title: “homework v”
author: “Sidhartha Amperayani, Moinuddin Memon, Amritha Menon”
date: “July 22, 2022”
output:
pdf_document:
latex_engine: lualatex
html_document:
df_print: paged
editor_options:
markdown:
wrap: sentence

INTRODUCTION

In this assignment, we further explore the NYC 311 Service Request dataset. We find meaningful connections between the column of the dataset. Through the analysis and visualizations we make, we seek the answers for the following questions :

1. What is the density distribution of the top 5 complaints?
2. Which agency has the most number of closed cases?
3. In which Borough does NYPD respond to the most?
4. How are different types of noise complaints distributed in each Borough?
5. Which month has the most service requests overall?
6. Which agency has the most pending cases?
7. Which agency takes the most time to resolve the service requests?
8. During which hour is the maximum service request reported?
9. Which are the most frequent word-sets in the description of a Service Request?

PARAMETERS

All the parameters required for this file are initialized here like if the file name, sample number of rows, if a full run or test run must be made and also the key for the Google map API.

```
mainFile <- '311_Service_Requests_from_2010_to_Present.csv'
#sampleFile <- 'mini311.csv'
sampleRows <- 400000
testRun <- FALSE
key <- "AIzaSyBn71X_SGI5wnpZvQLcFhC7dacG-DQXaHw"
```

INITIALIZATION

The packages to be used like tidyverse, ggplot, dplyr etc. are installed. The program runs according to the user’s setting of ***testRun*** variable. All the irrelevant columns are removed and the duplicate rows are also removed. In the ***Borough*** column the ***Unspecified*** value is changed to the respective borough name based on the given zip codes and our knowledge of NYC.

```r
packages <-
c(
"tidyverse",
"data.table",
#"devtools",
"ggmap",
"dplyr",
'ggplot2',
'tidyr',
'plyr',
'lubridate',
"xtable"
)
for (package in packages) {
library(package, character.only = TRUE)
}
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.9
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

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

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

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

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

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

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

##
## 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 objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
if (testRun){
  print(paste0('Reducing dataset size to ',sampleRows))
  df <- fread(mainFile, nrows=sampleRows)
} else {
  df <- fread(mainFile)
}

names(df) <-names(df) %>% stringr::str_replace_all("\\s", "")
df <- subset(df, select = -c(UniqueKey,City,LocationType,IncidentAddress,CrossStreet1, CrossStreet2, Int
nyc_with_dups<-nrow(df)
df <- df %>% distinct()
nyc_without_dups<-nrow(df)
nyc_with_dups-nyc_without_dups
```

```
## [1] 872266
```

```
df$IncidentZip <- as.integer(df$IncidentZip)
```

```
## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion to integer range
```

```
df$Borough[df$IncidentZip>=10001 & df$IncidentZip<10283]<-"MANHATTAN"
df$Borough[df$IncidentZip>=10301 & df$IncidentZip<10315]<-"STATEN ISLAND"
df$Borough[df$IncidentZip>=10451 & df$IncidentZip<10476]<-"BRONX"
df$Borough[df$IncidentZip>=11004 & df$IncidentZip<11110]<-"QUEENS"
```

```
df$Borough[df$IncidentZip>=11351 & df$IncidentZip<11698]<-"QUEENS"
df$Borough[df$IncidentZip>=11201 & df$IncidentZip<11257]<-"BROOKLYN"

library(xtable)
print(xtable(head(df)))
```

```
## % latex table generated in R 4.1.1 by xtable 1.8-4 package
## % Fri Jul 22 19:52:51 2022
## \begin{table}[ht]
## \centering
## \begin{tabular}{rllllllrllllrrrr}
##   \hline
##  & CreatedDate & ClosedDate & Agency & AgencyName & ComplaintType & Descriptor & IncidentZip & Stree
##   \hline
## 1 & 04/14/2015 02:14:40 AM & 04/14/2015 03:03:22 AM & NYPD & New York City Police Department & Vendi
73.82 \\
##   2 & 04/14/2015 02:10:12 AM &  & NYPD & New York City Police Department & Blocked Driveway & No Acc
73.94 \\
##   3 & 04/14/2015 02:03:01 AM &  & NYPD & New York City Police Department & Noise - Street/Sidewalk &
74.00 \\
##   4 & 04/14/2015 02:02:40 AM &  & NYPD & New York City Police Department & Noise - Street/Sidewalk &
73.96 \\
##   5 & 04/14/2015 02:00:04 AM & 04/14/2015 02:47:33 AM & NYPD & New York City Police Department & Nois
73.96 \\
##   6 & 04/14/2015 01:52:15 AM & 04/14/2015 02:11:10 AM & NYPD & New York City Police Department & Nois
73.96 \\
##    \hline
## \end{tabular}
## \end{table}
```

```
date<-parse_datetime(df$CreatedDate,format="%m/%d/%Y %H:%M:%S %p")
df[,"Created_year"]<-format(date,"%y")
df[,"Created_month"]<-(format(date,"%m"))
df[,"Created_day"]<-as.integer(format(date,"%d"))
df[,"Created_hour"]<-as.integer(format(date,"%H"))
df[,"Created_minute"]<-as.integer(format(date,"%M"))
df$HM<-paste(df$Created_hour,df$Created_minute)
date<-parse_datetime(df$ClosedDate,format="%m/%d/%Y %H:%M:%S %p")
df[,"Closed_month"]<-(format(date,"%m"))
df[,"Closed_day"]<-as.integer(format(date,"%d"))
df[,"Closed_hour"]<-as.integer(format(date,"%H"))
```

NYPD Complaints 2010-2015

To help us better understand the the way a how the administration works, we import the NYPD complaints data-set which represents all the crime reported to NYPD.We sampled this from NYC Open Data, which can be found here: https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i

The dataset's attributes describes details related to the type of crime (valid felony, misdemeanor, and violation crimes) that has been reported to the New York City Police Department from 2006 to 2019. Since the 311 Service Request data contains data between the years 2010 and 2015, we have filtered crimes reported between the years 2010 and 2015 from NYPD Complaints dataset.

This dataset includes date, location of the crime, victims ad suspects age group and other relevant features pertaining to the crime.

We clean the data by removing unnecessary columns like Unique Key, Date, Time(Since we already have the Date Time Column) and filtering rows that have a report between the year range 2010 to 2015. After cleaning the NYPD complaint data, it forms a table of nrow(alt_data)rows, which contain columns described as follows:

Data Dictionary

- **DATE_TIME(Type: Date)**
  -Exact date time of Inquiry to the call center
- **ADDR_PCT_CD**
  - The precinct in which the incident occurred
- **OFNS_DESC(Type: CHAR)**
  -Description of offense corresponding with key code
- **LAW_CAT_CD(TYPE: CHAR)**
  - Level of offense: felony, misdemeanor, violation
- **BORO_NM(TYPE:CHAR)**
  - The name of the borough in which the incident occurred
- **JURIS_DESC(TYPE:CHAR)**
  - Description of the jurisdiction code
- **Latitude(TYPE: FLOAT)**
  - Midblock Latitude coordinate for Global Coordinate System, WGS 1984, decimal degrees
- **Longitude(TYPE: FLOAT)**
  - Midblock Longitude coordinate for Global Coordinate System, WGS 1984, decimal degrees
- **SUSP_AGE_GROUP(TYPE: CHAR)**
  - Suspect's Age Group
- **VIC_AGE_GROUP(TYPE: CHAR)**
  - Victim's Age Group
- How the call was resolved.(For eg. CSMS SR-Call was resolved by submission of a Service Request)

Custom Columns

- **Day(Type: Number)**
  -Integer representing the Day, derived from Date_Time
- **Month(Type: Number)**
  -Integer representing the month, derived from Date_Time
- **Year(Type: Number)**
  -Integer representing the starting Year, derived from Date_Time

```
alt_data<-fread('NYPD_Complaint_Data_Historic.csv')
alt_data <- subset(alt_data, select = -c(CMPLNT_NUM,CMPLNT_TO_DT,CMPLNT_TO_TM,RPT_DT,
                                       KY_CD,PD_CD,CRM_ATPT_CPTD_CD,JURIS_DESC   ,JURISDICTION_CODE,TRA
                                       ADDR_PCT_CD,LOC_OF_OCCUR_DESC,PREM_TYP_DESC,
                                       PARKS_NM,HADEVELOPT,X_COORD_CD,Y_COORD_CD,Lat_Lon))
alt_data<-alt_data %>% distinct()
alt_data$CMPLNT_FR_DT<-parse_date_time(alt_data$CMPLNT_FR_DT, "%m/%d/%y")
alt_data <- mutate(alt_data,
  DAY = as.integer(format(alt_data$CMPLNT_FR_DT, format="%d")),
  MONTH = as.integer(format(alt_data$CMPLNT_FR_DT, format="%m")),
  YEAR = as.integer(format(alt_data$CMPLNT_FR_DT, format="%y")),
)
alt_data <- alt_data[alt_data$YEAR <= 15 & alt_data$YEAR >= 10]
```

MERGING DATA

Join on Borough, Year, Month, Day

```
#SNAPSHOT OF AGENCY WIDE COMPLAINTS FROM THE NYC311 DATASET
nyc311sum <- df %>%
  select(Agency, Borough, Created_day, Created_month, Created_year, ComplaintType) %>%
  filter(Agency=="NYPD", Created_year>=10)
nyc311sum <- nyc311sum %>%
  group_by(Borough, Created_year, Created_month, Created_day) %>%
  dplyr::summarize(Total_Complaints=n())
```

```
## `summarise()` has grouped output by 'Borough', 'Created_year', 'Created_month'.
## You can override using the `.groups` argument.
```

```
altdatasum <- alt_data %>%
  select(BORO_NM, YEAR, MONTH, DAY, LAW_CAT_CD) %>%
  filter(BORO_NM!="")
#Fixing the name of columns for simplicity
names(altdatasum)[names(altdatasum)=="BORO_NM"] <- "Borough"
names(altdatasum)[names(altdatasum)=="YEAR"] <- "Year"
names(altdatasum)[names(altdatasum)=="MONTH"] <- "Month"
names(altdatasum)[names(altdatasum)=="DAY"] <- "Day"
names(nyc311sum)[names(nyc311sum)=="Created_day"] <- "Day"
names(nyc311sum)[names(nyc311sum)=="Created_year"] <- "Year"
names(nyc311sum)[names(nyc311sum)=="Created_month"] <- "Month"
nyc311sum$Year <- (as.integer(nyc311sum$Year))
nyc311sum$Month <- (as.integer(nyc311sum$Month))
nyc311sum$Day <- (as.integer(nyc311sum$Day))
altdatasum <- altdatasum %>%
  group_by(Borough, Year, Month, Day) %>%
  dplyr::summarize(Total_Offenses=n())
```

```
## `summarise()` has grouped output by 'Borough', 'Year', 'Month'. You can
## override using the `.groups` argument.
```

```
altdatasum$Month[altdatasum$Month==1] <- "Jan"
altdatasum$Month[altdatasum$Month==2] <- "Feb"
altdatasum$Month[altdatasum$Month==3] <- "Mar"
altdatasum$Month[altdatasum$Month==4] <- "Apr"
altdatasum$Month[altdatasum$Month==5] <- "May"
altdatasum$Month[altdatasum$Month==6] <- "Jun"
altdatasum$Month[altdatasum$Month==7] <- "Jul"
altdatasum$Month[altdatasum$Month==8] <- "Aug"
altdatasum$Month[altdatasum$Month==9] <- "Sep"
altdatasum$Month[altdatasum$Month==10] <- "Oct"
altdatasum$Month[altdatasum$Month==11] <- "Nov"
altdatasum$Month[altdatasum$Month==12] <- "Dec"
nyc311sum$Month[nyc311sum$Month==1] <- "Jan"
nyc311sum$Month[nyc311sum$Month==2] <- "Feb"
nyc311sum$Month[nyc311sum$Month==3] <- "Mar"
nyc311sum$Month[nyc311sum$Month==4] <- "Apr"
nyc311sum$Month[nyc311sum$Month==5] <- "May"
nyc311sum$Month[nyc311sum$Month==6] <- "Jun"
nyc311sum$Month[nyc311sum$Month==7] <- "Jul"
nyc311sum$Month[nyc311sum$Month==8] <- "Aug"
nyc311sum$Month[nyc311sum$Month==9] <- "Sep"
nyc311sum$Month[nyc311sum$Month==10] <- "Oct"
nyc311sum$Month[nyc311sum$Month==11] <- "Nov"
nyc311sum$Month[nyc311sum$Month==12] <- "Dec"
# By Borough total
joined_data <- inner_join(nyc311sum, altdatasum, by=c("Borough", "Year", "Month","Day"))
joined_data<-filter(joined_data,Borough!="Unspecified")


joined_data_borough <-  joined_data %>%
  group_by(Borough) %>%
  dplyr::summarize(Total_Complaints=sum(Total_Complaints), Total_Offenses=sum(Total_Offenses))

# By year
joined_data_year <- joined_data %>%
  group_by(Borough, Year) %>%
  dplyr::summarize(Total_Complaints=sum(Total_Complaints), Total_Offenses=sum(Total_Offenses))
```

```
## `summarise()` has grouped output by 'Borough'. You can override using the
## `.groups` argument.
```

```
# By year,month total
joined_data_year_total <- joined_data %>%
  group_by(Year,Month) %>%
  dplyr::summarize(Total_Complaints=sum(Total_Complaints), Total_Offenses=sum(Total_Offenses))
```

```
## `summarise()` has grouped output by 'Year'. You can override using the
## `.groups` argument.
```

Head Of Joined Data

```
print(xtable(head(joined_data)))
```

```
## % latex table generated in R 4.1.1 by xtable 1.8-4 package
## % Fri Jul 22 19:54:17 2022
## \begin{table}[ht]
## \centering
## \begin{tabular}{rlrlrrr}
##   \hline
##  & Borough & Year & Month & Day & Total\_Complaints & Total\_Offenses \\
##   \hline
## 1 & BRONX &  10 & Jan &   1 &  36 & 462 \\
##   2 & BRONX &  10 & Jan &   2 &  32 & 264 \\
##   3 & BRONX &  10 & Jan &   3 &  28 & 264 \\
##   4 & BRONX &  10 & Jan &   4 &  37 & 237 \\
##   5 & BRONX &  10 & Jan &   5 &  31 & 303 \\
##   6 & BRONX &  10 & Jan &   6 &  26 & 321 \\
##    \hline
## \end{tabular}
## \end{table}
```

Borough name, year, month and, day are the attributes that as a team we collectively decided to join the datasets upon. With the help of this join, it would enable us to make a variety of informative visualizations.

First, a snapshot of the agency wise complaints from the 311 dataset with the attributes Agency, Borough, Created_day, Created_month, Created_year, ComplaintType and filtered it by just taking NYPD complaints with years greater than or equal to 2010. Then the data was grouped based on the total complaints for each row. A subset of the NYPD complaint dataset was also taken with the attributes BORO_NM, YEAR, MONTH, DAY, LAW_CAT_CD. The column names of this dataset was changed to the name of corresponding column name of th 311 dataset. All the months in both the datasets were converted from numbers to the respective months. The two datasets were joined using left join by the Borough, Year and, Month. The total number of offense were calculated by the same Borough, Year and, Month and the joined dataset is grouped by the total number of offenses. So now joined_data contains both the dataset.

TOP 5 COMPLAINTS ACROSS NYC

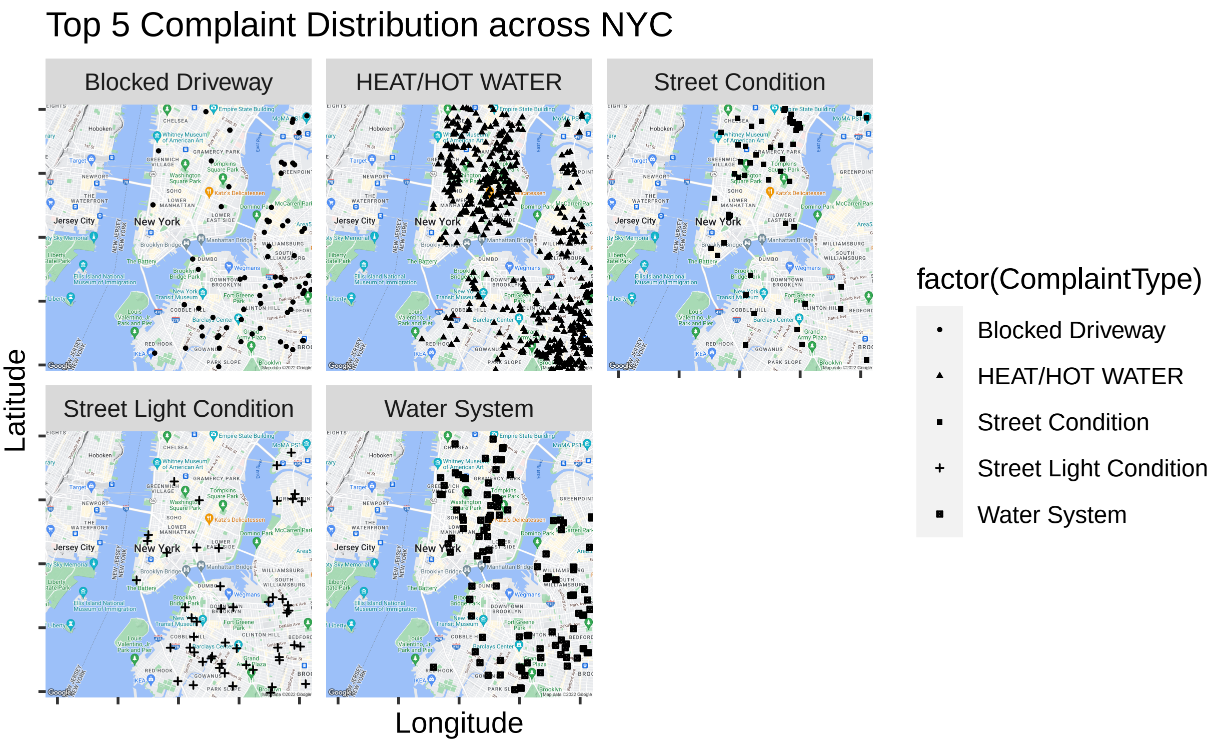
```
maps_df <- fread(mainFile, nrows=sampleRows)
names(maps_df) <-names(maps_df) %>% stringr::str_replace_all("\\s", "")
# getting a subset of the original dataset
maps_df.sub <- subset(maps_df, ComplaintType %in% dplyr::count(maps_df, ComplaintType, sort=T)[1:5]$Comp
maps_df.sub <- maps_df.sub %>% select(ComplaintType, Latitude, Longitude) %>% drop_na()
# getting the frequency of each complaint
counts <- ddply(maps_df.sub, .(ComplaintType), "count")
counts <- filter(counts, freq > 2)
counts$freq <- as.numeric(counts$freq)
counts$Longitude <- as.numeric(counts$Longitude)
counts$Latitude <- as.numeric(counts$Latitude)
# google map api
ggmap::register_google(key = key)
```

```
# map of NYC
nyc_map <- get_map(location = c(lon= -74.00, lat = 40.71), maptype = "terrain", zoom = 13)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=40.71,-74&zoom=13&size=640x640&scale=
EN&key=xxx-DQXaHw
```

```
ggmap(nyc_map)+ geom_point(data = counts, aes(x=Longitude,y=Latitude, shape=factor(ComplaintType)), siz
ggtitle('Top 5 Complaint Distribution across NYC') + theme(axis.text.y = element_blank(),axis.text.x =
```

```
## Warning: Removed 8204 rows containing missing values (geom_point).
```

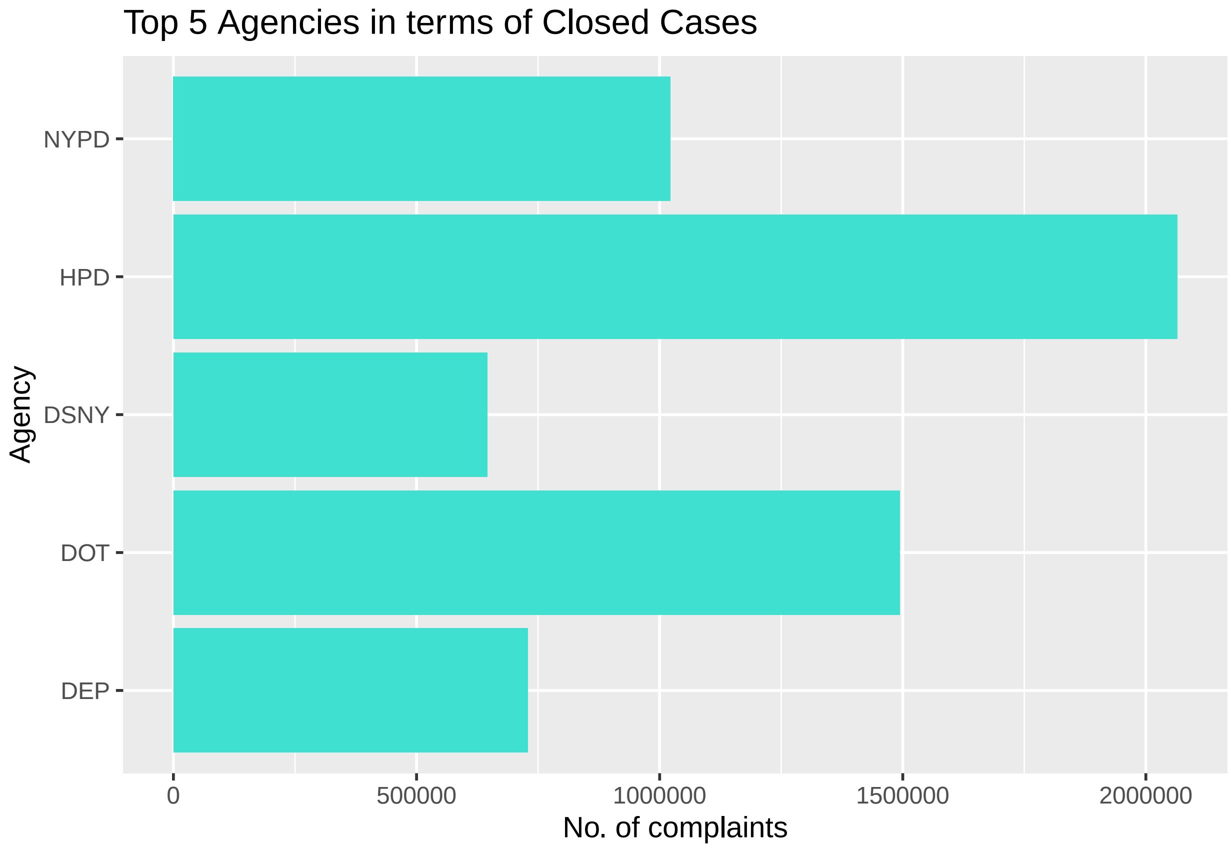


1. We can see that the top 5 complaints are Heat/Hot water, Water System, Blocked Driveway, Street Condition, and Street Light Conditions are the top 5 complaints in that order, based on their density distribution across the NYC map.

TOP 5 AGENCIES (BASED ON CLOSED CASES)

```
# selecting only closed cases
resolved_cases=df[df$ClosedDate!="",]
ggplot(subset(resolved_cases, Agency %in% dplyr::count(df, Agency, sort = T)[0:5,]$Agency), aes(Agency)
  coord_flip()+
  labs(title = "Top 5 Agencies in terms of Closed Cases")
```

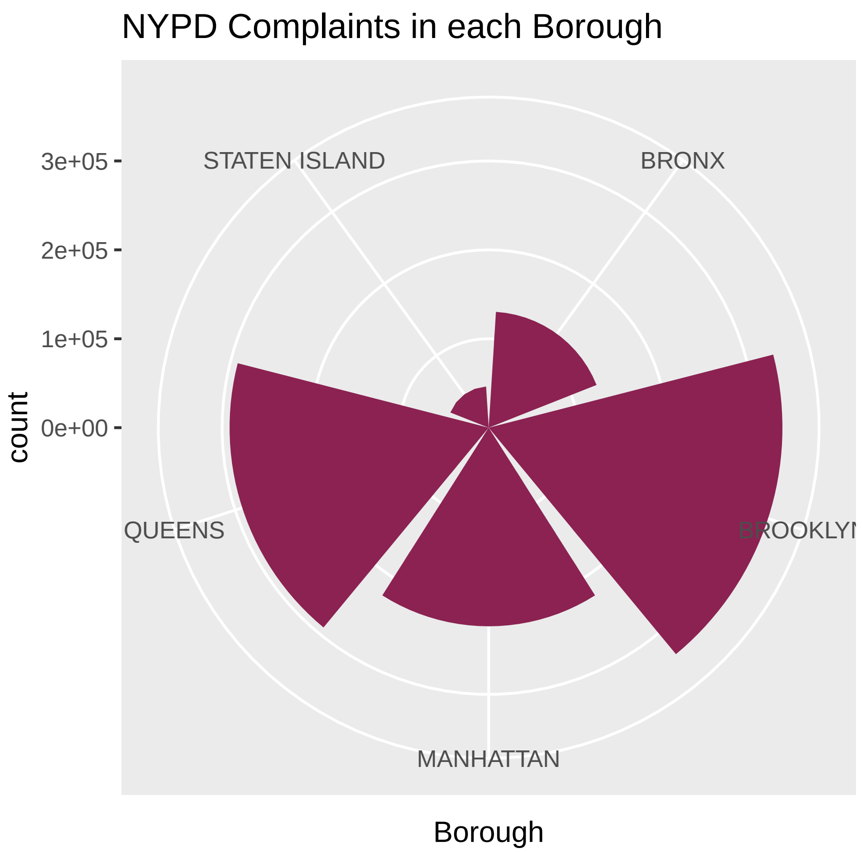
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



2. The HPD department, (Department of Housing, Preservation and Development) is the Agency which has closed the most cases, followed by DOT (Department of Transportation).

NYPD COMPLAINTS IN EACH BOROUGH

```r
# picking NYPD cases
nypd_cases=df[df$Agency=="NYPD",]
# filtering out Unspecified from each Borough
nypd_cases.sub <- nypd_cases %>%
  filter(Borough %in% c('BRONX', 'QUEENS','STATEN ISLAND', 'BROOKLYN','MANHATTAN'))
ggplot(nypd_cases.sub, aes(Borough))+geom_bar(stat ="count", fill="violetred4")+
coord_polar(theta = 'x') + ggtitle('NYPD Complaints in each Borough')
```



3. The Borough - Brooklyn has a lot of cases to which NYPD needs to respond.

NOISE COMPLAINT DISTRIBUTION IN EACH BOROUGH

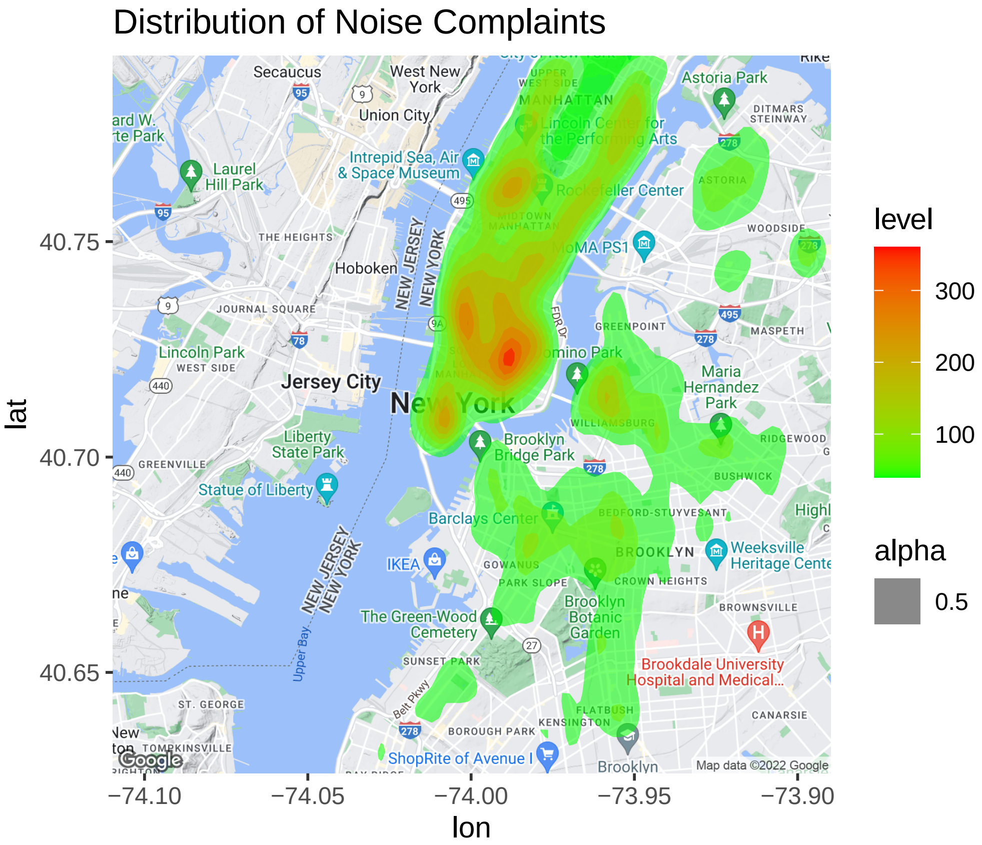
```
# filtering out the different noise complaints
dataset <- df %>% filter(ComplaintType %in% c("Noise", "Noise - Commercial", "Noise - Helicopter", "Noi

nyc_map <- get_map(location = c(lon= -74.0, lat = 40.71), maptype = "terrain", zoom = 12)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=40.71,-74&zoom=12&size=640x640&scale=
EN&key=xxx-DQXaHw
```

```
ggmap(nyc_map) + stat_density2d(
    aes(x = Longitude, y = Latitude, fill = ..level.., alpha = 0.5),
    size = 0.01, bins = 10, data = dataset,
    geom = "polygon"
  ) + scale_fill_gradient(low = "green", high = "red") +
  ggtitle("Distribution of Noise Complaints")
```

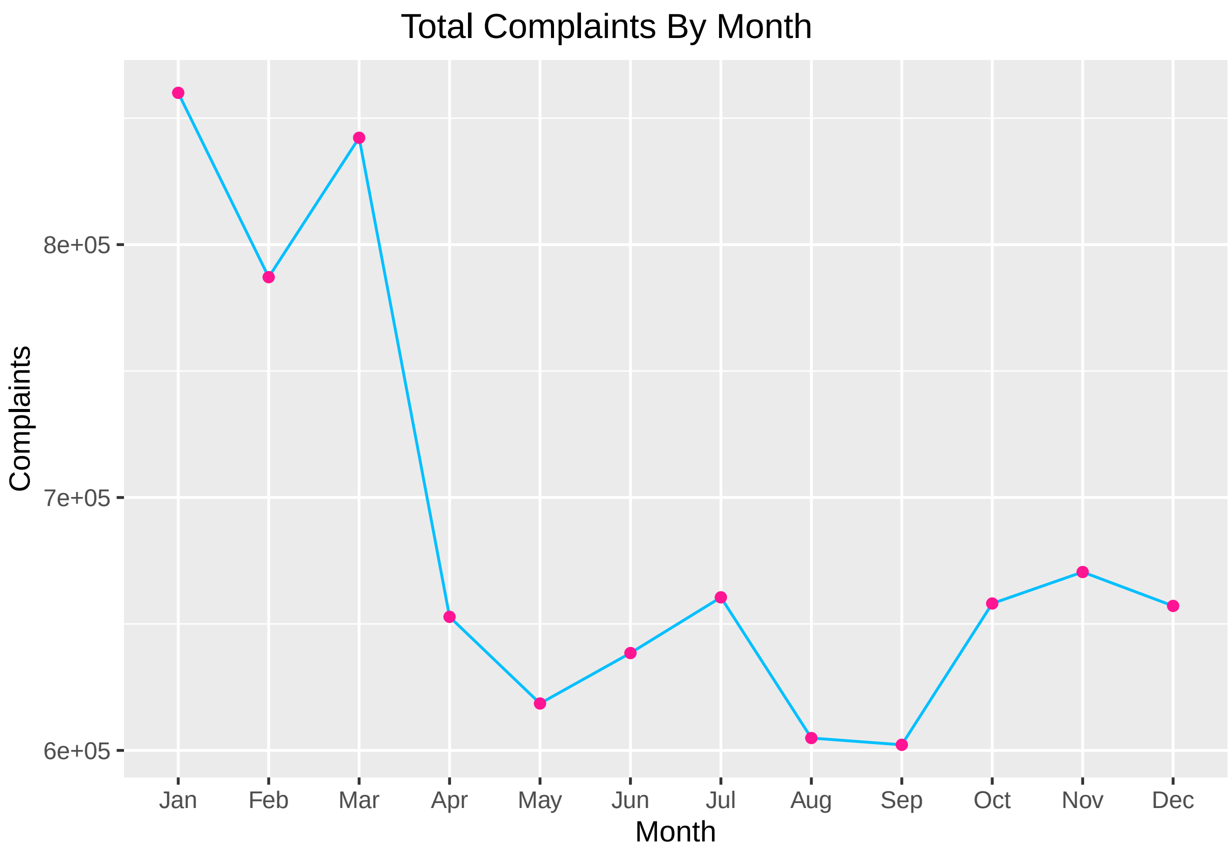
```
## Warning: Removed 228577 rows containing non-finite values (stat_density2d).
```



4. From the different types of noise complaints, we can see that helicopter noise, noise from house of worship, parks, streets and vehicles are on the lower side. But commercial noise and other noise complaints seem to be higher, especially in Manhattan. Staten Island and Bronx seem to have less number of noise complaints.

TOTAL NUMBER OF COMPLAINTS, MONTH WISE

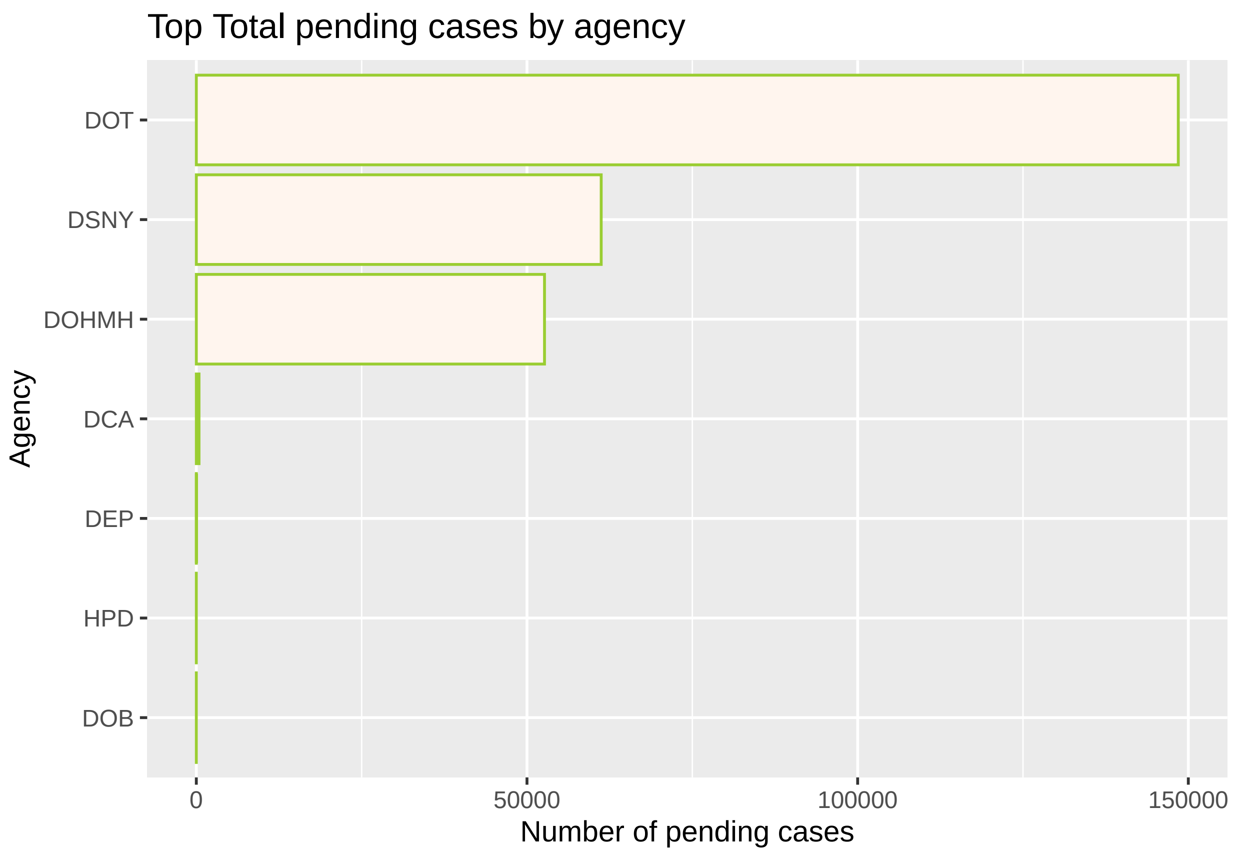
```r
# creating a sub dataset and grouping it by months
date_request<-
  subset(df,select=c(CreatedDate,ComplaintType,Borough)) %>%
  mutate(Month=month(mdy_hms(CreatedDate))) %>%
  group_by(Month) %>%
  dplyr::summarize(Complaints=n())
date_request$Month<- as.factor(month.abb[date_request$Month])
date_request$Month <- factor(date_request$Month, levels = date_request$Month)
ggplot(date_request, aes(x=Month, y= Complaints) ) +
    xlab("Month") +
    geom_line(aes(group=1),color='deepskyblue') +
    geom_point(color='deeppink')+
    labs(title="Total Complaints By Month")+
    theme(plot.title = element_text(hjust = 0.4))
```



5. From the line graph, it is seen that the total complaints are higher during the month of January.

PENDING CASES IN EACH AGENCY

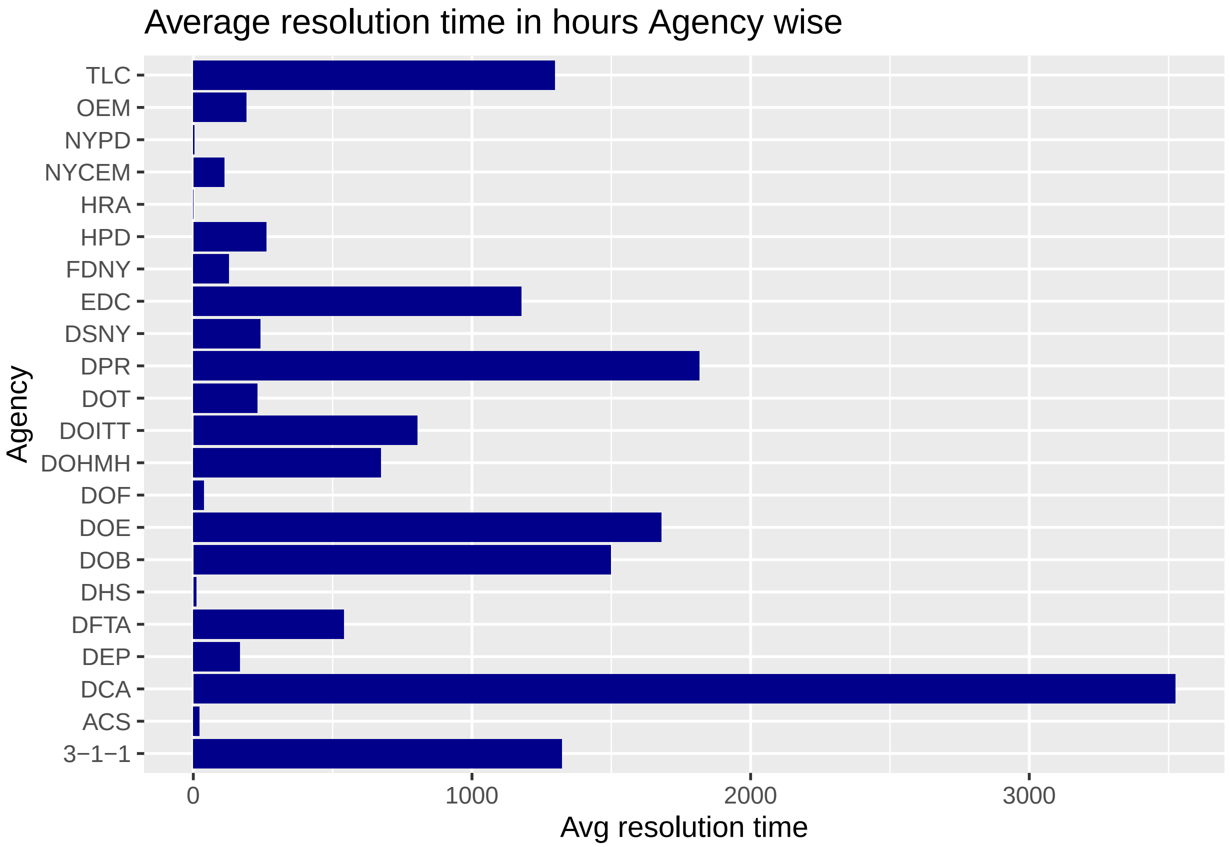
```
pend <- select(df, Agency, Status)
# filtering out pending cases
pend <- filter(pend, Status=="Pending")
# grouping by top 5 agencies
pending <-pend %>% group_by(Agency) %>% dplyr::summarize(count=n())
pending$Agency <- factor(pending$Agency,
                       levels=pending$Agency[order(pending$count,
                                                  decreasing=FALSE)])
(ggplot(pending, aes(x=Agency, y=count))+
    geom_bar(stat="identity", color ='yellowgreen',fill="seashell")+ coord_flip()+
    ggtitle(label="Top Total pending cases by agency")+
    ylab("Number of pending cases"))
```



6. The agency with the most pending cases is DOT (Department of Transportation).

AVERAGE RESOLUTION TIME, AGENCY WISE

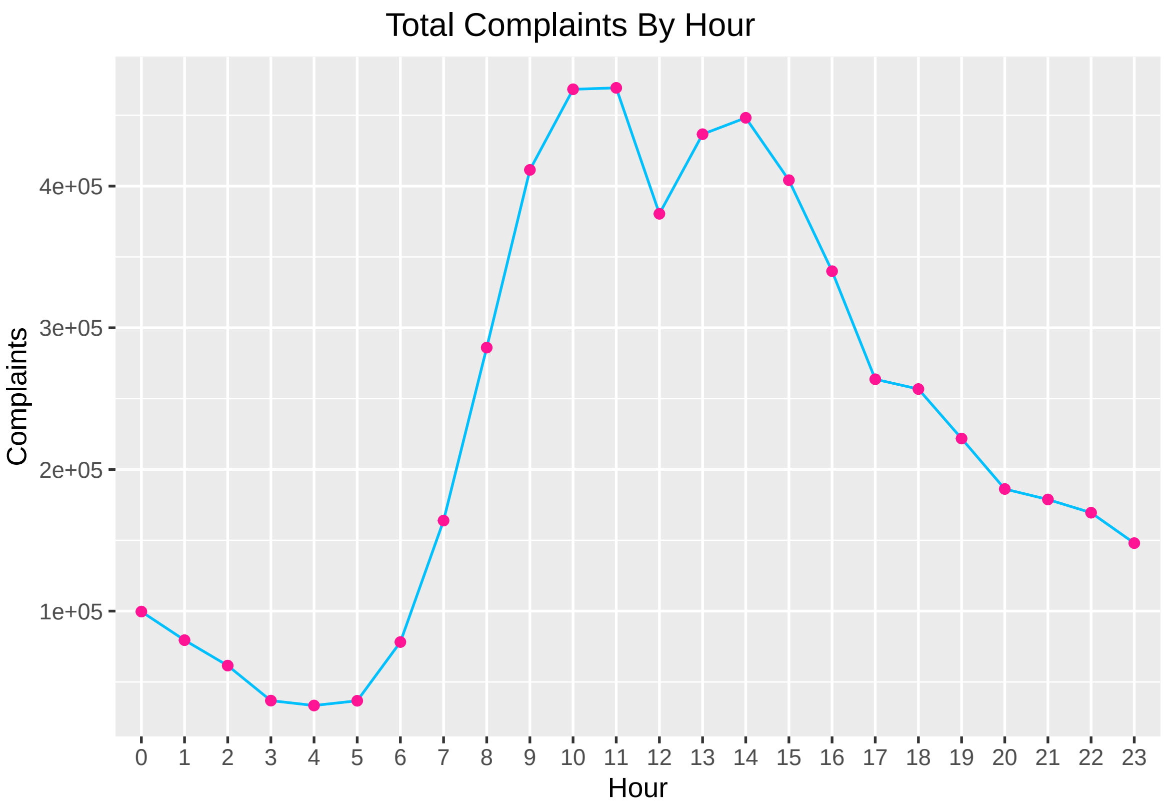
```
# calculating the resolution time in hours
resolution_hrs<- mdy_hms(df$ClosedDate)-mdy_hms(df$CreatedDate)
resolution_hrs<- round(as.numeric(resolution_hrs,units="hours"),2)
df$ResolutionTimeHrs<-resolution_hrs
# creating a subset to get avg resolution time
new_df <- df[df$ResolutionTimeHrs>0.0,] %>% subset(select=c(Agency, ResolutionTimeHrs))
new_df <- new_df %>% filter(!is.na(ResolutionTimeHrs))
new_df2 <- new_df %>% group_by(Agency) %>%
  dplyr::summarize(avgresptime = mean(ResolutionTimeHrs))
ggplot(new_df2, aes(Agency, avgresptime)) + geom_bar(stat='identity', fill='darkblue')+ggtitle(label="A
    ylab('Avg resolution time') + coord_flip()
```



7. From the average resolution time (in hours) calculated for each agency, it is seen that the Agency DCA (Department of Consumer Affairs) has the highest resolution time.

Complaints by the hour

```
complaintHour <- df[df$HM!="0 0"] %>%
 # group_by(Hour=hour(mdy_hms(CreatedDate))) %>%
#  dplyr::summarize(Complaints=n())
#ggplot(complaintHour,aes(x=Hour,y=count,group=1))+
#  geom_line(color="green")+
#  geom_point(color="green")+
#  labs(title = "Complaint By the hour",x="Hour",y="Count")
  group_by(Created_hour) %>%
  dplyr::summarize(Complaints=n())
complaintHour$Created_hour<- as.factor(complaintHour$Created_hour)
complaintHour$Created_hour <- factor(complaintHour$Created_hour, levels = complaintHour$Created_hou
ggplot(complaintHour, aes(x=Created_hour, y= Complaints) ) +
    xlab("Hour") +
    geom_line(aes(group=1),color='deepskyblue') +
    geom_point(color='deeppink')+
    labs(title="Total Complaints By Hour")+
    theme(plot.title = element_text(hjust = 0.4))
```



8.We can find in the pattern of the number of complaints reported against the hours in a day that the peak time of the reporting is between 9-12. We also removed the auto-generated complaint times which were set to 0 hour by default by chechking against the minute value, If it was also 0 then it was by default and can be removed.

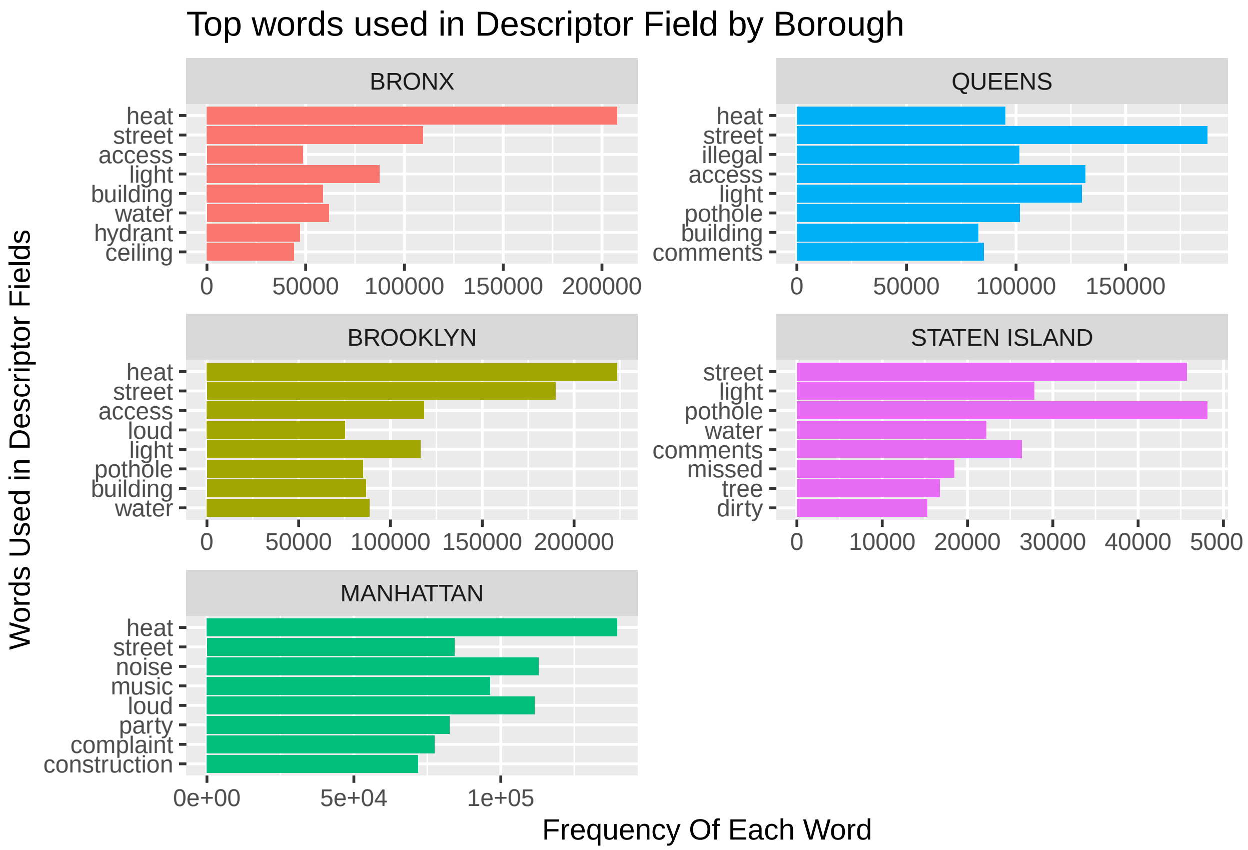
Top Complaint Words

```
library("stopwords")
library(tidyverse)
library(tidytext)
data(stop_words)
tokenized_desc <- df %>%
  select(ComplaintType, Descriptor,Borough) %>%
  filter(!str_detect(Borough,"Unspecified")) %>%
  filter(!str_detect(Descriptor,"NA")) %>%
  unnest_tokens(word, Descriptor) %>%
  anti_join(stop_words) %>%
  group_by(Borough, word) %>%
  tally()
```

```
## Joining, by = "word"
```

```
tokenized_desc %>%
  group_by(Borough) %>%
  top_n(8) %>%
  arrange(desc(n)) %>%
  ggplot(aes(x = reorder(word,n), y = n, fill = factor(Borough))) +
  geom_bar(stat = "identity") +
  theme(legend.position = "none") +
  facet_wrap(~Borough, scales = "free",dir="v") +
  coord_flip() +
  labs(x = "Words Used in Descriptor Fields",
       y = "Frequency Of Each Word",
       title = "Top words used in Descriptor Field by Borough")
```

```
## Selecting by n
```



9.We follow the word counts in the descriptor field of the data-set to get insights about how the complaints are reported. As we can see, "heat" is a keyword found the most in the descriptor field across the Boroughs Bronx, Brooklyn,Manhattan while "street" is found most in Staten Islands and Queens complaints. These insights give an idea about the context in which the complaints are reported and are very insightful.

CONCLUSION

Through this assignment we were able to analyze the NYC 311 service request dataset and explore more about and find new information. We asked a set of questions and through the visualizations made, we were able to answer those questions.

Initially, we cleaned the data and pre-processed it and made it ready for analysis. Using the latitude and longitude columns from the dataset we marked the areas of the top 5 complaints across NYC. It must be noted that, for this particular analysis we used a sample of the dataset which has 400,000 rows. Amongst the top complaints, ***Heat/Hot water*** complaint is prevalent throughout NYC. Following Heat/Hot Water complaint is ***Water System*** compliant which is equally distributed throughout the city, although less dense than the former complaint. An interesting observation we made is that in the northern region of NYC (Upper Manhattan), although being known as one of the busiest locations in the world, there are hardly any complaints of ***Blocked Driveways.***

We looked into the efficiency of top 5 agencies based on the closed cases. ***The Department of Housing Preservation and Development*** has closed the most cases. We also analyzed the pending cases of all the agencies and saw that the Agencies ***DOT, DSNY,*** and ***DOHMH*** have at least 50000 complaints, topping the list is the ***Department of Transportation (DOT)*** with registered pending complaints close to 150,000. While the remaining agencies have almost no pending cases but it could also mean the case could be pending but the Status could be ***Assigned*** or ***Email sent***, etc.

Amongst all the five boroughs, most complaints were registered to ***NYPD*** were from ***Brooklyn*** and ***Queens***. The lowest number of complaints originate from ***Staten Island.*** While studying the dataset, we understood that under the umbrella of noise complaints there are several varieties of noise complaints. To analyze in depth, we made a visualization which explains all the different noise complaints registered from different boroughs. According to the previous visualization, although Queens and Brooklyn have the most complaints registered to NYPD, Manhattan is where top 3 noise complaint types (Noise, Noise-Commercial, andNoise-Street/Sidewalk) are registered.

In NYC, the weather can get very cold during the first quarter of the year, this is the period where the most number of complaints are registered with January being the highest. This could be because of the high volume of complaints due to ***Heat/Hot Water.***

We made an important visualization which talks about how efficiently all the agencies resolve their respective service requests. We calculated the average resolution time taken (in hours) for each agency. ***The Department of Consumer Affairs*** take the most time to resolve their cases and is not very efficient.

We were also able to localize the time slot between which most of the complaints are reported in a day. The time around noon between 9-12 a.m is where most of the complaints are reported which makes sense as these are normal working hours of any agency office. One more peculiar observation is that even after normal office hours there is a steady incoming of complaints for more than 2 hours after which it declines rapidly.

We were also able to categorize the popular descriptor words used while complaining across the Boroughs. We see a divison of Complaint words across the 2 groups of Boroughs viz Manhattan,Bronx,Brooklyn and Staten Island and Queens, which gives us a better idea of where to replicate which resolution strategy.

We were also able to read in a new related dataset about the number of Inquiries that a call center handling an Agency's affairs gets. This dataset would be able to provide us more insight as to how each agency works when given a particular type of complaint. Also, since it contains the date and time value of a particular inquiry, we would be also able to find the time analysis of the type of inquiry that a customer makes. This dataset is between 2010 to 2015 which makes it perfect to be worked upon in conjunction with the current data-set. We were able to get rid of redundant columns and data. Alongside, we also created new columns based on the Date Time information that was present in the data-set.

For this assignment, we explored several datasets and found NYPD_Complaints_Data_Historic dataset is more relevant than others. With merging this data it would enable us to make interesting insights through visualizations. We can find the crime rate in each boroughs by year/ month.

Using dplyr package, we were able to perform a join between the nyc311 complaint and the nypd crime dataset. We were able to combine the data by projecting only the data between the year 2010 and 2015. Based on this , we could find that there were many columns like Boroughs, Year, Month, Time, etc. that we can use in joining the two datasets. We cleaned the new data in the same manner as we did for the nyc311 datasets in earlier assignments. We also created custom columns like the Year,month, day from the complaint time column of this dataset. As directed we created several short tables to signify the quality of the join on the datasets. So for a given Borough and time , we can point out how many complaints on the 311 helpline have been registered over the years and months,etc. and also report the crime statistics across the same demography.