NYPD Shooting Incident Data Report

2024-01-29

Purpose

Show core data science and R skills by importing, analyzing, and tidying the NYPD Shooting Incident data

Intro As concerns regarding gun violence intensify, I will explore the NYPD Shooting Incident data and delve into the numerical landscape to unveil patterns and insights. The aim is to distill meaningful trends without delving into broader societal debates. This report zeros in on the core details of the NYPD Shooting Incident data, focusing on statistical analyses to illuminate trends and relationships.

Step 1 Install `tidyverse` and `lubridate` which are the packages I'll need to preform my analysis

```
library(tidyverse)
library(lubridate)
```

Step 2 Read the data using `read_csv`

```
shooting_incident <-
  read_csv(
    "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
  )
```

Step 3 Check out first 6 rows using `head`. We are looking at incidents of shootings in the 5 boroughs of NYC. We have information surrounding the location, perp, and victims.

```
head(shooting_incident)
```

```
## # A tibble: 6 x 21
##   INCIDE~1 OCCUR~2 OCCUR~3 BORO  LOC_O~4 PRECI~5 JURIS~6 LOC_C~7 LOCAT~8 STATI~9
##      <dbl> <chr>   <time>  <chr> <chr>     <dbl>   <dbl> <chr>   <chr>   <lgl>
## 1   2.29e8 05/27/~ 21:30   QUEE~ <NA>        105       0 <NA>    <NA>    FALSE
## 2   1.37e8 06/27/~ 17:40   BRONX <NA>         40       0 <NA>    <NA>    FALSE
## 3   1.48e8 11/21/~ 03:56   QUEE~ <NA>        108       0 <NA>    <NA>    TRUE
## 4   1.47e8 10/09/~ 18:30   BRONX <NA>         44       0 <NA>    <NA>    FALSE
## 5   5.89e7 02/19/~ 22:58   BRONX <NA>         47       0 <NA>    <NA>    TRUE
## 6   2.20e8 10/21/~ 21:36   BROO~ <NA>         81       0 <NA>    <NA>    TRUE
## # ... with 11 more variables: PERP_AGE_GROUP <chr>, PERP_SEX <chr>,
## #   PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>, VIC_RACE <chr>,
## #   X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>, Longitude <dbl>,
## #   Lon_Lat <chr>, and abbreviated variable names 1: INCIDENT_KEY,
## #   2: OCCUR_DATE, 3: OCCUR_TIME, 4: LOC_OF_OCCUR_DESC, 5: PRECINCT,
## #   6: JURISDICTION_CODE, 7: LOC_CLASSFCTN_DESC, 8: LOCATION_DESC,
## #   9: STATISTICAL_MURDER_FLAG
```

Step 4 Lets take a look at the summary of the dataframe using `summary`

```r
summary(shooting_incident)
```

```
##   INCIDENT_KEY        OCCUR_DATE          OCCUR_TIME           BORO          
##  Min.   :  9953245   Length:27312       Length:27312      Length:27312      
##  1st Qu.: 63860880   Class :character   Class1:hms        Class :character  
##  Median : 90372218   Mode  :character   Class2:difftime   Mode  :character  
##  Mean   :120860536                      Mode  :numeric                      
##  3rd Qu.:188810230                                                          
##  Max.   :261190187                                                          
##                                                                             
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:27312       Min.   :  1.00   Min.   :0.0000    Length:27312      
##  Class :character   1st Qu.: 44.00   1st Qu.:0.0000    Class :character  
##  Mode  :character   Median : 68.00   Median :0.0000    Mode  :character  
##                     Mean   : 65.64   Mean   :0.3269                      
##                     3rd Qu.: 81.00   3rd Qu.:0.0000                      
##                     Max.   :123.00   Max.   :2.0000                      
##                                      NA's   :2                           
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP    
##  Length:27312       Mode :logical           Length:27312      
##  Class :character   FALSE:22046             Class :character  
##  Mode  :character   TRUE :5266              Mode  :character  
##                                                               
##                                                               
##                                                               
##                                                               
##    PERP_SEX          PERP_RACE         VIC_AGE_GROUP        VIC_SEX         
##  Length:27312       Length:27312       Length:27312       Length:27312      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude    
##  Length:27312       Min.   : 914928   Min.   :125757   Min.   :40.51  
##  Class :character   1st Qu.:1000029   1st Qu.:182834   1st Qu.:40.67  
##  Mode  :character   Median :1007731   Median :194487   Median :40.70  
##                     Mean   :1009449   Mean   :208127   Mean   :40.74  
##                     3rd Qu.:1016838   3rd Qu.:239518   3rd Qu.:40.82  
##                     Max.   :1066815   Max.   :271128   Max.   :40.91  
##                                                        NA's   :10     
##    Longitude        Lon_Lat         
##  Min.   :-74.25   Length:27312      
##  1st Qu.:-73.94   Class :character  
##  Median :-73.92   Mode  :character  
##  Mean   :-73.91                     
##  3rd Qu.:-73.88                     
##  Max.   :-73.70                     
##  NA's   :10
```

Step 5 Looks like the `OCCUR_DATE` is a character, lets try to change it to a date in a new column called `OCCUR_DATE_LUBRIDATE`

```
shooting_incident <- shooting_incident %>%
  mutate(OCCUR_DATE_LUBRIDATE = mdy(OCCUR_DATE))
```

Step 6 Confirm the date changes worked

```
summary(shooting_incident$OCCUR_DATE)
```

```
##    Length     Class      Mode
##     27312 character character
```

```
summary(shooting_incident$OCCUR_DATE_LUBRIDATE)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2006-01-01" "2009-07-18" "2013-04-29" "2014-01-06" "2018-10-15" "2022-12-31"
```

Step 7 It worked! Lets remove the old `OCCUR_DATE` and rename `OCCUR_DATE_LUBRIDATE` to `OCCUR_DATE`

```
shooting_incident <- shooting_incident %>%
  select(-c(OCCUR_DATE)) %>%
  rename(OCCUR_DATE = OCCUR_DATE_LUBRIDATE)
```

Step 8 Now we can take a look to see how the `VIC_AGE_GROUP`s are allocated

```
shooting_incident %>%
  group_by(VIC_AGE_GROUP) %>%
  summarise(Total = n()) %>%
  arrange(desc(Total))
```

```
## # A tibble: 7 x 2
##   VIC_AGE_GROUP Total
##   <chr>         <int>
## 1 25-44         12281
## 2 18-24         10086
## 3 <18            2839
## 4 45-64          1863
## 5 65+             181
## 6 UNKNOWN          61
## 7 1022              1
```

Step 9 There could be some funky values (1022 is obviously an error) so lets only include the values that are good. (I could exclude 1022, but I want to make sure I also exclude other "fat-fingered" values in the future, so I'll make it an include rather than exclude function)

```
shooting_incident <- shooting_incident %>%
  filter(VIC_AGE_GROUP %in% c("25-44", "18-24", "<18", "45-64", "65+", "UNKNOWN"))
```

Step 10 Check to make sure that looks better

```
shooting_incident %>%
  group_by(VIC_AGE_GROUP) %>%
  summarise(Total = n()) %>%
  arrange(desc(Total))
```

```
## # A tibble: 6 x 2
##   VIC_AGE_GROUP Total
##   <chr>         <int>
## 1 25-44         12281
## 2 18-24         10086
## 3 <18            2839
## 4 45-64          1863
## 5 65+             181
## 6 UNKNOWN          61
```

Step 11 I think I'll want to sort them as well, so lets rename them

```
shooting_incident <- shooting_incident %>%
  mutate(
    VIC_AGE_GROUP = case_when(
      VIC_AGE_GROUP == "<18" ~ "1. <18",
      VIC_AGE_GROUP == "18-24" ~ "2. 18-24",
      VIC_AGE_GROUP == "25-44" ~ "3. 25-44",
      VIC_AGE_GROUP == "45-64" ~ "4. 45-64",
      VIC_AGE_GROUP == "65+" ~ "5. 65+"
    )
  )
```

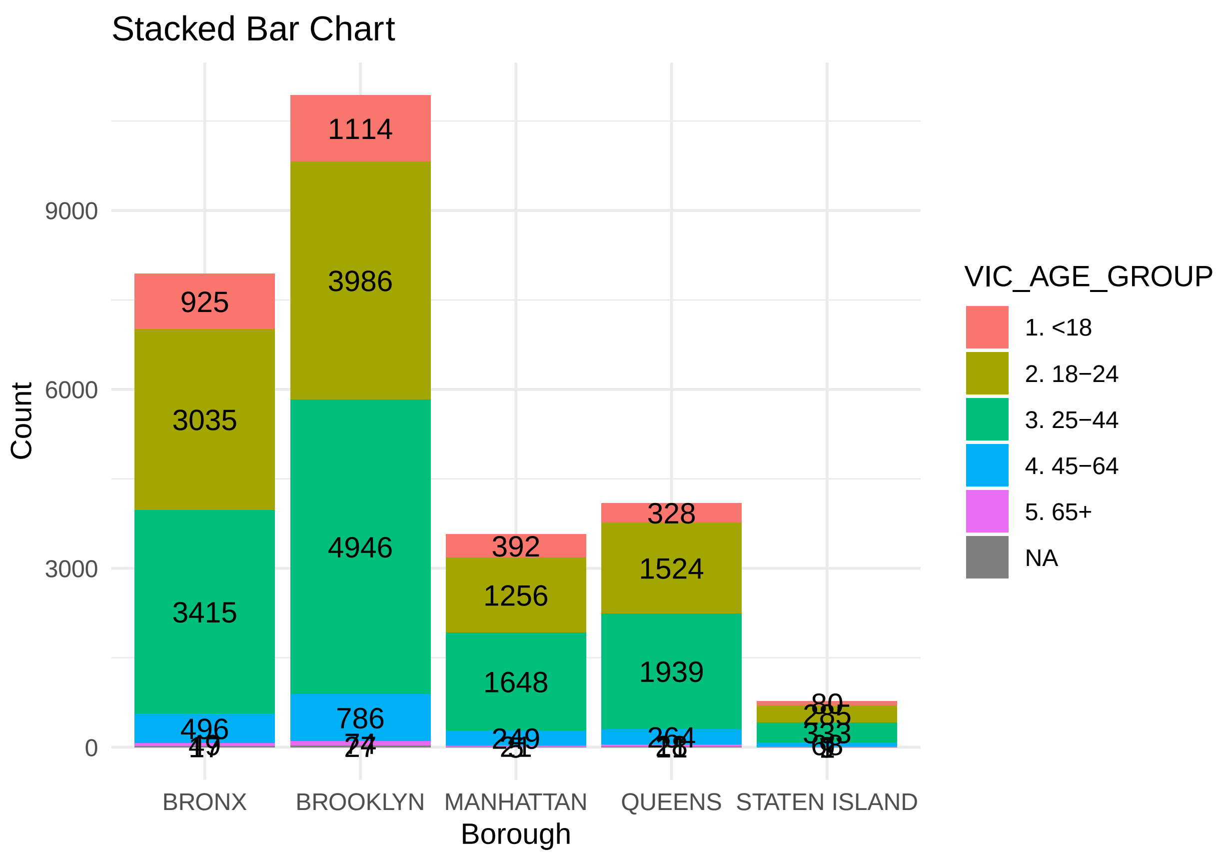
Step 12 Check to make sure that looks better, now arranging by `VIC_AGE_GROUP`

```
shooting_incident %>%
  group_by(VIC_AGE_GROUP) %>%
  summarise(Total = n()) %>%
  arrange(VIC_AGE_GROUP)
```

```
## # A tibble: 6 x 2
##   VIC_AGE_GROUP Total
##   <chr>         <int>
## 1 1. <18         2839
## 2 2. 18-24      10086
## 3 3. 25-44      12281
## 4 4. 45-64       1863
## 5 5. 65+          181
## 6 <NA>             61
```

Step 13 Let's create a stacked bar chart to see how the count is spread between the age groups in the different boroughs

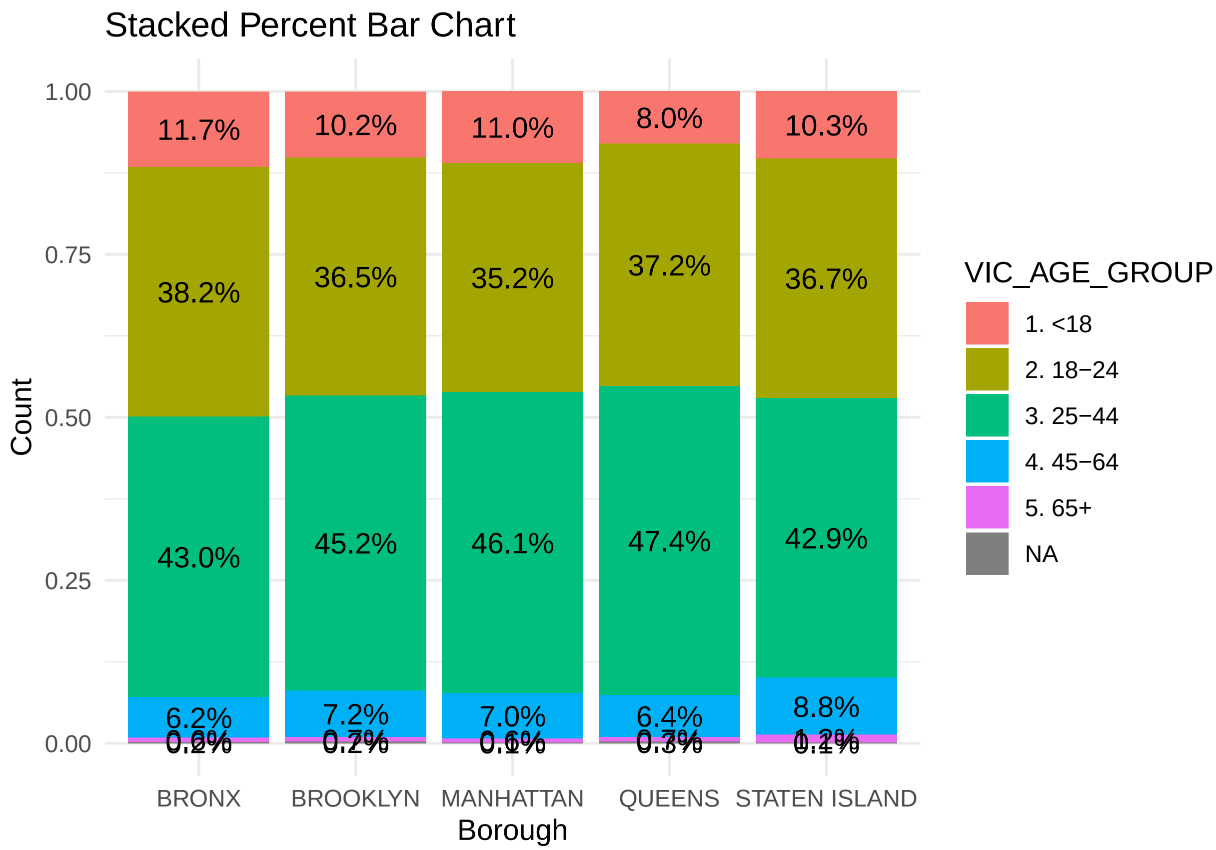
```r
ggplot(shooting_incident, aes(fill = VIC_AGE_GROUP, x = BORO)) +
  geom_bar(position = "stack", stat = "count") +
  geom_text(
    stat = 'count',
    aes(label = after_stat(count), group = VIC_AGE_GROUP),
    position = position_stack(vjust = 0.5)
  ) +
  labs(title = "Stacked Bar Chart",
       x = "Borough",
       y = "Count") +
  theme_minimal()
```



Step 14 That's a little hard to understand since the populations of the borough are pretty varied. We can make a stacked percent bar char to see if that helps

```r
shooting_incident %>%
  count(BORO, VIC_AGE_GROUP) %>%
  group_by(BORO) %>%
  mutate(pct = prop.table(n) * 100) %>%
  ggplot(aes(fill = VIC_AGE_GROUP, x = BORO, y = pct)) +
  geom_col(position = "fill") +
  labs(title = "Stacked Percent Bar Chart",
       x = "Borough",
       y = "Count") +
geom_text(aes(label = paste0(sprintf("%1.1f", pct), "%")), position = position_fill(vjust = 0.5)) +
theme_minimal()
```

```
  geom_text(aes(label = paste0(sprintf("%1.1f", pct), "%")), position = position_fill(vjust = 0.5)) +
  theme_minimal()
```



Step 15 We can see that as a % of all victims in the boroughs, 18-24 year olds make up a smaller % in Manhattan than the other boroughs, and 25-44 year-olds make up a smaller % in the Bronx, but I want to create a model to see if that is statistically significant. In order to do that, I want to preform a chi-squared test of homogeneity. To start, I need to create a contingency table

```
contingency_table <-
  table(shooting_incident$BORO, shooting_incident$VIC_AGE_GROUP)
print(contingency_table)
```

```
##
##                 1. <18 2. 18-24 3. 25-44 4. 45-64 5. 65+
##   BRONX            925     3035     3415      496     49
##   BROOKLYN        1114     3986     4946      786     74
##   MANHATTAN        392     1256     1648      249     21
##   QUEENS           328     1524     1939      264     28
##   STATEN ISLAND     80      285      333       68      9
```

Step 16 Now I can run the chi-squared test using `chisq.test`

```r
chi_squared_test <- chisq.test(contingency_table)
print(chi_squared_test)
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 72.13, df = 16, p-value = 4.215e-09
```

Step 17 The P value is super low! Lets dig into standardized residuals see where the biggest offenders are.

```r
chi_squared_test$stdres
```

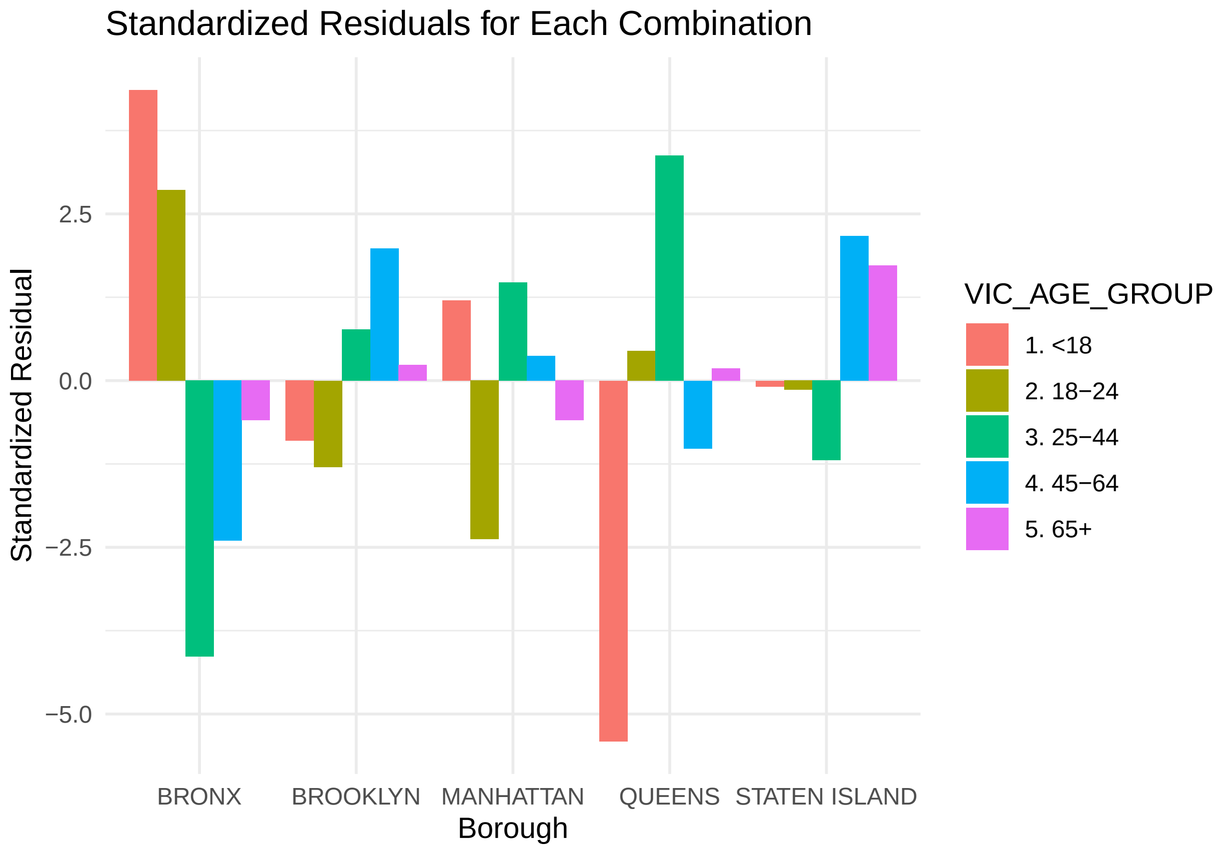
```
##
##                      1. <18    2. 18-24    3. 25-44    4. 45-64      5. 65+
##   BRONX          4.36130889  2.86213394 -4.13942715 -2.40353196 -0.59231168
##   BROOKLYN      -0.89951021 -1.29627712  0.76776056  1.97877949  0.23748047
##   MANHATTAN      1.20426435 -2.37637111  1.47577128  0.37034345 -0.59398710
##   QUEENS        -5.41034172  0.44874221  3.37296493 -1.01837163  0.18385613
##   STATEN ISLAND -0.08853754 -0.13959704 -1.19211805  2.16826016  1.72832299
```

Step 18 Lets graph the standard residual. But first we need to turn it into a data frame, and change the column names

```r
std_res_df <- as.data.frame(chi_squared_test$stdres) %>%
  rename(Borough = Var1,
         VIC_AGE_GROUP = Var2)
```

Step 19 Now we can graph it. The outsized negative standard residual in '<18' year-olds in Queens show they are less likely to be victims, while in the Bronx, the higher standard residual shows anyone less than 24 is more likely (and then the opposite for those aged 25-64).

```r
ggplot(std_res_df, aes(x = Borough, y = Freq, fill = VIC_AGE_GROUP)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Standardized Residuals for Each Combination",
       y = "Standardized Residual")  +
  theme_minimal()
```



Final thoughts - bias reduction In analyzing the NYPD Shooting Incident data, it's important to acknowledge potential biases despite efforts for objectivity. While statistical analyses, including chi-squared analysis, unveil patterns, it's crucial to note that statistical significance doesn't always translate practically. Inherent dataset limitations, such as underreporting and misclassification, introduce uncertainties, necessitating a cautious interpretation. Striving for a nuanced analysis, we are mindful of the broader societal implications within the complexities of the data.