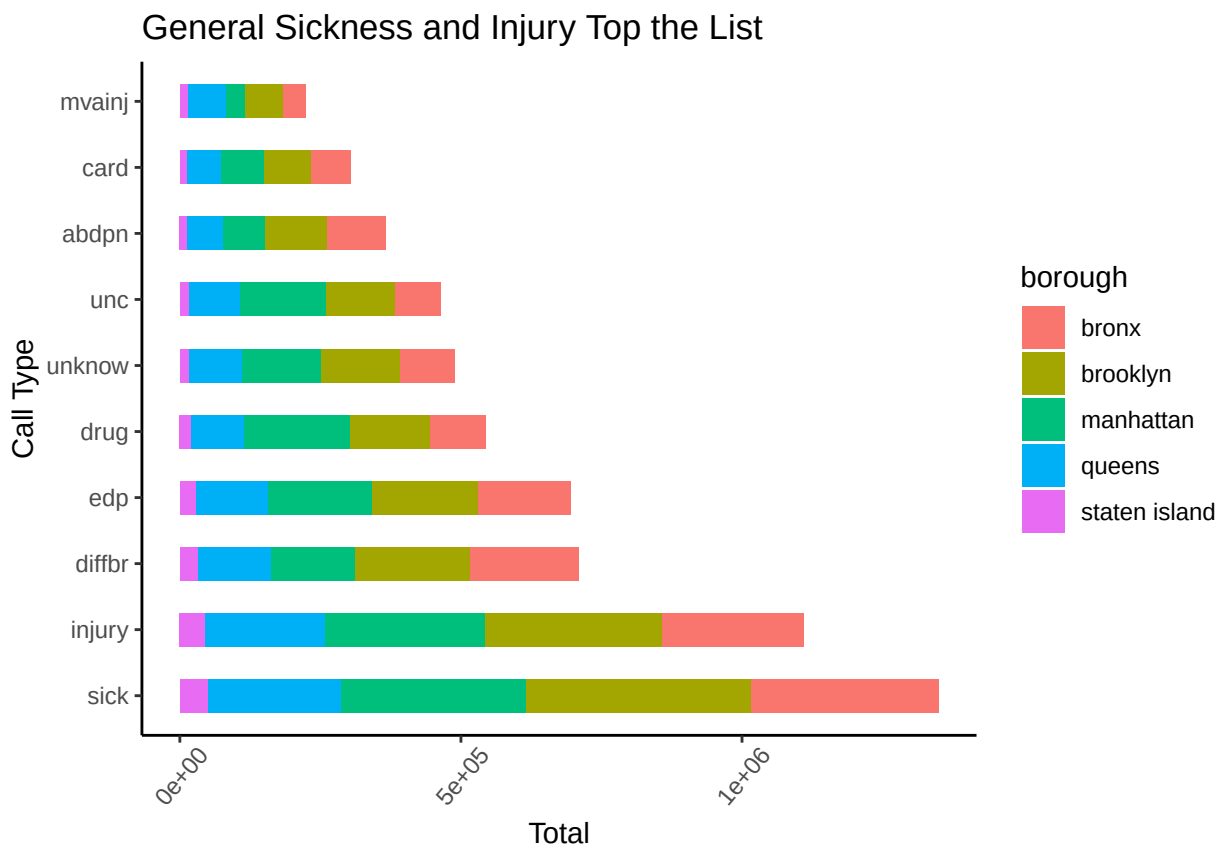
What were the top 10 emergency medical services calls received?

```
top_10 <- ems.2 %>%
  count(Call = initial.call.type,
        Severity = initial.severity.level.code, sort = TRUE, name = 'Total')

kable(top_10[1:10,])
```

Call	Severity	Total
sick	6	1332660
injury	5	1069355
diffbr	2	708971
edp	7	695037
drug	4	543584
unknow	4	488309
unc	2	464021
abdpn	5	363093
card	3	303260
mvainj	4	224313

How are these calls distributed throughout the five boroughs?

```
ems_calls <- ggplot(data = subset(ems.2, initial.call.type %in%
    count(ems.2, initial.call.type, sort = TRUE)[1:10,]$initial.call.type) %>%
    dplyr::filter(borough != 'unknown'),
    aes(initial.call.type)) +

  geom_bar(aes(x = forcats::fct_infreq(initial.call.type),
          fill = borough), width = 0.5) +
  theme(axis.text.x = element_text(angle = 50, vjust = 0.6)) +
  coord_flip() + ylab('Total') + xlab('Call Type') +
  ggtitle('General Sickness and Injury Top the List')


ems_calls
```



It looks like general sick calls total a few hundred thousand more than the second most called about emergency, injuury. Glancing at this plot tells us that Brooklyn receives the most calls, although it is not apparent by how much.

Let's view the totals in a crosstable:

```
chooseCRANmirror(graphics=FALSE, ind=1)
xtabC <-dplyr::filter(ems.2,
    initial.call.type == 'sick' | initial.call.type == 'injury' |
    initial.call.type == 'diffbr')
xtabD <-select(xtabC, borough, "initial.call.type")
CrossTable(xtabC$borough, xtabD$'initial.call.type')
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
```

```
##
## Total Observations in Table:  3168617
##
##
##                | xtabD$initial.call.type
## xtabC$borough |    diffbr |    injury |      sick | Row Total |
## --------------|-----------|-----------|-----------|-----------|
##         bronx |    192221 |    252189 |    334124 |    778534 |
##               |  1846.321 |  1534.261 |    19.062 |           |
##               |     0.247 |     0.324 |     0.429 |     0.246 |
##               |     0.271 |     0.227 |     0.248 |           |
##               |     0.061 |     0.080 |     0.105 |           |
## --------------|-----------|-----------|-----------|-----------|
##      brooklyn |    205250 |    313289 |    398904 |    917443 |
##               |     0.081 |   199.108 |   169.021 |           |
##               |     0.224 |     0.341 |     0.435 |     0.290 |
##               |     0.289 |     0.282 |     0.296 |           |
##               |     0.065 |     0.099 |     0.126 |           |
## --------------|-----------|-----------|-----------|-----------|
##     manhattan |    149639 |    285123 |    330043 |    764805 |
##               |  2717.607 |  1116.081 |    56.257 |           |
##               |     0.196 |     0.373 |     0.432 |     0.241 |
##               |     0.211 |     0.257 |     0.245 |           |
##               |     0.047 |     0.090 |     0.104 |           |
## --------------|-----------|-----------|-----------|-----------|
##        queens |    128543 |    214134 |    235987 |    578664 |
##               |     7.670 |   651.105 |   446.446 |           |
##               |     0.222 |     0.370 |     0.408 |     0.183 |
##               |     0.181 |     0.193 |     0.175 |           |
##               |     0.041 |     0.068 |     0.074 |           |
## --------------|-----------|-----------|-----------|-----------|
## staten island |     33674 |     44910 |     50586 |    129170 |
##               |   782.903 |     2.337 |   357.152 |           |
##               |     0.261 |     0.348 |     0.392 |     0.041 |
##               |     0.047 |     0.040 |     0.037 |           |
##               |     0.011 |     0.014 |     0.016 |           |
## --------------|-----------|-----------|-----------|-----------|
##       unknown |         0 |         0 |         1 |         1 |
##               |     0.224 |     0.350 |     0.774 |           |
##               |     0.000 |     0.000 |     1.000 |     0.000 |
##               |     0.000 |     0.000 |     0.000 |           |
##               |     0.000 |     0.000 |     0.000 |           |
## --------------|-----------|-----------|-----------|-----------|
##  Column Total |    709327 |   1109645 |   1349645 |   3168617 |
##               |     0.224 |     0.350 |     0.426 |           |
## --------------|-----------|-----------|-----------|-----------|
##
##
```

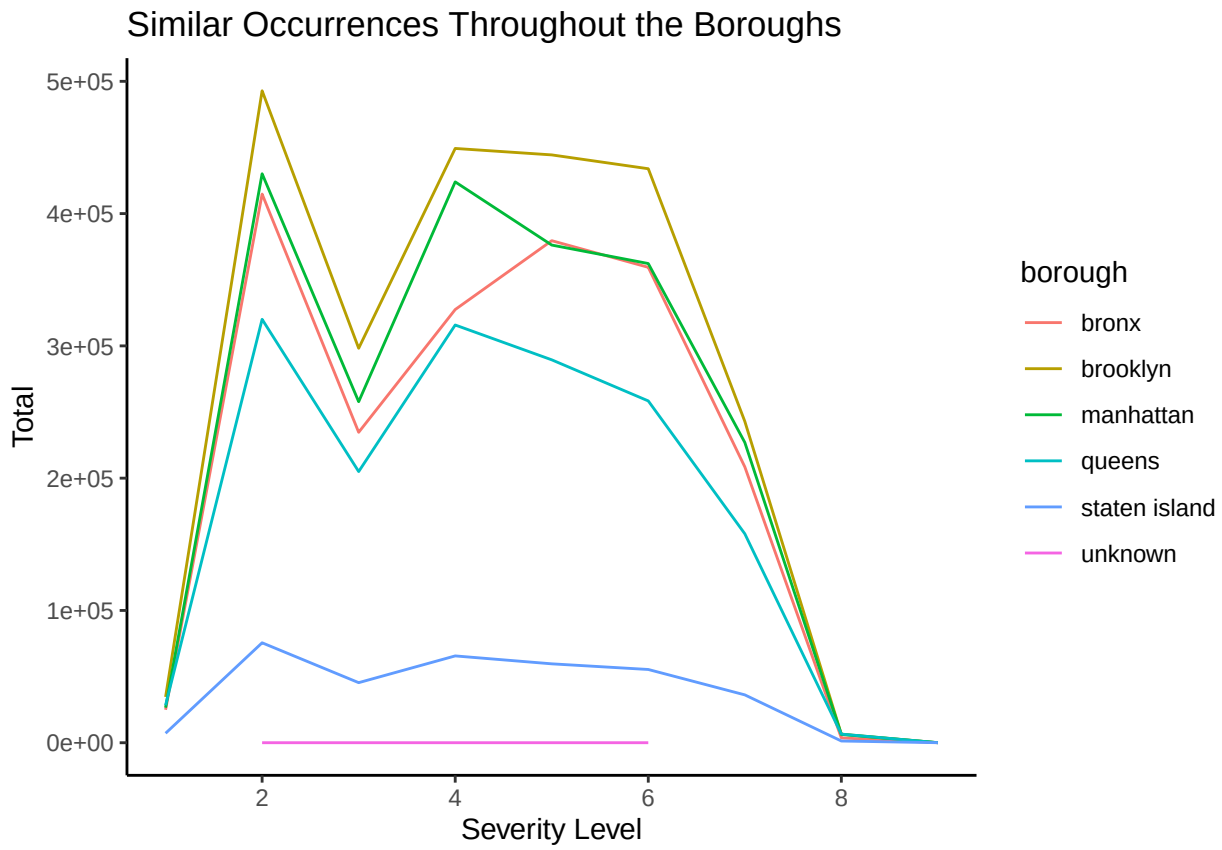
Here I have created the table to show the distribution of the top three ems calls: sick, injury, and diffbr. We see that calls in Brooklyn account for 29% of the total data, followed by Bronx at 24.6%, Manhattan at 24.1%, Queens at 18.3% and Staten Island with a very low 4% of the data. For the top complaint, sick calls, 29.6% of those came from Brooklyn, followed by 24.8% in Bronx. Regarding the injury calls, 28.2% came from Brooklyn followed 25.7% from Manhattan. Finally, for diffbr calls, 28.9% came from Brooklyn

and 27.1% from the Bronx. There is only one report from an unkown borough, so we can be sure that the data is kept with attention to detail.

How severe are EMS calls across the boroughs?

The data includes a severity level column ranging from 1 (least) to 9 (most).

```
ems.2 <- ems.2 %>%
  filter(borough != 'unkown')

severity <- ggplot(data = ems.2 %>%
  count(borough, Severity.Level = initial.severity.level.code, name = 'Total'),
  aes(x = Severity.Level, y = Total, color = borough)) +
  geom_line() + xlab('Severity Level') + scale_x_continuous(breaks = pretty_breaks()) +
  ggtitle('Similar Occurrences Throughout the Boroughs')

severity
```



Although the totals are different for each borough, a general pattern is seen for all of them. There are very few calls labelled with minimal severity (1) and maximum severity (8-9). For Brooklyn, Manhattan, Queens, and Staten Island, the two most occuring are labelled with a severity level of 2 or 4, while in the Bronx, they are labelled as 2 or 5.

What is the average severity level between 2013-2015?

```
mean(ems.3$initial.severity.level.code)
```

```
## [1] 4.298792
```

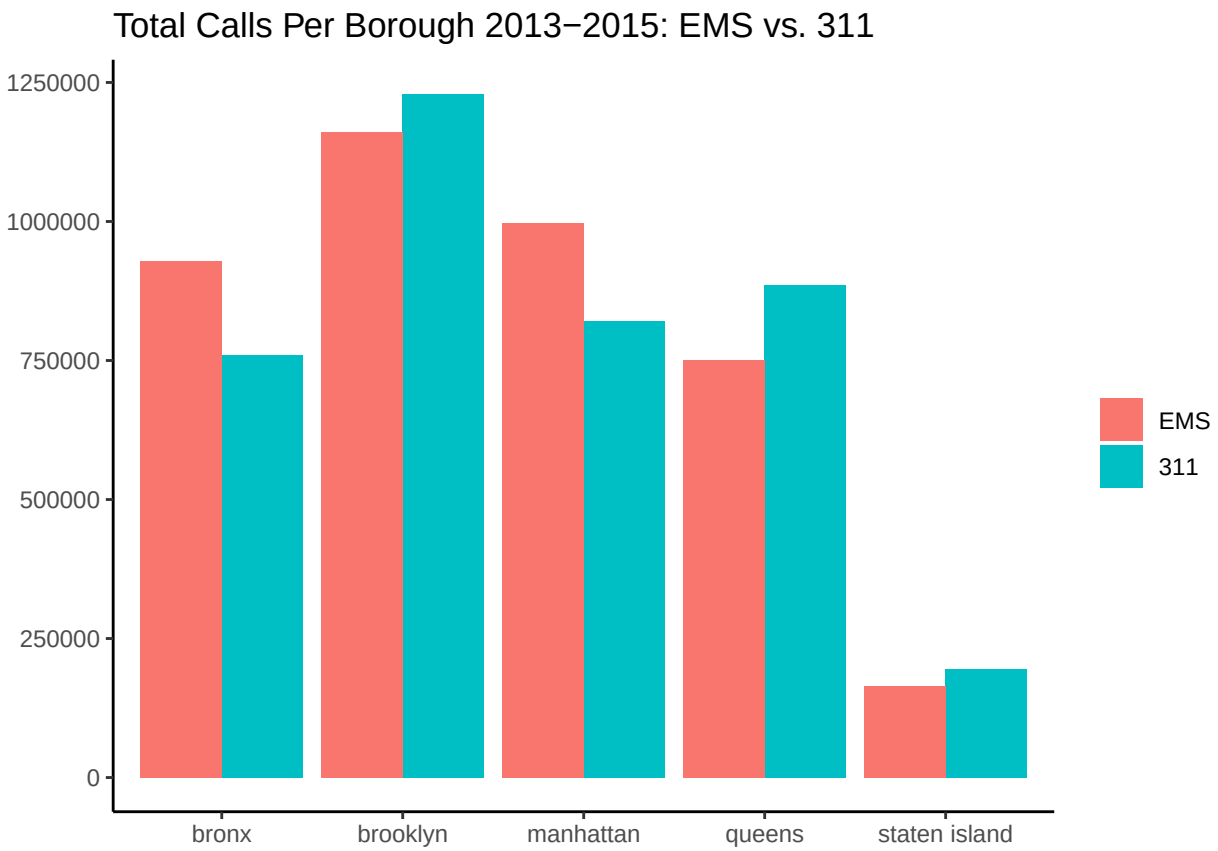
It appears that on average, emergency medical services deals with cases that are not severe enough to be life or death, but that do require fast attention.

Now that we have a general idea of what both datasets are capable of telling, let's compare:

For the remainder of this analysis, I will be aggregating and looking for trends on the hour, day, and month between the EMS and 311 data to see if we can gain any insight as to whether or not there is any sort of correlation between occurrences for which a 311 complaint was filed and emergency medical incidents. In order to match the two data sets accurately, I have filtered each one to include only occurences in 2013, 2014, and 2015.

First, let's take a look at the total calls by Borough for both sets side by side.

```
b1 <- ems.3 %>%
  count(borough, name = 'total.ems')

b2 <- nyc311.3 %>%
  count(borough, name = 'total.311')

b3 <- full_join(b1, b2, by = 'borough')

b4 <- melt(b3, id.vars = 'borough')

total_by_borough <- ggplot(b4, aes(x = borough, y = value, fill = variable)) +
  geom_bar(stat = 'identity', position = 'dodge') +
  theme(axis.title.x = element_blank(), axis.title.y = element_blank()) +
  ggtitle('Total Calls Per Borough 2013-2015: EMS vs. 311') +
  theme(legend.title = element_blank()) +
  scale_fill_discrete(labels = c('EMS', '311'))

total_by_borough
```

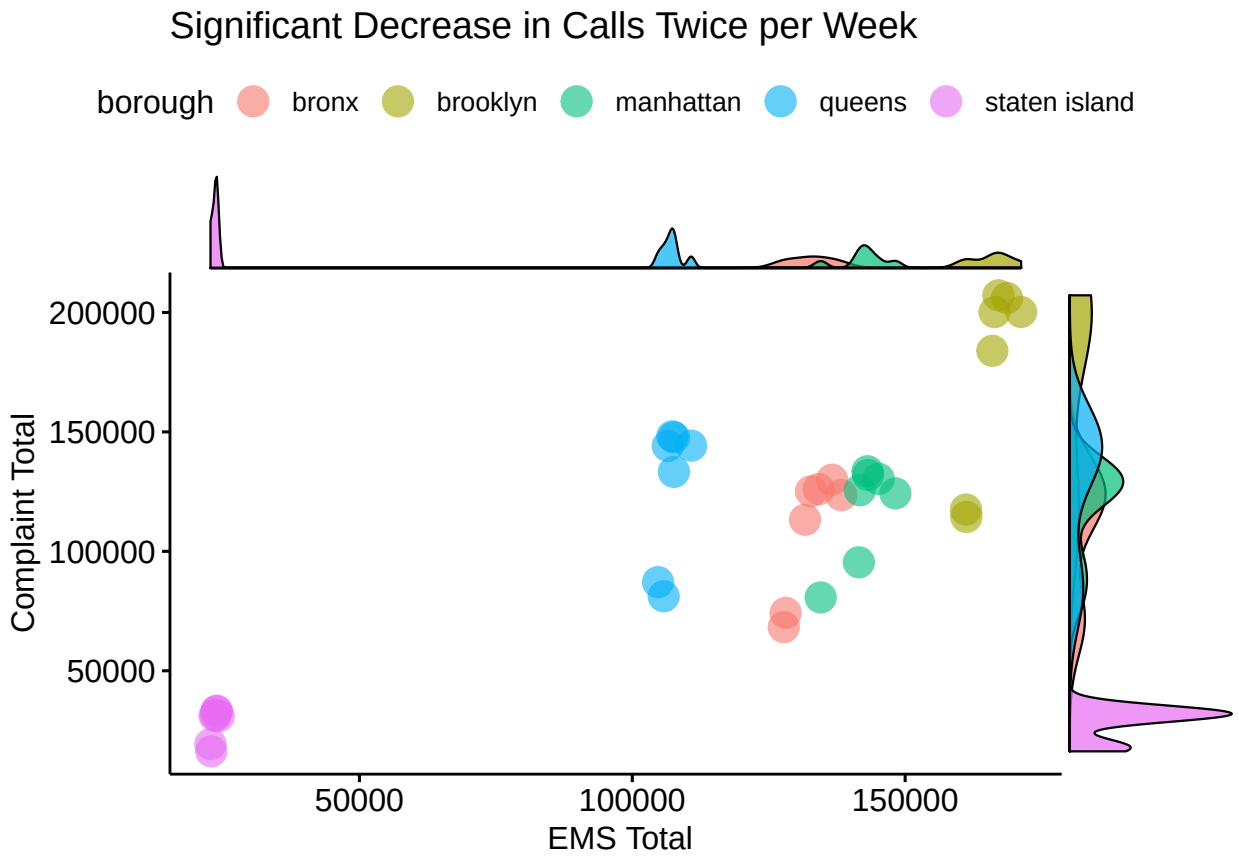


Here we see some difference in aggregates for both sets. The EMS data shows that EMS calls are highest in brooklyn, but then in order Manhattan, Bronx, Queens, and Staten Island.

What does comparing totals by day show about the relationship between EMS calls and complaint calls?

```
ems.3$day <- wday(as.Date(ems.3$incident.datetime))

nyc311.3$day <- wday(as.Date(nyc311.3$created.date))

total.ems.day <- ems.3 %>%
  group_by(borough, day) %>%
  tally(name = 'EMS.Total')

total.311.day <- nyc311.3 %>%
  group_by(borough, day) %>%
  tally(name = 'Complaint.Total')

daily <- full_join(total.ems.day, total.311.day,
          by = c('borough', 'day'))

daily_plot <- ggscatterhist(daily, x = 'EMS.Total', y = 'Complaint.Total',
    color = 'borough', size = 5, alpha = 0.6,
    margin.params = list(fill = 'borough', color = 'black', size = 0.4),
```

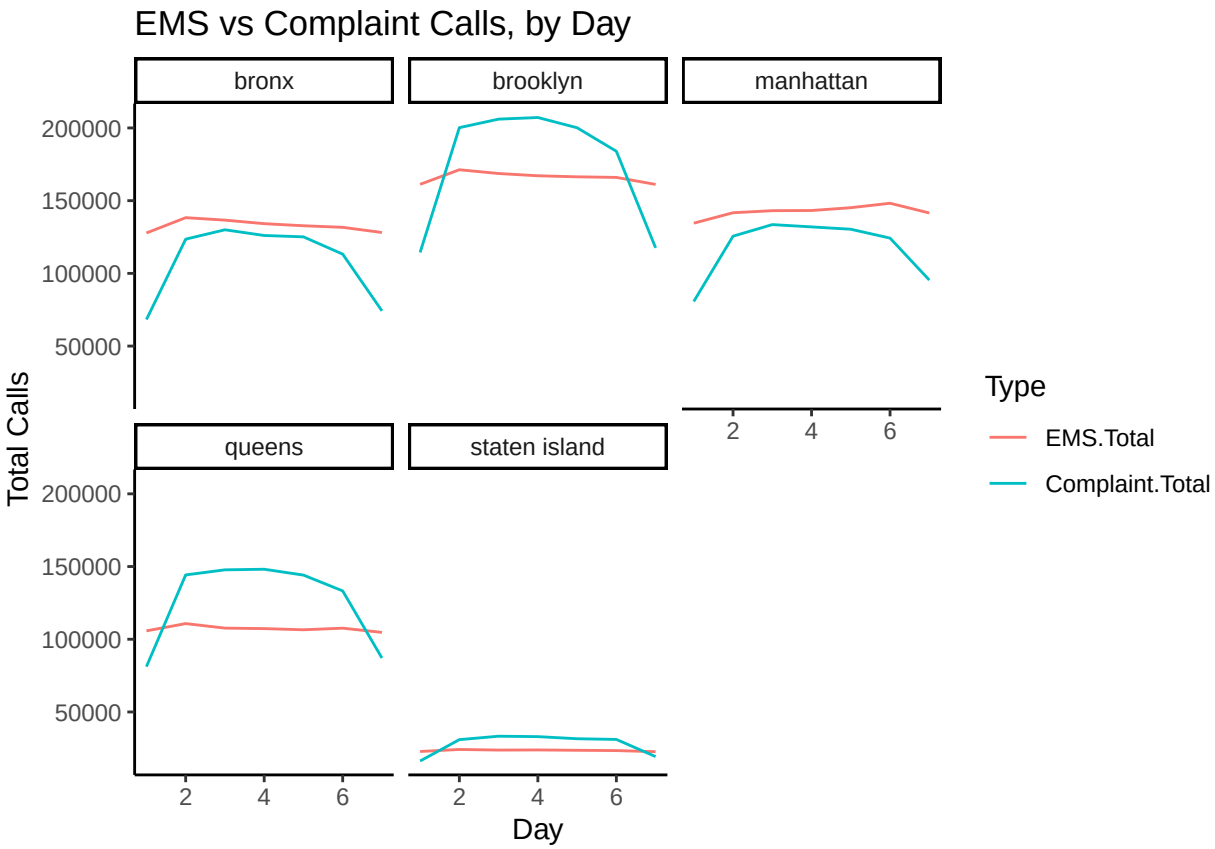
```
    xlab = ' EMS Total', ylab = 'Complaint Total',
    title = 'Significant Decrease in Calls Twice per Week')

daily_plot
```



From this plot, we can see that daily trends are somewhat consistent across all boroughs. Interestingly, it shows that for each borough there are two days during the week that complaint calls occur significantly less than the mean and EMS calls occur slightly less than the mean.

Let's view this data in a line plot and see what it tells us

```
daily2 <- melt(daily, id = c('day', 'borough'), variable.name = 'Type',
               value.name = 'total')

ggplot() + theme_set(theme_gray()) +
  geom_line(data = daily2, aes(x = day, y = total, color = Type)) +
  facet_wrap(~borough) + ggtitle("EMS vs Complaint Calls, by Day") +
  xlab('Day') + ylab('Total Calls')
```

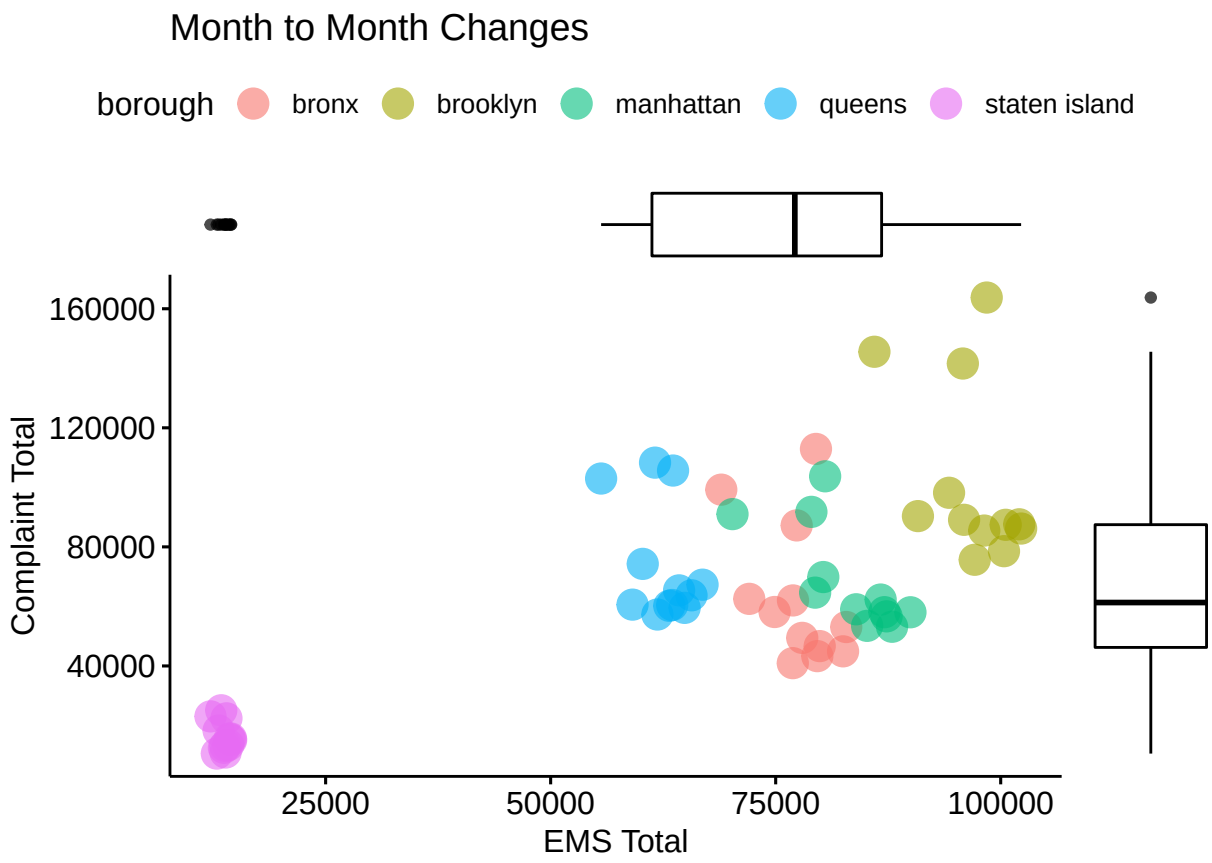


This is interesting. Looking at these totals day by day, we see a very consistent daily total for EMS calls, but some variability for complaint calls. As was expected, we see that there are two days per week where the total falls drastically for complaint calls. Curiously, this is always on Saturday and Sunday. Another interesting observation here is that Brookly and Queens see a large spike in complaint calls on Monday -Friday. Overall, there seems to be no real correlation between the data, given the generally flat pattern of EMS calls.

Will a monthly view give us a different view of the relationship?

```
total.ems.month <- ems.3 %>%
  group_by(borough, month(incident.datetime)) %>%
  tally(name = 'EMS.Total')

total.311.month <- nyc311.3 %>%
  group_by(borough, month(created.date)) %>%
  tally(name = 'Complaint.Total')

monthly <- full_join(total.ems.month, total.311.month,
          by = c('borough', 'month(incident.datetime)' = 'month(created.date)'))

setnames(monthly, 'month(incident.datetime)', 'month')

ggscatterhist(monthly, x = 'EMS.Total', y = 'Complaint.Total',
              color = 'borough', size = 5, alpha = 0.6,
```

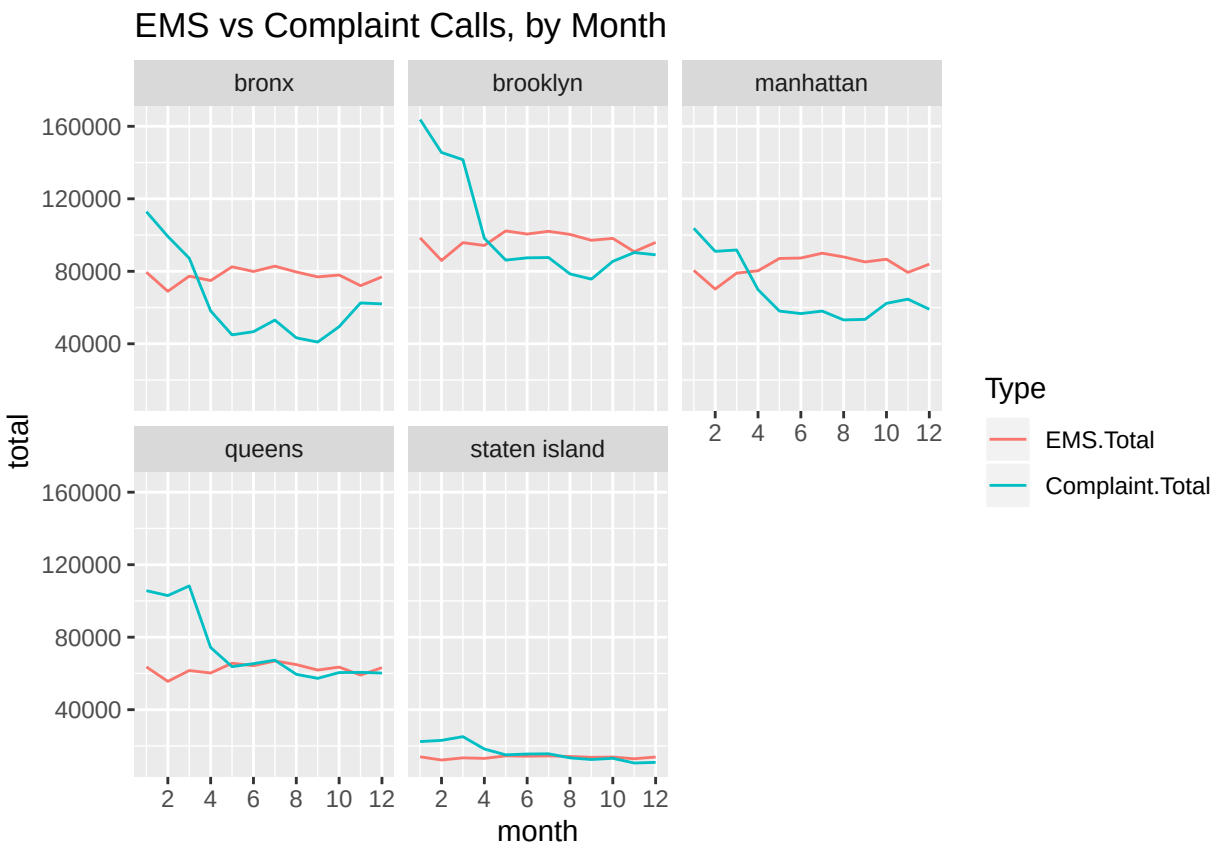
```
            margin.plot  = 'boxplot', xlab = 'EMS Total',
            ylab = 'Complaint Total', title = 'Month to Month Changes')
```



This looks a bit different from the daily view - there is much more variance in each cluster. We see a median of about 80,000 monthly EMS calls, and about 60,000 complaint calls. The curious takeaway from this plot is that instead of seeing outliers with lower values, there are outliers with higher values in the Complaint side, but lower/stagnant values on the EMS side.

What will the line plot tell us?

```
monthly2 <- melt(monthly, id = c('month', 'borough'), variable.name = 'Type',
                 value.name = 'total')

ggplot() +
  geom_line(data = monthly2, aes(x = month, y = total, color = Type)) +
  facet_wrap(~borough) + ggtitle("EMS vs Complaint Calls, by Month") +
  theme_set(theme_gray()) + scale_x_continuous(breaks = pretty_breaks())
```

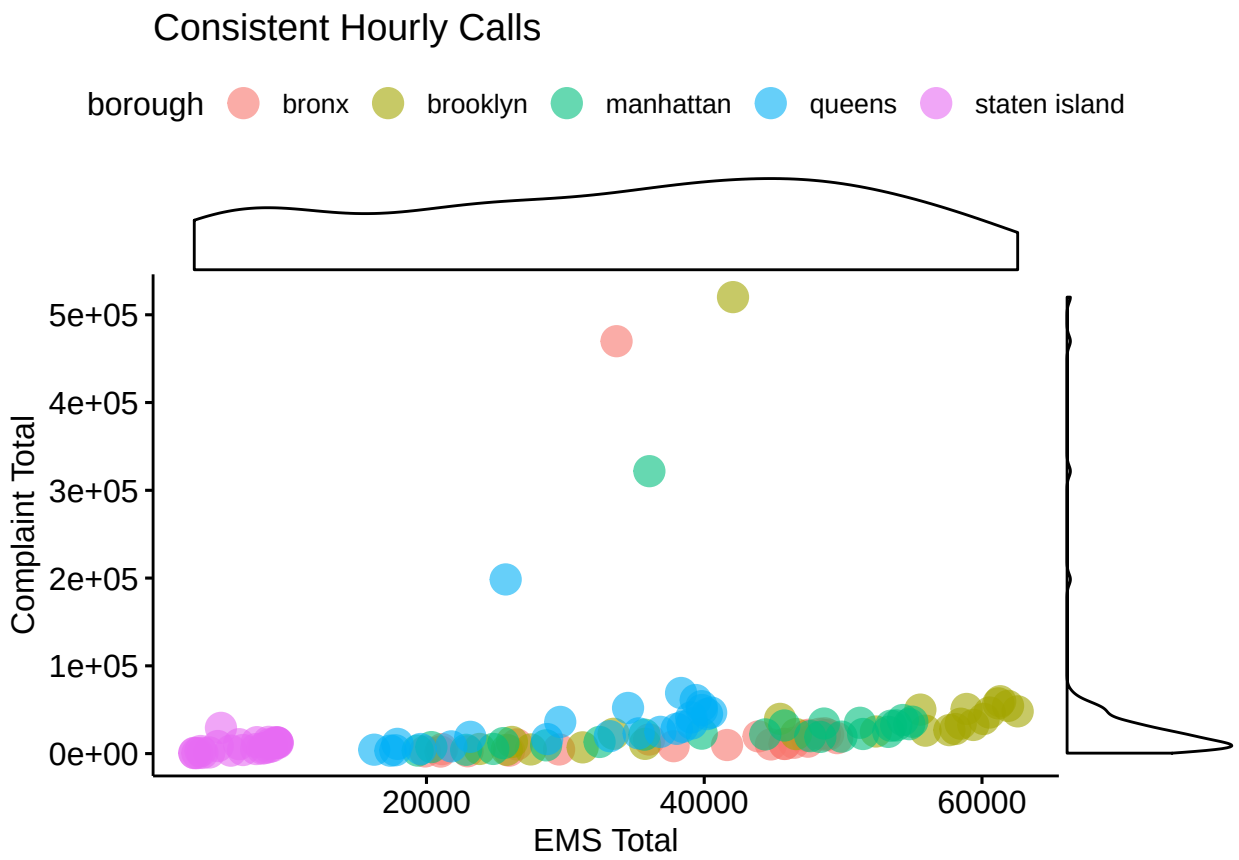


Again we see general consistency in EMS data month to month, but some variability in the complaint data, with differing levels in each borough. For all boroughs we see the highest amount of complaints come in the winter to spring months, with a bottoming out in September. Although we see this variability in the 311 data, the consisten levels in the EMS data tell us that there is basically no correlation between the two sets.

One more try, will an hourly analysis show any relationship?

```
total.ems.hour <- ems.3 %>%
  group_by(borough, hour(incident.datetime)) %>%
  tally(name = 'EMS.Total')

total.311.hour <- nyc311.3 %>%
  group_by(borough, hour(created.date)) %>%
  tally(name = 'Complaint.Total')

hourly <- full_join(total.ems.hour, total.311.hour,
          by = c('borough', 'hour(incident.datetime)' = 'hour(created.date)'))

setnames(hourly, 'hour(incident.datetime)', 'hour')

hourly.plot <- ggscatterhist(hourly, x = 'EMS.Total', y = 'Complaint.Total',
               color = 'borough', size = 5, alpha = 0.6,
               xlab = 'EMS Total', ylab = 'Complaint Total',
               title = 'Consistent Hourly Calls')
```

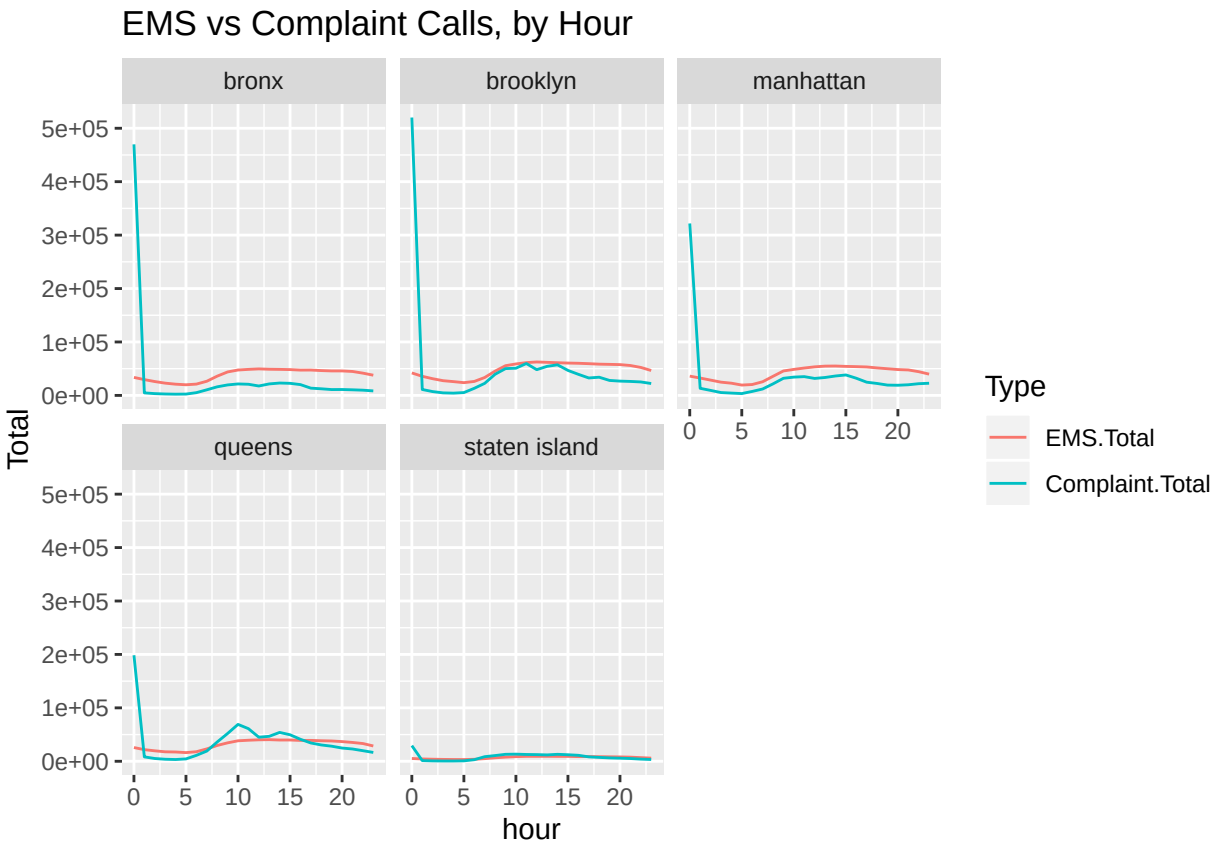
```
hourly.plot
```



From this plot, it looks like complaint calls follow a very consisten total while EMS calls show a lot of variance hour to hour.

What will the line plot show?

```
hourly2 <- melt(hourly, id = c('hour', 'borough'), variable.name = 'Type',
                value.name = 'Total')

ggplot() +
  geom_line(data = hourly2, aes(x = hour, y = Total, color = Type)) +
  facet_wrap(~borough) + ggtitle("EMS vs Complaint Calls, by Hour") +
  theme_set(theme_gray()) + scale_x_continuous(breaks = pretty_breaks())
```



Based on this plot, there isn't as much variability in EMS calls as I would have expected from the previous plot. Both datasets show consistency in call totals hour by hour, all with a small midday spike in activity. Also, it is important to note that for each borough there is an abnormal amount of calls registered as occuring at midnight - my initial though is that this is some sort of inaccuracy in the data.

What have we learned?

In conclusion there are a few key take aways from this analysis: * Both EMS and 311 Complaint calls occur most in Brooklyn and least in Staten Island * Although Brooklyn acounts for a high percentage of most complaint types, the other boroughs are not far behind in call totals, with the exception of Staten Island. * The 311 data is consisten with what would seem to be common complaints, and the most common have to do with heating issues, street conditions, plumbing, and construction issues. * Complaint calls tend to be made during the winter into spring months, levelling out in the summer and bottoming in September, across all boroughs. * EMS calls, whether on the hourly, daily, or monthly, show no signs of any major upward or downward trend during any time period. The amount remains generally steady over all time periods. * We cannot conclude that there is any significant relationship between the calls accounted for in both of these datasets. While this is true on the aggregate, exploring this data even further could prove that there are very specific relationship within the data.

Appendix: Data Dictionaries

NYC 311 Calls

```r
dict.311 <- fread('DataDictionaryNYC311.csv')
kable(dict.311)
```

Field Name	Field Size	Data Type	Data Format	Description
Unique Key	8	integer	Integer	Unique ID of each Service Request
Created Date	22	Date mm/dd/yy	character	Date the request was created
Closed Date	22	Date mm/dd/yy	character	Date the responding agency closed the
Agency	10	String	character	Acronym of responding city governmen
Agency Name	91	String	character	Full name of responding city governme
Complaint Type	104	String	character	General category of each complaint
Descriptor	104	String	character	Specific complaint under general catego
Location Type	36	String	character	Type of location of each complaint
Incident Zip	10	Integer	character	Zip code where each complaint was ma
Incident Address	47	string	character	Address where complaint was made
Street Name	42	String	character	Street name of location of complaint
Cross Street 1	32	string	character	First cross street based on geo location
Cross Street 2	36	String	character	Second cross street based on geo locati
Intersection Street 1	32	String	character	First cross street based on geo location
Intersection Street 2	32	string	character	Second cross street based on geo locati
Address Type	12	String	character	Type of location where complaint was
City	22	String	character	City where complaint was made, based
Landmark	48	string	character	Will only show data if the location of i
Facility Type	15	String	Character	Type of facility associated with the req
Status	28	String	character	Status of complaint (open, in progress,
Due Date	22	Date mm/dd/yy	character	Date when responding agency is expect
Resolution Action Updated Date	22	Date mm/dd/yy	character	Date when responding agency last upd
Community Board	25		character	
Borough	13	string	character	Borough where incident occurred, geo v
X Coordinate (State Plane)	7	float	Integer	Geo validated x coordinate of incident
Y Coordinate (State Plane)	7	float	Integer	Geo validated y coordinate of incident
Park Facility Name	95	string	character	Name of Parks Dept facility if incident
Park Borough	13	string	character	Name of the Borough in which the par
School Name	95	string	character	If incident happened in a school, the na
School Number	11	integer	character	If incident happened in a school, the sc
School Phone Number	11	String	character	If incident happened in a school, the sc
School Address	120	String	character	If incident happened in a school, it's ac
School City	19	string	character	If incident happened in a school, the ci
School State	11	string	character	“” state
School Zip	11	integer	character	“” Zip code
School Not Found	1	String	cahracter	
School or Citywide Complaint	18	String	character	
Vehicle Type	23	string	character	If incident is reported in a taxi, the bo
Taxi Company Borough	13	string	character	If incident is in a taxi, which Borough
Taxi Pickup Location	27	String	cahracter	If incident is in a taxi, the location of i
Bridge Highway Name	42	string	character	Bridge name if incident happened on a
Bridge Highway Direction	30	string	character	If on bridge, which direction the incide
Road Ramp	7	string	character	If on a bridge, differentiation as to whe
Bridge Highway Segment	100	string	character	Additional info if incident on a bridge
Garage Lot Name	27	string	character	
Ferry Direction	0	string	character	
Ferry Terminal Name	95	String	character	
Latitude	16	float	double	Geo based latitude of incident location
Longitude	17	Float	Double	Geo based longitude of incident
Location	40	Tuple	character	Combination latitude and longitude of

EMS Calls

```
dict.ems = fread('DataDictionaryHW5.csv')
kable(dict.ems)
```

Field Name	Field Size	Data Type	Data Format	Description
CAD_INCIDENT_ID	9	Integer	integer	Incident ID using Julian d
INCIDENT_DATETIME	22	Date ymd_hms	character	Date and Time the incider
INITIAL_CALL_TYPE	6	string	character	Call type at incident creat
INITIAL_SEVERITY_LEVEL_CODE	1	integer	Integer	Priority assigned to the in
FINAL_CALL_TYPE	6	string	Character	Call type at incident close
FINAL_SEVERITY_LEVEL_CODE	1	integer	integer	Priority assigned after clos
FIRST_ASSIGNMENT_DATETIME	22	Date ymd_hms	character	Datetime of first unit assig
VALID_DISPATCH_RSPNS_TIME_INDC	1	string	character	Indicator of valid compone
DISPATCH_RESPONSE_SECONDS_QY	5	integer	integer	Time elapsed in seconds be
FIRST_ACTIVATION_DATETIME	22	Date ymd_hms	character	Date and time the first un
FIRST_ON_SCENE_DATETIME	22	Date ymd_hms	character	Date and time the first un
VALID_INCIDENT_RSPNS_TIME_INDC	1	string	character	Indicator of valid compone
INCIDENT_TRAVEL_TM_SECONDS_QY	5	integer	integer	Time elapsed in seconds be
FIRST_TO_HOSP_DATETIME	22	Date ymd_hms	character	Time elapsed in seconds be
FIRST_HOSP_ARRIVAL_DATETIME	22	Date ymd_hms	character	Date and time the first un
INCIDENT_CLOSE_DATETIME	22	Date ymd_hms	character	Date and time the incident
HELD_INDICATOR	1	string	character	Indicator that a unit could
INCIDENT_DISPOSITION_CODE	2	integer	integer	Code indicating the final o
BOROUGH	24	string	character	Borough of incident
ATOM	NA	string	character	Smallest subdivision of the
ZIPCODE	5	String	integer	Zipcode of incident
POLICEPRECINCT	3	String	integer	Police precinct of they nci
CITY COUNCIL DISTRICT	2	String	integer	City council district of inci
COMMUNITY DISTRICT	3	String	integer	Community school district
CONGRESSIONAL DISTRICT	2	String	integer	Congressional district of in
REOPEN INDICATOR	1	String	character	Indicator that at some poi
SPECIAL EVENT INDICATOR	1	String	character	Indicator the incident was
STANDBY INDICATOR	1	String	character	Indicates that units were a
TRANSFER INDICATOR	1	string	character	Indicates that units were c