NYC Housing Complaints

Amit V Singh

6/16/2019

Problem Statement

The people of New Yorker use the 311 system to report complaints about the non-emergency problems to local authorities. In the last few years, the number of 311 complaints coming to the Department of Housing Preservation and Development has increased significantly. Although these complaints are not necessarily urgent, the large volume of complaints and the sudden increase is impacting the overall efficiency of operations of the agency.

Therefore, I have developed a solution to help the Department of Housing Preservation and Development to manage their large volume of 311 complaints they are receiving every year.

The project tries to answers several questions:

1. Which type of complaint should the Department of Housing Preservation and Development of New York City focus on first?
2. Should the Department of Housing Preservation and Development of New York City focus on any particular set of boroughs, ZIP codes, or street (where the complaints are severe) for the specific type of complaints you identified in response to Question 1?
3. Does the Complaint Type that you identified in response to question 1 have an obvious relationship with any particular characteristic or characteristics of the houses or buildings?
4. Can a predictive model be built for a future prediction of the possibility of complaints of the type that you have identified in response to question 1?

The project contains Rmd, R and pdf files each problem with 4 subsections, one for each problem statement. It contains data analysis along with nice visualisations.

Datasets

Two datasets have been used from the Department of Housing Preservation and Development of New York City to address their problems.

1. 311 complaint dataset (https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9)

2. PLUTO dataset for housing (https://data.cityofnewyork.us/City-Government/Primary-Land-Use-Tax-Lot-Output-PLUT xuk2-nczf)

(The details of the project overview can be found in the following link https://courses.edx.org/courses/course-v1:IBM+DS0720EN+1T2019/course/)

Library

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(leaflet)) install.packages("leaflet")
if(!require(randomForest)) install.packages("randomForest")
```

```r
library(tidyverse)
library(ggplot2)
library(caret)
library(lubridate)
library(leaflet)
library(randomForest)
```

Problem 1: Which type of complaint should the Department of Housing Preservation and Development of New York City focus on first?

```r
dl <- tempfile()

# download file. it may take few minutes (fileSize = 2.57GB, nrows ~ 5800000).
url   <- "https://data.cityofnewyork.us/resource/fhrw-4uyv.csv?$limit=100000000&Agency=HPD&$select=creat
download.file(url, dl)

df_NYC = read.csv(dl)

# get the idea of dataframe's number of rows and columns
dim(df_NYC)
```

```
## [1] 5792711      15
```

```r
rm(dl)
```

Basic Exploratory Analysis and Summary Statistics

```r
# 7 rows of the dataset with header
head(df_NYC)
```

```
##              created_date unique_key        complaint_type incident_zip
## 1 2013-01-11T13:25:34.000   24765056 HPD Literature Request           NA
## 2 2018-08-11T19:19:41.000   39981834           PAINT/PLASTER        11429
## 3 2018-08-11T19:19:41.000   39982698               APPLIANCE        11429
## 4 2018-08-11T19:19:41.000   39987943   UNSANITARY CONDITION        11429
## 5 2018-10-23T19:27:06.000   40636028             DOOR/WINDOW        11412
## 6 2018-10-23T19:27:06.000   40637746   UNSANITARY CONDITION        11412
##    incident_address street_name address_type           city
## 1
## 2 104-34 219 STREET  219 STREET      ADDRESS Queens Village
## 3 104-34 219 STREET  219 STREET      ADDRESS Queens Village
## 4 104-34 219 STREET  219 STREET      ADDRESS Queens Village
## 5 116-35 195 STREET  195 STREET      ADDRESS   Saint Albans
## 6 116-35 195 STREET  195 STREET      ADDRESS   Saint Albans
##
## 1                                                                                    The lite
## 2                                      The Department of Housing Preservation and Development inspect
## 3                                      The Department of Housing Preservation and Development inspect
## 4 The Department of Housing Preservation and Development inspected the following conditions. Violatio
## 5                                      The Department of Housing Preservation and Development inspect
## 6 The Department of Housing Preservation and Development inspected the following conditions. Violatio
```

```
##       borough latitude longitude             closed_date
## 1 Unspecified       NA        NA 2013-01-11T15:01:56.000
## 2      QUEENS 40.71154 -73.73572 2019-03-30T08:58:02.000
## 3      QUEENS 40.71154 -73.73572 2019-03-30T08:58:01.000
## 4      QUEENS 40.71154 -73.73572 2019-03-30T08:58:01.000
## 5      QUEENS 40.69372 -73.75712 2019-03-30T08:58:02.000
## 6      QUEENS 40.69372 -73.75712 2019-03-30T08:58:02.000
##          location_type status
## 1                      Closed
## 2 RESIDENTIAL BUILDING Closed
## 3 RESIDENTIAL BUILDING Closed
## 4 RESIDENTIAL BUILDING Closed
## 5 RESIDENTIAL BUILDING Closed
## 6 RESIDENTIAL BUILDING Closed
```

```r
# columns names
colnames(df_NYC)
```

```
##  [1] "created_date"           "unique_key"
##  [3] "complaint_type"         "incident_zip"
##  [5] "incident_address"       "street_name"
##  [7] "address_type"           "city"
##  [9] "resolution_description" "borough"
## [11] "latitude"               "longitude"
## [13] "closed_date"            "location_type"
## [15] "status"
```

```r
# datatype of columns
sapply(df_NYC, class)
```

```
##           created_date             unique_key         complaint_type
##               "factor"              "integer"               "factor"
##           incident_zip       incident_address            street_name
##              "integer"               "factor"               "factor"
##           address_type                   city resolution_description
##               "factor"               "factor"               "factor"
##                borough               latitude              longitude
##               "factor"              "numeric"              "numeric"
##            closed_date          location_type                 status
##               "factor"               "factor"               "factor"
```

```r
# basic summary statistics
summary(df_NYC)
```

```
##                   created_date       unique_key
##  2013-01-24T00:00:00.000:   7581   Min.   :15629728
##  2015-01-08T00:00:00.000:   7183   1st Qu.:22711060
##  2014-01-07T00:00:00.000:   6984   Median :28832675
##  2015-02-16T00:00:00.000:   6382   Mean   :28978207
##  2014-01-08T00:00:00.000:   6153   3rd Qu.:35168761
##  2012-01-04T00:00:00.000:   5887   Max.   :42992832
##  (Other)                :5752541
##              complaint_type     incident_zip
##  HEAT/HOT WATER      :1144631   Min.   :10001
##  HEATING             : 887869   1st Qu.:10452
##  PLUMBING            : 696090   Median :10469
```

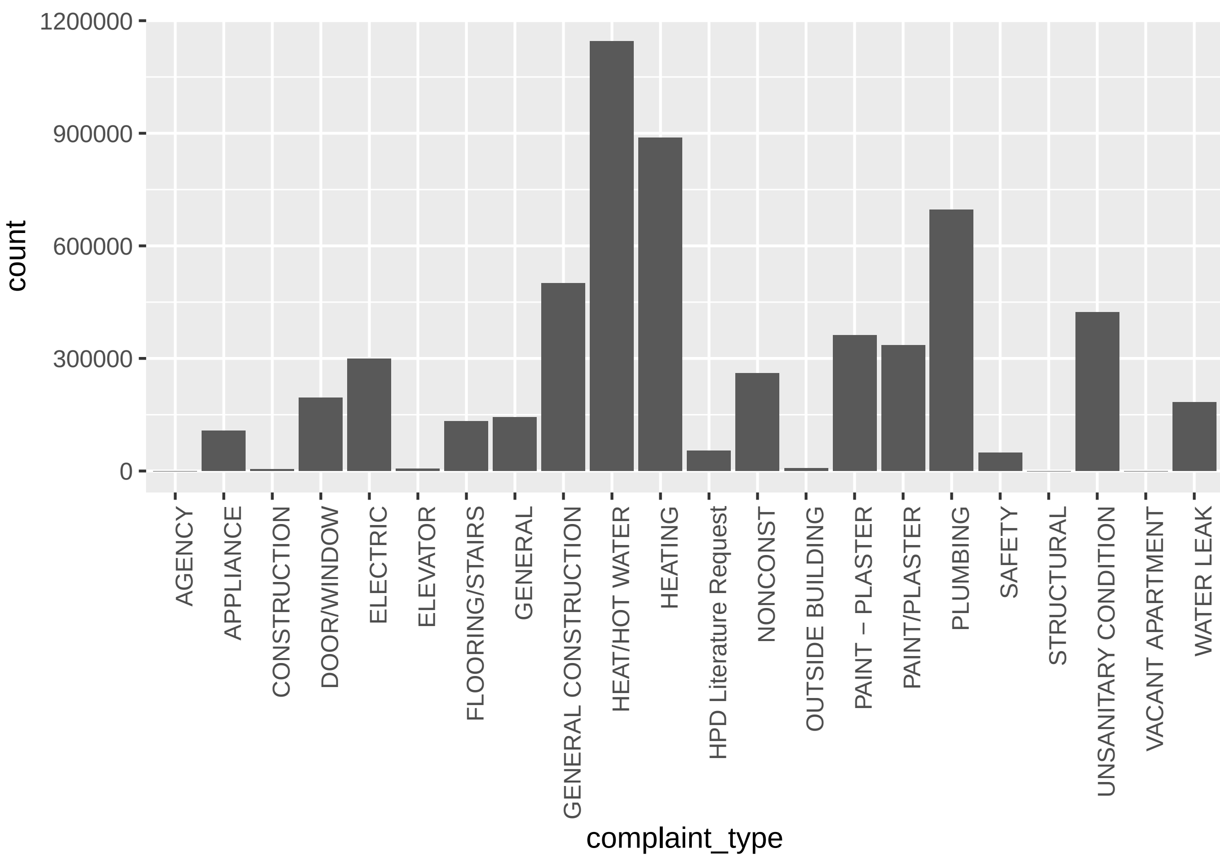
```
## GENERAL CONSTRUCTION: 500863   Mean   :10748
## UNSANITARY CONDITION: 423028   3rd Qu.:11224
## PAINT - PLASTER     : 361258   Max.   :12345
## (Other)             :1778972   NA's   :81898
##              incident_address              street_name
##                      :  54145   GRAND CONCOURSE   :  89149
## 34 ARDEN STREET      :  14248   BROADWAY          :  63396
## 89-21 ELMHURST AVENUE:  11406                     :  54145
## 1025 BOYNTON AVENUE  :   9835   OCEAN AVENUE      :  53307
## 3810 BAILEY AVENUE   :   7171   ST NICHOLAS AVENUE:  40049
## 2913 FOSTER AVENUE   :   4911   MORRIS AVENUE     :  39443
## (Other)              :5690995   (Other)           :5453222
##  address_type              city
##       :  78996   BROOKLYN     :1955914
## ADDRESS:5713715   BRONX        :1786188
##                  NEW YORK     :1154902
##                  STATEN ISLAND:  97982
##                               :  81497
##                  Jamaica      :  62359
##                  (Other)      : 653869
##
## The Department of Housing Preservation and Development inspected the following conditions. No viola
## The Department of Housing Preservation and Development inspected the following conditions. Violation
## The Department of Housing Preservation and Development was not able to gain access to inspect the fo
## The complaint you filed is a duplicate of a condition already reported by another tenant for a build
## The Department of Housing Preservation and Development responded to a complaint of no heat or hot wa
## The Department of Housing Preservation and Development was not able to gain access to your apartment
## (Other)
##          borough          latitude       longitude
## BRONX        :1543582   Min.   :40.50   Min.   :-74.25
## BROOKLYN     :1669223   1st Qu.:40.67   1st Qu.:-73.95
## MANHATTAN    :1005710   Median :40.76   Median :-73.92
## QUEENS       : 615683   Mean   :40.75   Mean   :-73.92
## STATEN ISLAND:  84013   3rd Qu.:40.84   3rd Qu.:-73.89
## Unspecified  : 874500   Max.   :40.91   Max.   :-73.70
##                         NA's   :81872   NA's   :81872
##                  closed_date                    location_type
##                         : 117906                       :  54144
## 2012-11-07T00:00:00.000:   7296   RESIDENTIAL BUILDING:5738567
## 2010-12-09T00:00:00.000:   6264
## 2011-11-28T00:00:00.000:   6005
## 2014-01-06T00:00:00.000:   5600
## 2013-01-28T00:00:00.000:   5598
## (Other)                 :5644042
##      status
## Assigned:      6
## Closed  :5667904
## Open    : 124799
## Pending :      2
##
##
##
```

Number of Housing Complaints

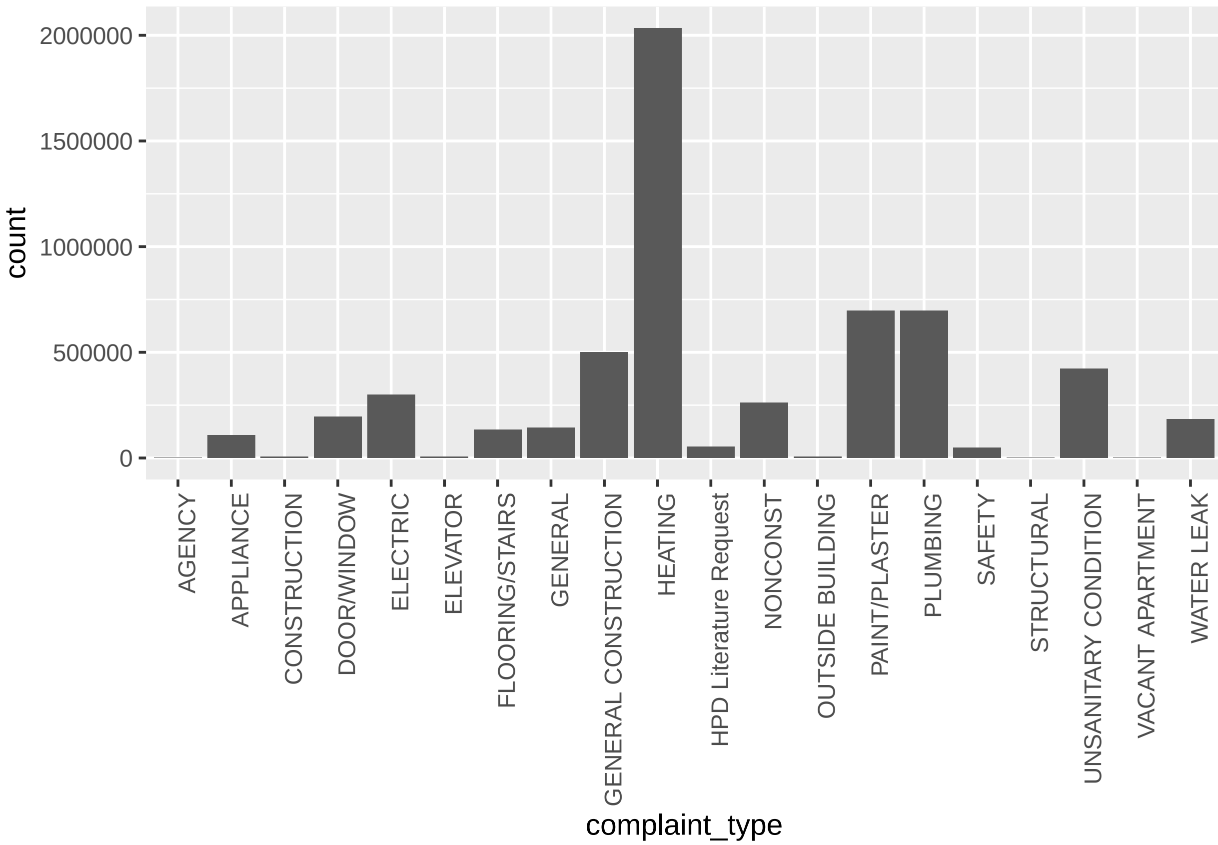
```
df_NYC%>%
group_by(complaint_type) %>%
summarize(count = n()) %>%
arrange(desc(count))
```

```
## # A tibble: 22 x 2
##    complaint_type          count
##    <fct>                   <int>
##  1 HEAT/HOT WATER        1144631
##  2 HEATING                887869
##  3 PLUMBING               696090
##  4 GENERAL CONSTRUCTION   500863
##  5 UNSANITARY CONDITION   423028
##  6 PAINT - PLASTER        361258
##  7 PAINT/PLASTER          335622
##  8 ELECTRIC               299646
##  9 NONCONST               260890
## 10 DOOR/WINDOW            195696
## # ... with 12 more rows
```

```
## we can visulize the complaints type and number of complainted in the bar plot

df_NYC %>% ggplot(aes(complaint_type))+
geom_bar()+
theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



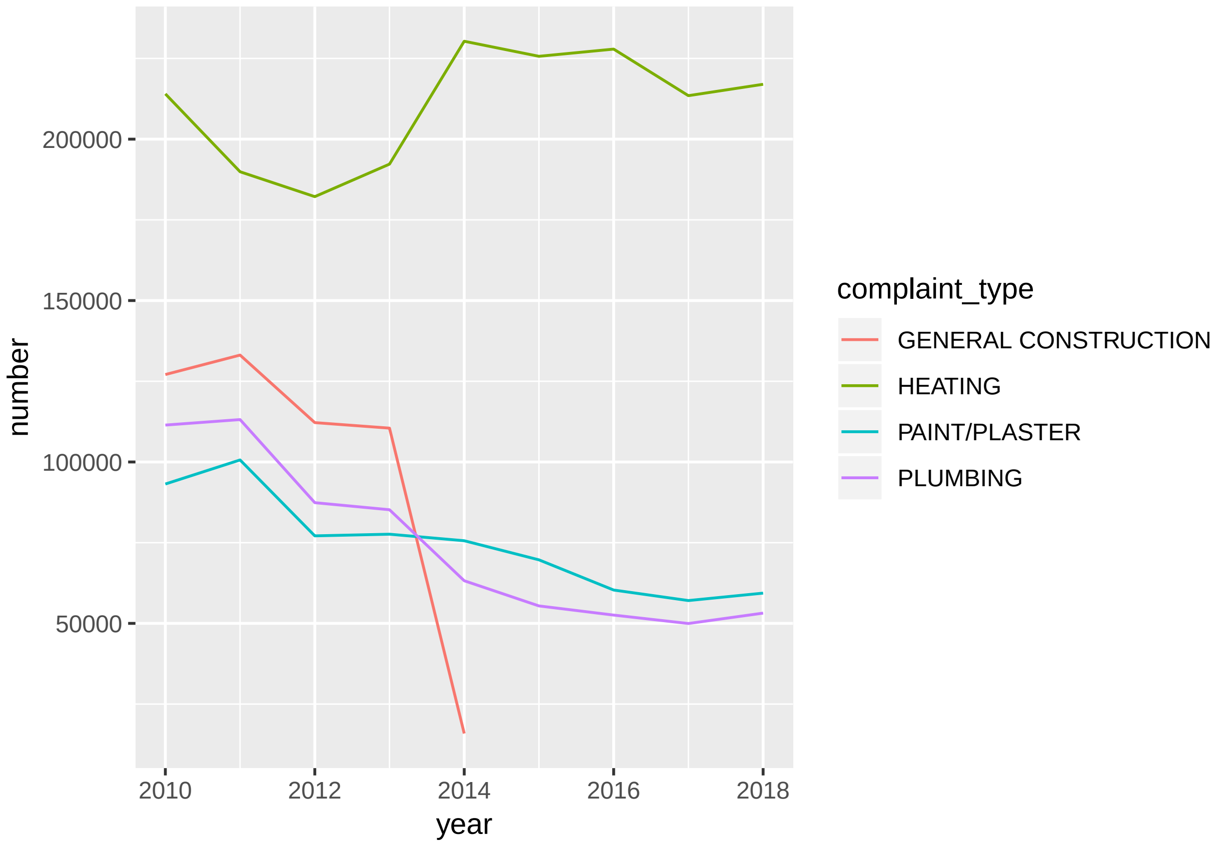
```
## After reading the New york city data file, one can see that HEAT/HOT WATER complaint column has been

df_NYC$complaint_type[df_NYC$complaint_type %in% "HEAT/HOT WATER"] <- "HEATING"
df_NYC$complaint_type[df_NYC$complaint_type %in% "PAINT - PLASTER"] <- "PAINT/PLASTER"

df_NYC %>% ggplot(aes(complaint_type))+
geom_bar()+
theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Temoral Evolution of Complaints Type

```
# Convert `timestamp to`POSIXct`
dt <- as.POSIXct(df_NYC$created_date)
df_NYC <- df_NYC %>% mutate(year = format(dt, "%Y"), month = format(dt, "%m"))

rm(dt)

complaint_year <- df_NYC %>%
na.omit() %>% # omit missing values
#select(year, complaint_type) %>% # select columns we are interested in
mutate(year = as.factor(year)) %>% # turn year in factors
mutate(year = as.numeric(levels(year))[year]) %>%
filter(year < 2019) %>%
group_by(year, complaint_type) %>% # group data by year and complaint_Type
summarise(number = n())  # count
```

```r
complaint_year %>%
filter(complaint_type %in% c("HEATING", "PLUMBING", "GENERAL CONSTRUCTION", "PAINT/PLASTER")) %>%
ggplot(aes(x = year, y = number)) +
geom_line(aes(color=complaint_type)) +
scale_fill_brewer(palette = "Paired")
```



Concluding Remarks: solution of problem 1: Based on the above plot it is clear that maximum number of complaints are coming from HEAT/HOT water category. So HPD should address the HEAT/HOT WATER complaint first. The problem remails all time high. It is clear with the time dependent plots.

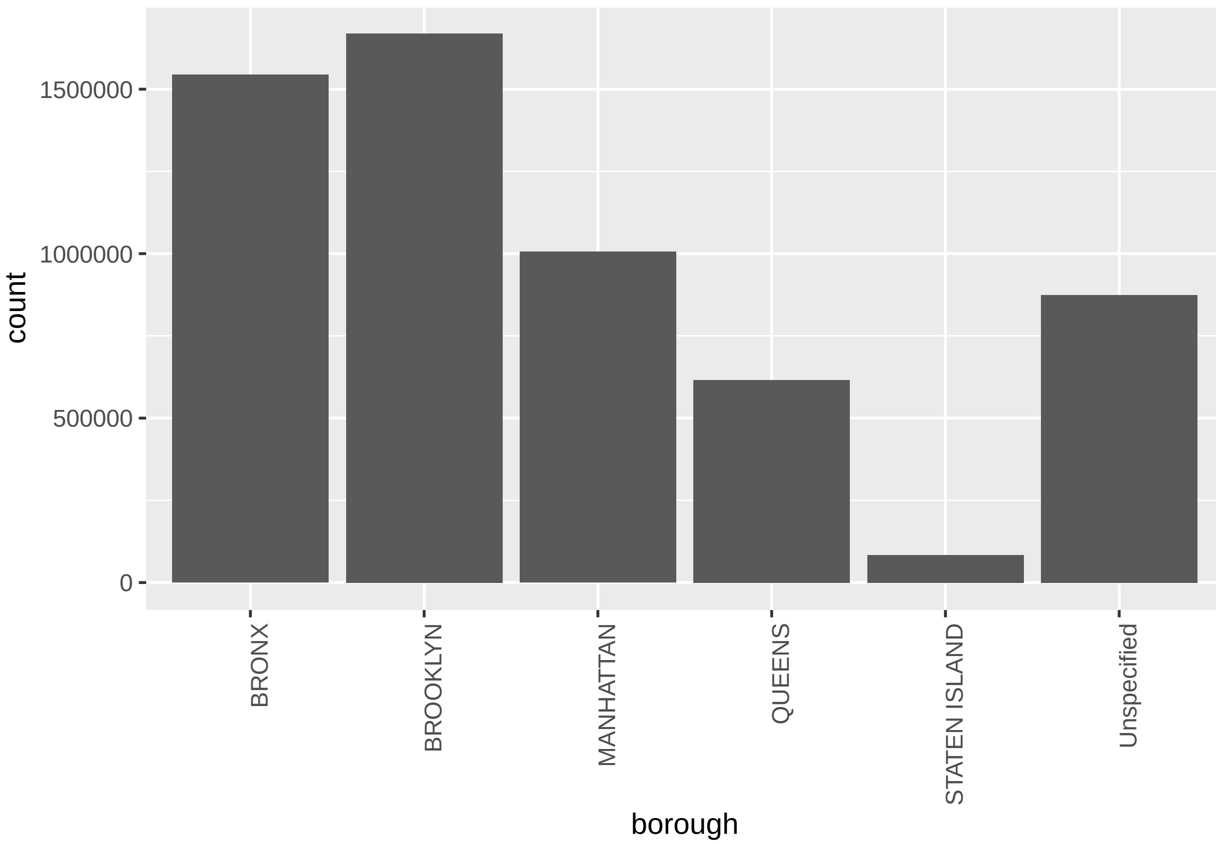
Problem 2: Should the Department of Housing Preservation and Development of New York City focus on any particular set of boroughs, ZIP codes, or street (where the complaints are severe) for the specific type of complaints you identified in response to Question 1?

```r
# number of complaints for each borough
df_NYC %>%
na.omit() %>% # omit missing values
#select(year, complaint_type) %>% # select columns we are interested in
group_by(borough) %>% # group data by year and complaint_Type
summarise(number = n())  # count
```

```
## # A tibble: 6 x 2
##   borough       number
```

```
##   <fct>           <int>
## 1 BRONX          1535504
## 2 BROOKLYN       1660537
## 3 MANHATTAN       999864
## 4 QUEENS          611494
## 5 STATEN ISLAND    83616
## 6 Unspecified     819798
```

```r
# bar plot for complaints in each borough
df_NYC %>% ggplot(aes(borough))+
geom_bar() +
theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



This chunk of code produces an interactive map of NYC housing complain area

```r
#lat <- df_NYC$latitude %>% na.omit()
#lng <- df_NYC$longitude %>% na.omit()

#df_geo <- data.frame(lat = runif(1000, min = min(lat), max = max(lat)),
#                     lng = runif(1000, min = min(lng), max = max(lng)))

# The interactive map shows cluster of complaint prone area.

#df_geo %>% leaflet() %>%
#  addTiles() %>%
#  addMarkers(clusterOptions = markerClusterOptions())
```

```
# Please note that this map can't be rendered in pdf format.
```

Concluding Remarks: solution of problem 2: The bar plot depicts that some boroughs are more severly affected by housing complaints than others. So for further analysis we should particularly focus on 4 boroughs namely 'BRONOX', 'BROOKLYN', 'MANHATTAN', 'QUEENS'. As a machine learning algorithm are more reliable if we have more feature sets. So in next section we will download some dataset for more details on Housing feathers in the mentioned boroughs.

Problem 3: Does the Complaint Type that you identified in response to question 1 have an obvious relationship with any particular characteristic or characteristics of the houses or buildings?

```
# THE ZIP FILE CAN BE DOWNLOADED FROM THE FOLLOWING LINK: "https://www1.nyc.gov/assets/planning/downloa

## download file. it may take couple of minutes (fileSize = 46MB).
#dl <- tempfile()
#zip.file.location <- "https://www1.nyc.gov/assets/planning/download/zip/data-maps/open-data/nyc_pluto_
#download.file(zip.file.location, dl)
#BK_18v1 <- read.csv(unzip(dl,PLUTO_for_WEB/BK_18v1.csv))

#rm(dl)

# I am using my local directory to access the PLUTO files
BK_18v1 <- read.csv('./PLUTO_for_WEB/BK_18v1.csv')
BX_18v1 <- read.csv('./PLUTO_for_WEB/BX_18v1.csv')
MN_18v1 <- read.csv('./PLUTO_for_WEB/MN_18v1.csv')
QN_18v1 <- read.csv('./PLUTO_for_WEB/QN_18v1.csv')

# dimension of data frame
dim(BK_18v1)
```

```
## [1] 277316     87
```

```
# This gives 87 columns (features). Only few of them are relevant. lets store features that are relevan


# The recommended fields are Address, BldgArea, BldgDepth, BuiltFAR, CommFAR, FacilFAR, Lot, LotArea, L


df_BK <- BK_18v1 %>% select('Address', 'BldgArea', 'BldgDepth', 'BuiltFAR', 'CommFAR', 'FacilFAR', 'Lot

df_BX <- BX_18v1 %>% select('Address', 'BldgArea', 'BldgDepth', 'BuiltFAR', 'CommFAR', 'FacilFAR', 'Lot

df_MN <- MN_18v1 %>% select('Address', 'BldgArea', 'BldgDepth', 'BuiltFAR', 'CommFAR', 'FacilFAR', 'Lot

df_QN <- QN_18v1 %>% select('Address', 'BldgArea', 'BldgDepth', 'BuiltFAR', 'CommFAR', 'FacilFAR', 'Lot

# new data frame with smaller features
dim(df_BK)
```

```
## [1] 277316     20
```

```r
# Merge all data frames by rows
df_pluto = rbind(df_BK, df_BX, df_MN, df_QN)

identical(nrow(df_pluto), nrow(df_BK)+nrow(df_BX)+nrow(df_MN)+nrow(df_QN))
```

```
## [1] TRUE
```

```r
rm('df_BK', 'df_BX', 'df_MN', 'df_QN', 'BK_18v1', 'BX_18v1', 'MN_18v1', 'QN_18v1')
# The above dataframes has been successfully merged
```

Exploratory Analysis

```r
# print the column names
print(colnames(df_NYC))
```

```
##  [1] "created_date"           "unique_key"
##  [3] "complaint_type"         "incident_zip"
##  [5] "incident_address"       "street_name"
##  [7] "address_type"           "city"
##  [9] "resolution_description" "borough"
## [11] "latitude"               "longitude"
## [13] "closed_date"            "location_type"
## [15] "status"                 "year"
## [17] "month"
```

```r
print(colnames(df_pluto))
```

```
##  [1] "Address"    "BldgArea"   "BldgDepth"  "BuiltFAR"   "CommFAR"
##  [6] "FacilFAR"   "Lot"        "LotArea"    "LotDepth"   "NumBldgs"
## [11] "NumFloors"  "OfficeArea" "ResArea"    "ResidFAR"   "RetailArea"
## [16] "YearBuilt"  "YearAlter1" "ZipCode"    "YCoord"     "XCoord"
```

```r
# merge complaint types which were renamed e.g. "HEAT/HOT WATER" to "HEATING" and "PAINT - PLASTER" to
 df_NYC$complaint_type[df_NYC$complaint_type %in% "HEAT/HOT WATER"] <- "HEATING"
 df_NYC$complaint_type[df_NYC$complaint_type %in% "PAINT - PLASTER"] <- "PAINT/PLASTER"

# remove NA entries
df_NYC <- df_NYC %>% na.omit()
```

Target defnition: Pluto dataset has all houses information for the given borrows. Some houses are register more complain more often. These particular houses have features that can help in predicting future complaints.

```r
df_target <- as.numeric(df_pluto$Address %in% df_NYC$incident_address)


df_pluto['target'] <- df_target

colnames(df_pluto)
```

```
##  [1] "Address"    "BldgArea"   "BldgDepth"  "BuiltFAR"   "CommFAR"
##  [6] "FacilFAR"   "Lot"        "LotArea"    "LotDepth"   "NumBldgs"
## [11] "NumFloors"  "OfficeArea" "ResArea"    "ResidFAR"   "RetailArea"
```

```
## [16] "YearBuilt"  "YearAlter1" "ZipCode"    "YCoord"     "XCoord"
## [21] "target"
```

```
# remove Address column
df_pluto <- df_pluto[-1]
colnames(df_pluto)
```

```
##  [1] "BldgArea"   "BldgDepth"  "BuiltFAR"   "CommFAR"    "FacilFAR"
##  [6] "Lot"        "LotArea"    "LotDepth"   "NumBldgs"   "NumFloors"
## [11] "OfficeArea" "ResArea"    "ResidFAR"   "RetailArea" "YearBuilt"
## [16] "YearAlter1" "ZipCode"    "YCoord"     "XCoord"     "target"
```

```
# to make sure every column has numeric/integer class
sapply(df_pluto, class)
```

```
##   BldgArea  BldgDepth   BuiltFAR    CommFAR   FacilFAR        Lot
##  "integer"  "numeric"  "numeric"  "numeric"  "numeric"  "integer"
##    LotArea   LotDepth   NumBldgs  NumFloors OfficeArea    ResArea
##  "integer"  "numeric"  "integer"  "numeric"  "integer"  "integer"
##   ResidFAR RetailArea  YearBuilt YearAlter1    ZipCode     YCoord
##  "numeric"  "integer"  "integer"  "integer"  "integer"  "integer"
##     XCoord     target
##  "integer"  "numeric"
```

Pearson correlation matrix heatmap

```
# correlation matrix
cormat <- round(cor(df_pluto),2)
head(cormat)
```

```
##           BldgArea BldgDepth BuiltFAR CommFAR FacilFAR  Lot LotArea
## BldgArea      1.00      0.21     0.06    0.18     0.17 0.08    0.25
## BldgDepth     0.21      1.00     0.05    0.16     0.24 0.00    0.01
## BuiltFAR      0.06      0.05     1.00    0.08     0.10 0.05    0.00
## CommFAR       0.18      0.16     0.08    1.00     0.55 0.12    0.00
## FacilFAR      0.17      0.24     0.10    0.55     1.00 0.14    0.00
## Lot           0.08      0.00     0.05    0.12     0.14 1.00    0.00
##           LotDepth NumBldgs NumFloors OfficeArea ResArea ResidFAR
## BldgArea      0.22     0.17      0.40       0.41    0.54     0.16
## BldgDepth     0.18     0.02      0.27       0.14    0.15     0.22
## BuiltFAR      0.00     0.00      0.14       0.05    0.05     0.10
## CommFAR       0.04    -0.02      0.37       0.26    0.06     0.45
## FacilFAR      0.00    -0.04      0.44       0.16    0.15     0.84
## Lot          -0.01     0.02      0.25       0.04    0.10     0.15
##           RetailArea YearBuilt YearAlter1 ZipCode YCoord XCoord target
## BldgArea        0.23      0.02       0.07      NA     NA     NA   0.06
## BldgDepth       0.13      0.27       0.19      NA     NA     NA   0.17
## BuiltFAR        0.03      0.03       0.04      NA     NA     NA   0.05
## CommFAR         0.16     -0.07       0.16      NA     NA     NA   0.04
## FacilFAR        0.12     -0.05       0.24      NA     NA     NA   0.24
## Lot             0.07     -0.03       0.00      NA     NA     NA   0.03
```

```
# define a function that may help to remove the redundancy in the correlation matrix

# Get lower triangle of the correlation matrix
```

```r
  get_lower_tri<-function(cormat){
    cormat[upper.tri(cormat)] <- NA
    return(cormat)
  }
  # Get upper triangle of the correlation matrix
  get_upper_tri <- function(cormat){
    cormat[lower.tri(cormat)]<- NA
    return(cormat)
  }

# use the above defined function to set redundant entries to NA
upper_tri <- get_upper_tri(cormat)
upper_tri
```

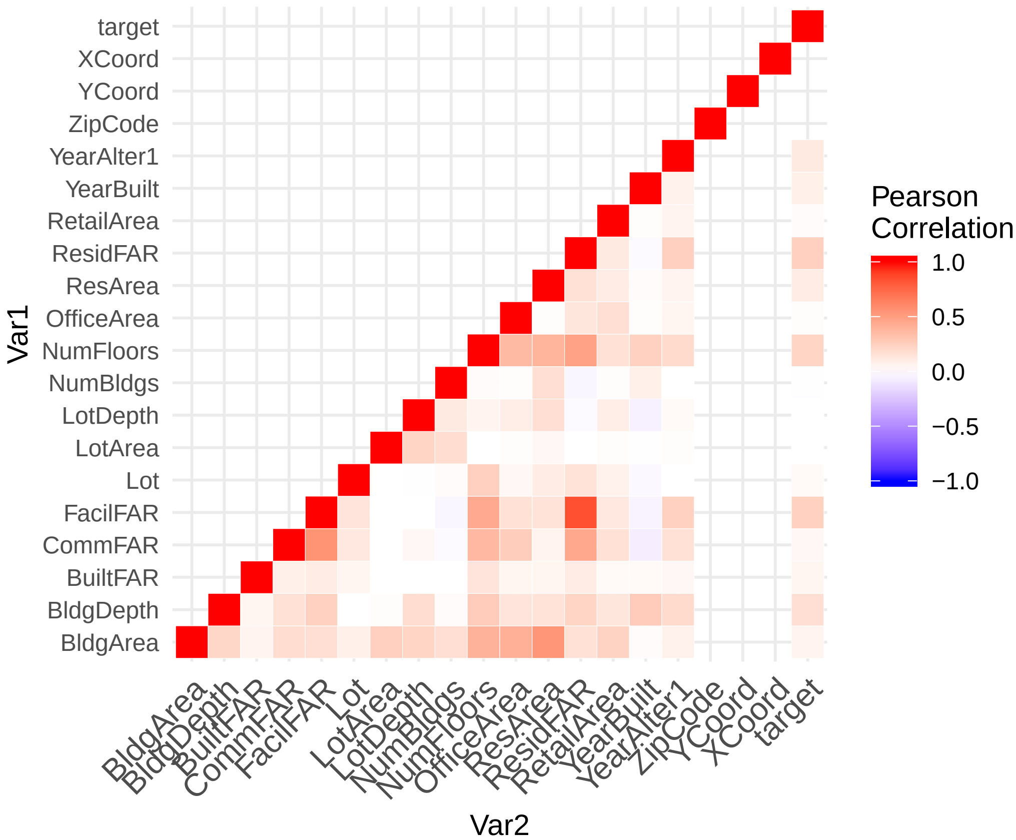
```
##             BldgArea BldgDepth BuiltFAR CommFAR FacilFAR  Lot LotArea
## BldgArea           1      0.21     0.06    0.18     0.17 0.08    0.25
## BldgDepth         NA      1.00     0.05    0.16     0.24 0.00    0.01
## BuiltFAR          NA        NA     1.00    0.08     0.10 0.05    0.00
## CommFAR           NA        NA       NA    1.00     0.55 0.12    0.00
## FacilFAR          NA        NA       NA      NA     1.00 0.14    0.00
## Lot               NA        NA       NA      NA       NA 1.00    0.00
## LotArea           NA        NA       NA      NA       NA   NA    1.00
## LotDepth          NA        NA       NA      NA       NA   NA      NA
## NumBldgs          NA        NA       NA      NA       NA   NA      NA
## NumFloors         NA        NA       NA      NA       NA   NA      NA
## OfficeArea        NA        NA       NA      NA       NA   NA      NA
## ResArea           NA        NA       NA      NA       NA   NA      NA
## ResidFAR          NA        NA       NA      NA       NA   NA      NA
## RetailArea        NA        NA       NA      NA       NA   NA      NA
## YearBuilt         NA        NA       NA      NA       NA   NA      NA
## YearAlter1        NA        NA       NA      NA       NA   NA      NA
## ZipCode           NA        NA       NA      NA       NA   NA      NA
## YCoord            NA        NA       NA      NA       NA   NA      NA
## XCoord            NA        NA       NA      NA       NA   NA      NA
## target            NA        NA       NA      NA       NA   NA      NA
##             LotDepth NumBldgs NumFloors OfficeArea ResArea ResidFAR
## BldgArea        0.22     0.17      0.40       0.41    0.54     0.16
## BldgDepth       0.18     0.02      0.27       0.14    0.15     0.22
## BuiltFAR        0.00     0.00      0.14       0.05    0.05     0.10
## CommFAR         0.04    -0.02      0.37       0.26    0.06     0.45
## FacilFAR        0.00    -0.04      0.44       0.16    0.15     0.84
## Lot            -0.01     0.02      0.25       0.04    0.10     0.15
## LotArea         0.22     0.18      0.00       0.01    0.04     0.00
## LotDepth        1.00     0.11      0.06       0.09    0.17    -0.02
## NumBldgs          NA     1.00      0.02       0.01    0.17    -0.04
## NumFloors         NA       NA      1.00       0.36    0.39     0.48
## OfficeArea        NA       NA        NA       1.00    0.01     0.13
## ResArea           NA       NA        NA         NA    1.00     0.16
## ResidFAR          NA       NA        NA         NA      NA     1.00
## RetailArea        NA       NA        NA         NA      NA       NA
## YearBuilt         NA       NA        NA         NA      NA       NA
## YearAlter1        NA       NA        NA         NA      NA       NA
## ZipCode           NA       NA        NA         NA      NA       NA
## YCoord            NA       NA        NA         NA      NA       NA
```

```
## XCoord           NA       NA         NA          NA       NA        NA
## target           NA       NA         NA          NA       NA        NA
##             RetailArea YearBuilt YearAlter1 ZipCode YCoord XCoord target
## BldgArea          0.23      0.02       0.07      NA     NA     NA   0.06
## BldgDepth         0.13      0.27       0.19      NA     NA     NA   0.17
## BuiltFAR          0.03      0.03       0.04      NA     NA     NA   0.05
## CommFAR           0.16     -0.07       0.16      NA     NA     NA   0.04
## FacilFAR          0.12     -0.05       0.24      NA     NA     NA   0.24
## Lot               0.07     -0.03       0.00      NA     NA     NA   0.03
## LotArea           0.01     -0.01       0.01      NA     NA     NA   0.00
## LotDepth          0.09     -0.06       0.03      NA     NA     NA   0.00
## NumBldgs          0.01      0.08       0.00      NA     NA     NA  -0.01
## NumFloors         0.16      0.24       0.19      NA     NA     NA   0.22
## OfficeArea        0.17      0.01       0.05      NA     NA     NA   0.01
## ResArea           0.10      0.02       0.06      NA     NA     NA   0.10
## ResidFAR          0.11     -0.02       0.25      NA     NA     NA   0.25
## RetailArea        1.00      0.01       0.06      NA     NA     NA   0.02
## YearBuilt           NA      1.00       0.07      NA     NA     NA   0.08
## YearAlter1          NA        NA       1.00      NA     NA     NA   0.11
## ZipCode             NA        NA         NA       1     NA     NA     NA
## YCoord              NA        NA         NA      NA      1     NA     NA
## XCoord              NA        NA         NA      NA     NA      1     NA
## target              NA        NA         NA      NA     NA     NA   1.00
```

```r
# Melt the correlation matrix
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

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

```r
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Heatmap
library(ggplot2)
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
   name="Pearson\nCorrelation") +
  theme_minimal()+
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()
```




```r
#Selecting correlated features. Here I set the threshold to be 0.12
cor_target = abs(cormat[,"target"])
cor_target<-cor_target[!is.na(cor_target)]
relevant_features = cor_target[cor_target>0.12]
print(relevant_features)
```

```
## BldgDepth  FacilFAR NumFloors  ResidFAR    target
##      0.17      0.24      0.22      0.25      1.00
```

```r
# let's find the features which are highly correlated among themselves. If so, we may use only one of t
df_corr_2features <- df_pluto%>% select("ResidFAR","FacilFAR")
round(cor(df_corr_2features),2)
```

```
##          ResidFAR FacilFAR
## ResidFAR     1.00     0.84
## FacilFAR     0.84     1.00
```

```r
# This implies we can drop FacilFAR feature and keep ResiFAR feature as they are highly correlated. So
```

Random forrest method for feature selection

```r
# Here we will drop only three features ("target", "XCoord", "YCoord") and train the model on rest of t
# data for features set and target

drops <- c("XCoord", "YCoord")
df_PLUTO <- df_pluto[ , !(names(df_pluto) %in% drops)]
```

```
df_PLUTO <- df_pluto[ , !(names(df_pluto) %in% drops)]

# Validation set will be 30% of pluto data
test_index <- createDataPartition(y = df_pluto$target, times = 1, p = 0.3, list = FALSE)

train_set <- df_PLUTO[-test_index,]
test_set <- df_PLUTO[test_index,]

# test dataset can be further modified by removing target column
test_set_CM <- test_set
test_set <- test_set %>% select(-target)
```

```
# In case of only 2 classifier, one can use linear regression, but here I am using Random Forest method

# convert target as factor
train_set$target <- as.character(train_set$target)
train_set$target <- as.factor(train_set$target)

tree_count <- 100
set.seed(1)
model<- randomForest(target~.,train_set,ntree=tree_count,importance=TRUE,na.action=na.omit)



# convert all iterations into matrix form
imp_score_matrix <- importance(model)
imp_score_matrix
```

```
##                     0           1 MeanDecreaseAccuracy MeanDecreaseGini
## BldgArea     58.07064  10.3413868             60.96335       14340.8476
## BldgDepth    63.22847  39.1884914             76.33508       10258.4894
## BuiltFAR     39.34641  10.9304326             47.47891       16011.6699
## CommFAR      21.42778  -6.0548737             20.97901         803.5120
## FacilFAR     26.69677  17.3208534             31.33298        2933.2561
## Lot          97.43836  11.0142032             92.99875       16694.5160
## LotArea      73.05922  14.5811908             78.72567       10590.3002
## LotDepth    102.69171  18.1112414            107.52054        7324.9440
## NumBldgs     46.85515   9.5420549             45.46514        1824.8036
## NumFloors    30.41020  25.0890736             42.50460        6109.7303
## OfficeArea   23.98086  -0.9130089             24.62711         916.3013
## ResArea      43.50018 123.7801364             64.37517       21454.0713
## ResidFAR     38.97624  17.4247331             43.31149        4592.3480
## RetailArea   16.17093  26.4605123             21.97592        2242.1306
## YearBuilt    86.46064  38.8229149             96.84888        8977.1389
## YearAlter1   40.04927   0.8526064             38.34342        2708.5971
## ZipCode      62.84478  37.6976475             73.93454       10217.8819
```

```
rm('df_PLUTO', 'model', 'df_NYC')
```

The Ransom forest method provides a table of feature importance. It shows two varible 'MeanDecreaseAccuracy', 'MeanDecreaseGini'. Larger the numbers are, greater their feature importance is. A cursory look at the table reveals that features like 'Lot', 'BuiltFAR', 'BldgArea', 'ResArea', 'NumFloors' are most important one.

Concluding Remarks: solution of problem 3: The Pearson correlation matrix shows that there are 4 important features: 'BldgDepth', 'FacilFAR', 'NumFloors', 'ResidFAR'. This is further confirmed by Random forest method. One can further use bootstrap method to find several feature importance for different randomized dataset and then take mean of it to have more accurate results.

Problem 4: Can a predictive model be built for a future prediction of the possibility of complaints of the type that you have identified in response to question 1?

So far, we have pointed out the important features in the pluto data set and did some exploratory analysis. Problem 4 poses a new set of challenge. It asks to predict the future. I don't know who can be well suited for the job 'Prophet', 'Philosopher', or 'Professor'. I beleive everyone will look for 'history' or in simple words time dependent feature sets.

However our analysis shows that features are static in nature. To predict the HEATING complaints, we may need additional data from external sources that has some temporal dependencies e.g. weather dataset over years. With the given dataset, I think it would be good to try our hand over 'Time series analysis' and get a rough future estimate about number of complaints.

Time Series Analysis

```
#colnames(df_NYC)
#dt <- as.POSIXct(df_NYC$created_date)
#df_NYC <- df_NYC %>% mutate(year_month = format(dt, "%Y-%m"))

#rm(dt)

#df_TS <- df_NYC %>%
#  select('year_month', 'unique_key')%>%
#  filter(year_month<2019)
# below chunk of codes have not been varified so I have not put them in my Pdf file
#df_TS %>%  sort(df_TS$year_month, decreasing = FALSE)

#TS_complaint <- df_TS %>%
#na.omit() %>% # omit missing values
#select(year, complaint_type) %>% # select columns we are interested in
#mutate(year_month = as.factor(year_month)) %>% # turn year in factors
#mutate(year_month = as.numeric(levels(year_month))[year_month]) %>%
#group_by(year, unique_key) %>% # group data by year and complaint_Type
#summarise(number = n())  # count


#TS_complaint %>%
#ggplot(aes(x = year_month, y = number)) +
#geom_line()
```

```
# Trend, Seasonality and error
#decomposedRes <- decompose(tsData, type="mult") # use type = "additive" for additive components
#plot (decomposedRes) # see plot below
#stlRes <- stl(tsData, s.window = "periodic")

# Further one can use ARIMA method for modelling. I am still in learning phase to use this method.
```

Concluding Remarks on the Project: In this project I have ingested data from external web resources. Performed some simple queries to access the relevant files. Furthermore, the problem was defined very clearly and was broken into smaller set of problems. A clear attack plan was made and susequent analysis is performed. The exploratory data analysis provided an insight into the data and helped to pick an appropriate model. Important features have been selected using Pearson correlation and Random Forest method. Results were found in good agreement. A deeper insight was obtained in the data and a conclusion was made that time dependent features will be needed to predict the future complaints. Neverthless, ARIMA, a time series based analysis have been suggested into this context.