homework iii

Parthivi Shrivastava, Khavya Seshadri

2019-09-24

Introduction

We have performed exploratory data analysis on 311 data and explored the relationship between the relevant columns of our pre-processed data. We have also depicted geographical maps with respect to complaint type, borough and agencies.

Initialization

Here we load the tidyverse packages and the `data.table` package and load the nyc311 data set. Then we fix the column names of the nyc311 data so that they have no spaces.

```
library(tidyverse)
```

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

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.1     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.5.2

## Warning: package 'tibble' was built under R version 3.5.2

## Warning: package 'tidyr' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.2

## Warning: package 'dplyr' was built under R version 3.5.2

## Warning: package 'stringr' was built under R version 3.5.2

## Warning: package 'forcats' was built under R version 3.5.2

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

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

```
## Warning: package 'data.table' was built under R version 3.5.2
```

```
##
## 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", ".")
```

Data pre-processing

Here we perform data pre-processing steps, by dropping irrelevant columns and removing duplicate rows from the dataset.

```
nyc311 <- nyc311[,c(-1,-10:-19,-23, -25:-49)]
nyc311nodups <- distinct(nyc311)
names(nyc311nodups)
```

```
##  [1] "Created.Date"                    "Closed.Date"
##  [3] "Agency"                          "Agency.Name"
##  [5] "Complaint.Type"                  "Descriptor"
##  [7] "Location.Type"                   "Incident.Zip"
##  [9] "Status"                          "Due.Date"
## [11] "Resolution.Action.Updated.Date" "Borough"
## [13] "Latitude"                        "Longitude"
## [15] "Location"
```

Exploration

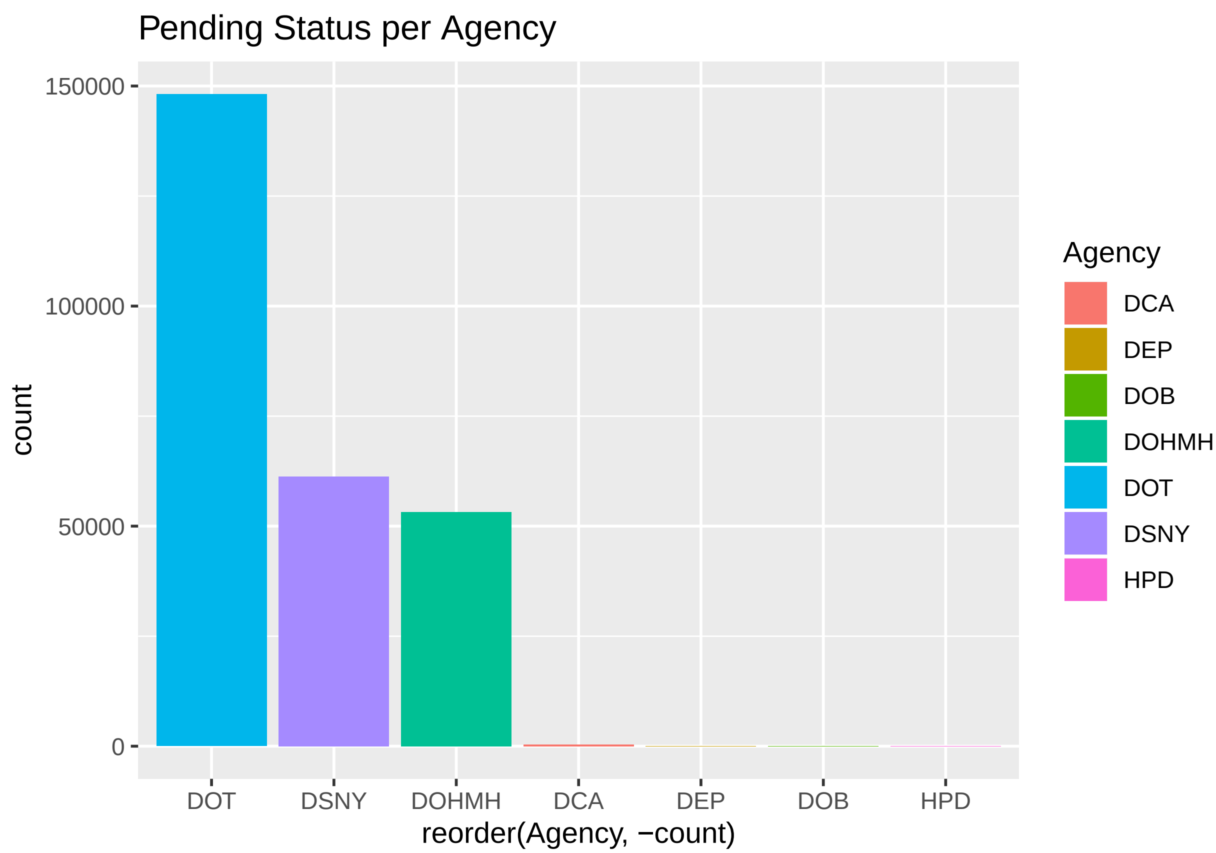
Here we explore the relationship between the columns in the data set, continuing from the previous exploration.

Plots

The following plot shows the pending complaints with respect to every agency.

```
pendingComp <- nyc311nodups %>%
      select(Agency,Status ) %>%
      filter(Status == "Pending")
agencyPending <- pendingComp %>%
      group_by(Agency) %>%
      summarize(count=n())
```

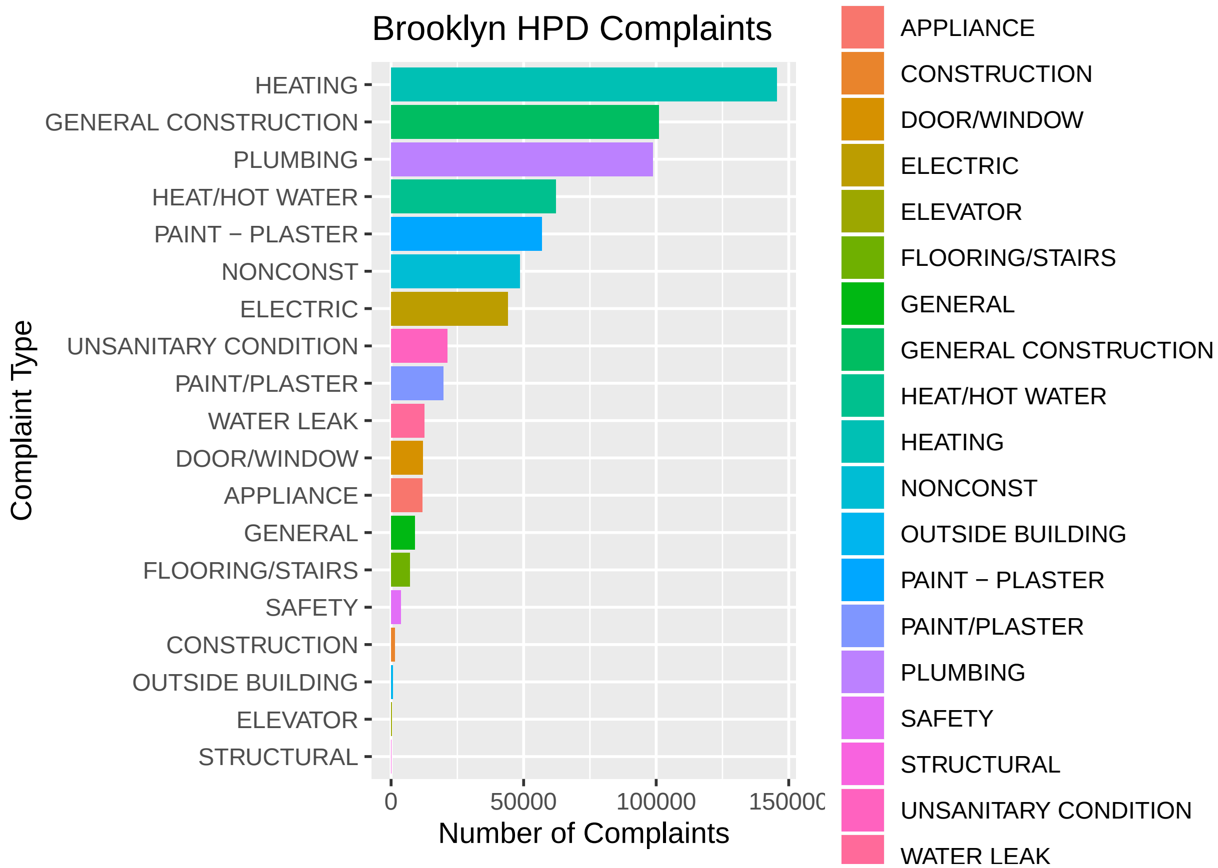
```
plotA <- ggplot(agencyPending,aes(x = reorder(Agency,-count), y = count,fill = Agency)) +
      geom_bar(stat = "identity") +
      ggtitle("Pending Status per Agency")
plotA
```



We found that the DOT agency had the most tickets with pending status and it looks like the DOB, HPD and DEP agencies are doing well, as they seem to have no pending tickets. This information can be used to advise the agencies to fasten the process of handling the pending service call requests.

In the following we are diving deep into showing the count of complaint types majoring in Brooklyn and handled by the HPD agency. We are exploring the complaints majoring in Brooklyn and handled by HPD.

```
brooklynComp <- nyc311nodups %>%
  select(Borough,Complaint.Type,Agency) %>%
  filter(Borough == "BROOKLYN"& Agency == "HPD")
brooklynHPD <- brooklynComp %>%
  group_by(Complaint.Type) %>%
  summarize(Complaints = length(Complaint.Type))
plotB <- ggplot(brooklynHPD, aes(x= reorder(Complaint.Type,Complaints), y=Complaints,
                                 fill = Complaint.Type )) +
         xlab("Complaint Type") + geom_bar(stat ="identity") +
         coord_flip() + ggtitle("Brooklyn HPD Complaints") +
         ylab("Number of Complaints")
plotB
```



From our previous exploration(hwii), we found that most complaints occured at Brooklyn and was handled by HPD agency. From the above plot, we see that the major complaint(Heating) seems to occur the most in Brooklyn as attended by HPD. This can be useful to know about the common complaints for people who wants to move in to Brooklyn.

Now we explore the average number of days taken by every agency to resolve the complaints(ignoring the empty dates).

```
resolveComplaints <- nyc311nodups %>%
  select(Complaint.Type,
     Created.Date,
     Closed.Date,
     Due.Date,
     Agency,
     Borough)
filteredData <-dplyr::filter(resolveComplaints,
              (str_trim(resolveComplaints$Closed.Date)!="" &
numOfDays <- abs(as.Date(filteredData$Closed.Date, format="%m/%d/%Y") -
                   as.Date(filteredData$Created.Date, format="%m/%d/%Y"))
filteredData <- data.frame(filteredData,numOfDays)
slowAgency <- filteredData %>%
  group_by(Agency) %>%
  summarize(averageTime = as.integer(mean(numOfDays)))
slowAgency <- slowAgency[order(-slowAgency$averageTime),]
slowAgency
```

```
## # A tibble: 28 x 2
```

```
##    Agency averageTime
##    <chr>        <int>
##  1 CHALL        41784
##  2 OPS          41277
##  3 DCAS         41259
##  4 WF1          40974
##  5 OATH         40972
##  6 CWI          40958
##  7 DOHMH         1803
##  8 DCA            147
##  9 DPR            114
## 10 TLC             77
## # ... with 18 more rows
```

```
topAgencies <- dplyr::filter(slowAgency, Agency=='HPD'|Agency=='DOT'|Agency=='NYPD')
topAgencies
```

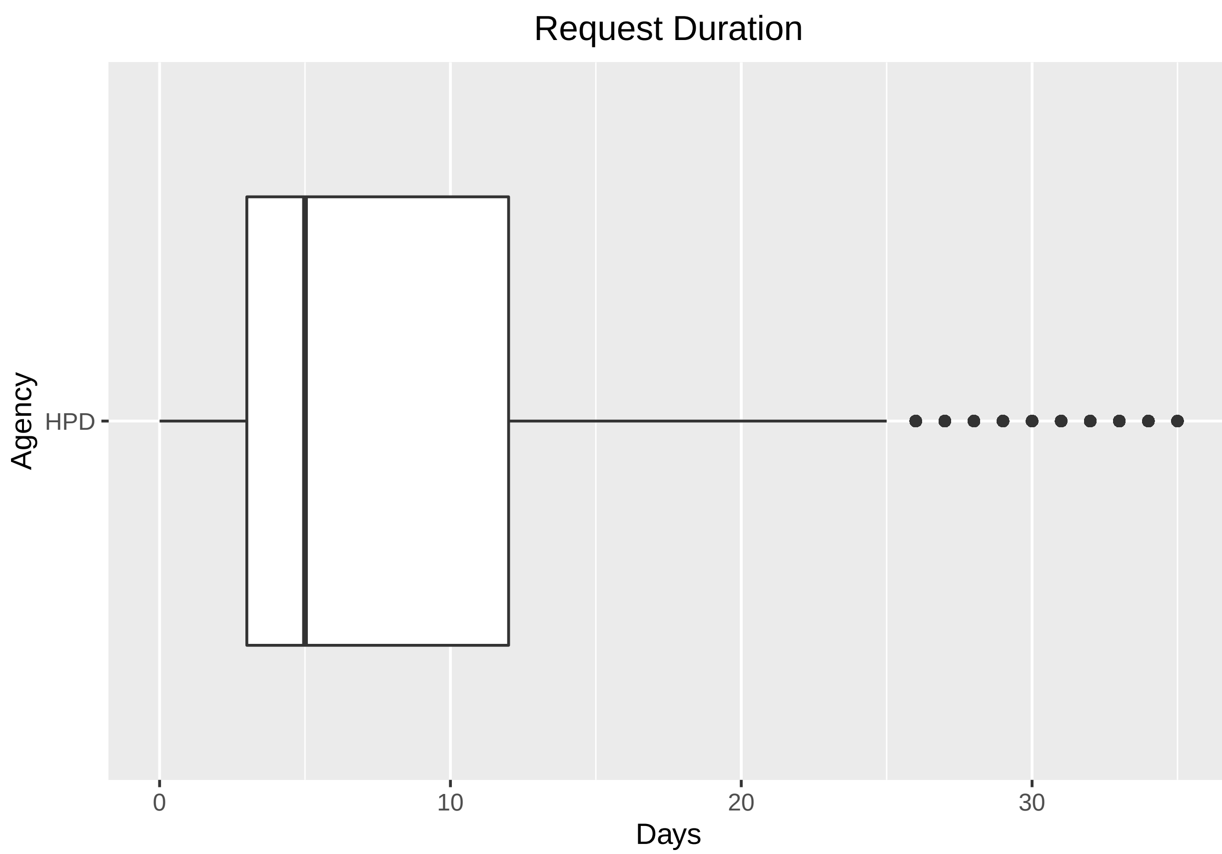
```
## # A tibble: 3 x 2
##   Agency averageTime
##   <chr>        <int>
## 1 HPD             10
## 2 DOT              8
## 3 NYPD             0
```

The number of days taken to resolve a complaint are computed using the created date and closed date. From the table we get to know the average time taken by the top agencies(as explored in hwii) in resolving the complaints.

The following can be useful to know about the duration for resolving HPD complaints.

```
hpdComplaints <- dplyr::filter(filteredData, (Agency=="HPD"))
duration <- abs(as.Date(hpdComplaints$Closed.Date, format="%m/%d/%Y") -
                  as.Date(hpdComplaints$Created.Date, format="%m/%d/%Y"))
plotC <- ggplot(hpdComplaints, aes(x=Agency, y=duration)) +
         geom_boxplot() + ylim(0,35) + ylab("Days") +
         ggtitle("Request Duration") +
         theme(plot.title = element_text(hjust = 0.5)) +coord_flip()
plotC
```

```
## Warning: Removed 86586 rows containing non-finite values (stat_boxplot).
```



The above shows a box plot depicting the request duration of the HPD complaints, which takes on an average of 10 days to resolve a complaint. This plot gives an idea of the varaition in the data with respect to the number of days taken by HPD to resolve the complaints.

The following can be used to find the duration taken for resolving the top three complaints.

```
complaintsData <- filteredData %>%
  group_by(Complaint.Type) %>%
  summarize(averageTime = as.integer(mean(numOfDays)))
complaintsData <- complaintsData[order(-complaintsData$averageTime),]

complaintsData[complaintsData$Complaint.Type=='HEATING'|
                 complaintsData$Complaint.Type=='Street Condition' |
                 complaintsData$Complaint.Type=='Street Light Condition',]
```

```
## # A tibble: 3 x 2
##   Complaint.Type         averageTime
##   <chr>                        <int>
## 1 Street Light Condition           7
## 2 Street Condition                 6
## 3 HEATING                          3
```

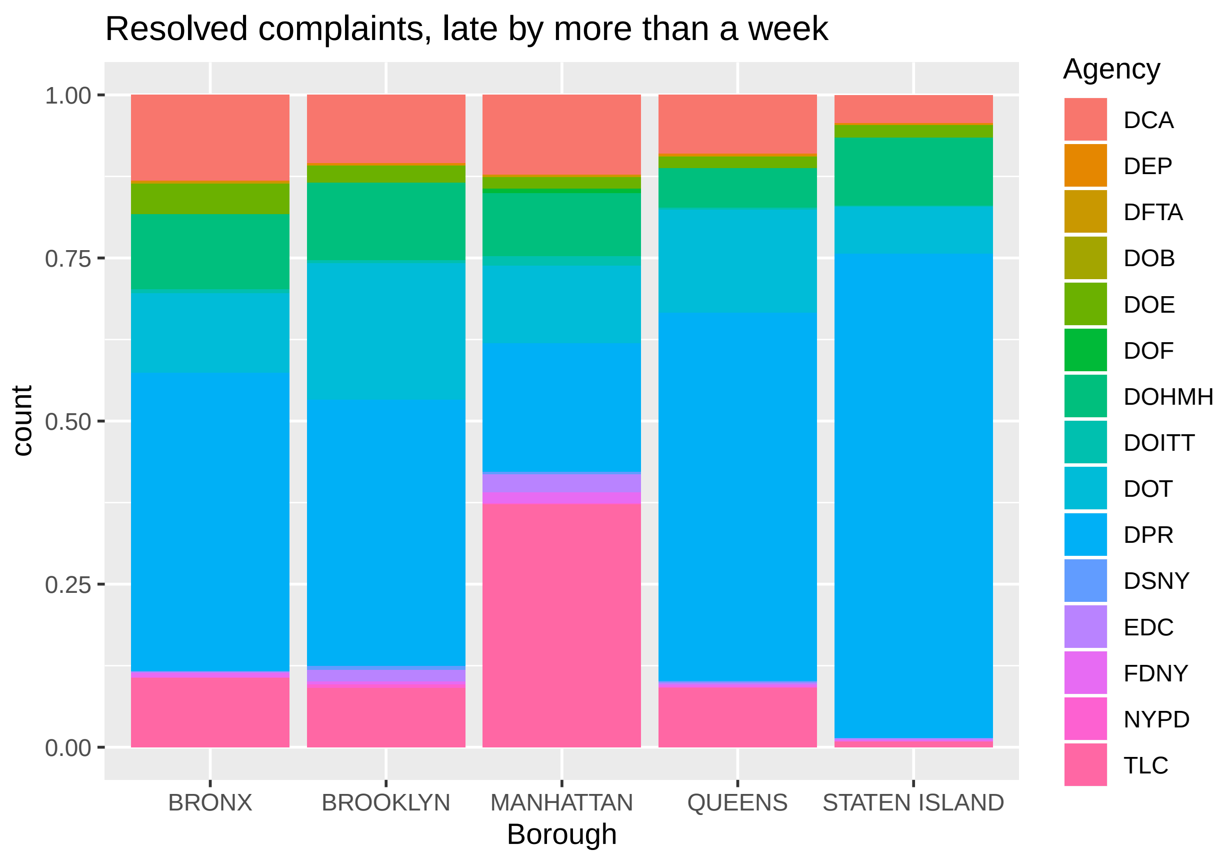
The above table indicates the average time taken to resolve the top 3 complaint types, which was found from our previous exploration. We have shown only the major three complaints because this gives us an idea about how fast these complaints have been resolved.

Now, we are interested in knowing about the complaints that are late by more than a week.

```r
lateComplaints <- dplyr::filter(resolveComplaints,
            as.Date(Due.Date, format="%m/%d/%Y")+6 <
              as.Date(Closed.Date, format="%m/%d/%Y"))
lateComp <- lateComplaints %>%
            filter(Borough!="Unspecified") %>%
            group_by(Borough,Agency) %>%
            summarize(count=n())
lateComp
```

```
## # A tibble: 69 x 3
## # Groups:   Borough [5]
##    Borough Agency count
##    <chr>   <chr>  <int>
##  1 BRONX   DCA     2390
##  2 BRONX   DEP       33
##  3 BRONX   DFTA      46
##  4 BRONX   DOE      856
##  5 BRONX   DOHMH   2094
##  6 BRONX   DOITT    117
##  7 BRONX   DOT     2208
##  8 BRONX   DPR     8350
##  9 BRONX   EDC       40
## 10 BRONX   FDNY     127
## # ... with 59 more rows
```

```r
plotD <- ggplot(lateComp,aes(x=Borough,y=count, fill=Agency)) +
         geom_bar(stat="identity", position = "fill") +
         ggtitle("Resolved complaints, late by more than a week")
plotD
```



The late complaints were computed using the due date and the closed date. The above plot shows the late complaints with respect to the agency and the borough. This information would be useful to know about which agencies lack behind in completion of the requests within the due date.

Geo Plots

Here, we are generating a random sample of size 10K from the pre-processed data.

```
mini311<-nyc311nodups[sample(nrow(nyc311nodups),10000),]
write.csv(mini311,"mini311.csv")
sample<-fread("mini311.csv")
```

Selecting the required columns to explore and we narrow down the data to include just Noise complaints.

```
complaintlocs <- sample %>%
  select(Complaint.Type,
    Longitude,
    Latitude
  )
noisecompl <- complaintlocs %>%
  filter(Complaint.Type == "Noise")
```

Including libraries required for map

```r
library(devtools)
```

```
## Warning: package 'devtools' was built under R version 3.5.2
```

```
## Loading required package: usethis
```

```
## Warning: package 'usethis' was built under R version 3.5.2
```

```r
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.5.2
```

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

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

Registering with Google API key

```r
key <- "AIzaSyClTqcMNpFm9_rFaaXH6ptzDpmTmAEwml4"
register_google(key=key)
```

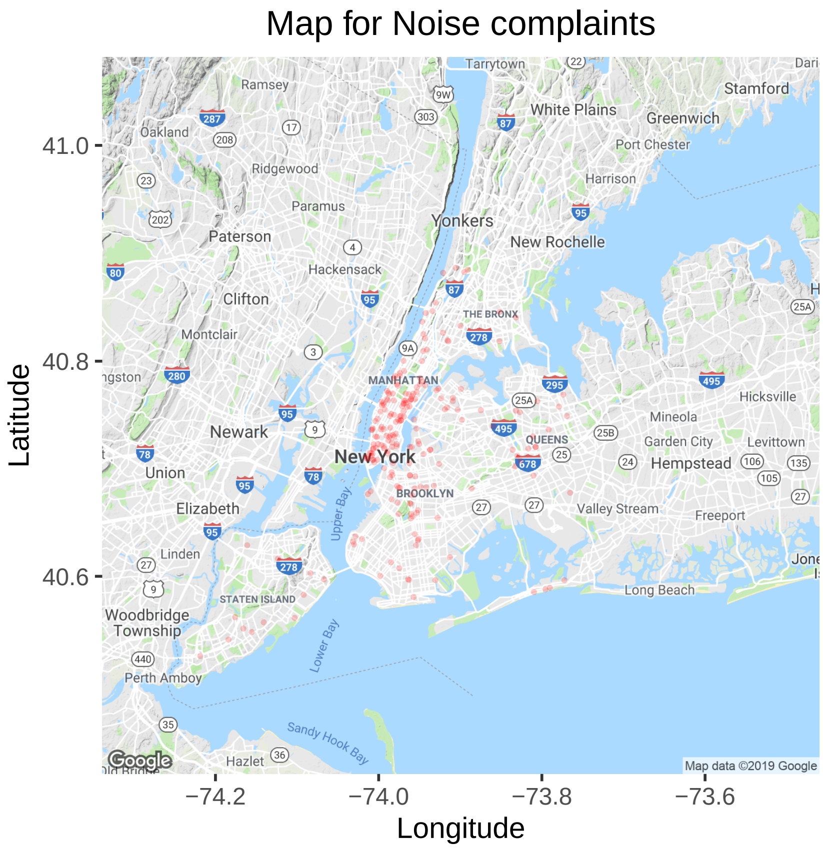
Generating map using the sample for Noise complaints.

```r
nyc_map <- get_map(location=c(lon=-73.9,lat=40.75),
           maptype="terrain",zoom=10)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=40.75,-73.9&zoom=10&size=640x640&scal
```

```r
map <- ggmap(nyc_map) +
  geom_point(data=noisecompl,aes(x=Longitude,y=Latitude),
         size=0.4,alpha=0.2,color="red") +
  ggtitle("Map for Noise complaints") +
  theme(plot.title=element_text(hjust=0.5)) +
  xlab("Longitude") + ylab("Latitude")
map
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```



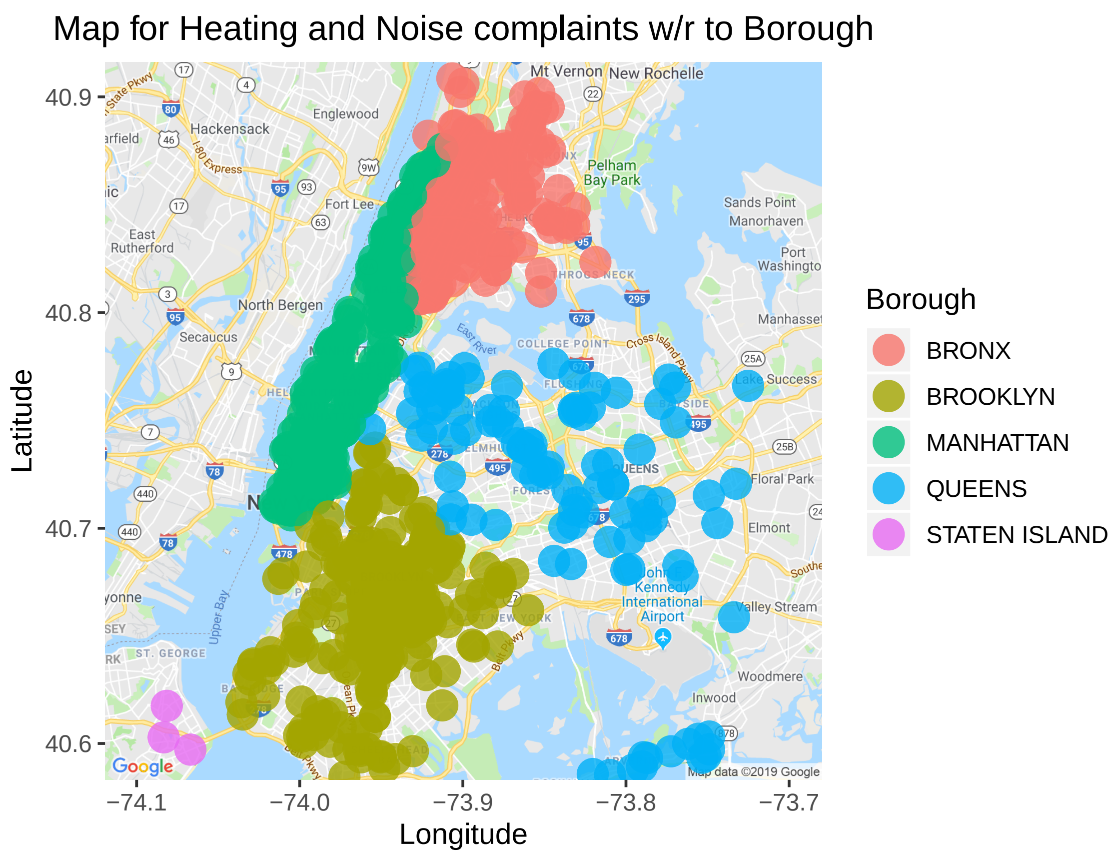
Considering Heating and Noise complaint types, we have generated a map with respect to the boroughs(ignoring Unspecified boroughs) differentiated using colors.

```r
geoBoroughMap <- sample %>%
  select(Complaint.Type,
    Longitude,
    Latitude,Borough
  )%>%
filter(Borough !="Unspecified")
geoBoroughMap <- geoBoroughMap %>%
  filter(Complaint.Type == "Noise" | Complaint.Type == "HEATING")
nyc_map <- get_map(location=c(lon=-73.9,lat=40.75),
           maptype="roadmap",zoom=11)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=40.75,-73.9&zoom=11&size=640x640&scale
```

```r
map1 <- ggmap(nyc_map) +
  geom_point(data=geoBoroughMap, aes(x=Longitude, y=Latitude, color=Borough)
        ,alpha=0.8, size=5)+
  ggtitle("Map for Heating and Noise complaints w/r to Borough") +
  theme(plot.title=element_text(hjust=0.5)) +
  xlab("Longitude") + ylab("Latitude")
map1
```

```
## Warning: Removed 25 rows containing missing values (geom_point).
```



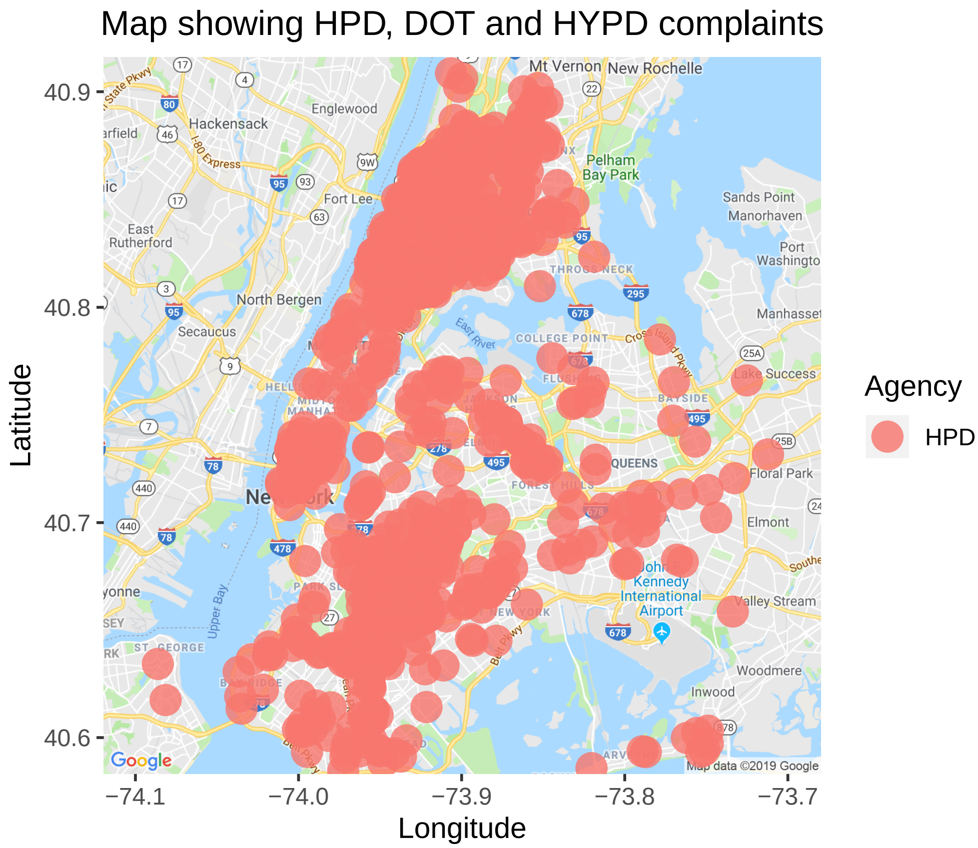
We have generated a map showing the complaints handled by HPD, DOT and NYPD agencies specific to Heating complaint.

```
geoAgencyMap <- sample %>%
  select(Complaint.Type,
    Longitude,
    Latitude,Agency
  )%>%
  filter(Agency =="HPD" | Agency =="DOT" | Agency == "NYPD")
geoAgencyMap <- geoAgencyMap %>%
  filter(Complaint.Type == "HEATING")
nyc_map <- get_map(location=c(lon=-73.9,lat=40.75),
           maptype="roadmap",zoom=11)
```

```
## Source : https://maps.googleapis.com/maps/api/staticmap?center=40.75,-73.9&zoom=11&size=640x640&scale
```

```
map2 <- ggmap(nyc_map) +
  geom_point(data=geoAgencyMap, aes(x=Longitude, y=Latitude, color=Agency)
        ,alpha=0.8, size=5)+
  ggtitle("Map showing HPD, DOT and HYPD complaints") +
  theme(plot.title=element_text(hjust=0.5)) +
  xlab("Longitude") + ylab("Latitude")
map2
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



Conclusion

In this document, we found relationship between the following columns: Complaint type, Borough and Agency. Initially we found the pending tickets per agency and explored by focussing on HPD complaints in Brooklyn. Then, we computed the average time taken by the agencies to resolve the complaints and found out the information regarding the complaints that were late by more than a week from the due date. Finally, we showed geographical maps specific to few complaint types and generated plots from a random sample with respect to agency and borough.