NYPD Shooting Incidents

5/3/2021

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

```r
url = "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"

shooting_data = read_csv(url)
summary(shooting_data)
```

```
##   INCIDENT_KEY        OCCUR_DATE          OCCUR_TIME            BORO
##  Min.   :  9953245   Length:23568       Length:23568      Length:23568
##  1st Qu.: 55317014   Class :character   Class1:hms        Class :character
##  Median : 83365370   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :102218616                      Mode  :numeric
##  3rd Qu.:150772442
##  Max.   :222473262
##
##     PRECINCT      JURISDICTION_CODE LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Min.   :0.0000    Length:23568       Mode :logical
##  1st Qu.: 44.00   1st Qu.:0.0000    Class :character   FALSE:19080
##  Median : 69.00   Median :0.0000    Mode  :character   TRUE :4488
##  Mean   : 66.21   Mean   :0.3323
##  3rd Qu.: 81.00   3rd Qu.:0.0000
##  Max.   :123.00   Max.   :2.0000
##                   NA's   :2
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
##  Length:23568       Length:23568       Length:23568       Length:23568
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_SEX            VIC_RACE           X_COORD_CD        Y_COORD_CD
##  Length:23568       Length:23568       Min.   : 914928   Min.   :125757
##  Class :character   Class :character   1st Qu.: 999900   1st Qu.:182565
##  Mode  :character   Mode  :character   Median :1007645   Median :193482
##                                        Mean   :1009363   Mean   :207312
##                                        3rd Qu.:1016807   3rd Qu.:239163
##                                        Max.   :1066815   Max.   :271128
##
##     Latitude       Longitude        Lon_Lat
##  Min.   :40.51   Min.   :-74.25   Length:23568
```

```
##  1st Qu.:40.67   1st Qu.:-73.94   Class :character
##  Median :40.70   Median :-73.92   Mode  :character
##  Mean   :40.74   Mean   :-73.91
##  3rd Qu.:40.82   3rd Qu.:-73.88
##  Max.   :40.91   Max.   :-73.70
##
```

We can see that several categories have a decent amount of missing data. Let's quantify exactly what percentage of info is missing for one of the features in the dataset:

```
mean(is.na(shooting_data$LOCATION_DESC)) #Check proportion of missing values for a single feature
```

```
## [1] 0.5762475
```

```
#md.pattern(shooting_data) #Check raw number of missing cases for each feature
sum(is.na(shooting_data)) #Total number of missing cell values
```

```
## [1] 38892
```

We have several variables missing many entries, some with over 50% of the values absent! There are a handful of ways to deal with this kind of data missing completely at random (MCAR). One method is imputation, in which the missing values are filled in using the existing values as a reference. This can be useful for smaller amounts of missing data, but when over half the values are missing for a feature, it's going to introduce too much bias. Imputing missing data generally works better for continuous values rather than categorical values as well, although there are still ways to impute for missing categorical data. Mode imputation is a spin on regular imputation; the most common category is assigned to all missing values in a feature, but similar to regular imputation, there is an increase in bias and a decrease in variance. Multinomial logistic regression imputation can be used as long as the feature has a small number of categories, so it might have been useful for imputing perpetrator sex if less data was missing. Predictive mean matching imputation can work well on ordered categorical data, such as perpetrator age group, but again the percentage of missing data is so high that the most logical solution is to simply exclude any features missing large swaths of data or exclude any observation that has data missing for any of the features.

In the case of this dataset, the solution depends on how important analysis of the perpetrator is, since most of the heavily missing data is focused on them. If perp analysis is valued here, remove incomplete observations and keep all of the features; if not, remove those perp features and keep all of the observations.

```
shooting_cleaned <- shooting_data %>%
  select(-c(INCIDENT_KEY, LOCATION_DESC, PERP_AGE_GROUP, PERP_SEX, PERP_RACE, X_COORD_CD, Y_COORD_CD,
            Lon_Lat)) %>%
  mutate(OCCUR_DATE = mdy(OCCUR_DATE), JURISDICTION_CODE = as.factor(JURISDICTION_CODE),
         STATISTICAL_MURDER_FLAG = as.factor(STATISTICAL_MURDER_FLAG), PRECINCT = as.factor(PRECINCT)) %
  mutate_if(is.character, as.factor) %>%
  na.omit()

shooting_cleaned
```

```
## # A tibble: 23,566 x 11
##    OCCUR_DATE OCCUR_TIME BORO      PRECINCT JURISDICTION_CO~ STATISTICAL_MURDER~
##    <date>     <time>     <fct>     <fct>    <fct>            <fct>
##  1 2019-08-23 22:10      QUEENS    103      0                FALSE
##  2 2019-11-27 15:54      BRONX     40       0                FALSE
```

```
##  3 2019-02-02 19:40      MANHATTAN 23       0                 FALSE
##  4 2019-10-24 00:52      STATEN I~ 121      0                 TRUE
##  5 2019-08-22 18:03      BRONX     46       0                 FALSE
##  6 2019-06-07 17:50      BROOKLYN  73       0                 FALSE
##  7 2019-03-11 16:30      BROOKLYN  81       0                 FALSE
##  8 2019-10-03 01:45      BROOKLYN  67       0                 TRUE
##  9 2019-02-17 03:00      QUEENS    114      2                 FALSE
## 10 2019-07-10 02:56      BROOKLYN  69       0                 FALSE
## # ... with 23,556 more rows, and 5 more variables: VIC_AGE_GROUP <fct>,
## #   VIC_SEX <fct>, VIC_RACE <fct>, Latitude <dbl>, Longitude <dbl>
```

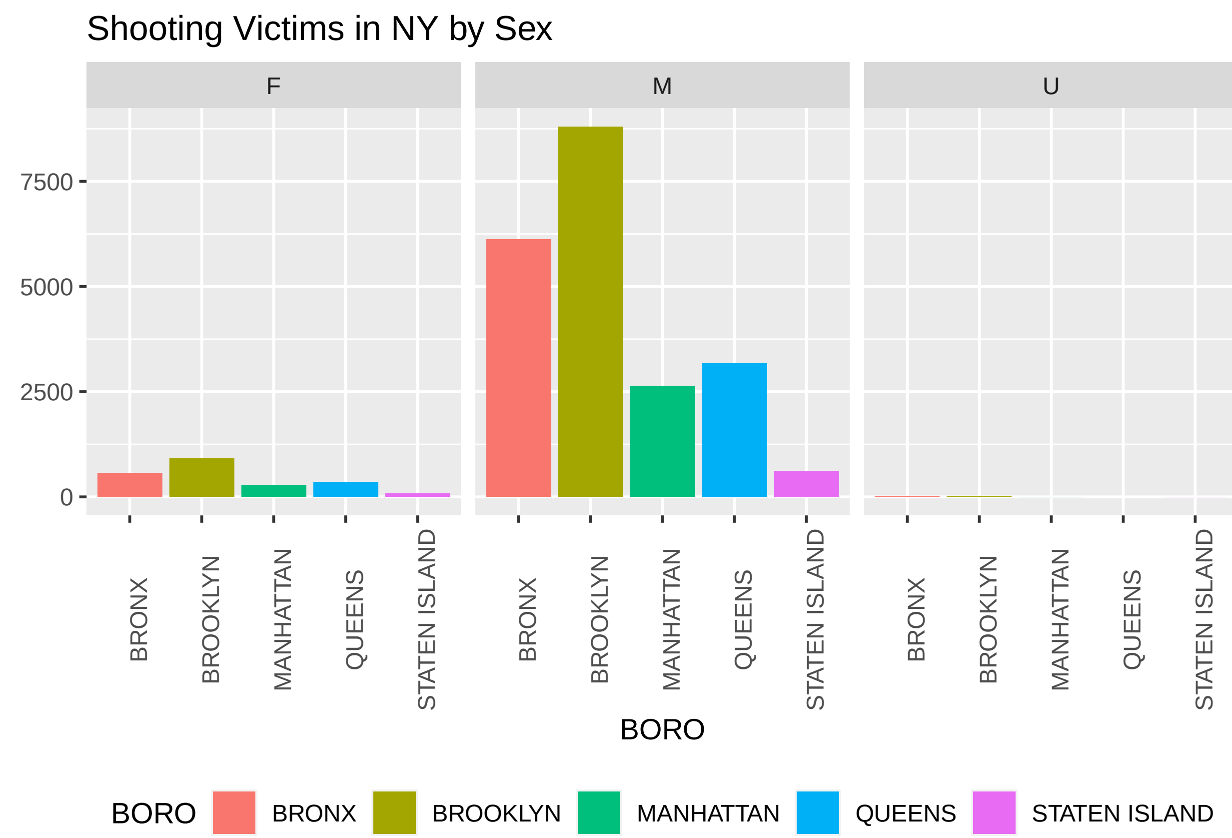
```r
summary(shooting_cleaned)
```

```
##    OCCUR_DATE          OCCUR_TIME                   BORO         PRECINCT
##  Min.   :2006-01-01   Length:23566      BRONX        :6700   75      : 1367
##  1st Qu.:2008-12-30   Class1:hms        BROOKLYN     :9722   73      : 1282
##  Median :2012-02-26   Class2:difftime   MANHATTAN    :2920   67      : 1102
##  Mean   :2012-10-03   Mode  :numeric    QUEENS       :3526   79      :  920
##  3rd Qu.:2016-02-27                     STATEN ISLAND: 698   44      :  842
##  Max.   :2020-12-31                                          47      :  815
##                                                              (Other):17238
##  JURISDICTION_CODE STATISTICAL_MURDER_FLAG VIC_AGE_GROUP   VIC_SEX
##  0:19624           FALSE:19078             <18     : 2525   F: 2195
##  1:   54           TRUE : 4488             18-24   : 8999   M:21351
##  2: 3888                                   25-44   :10286   U:   20
##                                            45-64   : 1536
##                                            65+     :  155
##                                            UNKNOWN:   65
##
##                              VIC_RACE        Latitude       Longitude
##  AMERICAN INDIAN/ALASKAN NATIVE:    9   Min.   :40.51   Min.   :-74.25
##  ASIAN / PACIFIC ISLANDER      :  320   1st Qu.:40.67   1st Qu.:-73.94
##  BLACK                         :16845   Median :40.70   Median :-73.92
##  BLACK HISPANIC                : 2244   Mean   :40.74   Mean   :-73.91
##  UNKNOWN                       :  102   3rd Qu.:40.82   3rd Qu.:-73.88
##  WHITE                         :  615   Max.   :40.91   Max.   :-73.70
##  WHITE HISPANIC                : 3431
```

I have chosen to remove the features that were missing too much data and keep the vast majority of the observations intact. Only a couple of observations were missing enough feature values that they had to be removed with `na.omit()`.

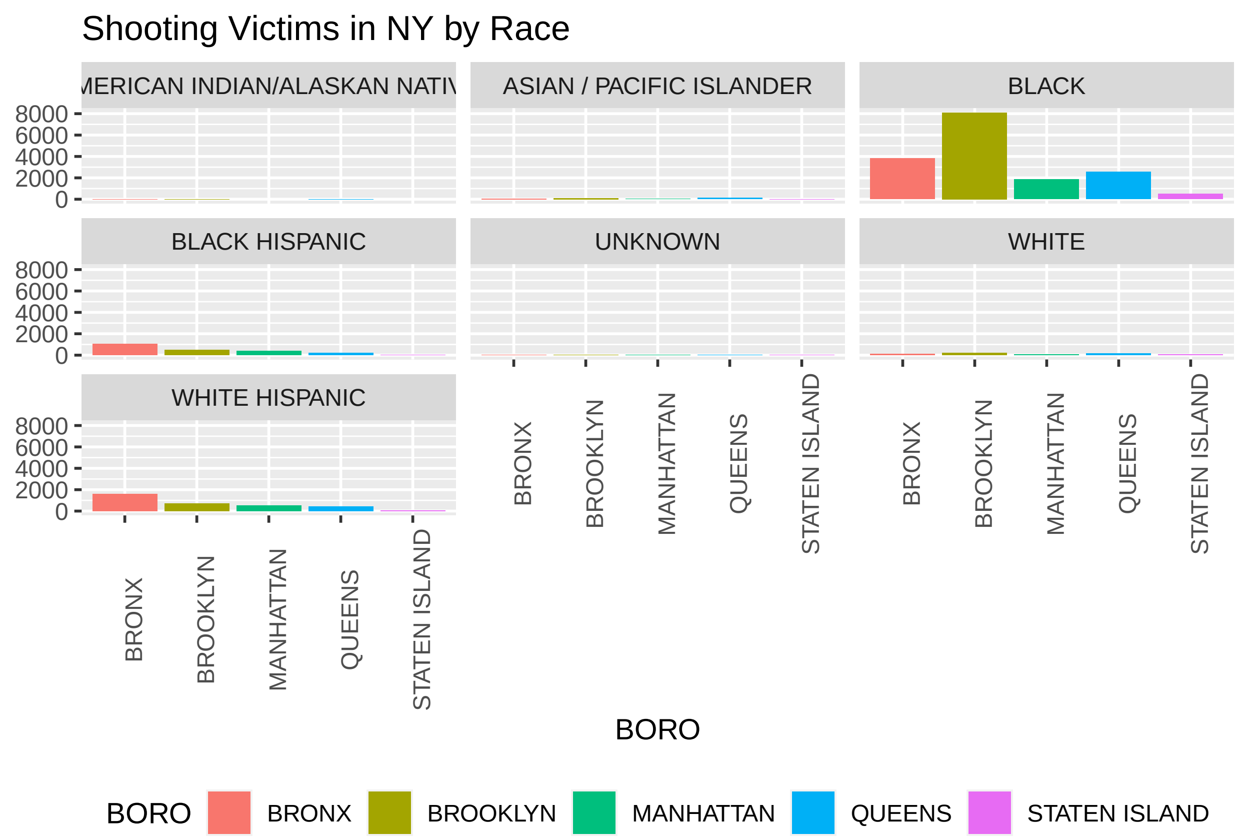
By faceting the number of shootings by the sex of the victim, we are able to see a sex breakdown for each of the five boroughs. We can easily see that males are overwhelmingly the victims of shootings in New York.

```r
ggplot(shooting_cleaned) +
  geom_bar(aes(x = BORO, fill = BORO)) +
  facet_wrap(~VIC_SEX) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shooting Victims in NY by Sex", y = NULL)
```



Similarly, we can also facet the shootings per borough by the racial attributes of the victims, revealing that the victims are also overwhelmingly black:

```r
ggplot(shooting_cleaned) +
  geom_bar(aes(x = BORO, fill = BORO)) +
  facet_wrap(~VIC_RACE) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Shooting Victims in NY by Race", y = NULL)
```



Finally, I will train a random forest model on a portion of the dataset and use the model to try to predict whether a shooting victim was murdered based on the victim's race, sex, age group, and the borough the crime occurred in:

```
train <- shooting_cleaned[1:20000, ]
test <- shooting_cleaned[20001:23566, ]

rf_model <- randomForest(STATISTICAL_MURDER_FLAG ~ VIC_RACE + VIC_SEX + VIC_AGE_GROUP + BORO, data = tra
#rf_model

test$predicted <- predict(rf_model, test)
confusionMatrix(test$STATISTICAL_MURDER_FLAG, test$predicted)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2869    2
##      TRUE    694    1
##
##                Accuracy : 0.8048
##                  95% CI : (0.7914, 0.8177)
##     No Information Rate : 0.9992
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0012
```

```
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.805220
##             Specificity : 0.333333
##          Pos Pred Value : 0.999303
##          Neg Pred Value : 0.001439
##              Prevalence : 0.999159
##          Detection Rate : 0.804543
##    Detection Prevalence : 0.805104
##       Balanced Accuracy : 0.569277
##
##        'Positive' Class : FALSE
##
```

The model appears to have a decent accuracy rate of correctly predicting the outcome about 80% of the time. That being said, less than 20% of the shootings victims die, so by simply guessing that the victim lives every time, the model would technically have a higher accuracy, although the model would never correctly predict a victim dying even once.

Conclusion

Both of the visualizations seem to indicate that Brooklyn is the most dangerous borough for gun violence by far, and Staten Island is the least dangerous borough for gun violence by far. However, this only takes into account the raw number of reported cases and does not consider population density, so violence per capita data could yield different results. There could be bias present in the way the data was reported and recorded. For example, the term 'shooting victim' could refer to a person who has a gun pulled on them in one precinct, in another precinct an actual shot had to have been fired for it to count as a victim, and in yet another the person might have had to been actually hit by the bullet for it to be recorded as a shooting victim. Another source of bias is that not all shootings will be reported. One could reasonably assume that a higher percentage of dead shootings victims are reported than victims of non-lethal shootings, for the simple reason that dead people cannot walk away from the crime scene and remain silent about what occurred. If the fatality rate of the shootings was examined, the biased reported data would likely overestimate the true population parameter of the shooting fatality rate. In terms of personal bias, I would say there is very little because the data set was chosen for me so I have no personal connection to it and the conclusions I drew from the data visualizations were overwhelmingly apparent and entirely unambiguous. That being said, the manner in which I chose to tidy and clean the data had personal bias, because I chose to exclude certain features due to the amount of missing data when I could have kept them and excluded observations that were missing data instead. This caused my analysis to focus more on the victims of the shootings because the features about the perpetrators of the shootings were largely removed.

```r
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 8.1 x64 (build 9600)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
```

```
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] randomForest_4.6-14 caret_6.0-86        lattice_0.20-38
##  [4] lubridate_1.7.10    forcats_0.5.1       stringr_1.4.0
##  [7] dplyr_1.0.5         purrr_0.3.4         readr_1.4.0
## [10] tidyr_1.1.3         tibble_3.1.1        ggplot2_3.3.3
## [13] tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2           jsonlite_1.7.2       splines_3.6.1
##  [4] foreach_1.5.1        prodlim_2019.11.13   modelr_0.1.8
##  [7] assertthat_0.2.1     highr_0.9            stats4_3.6.1
## [10] cellranger_1.1.0     yaml_2.2.1           ipred_0.9-11
## [13] pillar_1.6.0         backports_1.2.1      glue_1.4.2
## [16] pROC_1.17.0.1        digest_0.6.27        rvest_1.0.0
## [19] colorspace_2.0-0     recipes_0.1.16       htmltools_0.5.1.1
## [22] Matrix_1.2-17        plyr_1.8.6           timeDate_3043.102
## [25] pkgconfig_2.0.3      broom_0.7.6          haven_2.4.0
## [28] scales_1.1.1         gower_0.2.2          lava_1.6.9
## [31] proxy_0.4-25         farver_2.1.0         generics_0.1.0
## [34] ellipsis_0.3.1       withr_2.4.2          nnet_7.3-12
## [37] cli_2.4.0            survival_3.2-10      magrittr_2.0.1
## [40] crayon_1.4.1         readxl_1.3.1         evaluate_0.14
## [43] fs_1.5.0             fansi_0.4.2          nlme_3.1-140
## [46] MASS_7.3-51.4        xml2_1.3.2           class_7.3-15
## [49] tools_3.6.1          data.table_1.14.0    hms_1.0.0
## [52] lifecycle_1.0.0      munsell_0.5.0        reprex_2.0.0
## [55] e1071_1.7-6          compiler_3.6.1       rlang_0.4.10
## [58] grid_3.6.1           iterators_1.0.13     rstudioapi_0.13
## [61] labeling_0.4.2       rmarkdown_2.7        gtable_0.3.0
## [64] ModelMetrics_1.2.2.2 codetools_0.2-16     curl_4.3
## [67] DBI_1.1.1            reshape2_1.4.4       R6_2.5.0
## [70] knitr_1.33           utf8_1.2.1           stringi_1.5.3
## [73] Rcpp_1.0.6           vctrs_0.3.7          rpart_4.1-15
## [76] dbplyr_2.1.1         tidyselect_1.1.0     xfun_0.22
```