Crime Predictor Program

Brian Searles

12/8/2019

Introduction

The purpose of this program is to analyze the crime rates in Boston and, by using that data, predict the nature of a future crime. To do this, a dataset collected from https://www.kaggle.com/AnalyzeBoston/crimes-in-boston was used. The dataset contained records of emergency calls made in the Boston area. There are 17 columns in the dataset, each describing a property of the crime. They can be seen here:

```
##  [1] "INCIDENT_NUMBER"     "OFFENSE_CODE"        "OFFENSE_CODE_GROUP" 
##  [4] "OFFENSE_DESCRIPTION" "DISTRICT"            "REPORTING_AREA"     
##  [7] "SHOOTING"            "OCCURRED_ON_DATE"    "YEAR"               
## [10] "MONTH"               "DAY_OF_WEEK"         "HOUR"               
## [13] "UCR_PART"            "STREET"              "Lat"                
## [16] "Long"                "Location"
```

With dimensions:

```
## [1] 319073     17
```

From which the data takes the form:

```
head(crimedata)
```

```
##   INCIDENT_NUMBER OFFENSE_CODE              OFFENSE_CODE_GROUP
## 1      I182070945          619                         Larceny
## 2      I182070943         1402                       Vandalism
## 3      I182070941         3410                           Towed
## 4      I182070940         3114            Investigate Property
## 5      I182070938         3114            Investigate Property
## 6      I182070936         3820 Motor Vehicle Accident Response
##                           OFFENSE_DESCRIPTION DISTRICT REPORTING_AREA
## 1                          LARCENY ALL OTHERS      D14            808
## 2                                   VANDALISM      C11            347
## 3                         TOWED MOTOR VEHICLE       D4            151
## 4                        INVESTIGATE PROPERTY       D4            272
## 5                        INVESTIGATE PROPERTY       B3            421
## 6 M/V ACCIDENT INVOLVING PEDESTRIAN - INJURY      C11            398
##   SHOOTING    OCCURRED_ON_DATE YEAR MONTH DAY_OF_WEEK HOUR   UCR_PART
## 1          2018-09-02 13:00:00 2018     9      Sunday   13   Part One
## 2          2018-08-21 00:00:00 2018     8     Tuesday    0   Part Two
## 3          2018-09-03 19:27:00 2018     9      Monday   19 Part Three
## 4          2018-09-03 21:16:00 2018     9      Monday   21 Part Three
## 5          2018-09-03 21:05:00 2018     9      Monday   21 Part Three
## 6          2018-09-03 21:09:00 2018     9      Monday   21 Part Three
##        STREET      Lat      Long                    Location
## 1  LINCOLN ST 42.35779 -71.13937 (42.35779134, -71.13937053)
## 2    HECLA ST 42.30682 -71.06030 (42.30682138, -71.06030035)
## 3 CAZENOVE ST 42.34659 -71.07243 (42.34658879, -71.07242943)
## 4  NEWCOMB ST 42.33418 -71.07866 (42.33418175, -71.07866441)
## 5    DELHI ST 42.27537 -71.09036 (42.27536542, -71.09036101)
## 6  TALBOT AVE 42.29020 -71.07159 (42.29019621, -71.07159012)
```

I believe that this information is sufficient to be able to predict the type of crime, given by "CRIME_CODE_GROUP". The program will behave as the following: by filtering each column of the dataset to a specific value, i.e. setting district to B2 and Hour to 21, and then finding the most likely crime that occurs on that filtered entry. The program will predict the most commonly occuring crime, outputted as "CRIME_CODE_GROUP" for the specified filter, and place the prediction in a list. This prediction list will then be compared to the real crime data to judge the accuracy of the program.

Analysis

First, let us analyze the imported dataset.

```
head(crimedata)
```

```
##   INCIDENT_NUMBER OFFENSE_CODE              OFFENSE_CODE_GROUP
## 1      I182070945          619                         Larceny
## 2      I182070943         1402                       Vandalism
## 3      I182070941         3410                           Towed
## 4      I182070940         3114            Investigate Property
## 5      I182070938         3114            Investigate Property
## 6      I182070936         3820 Motor Vehicle Accident Response
##                            OFFENSE_DESCRIPTION DISTRICT REPORTING_AREA
## 1                         LARCENY ALL OTHERS      D14            808
## 2                                  VANDALISM      C11            347
## 3                        TOWED MOTOR VEHICLE       D4            151
## 4                       INVESTIGATE PROPERTY       D4            272
## 5                       INVESTIGATE PROPERTY       B3            421
## 6 M/V ACCIDENT INVOLVING PEDESTRIAN - INJURY      C11            398
##   SHOOTING    OCCURRED_ON_DATE YEAR MONTH DAY_OF_WEEK HOUR   UCR_PART
## 1          2018-09-02 13:00:00 2018     9      Sunday   13   Part One
## 2          2018-08-21 00:00:00 2018     8     Tuesday    0   Part Two
## 3          2018-09-03 19:27:00 2018     9      Monday   19 Part Three
## 4          2018-09-03 21:16:00 2018     9      Monday   21 Part Three
## 5          2018-09-03 21:05:00 2018     9      Monday   21 Part Three
## 6          2018-09-03 21:09:00 2018     9      Monday   21 Part Three
##        STREET      Lat      Long                    Location
## 1  LINCOLN ST 42.35779 -71.13937 (42.35779134, -71.13937053)
## 2    HECLA ST 42.30682 -71.06030 (42.30682138, -71.06030035)
## 3 CAZENOVE ST 42.34659 -71.07243 (42.34658879, -71.07242943)
## 4  NEWCOMB ST 42.33418 -71.07866 (42.33418175, -71.07866441)
## 5    DELHI ST 42.27537 -71.09036 (42.27536542, -71.09036101)
## 6  TALBOT AVE 42.29020 -71.07159 (42.29019621, -71.07159012)
```

There are multiple unneeded columns included in this dataset, so we need to remove them. These columns include incident report number (each incident will have a unique number, so it will have zero prediciton value), and lat/long location (location data is already stored in the DISTRICT column, so this is unneeded). This leaves us with:

```
## [1] "OFFENSE_CODE"       "DISTRICT"           "MONTH"
## [4] "DAY_OF_WEEK"        "HOUR"               "OFFENSE_CODE_GROUP"
```

These will be the columns that we use to predict the crime. DISTRICT is used to analyze trends in location, while MONTH, DAY_OF_WEEK, and HOUR observe trends in time. Next, after studying the data, notice that there are some rows that contain empty entries for specific columns, an empty entry for DISTRICT for example. These rows must be removed, as to not influence the learning.

```r
selecteddata <- selecteddata[! selecteddata$DISTRICT %in% "",]
selecteddata <- selecteddata[! selecteddata$OFFENSE_CODE %in% "",]
selecteddata <- selecteddata[! selecteddata$MONTH %in% "",]
selecteddata <- selecteddata[! selecteddata$DAY_OF_WEEK %in% "",]
selecteddata <- selecteddata[! selecteddata$HOUR %in% "",]
selecteddata <- selecteddata[! selecteddata$OFFENSE_CODE_GROUP %in% "",]
cleandata <- selecteddata
```

This leaves us with a dataset called cleandata which has been sufficiently cleaned for our needs. Now, all we must do is construct a training and test set.

```r
set.seed(1, sample.kind="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

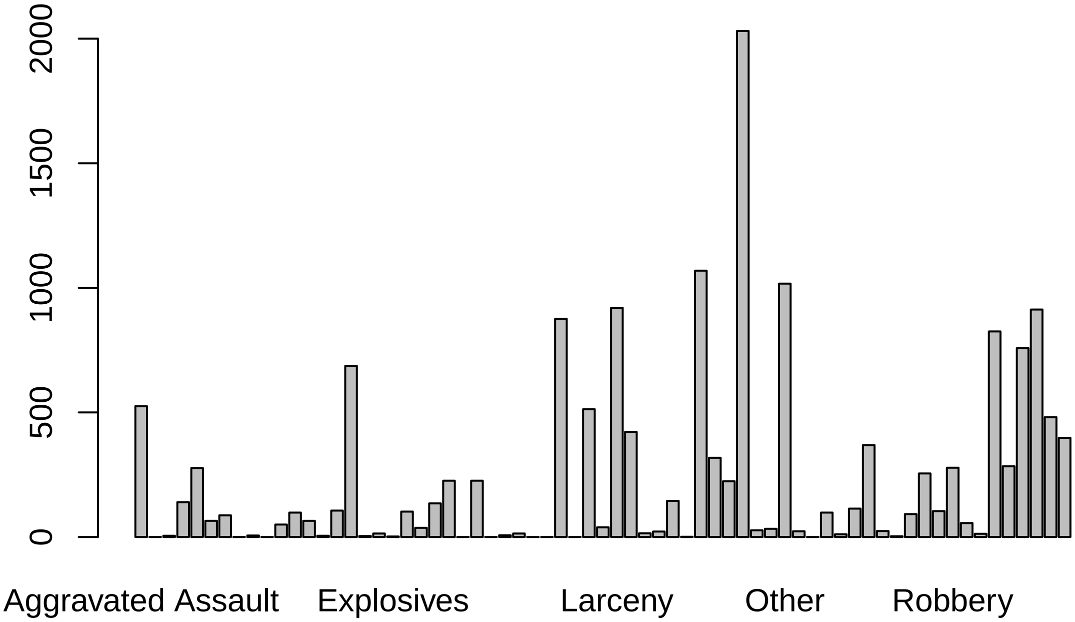
```r
index <- sample(nrow(cleandata), 100000)
x <-  cleandata[index,]

index1 <- sample(nrow(cleandata), 1000)
x_test <-  cleandata[index1,]
```

So x will be our training set and x_test will be our testing set. Note, the seed is set to 1 if you would like to attempt to replicate my results.

Next, let us begin to analyze out data.

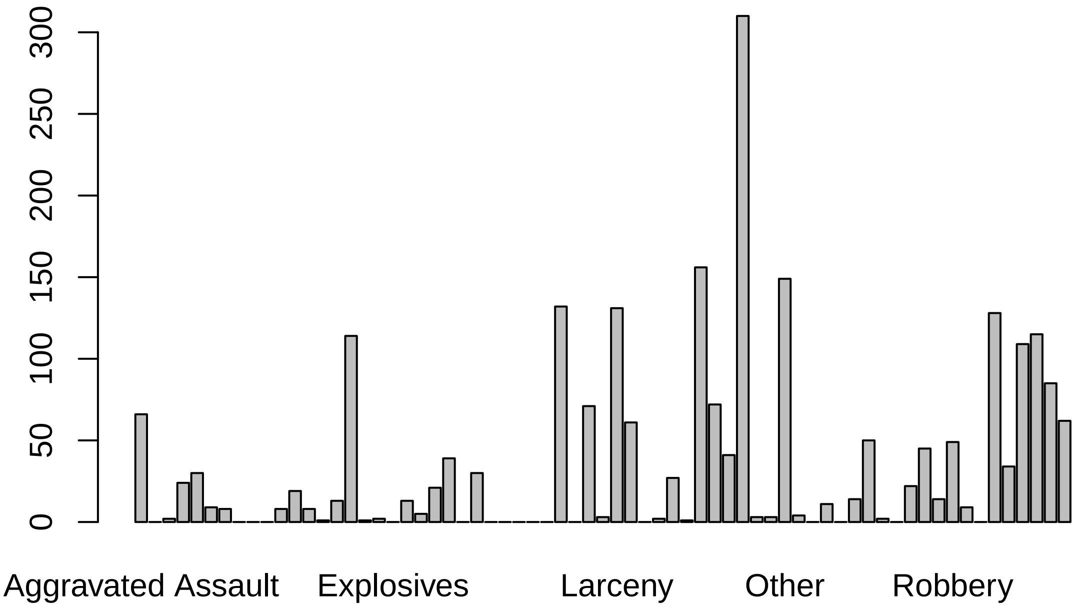
Let us first analyze how the crime rate differs between districts. For district B2, we have



Each bar represents a different crime, and the height of the bar represents the number of occurences of that crime. This gives us a most common crime of

```
## [1] "Motor Vehicle Accident Response"
```

Next, lets see how filtering for a specific Day effects the crime rate. Let's filter for Friday in district B2



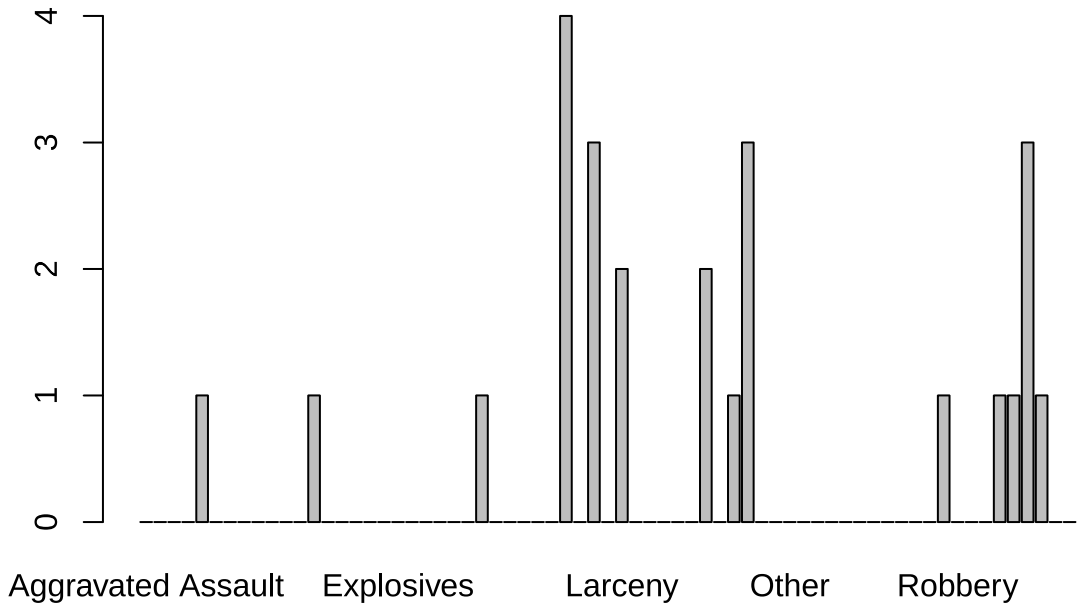
We can see that when we compare this graph to the previous graph, the distribution has changed. Our most common crime for Friday in B2 is:

```
## [1] "Motor Vehicle Accident Response"
```

Which is the same as the previous distribution. This is what we expect as this distribution is a subset of the prevoius one.

Eventually we will filter our dataset to as specific as we can possibly get. An example of this is as shown:

```
B3M9DFH22 <- cleandata %>% filter(DISTRICT %in% "B3") %>% filter(MONTH %in% "9") %>% filter(DAY_OF_WEEK
plot(B3M9DFH22$OFFENSE_CODE_GROUP)
```



```
B3M9DFH22Table <- table(B3M9DFH22$OFFENSE_CODE_GROUP)
B3M9DFH22Guess  <- names(which(max(B3M9DFH22Table)==B3M9DFH22Table))
B3M9DFH22Guess
```

```
## [1] "Investigate Person"
```

This is for District B3 on Month 9, on Friday, at Hour 22.

So our model will work by filtering down to these specific catagories and then using the most common crime as its prediction. To do this, we will build a function that takes our dataset and builds a list of predicted crimes, positioned in the same order as they appear in the dataset.

The function takes the form:

```
GuessList <- function(CrimeList){
  tempList <- list()
  for(i in 1:nrow(CrimeList)){
    tempDI <- CrimeList[i,2] #District of the indexed row
    tempM <- CrimeList[i,3] #Month of the indexed row
    tempDAY <- CrimeList[i,4] #Day of the indexed row
    tempH <- CrimeList[i,5] #Hour of the indexed row
    tempG <- x %>% filter(DISTRICT %in% tempDI) %>% filter(MONTH %in% tempM) %>% filter(DAY_OF_WEEK %in
    tempTABLE  <- table(tempG$OFFENSE_CODE_GROUP)
    tempGuess  <- names(which(max(tempTABLE)==tempTABLE))
    tempList[i] <- tempGuess[1] #Adding the most common crime to a List. If there are more than 1 most
  }
  tempList
}
```

Now all that is left is to create a few prediction lists and test their accuracy.

Results

In order to test the accuracy of our prediction lists, we will create an accuracy test function. This function will preform a boolean check to see if the crime from the predicted list is the same as from the actual list, then it will add up the results and take the mean of boolean values. This mean will be the accuracy of the prediction list compared to the actual list. Accuracy test function:

```
accuracy_test <- function(test,true){
  p <- 0
  for(i in 1:length(test)){
    j <- ifelse(test[i] == true[i,6],1,0)
    p <- p+j
  }
  l = p/length(test)
  print(l)
}
```

using our training list first, we have:

```
Training <- GuessList(x)
accuracy_test(Training, x)
```

```
## [1] 0.32261
```

This provides an accuracy of 0.323. This is not a very impressive accuracy, so lets see what happens when we increase the sample size to the entire dataset.

```
Total <- GuessList(cleandata)
accuracy_test(Total, cleandata)
```

```
## [1] 0.1548338
```

This again is not a very impressive accuracy.

One change that may increase the accuracy of the program is changing the GuessList function. The training set x is used to build the list, but it can be replaced with a larger dataset, providing a more accurate guess.

Conclusion

In conclusion, this program was written with the purpose of predicting the type of crime committed in Boston given data of the previous crimes location and time of occurance. Analysis of the dataset seemed to imply certain trends within the location and time data that lead to certain crimes being commited. However, when accuaracy of our predictions were measured, we found them to be unsatisfactory. Perhaps if the dataset contained more usefull information, or was larger, then the accuracy would improve.