Examining the Shooting Events in New York City

By Daniel Lewis

Student at Syracuse University

Contents

1 Introduction

As a once long-term resident of the great state of New York. The author aims to learn more about its most populous city through this project. The City of New York, New York (NYC) has generated an Open Data imitative in order to be fair and open with its constituents. As the author may be taking jobs in the city as early as 2020, there must be a way to determine the dangerous areas of the city from the crime data. In order to do this the student has referenced the New York Police Department (NYPD) public facing records for shooting crime incidents for the period of 2013 through 2018. Throughout this project chunks of **R** code will be displayed inline as a reference to better understand the process.

---

2 Business Questions

This project will attempt to answer the following three questions:

- Does time of day and location have a direct correlation to a shooting event?
- Has any part of the city been consistently high in shooting incidents?
- Is any one particular demographic group at high risk ?

As the purpose of this project is to determine the dangerous parts of NYC. Understanding the time of day that most shootings occur as well as which areas of the city these incidents occur in will be key to a new resident choosing their domicile. Additionally, as a bi-product of the information we can generate information on at risk demographics. While project will not predict the housing market prices or the cost of living, it will give a reader a reasonable glimpse at the public order of the city.

---

3 Data Acquisition, Cleansing Transformation, Munging

3.1 Problem Definition

The purpose of this project is to generate actionable data for a potential future resident of NYC. Actionable data is defined by this project as tables, charts, and modeling to provide graphical information for the reader to understand where the shooting hotspots in the last five years of shooting incidents in NYC. The data from

New York Police Department is assigned several flags that can better help define the demographics of the project. There are some key attributes within this dataset that can be used to generate actionable data.

Exmaple Attributes:

- Ages of Accused & Victims
- Sex of Accused & Victims
- Location of the Crime
- Time and Date of the Crime

3.2 Data Acquisition

The **NYPD Shooting Incident Data**, which is available freely from the NYC Open Data website contains one set of data. This project will utalize the powerful data science oriented programing language of **R** to analyze this data. Upon intial review of the information provided to the author via the NYPD. Reading the source documentation for this file it was determined that the best method to import would be a Comma Seperated Values (CSV). Addtionally the document retrieved should contain a total of 6,407 rows with 18 attributes for a total of 115,326 data points. In order to start processing the data it was first imported into the system and stored in the variable rawCSV.

```
urlToImport <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
rawCSV <- data.frame((read.csv(urlToImport, stringsAsFactors = FALSE)))
```

Generating the vector of rawCSV, we created a data frame with the contents of the download csv button from our selected source. Wtih this data now stored within our project, a data scieentist must confirm that the information has arrived correct and that it is able to be used within the project.

3.3 Data Clensing Process

Although the data retrived is in relativly good shape there are some peculiarities that need to be addressed during the porocess in order to properly utalize this data within the project. The first piece that could be easily recognized just in a look at the data is that alot of the information is in capital letters. Due to several proceses within this project requiring the lowercase data, a to lower operation was required to be conducted on a few fields.

```r
cleanCSV <- rawCSV

colnames(cleanCSV) <- tolower(colnames(cleanCSV))

cleanCSV$boro<-tolower(cleanCSV$boro)

cleanCSV$location_desc<-tolower(cleanCSV$location_desc)

cleanCSV$perp_sex<-tolower(cleanCSV$perp_sex)

cleanCSV$perp_race<-tolower(cleanCSV$perp_race)

cleanCSV$vic_sex<-tolower(cleanCSV$vic_sex)

cleanCSV$vic_race<-tolower(cleanCSV$vic_race)
```

The next issue with the intial importing of the CSV file is that the file has not properly converted all the fields to their proper data types. Upon pulling the data frames structure we can see that some fields that should be numbers are characters, as well as fields that should be dates or times are also treated as a character field.

```r
str(cleanCSV)
```

In order to correct these issues, the first column to be targeted was that of the occur_date. From the above structure you can see that this is shown as both a factor and additionally it contained both an unformated date as well as some erronious informationn inserted somewhere along the way during the csv file creation process.

```r
removeTime <- function(inputVector){

  inputVector <-gsub(" 12:00:00 AM","",inputVector)

  inputVector <- gsub(" ", "", inputVector)

  inputVector <- as.Date(inputVector, format='%m/%d/%Y')

  return(inputVector)

}

cleanCSV$occur_date<-removeTime(cleanCSV$occur_date)
```

The column titled location_desc is filled with limited information for the purposes of this report. When it is filled with data the information is not consistant or actionable. Additionally, there is a column to indicate what type of officer was involved in this incident. These columns were removed from the data frame and than verified that it was indeed removed.

```r
cleanCSV<-cleanCSV[,-6:-7]

colnames(cleanCSV)
```

Finally, after all of this cleaning has been completed only complete rows should remain. This will be rows that do not have any blank values that cannot be used within this report.

```
cleanCSV<-cleanCSV[complete.cases(cleanCSV),]
```

After performing all of these operations to prepare the data for use within the report the data contains 6407 rows and 17 attributes to perform work on for a total of 108,919 data points.

3.4 Data Dictionary

The following tables displays the list of data items used within this report.

Variable Name	Variable Type	Variable Description
incident_key	int	Incidient number, duplicates are victims
occur_date	date	Indicates the date of occurance
occur_time	chr	Indicates the time of the incident
boro	chr	Indicates which borrough the even occured in
precinct	int	Indicates which police department responded
statistical_murder_flag	chr	Indicates if victim died
perp_age_group	chr	If known, age range of perpetrator
perp_sex	chr	If known, sex of perpetrator
perp_race	chr	If known race of perpetrator
vic_age_group	chr	Victims Age Range
vic_sex	chr	Victims Sex
vic_race	chr	Victims Race
x_coord_cd	int	X grid coordinate for event
y_coord_cd	int	Y grid coordinate for event
latitude	num	Lattitude of event
longitude	num	Longitude of event

4 Descriptive Statistics

4.1 Summary Statistics

Prior to doing any work with the data, the author wanted to understand the data that they were using. Using the summary command they were able to get the minimum and max values, as well as the mean and median values. As the incident key is an auto generated number for each incident it is not included in these tables.

occur_date	occur_time	boro	precinct	statistical_murder_flag
Min. :2013-01-01	Length:6407	Length:6407	Min. : 1.00	Length:6407
1st Qu.:2014-03-30	Class :character	Class :character	1st Qu.: 44.00	Class :character
Median :2015-05-05	Mode :character	Mode :character	Median : 69.00	Mode :character
Mean :2015-05-13	NA	NA	Mean : 66.69	NA
3rd Qu.:2016-07-13	NA	NA	3rd Qu.: 81.00	NA
Max. :2017-12-31	NA	NA	Max. :123.00	NA

perp_age_group	perp_sex	perp_race	vic_age_group	vic_sex
Length:6407	Length:6407	Length:6407	Length:6407	Length:6407
Class :character	Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character	Mode :character

vic_race	x_coord_cd	y_coord_cd	latitude	longitude
Length:6407	Min. : 922884	Min. :132099	Min. :40.53	Min. :-74.22
Class :character	1st Qu.: 999970	1st Qu.:181481	1st Qu.:40.66	1st Qu.:-73.94
Mode :character	Median :1007605	Median :193359	Median :40.70	Median :-73.92
NA	Mean :1008757	Mean :206795	Mean :40.73	Mean :-73.91
NA	3rd Qu.:1016320	3rd Qu.:239453	3rd Qu.:40.82	3rd Qu.:-73.88
NA	Max. :1063056	Max. :269205	Max. :40.91	Max. :-73.72

4.2 Data Structure

In order to process and manipulate the data within the project, we must get a firm grasp for what each fields data type is in **R**. This is reflected by using the

```
## 'data.frame':    6407 obs. of  16 variables:
##  $ incident_key           : int  138817042 156642467 89216118 138898674 90370145 157022340 166728123
##  $ occur_date             : Date, format: "2014-09-21" "2016-09-12" ...
##  $ occur_time             : chr  "23:15:00" "17:30:00" "03:30:00" "17:48:00" ...
##  $ boro                   : chr  "brooklyn" "queens" "queens" "brooklyn" ...
##  $ precinct               : int  67 105 102 73 71 45 47 13 43 7 ...
##  $ statistical_murder_flag: chr  "false" "false" "false" "false" ...
##  $ perp_age_group         : chr  "" "" "" "" ...
##  $ perp_sex               : chr  "" "" "" "" ...
##  $ perp_race              : chr  "" "" "" "" ...
##  $ vic_age_group          : chr  "25-44" "25-44" "45-64" "<18" ...
##  $ vic_sex                : chr  "m" "m" "m" "m" ...
##  $ vic_race               : chr  "black" "black" "asian / pacific islander" "black" ...
##  $ x_coord_cd             : int  996949 1051945 1022214 1007962 1003564 1032140 1025683 986464 102558
##  $ y_coord_cd             : int  176623 180331 188265 183992 181625 242004 261870 208227 236918 20105
##  $ latitude               : num  40.7 40.7 40.7 40.7 40.7 ...
##  $ longitude              : num  -74 -73.8 -73.9 -73.9 -73.9 ...
```

4.3 Graphs, Charts, Tables

The first part of generating charts and graphs is to generate the data into easier to read tables for the purpose of genarting graphs to be read.

```
deaths <- sqldf("SELECT *
                FROM cleanCSV
                WHERE statistical_murder_flag IN ('true')")
injuries <- sqldf("SELECT *
                FROM cleanCSV
                  WHERE statistical_murder_flag NOT IN ('true')")


incidentsByLocation <- sqldf("SELECT DISTINCT incident_key, x_coord_cd,
                            y_coord_cd, latitude, longitude
                             FROM cleanCSV")
```

---

5 Modeling Techniques

---

6 Data Interpretation

---

7 Summary

Actionable ideas or insights

---

8 Appendix

A.1 R Code

```r
#### Generate a Function to Ensure a Package is Installed ###
EnsurePackage<-function(x){
  x<-as.character(x)
  if (!require(x,character.only=TRUE)){
    install.packages(pkgs=x, repos="http://cran.r-project.org")
    require(x, character.only=TRUE)
  }
}
removeTime <- function(inputVector){
  inputVector <-gsub(" 12:00:00 AM","",inputVector)
  inputVector <- gsub(" ", "", inputVector)
  inputVector <- as.Date(inputVector, format='%m/%d/%Y')
  return(inputVector)
}

###########Import All Required Packages######################
EnsurePackage("compare")
EnsurePackage("ggplot2")
EnsurePackage("ggmap")
EnsurePackage("gridExtra")
EnsurePackage("rgeos")
EnsurePackage("maptools")
EnsurePackage("RJSONIO")
EnsurePackage("sqldf")

#############Import Source Data for Project##################
urlToImport <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
rawCSV <- data.frame((read.csv(urlToImport, stringsAsFactors = FALSE)))

########################Pre-Processing Steps#################
cleanCSV <- rawCSV
colnames(cleanCSV) <- tolower(colnames(cleanCSV))
cleanCSV$boro<-tolower(cleanCSV$boro)
cleanCSV$location_desc<-tolower(cleanCSV$location_desc)
cleanCSV$perp_sex<-tolower(cleanCSV$perp_sex)
cleanCSV$perp_race<-tolower(cleanCSV$perp_race)
cleanCSV$vic_sex<-tolower(cleanCSV$vic_sex)
cleanCSV$vic_race<-tolower(cleanCSV$vic_race)


cleanCSV$occur_date<-removeTime(cleanCSV$occur_date)

cleanCSV<-cleanCSV[,-6:-7]
colnames(cleanCSV)
```

```r
cleanCSV<-cleanCSV[complete.cases(cleanCSV),]

####################Creating Descriptive Statistics###########

knitr::kable(summary(cleanCSV[2:6]), "latex", booktabs=T)
knitr::kable(summary(cleanCSV[7:11]), "latex", booktabs=T)
knitr::kable(summary(cleanCSV[12:16]), "latex", booktabs=T)

str(cleanCSV)

#################Generating Graph Data########################

deaths <- sqldf("SELECT *
                FROM cleanCSV
                WHERE statistical_murder_flag IN ('true')")
injuries <- sqldf("SELECT *
                FROM cleanCSV
                  WHERE statistical_murder_flag NOT IN ('true')")

incidentsByLocation <- sqldf("SELECT DISTINCT incident_key, x_coord_cd, y_coord_cd, latitude, longitude
                             FROM cleanCSV")


########Create the Data Dictionary############################
varName<-colnames(cleanCSV)
varType<-c("int",
           "date",
           "chr",
           "chr",
           "int",
           "chr",
           "chr",
           "chr",
           "chr",
           "chr",
           "chr",
           "chr",
           "int",
           "int",
           "num",
           "num")
varDesc<-c("Incidient number, duplicates are victims",
           "Indicates the date of occurance",
           "Indicates the time of the incident",
           "Indicates which borrough the even occured in",
           "Indicates which police department responded",
           "Indicates if victim died",
           "If known, age range of perpetrator",
           "If known, sex of perpetrator",
           "If known race of perpetrator ",
           "Victims Age Range","Victims Sex",
           "Victims Race",
           "X grid coordinate for event",
           "Y grid coordinate for event",
           "Lattitude of event",
           "Longitude of event")
dataDict<-data.frame(varName,varType,varDesc)
colnames(dataDict)<-c("Variable Name", "Variable Type", "Variable Description")

knitr::kable(dataDict, "latex", booktabs=T)

########Create the map to plot on#############################

hist_events <- ggplot(data=cleanCSV, aes(x=occur_date))+geom_histogram(aes(color='orange'))
```

```
hist_events


line_deaths <- ggplot(data=cleanCSV, aes(x=occur_date, y=statistical_murder_flag))+geom_line()

line_deaths
```

A.2 Notes