Final Project IS607

Marco Siqueira Campos, KaMan Chan, Sharon Morris, Talha Muhammad

12/18/2016

Introduction

This is the final project for IS607 Data Acquisition and Management class in the M.S. of Data Analytics program at The City University of New York.

This is an observational study.

Motivation

The motivation of this project is to understand the relationship between crime and property values in New York City. The majority of the project team are New York City residents and this is of interest.

The results of this study could be useful to those interested in purchasing real estate in New York City.

The Data

The data were obtained from two sources. Crime data were collected from New York City Police Department complaint data. The crime dataset includes all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD) during 2015. These data represent criminal offenses according to New York State Penal Law definitions, not FBI Uniform Crime Report definitions, and are therefore not comparable to UCR reported crime.

Each row represents a complaint reported to NYPD. Only valid complaints are included in this release. Complaints deemed unfounded due to reporter error or misinformation are excluded from the data set, as they are not reflected in official figures nor are they considered to have actually occurred in a criminal context. Similarly, complaints that were voided due to internal error are also excluded from the data set.

Property sales data from August to November 2016 were scraped from the real estate listing site Trulia.

Obtain the data

```
library(dplyr)
library(plyr)
library(corrplot)
library(ggmap)
library(RMySQL)
```

```
temp <- tempfile()
download.file("https://raw.githubusercontent.com/cunyauthor/FinalProject/master/NYPD_Complaint_Data_His
NYPD <- read.csv(unz(temp, "NYPD_Complaint_Data_Historic.csv"), encoding="UTF-8", na.strings=c("","NA")
unlink(temp)

str(NYPD)
head(NYPD)
```

Scrub and explore the data

A subset of the crime data was created with only variables required for the analysis. All missing data were removed from the dataset, the resulting dataset contained 4500,000+ cases.

A random sample was taken to create a sample of 11,700 cases. The sample was broken into 5 smaller samples in order to add Google API to match longitude and latitude in the data to street addresses. Google API allowed 2,500 free downloads per 24 hour period.

We had a challenge here as there is no single standard defining of violent crime, there are two standards, NCVS Bureau of Justice Statistics National Crime Victimization Survey and the UCR Federal Bureau of Investigation's Uniform Crime Report and they were not fully compatible with the description of the NYC PD.

To solve this we create our own definition of violent crime, we filter from the main the follow crime categories: DANGEROUS WEAPONS,FELONY ASSAULT,KIDNAPPING & RELATED OFFENSES,MURDER & NON-NEGL. MANSLAUGHTER,ROBBERY and SEX CRIMES.

```
class(NYPD) #view class
dim(NYPD)   #view dimensions
names(NYPD) #explore the column names
unstand the structure
str(NYPD)

new <- select(NYPD, RPT_DT, OFNS_DESC, OFNS_DESC, LAW_CAT_CD, BORO_NM, ADDR_PCT_CD,
              Latitude, Longitude, Lat_Lon) # data with required variables

sapply(new, function(x) sum(is.na(x))) # review NAs in the data
noNas <- na.omit(new) # remove NAs from data
sapply(noNas, function(x) sum(is.na(x)))
glimpse(noNas) #look at the data

summary(noNas) #summarize the data
```

Create datasets with Google API addresses, crime index

```
#set.seed(123) # set the seed to standartize
#nyc<-noNas[sample(nrow(noNas),458557),] # random the sample to standardize the sequence
#nyc$id<-1:458557 # create a id to check

# Create datasets with Google API
#nyc_1<-nyc[1:2400,]#
#res <- mapply(FUN = function(lon, lat) {
        #revgeocode(c(lon, lat), output = "more")
#},
#nyc_1$Longitude, nyc_1$Latitude
#)
#res1<-rbind_all(lapply(res, as.data.frame))
#View(res1)
#nyc_1<-cbind(nyc_1,res1) # add full address to data base
#write.csv(nyc_51 file = 'nyc_1.csv', row.names = FALSE) #write to csv
#head(nyc_1)

#nyc_2<-nyc[2401:4800,]
#res <- mapply(FUN = function(lon, lat) {
        #revgeocode(c(lon, lat), output = "more")
```

```
#},
#nyc_2$Longitude, nyc_2$Latitude
#)
#res1<-rbind_all(lapply(res, as.data.frame))
#View(res1)
#nyc_2<-cbind(nyc_2,res1) # add full address to data base
#write.csv(nyc_2, file = 'nyc_2.csv', row.names = FALSE) #write to csv
#head(nyc_2)

#nyc_3<-nyc[4801:6900,]# 2nd Dec Sharon sample # (for today)
#res <- mapply(FUN = function(lon, lat) {
        #revgeocode(c(lon, lat), output = "more")
#},
#nyc_3$Longitude, nyc_3$Latitude
#)
#res1<-rbind_all(lapply(res, as.data.frame))
#View(res1)
#nyc_3<-cbind(nyc_3,res1) # add full address to data base
#write.csv(nyc_3, file = 'nyc_3.csv', row.names = FALSE) #write to csv
#head(nyc_3)

#nyc_4<-nyc[7201:9600,]#
#res <- mapply(FUN = function(lon, lat) {
        #revgeocode(c(lon, lat), output = "more")
#},
#nyc_4$Longitude, nyc_4$Latitude
#)
#res1<-rbind_all(lapply(res, as.data.frame))
#View(res1)
#nyc_4<-cbind(nyc_4,res1) # add full address to data base
#write.csv(nyc_4, file = 'nyc_4.csv', row.names = FALSE) #write to csv
#head(nyc_4)

#nyc_5<-nyc[9600:12000,]#
#res <- mapply(FUN = function(lon, lat) {
        #revgeocode(c(lon, lat), output = "more")
#},
#nyc_5$Longitude, nyc_5$Latitude
#)
#res1<-rbind_all(lapply(res, as.data.frame))
#View(res1)
#nyc_5<-cbind(nyc_5,res1) # add full address to data base
#write.csv(nyc_5, file = 'nyc_5.csv', row.names = FALSE) #write to csv
#head(nyc_5)

# read the nyc data
urlfile<-"https://raw.githubusercontent.com/cunyauthor/FinalProject/master/nyc_15.csv"
nyc_1<-read.csv(url(urlfile), encoding="UTF-8", na.strings=c("","NA"), stringsAsFactors = F)

nyc_1$postal_code<-as.character(nyc_1$postal_code)

# create new column for violent crime
nyc_1<-nyc_1[-10639,]
```

```r
index<-c("DANGEROUS WEAPONS","FELONY ASSAULT","KIDNAPPING & RELATED OFFENSES","MURDER & NON-NEGL. MANSLA
nyc_1$crime_v<-index[(match(nyc_1$OFNS_DESC,index))]
nyc_1$crime_v<-as.factor(nyc_1$crime_v)

# agrregate the crime data by zip code
nyc_s1<-aggregate(OFNS_DESC ~ postal_code, nyc_1, length)
nyc_s2<-aggregate(crime_v ~ postal_code, nyc_1, length)

colnames(nyc_s1) <- c("zip","freq_crime")
colnames(nyc_s2) <- c("zip","freq_violent_crime")

# joint two tables of crime data
nyc_s12<-left_join(nyc_s1, nyc_s2, "zip")
# change NA for zero
nyc_s12[is.na(nyc_s12)]<-0
# Crime table
head(nyc_s12)
```

```
##     zip freq_crime freq_violent_crime
## 1 10001        161                  8
## 2 10002        141                 10
## 3 10003        140                  4
## 4 10004         18                  0
## 5 10005          7                  1
## 6 10006          6                  0
```

```r
#crime summary
summary(nyc_s12$freq_crime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   27.00   51.00   63.21   93.00  222.00
```

```r
sd(nyc_s12$freq_crime)
```

```
## [1] 48.22886
```

```r
summary(nyc_s12$freq_violent_crime)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    1.00    5.00    6.73   10.00   39.00
```

```r
sd(nyc_s12$freq_violent_crime)
```

```
## [1] 6.985422
```

```r
# Read data from real state
urlfile<-"https://raw.githubusercontent.com/cunyauthor/FinalProject/master/trulia_sqft.csv"
trulia<-read.csv(url(urlfile), encoding="UTF-8", na.strings=c("","NA"), stringsAsFactors = F)

# Read the table
urlfile<-"https://raw.githubusercontent.com/cunyauthor/FinalProject/master/zip_neig.csv"
zip_neig<-read.csv(url(urlfile), encoding="UTF-8", na.strings=c("","NA"), stringsAsFactors = F)

#joint crime data with borough and neighborhood
 zip_neig$zip<-as.character(zip_neig$zip)
nyc_s12<-left_join(nyc_s12,zip_neig, "zip")
head(nyc_s12)
```

```
##     zip freq_crime freq_violent_crime   Borough        Neighborhood
## 1 10001        161                  8 Manhattan Chelsea and Clinton
## 2 10002        141                 10 Manhattan     Lower East Side
## 3 10003        140                  4 Manhattan     Lower East Side
## 4 10004         18                  0 Manhattan     Lower Manhattan
## 5 10005          7                  1 Manhattan     Lower Manhattan
## 6 10006          6                  0 Manhattan     Lower Manhattan
```

```
# Statistics summary of NYC prices
summary(trulia$Median_sales_price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  400000  879000 1040000 1271000 1656000 3200000
```

```
summary(trulia$Price_sqft)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     515    1100    1374    1348    1684    2074
```

```
# Trulia table price by zip
head(trulia)
```

```
##   X   zip Median_sales_price Price_sqft
## 1 1 10001            1687500       2074
## 2 2 10002             875000       1216
## 3 3 10003            2395000       1837
## 4 4 10004            1260000       1450
## 5 5 10005            1513779       1404
## 6 6 10006             749000       1276
```

```
# Change zip to as.character
trulia$zip<-as.character(trulia$zip)
# Join the two tables
nyc_full<-inner_join(trulia,nyc_s12, "zip")
head(nyc_full)
```

```
##   X   zip Median_sales_price Price_sqft freq_crime freq_violent_crime
## 1 1 10001            1687500       2074        161                  8
## 2 2 10002             875000       1216        141                 10
## 3 3 10003            2395000       1837        140                  4
## 4 4 10004            1260000       1450         18                  0
## 5 5 10005            1513779       1404          7                  1
## 6 6 10006             749000       1276          6                  0
##     Borough        Neighborhood
## 1 Manhattan Chelsea and Clinton
## 2 Manhattan     Lower East Side
## 3 Manhattan     Lower East Side
## 4 Manhattan     Lower Manhattan
## 5 Manhattan     Lower Manhattan
## 6 Manhattan     Lower Manhattan
```

```
# read population by ZIP frm 2010 US Census
urlfile<-"https://raw.githubusercontent.com/cunyauthor/FinalProject/master/pop_zip_census.csv"
zip_pop<-read.csv(url(urlfile), encoding="UTF-8", na.strings=c("","NA"), stringsAsFactors = F)

colnames(zip_pop) <- c("zip","pop")
zip_pop$zip<-as.character(zip_pop$zip)
```

```r
# Join the two tables, add population data
nyc_full<-inner_join(nyc_full,zip_pop, "zip")

# Include crime rate data per 1000.000 habit per year by zip
nyc_full$crime_rate<-(nyc_full$freq_crime/nyc_full$pop)*100000 * 458557/nrow(nyc_1)
nyc_full$v_crime_rate<-(nyc_full$freq_violent_crime/nyc_full$pop)*100000 * 458557/nrow(nyc_1)
```

```r
# Database schema
source("logincredentiasl.R")
connection <- dbConnect(MySQL(), user=MySQL_Username, password=MySQL_Password)

dbSendQuery(connection, 'CREATE SCHEMA IF NOT EXISTS SYS;')
```

```
## <MySQLResult:775108397,0,0>
```

```r
dbSendQuery(connection, 'USE SYS;')
```

```
## <MySQLResult:775108397,0,1>
```

```r
dbSendQuery(connection, 'DROP TABLE IF EXISTS nyc_1;')
```

```
## <MySQLResult:942551092,0,2>
```

```r
dbSendQuery(connection, 'Drop Table If Exists nyc_full;')
```

```
## <MySQLResult:925773876,0,3>
```

```r
dbSendQuery(connection, 'Drop Table If Exists nyc_s1;')
```

```
## <MySQLResult:942551092,0,4>
```

```r
dbSendQuery(connection, 'Drop Table If Exists nyc_s12;')
```

```
## <MySQLResult:775108397,0,5>
```

```r
dbSendQuery(connection, 'Drop Table If Exists nyc_s2;')
```

```
## <MySQLResult:925773876,0,6>
```

```r
dbSendQuery(connection, 'Drop Table If Exists trulia;')
```

```
## <MySQLResult:775108397,0,7>
```

```r
dbSendQuery(connection, 'Drop Table If Exists zip_pop;')
```

```
## <MySQLResult:942551092,0,8>
```

```r
urlfile<-"https://raw.githubusercontent.com/cunyauthor/FinalProject/master/nyc_15.csv"
nyc_1<-read.csv(url(urlfile), encoding="latin1", na.strings=c("","NA"), stringsAsFactors = F)

dbWriteTable(connection, "tbl_nyc1", nyc_1, append = TRUE, row.names = FALSE)
```

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

```r
dbSendQuery(connection, "ALTER TABLE tbl_nyc1
            MODIFY COLUMN id varchar(100) NOT NULL,
            MODIFY COLUMN RPT_DT varchar(100) NOT NULL,
            MODIFY COLUMN OFNS_DESC varchar(100) NOT NULL,
            MODIFY COLUMN BORO_NM varchar(100) NOT NULL,
            MODIFY COLUMN ADDR_PCT_CD varchar(100) NULL,
            MODIFY COLUMN Latitude varchar(100) NULL,
```

```
            MODIFY COLUMN Longitude varchar(100) NULL,
            MODIFY COLUMN Lat_Lon varchar(100) NULL,
            MODIFY COLUMN address varchar(200) NULL,
            MODIFY COLUMN street_number varchar(100) NULL,
            MODIFY COLUMN route varchar(100) NULL,
            MODIFY COLUMN neighborhood varchar(100) NULL,
            MODIFY COLUMN political varchar(100) NULL,
            MODIFY COLUMN administrative_area_level_2 varchar(100) NULL,
            MODIFY COLUMN administrative_area_level_1 varchar(100) NULL,
            MODIFY COLUMN country varchar(100) NULL,
            MODIFY COLUMN postal_code varchar(100) NULL,
            MODIFY COLUMN postal_code_suffix varchar(100) NULL,
            MODIFY COLUMN locality varchar(100) NULL,
            MODIFY COLUMN premise varchar(100) NULL,
            MODIFY COLUMN crime_v varchar(100) NULL
            ;")
```

```
## <MySQLResult:942551092,0,11>
```

```
urlfile<-"https://raw.githubusercontent.com/cunyauthor/FinalProject/master/pop_zip_census.csv"
zip_pop<-read.csv(url(urlfile), encoding="UTF-8", na.strings=c("","NA"), stringsAsFactors = F)
colnames(zip_pop) <- c("zip","pop")# rename variables
dbWriteTable(connection, "tbl_zip_pop", zip_pop, append = TRUE, row.names = FALSE)
```

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

```
dbSendQuery(connection, "ALTER TABLE tbl_zip_pop
            MODIFY COLUMN zip varchar(10) NOT NULL,
            MODIFY COLUMN pop varchar(50) NOT NULL
            ;")
```

```
## <MySQLResult:-1147955784,0,14>
```

```
urlfile<-"https://raw.githubusercontent.com/cunyauthor/FinalProject/master/zip_neig.csv"
zip_neig<-read.csv(url(urlfile), encoding="UTF-8", na.strings=c("","NA"), stringsAsFactors = F)

dbWriteTable(connection, "tbl_zip_neig", zip_neig, append = TRUE, row.names = FALSE)
```

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

```
dbSendQuery(connection, "ALTER TABLE tbl_zip_neig
            MODIFY COLUMN Borough varchar(100) NOT NULL,
            MODIFY COLUMN Neighborhood varchar(100) NOT NULL,
            MODIFY COLUMN zip varchar(10) NOT NULL
            ;")
```

```
## <MySQLResult:-1134417840,0,17>
```

```
urlfile<-"https://raw.githubusercontent.com/cunyauthor/FinalProject/master/trulia_sqft.csv"
trulia<-read.csv(url(urlfile), encoding="UTF-8", na.strings=c("","NA"), stringsAsFactors = F)

dbWriteTable(connection, "tbl_trulia", trulia, append = TRUE, row.names = FALSE)
```

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

```
dbSendQuery(connection, "ALTER TABLE tbl_trulia
            MODIFY COLUMN x varchar(10) NOT NULL,
            MODIFY COLUMN zip varchar(10) NOT NULL,
```

```
            MODIFY COLUMN Median_sales_price varchar(50) NOT NULL,
            MODIFY COLUMN Price_sqft varchar(50) NOT NULL
            ;")
```

```
## <MySQLResult:908996660,0,20>
```

```
dbSendQuery(connection, 'DROP TABLE IF EXISTS nyc_1;')
```

```
## <MySQLResult:-1108341144,0,21>
```

```
dbSendQuery(connection, 'Drop Table If Exists nyc_full;')
```

```
## <MySQLResult:925773876,0,22>
```

```
dbSendQuery(connection, 'Drop Table If Exists nyc_s1;')
```

```
## <MySQLResult:775108397,0,23>
```

```
dbSendQuery(connection, 'Drop Table If Exists nyc_s12;')
```

```
## <MySQLResult:925773876,0,24>
```

```
dbSendQuery(connection, 'Drop Table If Exists nyc_s2;')
```

```
## <MySQLResult:908996660,0,25>
```

```
dbSendQuery(connection, 'Drop Table If Exists trulia;')
```

```
## <MySQLResult:775108397,0,26>
```

```
dbSendQuery(connection, 'Drop Table If Exists zip_pop;')
```

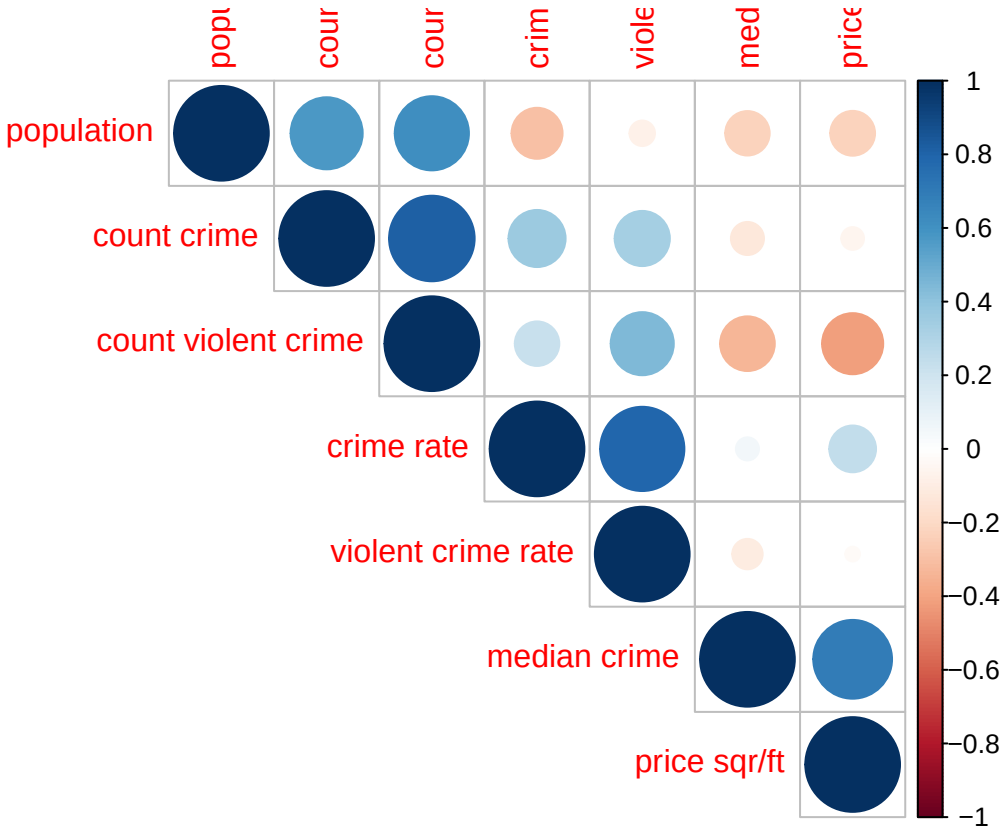
```
## <MySQLResult:942551092,0,27>
```

```
dbDisconnect(connection)
```

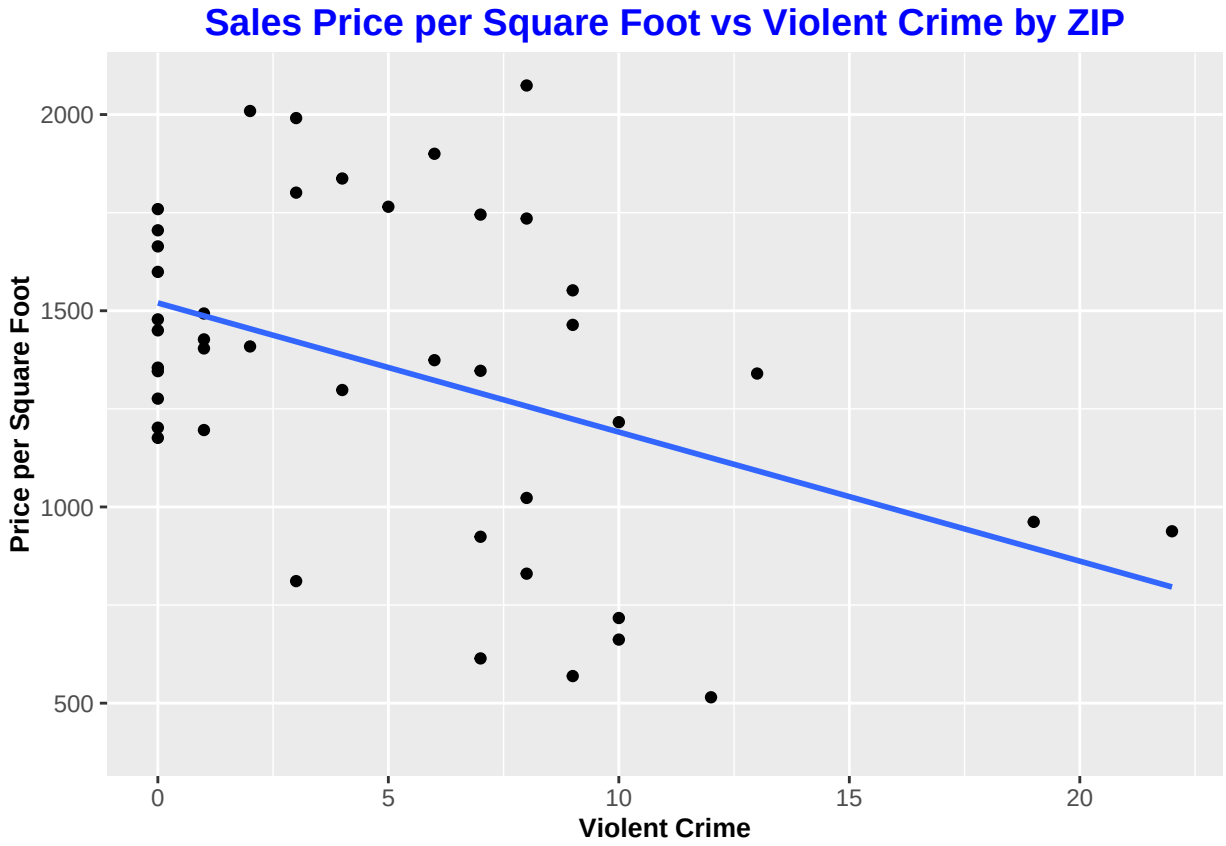
```
## [1] TRUE
```

Model the Data

```
#Correlation matrix
nyc_full$X <- NULL
m1<- subset(nyc_full, select=-c(zip,Borough,Neighborhood))
colnames(m1)<-c("median crime", "price sqr/ft", "count crime", "count violent crime",
                "population", "crime rate", "violent crime rate")
m<-cor(m1)
corrplot(m, type="upper", order="hclust", t1.col="black",t1.srt=45)
```




```
# Scatterplot with regression line
ggplot(nyc_full,aes(y = Price_sqft, x = freq_violent_crime)) +
    geom_point() + ggtitle("Sales Price per Square Foot vs Violent Crime by ZIP") +
    xlab("Violent Crime") + ylab("Price per Square Foot") +
    geom_smooth(method="lm", fill=NA) +
    theme(plot.title = element_text(color="blue", size=14, face="bold"),
          axis.title.x = element_text(color="black", size=10, face="bold"),
          axis.title.y = element_text(color="black", size=10, face="bold"))
```




```
# Regression analysis
lmt<-lm(Price_sqft ~ freq_violent_crime, nyc_full)
summary(lmt)
```

```
##
## Call:
## lm(formula = Price_sqft ~ freq_violent_crime, data = nyc_full)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -675.57 -267.43    6.02  244.03  817.33
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1519.88      82.50   18.42  < 2e-16 ***
## freq_violent_crime   -32.90      11.23   -2.93  0.00552 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 379.8 on 41 degrees of freedom
## Multiple R-squared:  0.1731, Adjusted R-squared:  0.153
## F-statistic: 8.585 on 1 and 41 DF,  p-value: 0.005518
```

Findings and Conclusion

Summary Analysis

All data was join by ZIP code, ZIP was the better standard to connect all data base, we try by neighborhood but each source had its own definition, so we abandoned. We aggregate all data by ZIP code.

There are differences between the full sales price among zip codes. The median sales price is different from mean, in some zipcode there is big change in the mean. The best central tendency here is median, which better reflect the reality."

The price per square/foot is more stable with lower difference between media and median, here mean can be adopted".

In our analysis we focus in the price per square/foot this better to do comparison, remove the size, in the value"

In the crime data by zip there are large variability, for both crimes. The variability illustrates some areas are more violent than others.

The times series plot by crime (See charts Tableau Public.) The mosaic plot shows, petit larceny is the most frequent, followed by harassment during the summer months so do overall crime rates.

Correlation Matrix Plot

The price square/foot has negative relationship with violent crime. The common crime don't have relationship with price square/foot or median sales price. The price square/foot don't have relationship with crime rate (crime/population).

Regression Analysis

There is negative relationship between violent crime and price square/foot, for each occurrence of violent crime the price drops US$ 32.90, at 0.05 significant level. There is a ($500) mean drop in price between areas with the most or least violent crimes. The crime frequency explain 17% of property value.

There is no relationship with price and crime rate or violent crime rate, this is an interesting finding, for people interested in how the crime impacts price. Is not taken into account in the number of people (population). This can lead to an incorrect crime perception.

"'

References

The definition for violent crime was created using two existing standards NCVS Bureau of Justice Statistics National Crime Victimization Survey and the UCR Federal Bureau of Investigation's Uniform Crime Report

1. Trulia real state https://www.trulia.com/home_prices/New_York/New_York-heat_map/city_by_neighborhood/ALP/nh/

2. NYC Opendata – City crime data https://data.cityofnewyork.us/Public-Safety/Historical-New-York-City-Crime-Data/hqhv-9zeg

3. US Census 2010 http://www.census.gov/2010census/data/

4. New York State https://www.health.ny.gov/statistics/cancer/registry/appendix/neighborhoods.htm

5. Violent crime definition https://en.wikipedia.org/wiki/Violent_crime

6. Google API https://developers.google.com/maps/pricing-and-plans/

7. R ggmap (Google maps) package https://cran.r-project.org/web/packages/ggmap/ggmap.pdf