Analysis of NYPD Shooting Incident Dataset

MG

April 11, 2024

Summary

This report analyses the NYPD Shooting Incidents Dataset from https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic.

This report will study how borough, location type, season of the year, and time of the day affect the fatal outcome of a shooting (predicting the `STATISTICAL_MURDER_FLAG` variable).

Importing and Cleaning the data

```
set.seed(12345)

# required libraries
library(lubridate)
library(dplyr)
library(caret)
library(randomForest)
library(gbm)
library(C50)

# multi-core processing for training
#library(doMC)
#registerDoMC(cores = 3)
```

First, lets import the data from the CSV file, and replace any blank cells with `NA`

```
nypd_raw <- read.csv(
  'https://raw.githubusercontent.com/mgribov/msds-dsaaf/main/nypd/NYPD_Shooting_Incident_Data__Historic
  header=T,
  na.strings=c("","NA")
)
```

Next, we can see what the column names are, and we can look up their description from the metadata information included with the dataset. (https://data.cityofnewyork.us/api/views/833y-fsy8/columns.json)

```
colnames(nypd_raw)
```

```
##  [1] "INCIDENT_KEY"            "OCCUR_DATE"
##  [3] "OCCUR_TIME"              "BORO"
##  [5] "PRECINCT"                "JURISDICTION_CODE"
##  [7] "LOCATION_DESC"           "STATISTICAL_MURDER_FLAG"
##  [9] "PERP_AGE_GROUP"          "PERP_SEX"
## [11] "PERP_RACE"               "VIC_AGE_GROUP"
```

```
## [13] "VIC_SEX"                 "VIC_RACE"
## [15] "X_COORD_CD"              "Y_COORD_CD"
## [17] "Latitude"                "Longitude"
## [19] "Lon_Lat"
```

We will use only some of the columns in the dataset, so we'll first add new columns for our analysis, and then create a new dataframe with only the desired columns.

First, create a new field `timestamp` from `OCCUR_DATE` and `OCCUR_TIME`

```
nypd_raw <- within(
  nypd_raw,
  {timestamp=strptime(paste(OCCUR_DATE, ' ', OCCUR_TIME), "%m/%d/%Y%H:%M:%S")}
)
```

Next, lets create a new column which represents the season (Spring, Summer, Fall, Winter), based on the date.

We will use `quarters()` function to determine which quarter the date belongs to, which would be roughly the same as a season.

```
nypd_raw$season <- as.factor(quarters(nypd_raw$timestamp))
```

Next, we will create a new column which represents the part of the day for the incident (Night, Morning, Afternoon, Evening).

```
breaks <- hour(hm("00:00", "6:00", "12:00", "18:00", "23:59"))
labels <- c("Night", "Morning", "Afternoon", "Evening")
nypd_raw$daypart <- as.factor(
  cut(
    x=hour(nypd_raw$timestamp),
    breaks = breaks,
    labels = labels,
    include.lowest=TRUE
  )
)
```

Now, create the new data frame for analysis and modeling, and simplify column names.

```
nypd <- data.frame(
  as.factor(nypd_raw$BORO),
  as.factor(nypd_raw$LOCATION_DESC),
  as.factor(nypd_raw$STATISTICAL_MURDER_FLAG),
  nypd_raw$season,
  nypd_raw$daypart
)

# clean up column names
names(nypd)[1] <- "boro"
names(nypd)[2] <- "location_desc"
names(nypd)[3] <- "is_fatal"
names(nypd)[4] <- "season"
names(nypd)[5] <- "daypart"
```

Next, make sure to drop any rows which have missing values

```
nypd <- na.omit(nypd)
```

Summary of our new dataset

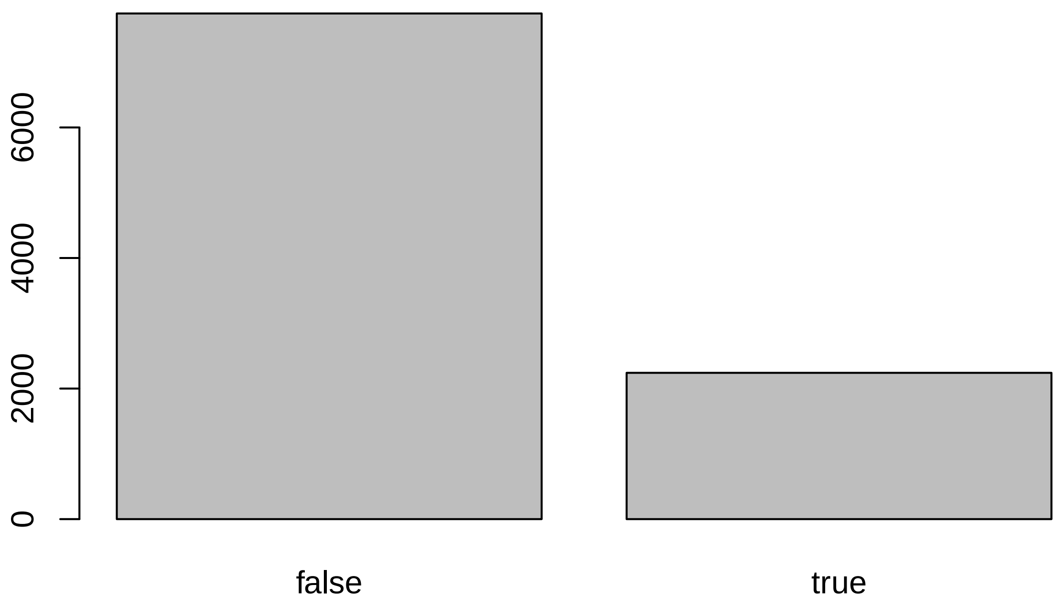
```r
summary(nypd)
```

```
##             boro                          location_desc   is_fatal    season
##  BRONX        :2668   MULTI DWELL - PUBLIC HOUS:4230   false:7746   Q1:2008
##  BROOKLYN     :4285   MULTI DWELL - APT BUILD  :2551   true :2241   Q2:2600
##  MANHATTAN    :1371   PVT HOUSE                : 858                Q3:3048
##  QUEENS       :1365   GROCERY/BODEGA           : 572                Q4:2331
##  STATEN ISLAND: 298   BAR/NIGHT CLUB           : 558
##                       COMMERCIAL BLDG          : 234
##                       (Other)                  : 984
##       daypart
##  Night    :3786
##  Morning  : 773
##  Afternoon:2142
##  Evening  :3286
##
##
##
```

Analysis and Visualization

Next, lets see what values the `STATISTICAL_MURDER_FLAG` column has. According to the data, most shooting incidents are not fatal.

```r
plot(nypd$is_fatal)
```



Analysis of fatal outcomes

First, we will create a subset of the data with only fatal outcomes for the plots.

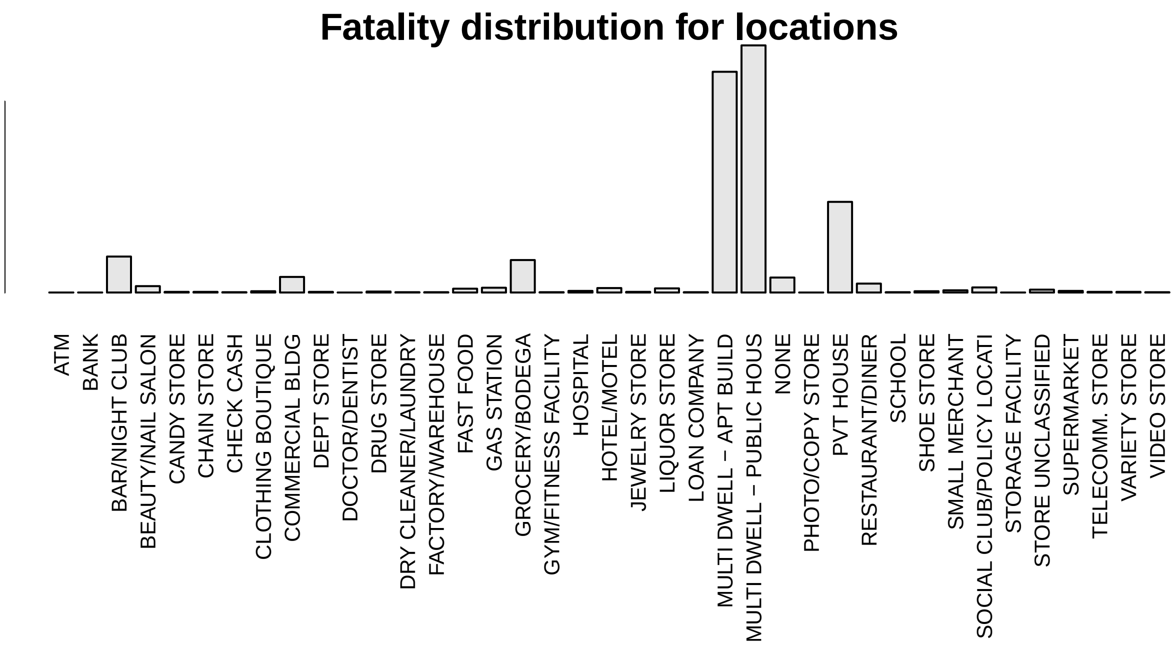
```r
fatal <- subset(nypd, is_fatal == 'true')
```

Location type (`LOCATION_DESC` column).

```
tab <- table(
  fatal$is_fatal,
  fatal$location_desc
)

par(mar=c(15, 0, 1, 1))

barplot(
  tab,
  main="Fatality distribution for locations",
  las=2,
  cex.axis=0.1,
  cex.names=0.7
)
```



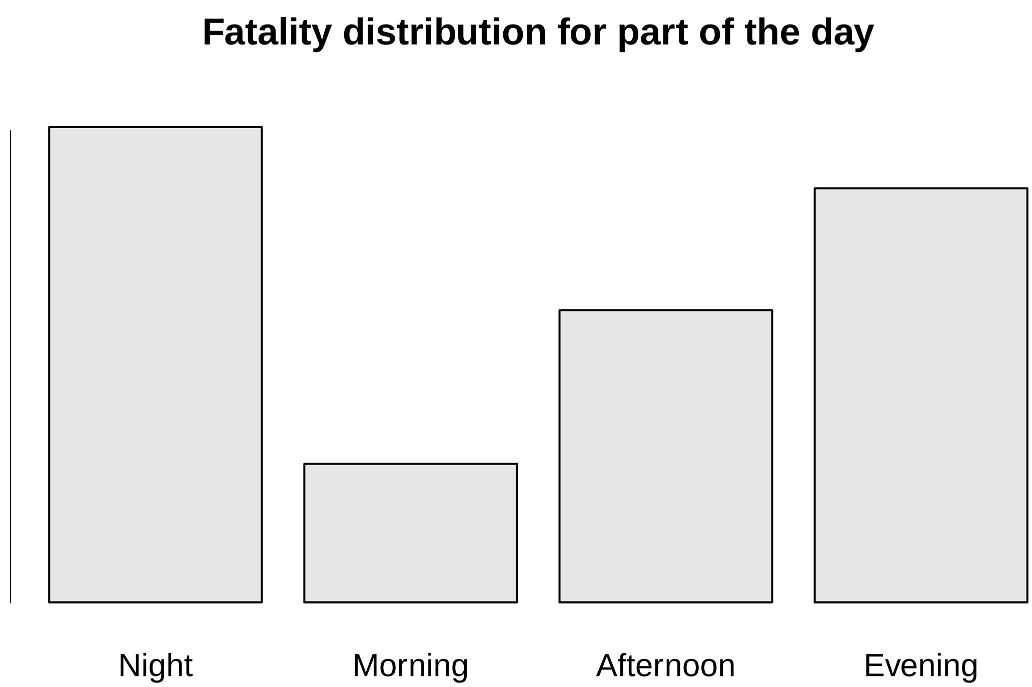
According to the plot, apartment buildings and public housing are responsible for majority of the fatal incidents.

Part of the day (`daypart` column).

```
tab <- table(fatal$is_fatal, fatal$daypart)

par(mar=c(5, 0, 5, 5))

barplot(
  tab,
  main="Fatality distribution for part of the day"
)
```



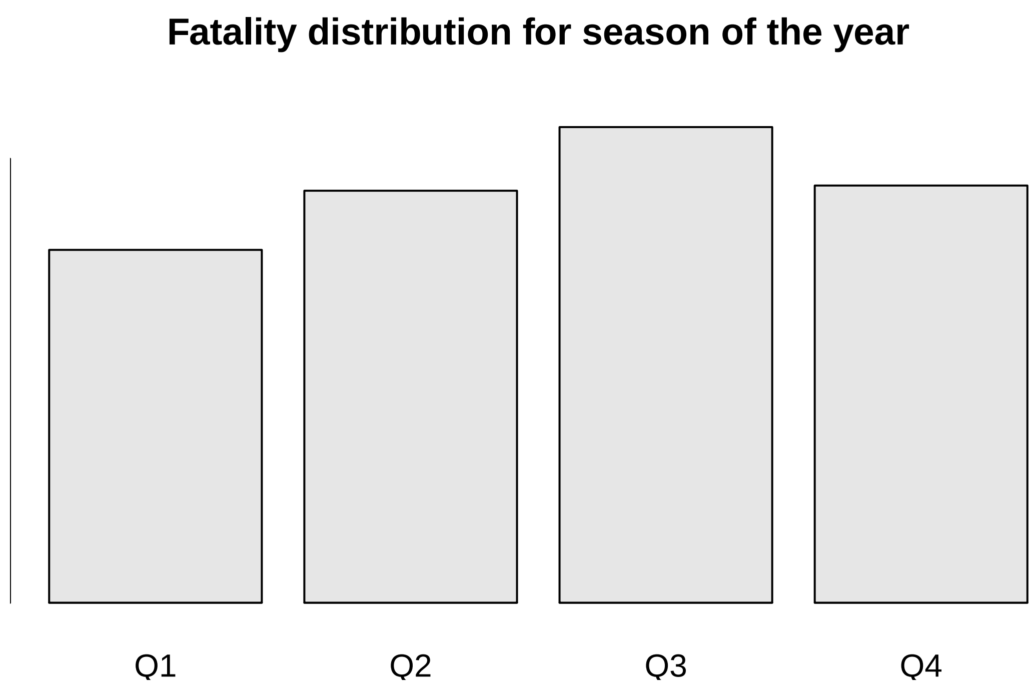
Accoding to the plot, morning has the least fatal accidents, and night time has the most.

Season of the year (season column).

```
tab <- table(fatal$is_fatal, fatal$season)

par(mar=c(5, 0, 5, 5))

barplot(
  tab,
  main="Fatality distribution for season of the year"
)
```



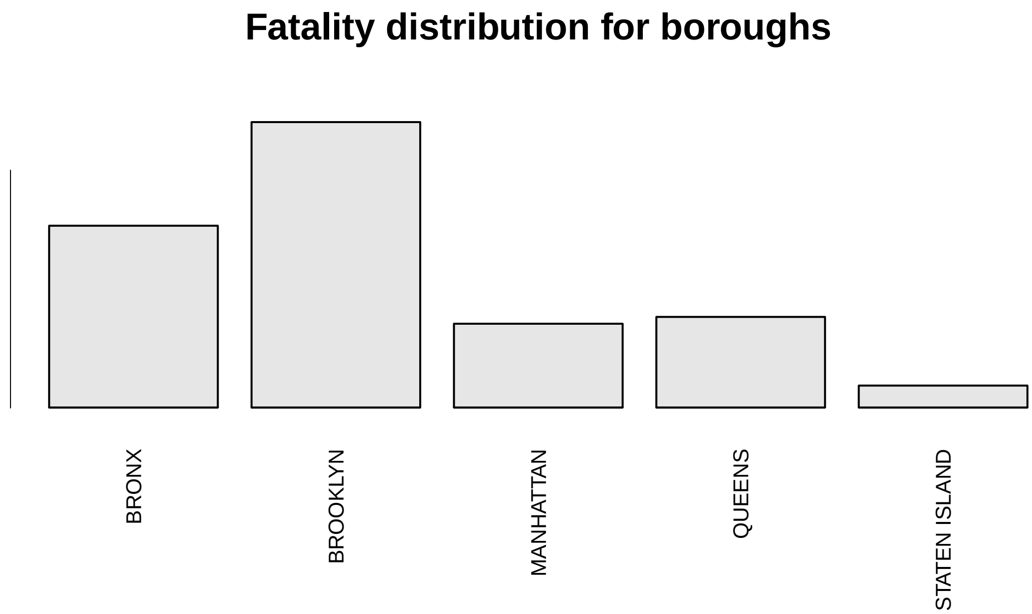
According to the plot, summer has the most fatal incidents, and winter has the least.

Borough (season column).

```
tab <- table(fatal$is_fatal, fatal$boro)

par(mar=c(10, 0, 5, 5))

barplot(
  tab,
  main="Fatality distribution for boroughs",
  las=2,
  cex.axis=0.1,
  cex.names=0.7
)
```



According to the plot, Brooklyn has the most fatal incidents.

Building a model

Predicting an incident outcome based our data is a classfication problem.

The data has significantly more non-fatal outcomes than fatal, so first we will create a data set with equal number of fatal and non-fatal outcomes, and create a training and testing sets.

```
nonfatal <- subset(nypd, is_fatal == 'false')
nonfatal <- sample_n(nonfatal, nrow(fatal))

model_data <- rbind(fatal, nonfatal)

model_data_train <- sample_frac(model_data, 0.9)
model_data_test <- sample_frac(model_data, 0.1)
```

We will train and evaluate several models used with classfication tasks: Random Forests, Decision Trees using Stochastic Gradient Boosting, Decision Trees using C5.0 algorithm, and K-Nearest Neighbors.

NOTE: the code to generate the models is included, but commented out and instead replaced with loading saved models from original training, to speed up the knitting process.

```
#fitControl <- trainControl(
#  allowParallel = TRUE,
#  ## 10-fold CV
#  method = "repeatedcv",
#  number = 10,
#  ## repeated ten times
#  repeats = 10
#)

#rf <- train(
#  is_fatal ~ .,
#  data=model_data_train,
#  method="rf",
```

```
#  metric="Kappa",
#  trControl=fitControl
#)
#saveRDS(rf, "rf_full_model.rds")
download.file('https://raw.githubusercontent.com/mgribov/msds-dsaaf/main/nypd/rf_full_model.rds', destfi
rf <- readRDS("./rf_full_model.rds")

# one of the initial hypotheses was that location of the shooting
# would affect the outcome, this was not the case
#rf2 <- train(
#  is_fatal ~ location_desc,
#  data=model_data_train,
#  method="rf",
#  metric="Kappa",
#  trControl=fitControl
#)
#saveRDS(rf2, "rf_location_desc_model.rds")
#rf2 <- readRDS("rf_location_desc_model.rds")

# @todo for some reason this fails in knittr, but works in console
# so not using this model
#pred_glmn <- predict(glmn, model_data_test)
#glmn <- train(
#  is_fatal ~ .,
#  data=model_data_train,
#  method="glmnet",
#  family = 'binomial',
#  trControl=fitControl
#)
#saveRDS(glmn, "glmnet_full_model.rds")
#glmn <- readRDS("glmnet_full_model.rds")

#gb <- train(
#  is_fatal ~ .,
#  data=model_data_train,
#  method="gbm",
#  trControl=fitControl,
#  verbose=FALSE,
#  metric="Kappa",
#  na.action = na.omit
#)
#saveRDS(gb, "gbm_full_model.rds")
download.file('https://raw.githubusercontent.com/mgribov/msds-dsaaf/main/nypd/gbm_full_model.rds', desti
gb <- readRDS("./gbm_full_model.rds")

#c50 <- train(
#  is_fatal ~ .,
#  data=model_data_train,
#  method="C5.0",
#  trControl=fitControl,
#  verbose=FALSE,
#  metric="Kappa"
#)
```

```
#saveRDS(c50, "c50_full_model.rds")
download.file('https://raw.githubusercontent.com/mgribov/msds-dsaaf/main/nypd/c50_full_model.rds', dest
c50 <- readRDS("./c50_full_model.rds")

#knn <- train(
#  is_fatal ~ .,
#  data=model_data_train,
#  method="knn",
#  metric="Kappa",
#  trControl=fitControl
#)
#saveRDS(knn, "knn_full_model.rds")
download.file('https://raw.githubusercontent.com/mgribov/msds-dsaaf/main/nypd/knn_full_model.rds', dest
knn <- readRDS("./knn_full_model.rds")
```

Analyzing model performance

```
# generate predictions for each model from test dataset
pred_rf <- predict(rf, model_data_test)
pred_gb <- predict(gb, model_data_test)
pred_c50 <- predict(c50, model_data_test)
pred_knn <- predict(knn, model_data_test)
```

```
# confusion matrix accuracy for random forest
confusionMatrix(pred_rf, model_data_test$is_fatal)$overall[["Accuracy"]]
```

```
## [1] 0.546875
```

```
# confusion matrix accuracy for k-nearest neighbors
confusionMatrix(pred_knn, model_data_test$is_fatal)$overall[["Accuracy"]]
```

```
## [1] 0.53125
```

```
# confusion matrix accuracy for stochastic gradient boost decision trees
confusionMatrix(pred_gb, model_data_test$is_fatal)$overall[["Accuracy"]]
```

```
## [1] 0.5267857
```

```
# confusion matrix accuracy for C.50 decision trees
confusionMatrix(pred_c50, model_data_test$is_fatal)$overall[["Accuracy"]]
```

```
## [1] 0.5267857
```

Conclusion

Based on the data exploration and visualization, most of the incidents are non-fatal, and night time, summer, Brooklyn, and apartment buildings have the highest occurrence of fatal incidents.

All of the models had relatively low accuracy on the test data set, less than 60%, which suggests that none of the studied factors - time of the day, season of the year, the borough, or the location type of the incident - have a significant impact on fatal vs. non-fatal incident outcome.

Random Forest model had the highest accuracy value, so we can examine it for the top 20 most important features:

```
varImp(rf, scale=FALSE)
```

```
## rf variable importance
```

```
##
##   only 20 most important variables shown (out of 48)
##
##                                         Overall
## location_descPVT HOUSE                   5.9512
## location_descMULTI DWELL - PUBLIC HOUS  4.8153
## daypartMorning                           3.7902
## location_descMULTI DWELL - APT BUILD    2.3499
## location_descDRY CLEANER/LAUNDRY        1.3752
## location_descHOTEL/MOTEL                1.1078
## boroQUEENS                               1.0876
## seasonQ4                                 0.9115
## location_descRESTAURANT/DINER           0.8831
## location_descGROCERY/BODEGA             0.7877
## location_descBEAUTY/NAIL SALON          0.7432
## boroBROOKLYN                             0.7318
## seasonQ2                                 0.7158
## daypartAfternoon                         0.6794
## boroMANHATTAN                            0.6731
## location_descNONE                        0.6622
## daypartEvening                           0.6531
## location_descBAR/NIGHT CLUB             0.6427
## location_descLIQUOR STORE               0.6069
## seasonQ3                                 0.5861
```

According to the variable importance table, location type is a very important feature, and its high importance values (e.g. `PVT_HOUSE`, `MULTI DWELL - PUBLIC HOUS`, `MULTI DWELL - APT BUILD`) reflect the findings from the visualization for fatality distribution across all the possible location types, but since the model accuracy is low, we still cannot conclude that location type has significant predictive power for the fatal outcome of a shooting, the model simply confirms what we already saw from the summary of the raw data - majority of shootings, both fatal and not, happen in someone's home.

Summary of location type for all shooting outcomes:

```r
summary(nypd[c('location_desc', 'is_fatal')])
```

```
##                    location_desc    is_fatal
##  MULTI DWELL - PUBLIC HOUS:4230   false:7746
##  MULTI DWELL - APT BUILD  :2551   true :2241
##  PVT HOUSE                : 858
##  GROCERY/BODEGA           : 572
##  BAR/NIGHT CLUB           : 558
##  COMMERCIAL BLDG          : 234
##  (Other)                  : 984
```

Summary of location type for non-fatal outcomes only:

```r
summary(nonfatal[c('location_desc', 'is_fatal')])
```

```
##                    location_desc    is_fatal
##  MULTI DWELL - PUBLIC HOUS:1021   false:2241
##  MULTI DWELL - APT BUILD  : 521   true :   0
##  PVT HOUSE                : 151
##  BAR/NIGHT CLUB           : 132
##  GROCERY/BODEGA           : 131
##  COMMERCIAL BLDG          :  58
##  (Other)                  : 227
```

Biases in the data

Based on the analysis, Brooklyn and apartment buildings contain the highest number of fatal incidents.

However, Brooklyn is the most populous borough in New York City (https://www.census.gov/quickfacts/fact/table/newyorkcitynewyork,bronxcountybronxboroughnewyork,kingscountybrooklynboroughnewyork,newyorkcountymanhattanboroughnewyork,queenscountyqueensboroughnewyork,richmondcountystatenislandboroughnewyork/PST045219), and New York City is a very densely populated metropolis, so apartment buildings represent majority of available real estate (https://www.valuepenguin.com/new-york-city-renters-statistics#building-size).

This introduces a bias to the data analysis and modeling, and any conclusions based on this data can be applied only to New York City, or a metropolis with similar population and real estate breakdown.