DTSA5301 Assignment Week 3

2024-03-31

NYPD Shooting incident Data Report

The dataset consists of information around shooting incidents that occured in NYC from 2006 through the end of the previous calendar year. The data is manually extracted every quarter and reviewed by the Office of Management Analysis and Planning. The attached footnotes and data dictionary gives detailed information about the assumptions, method of collection and structure of the data.

This document does an exploratory analysis on the data and tests a few hypotheses. The goal of this is to demonstrate how reproducable research has to be performed.

Step 0 : Load necessary packages

Packages lubridate for manipulation on dates and Tidyverse for data cleaning, transformation etc.

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

Step 1 : Import and load the data

The data is available through catalog of open data *https://catalog.data.gov/dataset*. Link to the data has been copied and is directly being loaded from the internet. Showing the first few rows of the data below.

```
data_link = 'https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD'
nypd_shootings = read.csv(data_link)
head(nypd_shootings)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO LOC_OF_OCCUR_DESC PRECINCT
## 1    228798151 05/27/2021   21:30:00   QUEENS                       105
## 2    137471050 06/27/2014   17:40:00    BRONX                        40
## 3    147998800 11/21/2015   03:56:00   QUEENS                       108
## 4    146837977 10/09/2015   18:30:00    BRONX                        44
## 5     58921844 02/19/2009   22:58:00    BRONX                        47
## 6    219559682 10/21/2020   21:36:00 BROOKLYN                        81
##   JURISDICTION_CODE LOC_CLASSFCTN_DESC LOCATION_DESC STATISTICAL_MURDER_FLAG
## 1                 0                                                   false
## 2                 0                                                   false
## 3                 0                                                    true
## 4                 0                                                   false
## 5                 0                                                    true
## 6                 0                                                    true
```

```
##   PERP_AGE_GROUP PERP_SEX PERP_RACE VIC_AGE_GROUP VIC_SEX       VIC_RACE
## 1                                          18-24       M          BLACK
## 2                                          18-24       M          BLACK
## 3                                          25-44       M          WHITE
## 4                                            <18       M WHITE HISPANIC
## 5          25-44        M     BLACK         45-64       M          BLACK
## 6                                          25-44       M          BLACK
##   X_COORD_CD Y_COORD_CD Latitude Longitude
## 1    1058925   180924.0 40.66296 -73.73084
## 2    1005028   234516.0 40.81035 -73.92494
## 3    1007668   209836.5 40.74261 -73.91549
## 4    1006537   244511.1 40.83778 -73.91946
## 5    1024922   262189.4 40.88624 -73.85291
## 6    1004234   186461.7 40.67846 -73.92795
##                                      Lon_Lat
## 1 POINT (-73.73083868899994 40.662964620000025)
## 2  POINT (-73.92494232599995 40.81035186300006)
## 3  POINT (-73.91549174199997 40.74260663300004)
## 4  POINT (-73.91945661499994 40.83778200300003)
## 5  POINT (-73.85290950899997 40.88623791800006)
## 6 POINT (-73.92795224099996 40.678456718000064)
```

Step 2 : Check and understand the data

Here we make some basic checks on the data. We note the number of rows and columns in the data so that we can tally it later after cleaning. We also summarize the data and identify null values.

```
# Check number of rows and columns
dim(nypd_shootings)
```

```
## [1] 27312    21
```

```
# View Summary
summary(nypd_shootings)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:27312       Length:27312       Length:27312
##  1st Qu.: 63860880   Class :character   Class :character   Class :character
##  Median : 90372218   Mode  :character   Mode  :character   Mode  :character
##  Mean   :120860536
##  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       Length:27312           Length:27312
##  Class :character   Class :character       Class :character
##  Mode  :character   Mode  :character       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
```

```
# Check for missing values
colSums(is.na(nypd_shootings))
```

```
##            INCIDENT_KEY               OCCUR_DATE               OCCUR_TIME
##                       0                        0                        0
##                    BORO       LOC_OF_OCCUR_DESC                 PRECINCT
##                       0                        0                        0
##       JURISDICTION_CODE      LOC_CLASSFCTN_DESC           LOCATION_DESC
##                       2                        0                        0
## STATISTICAL_MURDER_FLAG          PERP_AGE_GROUP                 PERP_SEX
##                       0                        0                        0
##                PERP_RACE           VIC_AGE_GROUP                  VIC_SEX
##                       0                        0                        0
##                VIC_RACE              X_COORD_CD              Y_COORD_CD
##                       0                        0                        0
##                Latitude               Longitude                 Lon_Lat
##                      10                       10                        0
```

Only 10 data from Lat long missing and 2 data from jurisdiction is missing. For our analysis, we will not need these columns. Hence we can move to the next stage.

Step 3 : Tidying and transforming

On closer observation, it is found that though there are no null values, there are some blank strings and other anomalies. For example, the PERP_RACE column shows the non-compliant strings. We substitute incorrect values with the string 'UNKNOWN'.

```
unique(nypd_shootings$PERP_RACE)
```

```
## [1] ""                                "BLACK"
## [3] "UNKNOWN"                         "BLACK HISPANIC"
## [5] "ASIAN / PACIFIC ISLANDER"        "WHITE HISPANIC"
## [7] "WHITE"                           "(null)"
## [9] "AMERICAN INDIAN/ALASKAN NATIVE"
```

```
#Clean the data through a pipeline
nypd_shootings=nypd_shootings%>%mutate(PERP_AGE_GROUP=case_when(
  PERP_AGE_GROUP %in% c('940','','(null)','224','1020')~'UNKNOWN',
  TRUE ~ PERP_AGE_GROUP
  ))%>%mutate(PERP_SEX=case_when(
  PERP_SEX %in% c('','(null)')~'UNKNOWN',
  TRUE ~ PERP_SEX
  ))%>%mutate(PERP_RACE=case_when(
  PERP_RACE %in% c('','(null)')~'UNKNOWN',
  TRUE ~ PERP_RACE
  ))%>%mutate(VIC_AGE_GROUP=case_when(
  VIC_AGE_GROUP %in% c('1022')~'UNKNOWN',
  TRUE ~ VIC_AGE_GROUP
  ))
# Finally check if the dimensions match
dim(nypd_shootings)
```

```
## [1] 27312    21
```

Step 4 : Converting data types

We look up the data dictionary from the provider of the data and make appropriate changes to the data types

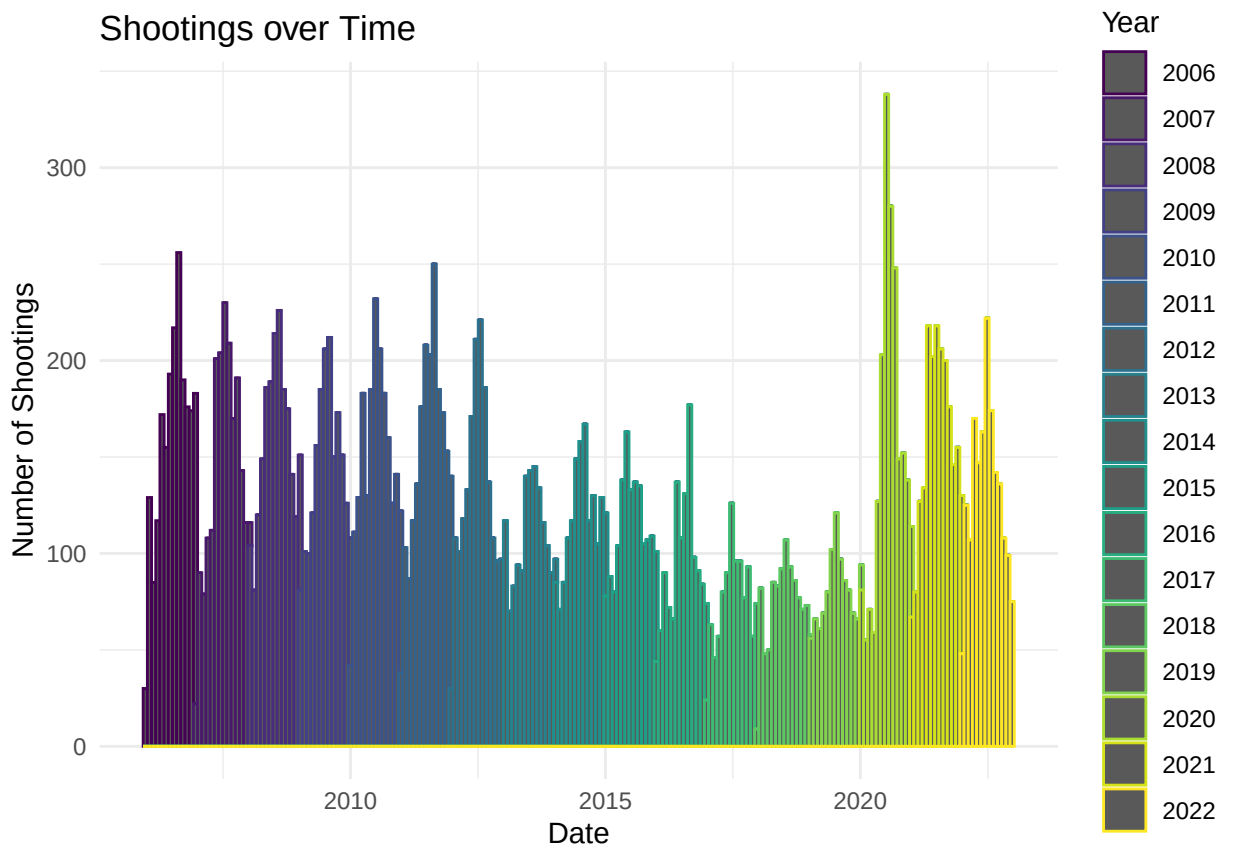
```
# Ensure proper data types
nypd_shootings$OCCUR_DATE <- as.Date(nypd_shootings$OCCUR_DATE,format = "%m/%d/%Y")
nypd_shootings$OCCUR_TIME <- hms(nypd_shootings$OCCUR_TIME)

# Convert categorical columns to factors
nypd_shootings$BORO <- as.factor(nypd_shootings$BORO)
# ... and similarly for other categorical columns
nypd_shootings$STATISTICAL_MURDER_FLAG <- as.factor(nypd_shootings$STATISTICAL_MURDER_FLAG)
nypd_shootings$PERP_AGE_GROUP <- as.factor(nypd_shootings$PERP_AGE_GROUP)
nypd_shootings$PERP_SEX <- as.factor(nypd_shootings$PERP_SEX)
nypd_shootings$PERP_RACE <- as.factor(nypd_shootings$PERP_RACE)
nypd_shootings$VIC_AGE_GROUP <- as.factor(nypd_shootings$VIC_AGE_GROUP)
nypd_shootings$VIC_SEX <- as.factor(nypd_shootings$VIC_SEX)
nypd_shootings$VIC_RACE <- as.factor(nypd_shootings$VIC_RACE)
```

Finally, the data is clean for some visualizations and exploratory analyses.

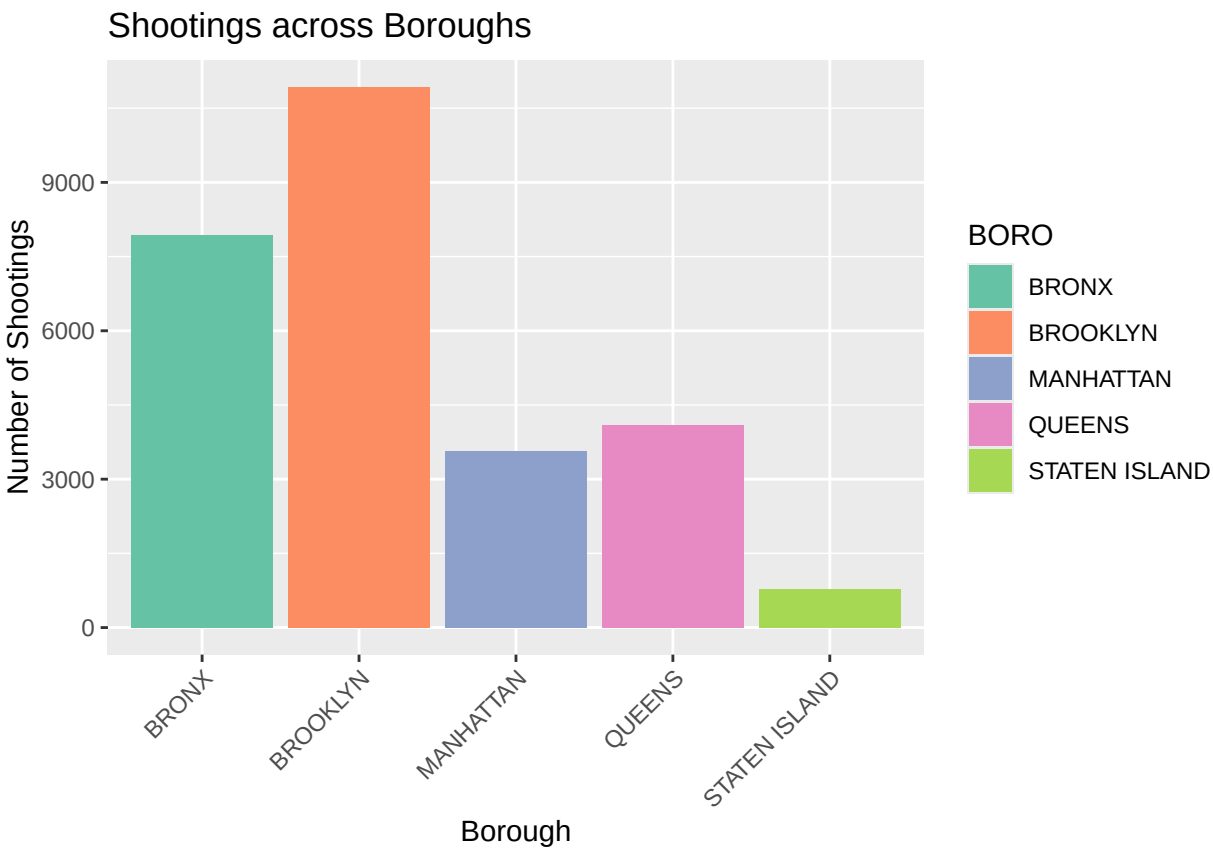
Step 5 : Visualizing the data

We observe number of shooting incidents by year and see how the timeseries is behaving. We also see the behaviour by boroughs.

```
# Time series with color by year
ggplot(nypd_shootings, aes(x = OCCUR_DATE, color = factor(year(OCCUR_DATE)))) +
  geom_histogram(binwidth = 30) +
  scale_color_viridis_d() + # Discrete color scale
  labs(x = "Date", y = "Number of Shootings",
       title = "Shootings over Time", color = "Year") +
  theme_minimal()
```




```
# Shootings by borough
ggplot(nypd_shootings, aes(x = BORO, fill = BORO)) +
  geom_bar() +
  scale_fill_brewer(palette = "Set2") + # Colorful qualitative palette
  labs(x = "Borough", y = "Number of Shootings",
       title = "Shootings across Boroughs") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

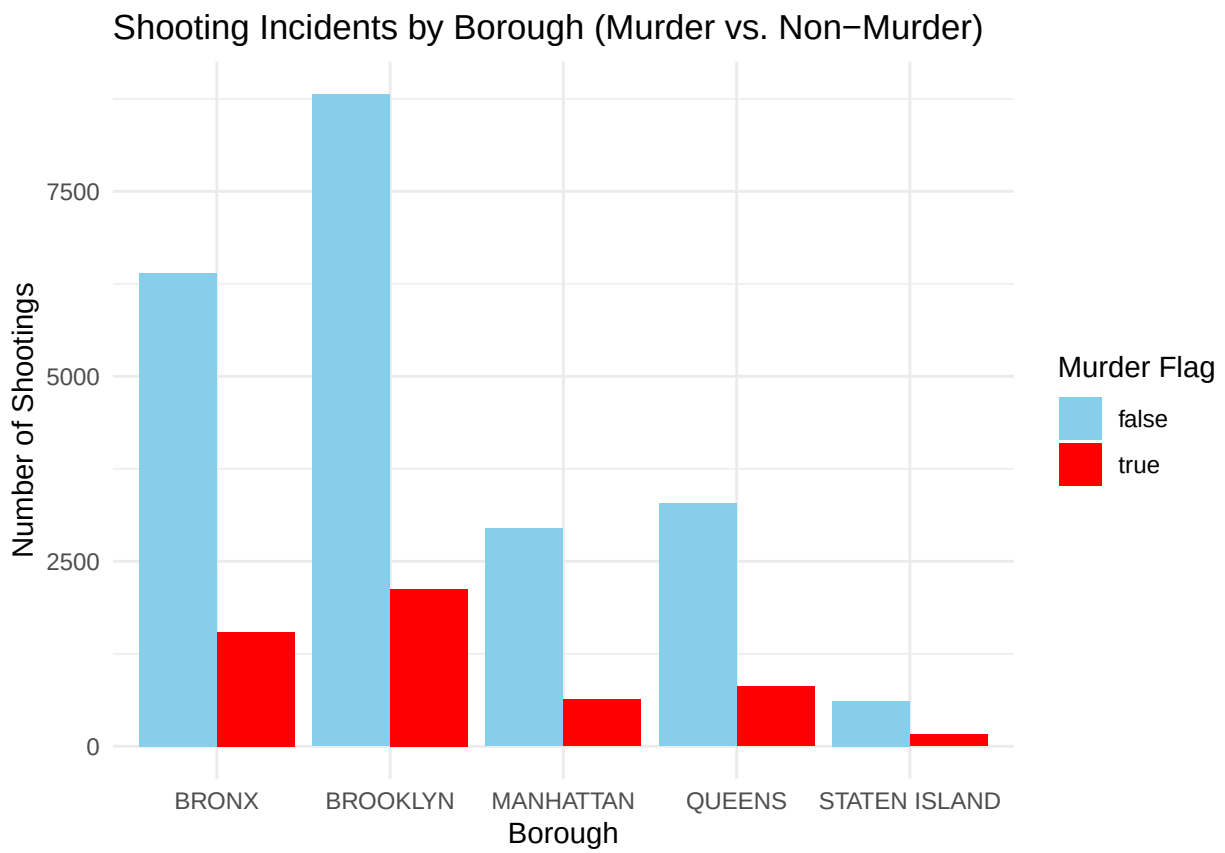


Step 6 : Analyzing the data

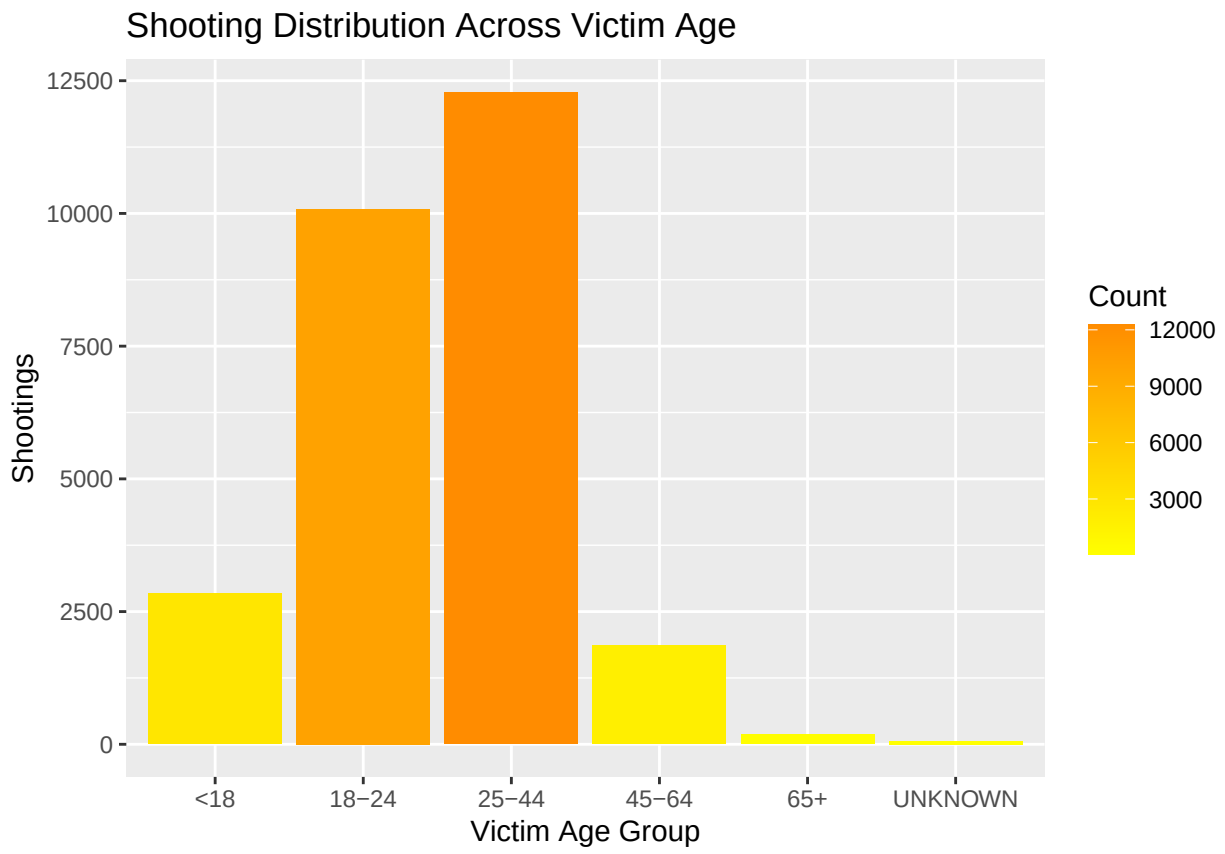
We see the relation between the number of murders and non-murders by boroughs. We also observe the victims age group and analyze that younger people are disproportionately more affected.

```
# Murder vs Non-murder shootings
ggplot(nypd_shootings, aes(x = BORO, y = ..count.., fill = STATISTICAL_MURDER_FLAG)) +
  geom_bar(position = "dodge") +  # For side-by-side comparison
  scale_fill_manual(values = c("skyblue", "red")) +
  labs(x = "Borough", y = "Number of Shootings", fill = "Murder Flag",
       title = "Shooting Incidents by Borough (Murder vs. Non-Murder)") +
  theme_minimal()
```

```
## Warning: The dot-dot notation (‘..count..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(count)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
# Victim age data
ggplot(nypd_shootings, aes(x = VIC_AGE_GROUP, fill = ..count..)) +
  geom_bar() +
  scale_fill_gradient(low = "yellow", high = "darkorange") +
  labs(x = "Victim Age Group", y = "Shootings", fill = "Count",
       title = "Shooting Distribution Across Victim Age")
```



Step 7 : Modelling the data

Finally, we model the relation between Borough and perpetrators age group and find that different boroughs have different age group association. This relationship may be vital to law enforcement who can look for differential strategies across the boroughs instead of the usual 'one size fits all' method.

We test the hypothesis that the perpetrators age group is independent of the boro in which the crime has occured. The low p-value of this test shows that the hypothesis is false. The image in the end section visually corroborates this finding.

```
#linm = lm(STATISTICAL_MURDER_FLAG~BORO,data = nypd_shootings)
#summary(linm)
# The image file has been generated thus
#library(vcd)
#png('mos1.png',height=13,width=13,units='in',res=100)
#mosaic(~ BORO + PERP_AGE_GROUP, data = nypd_shootings, shade = TRUE,abbreviate=1)
#dev.off()
chisq.test(table(nypd_shootings$BORO,nypd_shootings$PERP_AGE_GROUP))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(nypd_shootings$BORO, nypd_shootings$PERP_AGE_GROUP)
## X-squared = 458.32, df = 20, p-value < 2.2e-16
```

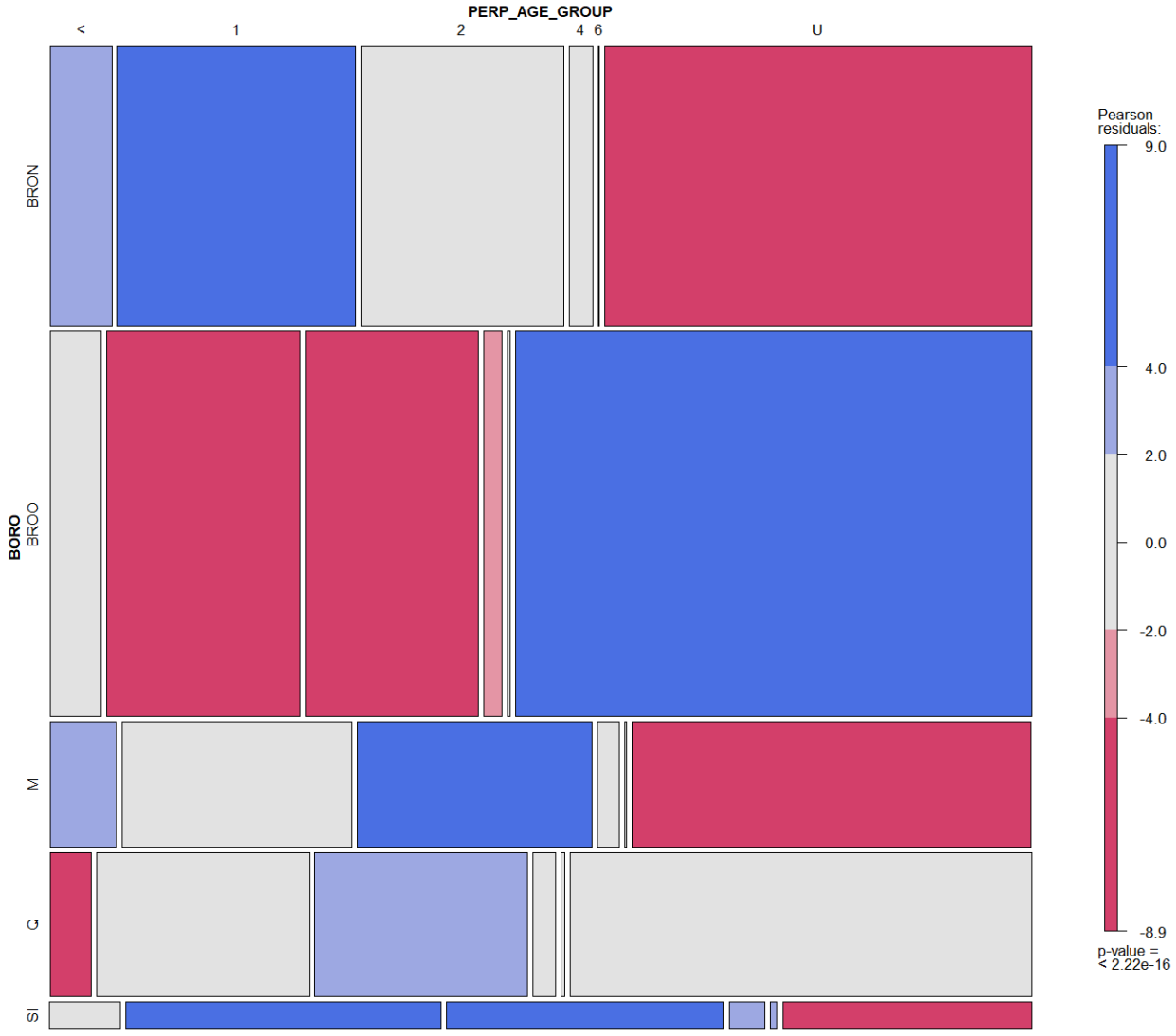


Figure 1: Mosaic Plot

Step 8 : Identifying possible sources of bias

The possible sources of bias are listed below: 1. Missing data : Though we have not filtered out missing data from the set, there are a number of datapoints with label 'UNKNOWN'. These may affect the findings. 2. Data collection inaccuracies : The foot notes describe some sources of bias. For example, if the shooting incident happened in a train, the next stop is taken as the location. + Other sources may include outliers or unreasonable values 3. Sampling bias : It is possible that the dataset doesnot reflect the true proportions of shooting incidents across different locations, time or demographic groups. 4. Implicit bias in reporting : There may be certain types of incidents which may be more likely to be reported. Language of reporting may also introduce bias. 5. Personal bias: Given the sensitive nature of the data, I may have held some pre-conceived notions against certain neighbourhood, race or age group. I hope that the same would be mitigated through standardized reporting and oversight from peers.

Conclusion:

```
sessionInfo()
```

Our analysis of the NYPD shooting incident dataset revealed several key insights. we observed seasonal trends in shooting and disparate distribution of perpetretors age group across boroughs. Additionally, certain precincts appear to have a disproportionate number of shootings, even after accounting for other variables. However, data accuracy and bias has to be closely investigated for a final conclusion on the topic.

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22631)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_India.utf8  LC_CTYPE=English_India.utf8
## [3] LC_MONETARY=English_India.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_India.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2
##  [5] readr_2.1.5     tidyr_1.3.1     tibble_3.2.1    ggplot2_3.5.0
##  [9] tidyverse_2.0.0 lubridate_1.9.3
##
## loaded via a namespace (and not attached):
##  [1] highr_0.10         RColorBrewer_1.1-3 pillar_1.9.0       compiler_4.2.2
##  [5] tools_4.2.2        digest_0.6.35      viridisLite_0.4.2  evaluate_0.23
##  [9] lifecycle_1.0.4    gtable_0.3.4       timechange_0.3.0   pkgconfig_2.0.3
## [13] rlang_1.1.3        cli_3.6.2          rstudioapi_0.15.0  yaml_2.3.8
## [17] xfun_0.42          fastmap_1.1.1      withr_3.0.0        knitr_1.45
## [21] generics_0.1.3     vctrs_0.6.5        hms_1.1.3          grid_4.2.2
## [25] tidyselect_1.2.1   glue_1.7.0         R6_2.5.1           fansi_1.0.6
## [29] rmarkdown_2.26     farver_2.1.1       tzdb_0.4.0         magrittr_2.0.3
```

```
## [33] scales_1.3.0       htmltools_0.5.7    colorspace_2.1-0   labeling_0.4.3
## [37] utf8_1.2.4         stringi_1.8.3      munsell_0.5.0      crayon_1.5.2
```