NYC 311 Analysis

Title: VI , Author: Marwan Albanna

30th June 2022

Project Goal

NYC311 Service Requests & Resolution Analysis

Explore and analyze NYC311 Service requests (historical data sets) to understand diverse patterns, regular themes and trends, as well as community satisfaction levels and sentiment pulse (social network feeds) derived from resolution categories and timing.

Introduction: Preliminary description of the data

NYC311's mission is to provide the public with quick, easy access to all New York City government services and information while offering the best customer service. We help Agencies improve service delivery by allowing them to focus on their core missions and manage their workload efficiently. We also provide insight to improve City government through accurate, consistent measurement and analysis of service delivery. NYC311 is available 24 hours a day, 7 days a week, 365 days a year. We work to make government services accessible for all

This data is the New York City's primary source of government information and non-emergency services. NYC311 data originated from reports by New Yorkers to the authorities of a strange smell that was feared that it might cause chemical warfare especially from Bloods & Crips.

The data was enhanced more on January 29, 2009, when another maple syrup event commenced in northern Manhattan. Launched in March 2003, 311 now fields on average more than 50,000 calls a day, offering information about more than 3,600 topics: school closings, recycling rules, homeless shelters, park events, pothole repairs.

This data formed a giant arrow aiming at a group of industrial plants in north eastern New Jersey. The service has translators on call to handle some 180 different languages.

NYC Open Data makes the public data generated by various New York City Agencies and other City organizations available for public use. The data sets are available in a variety of machine-readable formats and are refreshed when new data becomes available. Data is presented by category, by City Agency, or by other City organization.

NYC311 works continuously to make government services more accessible to non-English speakers, with 180 languages available in the call center and more than 50 languages available online. The 311service request dataset is available from 2010 to present, is updated daily, and contains over 9 million rows of data.

The Data Set

• NYC311 is a very massive dataset with approximately 9,124,937 observations and approximately 52 variables.

• Due to the presence of so many duplicated observations, about 1,670,064 observations are dropped and 37 redundant and missing value columns also dropped. NYC311 dataset is left with about 7,454,873 observations and 15 columns for this specific analysis.

- By mutating the datetime using the lubridate function, I am able to get 5 new variables namely: year, month, wday, timetaken (difference between Created date and Closed date) and resolvetime (difference between due date and resolution action updated date)

- For this specific assignment, I filtered the data with Agency as the key which allowed me to only deal with Complaints that involved the Biggest Agency.

- The Department of Housing Preservation and Development (HPD) emerged as the biggest agency. Heating Complaint was visualized as the most complaint that 311 was used on concerning HPD Agency.

- 30% of complaints are being resolved within 24 hours while the rest takes from 1 to 20 or more days.

- Complaints from unspecified Boroughs take longer time to be solved.

Second dataset: DOL annual Population Estimates for NY State & Boroughs

Each year, the Census Bureau's Population Estimates Program (PEP) utilizes current data on births, deaths, and migration to calculate population change and produce a time series of estimates of population, demographic components of change, and housing units. The annual time series of estimates begins with the most recent decennial census data and extends to the vintage year.

Each vintage of estimates includes all years since the most recent decennial census, and the latest vintage of data available supersedes all previously-produced estimates for those dates.

The data includes intercensal estimates for 1970-2009 and postcensal estimates for 2010 and later for New York State and all New York State Counties, and the decennial Census counts for 1970-2010.

Intercensal estimates are population estimates produced for the years between two decennial censuses when both the beginning and ending populations are known. They are produced once a decade by adjusting the existing time series of postcensal estimates for the entire decade to smooth the transition from one decennial census count to the next.

They differ from the postcensal estimates that are released annually because they rely on a mathematical formula that redistributes the difference between the April 1 postcensal estimate and April 1 census count for the end of the decade across the estimates for that decade.

For dates when both postcensal and intercensal estimates are available, intercensal estimates are preferred.

Joining the two dataset: NYC311 AND NYCPopulation DATA

This is aimed at establishing the relationship between the population of New York City and the complaints that are made through 311. This data of population is joined together with the 311 one to find out the following:

- What is the rate of complaining per population?
- What is the demographic distribution in New York City within the Boroughs?
- Is there any relationship between population growth over the years to complaints growth over the years?

Motivation

I am therefore motivated by the above to establish how effective 311 has been in New York city and how easy it is to know which Borough to live in case one decides to relocate to New York. I am excited to juggle around this massive dataset with my limited knowledge in visualization and help me in my discovery journey.

Packages & its purpose

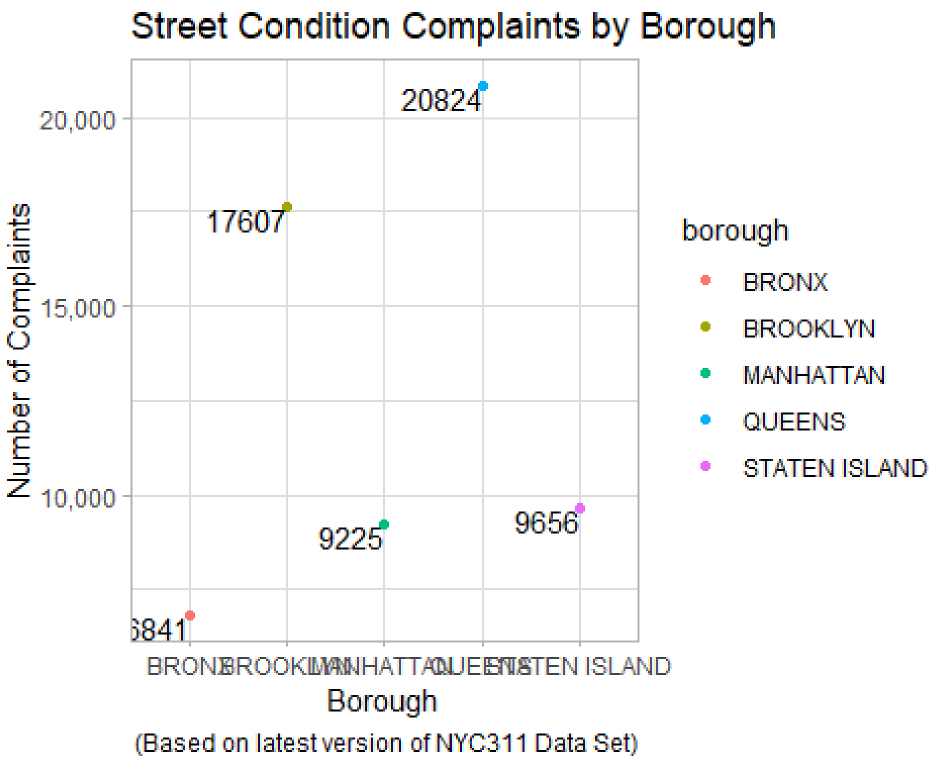
1. tidyverse : give essential data transformation capabilities in a unified package.
2. dplyr : offering a dependable set of verbs that assist you in resolving the most typical data manipulation problems.
3. lubridate : makes working with dates and times easier.
4. data.table : combines database procedures like subset, with, and by, and offers faster joins than merge does.
5. devtools : By offering R functions that accelerate and simplify typical operations, we may "make package development simpler.
6. ggmap : The capacity to view data and models overlaid with their fundamental social landmarks and geographic context is crucial in spatial statistics, For plotting maps.
7. rmarkdown : maintain the formatting, code output, and text in the original.Rmd file.
8. tidytext : Provide tools and data sources to assist the conversion of text between tidy formats.
9. jsonlite : a decently quick JSON parser and generator that is tailored for the web and statistics data provides straightforward, adaptable tools for working with JSON in R.
10. scales : To scale the values in a vector, matrix, or data frame, use the scale() function.
11. wordcloud : Visualization of textual data.
12. janitor : used to clean the column's names.
13. Amelia : used to display data that has missing values.
14. Skimr : Used to view a data summary.
15. Leaflet : For interactive map

Loading nyc311 dataset

```
nyc311<-
data.table::fread("311_Service_Requests_from_2010_to_Present.csv",nrows=10000
00) %>%
  clean_names()



# Convert blank values to NA
nyc311[nyc311==""]<-NA

# Converting the date & time to required format
nyc311 <- nyc311 %>%
  mutate(closed_date = mdy_hms(closed_date),
         created_date = mdy_hms(created_date),
         resolution_action_updated_date =
mdy_hms(resolution_action_updated_date),
         due_date = mdy_hms(due_date),
         wday = wday(created_date, label = TRUE),
         month = month(created_date, label = TRUE),
         year = year(created_date),
         timetaken = (created_date - closed_date),
         resolvetime = (due_date - resolution_action_updated_date))
```

Street condition by Borough

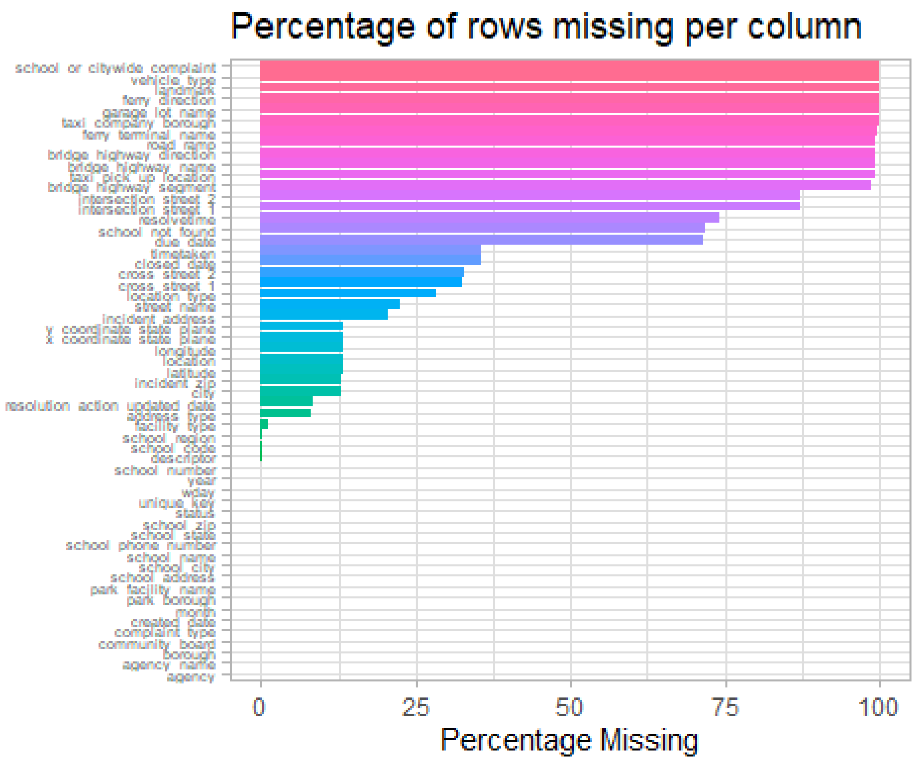
```
nyc311 %>%
  filter(complaint_type=="Street Condition") %>%
  count(borough,complaint_type) %>%
ggplot(aes(x = borough,y=n)) +
  geom_point(aes(color = borough))  +
  geom_text(aes(label=n),hjust=1,vjust=1)+
  theme_light()+
  ggtitle("Street Condition Complaints by Borough ") +
  xlab("Borough") + ylab("Number of Complaints")  +
  labs(caption = "(Based on latest version of NYC311 Data Set)") +
scale_y_continuous(labels = scales::comma)
```


Street Condition Complaints by Borough

(Based on latest version of NYC311 Data Set)

Checking missing data in the dataset

```
data.frame(n=colSums(is.na(nyc311))) %>%
  arrange(desc(n)) %>%
  rownames_to_column() %>%
  mutate(per_missing=n/nrow(nyc311)*100,
         rowname=fct_reorder(rowname,per_missing)) %>%
  ggplot(aes(x=per_missing,y=rowname,fill=rowname))+
  geom_col(show.legend = FALSE)+
  labs(title="Percentage of rows missing per column",
       x="Percentage Missing",
       y="")+
  theme(axis.text.y=element_text(size=5))
```



Sub Data Sets

nyc311 data set contains 52 columns, so at first step we are going to select some important variables which we are going to used for insights. first we will create the subset of nyc311 for text analysis and second data set for visualizations.

```
nyc311_sub <- nyc311 %>%
  select(created_date,closed_date,agency,agency_name,complaint_type,

incident_zip,status,due_date,resolution_action_updated_date,borough,latitude,
longitude,location,
                                landmark, park_borough,
school_city,school_state,school_zip,location_type,community_board,

incident_address,taxi_company_borough,descriptor,
         resolution_action_updated_date,due_date ,
wday,month,year,timetaken,resolvetime) %>%
  ungroup()
```

Summary of the data

```
summary(nyc311_sub)

##   created_date                    closed_date                      agency
##  Min.   :2014-10-17 08:49:04   Min.   :1900-01-01 00:00:00
Length:1000000
##  1st Qu.:2014-12-03 08:59:47   1st Qu.:2014-12-08 00:00:00   Class
:character
```

```
##  Median :2015-01-20 17:18:05   Median :2015-01-24 12:40:00   Mode
:character
##  Mean   :2015-01-16 23:44:39   Mean   :2015-01-18 12:19:51
##  3rd Qu.:2015-03-01 16:45:21   3rd Qu.:2015-03-03 12:07:19
##  Max.   :2015-04-14 02:14:40   Max.   :2015-04-15 01:06:54
##                                NA's   :356826
##  agency_name        complaint_type     incident_zip          status
##  Length:1000000     Length:1000000     Length:1000000     Length:1000000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     due_date                    resolution_action_updated_date
##  Min.   :2014-10-17 13:57:08   Min.   :2014-01-23 10:35:00
##  1st Qu.:2014-12-11 15:54:39   1st Qu.:2014-12-05 00:00:00
##  Median :2015-01-30 22:06:45   Median :2015-01-22 00:00:00
##  Mean   :2015-01-28 09:50:11   Mean   :2015-01-18 11:02:09
##  3rd Qu.:2015-03-15 23:25:14   3rd Qu.:2015-03-02 15:38:08
##  Max.   :2016-04-12 09:50:48   Max.   :2015-04-15 01:06:54
##  NA's   :715324                NA's   :84099
##    borough              latitude        longitude        location
##  Length:1000000     Min.   :40.50    Min.   :-74.25   Length:1000000
##  Class :character   1st Qu.:40.67    1st Qu.:-73.96   Class :character
##  Mode  :character   Median :40.73    Median :-73.93   Mode  :character
##                     Mean   :40.74    Mean   :-73.92
##                     3rd Qu.:40.82    3rd Qu.:-73.88
##                     Max.   :40.91    Max.   :-73.70
##                     NA's   :132920   NA's   :132920
##    landmark          park_borough       school_city        school_state
##  Length:1000000     Length:1000000     Length:1000000     Length:1000000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   school_zip        location_type      community_board    incident_address
##  Length:1000000     Length:1000000     Length:1000000     Length:1000000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  taxi_company_borough  descriptor         wday           month
##  Length:1000000       Length:1000000     Sun:100407   Mar    :186594
##  Class :character     Class :character   Mon:169949   Feb    :185576
##  Mode  :character     Mode  :character   Tue:160847   Jan    :171653
##                                           Wed:157230   Nov    :158788
##                                           Thu:150870   Dec    :154122
##                                           Fri:150616   Oct    : 76429
##                                           Sat:110081   (Other): 66838
##       year       timetaken        resolvetime
##  Min.   :2014   Length:1000000    Length:1000000
##  1st Qu.:2014   Class :difftime   Class :difftime
##  Median :2015   Mode  :numeric    Mode  :numeric
##  Mean   :2015
##  3rd Qu.:2015
##  Max.   :2015
##

skim(nyc311_sub)
```

Data summary

Name	nyc311_sub
Number of rows	1000000
Number of columns	28
_______________________	
Column type frequency:	
character	17
difftime	2
factor	2
numeric	3
POSIXct	4
________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
agency	0	1.00	2	10	0	45	0
agency_name	0	1.00	2	91	0	552	0
complaint_type	0	1.00	3	41	0	204	0
incident_zip	129281	0.87	1	10	0	451	0
status	0	1.00	4	10	0	7	0
borough	0	1.00	5	13	0	6	0
location	132920	0.87	26	40	0	237741	0
landmark	999840	0.00	4	27	0	61	0

park_borough	0	1.00	5	13	0	6	0
school_city	0	1.00	5	19	0	45	0
school_state	0	1.00	2	11	0	2	0
school_zip	0	1.00	3	11	0	168	0
location_type	285033	0.71	3	36	0	122	0
community_board	0	1.00	8	25	0	77	0
incident_address	205630	0.79	1	80	0	224716	0
taxi_company_borough	999097	0.00	5	13	0	5	0
descriptor	971	1.00	3	82	0	1112	0

Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
timetaken	356826	0.64	-15116534 secs	3636492524 secs	-72360 secs	153682
resolvetime	742864	0.26	-14857334 secs	31535954 secs	26169 secs	116767

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
wday	0	1	TRUE	7	Mon: 169949, Tue: 160847, Wed: 157230, Thu: 150870
month	0	1	TRUE	7	Mar: 186594, Feb: 185576, Jan: 171653, Nov: 158788

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
latitude	132920	0.87	40.74	0.09	40.50	40.67	40.73	40.82	40.91	▁▆▇▆▆
longitude	132920	0.87	-73.92	0.08	-74.25	-73.96	-73.93	-73.88	-73.70	▁▁▇▇▁
year	0	1.00	2014.61	0.49	2014.00	2014.00	2015.00	2015.00	2015.00	▇▁▁▁▇

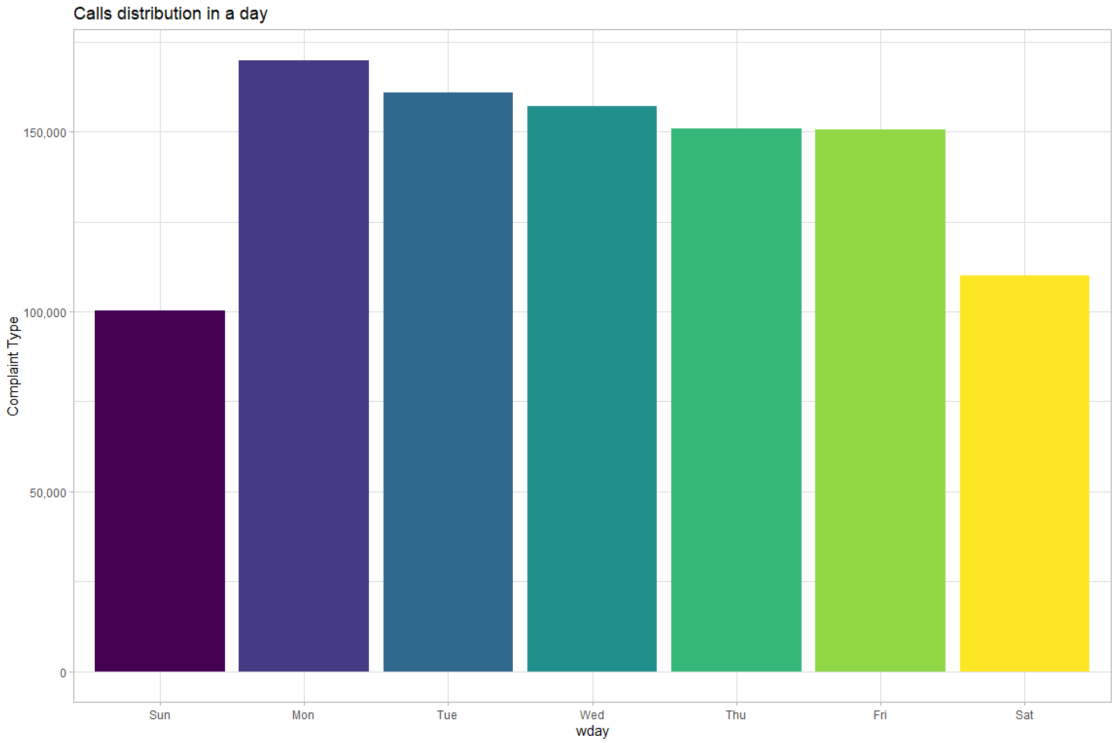
Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
created_date	0	1.00	2014-10-17 08:49:04	2015-04-14 02:14:40	2015-01-20 17:18:05	476559
closed_date	356826	0.64	1900-01-01 00:00:00	2015-04-15 01:06:54	2015-01-24 12:40:00	323015
due_date	715324	0.28	2014-10-17 13:57:08	2016-04-12 09:50:48	2015-01-30 22:06:45	261626
resolution_action_updated_date	84099	0.92	2014-01-23 10:35:00	2015-04-15 01:06:54	2015-01-22 00:00:00	317780

Exploring Patterns in the data

Which day of the week was 311 busiest?

```
nyc311_sub %>%
  group_by(wday) %>%
  dplyr::summarise(count=n()) %>%
  ggplot(mapping = aes(x=wday,y=count))+
  geom_col(aes(fill = wday),show.legend = FALSE) +
  labs(title="Calls distribution in a day",x="wday",y="Complaint Type")+
  scale_y_continuous(labels=scales::comma_format())
```

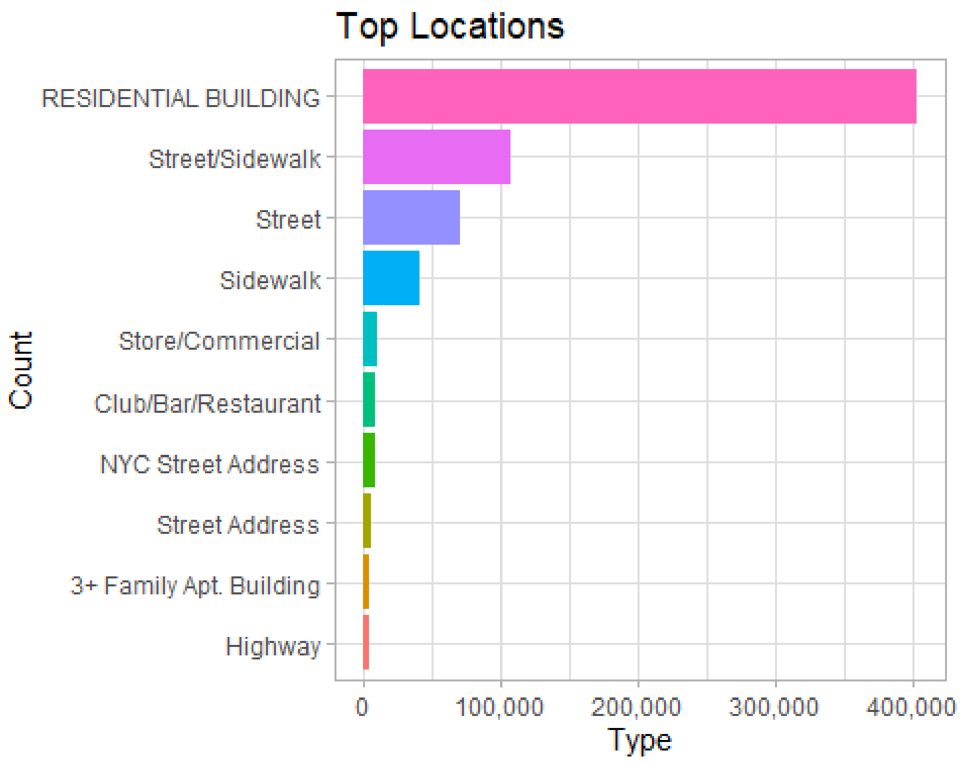

Calls distribution in a day

Observations form above plot:

- Weekends are the less busiest days of the week with Sunday being the least
- New York residents make more calls on Monday through Friday respectively
- The beginning of the week see 311 very busy and eases up as weekend draws near.

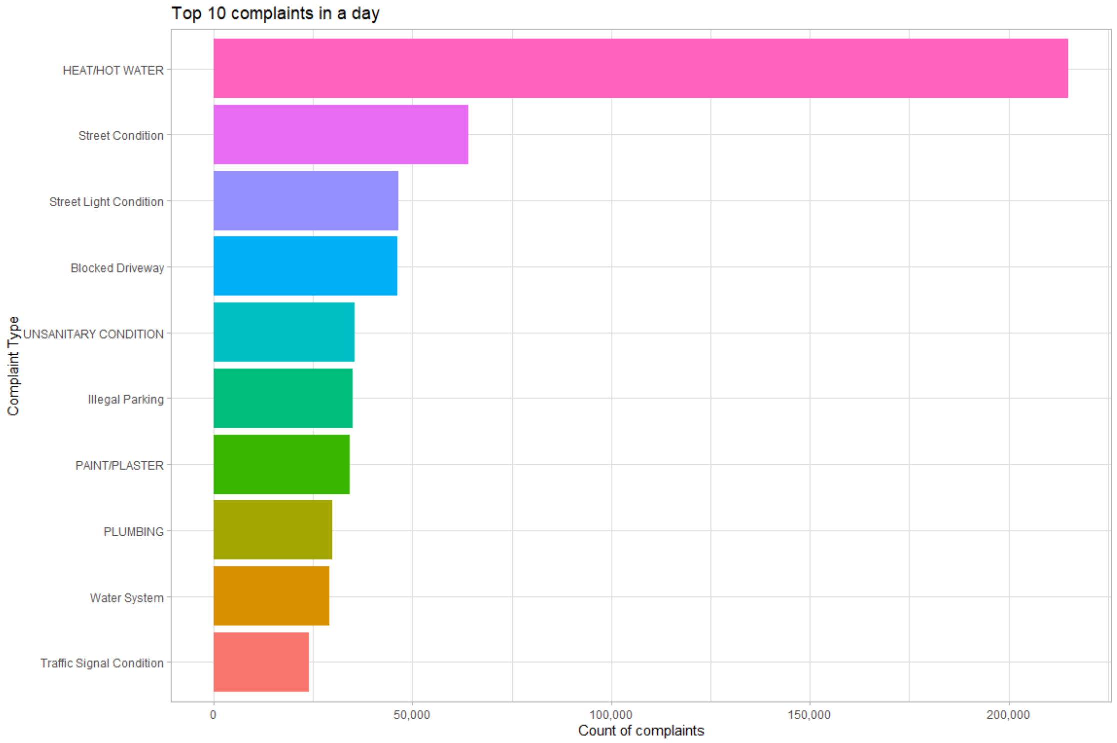
What are the top locations?

```
nyc311_sub %>%
  group_by(location_type) %>%
  dplyr::summarise(count=n()) %>%
  arrange(desc(count)) %>%
  na.omit() %>%
  head(10) %>%
  mutate(location_type=fct_reorder(location_type,count)) %>%
  ggplot(mapping = aes(y=location_type,x=count))+
  geom_col(aes(fill = location_type),show.legend = FALSE) +
  labs(title="Top Locations",x="Type",y="Count")+
  scale_x_continuous(labels=scales::comma_format())
```



What were the top 10 most complained that HPD received through 311?

```
top10 <- nyc311_sub %>%
  group_by(complaint_type) %>%
  summarize(count=n()) %>%
  slice_max(count,n=10) %>%
  mutate(complaint_type=fct_reorder(complaint_type,count))

ggplot(top10,aes(y=complaint_type,x=count)) +
  geom_bar(stat="identity",aes(fill=complaint_type),show.legend = FALSE) +
  labs(title="Top 10 complaints in a day",x="Count of
complaints",y="Complaint Type")+
  scale_x_continuous(labels=scales::comma_format())
```

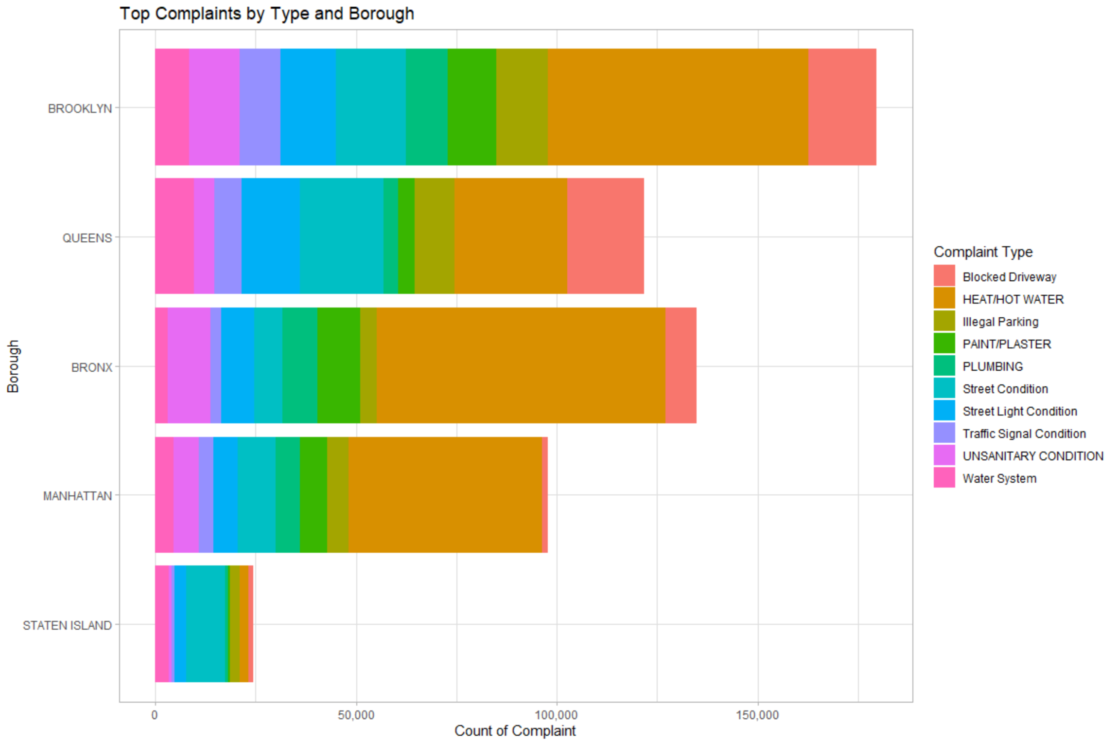


Observations form above plot:

- Heating was the most complaint that New Yorkers called 311 to complain about.
- The Department of Housing Preservation and Development (HPD) received approximately 659,046 Complaints from New York residents on HEATING.

Which Borough raised the highest number of Complaints?

```
nyc311_sub %>%
  filter(complaint_type %in% (top10 %>% pull(complaint_type))) %>%
  group_by(borough,complaint_type) %>%
  dplyr::summarise(count=n()) %>%
  ungroup() %>%
  filter(borough!="Unspecified") %>%
  mutate(borough=fct_reorder(borough,count)) %>%
ggplot(aes(x=count, y=borough, fill=complaint_type)) +
geom_col(show.legend = TRUE) +
labs(title="Top Complaints by Type and Borough",
     x="Count of Complaint",
     y="Borough",
     fill="Complaint Type")+
  scale_x_continuous(labels=scales::comma_format())
```

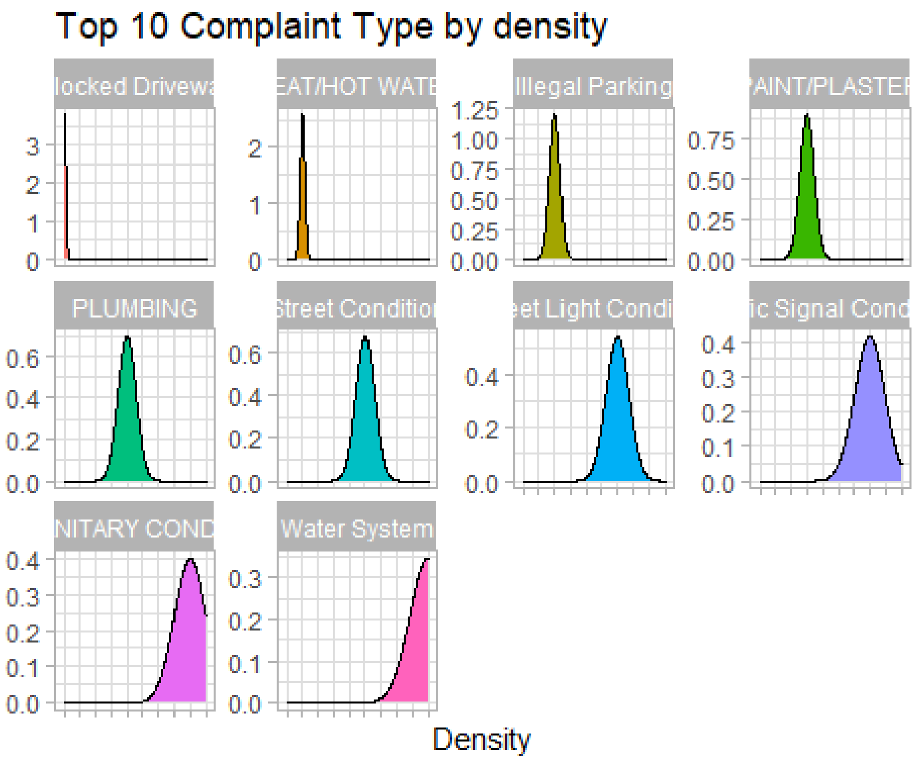


Observations form above plot:

- The analysis above indicates that Brooklyn Borough kept calling 311 frequently and complained more on Heating.
- Staten Island Borough on the other hand had the lowest amount of Complaints.

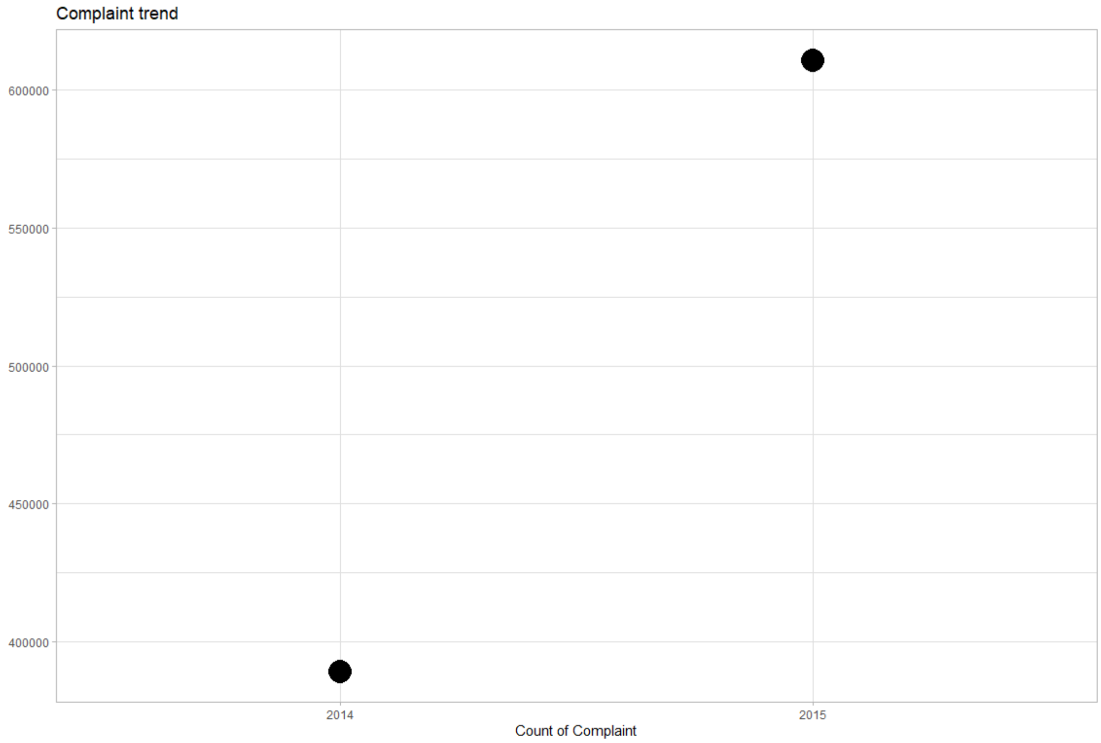
Top 10 Complaint Denstiy

```
nyc311_sub %>%
  filter(complaint_type %in% (top10 %>% pull(complaint_type))) %>%
  ggplot(aes(x=complaint_type))+
  geom_density((aes(fill=complaint_type)))+
  facet_wrap(~complaint_type,scales="free_y")+
  labs(title="Top 10 Complaint Type by density",
       fill="Complaint Type",
       y="",
       x="Density")+
  theme(legend.position = "none",
        axis.text.x = element_blank())
```



What is the relationship between the complaints and the number of years? Have the complaints increased, decreased or stagnated?

```
nyc311_sub %>%
  group_by(year) %>%
  dplyr::summarise(count=n()) %>%
  ggplot(aes(x=as.factor(year), y=count)) +
  geom_point(size=8,aes(group=year))+
  labs(title="Complaint trend",
     x="Count of Complaint",
     y="")
```

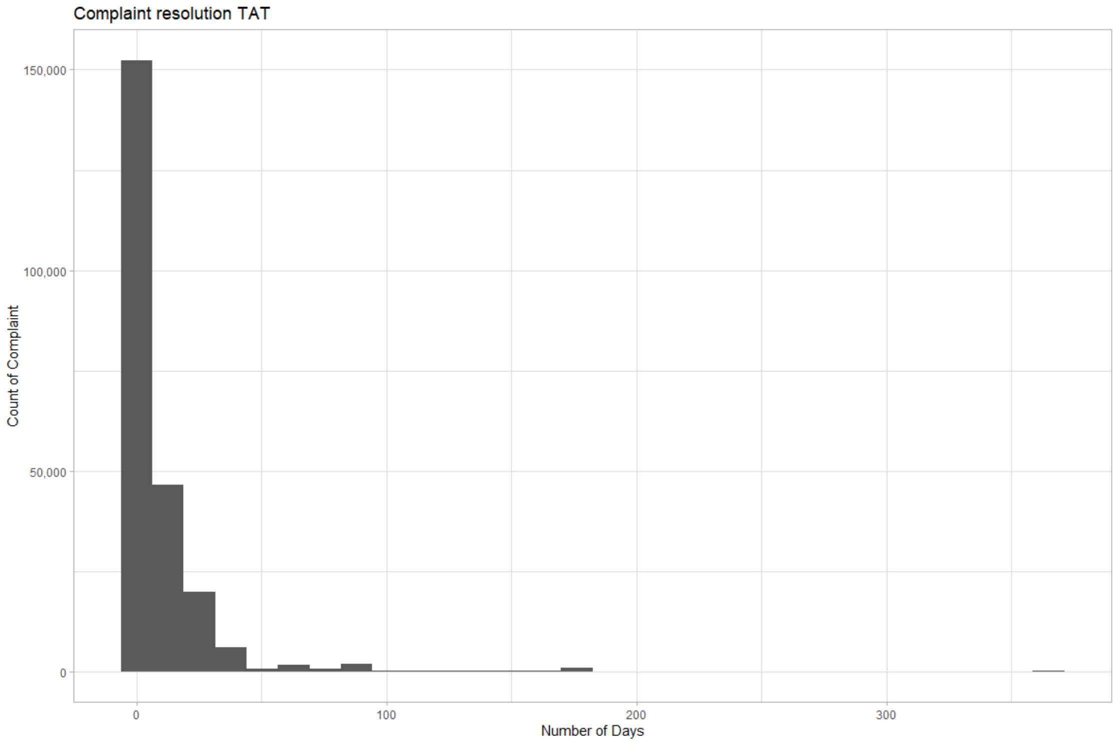


Observations:

- The above analysis indicates that as years have progressed, complaints to 311 have increased.
- Most of the unspecified Boroughs were from complaints between 2010 and 2011 when the platform was still being developed.
- Between 2012 and 2015, there are no observations from unspecified Boroughs.
- Complaints have increased as years have progressed with 2014 registering the highest number of complaints.

How long does it take for HPD Agency to close a complaint once it is created?

```
nyc311_sub %>%
  filter(resolvetime>0) %>%
  ggplot(aes(x=resolvetime/(60*60*24))) +
  geom_histogram()+
   labs(title="Complaint resolution TAT",
     x="Number of Days",
     y="Count of Complaint")+
  scale_y_continuous(labels = scales::comma_format())
```

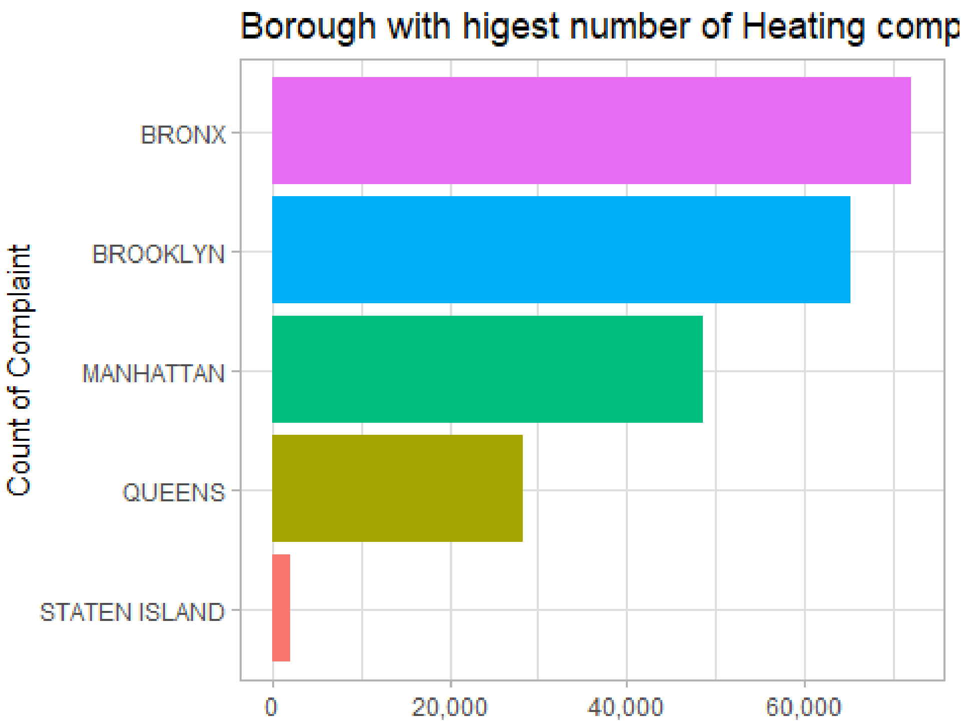


Observations

- Most of the coplaints are sorted within a week, there are some outliers which takes upwards of few months

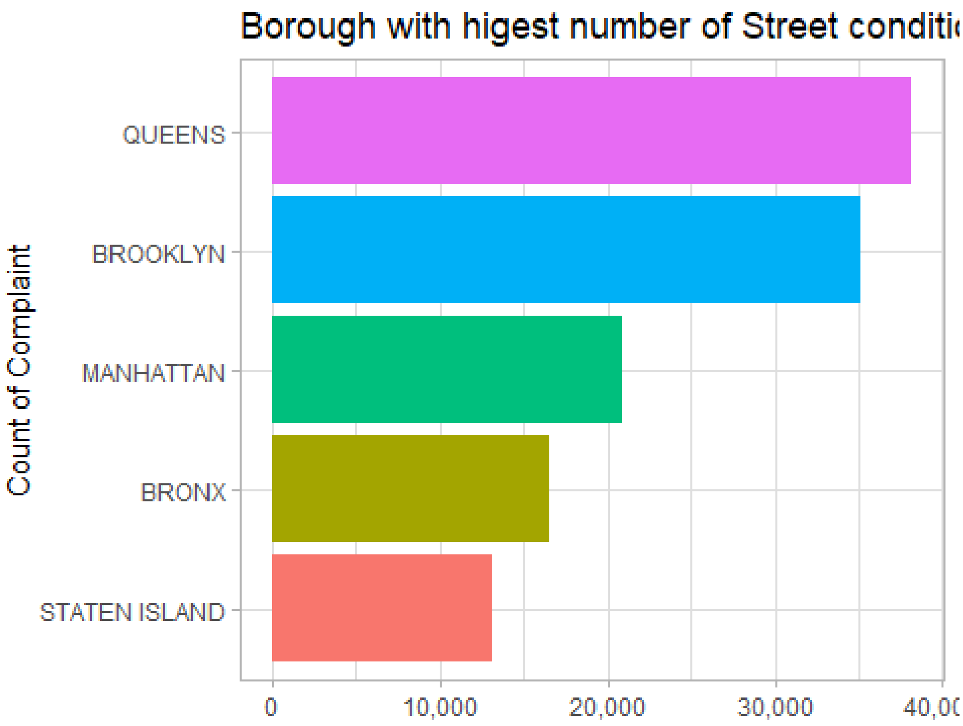
Which borough has the highest number of Heating complaints?

```
ds_heat <- nyc311_sub %>%
  mutate(complaint_type=tolower(complaint_type)) %>%
  filter(str_detect(complaint_type,"heat")) %>%
  group_by(borough) %>%
  dplyr::summarise(Count = n()) %>%
  ungroup() %>%
  mutate(borough = fct_reorder(borough,Count))

ds_heat %>%
  ggplot(aes(y = borough,x = Count)) +
  geom_col(aes(fill=borough),show.legend = FALSE) +
  labs(title="Borough with higest number of Heating complaints",
     x= "",
     y="Count of Complaint")+
  scale_x_continuous(labels=scales::comma_format())
```



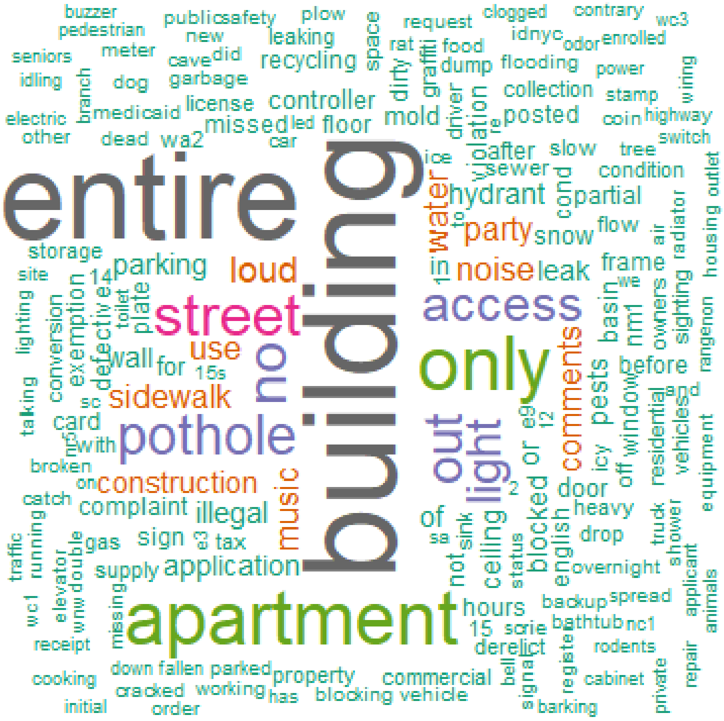
Which borough has the highest number of Street condition problem?

```
ds_street <-  nyc311_sub %>%
  mutate(complaint_type=tolower(complaint_type)) %>%
  filter(str_detect(complaint_type,"street")) %>%
  group_by(borough) %>%
  dplyr::summarise(Count = n()) %>%
  ungroup() %>%
  mutate(borough = fct_reorder(borough,Count)) %>%
  filter(borough!="Unspecified")

ds_street %>%
  ggplot(aes(y = borough,x = Count)) +
  geom_col(aes(fill=borough),show.legend = FALSE) +
  labs(title="Borough with higest number of Street condition complaints",
     x= "",
     y="Count of Complaint")+
  scale_x_continuous(labels=scales::comma_format())
```



What are the top words in the complaint

```
words <- nyc311_sub %>%
  unnest_tokens(word,descriptor) %>%
  count(word)


wordcloud(words = words$word, freq = words$n, min.freq = 1,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```

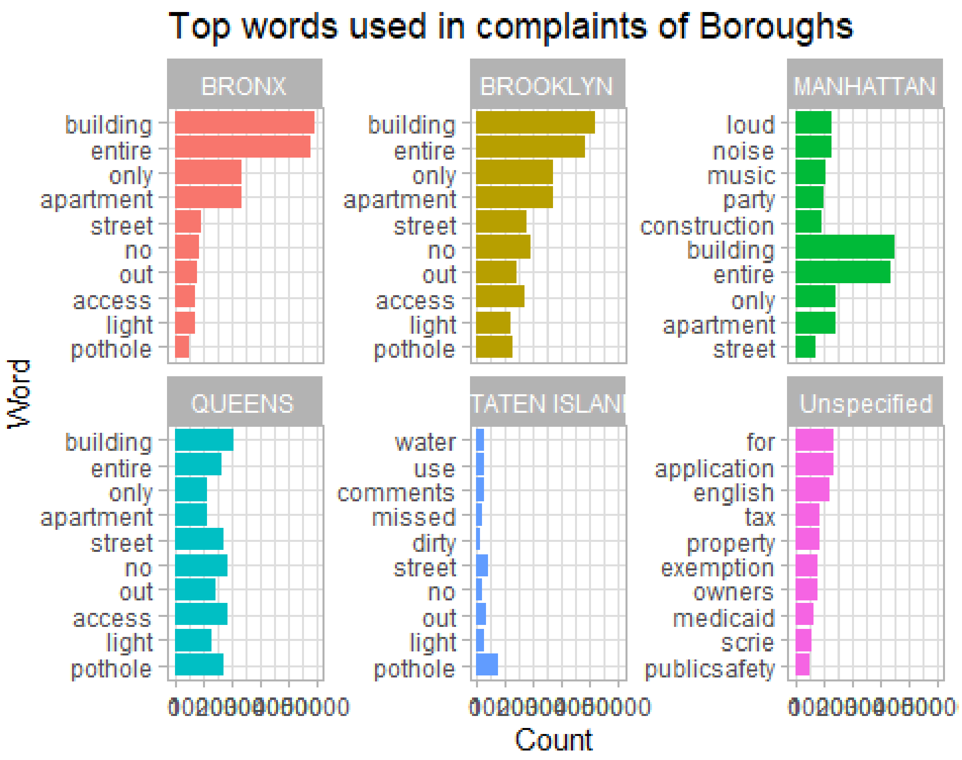


Observations

- Most common problem are potholes, loud music, heat & light issues

Top word used in main boroughs

```
nyc311_sub %>%
  unnest_tokens(word,descriptor) %>%
  count(borough,word) %>%
  group_by(borough) %>%
  slice_max(n,n=10) %>%
  mutate(word=fct_reorder(word,n)) %>%
  ggplot(aes(x=n,y=word,fill=borough))+
  geom_col(show.legend = FALSE)+
  facet_wrap(~borough,scale="free_y")+
  labs(title="Top words used in complaints of Boroughs",
       x="Count",
       y="Word")
```



Findings : Complaint Closing Date

As per above graph, January and March has the highest ratio of closing, before we visualize the complaints logged dates i.e. created date and found that same both January and March have the highest ratio, this means mostly complaints closed on same month. there is no such fluctuations in complaint creation date and closing date.

Reading DOL dataset From Web API

```
nycp<-fromJSON("https://data.ny.gov/resource/krt9-ym2k.json")

str(nycp)

## 'data.frame':    1000 obs. of  5 variables:
##  $ fips_code   : chr  "36000" "36001" "36003" "36005" ...
##  $ geography   : chr  "New York State" "Albany County" "Allegany County"
"Bronx County" ...
##  $ year        : chr  "2021" "2021" "2021" "2021" ...
##  $ program_type: chr  "Postcensal Population Estimate" "Postcensal
Population Estimate" "Postcensal Population Estimate" "Postcensal Population
Estimate" ...
##  $ population  : chr  "19835913" "313743" "46106" "1424948" ...

nycp <- nycp %>%
  rename("borough"="geography")
```

Summary DOL

```
summary(nycp)
```

```
##   fips_code           borough               year           program_type      
##  Length:1000        Length:1000        Length:1000        Length:1000       
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##   population       
##  Length:1000       
##  Class :character  
##  Mode  :character  
```

Data Descriptor Population Data Set

```
skim(nycp)
```

Data summary

Name	nycp
Number of rows	1000
Number of columns	5
_______________________	
Column type frequency:	
character	5
________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
fips_code	0	1	5	5	0	63	0
borough	0	1	11	19	0	63	0
year	0	1	4	4	0	14	0
program_type	0	1	22	31	0	3	0
population	0	1	4	8	0	995	0

Counties to Borough

Annual Population data used counties name not Borough so we will convert counties to Borough as below:

Bronx County TO BRONX

Kings County TO BROOKLYN

New York County TO MANHATTAN

Richmond County TO STATEN ISLAND

Queens County TO QUEENS

```
nycp_sub <- nycp %>%
  filter(program_type=="Postcensal Population Estimate") %>%
 mutate(borough = case_when(str_detect(borough,"Bronx County")~"BRONX",
                            str_detect(borough,"Kings County")~"BROOKLYN",
                            str_detect(borough,"New York
County")~"MANHATTAN",
                            str_detect(borough,"Richmond County")~"STATEN
ISLAND",
                            str_detect(borough,"Queens County")~"QUEENS",
                            TRUE ~ borough)) %>%
  select(borough, year, population) %>%
  mutate(year=as.numeric(year)) %>%
  filter(borough %in% c("BROOKLYN", "BRONX", "MANHATTAN", "STATEN ISLAND",
"QUEENS")) %>%
  filter(year %in% c(2010,2011, 2012,2013,2014,2015))
```

Grouping Data Set For Joining

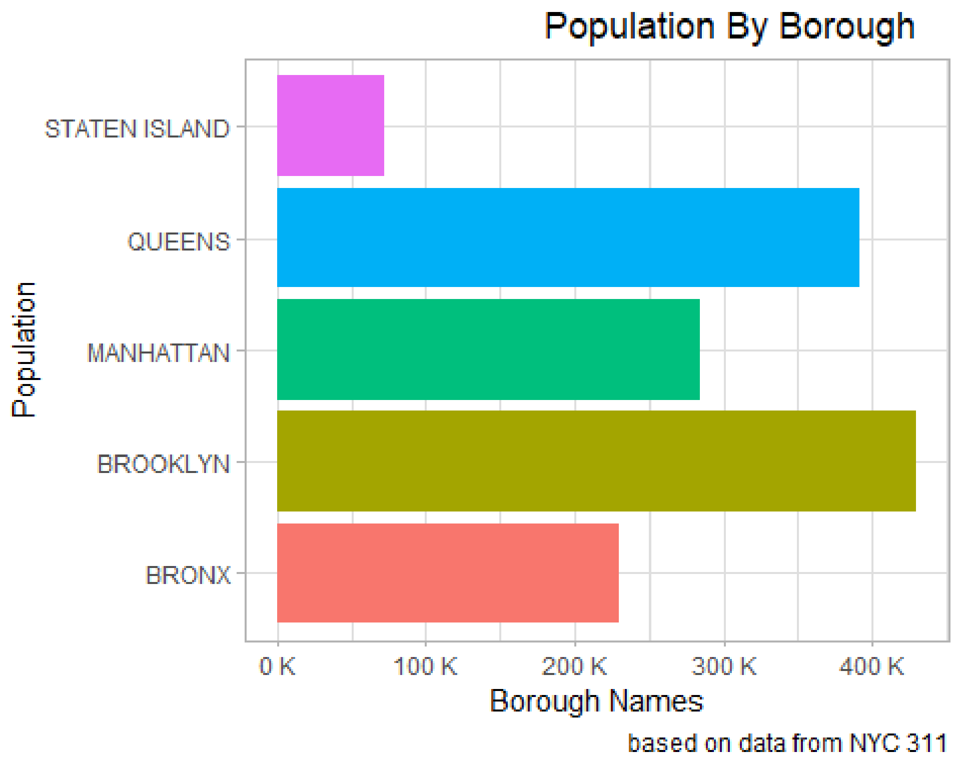
```
nyc311_fil <- nyc311_sub %>%
  group_by(agency, borough, complaint_type, year) %>%
  summarize(count=n())
```

Joiing two datasets - NYC311 AND NYCPopulation DATA

```
nyc_join <- nyc311_fil %>%
 left_join(nycp_sub)
```

Population per Borough in NYC

```
nyc_join %>%
#  ungroup() %>%
  filter(year==max(year),
         borough!="Unspecified") %>%
  mutate(population=as.numeric(population),
    borough=fct_reorder(borough,population)) %>%
ggplot(aes(fill=borough, x=as.numeric(population), y=borough)) +
geom_bar(stat="identity",show.legend = FALSE) +
     theme(plot.title = element_text(hjust = 0.9)) +
  scale_x_continuous(labels=scales::comma_format())+
    labs(title = "Population By Borough",
       x = "Borough Names",
       y = "Population",
       caption = "based on data from NYC 311",
       fill="Borough") +
    scale_x_continuous(labels = scales::unit_format(unit = "K", scale = 1e-
6))
```

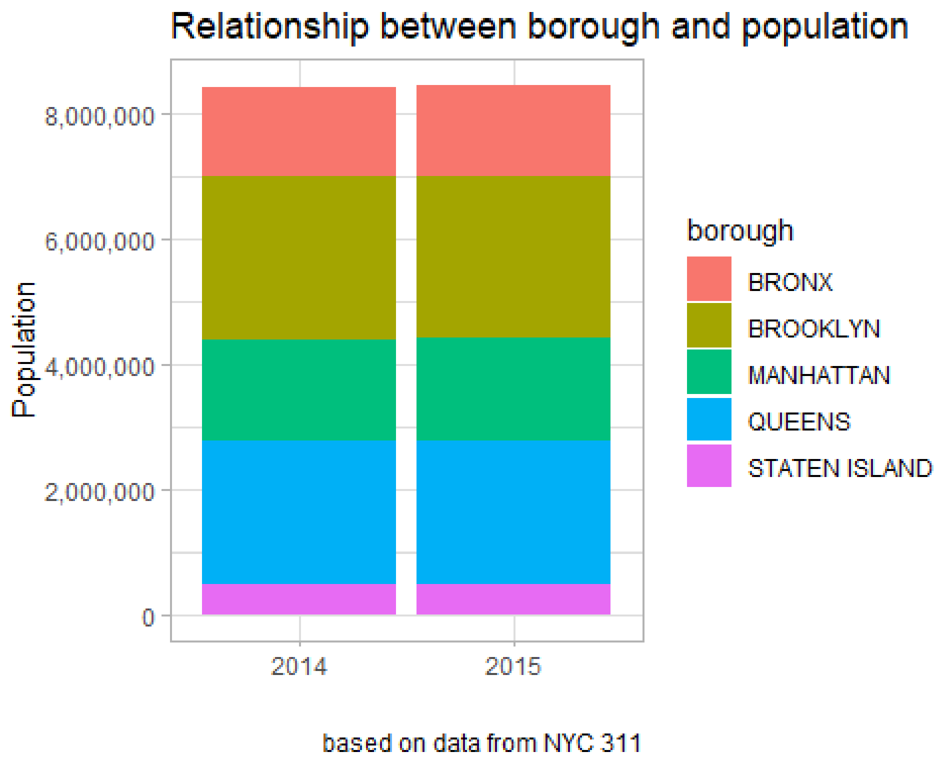


#Findings : Population By Borough

BROOKLYN & QUEENS is highest populous borough in New York.

NYC Population Trend

```
pop_2015<- nyc_join %>%
  filter(year %in% c(2010:2015))

pop_2015 %>%
  mutate(population=as.numeric(population)) %>%
  group_by(borough,year) %>%
  dplyr::summarise(pop=mean(population,na.rm=TRUE)) %>%
  ggplot(aes(fill=borough, y=pop, x=as.factor(year))) +
    geom_col() +
    labs(title = "Relationship between borough and population",  x= "",  y=
"Population",caption = "based on data from NYC 311") +
   scale_y_continuous(labels=scales::comma_format())
```



Findings : Reletionship between borough & polulation

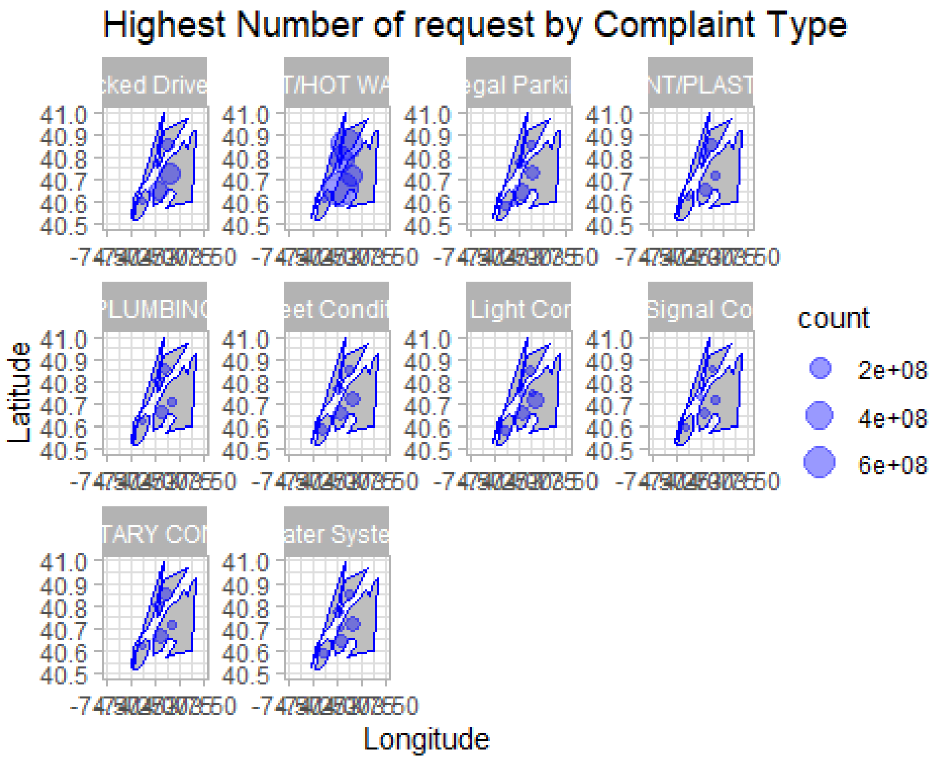
The chart shows that the population is reducing slightly from 2017 to 2020. Addition to that, BROOLYN borough has the highest population through all years. However, STATEN ISLAND borough has the lowest.

From the below two maps It is clear that Manhattan is producing the highest number of Noise Complaints, and some Noise complaints are coming from Brooklyn and Bronx. The lowest are coming from Staten Island. (According to NY City map: Bronx is located at North, Manhattan at middle below Manhattan, Queens at East, Brooklyn at South and Staten Island is the Island at the South West)

let's plot langitude and latitude points, and analyze highest no. of service requests by complaint types

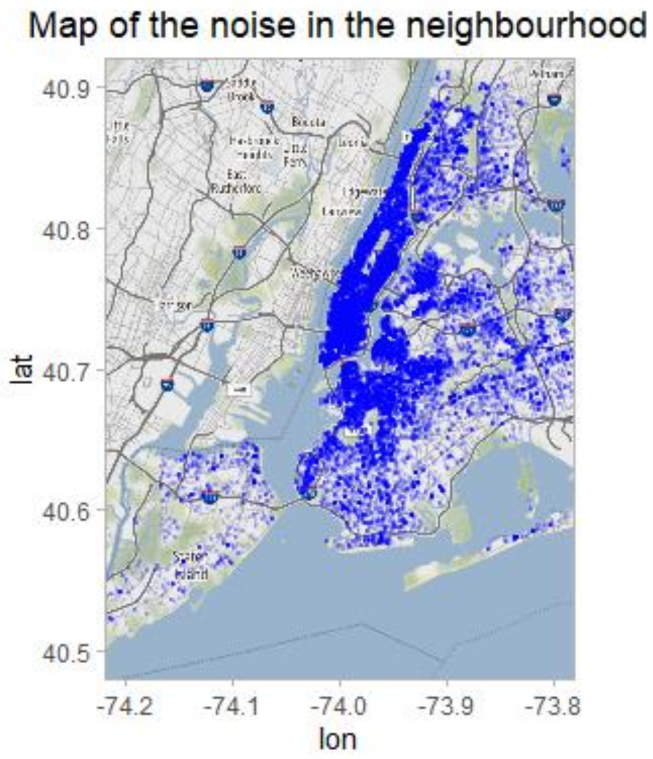
```
count_complaint <- nyc311_sub %>%
  group_by(complaint_type,borough) %>%
  add_count() %>%ungroup() %>%
  select(borough,complaint_type,longitude,latitude,n) %>%
  distinct(borough,complaint_type,n,longitude,latitude) %>%
  filter(complaint_type %in% (top10 %>% pull(complaint_type))) %>%
  group_by(borough, complaint_type) %>%
  summarise(longitude=mean(longitude,na.rm=TRUE),
            latitude=mean(latitude,na.rm=TRUE),
            count=sum(n))
```

```
library(rworldmap)

newmap <- getMap(resolution = "high")
nyc_coorflimits <- data.frame( long = c(-74.5, -73.5), lat = c(40.5, 41),
stringsAsFactors = FALSE)

nyc <- ggplot() + geom_polygon(data = newmap, aes(x=long, y = lat, group =
group), fill = "gray", color = "blue",)  + xlim(-74.5, -73.5) + ylim(40.5,
41)

nyc +
  geom_point(data=count_complaint, aes(longitude, latitude, size=count),
colour="blue",alpha=0.4)  +
  facet_wrap(~complaint_type, scales = "free") +
  labs(x = "Longitude", y = "Latitude", title = "Highest Number of request by
Complaint Type",color="Number of requests")
```



Noise Map

Here we created a map that shows the areas that produce the highest number of noise complaints.



Conclusion

This document has been created by importing data from nyc311 dataset to R software. The data has gone through the stage of tydying and transformation in order to help in Visualization, Modelling and Communication.

This being a massive data, visualization and modelling has been limited to some observations to allow the me deduce a few things from the sample.

nyc311 dataset is very informative and several things have been realized from the sample data that has been visualized:

- Brooklyn had the highest incidences of complaints compared to other Boroughs and Brooklyn had more complaint types described by both floor and pests than other Boroughs.
- HPD Agency is the biggest in terms of complaints made through 311
- Heat related complaints- Heat related complaints are the most common complaints of New Yorkers across boroughs.
- The number of complaints received by HPD are enormous compared to other boroughs and consequently HPD takes longer resolution time.
- May be HPD can consider employing more personnel to assist especially in the months that heating complaints become too much.
- Weekends complaints for HPD are lower compared to other complaints like noise.
- Brooklyn has the highest volume of complaints.
- January is the month that receives the highest calls of complainants especially on heating.
- This can be attributed to the largest complaint to HPD Agency of heating.
- May to September are the months that give HPD Agency some rest since the complaints fluctuate and stabilize.
- October to December see HPD Agency running again to solve issues which keep increasing.
- HPD Agency has been resolving complaints from 311 callers of between 300 to 3000 in a day over the years.
- The year 2013 saw actions taken over complaints rising to 5000 in a day.
- Brooklyn has the highest population and as established; population is positively related to complaints, thus, there are more complaints from Brooklyn compared to other Boroughs. However, Queens being second largest population wise; it lead in:

- Unsanitary conditions
- Paint-Plaster
- Heat or hot water
- Electric complaints conditions.
- Staten Island has the lowest population and her complaints too to HPD Agency were not as many as for the rest of Boroughs.
- The following were least complained about:
- Water leak
- Unsanitary condition
- Paint-plaster
- General
- Door or window
- Flooring or stairs
- Manhattan Borough topped in the complaints of water leak, door or window and appliances.

Accident data set exporatory analysis

```
col <- read.csv("nyc_accidents.csv",stringsAsFactors = F,header=T)
str(col)
```

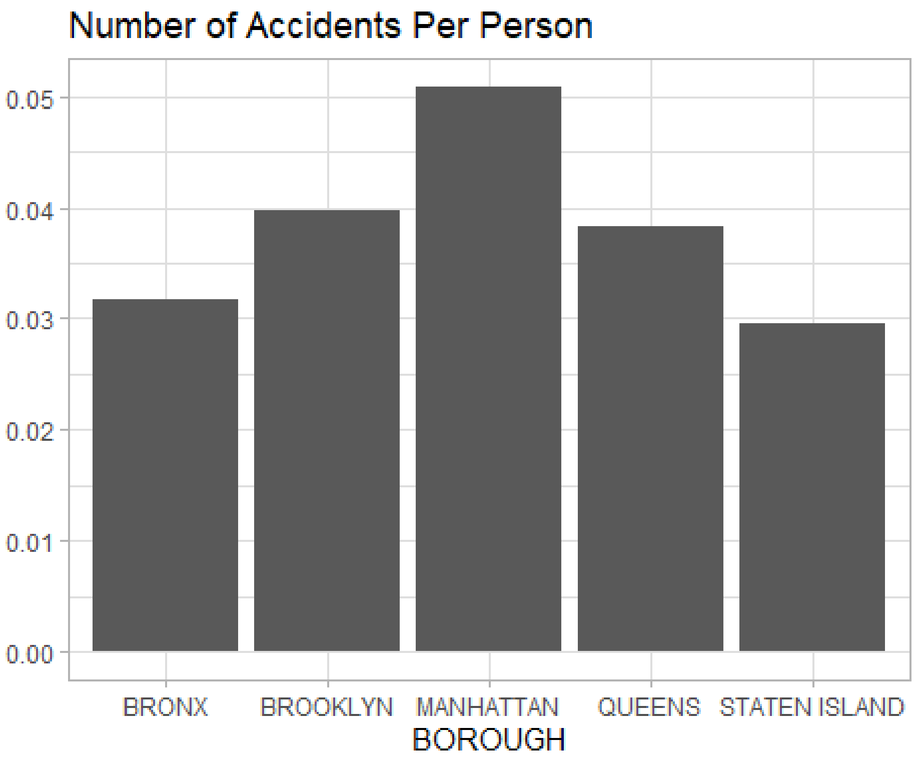
```
## 'data.frame':    477732 obs. of  29 variables:
##  $ UNIQUE.KEY         : int  3146911 3146180 3146384 3146013 3146120
3145968 3146102 3146344 3145900 3145960 ...
##  $ DATE               : chr  "01/01/2015" "01/01/2015" "01/01/2015"
"01/01/2015" ...
##  $ TIME               : chr  "0:20" "0:20" "0:21" "0:30" ...
##  $ BOROUGH            : chr  "QUEENS" "" "BROOKLYN" "BROOKLYN" ...
##  $ ZIP.CODE           : int  11358 NA 11205 11213 NA 11203 11105 11203
10024 11223 ...
##  $ LATITUDE           : num  40.8 40.8 40.7 40.7 NA ...
##  $ LONGITUDE          : num  -73.8 -73.9 -74 -73.9 NA ...
##  $ LOCATION           : chr  "(40.7518471, -73.787862)" "(40.7712888, -
73.9466928)" "(40.6894449, -73.9551212)" "(40.6738445, -73.9250801)" ...
##  $ ON.STREET.NAME     : chr  "47 AVENUE" "" "BEDFORD AVENUE" "BUFFALO
AVENUE" ...
##  $ CROSS.STREET.NAME  : chr  "193 STREET" "" "LAFAYETTE AVENUE" "SAINT
MARKS AVENUE" ...
##  $ OFF.STREET.NAME    : chr  "" "" "" "" ...
##  $ PERSONS.INJURED    : int  0 1 0 0 0 2 0 0 1 0 ...
##  $ PERSONS.KILLED     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PEDESTRIANS.INJURED: int  0 0 0 0 0 0 0 0 1 0 ...
##  $ PEDESTRIANS.KILLED : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CYCLISTS.INJURED   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CYCLISTS.KILLED    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOTORISTS.INJURED  : int  0 1 0 0 0 2 0 0 0 0 ...
##  $ MOTORISTS.KILLED   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ VEHICLE.1.TYPE     : chr  "SPORT UTILITY/STATION WAGON" "PASSENGER
VEHICLE" "PASSENGER VEHICLE" "BUS" ...
##  $ VEHICLE.2.TYPE     : chr  "" "" "UNKNOWN" "PASSENGER VEHICLE" ...
##  $ VEHICLE.3.TYPE     : chr  "" "" "" "" ...
##  $ VEHICLE.4.TYPE     : chr  "" "" "" "" ...
##  $ VEHICLE.5.TYPE     : chr  "" "" "" "" ...
##  $ VEHICLE.1.FACTOR   : chr  "TRAFFIC CONTROL DISREGARDED" "ANIMALS
ACTION" "FATIGUED/DROWSY" "LOST CONSCIOUSNESS" ...
##  $ VEHICLE.2.FACTOR   : chr  "" "" "UNSPECIFIED" "" ...
##  $ VEHICLE.3.FACTOR   : chr  "" "" "" "" ...
##  $ VEHICLE.4.FACTOR   : chr  "" "" "" "" ...
##  $ VEHICLE.5.FACTOR   : chr  "" "" "" "" ...
```

```
col$DATE_TIME <- paste(col$DATE,col$TIME)
col$DATE <- mdy(col$DATE)
col$DATE_TIME <-mdy_hm(col$DATE_TIME)
col$day <- wday(col$DATE_TIME,label = T)
col$month <- month(col$DATE_TIME,label = T)
col$hour <- hour(col$DATE_TIME)
```

Number of Accidents

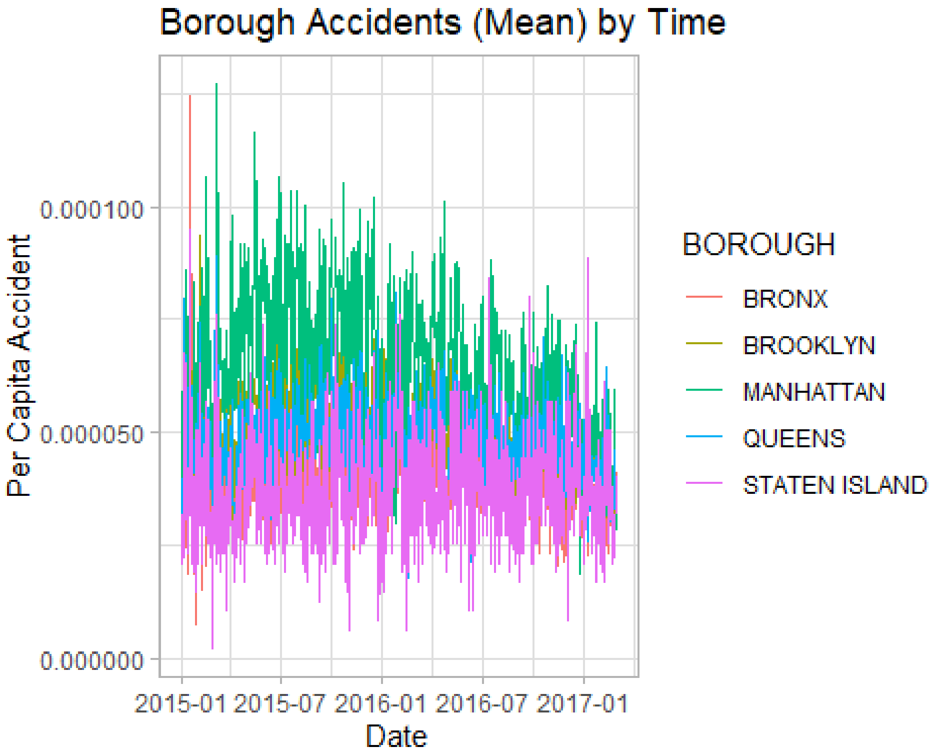
Number of Accidents Per Person

```
temp_1<- col %>% group_by(BOROUGH) %>% dplyr::summarise(n=n())
temp_1$pop <- rep(0,dim(temp_1)[1])
temp_1$pop <- ifelse(temp_1$BOROUGH=="MANHATTAN",1644158,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="BRONX",1455444,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="QUEENS",2339150,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="STATEN ISLAND",474558,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="BROOKLYN",2636735,temp_1$pop)
temp_1$per_cap <- temp_1$n/temp_1$pop

temp_1 %>%
  filter(BOROUGH!="") %>%

ggplot(aes(x=BOROUGH,y=per_cap))+geom_bar(stat="identity")+labs(title="Number
of Accidents Per Person",y="")
```



Number of Collisions by Day and Hour and Area

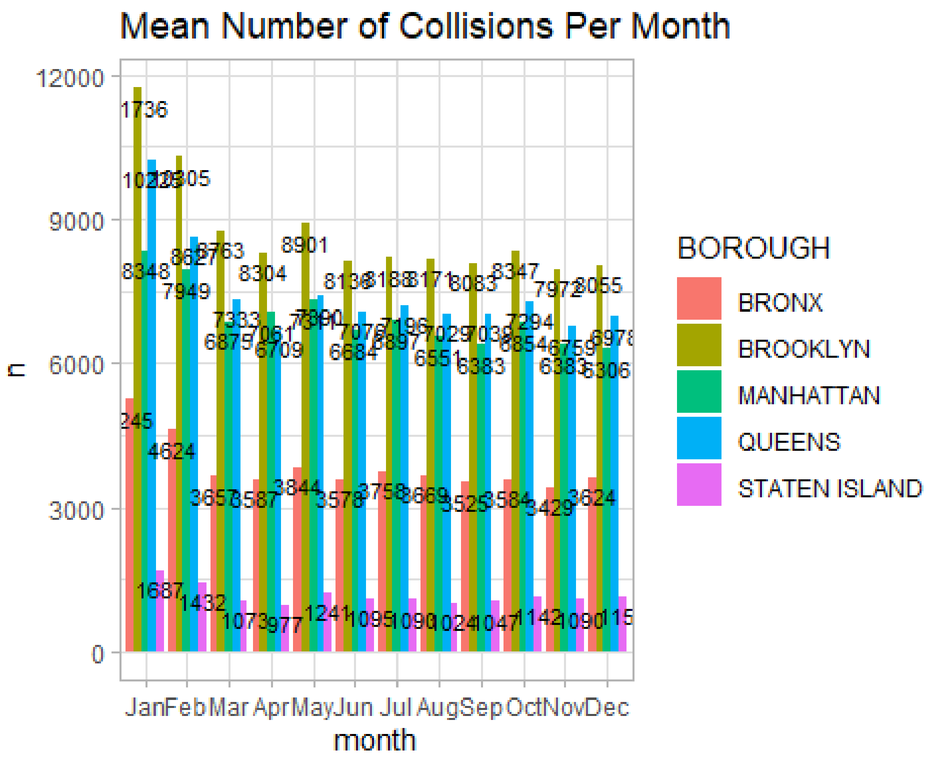
```
col %>%
  filter(BOROUGH!="") %>%
  group_by(DATE,BOROUGH) %>%
  dplyr::summarise(n=mean(n())) %>%
  na.omit() %>%
  left_join(temp_1,by="BOROUGH") %>%
  mutate(per_cap=n.x/pop) %>%
  ggplot(aes(x=DATE, y=per_cap, colour=BOROUGH, group=BOROUGH)) +
  geom_line()+
  scale_fill_hc("darkunica") +ggtitle("Borough Accidents (Mean) by Time")+
  scale_y_continuous(labels=scales::comma_format())+
  labs(y="Per Capita Accident",
       x="Date")
```



The lowest number of accidents occured in Staten Island. This borough also has the least ratio of accidents to persons. The most populous boroughs, Brooklyn, Manhattan, and Queens, also have the highest number of accidents at any time of day.

Mean Number of Collisions per Month

```
col %>% filter(BOROUGH!="") %>% group_by(month,BOROUGH) %>%
dplyr::summarise(n=mean(n())) %>% na.omit() %>%
  ggplot(aes(x=month, y=n, fill=BOROUGH)) +
geom_bar(position="dodge",stat = "identity")+geom_text(aes(label=n),
vjust=1.5, colour="black",
position=position_dodge(.9), size=3)+ggtitle("Mean Number of Collisions Per
Month")
```

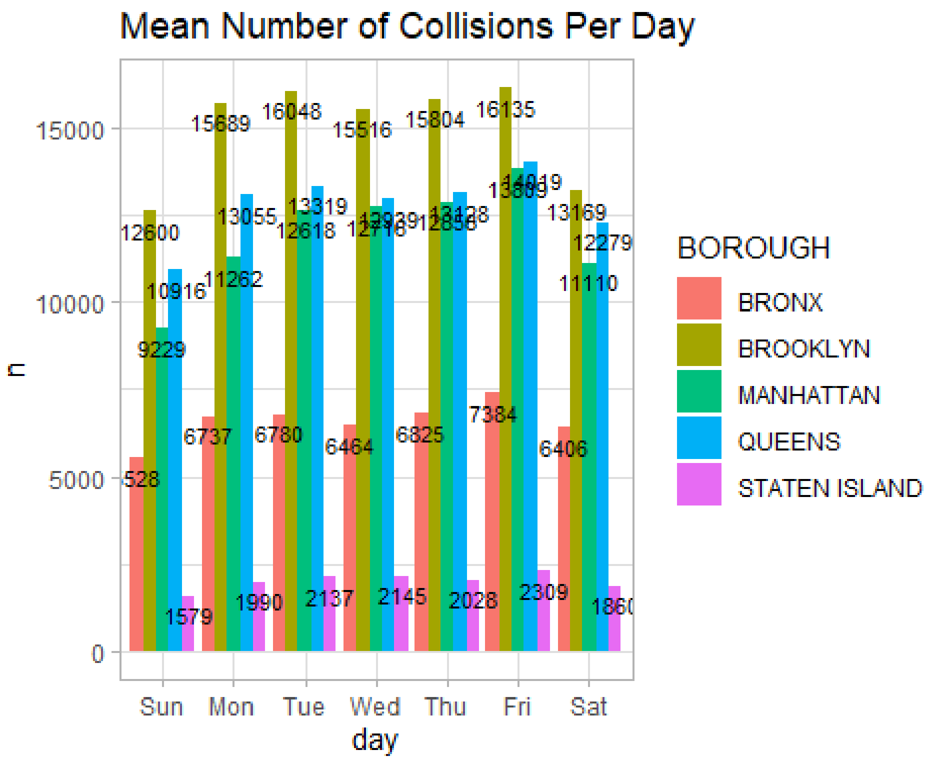


Brooklyn records the highest number of collisions in each month throught the year. January has the highest number of collisions in all the five boroughs.

Which Day had the highest mean number of Accidents?

Do weekends have higher accidents than week days?

```
col %>% group_by(BOROUGH,day)%>% dplyr::summarise(n=mean(n())) %>%
filter(BOROUGH!="") %>%
  ggplot(aes(x=day, y=n, fill=BOROUGH)) +
geom_bar(position="dodge",stat = "identity")+geom_text(aes(label=n),
vjust=1.5, colour="black",
position=position_dodge(.9), size=3)+ggtitle("Mean Number of Collisions Per
Day")
```


Mean Number of Collisions Per Day
n
15000
10000
5000
0
Sun
Mon
Tue
Wed
Thu
Fri
Sat
day
BOROUGH
BRONX
BROOKLYN
MANHATTAN
QUEENS
STATEN ISLAND
528
12600
9229
10916
1579
6737
15689
11262
13055
1990
6780
16048
12618
13319
2137
6464
15516
12718
12939
2145
6825
15804
12856
13128
2028
7384
16135
13809
14019
2309
6406
13169
11110
12279
1860

APPENDICES

NYC311 DATA DICTIONARY

- Unique.Key is the unique identify a Service Requester (SR) is given
- Agency is the Acronym of the government agency concerned with resolving the complaint
- Agency.Name is the full name of the Agency
- Complaint.Type is the first level hierachy of identifying the problem
- Descriptor is the second level hierachy of describing the problem. It is directly related to the Complaint.Type
- Location Type describes the location used in the address information
- Incident Zip is the incident location zip code provided by geo validation
- Incident Address is the address the SR provides
- Street Name is the street name of the incident submited by the SR
- Cross Street 1 is the first cross street based on the geo validated incident location
- Cross Street 2 is the second cross street based on the geo validated incident location
- Intersection Street 1 is the first intersection street based on the geo validated incident location
- Intersection Street 2 is the second intersection street based on the geo validated incident location
- Address Type is the type of address provided by the SR
- City is where the incident has been reported to happen
- Landmark is the nearest location to the incident location
- Facility Type is the kind of facility where the incident is reported to be by the SR
- Status is the state of the action that has been taken upon submitting a SR
- Due Date is when the SR is supposed to have been attended
- Resolution Action Updated Date is the date when action has been taken on resolving SR
- Borough is the city where the incident has been reported to have happened
- School Name is the name of the school with the incident

- School Number the number of of the school with the incident
- School Region is the region of the school with the incident
- School Code is the code of the school with the incident
- School Phone Number this is the number to call of the school with the incident
- School Address is the physical address of the school with the incident
- School City is where the school is found
- School State is the state where the school with the incident is found
- School or Citywide Complaint describes whether the complaint involves a school or the whole city
- Vehicle Type the kind of the vehicle with the incident

*Taxi Company Borough is the city where taxi with the incident operates from

- Taxi Pick Up Location is the specific location where the taxi picks client
- Bridge Highway Name is the bridge name along the highway
- Latitude is the physical location of the incident
- Longitude Location is the physical location of the incident

SECOND DATASET DATA DICTIONARY

- FIPS Code is a Text Data: Federal Information Processing Standards (FIPS) codes, issued by the National Institute of Standards and Technology (NIST), that identify each geographic area.
- Geography is a Text Data which indicates the Geographic Area Name.
- Year is a Numeric Data which indicates the year for which the population is calculated.

*Program Type is a Text Data which indicates Census Base Population-

The population count or estimate used as the starting point in the estimates process. It can be the most recent updated Census count or the estimate for a previous date within the same vintage. The April 1 estimates base population may differ from the April 1 Census count due to legal boundary updates, other geographic program changes, and Count Question Resolution actions. Intercensal- Population Estimates Program (PEP) estimate of the population between two completed decennial censuses. Postcensal- Population Estimates Program (PEP) estimate of the population following the most recent decennial census.

The latest vintage of data available supersedes all previously produced estimates for those dates.

CODE

```
knitr::opts_chunk$set(echo=TRUE, warning=FALSE, message=FALSE)

library(tidyverse)
library(janitor)
library(Amelia)
library(skimr)
library(lubridate)
library(tidytext)
library(wordcloud)
library(jsonlite)
library(plotly)
library(highcharter)
library(lubridate)
library(ggthemes)
library(viridis)
library(leaflet)
library(ggmap)

theme_set(theme_light())

nyc311<-
data.table::fread("311_Service_Requests_from_2010_to_Present.csv",nrows=10000
00) %>%
  clean_names()


# Convert blank values to NA
nyc311[nyc311==""]<-NA

# Converting the date & time to required format
nyc311 <- nyc311 %>%
  mutate(closed_date = mdy_hms(closed_date),
         created_date = mdy_hms(created_date),
         resolution_action_updated_date =
mdy_hms(resolution_action_updated_date),
         due_date = mdy_hms(due_date),
         wday = wday(created_date, label = TRUE),
         month = month(created_date, label = TRUE),
         year = year(created_date),
         timetaken = (created_date - closed_date),
         resolvetime = (due_date - resolution_action_updated_date))
nyc311 %>%
  filter(complaint_type=="Street Condition") %>%
  count(borough,complaint_type) %>%
ggplot(aes(x = borough,y=n)) +
  geom_point(aes(color = borough))  +
  geom_text(aes(label=n),hjust=1,vjust=1)+
  theme_light()+
```

```r
  ggtitle("Street Condition Complaints by Borough ") +
  xlab("Borough") + ylab("Number of Complaints")  +
  labs(caption = "(Based on latest version of NYC311 Data Set)") +
scale_y_continuous(labels = scales::comma)

data.frame(n=colSums(is.na(nyc311))) %>%
  arrange(desc(n)) %>%
  rownames_to_column() %>%
  mutate(per_missing=n/nrow(nyc311)*100,
         rowname=fct_reorder(rowname,per_missing)) %>%
  ggplot(aes(x=per_missing,y=rowname,fill=rowname))+
  geom_col(show.legend = FALSE)+
  labs(title="Percentage of rows missing per column",
       x="Percentage Missing",
       y="")+
  theme(axis.text.y=element_text(size=5))
nyc311_sub <- nyc311 %>%
  select(created_date,closed_date,agency,agency_name,complaint_type,

incident_zip,status,due_date,resolution_action_updated_date,borough,latitude,
longitude,location,
                                landmark, park_borough,
school_city,school_state,school_zip,location_type,community_board,

incident_address,taxi_company_borough,descriptor,
         resolution_action_updated_date,due_date ,
wday,month,year,timetaken,resolvetime) %>%
  ungroup()
summary(nyc311_sub)


skim(nyc311_sub)
nyc311_sub %>%
  group_by(wday) %>%
  dplyr::summarise(count=n()) %>%
  ggplot(mapping = aes(x=wday,y=count))+
  geom_col(aes(fill = wday),show.legend = FALSE) +
  labs(title="Calls distribution in a day",x="wday",y="Complaint Type")+
  scale_y_continuous(labels=scales::comma_format())
nyc311_sub %>%
  group_by(location_type) %>%
  dplyr::summarise(count=n()) %>%
  arrange(desc(count)) %>%
  na.omit() %>%
  head(10) %>%
  mutate(location_type=fct_reorder(location_type,count)) %>%
  ggplot(mapping = aes(y=location_type,x=count))+
  geom_col(aes(fill = location_type),show.legend = FALSE) +
  labs(title="Top Locations",x="Type",y="Count")+
  scale_x_continuous(labels=scales::comma_format())
```

```r
top10 <- nyc311_sub %>%
  group_by(complaint_type) %>%
  summarize(count=n()) %>%
  slice_max(count,n=10) %>%
  mutate(complaint_type=fct_reorder(complaint_type,count))

ggplot(top10,aes(y=complaint_type,x=count)) +
  geom_bar(stat="identity",aes(fill=complaint_type),show.legend = FALSE) +
  labs(title="Top 10 complaints in a day",x="Count of
complaints",y="Complaint Type")+
  scale_x_continuous(labels=scales::comma_format())


nyc311_sub %>%
  filter(complaint_type %in% (top10 %>% pull(complaint_type))) %>%
  group_by(borough,complaint_type) %>%
  dplyr::summarise(count=n()) %>%
  ungroup() %>%
  filter(borough!="Unspecified") %>%
  mutate(borough=fct_reorder(borough,count)) %>%
ggplot(aes(x=count, y=borough, fill=complaint_type)) +
geom_col(show.legend = TRUE) +
labs(title="Top Complaints by Type and Borough",
     x="Count of Complaint",
     y="Borough",
     fill="Complaint Type")+
  scale_x_continuous(labels=scales::comma_format())

nyc311_sub %>%
  filter(complaint_type %in% (top10 %>% pull(complaint_type))) %>%
  ggplot(aes(x=complaint_type))+
  geom_density((aes(fill=complaint_type)))+
  facet_wrap(~complaint_type,scales="free_y")+
  labs(title="Top 10 Complaint Type by density",
       fill="Complaint Type",
       y="",
       x="Density")+
  theme(legend.position = "none",
        axis.text.x = element_blank())


nyc311_sub %>%
  group_by(year) %>%
  dplyr::summarise(count=n()) %>%
  ggplot(aes(x=as.factor(year), y=count)) +
  geom_point(size=8,aes(group=year))+
  labs(title="Complaint trend",
     x="Count of Complaint",
     y="")

nyc311_sub %>%
  filter(resolvetime>0) %>%
  ggplot(aes(x=resolvetime/(60*60*24))) +
  geom_histogram()+
   labs(title="Complaint resolution TAT",
     x="Number of Days",
     y="Count of Complaint")+
  scale_y_continuous(labels = scales::comma_format())


ds_heat <- nyc311_sub %>%
  mutate(complaint_type=tolower(complaint_type)) %>%
  filter(str_detect(complaint_type,"heat")) %>%
  group_by(borough) %>%
  dplyr::summarise(Count = n()) %>%
  ungroup() %>%
  mutate(borough = fct_reorder(borough,Count))

ds_heat %>%
  ggplot(aes(y = borough,x = Count)) +
  geom_col(aes(fill=borough),show.legend = FALSE) +
  labs(title="Borough with higest number of Heating complaints",
     x= "",
     y="Count of Complaint")+
  scale_x_continuous(labels=scales::comma_format())



ds_street <-  nyc311_sub %>%
  mutate(complaint_type=tolower(complaint_type)) %>%
  filter(str_detect(complaint_type,"street")) %>%
  group_by(borough) %>%
  dplyr::summarise(Count = n()) %>%
  ungroup() %>%
  mutate(borough = fct_reorder(borough,Count)) %>%
  filter(borough!="Unspecified")

ds_street %>%
  ggplot(aes(y = borough,x = Count)) +
  geom_col(aes(fill=borough),show.legend = FALSE) +
  labs(title="Borough with higest number of Street condition complaints",
     x= "",
     y="Count of Complaint")+
  scale_x_continuous(labels=scales::comma_format())
```

```
words <- nyc311_sub %>%
  unnest_tokens(word,descriptor) %>%
  count(word)


wordcloud(words = words$word, freq = words$n, min.freq = 1,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
nyc311_sub %>%
  unnest_tokens(word,descriptor) %>%
  count(borough,word) %>%
  group_by(borough) %>%
  slice_max(n,n=10) %>%
  mutate(word=fct_reorder(word,n)) %>%
  ggplot(aes(x=n,y=word,fill=borough))+
  geom_col(show.legend = FALSE)+
  facet_wrap(~borough,scale="free_y")+
  labs(title="Top words used in complaints of Boroughs",
       x="Count",
       y="Word")

nycp<-fromJSON("https://data.ny.gov/resource/krt9-ym2k.json")

str(nycp)

nycp <- nycp %>%
  rename("borough"="geography")

summary(nycp)
skim(nycp)

nycp_sub <- nycp %>%
  filter(program_type=="Postcensal Population Estimate") %>%
 mutate(borough = case_when(str_detect(borough,"Bronx County")~"BRONX",
                            str_detect(borough,"Kings County")~"BROOKLYN",
                            str_detect(borough,"New York
County")~"MANHATTAN",
                            str_detect(borough,"Richmond County")~"STATEN
ISLAND",
                            str_detect(borough,"Queens County")~"QUEENS",
                            TRUE ~ borough)) %>%
  select(borough, year, population) %>%
  mutate(year=as.numeric(year)) %>%
  filter(borough %in% c("BROOKLYN", "BRONX", "MANHATTAN", "STATEN ISLAND",
"QUEENS")) %>%
  filter(year %in% c(2010,2011, 2012,2013,2014,2015))
```

```
nyc311_fil <- nyc311_sub %>%
  group_by(agency, borough, complaint_type, year) %>%
  summarize(count=n())


nyc_join <- nyc311_fil %>%
 left_join(nycp_sub)


nyc_join %>%
#  ungroup() %>%
  filter(year==max(year),
         borough!="Unspecified") %>%
  mutate(population=as.numeric(population),
    borough=fct_reorder(borough,population)) %>%
ggplot(aes(fill=borough, x=as.numeric(population), y=borough)) +
geom_bar(stat="identity",show.legend = FALSE) +
     theme(plot.title = element_text(hjust = 0.9)) +
  scale_x_continuous(labels=scales::comma_format())+
    labs(title = "Population By Borough",
       x = "Borough Names",
       y = "Population",
       caption = "based on data from NYC 311",
       fill="Borough") +
    scale_x_continuous(labels = scales::unit_format(unit = "K", scale = 1e-
6))


pop_2015<- nyc_join %>%
  filter(year %in% c(2010:2015))

pop_2015 %>%
  mutate(population=as.numeric(population)) %>%
  group_by(borough,year) %>%
  dplyr::summarise(pop=mean(population,na.rm=TRUE)) %>%
  ggplot(aes(fill=borough, y=pop, x=as.factor(year))) +
    geom_col() +
    labs(title = "Relationship between borough and population",  x= "",  y=
"Population",caption = "based on data from NYC 311") +
   scale_y_continuous(labels=scales::comma_format())


complaintlocs1 <- nyc311_sub %>%
  select(
    complaint_type,
    longitude,
    latitude
  )
noisecompl <- complaintlocs1 %>%
```

```r
  filter(str_detect(complaint_type,"Noise"))

count_complaint <- nyc311_sub %>%
  group_by(complaint_type,borough) %>%
  add_count() %>%ungroup() %>%
  select(borough,complaint_type,longitude,latitude,n) %>%
  distinct(borough,complaint_type,n,longitude,latitude) %>%
  filter(complaint_type %in% (top10 %>% pull(complaint_type))) %>%
  group_by(borough, complaint_type) %>%
  summarise(longitude=mean(longitude,na.rm=TRUE),
            latitude=mean(latitude,na.rm=TRUE),
            count=sum(n))

library(rworldmap)

newmap <- getMap(resolution = "high")
nyc_coorflimits <- data.frame( long = c(-74.5, -73.5), lat = c(40.5, 41),
stringsAsFactors = FALSE)

nyc <- ggplot() + geom_polygon(data = newmap, aes(x=long, y = lat, group =
group), fill = "gray", color = "blue",)  + xlim(-74.5, -73.5) + ylim(40.5,
41)

nyc +
  geom_point(data=count_complaint, aes(longitude, latitude, size=count),
colour="blue",alpha=0.4)  +
  facet_wrap(~complaint_type, scales = "free") +
  labs(x = "Longitude", y = "Latitude", title = "Highest Number of request by
Complaint Type",color="Number of requests")


lat <- c(40.5,40.9)
long <- c(-73.8,-74.2)
bbox <- make_bbox(long,lat,f=0.05)
nyc_map <- get_map(bbox,maptype="toner-lite",source="stamen")

 map <- ggmap(nyc_map) +
  geom_point(
    data = noisecompl, aes(x = longitude, y = latitude),
    size = 0.1, alpha = 0.1, color = "blue"
  ) +
  ggtitle("Map of the noise in the neighbourhood") +
  theme(plot.title = element_text(hjust = 0.5))
map

col <- read.csv("nyc_accidents.csv",stringsAsFactors = F,header=T)
str(col)
col$DATE_TIME <- paste(col$DATE,col$TIME)
col$DATE <- mdy(col$DATE)
```

```r
col$DATE_TIME <-mdy_hm(col$DATE_TIME)
col$day <- wday(col$DATE_TIME,label = T)
col$month <- month(col$DATE_TIME,label = T)
col$hour <- hour(col$DATE_TIME)

temp_1<- col %>% group_by(BOROUGH) %>% dplyr::summarise(n=n())
temp_1$pop <- rep(0,dim(temp_1)[1])
temp_1$pop <- ifelse(temp_1$BOROUGH=="MANHATTAN",1644158,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="BRONX",1455444,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="QUEENS",2339150,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="STATEN ISLAND",474558,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="BROOKLYN",2636735,temp_1$pop)
temp_1$per_cap <- temp_1$n/temp_1$pop

temp_1 %>%
  filter(BOROUGH!="") %>%

ggplot(aes(x=BOROUGH,y=per_cap))+geom_bar(stat="identity")+labs(title="Number
of Accidents Per Person",y="")

col %>%
  filter(BOROUGH!="") %>%
  group_by(DATE,BOROUGH) %>%
  dplyr::summarise(n=mean(n())) %>%
  na.omit() %>%
  left_join(temp_1,by="BOROUGH") %>%
  mutate(per_cap=n.x/pop) %>%
  ggplot(aes(x=DATE, y=per_cap, colour=BOROUGH, group=BOROUGH)) +
  geom_line()+
  scale_fill_hc("darkunica") +ggtitle("Borough Accidents (Mean) by Time")+
  scale_y_continuous(labels=scales::comma_format())+
  labs(y="Per Capita Accident",
       x="Date")


col %>% filter(BOROUGH!="") %>% group_by(month,BOROUGH) %>%
dplyr::summarise(n=mean(n())) %>% na.omit() %>%
  ggplot(aes(x=month, y=n, fill=BOROUGH)) +
geom_bar(position="dodge",stat = "identity")+geom_text(aes(label=n),
vjust=1.5, colour="black",
position=position_dodge(.9), size=3)+ggtitle("Mean Number of Collisions Per
Month")

col %>% group_by(BOROUGH,day)%>% dplyr::summarise(n=mean(n())) %>%
filter(BOROUGH!="") %>%
  ggplot(aes(x=day, y=n, fill=BOROUGH)) +
geom_bar(position="dodge",stat = "identity")+geom_text(aes(label=n),
vjust=1.5, colour="black",
position=position_dodge(.9), size=3)+ggtitle("Mean Number of Collisions Per
```

```
Day")




knitr::opts_chunk$set(echo=TRUE, warning=FALSE, message=FALSE)

library(tidyverse)
library(janitor)
library(Amelia)
library(skimr)
library(lubridate)
library(tidytext)
library(wordcloud)
library(jsonlite)
library(plotly)
library(highcharter)
library(lubridate)
library(ggthemes)
library(viridis)
library(leaflet)
library(ggmap)

theme_set(theme_light())
nyc311<-
data.table::fread("311_Service_Requests_from_2010_to_Present.csv",nrows=10000
00) %>%
  clean_names()
# Convert blank values to NA

nyc311[nyc311==""]<-NA

# Converting the date & time to required format
nyc311 <- nyc311 %>%
  mutate(closed_date = mdy_hms(closed_date),
         created_date = mdy_hms(created_date),
         resolution_action_updated_date =
mdy_hms(resolution_action_updated_date),
         due_date = mdy_hms(due_date),
         wday = wday(created_date, label = TRUE),
         month = month(created_date, label = TRUE),
         year = year(created_date),
         timetaken = (created_date - closed_date),
         resolvetime = (due_date - resolution_action_updated_date))

data.frame(n=colSums(is.na(nyc311))) %>%
  arrange(desc(n)) %>%
  rownames_to_column() %>%
  mutate(per_missing=n/nrow(nyc311)*100,
         rowname=fct_reorder(rowname,per_missing)) %>%
  ggplot(aes(x=per_missing,y=rowname,fill=rowname))+
  geom_col(show.legend = FALSE)+
  labs(title="Percentage of rows missing per column",
       x="Percentage Missing",
       y="")

nyc311_sub <- nyc311 %>%
  select(created_date,closed_date,agency,agency_name,complaint_type,

incident_zip,status,due_date,resolution_action_updated_date,borough,latitude,
longitude,location,
                                landmark, park_borough,
school_city,school_state,school_zip,location_type,community_board,

incident_address,taxi_company_borough,descriptor,
         resolution_action_updated_date,due_date ,
wday,month,year,timetaken,resolvetime) %>%
  ungroup()

summary(nyc311_sub)


skim(nyc311_sub)

nyc311_sub %>%
  group_by(wday) %>%
  dplyr::summarise(count=n()) %>%
  ggplot(mapping = aes(x=wday,y=count))+
  geom_col(aes(fill = wday),show.legend = FALSE) +
  labs(title="Calls distribution in a day",x="wday",y="Complaint Type")

nyc311_sub %>%
  group_by(location_type) %>%
  dplyr::summarise(count=n()) %>%
  arrange(desc(count)) %>%
  na.omit() %>%
  head(10) %>%
  mutate(location_type=fct_reorder(location_type,count)) %>%
  ggplot(mapping = aes(y=location_type,x=count))+
  geom_col(aes(fill = location_type),show.legend = FALSE) +
  labs(title="Top Locations",x="Type",y="Count")

top10 <- nyc311_sub %>%
  group_by(complaint_type) %>%
  summarize(count=n()) %>%
  slice_max(count,n=10) %>%
  mutate(complaint_type=fct_reorder(complaint_type,count))
```

```
ggplot(top10,aes(y=complaint_type,x=count)) +
  geom_bar(stat="identity",aes(fill=complaint_type),show.legend = FALSE) +
  labs(title="Top 10 complaints in a day",x="Count of
complaints",y="Complaint Type")

nyc311_sub %>%
  filter(complaint_type %in% (top10 %>% pull(complaint_type))) %>%
  group_by(borough,complaint_type) %>%
  dplyr::summarise(count=n()) %>%
  ungroup() %>%
  mutate(borough=fct_reorder(borough,count)) %>%
ggplot(aes(x=count, y=borough, fill=complaint_type)) +
geom_col(show.legend = TRUE) +
labs(title="Top Complaints by Type and Borough",
     x="Count of Complaint",
     y="Borough",
     fill="Complaint Type")

nyc311_sub %>%
  group_by(year) %>%
  dplyr::summarise(count=n()) %>%
  ggplot(aes(x=as.factor(year), y=count)) +
  geom_point(size=8,aes(group=year))+
  labs(title="Complaint trend",
     x="Count of Complaint",
     y="")

nyc311_sub %>%
  ggplot(aes(x=resolvetime/(60*60*24))) +
  geom_histogram()+
   labs(title="Complaint resolution TAT",
     x="Number of Days",
     y="Count of Complaint")+
  scale_x_continuous(labels = scales::comma_format())

ds_heat <- nyc311_sub %>%
  mutate(complaint_type=tolower(complaint_type)) %>%
  filter(str_detect(complaint_type,"heat")) %>%
  group_by(borough) %>%
  dplyr::summarise(Count = n()) %>%
  ungroup() %>%
  mutate(borough = fct_reorder(borough,Count))

ds_heat %>%
  ggplot(aes(y = borough,x = Count)) +
  geom_col(aes(fill=borough),show.legend = FALSE) +
  labs(title="Borough with higest number of Heating complaints",
     x= "",
     y="Count of Complaint")

words <- nyc311_sub %>%
  unnest_tokens(word,descriptor) %>%
  count(word)


wordcloud(words = words$word, freq = words$n, min.freq = 1,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))


nyc311_sub %>%
  unnest_tokens(word,descriptor) %>%
  count(borough,word) %>%
  group_by(borough) %>%
  slice_max(n,n=10) %>%
  mutate(word=fct_reorder(word,n)) %>%
  ggplot(aes(x=n,y=word,fill=borough))+
  geom_col(show.legend = FALSE)+
  facet_wrap(~borough,scale="free_y")+
  labs(title="Top words used in complaints of Boroughs",
       x="Count",
       y="Word")

nycp<-fromJSON("https://data.ny.gov/resource/krt9-ym2k.json")

str(nycp)

nycp <- nycp %>%
  rename("borough"="geography")

summary(nycp)

skim(nycp)

nycp_sub <- nycp %>%
  filter(program_type=="Postcensal Population Estimate") %>%
 mutate(borough = case_when(str_detect(borough,"Bronx County")~"BRONX",
                            str_detect(borough,"New York County")~"BROOKLYN",
                            str_detect(borough,"Richmond
County")~"MANHATTAN",
                            str_detect(borough,"Queens County")~"QUEENS",
                            TRUE ~ borough)) %>%
  select(borough, year, population) %>%
  mutate(year=as.numeric(year)) %>%
  filter(borough %in% c("BROOKLYN", "BRONX", "MANHATTAN", "STATEN ISLAND",
"QUEENS")) %>%
  filter(year %in% c(2010,2011, 2012,2013,2014,2015))
```

```
nyc311_fil <- nyc311_sub %>%
  group_by(agency, borough, complaint_type, year) %>%
  summarize(count=n()) %>%
  filter(count>50)

nyc_join <- nyc311_fil %>%
 left_join(nycp_sub)

nyc_join %>%
  ungroup() %>%
  filter(year==max(year),
         borough!="Unspecified") %>%
  mutate(population=as.numeric(population),
    borough=fct_reorder(borough,population)) %>%
ggplot(aes(fill=borough, x=as.numeric(population), y=borough)) +
geom_bar(stat="identity",show.legend = FALSE) +
     theme(plot.title = element_text(hjust = 0.9)) +
  scale_x_continuous(labels=scales::comma_format())+
    labs(title = "Population By Borough",
       x = "Borough Names",
       y = "Population",
       caption = "based on data from NYC 311",
       fill="Borough")

pop_2015<- nyc_join %>%
  filter(year %in% c(2010:2015))

pop_2015 %>%
  mutate(population=as.numeric(population)) %>%
  group_by(borough,year) %>%
  dplyr::summarise(pop=mean(population,na.rm=TRUE)) %>%
  ggplot(aes(fill=borough, y=pop, x=as.factor(year))) +
    geom_col() +
    labs(title = "Relationship between borough and population",  x= "",  y=
"Population",caption = "based on data from NYC 311") +
   scale_y_continuous(labels=scales::comma_format())

complaintlocs1 <- nyc311_sub %>%
  select(
    complaint_type,
    longitude,
    latitude
  )
noisecompl <- complaintlocs1 %>%
  filter(str_detect(complaint_type,"Noise"))

count_complaint <- nyc311_sub %>%
```

```r
  group_by(complaint_type,borough) %>%
  add_count() %>%ungroup() %>%
  select(borough,complaint_type,longitude,latitude,n) %>%
  distinct(borough,complaint_type,n,longitude,latitude) %>%
  filter(complaint_type %in% (top10 %>% pull(complaint_type))) %>%
  group_by(borough, complaint_type) %>%
  summarise(longitude=mean(longitude,na.rm=TRUE),
            latitude=mean(latitude,na.rm=TRUE),
            count=sum(n))

library(rworldmap)

newmap <- getMap(resolution = "high")
nyc_coorflimits <- data.frame( long = c(-74.5, -73.5), lat = c(40.5, 41),
stringsAsFactors = FALSE)

nyc <- ggplot() + geom_polygon(data = newmap, aes(x=long, y = lat, group =
group), fill = "gray", color = "blue")  + xlim(-74.5, -73.5) + ylim(40.5, 41)

nyc +
  geom_point(data=count_complaint, aes(longitude, latitude, size=count),
colour="blue")  +
  facet_wrap(~complaint_type, scales = "free") +
  labs(x = "Longitude", y = "Latitude", title = "Highest Number of request by
Complaint Type",color="Number of requests")




lat <- c(40.5,40.9)
long <- c(-73.8,-74.2)
bbox <- make_bbox(long,lat,f=0.05)
nyc_map <- get_map(bbox,maptype="toner-lite",source="stamen")

 map <- ggmap(nyc_map) +
  geom_point(
    data = noisecompl, aes(x = longitude, y = latitude),
    size = 0.6, alpha = 0.2, color = "red"
  ) +
  ggtitle("Noise Map") +
  theme(plot.title = element_text(hjust = 0.5)) +
  xlab("Longitude") + ylab("Latitude")
map

webshot::install_phantomjs()

leaflet(data = noisecompl) %>%   # The function leaflet() returns a Leaflet
map widget
  addTiles() %>%
  setView(-73.9, 40.6, zoom =10) %>%
  addPopups(-73.9, 40.9, 'Noise in NY') %>%
  addMarkers(
  clusterOptions = markerClusterOptions())

col <- read.csv("nyc_accidents.csv",stringsAsFactors = F,header=T)
str(col)
col$DATE_TIME <- paste(col$DATE,col$TIME)
col$DATE <- mdy(col$DATE)
col$DATE_TIME <-mdy_hm(col$DATE_TIME)
col$day <- wday(col$DATE_TIME,label = T)
col$month <- month(col$DATE_TIME,label = T)
col$hour <- hour(col$DATE_TIME)

temp_1<- col %>% group_by(BOROUGH) %>% dplyr::summarise(n=n())
temp_1$pop <- rep(0,dim(temp_1)[1])
temp_1$pop <- ifelse(temp_1$BOROUGH=="MANHATTAN",1644158,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="BRONX",1455444,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="QUEENS",2339150,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="STATEN ISLAND",474558,temp_1$pop)
temp_1$pop <- ifelse(temp_1$BOROUGH=="BROOKLYN",2636735,temp_1$pop)
temp_1$per_cap <- temp_1$n/temp_1$pop
temp_1 %>%
ggplot(aes(x=BOROUGH,y=per_cap))+geom_bar(stat="identity")+ggtitle("Number of
Accidents Per Person")

col %>% filter(BOROUGH!="") %>%  group_by(DATE,BOROUGH) %>%
dplyr::summarise(n=mean(n())) %>% na.omit() %>%
  ggplot(aes(x=DATE, y=n, colour=BOROUGH, group=BOROUGH)) +
  geom_line()+
  scale_fill_hc("darkunica") +ggtitle("Borough Accidents(Mean) by Time")

col %>% filter(BOROUGH!="") %>% group_by(month,BOROUGH) %>%
dplyr::summarise(n=mean(n())) %>% na.omit() %>%
  ggplot(aes(x=month, y=n, fill=BOROUGH)) +
geom_bar(position="dodge",stat = "identity")+geom_text(aes(label=n),
vjust=1.5, colour="black",
position=position_dodge(.9), size=3)+ggtitle("Mean Number of Collisions Per
Month")


col %>% group_by(BOROUGH,day)%>% dplyr::summarise(n=mean(n())) %>%
filter(BOROUGH!="") %>%
  ggplot(aes(x=day, y=n, fill=BOROUGH)) +
geom_bar(position="dodge",stat = "identity")+geom_text(aes(label=n),
vjust=1.5, colour="black",
position=position_dodge(.9), size=3)+ggtitle("Mean Number of Collisions Per
Day")
```