homework vi

Nirbhay Pherwani, Dhiren Chandnani

2019-10-11

Part A : Cleaning NYC Data

Summary of what we did here - We observed that there were 243 unique complaint types. A lot of complaints had just 1 or 2 instances. So for analysis only major complaints would be required. Hence, only complaints with counts greater than 60 are selected to maintain relevancy. Complaint types had unwanted characters and also were not in a formatted manner. So we made it all upper case and replaced those characters. We observed that there were 138 unique Location types. A lot of locations had just 1 or 2 instances. So for analysis only major locations would be required. Hence, only locations with counts greater than 10 are selected to maintain relevancy. We observed that there were 64 unique Agency types. A lot of Agencies had just 1 or 2 instances. So for analysis only major agencies would be required. Hence, only agencies with counts greater than 10 are selected to maintain relevancy. The zipcodes were not in a correct 5-digit format. We used the zipcode package to clean the zip codes. We also replaced all missing values with NA.

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## 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 ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

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

```
nyc311<-fread("311_Service_Requests_from_2010_to_Present.csv")
names(nyc311)<-names(nyc311) %>%
  stringr::str_replace_all("\\s", ".")
```

```r
# removing unwanted columns
nyc311 <- subset(nyc311, select= -c(Landmark, Park.Borough, School.City, School.State, School.Zip, Taxi

# removing complaints with count < 60
nyc311<-nyc311 %>%
  group_by(Complaint.Type) %>%
  filter(n() >= 60)

# making complaint values uppercase
library(dplyr)
nyc311$Complaint.Type <- toupper(nyc311$Complaint.Type)
nyc311$Complaint.Type <- gsub('/', ' ', nyc311$Complaint.Type)
nyc311$Complaint.Type <- gsub('-', ' ', nyc311$Complaint.Type)

# removing location types with count < 10
nyc311<-nyc311 %>%
  group_by(Location.Type) %>%
  filter(n() >= 10)

# removing agencies with count < 10
nyc311<-nyc311 %>%
  group_by(Agency) %>%
  filter(n() >= 10)

# splitting the community board field
nyc311<-nyc311 %>% separate(Community.Board, c("Community.Code"), sep=" ", extra = "drop")

# splitting incident address
nyc311<-nyc311 %>% separate(Incident.Address, c("Incident.Code"), sep=" ",extra = "drop")

# cleaning zipcodes
library(zipcode)
nyc311$Incident.Zip<-clean.zipcodes(nyc311$Incident.Zip)

# changing case for city
nyc311$City <- toupper(nyc311$City)

# replacing missing values with NA
nyc311 <- nyc311 %>% mutate_at(vars(-group_cols()),na_if,"N/A")
nyc311 <- nyc311 %>% mutate_at(vars(-group_cols()),na_if,"")
nyc311 <- nyc311 %>% mutate_at(vars(-group_cols()),na_if," ")
nyc311 <- nyc311 %>% mutate_at(vars(-group_cols()),na_if,"N / A")

nyc311_backup <- nyc311
```

Part B : Exploration of NYC 311 Data

Exploring the Agency Field

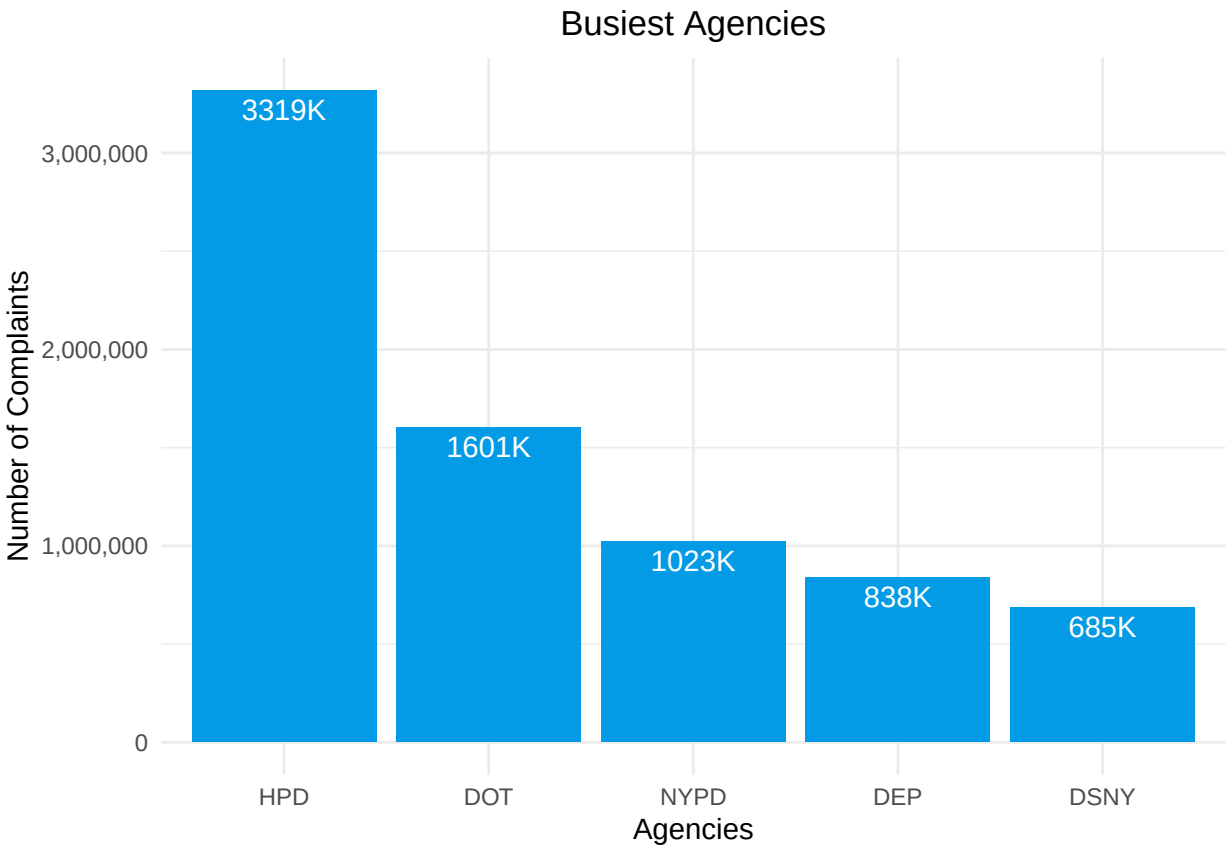
Busiest Agencies

The following plot shows the distribution of the service requests in various agencies. It can be seen that HPD (Department of Housing Preservation and Development) receives the most service calls followed by the DOT (Department of Transportation).

```
busyAgencies <- nyc311 %>%
   group_by(Agency) %>%
   summarize(Count=n()) %>%
   top_n(n=5, wt = Count)

busyAgencies$Agency<-factor(busyAgencies$Agency,
   levels=busyAgencies$Agency[order(busyAgencies$Count, decreasing = TRUE)])

library(scales)
```

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

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

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

```
busiestAgenciesPlot<-ggplot(busyAgencies,aes(x=Agency,y=Count))+
  geom_bar(stat="identity", fill = "#039BE5") +
  scale_y_continuous(labels = comma) +
labs(title = "Busiest Agencies", x = "Agencies", y = "Number of Complaints") +
geom_text(aes(label = paste(floor(Count/1000), "K", sep = "")), vjust = 1.4, color = "White") +
theme_minimal() +
theme(plot.title = element_text(hjust = 0.5))
busiestAgenciesPlot
```



Status of agencies

HPD clearly stands out for the maximum number of complaints being reported.

```
statusTab<-dplyr::filter(nyc311,
   Status=='Closed' |
   Status=='Open' |
   Status=='Assigned' |
   Status=='Pending'
)
agencyStatusTab<-select(statusTab,Agency,"Status")
library(gmodels)
CrossTable(agencyStatusTab$Agency,agencyStatusTab$'Status')
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  9091856
##
##
##                       | agencyStatusTab$Status
```

agencyStatusTab$Agency	Assigned	Closed	Open	Pending	Row Total
3-1-1	0	20583	1916	0	22499
	343.358	89.933	42.394	665.978	
	0.000	0.915	0.085	0.000	0.002
	0.000	0.003	0.002	0.000	
	0.000	0.002	0.000	0.000	
DCA	309	124478	351	407	125545
	1347.780	2678.508	11712.270	2946.749	
	0.002	0.992	0.003	0.003	0.014
	0.002	0.016	0.000	0.002	
	0.000	0.014	0.000	0.000	
DEP	2664	694054	130365	5	827088
	7856.462	285.104	28957.350	24472.084	
	0.003	0.839	0.158	0.000	0.091
	0.019	0.089	0.145	0.000	
	0.000	0.076	0.014	0.000	
DFTA	16	1949	12	0	1977
	6.656	38.719	172.073	58.520	
	0.008	0.986	0.006	0.000	0.000
	0.000	0.000	0.000	0.000	
	0.000	0.000	0.000	0.000	
DHS	1	3321	62	0	3384
	49.663	61.793	221.849	100.168	
	0.000	0.981	0.018	0.000	0.000
	0.000	0.000	0.000	0.000	
	0.000	0.000	0.000	0.000	
DOB	13356	386283	76030	1	475670
	5120.510	1087.788	17933.794	14077.992	
	0.028	0.812	0.160	0.000	0.052
	0.096	0.050	0.085	0.000	
	0.001	0.042	0.008	0.000	
DOE	293	11710	5	0	12008
	65.723	198.068	1176.459	355.441	
	0.024	0.975	0.000	0.000	0.001
	0.002	0.002	0.000	0.000	
	0.000	0.001	0.000	0.000	
DOF	22	129415	140	0	129577
	1933.722	3068.995	12524.246	3835.523	
	0.000	0.999	0.001	0.000	0.014
	0.000	0.017	0.000	0.000	
	0.000	0.014	0.000	0.000	
DOHMH	47817	149403	12806	53254	263280
	477450.668	25654.825	6705.368	265191.788	
	0.182	0.567	0.049	0.202	0.029
	0.345	0.019	0.014	0.198	
	0.005	0.016	0.001	0.006	
DOITT	21	4676	3	0	4700
	35.875	105.371	458.398	139.122	
	0.004	0.995	0.001	0.000	0.001
	0.000	0.001	0.000	0.000	
	0.000	0.001	0.000	0.000	
DOT	21091	1383770	43263	153229	1601353
	458.473	113.503	83523.605	236276.662	
	0.013	0.864	0.027	0.096	0.176
	0.152	0.178	0.048	0.569	
	0.002	0.152	0.005	0.017	

```
##                     DPR |     46227 |    317310 |     42397 |         0 |    405934 |
##                         | 258688.004 |  2642.047 |   130.651 | 12015.783 |           |
##                         |     0.114 |     0.782 |     0.104 |     0.000 |     0.045 |
##                         |     0.333 |     0.041 |     0.047 |     0.000 |           |
##                         |     0.005 |     0.035 |     0.005 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                    DSNY |      2956 |    585500 |     34506 |     62224 |    685186 |
##                         |  5380.273 |     2.657 | 16274.662 | 86735.842 |           |
##                         |     0.004 |     0.855 |     0.050 |     0.091 |     0.075 |
##                         |     0.021 |     0.075 |     0.038 |     0.231 |           |
##                         |     0.000 |     0.064 |     0.004 |     0.007 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                     EDC |         0 |      4656 |        98 |         0 |      4754 |
##                         |    72.551 |    84.059 |   294.160 |   140.720 |           |
##                         |     0.000 |     0.979 |     0.021 |     0.000 |     0.001 |
##                         |     0.000 |     0.001 |     0.000 |     0.000 |           |
##                         |     0.000 |     0.001 |     0.000 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                    FDNY |         0 |     23613 |        78 |         0 |     23691 |
##                         |   361.549 |   545.138 |  2187.363 |   701.262 |           |
##                         |     0.000 |     0.997 |     0.003 |     0.000 |     0.003 |
##                         |     0.000 |     0.003 |     0.000 |     0.000 |           |
##                         |     0.000 |     0.003 |     0.000 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                     HPD |        16 |   2774289 |    544748 |         2 |   3319055 |
##                         | 50620.171 |  1623.694 | 143343.979 | 98241.146 |           |
##                         |     0.000 |     0.836 |     0.164 |     0.000 |     0.365 |
##                         |     0.000 |     0.356 |     0.606 |     0.000 |           |
##                         |     0.000 |     0.305 |     0.060 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                     HRA |         0 |     37505 |         0 |         0 |     37505 |
##                         |   572.365 |   903.960 |  3705.641 |  1110.161 |           |
##                         |     0.000 |     1.000 |     0.000 |     0.000 |     0.004 |
##                         |     0.000 |     0.005 |     0.000 |     0.000 |           |
##                         |     0.000 |     0.004 |     0.000 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                   NYCEM |         1 |        12 |         0 |         0 |        13 |
##                         |     3.239 |     0.068 |     1.284 |     0.385 |           |
##                         |     0.077 |     0.923 |     0.000 |     0.000 |     0.000 |
##                         |     0.000 |     0.000 |     0.000 |     0.000 |           |
##                         |     0.000 |     0.000 |     0.000 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                    NYPD |        90 |   1022623 |       400 |         0 |   1023113 |
##                         | 15434.267 | 24495.328 | 100289.156 | 30284.489 |           |
##                         |     0.000 |     1.000 |     0.000 |     0.000 |     0.113 |
##                         |     0.001 |     0.131 |     0.000 |     0.000 |           |
##                         |     0.000 |     0.112 |     0.000 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                     OEM |        66 |       793 |         0 |         0 |       859 |
##                         |   213.395 |     4.480 |    84.873 |    25.427 |           |
##                         |     0.077 |     0.923 |     0.000 |     0.000 |     0.000 |
##                         |     0.000 |     0.000 |     0.000 |     0.000 |           |
##                         |     0.000 |     0.000 |     0.000 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                     OPS |         0 |         0 |         4 |         0 |         4 |
##                         |     0.061 |     3.425 |    32.879 |     0.118 |           |
##                         |     0.000 |     0.000 |     1.000 |     0.000 |     0.000 |
##                         |     0.000 |     0.000 |     0.000 |     0.000 |           |
##                         |     0.000 |     0.000 |     0.000 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##                     TLC |      3805 |    109729 |     11127 |         0 |    124661 |
##                         |  1902.638 |    83.046 |   114.970 |  3690.008 |           |
##                         |     0.031 |     0.880 |     0.089 |     0.000 |     0.014 |
##                         |     0.027 |     0.014 |     0.012 |     0.000 |           |
##                         |     0.000 |     0.012 |     0.001 |     0.000 |           |
## ------------------------|-----------|-----------|-----------|-----------|-----------|
##            Column Total |    138751 |   7785672 |    898311 |    269122 |   9091856 |
##                         |     0.015 |     0.856 |     0.099 |     0.030 |           |
```

```
## ----------------------|----------|----------|----------|----------|----------|
##
##
```

Which agency is taking care of complaints expeditiously?

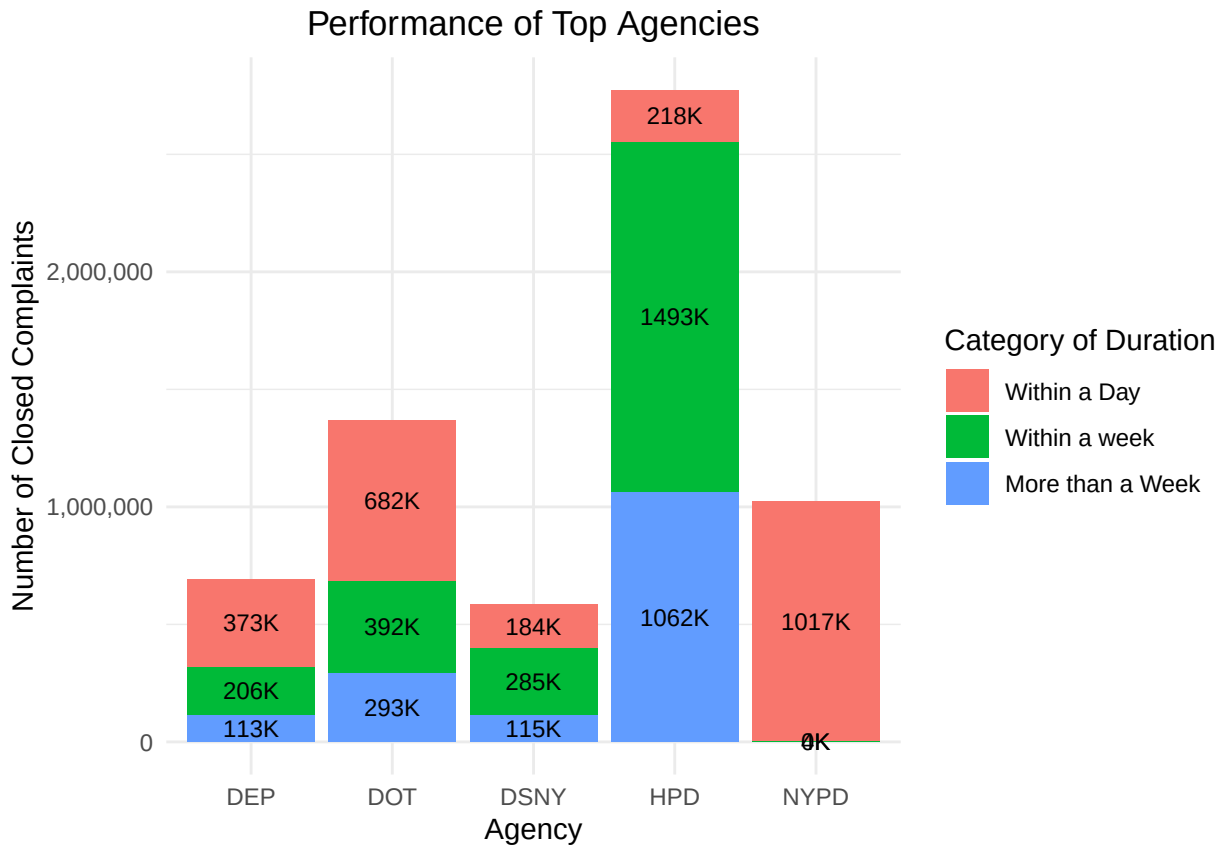
Lets see the performance of these agencies for the complaints they have closed.

```
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
```

```
# Filter closed status agency
agencyFilter <- nyc311 %>%
  select(Agency, Status, Created.Date, Closed.Date, Unique.Key) %>%
  filter(Status == "Closed")

# Calulate duration in days
agencyFilter$durations<-mdy_hms(agencyFilter$Closed.Date)-mdy_hms(agencyFilter$Created.Date)
agencyFilter$durations<-round(as.numeric(agencyFilter$durations,units='days'),2)

# Categorise duration
agencyDurations <- agencyFilter %>%
    filter(durations>=0, Agency == "HPD"| Agency == "DOT" | Agency == "NYPD" | Agency == "DEP" | Agency
    mutate(Duration.Category = case_when(
                                  durations<=1 ~ 'Inside One Day',
                                  durations <= 7 ~ 'Within One Week',
                                  durations > 7 ~ 'More than a Week',
                                  ))

# Spreading duration
agencyDurationsBackup <- agencyDurations
agencyDurations$durationsCount <- rep(1, nrow(agencyDurations))
agencyDurations <- agencyDurations %>% spread(Duration.Category, durationsCount)

agencyDurations$`Inside One Day`[is.na(agencyDurations$`Inside One Day`)] <- 0
agencyDurations$`Within One Week`[is.na(agencyDurations$`Within One Week`)] <- 0
agencyDurations$`More than a Week`[is.na(agencyDurations$`More than a Week`)] <- 0

# Getting count of each duration category
library("dplyr")
agency_data <- agencyDurations %>%
```

```
  select(Agency, `Inside One Day`, `Within One Week`, `More than a Week`) %>%
   group_by(Agency) %>%
   dplyr::summarise(
      OneDayCount = sum(`Inside One Day`),
      OneWeekCount = sum(`Within One Week`),
      MoreOneWeekCount = sum(`More than a Week`)
)

# Chart
meltedAgencyPerformance <- melt(agency_data, "Agency")

colnames(meltedAgencyPerformance)[colnames(meltedAgencyPerformance)=="variable"] <- "Category of Duratio
colnames(meltedAgencyPerformance)[colnames(meltedAgencyPerformance)=="value"] <- "Number of Closed Compl

agencyPerfomancePlot<-ggplot(meltedAgencyPerformance, aes(x = Agency, y = `Number of Closed Complaints`
  labs(title = "Performance of Top Agencies") +
  geom_bar(stat = 'identity', position = 'stack') +
  geom_text(aes(label = paste(floor(`Number of Closed Complaints`/1000), "K", sep = "")), size = 3, vjus
  scale_fill_discrete(labels = c("Within a Day", "Within a week", "More than a Week")) +
  scale_y_continuous(labels = comma) +
  theme_minimal() +
  theme(axis.ticks.x = element_blank(), plot.title = element_text(hjust = 0.5))
agencyPerfomancePlot
```



The above graph clearly shows that NYPD closes the most complaints within a day. They seem to be the best performing agency, at the same time HPD seems to be having too much workload, and is performing

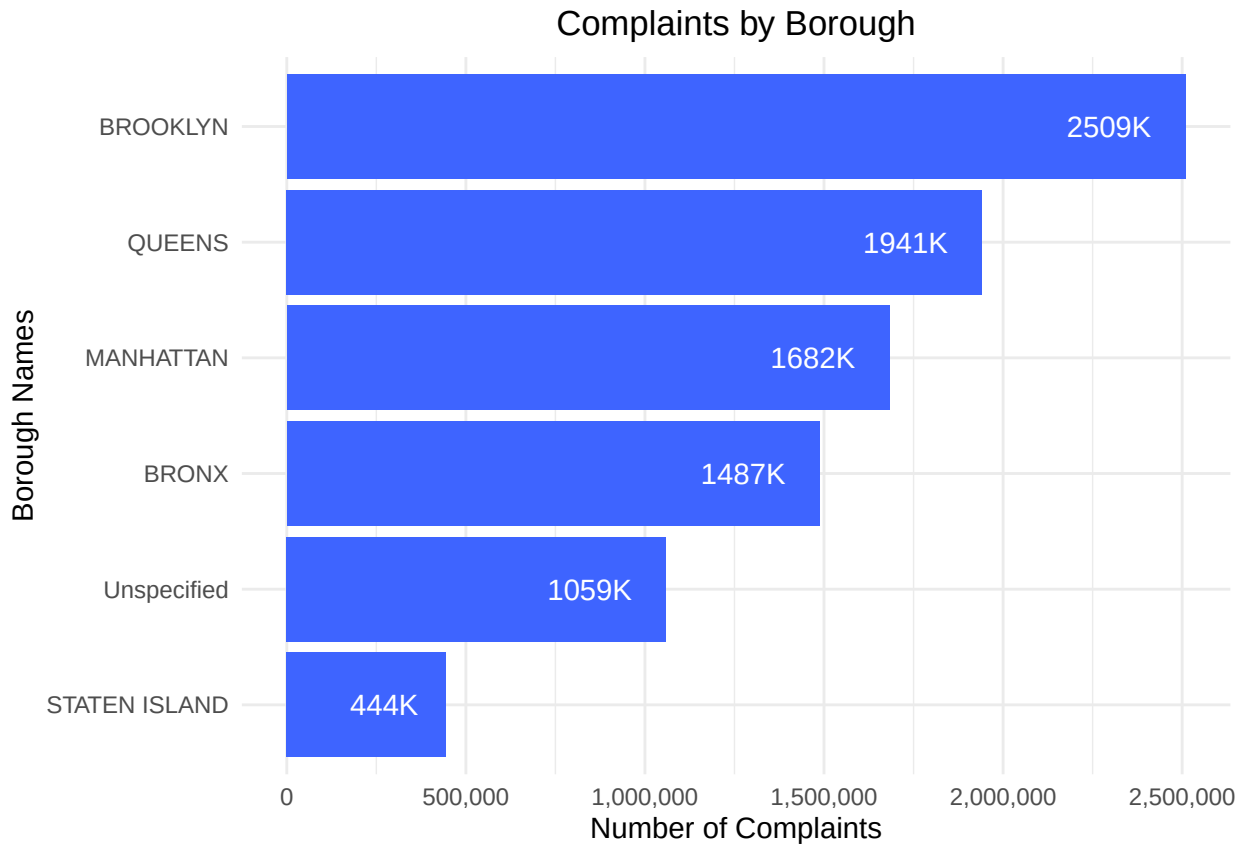
better than average, we could say.

Complaint Types & Boroughs

Complaints by Boroughs

The below graph talks about the Borough with most complaints. Brooklyn has the highest number of complaints followed by Queens.

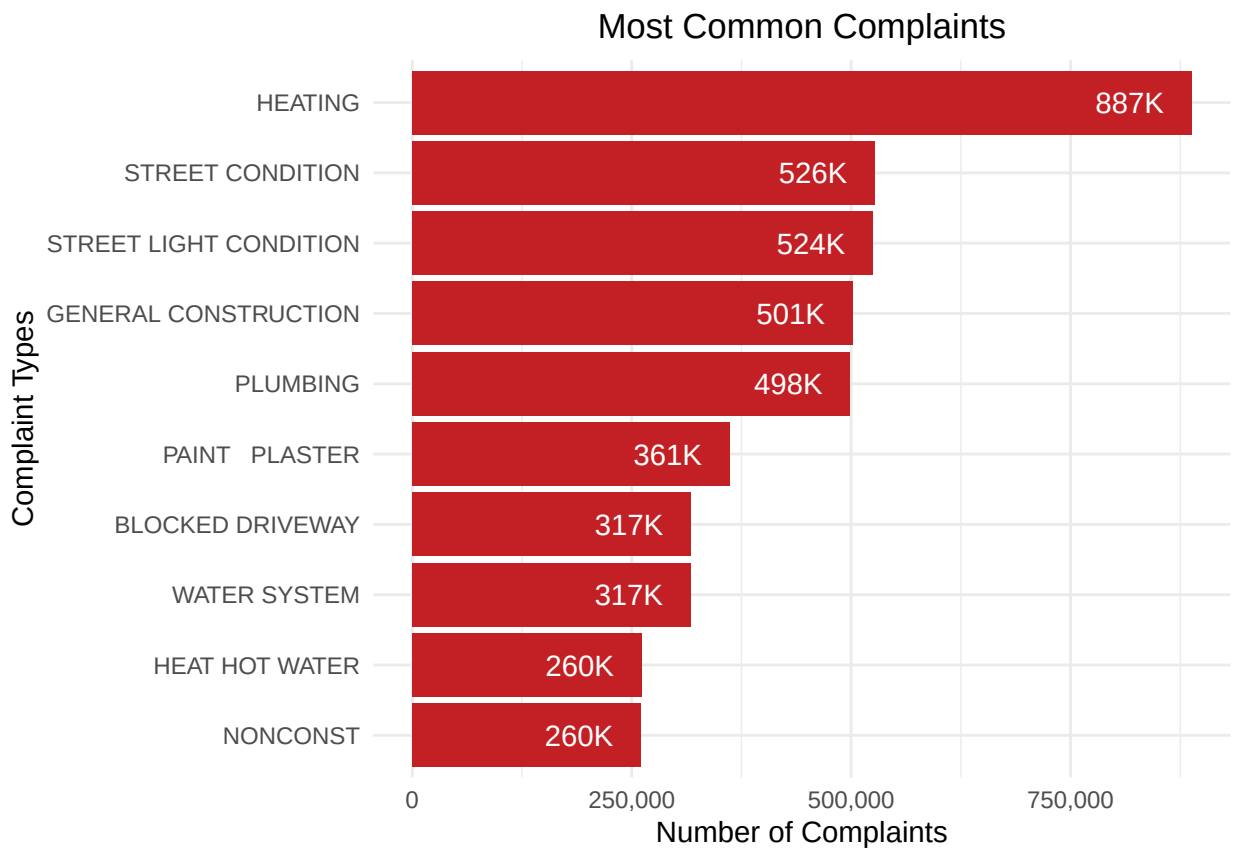
```
boroughComplaints <- nyc311 %>%
   group_by(Borough) %>%
   summarize(Count=n())

boroughComplaintsPlot<-ggplot(boroughComplaints,aes(x=reorder(Borough, Count),y=Count)) +
   geom_bar(stat="identity", fill = "#3E64FF") +
   scale_y_continuous(labels = comma) +
   labs(title = "Complaints by Borough", x = "Borough Names", y = "Number of Complaints") +
   geom_text(aes(label = paste(floor(Count/1000), "K", sep = "")), hjust = 1.4, color = "White") +
     theme_minimal() +
     theme(plot.title = element_text(hjust = 0.5)) +
     coord_flip()
boroughComplaintsPlot
```



Most common complaints

The below graph gives us a crucial information about the common types of complaint. It shows that the highest count of complaints recieved is regarding the heating problem.

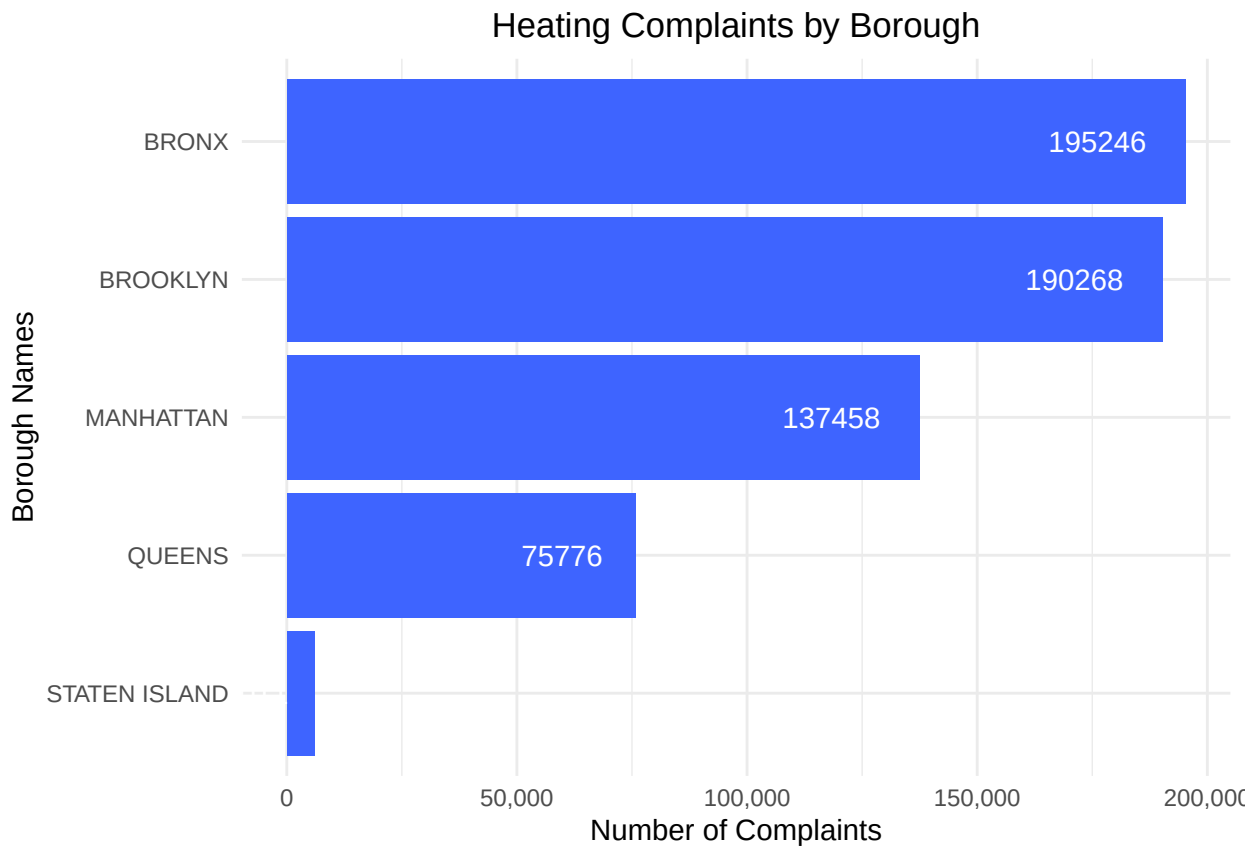
```
mostCommonComplaints <- nyc311 %>%
   group_by(Complaint.Type) %>%
   summarize(Count=n()) %>%
   top_n(n=10, wt = Count)

mostCommonComplaints$Complaint.Type<-factor(mostCommonComplaints$Complaint.Type,
   levels=mostCommonComplaints$Complaint.Type[order(mostCommonComplaints$Count)])

mostCommonComplaintsPlot<-ggplot(mostCommonComplaints,aes(x=Complaint.Type,y=Count)) +
    geom_bar(stat="identity", fill = "#C32026") +
    scale_y_continuous(labels = comma) +
    labs(title = "Most Common Complaints", x = "Complaint Types", y = "Number of Complaints") +
    geom_text(aes(label = paste(floor(Count/1000), "K", sep = "")), hjust = 1.4, color = "White") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5)) +
    coord_flip()
mostCommonComplaintsPlot
```



Selecting all the heating complaints by Borough

It can be seen that Heating complaints prevail in the Bronx Borough.

```
heatingBoroughCounts<-nyc311 %>%
  filter(Complaint.Type=="HEATING", Borough != "Unspecified") %>%
  group_by(Borough) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  ungroup() %>%
  mutate(Borough = reorder(Borough,Count))

ggplot(data=heatingBoroughCounts, aes(x = Borough,y = Count)) +
  geom_bar(stat="identity", fill = "#3E64FF") +
   scale_y_continuous(labels = comma) +
   labs(title = "Heating Complaints by Borough", x = "Borough Names", y = "Number of Complaints") +
   geom_text(aes(label = Count), hjust = 1.4, color = "White") +
     theme_minimal() +
     theme(plot.title = element_text(hjust = 0.5)) +
     coord_flip()
```

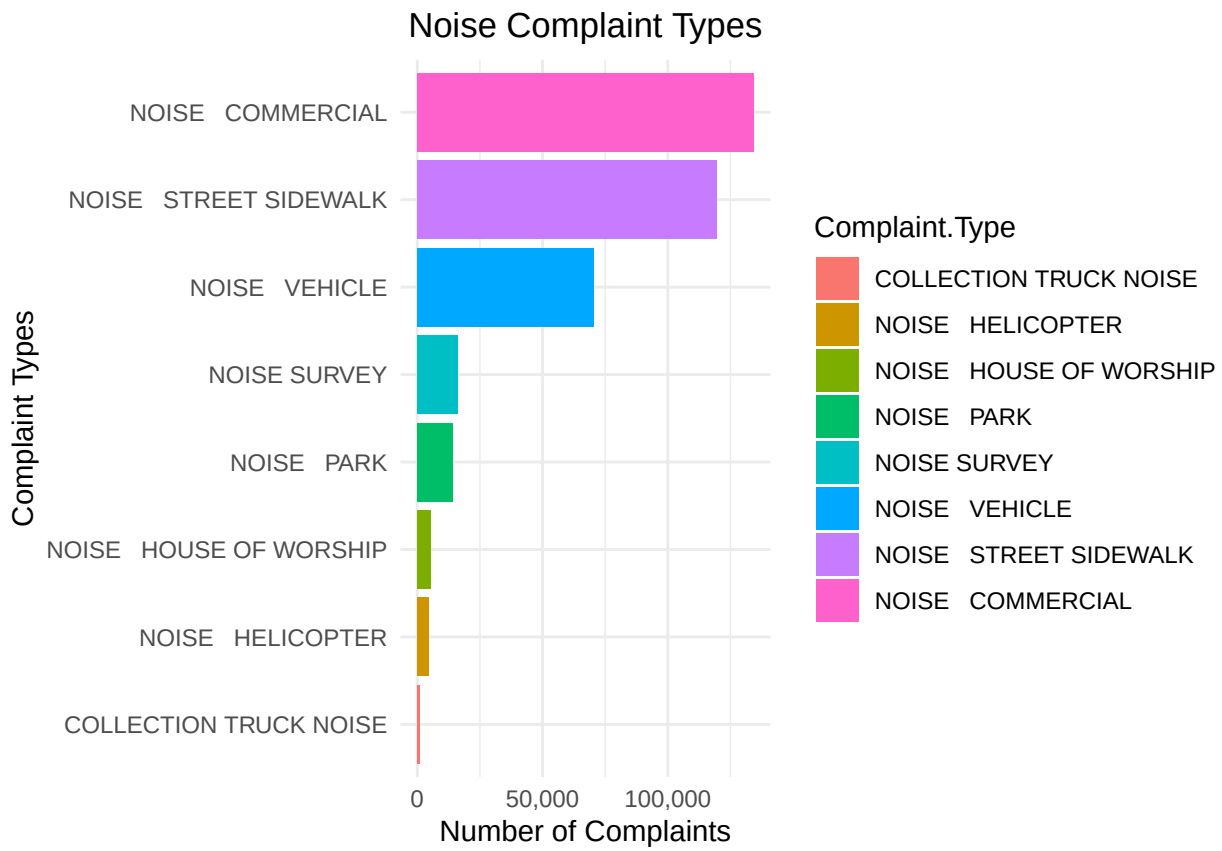


Exploring the Noise Complaints in NYC311

In this section we are trying to display all the noise complaints in New York and explore them more categorically.

What are the different noise complaints and which has the highest count after general noise complaints?

```
noiseComplaintLocations <- nyc311 %>%
  select(Complaint.Type,
    Longitude,
    Latitude,
    Agency
  ) %>%
  filter(str_detect(Complaint.Type,"NOISE"), Complaint.Type != "NOISE")

noiseComplaintData<-noiseComplaintLocations %>%
  group_by(Complaint.Type) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  ungroup() %>%
  mutate(Complaint.Type = reorder(Complaint.Type,Count))

ggplot(data=noiseComplaintData, aes(x = Complaint.Type,y = Count, fill = Complaint.Type)) +
   geom_bar(stat="identity") +
   scale_y_continuous(labels = comma) +
   labs(title = "Noise Complaint Types", x = "Complaint Types", y = "Number of Complaints") +
   #geom_text(aes(label = Count), hjust = 1.4, color = "White") +
   theme_minimal() +
   theme(plot.title = element_text(hjust = 0.5)) +
   coord_flip()
```

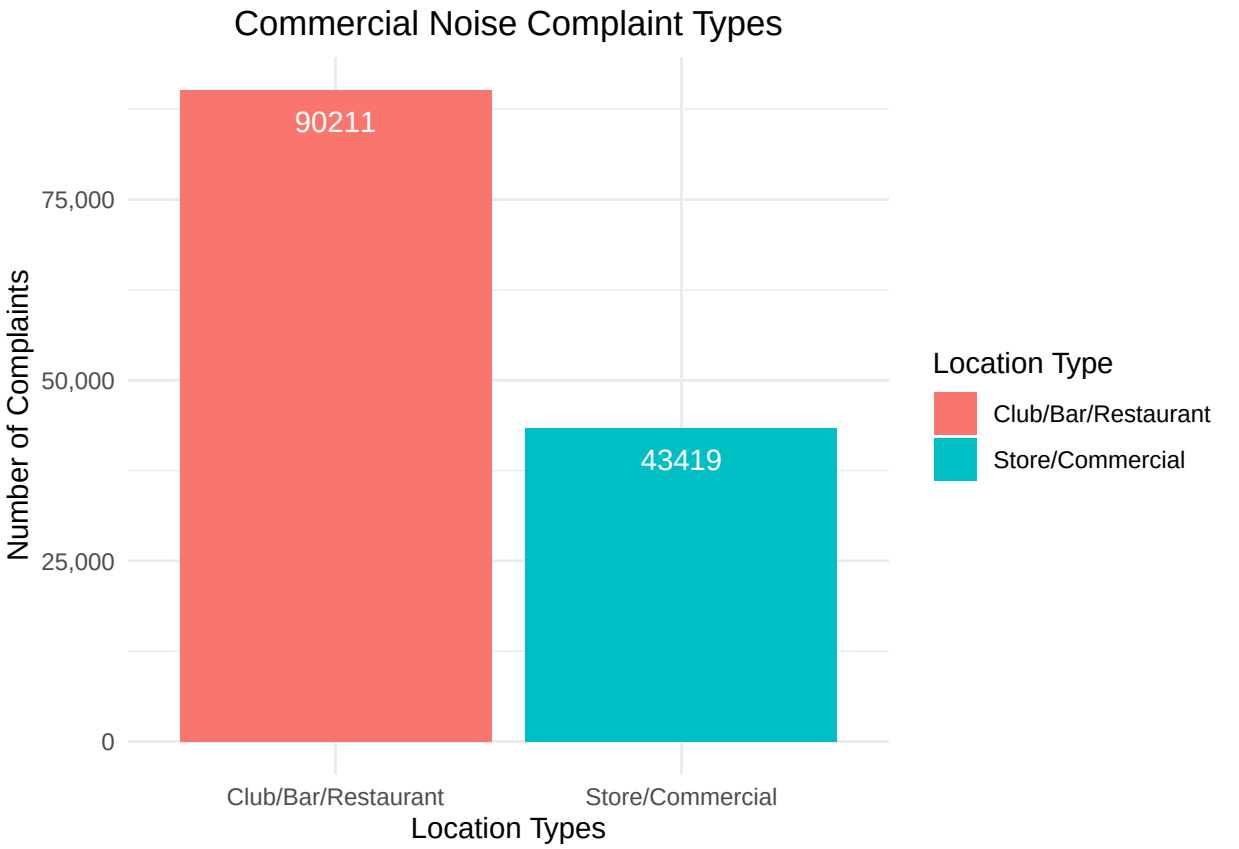


As we can see, Noise - Commercial is the category with second highest number of noise complaints. Next, we will see the distribution for commercial noise complaints.

Commercial-Noise complaints

```
require(dplyr)
commercialNoiseLocationsData <- nyc311 %>%
  dplyr::select(Location.Type,
         Complaint.Type,
         Longitude,
         Latitude,
         Agency) %>%
  dplyr::filter(str_detect(Complaint.Type,"COMMERCIAL"))

commercialNoiseCounts <- commercialNoiseLocationsData %>%
  group_by(Location.Type) %>%
  summarise(Count = n()) %>%
  filter(!is.na(Location.Type))

ggplot(data=commercialNoiseCounts, aes(x = Location.Type,y = Count, fill = Location.Type)) +
   geom_bar(stat="identity") +
   scale_y_continuous(labels = comma) +
   labs(title = "Commercial Noise Complaint Types", x = "Location Types", y = "Number of Complaints", fi
   geom_text(aes(label = Count), vjust = 2.0, color = "White") +
   theme_minimal() +
```

```
   theme(plot.title = element_text(hjust = 0.5))
```

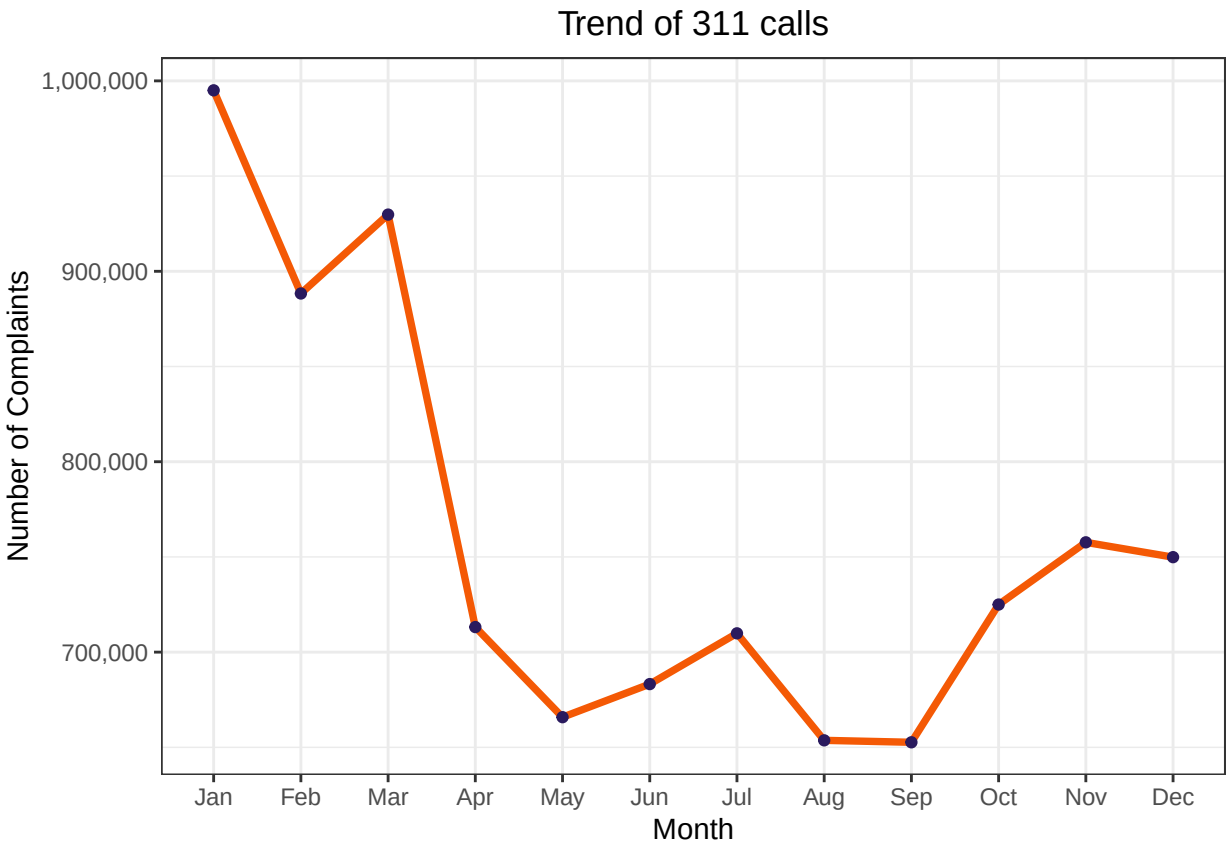


As we can see, Clubs/Bars/Restaurants contribute to the most number of commercial noise complaints. Lets see what it looks like on a map.

Trend of 311 calls

This is one of the most important graphs of this analysis. It shows the trend of 311 calls across various months. As you can see, January and March are the top 2 busiest months of service request.

```
library(lubridate)
trendPlot<-nyc311 %>%
   mutate(Month = month.abb[month(mdy_hms(Created.Date))]) %>%
   filter(!is.na(Month)) %>%
   group_by(Month) %>%
   summarise(Count = n()) %>%
   ggplot(aes(x = Month,y = Count, group=1)) +
   geom_line(color = "#F45905", size = 1.3) +
   geom_point(color = "#2A1A5E") +
   theme_bw() +
   scale_x_discrete(limits = month.abb) +
   scale_y_continuous(labels = comma) +
   labs(title = "Trend of 311 calls", x = "Month", y = "Number of Complaints") +
   theme(plot.title = element_text(hjust = 0.5))
trendPlot
```

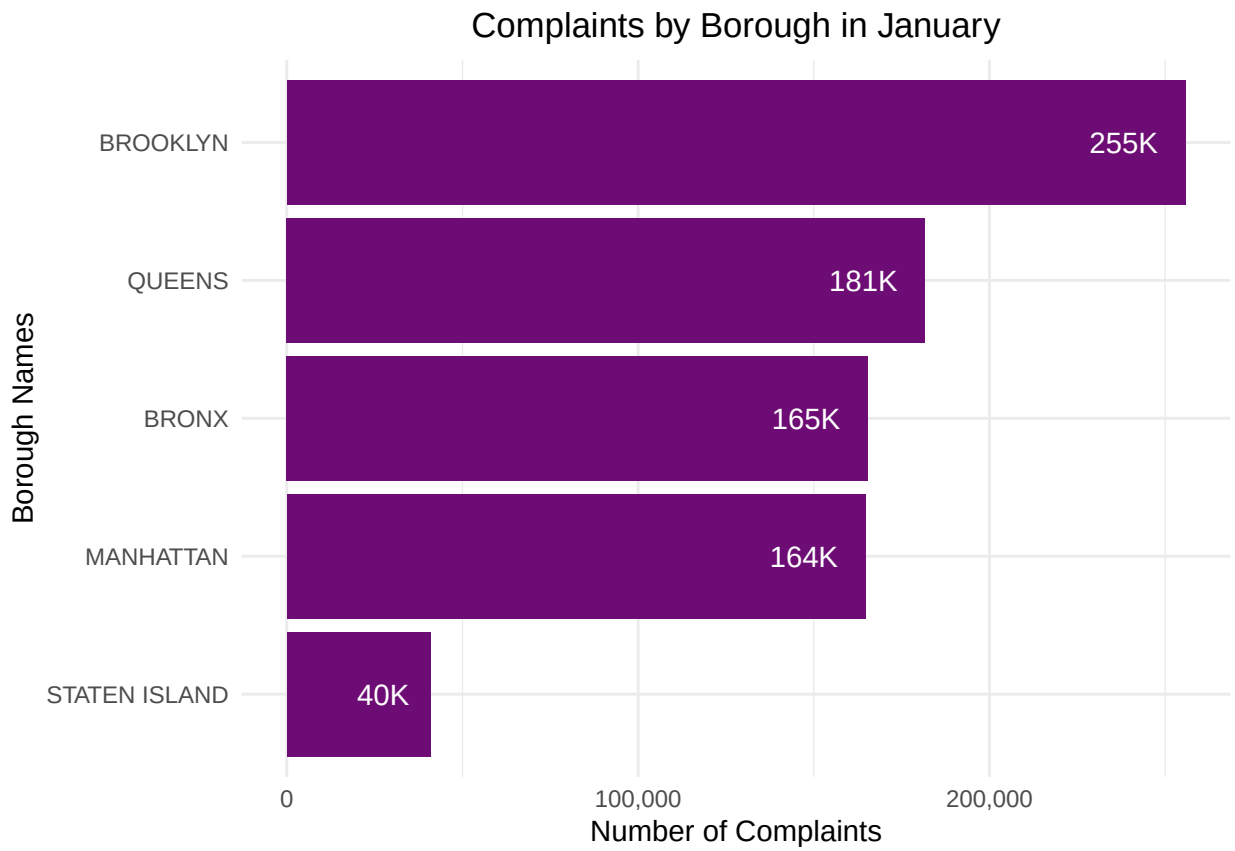


We can clearly see that January has the highest number of complaints getting reported.

Let's see January's complaints borough wise.

Plotting complaints by borough in January

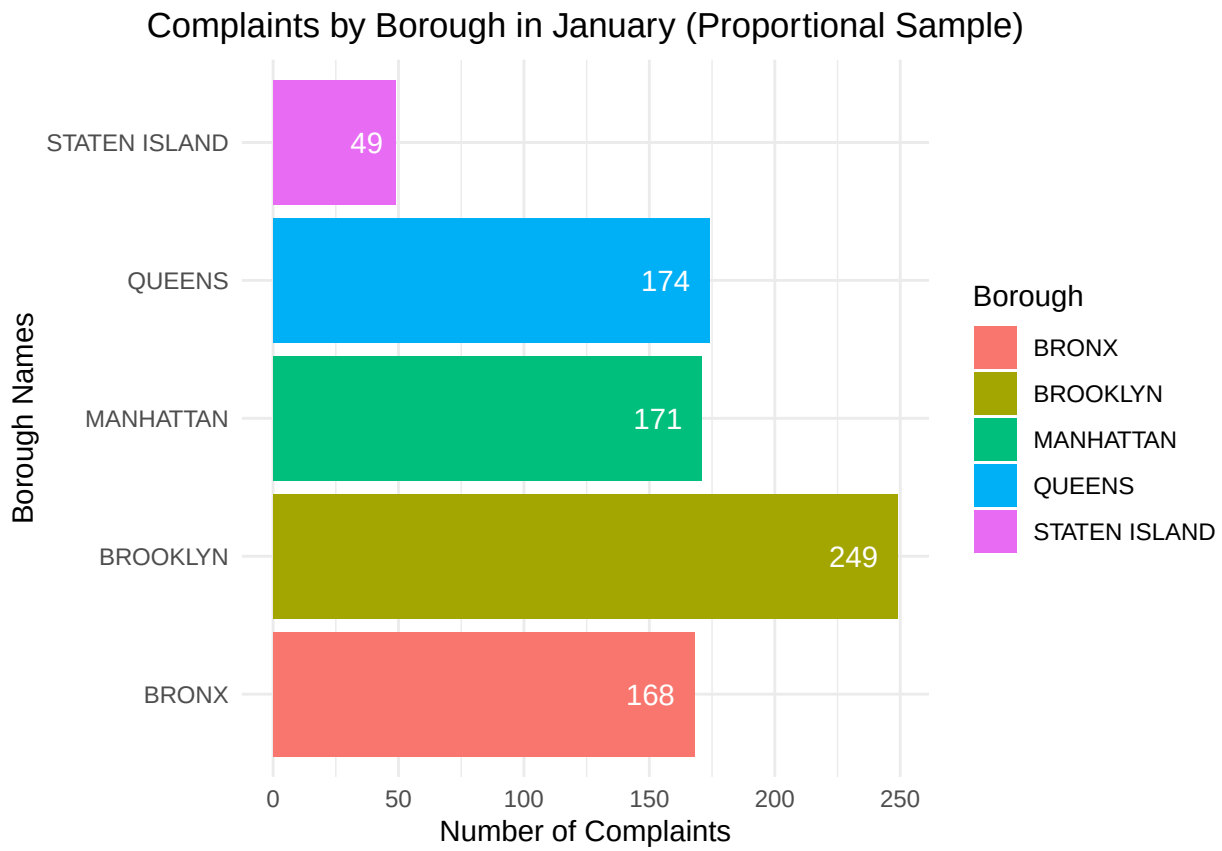
```
janBoroughCounts<- nyc311 %>%
  mutate(month = month.abb[month(mdy_hms(Created.Date))]) %>%
  filter(!is.na(month), month=='Jan', Borough != "Unspecified") %>%
  group_by(Borough) %>%
  dplyr::summarise(Count = n()) %>%
  arrange(desc(Count)) %>%
  ungroup() %>%
  mutate(Borough = reorder(Borough,Count))


ggplot(data=janBoroughCounts, aes(x = Borough,y = Count)) +
  geom_bar(stat="identity", fill = "#6D0C74") +
   scale_y_continuous(labels = comma) +
   labs(title = "Complaints by Borough in January", x = "Borough Names", y = "Number of Complaints") +
   geom_text(aes(label = paste(floor(Count/1000), "K", sep = "")), hjust = 1.4, color = "White") +
     theme_minimal() +
     theme(plot.title = element_text(hjust = 0.5)) +
     coord_flip()
```



As you can see, Brooklyn has the most number of complaints reported in January. Lets take a proportional sample to see it on a map.

Stratified Sampling

```
library("splitstackshape")
sampleJanBoroughComplaints<-splitstackshape::stratified(nyc311, "Borough", 0.001, select = list(Borough

sampleJanBoroughCounts<- sampleJanBoroughComplaints %>%
  mutate(month = month.abb[month(mdy_hms(Created.Date))]) %>%
  filter(!is.na(month), month=='Jan', Borough != "Unspecified") %>%
  group_by(Borough) %>%
  dplyr::summarise(Count = n())

ggplot(data=sampleJanBoroughCounts, aes(x = Borough,y = Count, fill = Borough)) +
  geom_bar(stat="identity") +
   scale_y_continuous(labels = comma) +
   labs(title = "Complaints by Borough in January (Proportional Sample)", x = "Borough Names", y = "Numb
   geom_text(aes(label = Count), hjust = 1.4, color = "White") +
     theme_minimal() +
     theme(plot.title = element_text(hjust = 0.5)) +
     coord_flip()
```



The above graph shows that the sampling performed was proportionate to the actual data. So we can now go ahead and see the distribution on the map.

Plotting complaints - Borough/January/Sampled

```
sampleJanBoroughLocations <- sampleJanBoroughComplaints %>%
  select(Created.Date, Borough,
    Longitude,
    Latitude
  ) %>%
  mutate(month = month.abb[month(mdy_hms(Created.Date))]) %>%
  filter(!is.na(month), month=='Jan', !is.na(Latitude), Latitude < 42)

sampleJanBoroughLocations<- sampleJanBoroughLocations %>%
  select(Borough,
    Longitude,
    Latitude
)

library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.
```

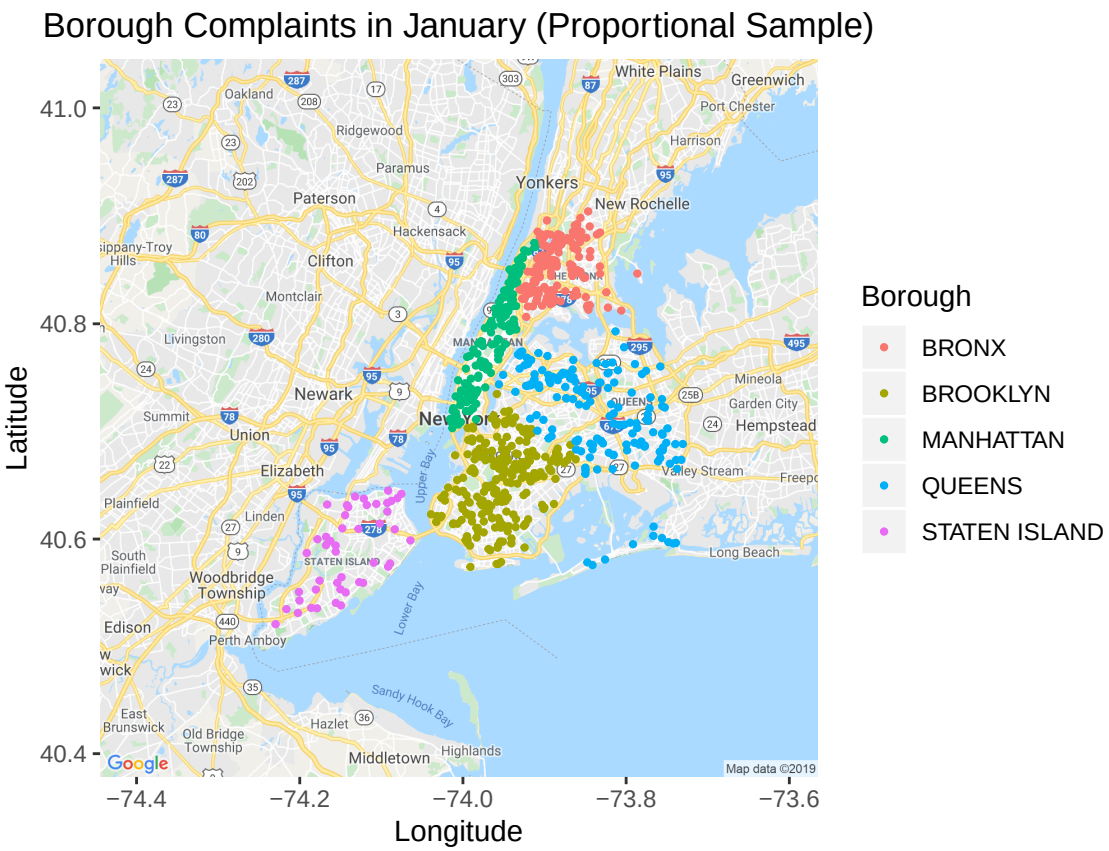
```
library(curl)
```

```
##
## Attaching package: 'curl'

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

```
key <- "AIzaSyClTqcMNpFm9_rFaaXH6ptzDpmTmAEwml4"
register_google(key=key)
nyc_map <- get_map(location="New York City",
           maptype="roadmap",zoom=10)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=New%20York%20City&zoom=10&size=640x640

## Source : https://maps.googleapis.com/maps/api/geocode/json?address=New+York+City&key=xxx
```

```
map <- ggmap(nyc_map) +
  geom_point(data=sampleJanBoroughLocations,aes(x=Longitude,y=Latitude, color=Borough),
          size=0.8) +
  ggtitle("Borough Complaints in January (Proportional Sample)") +
  theme(plot.title=element_text(hjust=0.5)) +
  xlab("Longitude") + ylab("Latitude")
map
```



Again, we can see through the map too that has been made using a proportionate sample, that Brooklyn seems to have max number of complaints being reported in January.

For Information Purposes - CrossTab Between Borough and Status

```
statusTab<-dplyr::filter(nyc311,
   Status=='Closed' |
   Status=='Open' |
   Status=='Assigned' |
   Status=='Pending'
)
boroughTab<-select(statusTab,Borough, Agency, "Status")
library(gmodels)
CrossTable(boroughTab$Borough,boroughTab$'Status')
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  9091856
##
##
##                    | boroughTab$Status
## boroughTab$Borough |  Assigned |    Closed |      Open |   Pending | Row Total |
## -------------------|-----------|-----------|-----------|-----------|-----------|
##              BRONX |     23154 |   1190437 |    203939 |     68632 |   1486162 |
##                    |     9.892 |  5311.213 | 22204.323 | 13802.493 |           |
##                    |     0.016 |     0.801 |     0.137 |     0.046 |     0.163 |
##                    |     0.167 |     0.153 |     0.227 |     0.255 |           |
##                    |     0.003 |     0.131 |     0.022 |     0.008 |           |
## -------------------|-----------|-----------|-----------|-----------|-----------|
##           BROOKLYN |     39970 |   2119823 |    274274 |     71077 |   2505144 |
##                    |    79.096 |   301.185 |  2892.238 |   127.607 |           |
##                    |     0.016 |     0.846 |     0.109 |     0.028 |     0.276 |
##                    |     0.288 |     0.272 |     0.305 |     0.264 |           |
##                    |     0.004 |     0.233 |     0.030 |     0.008 |           |
## -------------------|-----------|-----------|-----------|-----------|-----------|
##          MANHATTAN |     25847 |   1428962 |    182042 |     43514 |   1680365 |
##                    |     1.606 |    69.396 |  1544.882 |   779.166 |           |
##                    |     0.015 |     0.850 |     0.108 |     0.026 |     0.185 |
##                    |     0.186 |     0.184 |     0.203 |     0.162 |           |
##                    |     0.003 |     0.157 |     0.020 |     0.005 |           |
## -------------------|-----------|-----------|-----------|-----------|-----------|
##             QUEENS |     41126 |   1673016 |    161318 |     62725 |   1938185 |
##                    |  4507.996 |   106.271 |  4757.015 |   499.662 |           |
##                    |     0.021 |     0.863 |     0.083 |     0.032 |     0.213 |
##                    |     0.296 |     0.215 |     0.180 |     0.233 |           |
##                    |     0.005 |     0.184 |     0.018 |     0.007 |           |
## -------------------|-----------|-----------|-----------|-----------|-----------|
##      STATEN ISLAND |      8194 |    366974 |     47364 |     21005 |    443537 |
##                    |   300.069 |   434.211 |   286.088 |  4725.003 |           |
##                    |     0.018 |     0.827 |     0.107 |     0.047 |     0.049 |
##                    |     0.059 |     0.047 |     0.053 |     0.078 |           |
##                    |     0.001 |     0.040 |     0.005 |     0.002 |           |
## -------------------|-----------|-----------|-----------|-----------|-----------|
##        Unspecified |       460 |   1006460 |     29374 |      2169 |   1038463 |
```

```
##                    | 14941.357 | 15443.020 | 52265.536 | 26553.904 |           |
##                    |     0.000 |     0.969 |     0.028 |     0.002 |     0.114 |
##                    |     0.003 |     0.129 |     0.033 |     0.008 |           |
##                    |     0.000 |     0.111 |     0.003 |     0.000 |           |
## -------------------|-----------|-----------|-----------|-----------|-----------|
##       Column Total |    138751 |   7785672 |    898311 |    269122 |   9091856 |
##                    |     0.015 |     0.856 |     0.099 |     0.030 |           |
## -------------------|-----------|-----------|-----------|-----------|-----------|
##
##
```

23154 of all SRs with a status 'Assigned' are from the Bronx borough. 71077 of all SRs with a status 'Pending' are from the Brooklyn borough. Of all the 1038463 Unspecified location service requests, 1006460 have been closed.

PART C : Housing Units by Building Dataset

Preprocessing for Housing NY Units by Building Data

We selected the Housing New York Units by Building data. The data is provided by Department of Housing Preservation and Development (HPD) and is avaliable from the NYC Open Data website (https://data.cityofnewyork.us/Housing-Development/Housing-New-York-Units-by-Building/hg8x-zxpr)

```
nycHousingData <-fread("https://data.cityofnewyork.us/api/views/hg8x-zxpr/rows.csv")
```

```
## Warning in require_bit64(): Some columns are type 'integer64' but
## package bit64 is not installed. Those columns will print as strange
## looking floating point data. There is no need to reload the data. Simply
## install.packages('bit64') to obtain the integer64 print method and print
## the data again.
```

```
names(nycHousingData)<-names(nycHousingData) %>%
   stringr::str_replace_all("\\s", ".")
nycHousingData[nycHousingData=="" | nycHousingData==" "]<-NA

# making borough upper case
nycHousingData$Borough <- toupper(nycHousingData$Borough)

# filter for getting data just for 2014
nycHousingData<- nycHousingData %>% separate(Project.Start.Date, c("Month", "Day", "Year"), sep="/")
nycHousingData<-filter(nycHousingData, Year == "2014")
```

Summarize Housing Data

In this section we have grouped the housing data on the basis of postcode(zipcode) and summarized it to get sum values for Total Units which we have used later.

```
buildingUnits<- nycHousingData %>%
group_by(Postcode) %>%
dplyr::summarise(
     TotalUnits = sum(Total.Units)
   )
```

Joining 311 Calls and Housing datasets

In this section we have joined the two datasets based on zipcodes.

```
# converting df to character type
buildingUnits$Postcode<-as.character(buildingUnits$Postcode)

# renaming field
colnames(buildingUnits)[colnames(buildingUnits)=="Postcode"] <- "Incident.Zip"

# joining
joined_data <- nyc311 %>% inner_join(buildingUnits, by = "Incident.Zip")
```

Filtering the joined data

Here we have filtered the data and selected the columns we would finally want to perform analysis on. The second dataset chosen is from the year beginning 2014. The NYC 311 calls dataset ranges from 2010 to 2014. So we have filtered and just kept 2014.

We have performed further grouping, for eg: group on zipcode and get number of noise complaints for each zipcode and then see if there is a relation with the total construction units.

```
# selection
joined_data <- select(joined_data, Agency, Complaint.Type, Descriptor, Incident.Zip, Borough, Status, C

joined_data_filter_backup <- joined_data

joined_data<-joined_data %>% separate(Created.Date, c("Date"), sep=" ",extra = "drop")
joined_data<-joined_data %>% separate(Date, c("Month", "Date", "Year"), sep="/")
joined_data<-filter(joined_data, Year == "2014")
```

Spreading Status in Joined Data

Here we are spreading the status column to find out the counts of each kind of status in the joined data so that we can then use it to create the grouped stacked chart in the next section

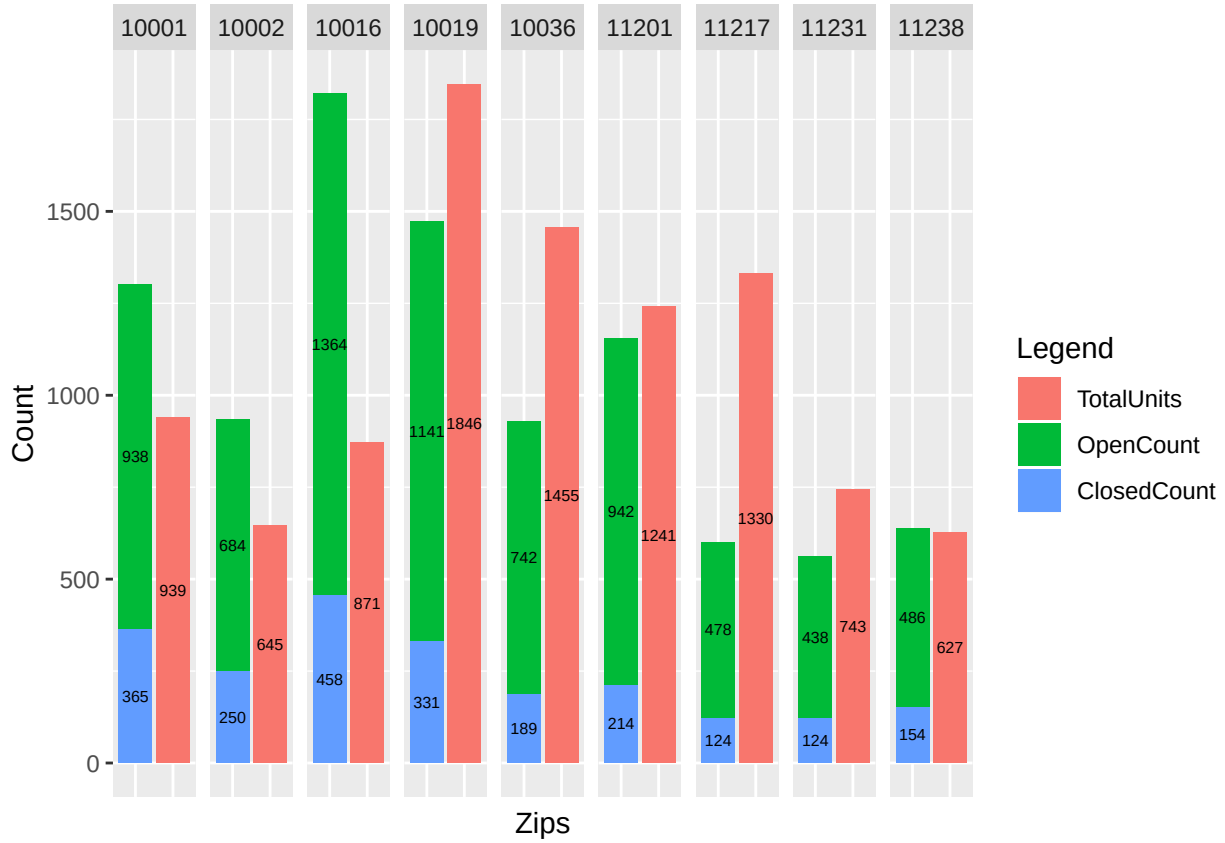
```
joined_data$StatusCount <- rep(1, nrow(joined_data))
joined_data$Status<-gsub("Started", "Open", joined_data$Status)
joined_data <- joined_data %>% spread(Status, StatusCount)
joined_data$Open[is.na(joined_data$Open)] <- 0
joined_data$Closed[is.na(joined_data$Closed)] <- 0
```

Grouped Stacked chart for noise complaints alongside total units

In this section we have plotted the total units alongside the status of the noise complaints. This showcases that number of construction units in a zip code somewhat does affect the noise complaints reported. We stacked the status bar to display the division of open and closed cases.

```
noise_data<-filter(joined_data, Complaint.Type == "NOISE")
zipComplaints <- noise_data %>%
   group_by(Incident.Zip) %>%
```

```
   dplyr::summarise(
      TotalUnits = mean(`TotalUnits`),
      OpenCount = sum(Open),
      ClosedCount = sum(Closed)
)

zipComplaints<-filter(zipComplaints,TotalUnits>400, (OpenCount + ClosedCount)>400)

#Chart
meltedZipComplaints <- melt(zipComplaints, "Incident.Zip")

meltedZipComplaints$Zips <- ''
meltedZipComplaints[meltedZipComplaints$variable == 'TotalUnits',]$Zips <- "TotalUnits"
meltedZipComplaints[meltedZipComplaints$variable != 'TotalUnits',]$Zips <- "ComplaintCount"
colnames(meltedZipComplaints)[colnames(meltedZipComplaints)=="variable"] <- "Legend"
colnames(meltedZipComplaints)[colnames(meltedZipComplaints)=="value"] <- "Count"

ggplot(meltedZipComplaints, aes(x = Zips, y = Count, fill = Legend)) +
   geom_bar(stat = 'identity', position = 'stack') + facet_grid(~ Incident.Zip) +
   geom_text(aes(label = Count), size = 2, position = position_stack(vjust = 0.5)) +
   theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())
```

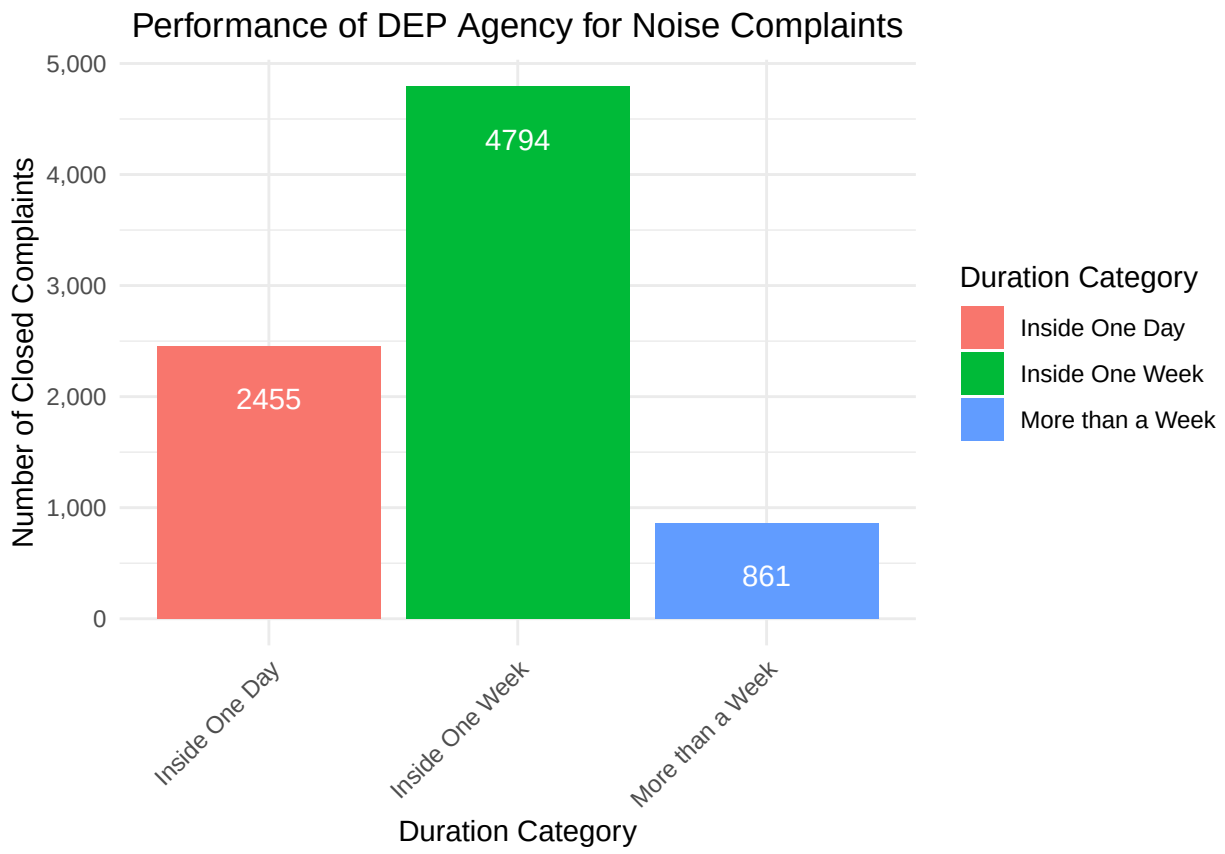


There seems to be a lot of open noise complaints. And DEP, the department taking care of these complaints should buck up. Let's see how quickly they are closing complaints.

How quickly are these complaints being taken care of?

Now that we have seen the above plot, lets look at how quickly these reported complaints are being taken care of.

```
# Filtering joined data
library(lubridate)
joined_data_filtered <- joined_data_filter_backup %>%
  filter(str_detect(Created.Date, "2014"), Status == "Closed", Complaint.Type == "NOISE")
joined_data_filtered$durations<-mdy_hms(joined_data_filtered$Closed.Date)-mdy_hms(joined_data_filtered$C
joined_data_filtered$durations<-round(as.numeric(joined_data_filtered$durations,units='days'),2)

# Categorizing durations
joined_data_durations <- joined_data_filtered %>%
    filter(durations>=0) %>%
    mutate(Duration.Category = case_when(
                                  durations<=1 ~ 'Inside One Day',
                                  durations <= 7 ~ 'Inside One Week',
                                  durations > 7 ~ 'More than a Week'
                                  ))

durationCategoryCounts<- joined_data_durations %>%
  group_by(Duration.Category) %>%
  dplyr::summarise(Count = n())

# Chart
ggplot(data=durationCategoryCounts, aes(x = Duration.Category, y = Count, fill=Duration.Category)) +
  geom_bar(stat="identity") +
   scale_y_continuous(labels = comma) +
   labs(title = "Performance of DEP Agency for Noise Complaints", x = "Duration Category", y = "Number
   geom_text(aes(label = Count), vjust = 3.0, color = "White") +
     theme_minimal() +
     theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle = 45, hjust = 1))
```



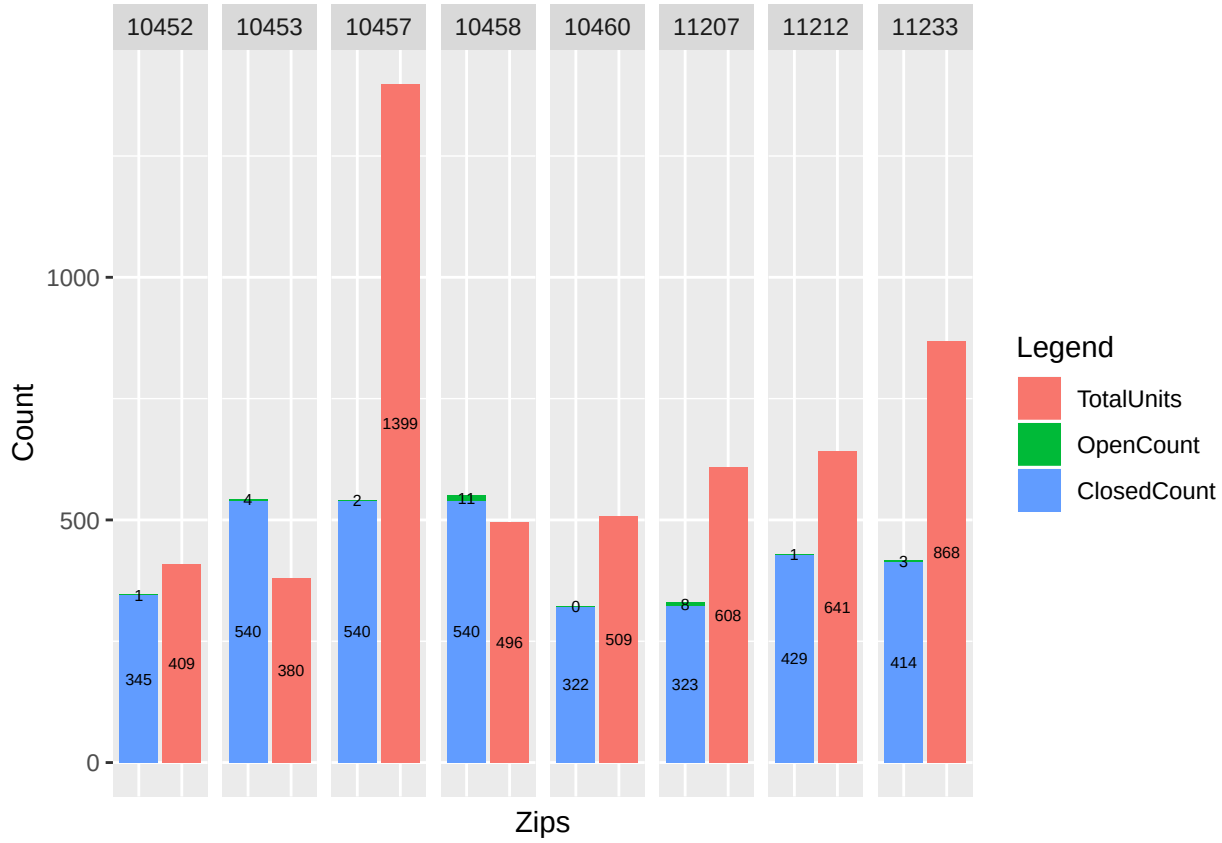
As we can see, most of the noise complaints in the zips where constructions (nyc building units) are in progress, have been resolved inside a week. There are too many open complaints though as we saw earlier.

Grouped Stacked chart for Construction complaints and total units

In this section we have plotted the total units alongside the status of the general construction complaints. This showcases that number of construction units in a zip code somewhat does affect the general construction complaints reported. We stacked the status bar to display the division of open and closed cases.

```
gc_data<-filter(joined_data, Complaint.Type == "GENERAL CONSTRUCTION")
zipComplaints <- gc_data %>%
   group_by(Incident.Zip) %>%
   dplyr::summarise(
      TotalUnits = mean(`TotalUnits`),
      OpenCount = sum(Open),
      ClosedCount = sum(Closed)
)

zipComplaints<-filter(zipComplaints,TotalUnits>300, (OpenCount + ClosedCount)>300)

#Chart
meltedZipComplaints <- melt(zipComplaints, "Incident.Zip")

meltedZipComplaints$Zips <- ''
meltedZipComplaints[meltedZipComplaints$variable == 'TotalUnits',]$Zips <- "TotalUnits"
```

```
meltedZipComplaints[meltedZipComplaints$variable != 'TotalUnits',]$Zips <- "ComplaintCount"
colnames(meltedZipComplaints)[colnames(meltedZipComplaints)=="variable"] <- "Legend"
colnames(meltedZipComplaints)[colnames(meltedZipComplaints)=="value"] <- "Count"

ggplot(meltedZipComplaints, aes(x = Zips, y = Count, fill = Legend)) +
   geom_bar(stat = 'identity', position = 'stack') + facet_grid(~ Incident.Zip) +
   geom_text(aes(label = Count), size = 2, position = position_stack(vjust = 0.5)) +
   theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())
```



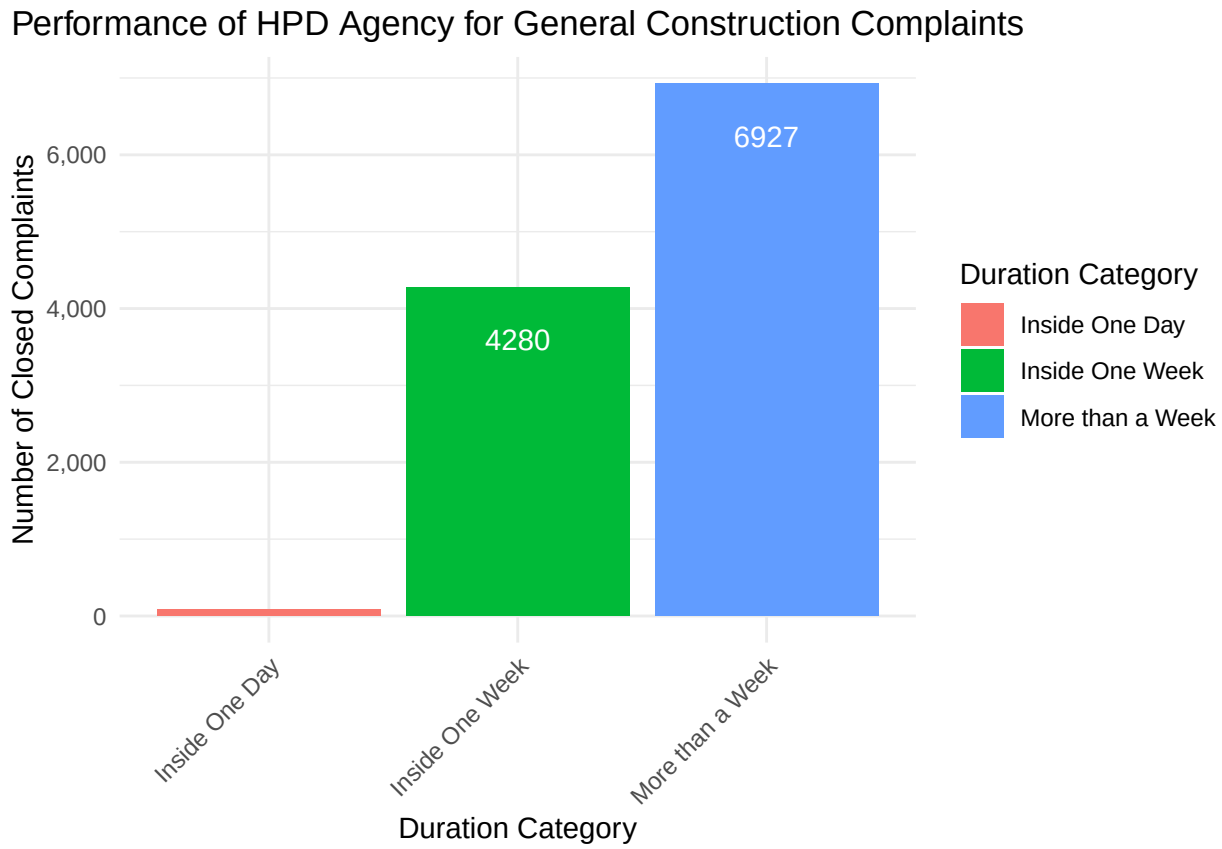
Maximum number of complaints have been closed. It will be interesting to see the rate at which these complaints are being closed.

How quickly are these complaints being taken care of?

Now that we have seen the above plot, lets look at how quickly these reported complaints are being taken care of.

```
# Filtering joined data
library(lubridate)
joined_data_filtered <- joined_data_filter_backup %>%
  filter(str_detect(Created.Date, "2014"), Status == "Closed", Complaint.Type == "GENERAL CONSTRUCTION"
joined_data_filtered$durations<-mdy_hms(joined_data_filtered$Closed.Date)-mdy_hms(joined_data_filtered$
joined_data_filtered$durations<-round(as.numeric(joined_data_filtered$durations,units='days'),2)

# Categorizing durations
```

```r
joined_data_durations <- joined_data_filtered %>%
    filter(durations>=0) %>%
    mutate(Duration.Category = case_when(
                                durations<=1 ~ 'Inside One Day',
                                durations <= 7 ~ 'Inside One Week',
                                durations > 7 ~ 'More than a Week'
                                ))

durationCategoryCounts<- joined_data_durations %>%
  group_by(Duration.Category) %>%
  dplyr::summarise(Count = n())

# Chart
ggplot(data=durationCategoryCounts, aes(x = Duration.Category, y = Count, fill=Duration.Category)) +
  geom_bar(stat="identity") +
   scale_y_continuous(labels = comma) +
   labs(title = "Performance of HPD Agency for General Construction Complaints", x = "Duration Category
   geom_text(aes(label = Count), vjust = 3.0, color = "White") +
     theme_minimal() +
     theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle = 45, hjust = 1))
```



As we can see, most of the general construction complaints in the zips where constructions (nyc building units) are in progress, have taken more than a week to be resolved. So the HPD agency, although has most of the complaints closed as we saw earlier, it still is not as expeditious.

APPENDIX

PART A - Data Dictionary [NYC311]

Unique Key - A key to uniquely identify each service request (SR)

Created Date - Date of creation of service request (Format: MM/DD/YY HH:MM:SS AM/PM)

Closed Date - Date on which the SR was closed ((Format: MM/DD/YY HH:MM:SS AM/PM))

Agency - The responding City Government Agency acronym

Agency Name - Full name of the responding City Government Agency

Complaint Type - Information about the topic of the incident or condition.

Descriptor - Dependent on the complaint type. Contains more information on the incident or condition.

Status - Status of SR submitted (Suggested values: Assigned, Cancelled, Closed, etc.)

Due Date - Date when SR is supposed to be updated (Format: MM/DD/YY HH:MM:SS AM/PM)

Resolution Action Updated Date - Date when responding agency last updated the SR. (Format: MM/DD/YY HH:MM:SS AM/PM)

Location Type - Describes the type of location used in the address information

Incident Zip - Incident location zip code, provided by geo validation.

Incident Address - House number of incident address provided by submitter.

Incident Code - Street Code

Street Name - Street name of incident address provided by the submitter

Cross Street 1 - First Cross street based on the geo validated incident location

Cross Street 2 - Second Cross Street based on the geo validated incident location

Intersection Street 1 - First intersecting street based on geo validated incident location

Intersection Street 2 - Second intersecting street based on geo validated incident location

Address Type - Type of incident location information available (Values: Address, Block face, Intersection, LatLong, Placename).

City - City of the incident location provided by geo validation.

Facility Type - If available, this field describes the type of city facility associated to the SR

Community Code - Code for Borough

Borough - Provided by the submitter and confirmed by geo validation.

X Coordinate (State Plane) - Geo validated, X coordinate of the incident location.

Y Coordinate (State Plane) - Geo validated, Y coordinate of the incident location.

Latitude - Geo based Lat of the incident location

Longitude - Geo based Long of the incident location

Location - Combination of the geo based lat & long of the incident location

Park Facility Name - If the incident location is a Parks Dept facility, the Name of the facility will appear here

Vehicle Type - If the incident is a taxi, this field describes the type of TLC vehicle.

Taxi Pick Up Location - If the incident is identified as a taxi, this field displays the taxi pick up location

Bridge Highway Name - If the incident is identified as a Bridge/Highway, the name will be displayed here.

Bridge Highway Direction - If the incident is identified as a Bridge/Highway, the direction where the issue took place would be displayed here.

Road Ramp - If the incident location was Bridge/Highway this column differentiates if the issue was on the Road or the Ramp.

Bridge Highway Segment - Additional information on the section of the Bridge/Highway were the incident took place.

PART B - Glimpse [NYC311]

```
glimpse(nyc311)
```

```
## Observations: 9,124,100
## Variables: 46
## Groups: Agency [30]
## $ Unique.Key                      <int> 30387854, 30388338, 30395236, 3...
## $ Created.Date                    <chr> "04/14/2015 02:14:40 AM", "04/1...
## $ Closed.Date                     <chr> "04/14/2015 03:03:22 AM", NA, N...
## $ Agency                          <chr> "NYPD", "NYPD", "NYPD", "NYPD",...
## $ Agency.Name                     <chr> "New York City Police Departmen...
## $ Complaint.Type                  <chr> "VENDING", "BLOCKED DRIVEWAY", ...
## $ Descriptor                      <chr> "In Prohibited Area", "No Acces...
## $ Location.Type                   <chr> "Street/Sidewalk", "Street/Side...
## $ Incident.Zip                    <chr> "10465", "11234", "11204", "112...
## $ Incident.Code                   <chr> "3775", "1524", NA, "361", NA, ...
## $ Street.Name                     <chr> "EAST TREMONT AVENUE", "RYDER S...
## $ Cross.Street.1                  <chr> "RANDALL AVENUE", "FLATLANDS AV...
## $ Cross.Street.2                  <chr> "ROOSEVELT AVENUE", "AVENUE P",...
## $ Intersection.Street.1           <chr> NA, NA, "71 STREET", NA, "WEST ...
## $ Intersection.Street.2           <chr> NA, NA, "16 AVENUE", NA, "COLUM...
## $ Address.Type                    <chr> "ADDRESS", "ADDRESS", "INTERSEC...
## $ City                            <chr> "BRONX", "BROOKLYN", "BROOKLYN"...
## $ Facility.Type                   <chr> "Precinct", "Precinct", "Precin...
## $ Status                          <chr> "Closed", "Open", "Open", "Assi...
## $ Due.Date                        <chr> "04/14/2015 10:14:40 AM", "04/1...
## $ Resolution.Action.Updated.Date <chr> "04/14/2015 03:03:05 AM", NA, N...
## $ Community.Code                  <chr> "10", "18", "11", "01", "07", "...
## $ Borough                         <chr> "BRONX", "BROOKLYN", "BROOKLYN"...
## $ `X.Coordinate.(State.Plane)`    <int> 1033758, 1001544, 984678, 99647...
## $ `Y.Coordinate.(State.Plane)`    <int> 240162, 164726, 164647, 199445,...
## $ Park.Facility.Name              <chr> "Unspecified", "Unspecified", "...
## $ School.Name                     <chr> "Unspecified", "Unspecified", "...
## $ School.Number                   <chr> "Unspecified", "Unspecified", "...
## $ School.Region                   <chr> "Unspecified", "Unspecified", "...
## $ School.Code                     <chr> "Unspecified", "Unspecified", "...
## $ School.Phone.Number             <chr> "Unspecified", "Unspecified", "...
## $ School.Address                  <chr> "Unspecified", "Unspecified", "...
## $ School.Not.Found                <chr> "N", "N", "N", "N", "N", "N", "...
## $ School.or.Citywide.Complaint    <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
```

```
## $ Vehicle.Type                  <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Taxi.Pick.Up.Location         <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Bridge.Highway.Name           <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Bridge.Highway.Direction      <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Road.Ramp                     <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Bridge.Highway.Segment        <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Garage.Lot.Name               <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Ferry.Direction               <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Ferry.Terminal.Name           <chr> NA, NA, NA, NA, NA, NA, NA, NA,...
## $ Latitude                      <dbl> 40.82573, 40.61879, 40.61859, 4...
## $ Longitude                     <dbl> -73.82111, -73.93771, -73.99846...
## $ Location                      <chr> "(40.8257259931145, -73.8211142...
```

PART C - Data Dictionary [JOINED DATA : NYC 311_HOUSING DATA]

Complaint.Type - Information about the topic of the incident or condition.

Descriptor - Dependent on the complaint type. Contains more information on the incident or condition.

Incident.Zip - Incident location zip code, provided by geo validation.

Borough - Provided by the submitter and confirmed by geo validation.

Status - Status of Service Request submitted (Suggested values: Assigned, Cancelled, Closed, etc.)

Created.Date - Date of creation of service request (Format: MM/DD/YY HH:MM:SS AM/PM)

Closed.Date - Date on which the SR was closed (Format: MM/DD/YY HH:MM:SS AM/PM)

Unique.Key - A key to uniquely identify each service request (SR)

TotalUnits - The Total Units field indicates the total number of units in each zip.

Latitude - Geo based Latitude of the incident location

Longitude - Geo based Longitude of the incident location

PART D - Glimpse [JOINED DATA : NYC 311_HOUSING DATA]

```
glimpse(joined_data_filter_backup)
```

```
## Observations: 5,086,050
## Variables: 12
## Groups: Agency [30]
## $ Agency         <chr> "NYPD", "NYPD", "NYPD", "NYPD", "DPR", "DOT", "...
## $ Complaint.Type <chr> "NOISE   STREET SIDEWALK", "NOISE   STREET SIDE...
## $ Descriptor     <chr> "Loud Talking", "Loud Talking", "Loud Talking",...
## $ Incident.Zip   <chr> "11211", "10025", "11205", "10001", "11213", NA...
## $ Borough        <chr> "BROOKLYN", "MANHATTAN", "BROOKLYN", "MANHATTAN...
## $ Status         <chr> "Assigned", "Closed", "Closed", "Open", "Open",...
## $ Created.Date   <chr> "04/14/2015 02:02:40 AM", "04/14/2015 02:00:04 ...
## $ Closed.Date    <chr> NA, "04/14/2015 02:47:33 AM", "04/14/2015 02:11...
## $ Unique.Key     <int> 30394595, 30390517, 30389560, 30388819, 3038782...
## $ TotalUnits     <int> 154, 74, 462, 939, 95, 434, 353, 1330, 1455, 43...
## $ Latitude       <dbl> 40.71410, 40.79792, 40.68833, 40.74805, NA, NA,...
## $ Longitude      <dbl> -73.95589, -73.96385, -73.96481, -74.00104, NA,...
```