Peer-graded Assignment: NYPD Shooting Incident Data Report

Rogerio Oliveira

10-Sep-2021

1 - Import Libraries

```
# Install libraries
# install.packages("gtools")
# install.packages("tidyverse")
# install.packages("lubridate")
# install.packages("kableExtra")

# Load libraries
library(gtools)
library(tidyverse)
library(lubridate)
library(kableExtra)
```

```
## Warning: package 'kableExtra' was built under R version 4.1.1
```

2 - Load Datasets

2.1 - NYPD Shooting Incidents

- Source: data extracted from NYC OpenData website at https://data.cityofnewyork.us

```
url_in = "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"

ny_shootings = read_csv(url_in)

# Quick peek at the dataframe
head(ny_shootings) %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12) %>%
  scroll_box(width = "100%")
```

INCIDENT_KEY	OCCUR_DATE	OCCUR_TIME	BORO	PRECINCT	JURISDICTION_CODE	LOCATION_DESC	ST
201575314	08/23/2019	22:10:00	QUEENS	103	0	NA	FA

INCIDENT_KEY	OCCUR_DATE	OCCUR_TIME	BORO	PRECINCT	JURISDICTION_CODE	LOCATION_DESC	ST
205748546	11/27/2019	15:54:00	BRONX	40	0	NA	FA
193118596	02/02/2019	19:40:00	MANHATTAN	23	0	NA	FA
204192600	10/24/2019	00:52:00	STATEN ISLAND	121	0	PVT HOUSE	TR
201483468	08/22/2019	18:03:00	BRONX	46	0	NA	FA
198255460	06/07/2019	17:50:00	BROOKLYN	73	0	NA	FA

```
# Dataframe structure
str(ny_shootings)
```

```
## spec_tbl_df [23,568 x 19] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ INCIDENT_KEY          : num [1:23568] 2.02e+08 2.06e+08 1.93e+08 2.04e+08 2.01e+08 ...
##  $ OCCUR_DATE            : chr [1:23568] "08/23/2019" "11/27/2019" "02/02/2019" "10/24/2019"
...
##  $ OCCUR_TIME            : chr [1:23568] "22:10:00" "15:54:00" "19:40:00" "00:52:00" ...
##  $ BORO                  : chr [1:23568] "QUEENS" "BRONX" "MANHATTAN" "STATEN ISLAND" ...
##  $ PRECINCT              : num [1:23568] 103 40 23 121 46 73 81 67 114 69 ...
##  $ JURISDICTION_CODE     : num [1:23568] 0 0 0 0 0 0 0 0 2 0 ...
##  $ LOCATION_DESC         : chr [1:23568] NA NA NA "PVT HOUSE" ...
##  $ STATISTICAL_MURDER_FLAG: logi [1:23568] FALSE FALSE FALSE TRUE FALSE FALSE ...
##  $ PERP_AGE_GROUP        : chr [1:23568] NA "<18" "18-24" "25-44" ...
##  $ PERP_SEX              : chr [1:23568] NA "M" "M" "M" ...
##  $ PERP_RACE             : chr [1:23568] NA "BLACK" "WHITE HISPANIC" "BLACK" ...
##  $ VIC_AGE_GROUP         : chr [1:23568] "25-44" "25-44" "18-24" "25-44" ...
##  $ VIC_SEX               : chr [1:23568] "M" "F" "M" "F" ...
##  $ VIC_RACE              : chr [1:23568] "BLACK" "BLACK" "BLACK HISPANIC" "BLACK" ...
##  $ X_COORD_CD            : num [1:23568] 1037451 1006789 999347 938149 1008224 ...
##  $ Y_COORD_CD            : num [1:23568] 193561 237559 227795 171781 250621 ...
##  $ Latitude              : num [1:23568] 40.7 40.8 40.8 40.6 40.9 ...
##  $ Longitude             : num [1:23568] -73.8 -73.9 -73.9 -74.2 -73.9 ...
##  $ Lon_Lat               : chr [1:23568] "POINT (-73.80814071699996 40.697805308000056)" "POIN
T (-73.91857061799993 40.81869973000005)" "POINT (-73.94547965999999 40.791916091000076)" "POINT
(-74.16610830199996 40.63806398200006)" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   INCIDENT_KEY = col_double(),
##   ..   OCCUR_DATE = col_character(),
##   ..   OCCUR_TIME = col_character(),
##   ..   BORO = col_character(),
##   ..   PRECINCT = col_double(),
##   ..   JURISDICTION_CODE = col_double(),
##   ..   LOCATION_DESC = col_character(),
##   ..   STATISTICAL_MURDER_FLAG = col_logical(),
##   ..   PERP_AGE_GROUP = col_character(),
##   ..   PERP_SEX = col_character(),
##   ..   PERP_RACE = col_character(),
##   ..   VIC_AGE_GROUP = col_character(),
##   ..   VIC_SEX = col_character(),
##   ..   VIC_RACE = col_character(),
##   ..   X_COORD_CD = col_number(),
##   ..   Y_COORD_CD = col_number(),
##   ..   Latitude = col_double(),
##   ..   Longitude = col_double(),
##   ..   Lon_Lat = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

2.1.1 - Date Formatting

- An initial view of the NYPD Shooting Incidents dataframe structure reveals that following steps will need to be carried out prior to the data cleanup step:
    - Convert OCCUR_DATE to date object
    - Convert OCCUR_TIME TO time object

```
# Format date and time
ny_shootings = ny_shootings %>% mutate(OCCUR_DATE = mdy(OCCUR_DATE))
ny_shootings = ny_shootings %>% mutate(OCCUR_TIME = hms(OCCUR_TIME))
```

2.2 - New York City Population

- Some of the analysis will be performed in relation to the NYC borough populations, therefore we will need to add a population dataset to the report.
- Source: data extracted from “Open NY” website at https://data.ny.gov/

```
url_in = "https://data.ny.gov/api/views/krt9-ym2k/rows.csv?accessType=DOWNLOAD&sorting=true"

ny_population = read_csv(url_in)

# Quick peek at the dataframe
# rbind(head(ny_population, 5), tail(ny_population, 5)) %>%
#  kbl() %>%
#    kable_styling(bootstrap_options = c("striped", "condensed"),
#                  full_width = FALSE, position = "left", font_size = 12) %>%
#  scroll_box(width = "100%")

# Dataframe structure
str(ny_population)
```

```
## spec_tbl_df [3,528 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ FIPS Code   : num [1:3528] 36000 36001 36003 36005 36007 ...
##  $ Geography   : chr [1:3528] "New York State" "Albany County" "Allegany County" "Bronx County"
...
##  $ Year        : num [1:3528] 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ Program Type: chr [1:3528] "Postcensal Population Estimate" "Postcensal Population Estimate"
"Postcensal Population Estimate" "Postcensal Population Estimate" ...
##  $ Population  : num [1:3528] 19336776 303654 45587 1401142 189420 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `FIPS Code` = col_double(),
##   ..   Geography = col_character(),
##   ..   Year = col_double(),
##   ..   `Program Type` = col_character(),
##   ..   Population = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

2.2.1 - Filter Out non-NYC Counties

- The population dataset includes all counties of the NY state. We need to filter out non-NYC counties.
- In addition, we need to convert the resulting county names to borough names as used in the NYPD shooting dataset.

```
# Function to map NY county names to NYC borough names
ny_county_to_boro <- function(ny_county) {
    case_when(ny_county == 'Bronx County' ~ 'BRONX',
              ny_county == 'Kings County' ~ 'BROOKLYN',
              ny_county == 'New York County' ~ 'MANHATTAN',
              ny_county == 'Queens County' ~ 'QUEENS',
              ny_county == 'Richmond County' ~ 'STATEN ISLAND')
}

# Select subset of columns from the NY state population dataset and
#   filter by the counties that are part of NYC
ny_population = ny_population %>% select(c(Geography, Year, Population)) %>%
    filter(Geography %in%
           c('New York County', 'Kings County', 'Bronx County', 'Richmond County', 'Queens County'
)) %>%
     setNames(c('BORO', 'YEAR', 'POPULATION'))

# The population dataset lists the geographies by county
# We need to map them to NYC borough names
ny_population$BORO = ny_population$BORO %>% map(ny_county_to_boro)
```

2.2.2 - Filter Out Years Not Covered by NYPD Shooting Dataset

- To ensure both datasets cover the same timeframe, we will limit the entries in population dataset to those present in the NYPD Shooting dataset.

```
# Extract years covered by NYPD shootings dataset
nypd_shootings_years = year(ny_shootings$OCCUR_DATE) %>% unique()

# Filter out years in the population dataset that are not covered by NYPD shooting dataset
ny_population = ny_population[ny_population$YEAR %in% nypd_shootings_years, ]

# Convert boro column from list to string
ny_population$BORO = as.character(ny_population$BORO)

# Quick peek at the resulting dataframe
rbind(head(ny_population, 5), tail(ny_population, 5)) %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12) %>%
  scroll_box(width = "100%")
```

BORO	YEAR	POPULATION
BRONX	2020	1401142
BROOKLYN	2020	2538934
MANHATTAN	2020	1611989
QUEENS	2020	2225821
STATEN ISLAND	2020	475327
BRONX	2006	1348164

BORO	YEAR	POPULATION
BROOKLYN	2006	2436132
MANHATTAN	2006	1578171
QUEENS	2006	2173862
STATEN ISLAND	2006	457577

2.3 - 2019 NYC Poverty Rates

- The 2019 NYC Poverty Rates will be used later in the analysis to try to find correlations between higher shooting rates and poverty rates.
- Source: data extracted from the US Census website at https://www.census.gov

```
# Dataset with poverty rates in NYC as of 2019
url_in = "https://www.census.gov/quickfacts/fact/csv/newyorkcitynewyork,bronxcountynewyork,kingsco
untynewyork,newyorkcountynewyork,queenscountynewyork,richmondcountynewyork/PST045219"

ny_poverty = read_csv(url_in)

# Parse and wrangle the data
ny_poverty = ny_poverty %>% filter(Fact == 'Persons in poverty, percent') %>%
    select('Fact',
           'Bronx County, New York',
           'Kings County, New York',
           'New York County, New York',
           'Queens County, New York',
           'Richmond County, New York') %>%
    setNames(c('Fact', 'BRONX', 'BROOKLYN', 'MANHATTAN', 'QUEENS', 'STATEN ISLAND'))

# Transpose table and set column names
ny_poverty = stack(ny_poverty[, -1]) %>% setNames(c('2019_POVERTY_RATES', 'BORO'))

# Reorder columns
ny_poverty = ny_poverty[, c(2, 1)]

# Quick peek at the dataframe
ny_poverty %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

BORO	2019_POVERTY_RATES
BRONX	26.2%
BROOKLYN	17.7%
MANHATTAN	14.1%
QUEENS	11.0%

BORO	2019_POVERTY_RATES
STATEN ISLAND	9.1%

3 - Bias Identification

- The dataset contains columns indicating the perpetrator race and the victim(s) race. In my opinion, these columns serve no purpose in advancing the analysis or in uncovering insights and can lead to conscious or unconscious bias.
- I have identified these columns as a potential **bias sources** and will therefore remove them from the dataframe.

```
# Remove race columns from dataframe
ny_shootings = ny_shootings %>% select(-c(PERP_RACE, VIC_RACE))
```

4 - Data Cleanup and Transformation

4.1 - Check for Columns with NA Values

- The first step in the data cleaning process will be dealing with NA values. Let's check for columns with NA values.

```
# Check for NAs
  apply(ny_shootings, 2, function(x) any(is.na(x))) %>%
    kbl() %>%
      kable_styling(bootstrap_options = c("striped", "condensed"),
                    full_width = FALSE, position = "left", font_size = 12)
```

	x
INCIDENT_KEY	FALSE
OCCUR_DATE	FALSE
OCCUR_TIME	FALSE
BORO	FALSE
PRECINCT	FALSE
JURISDICTION_CODE	TRUE
LOCATION_DESC	TRUE
STATISTICAL_MURDER_FLAG	FALSE
PERP_AGE_GROUP	TRUE
PERP_SEX	TRUE
VIC_AGE_GROUP	FALSE
VIC_SEX	FALSE
X_COORD_CD	FALSE
Y_COORD_CD	FALSE

	x
Latitude	FALSE
Longitude	FALSE
Lon_Lat	FALSE

- Several columns contain NA values and these values will be converted to “UNKNOWN” for better readability.
- JURISDICTION_CODE is a candidate for a factor, so the NA values will be changed to -1 and an “UNKNOWN” label will be assigned to it when the column is converted to factor later in the data cleaning step.

```
# Replace NAs with "UNKNOWN"
ny_shootings$LOCATION_DESC = ny_shootings$LOCATION_DESC %>%
  replace_na('UNKNOWN')
ny_shootings$PERP_AGE_GROUP = ny_shootings$PERP_AGE_GROUP %>% replace_na('UNKNOWN')
ny_shootings$PERP_SEX = ny_shootings$PERP_SEX %>% replace_na('UNKNOWN')

# JURISDICTION_CODE is a candidate for a factor, -1 will be assigned to NA values
ny_shootings$JURISDICTION_CODE = ny_shootings$JURISDICTION_CODE %>% replace_na(-1)
```

4.2 - Check for invalid values or misspellings

- The next step in the data cleaning process will consist of checking individual columns for invalid values or misspellings. One way to do it is to group values and count the occurrences. This will quickly show any potential values that need to be cleaned up.

4.2.1 - BORO column

```
ny_shootings %>% group_by(BORO) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

BORO	n
BRONX	6700
BROOKLYN	9722
MANHATTAN	2921
QUEENS	3527
STATEN ISLAND	698

- There are no apparent invalid/misspelled values in the BORO column.

4.2.2 - JURISDICTION_CODE column

```
ny_shootings %>% group_by(JURISDICTION_CODE) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

JURISDICTION_CODE	n
-1	2
0	19624
1	54
2	3888

- There are no apparent invalid/misspelled values in the JURISDICTION_CODE column.

4.2.3 - LOCATION_DESC column

```
ny_shootings %>% group_by(LOCATION_DESC) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

LOCATION_DESC	n
ATM	1
BANK	1
BAR/NIGHT CLUB	558
BEAUTY/NAIL SALON	100
CANDY STORE	6
CHAIN STORE	5
CHECK CASH	1
CLOTHING BOUTIQUE	14
COMMERCIAL BLDG	234
DEPT STORE	5
DOCTOR/DENTIST	1
DRUG STORE	11
DRY CLEANER/LAUNDRY	30
FACTORY/WAREHOUSE	6
FAST FOOD	98
GAS STATION	53
GROCERY/BODEGA	572
GYM/FITNESS FACILITY	3
HOSPITAL	38
HOTEL/MOTEL	24
JEWELRY STORE	12
LIQUOR STORE	36

LOCATION_DESC	n
LOAN COMPANY	1
MULTI DWELL - APT BUILD	2551
MULTI DWELL - PUBLIC HOUS	4230
NONE	175
PHOTO/COPY STORE	1
PVT HOUSE	858
RESTAURANT/DINER	188
SCHOOL	1
SHOE STORE	9
SMALL MERCHANT	25
SOCIAL CLUB/POLICY LOCATI	66
STORAGE FACILITY	1
STORE UNCLASSIFIED	35
SUPERMARKET	19
TELECOMM. STORE	5
UNKNOWN	13581
VARIETY STORE	11
VIDEO STORE	2

- There are no apparent invalid/misspelled values in the LOCATION_DESC column.

4.2.4 - STATISTICAL_MURDER_FLAG column

```
ny_shootings %>% group_by(STATISTICAL_MURDER_FLAG) %>% count()%>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

STATISTICAL_MURDER_FLAG	n
FALSE	19080
TRUE	4488

- There are no apparent invalid/misspelled values in the STATISTICAL_MURDER_FLAG column.

4.2.5 - PERP_AGE_GROUP column

```
ny_shootings %>% group_by(PERP_AGE_GROUP) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

PERP_AGE_GROUP	n
<18	1354
1020	1
18-24	5448
224	1
25-44	4613
45-64	481
65+	54
940	1
UNKNOWN	11615

- There are clearly some invalid/misspelled values in the PERP_AGE_GROUP column and will be fixed by converting the invalid values to “UNKNOWN”:

```
# Convert invalid ages to "UNKNOWN"
ny_shootings$PERP_AGE_GROUP[ny_shootings$PERP_AGE_GROUP
                            %in% c('1020', '224', '940')] = "UNKNOWN"
ny_shootings %>% group_by(PERP_AGE_GROUP) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

PERP_AGE_GROUP	n
<18	1354
18-24	5448
25-44	4613
45-64	481
65+	54
UNKNOWN	11618

4.2.6 - PERP_SEX column

```
ny_shootings %>% group_by(PERP_SEX) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

PERP_SEX	n

PERP_SEX	n
F	334
M	13305
U	1504
UNKNOWN	8425

- There are values marked as “U” and others as “UNKNOWN” in the PERP_SEX column, so I will change the “U” values to “UNKNOWN” for consistency:

```
# Convert "U" in PERP_SEX to "UNKNOWN"
ny_shootings$PERP_SEX[ny_shootings$PERP_SEX %in% c('U')] = "UNKNOWN"

ny_shootings %>% group_by(PERP_SEX) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

PERP_SEX	n
F	334
M	13305
UNKNOWN	9929

4.2.7 - VIC_AGE_GROUP column

```
ny_shootings %>% group_by(VIC_AGE_GROUP) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

VIC_AGE_GROUP	n
<18	2525
18-24	9000
25-44	10287
45-64	1536
65+	155
UNKNOWN	65

- There are no apparent invalid/misspelled values in VIC_AGE_GROUP column.

4.2.8 - VIC_SEX column

```
ny_shootings %>% group_by(VIC_SEX) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

VIC_SEX	n
F	2195
M	21353
U	20

- There are values marked as “U” in the VIC_SEX column, so I will change the “U” values to “UNKNOWN” for consistency:

```
# Convert "U" in VIC_SEX to "UNKNOWN"
ny_shootings$VIC_SEX[ny_shootings$VIC_SEX %in% c('U')] = "UNKNOWN"

ny_shootings %>% group_by(VIC_SEX) %>% count() %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

VIC_SEX	n
F	2195
M	21353
UNKNOWN	20

4.3 - Drop Unused Columns

- Lat/Long or X,Y coordinates columns will not be used, therefore they will be dropped from the dataframe.

```
# Drop unused columns from dataframe
ny_shootings = ny_shootings %>%
  select(-c(X_COORD_CD, Y_COORD_CD, Latitude, Longitude, Lon_Lat))
```

4.4 - Convert Categorical Values to Factors

- As per column specification in the metadata documentation, JURISDICTION_CODE contains categorical values. We will represent it as factors for better readability.

```
ny_shootings$JURISDICTION_CODE =
  factor(ny_shootings$JURISDICTION_CODE,
         labels = c('UNKNOWN', 'PATROL', 'TRANSIT', 'HOUSING'))
```

5 - Data Engineering/Transformation

5.1 - Add a Column to Represent Time of The Day

- The dataframe contains the timestamp of the incidents. It will be interesting to aggregate the incidents in time periods in order to assess if there are more incidents during a particular time of the day.
- For the purpose of this analysis, hours between 1AM and 6AM will be considered “NIGHT”, between 7AM and 12PM will be “MORNING”, between 1PM and 6PM will be “AFTERNOON” and between 7PM and 12AM will be “EVENING”.

```
# Function to categorize time of the day as a function of the hour of the incident
time_of_day <- function(time) {
    hr = hour(time)
    case_when(hr >= 0 & hr < 6 ~ 'NIGHT',
        hr >= 6 & hr < 12 ~ 'MORNING',
        hr >= 12 & hr < 18 ~ 'AFTERNOON',
        TRUE ~ 'EVENING')
}

# Create new column
ny_shootings$OCCUR_TIMEDAY = time_of_day(ny_shootings$OCCUR_TIME)
```

5.2 - Add a Column to Represent the Number of Victims

- The metadata documentation states that the incident key can be duplicated in the cases when there are multiple victims in a given incident.
- We can use this fact to create a column that represents the number of victims per incident, which could be useful for the analysis later.
- It is also desirable to create a new dataframe to represent the unique incidents, i.e., combine all victims of the same incident into one single occurrence. This could help produce a more accurate analysis.

```
# Create a new dataframe with number of victims per incident
ny_shootings_num_vic = ny_shootings %>% group_by(INCIDENT_KEY) %>%
    count() %>%
    rename('NUM_VIC' = n)

# Add number of victims per incident to NY shootings dataframe
ny_shootings = ny_shootings %>% left_join(ny_shootings_num_vic, by = c('INCIDENT_KEY'))

# New dataframe to represent unique incidents
ny_shootings_unique = ny_shootings[!duplicated(ny_shootings$INCIDENT_KEY), ]

# Print rowcounts
data.frame(c(count(ny_shootings), count(ny_shootings_unique))) %>%
  setNames(c('Total Incidents','Total Unique Incidents')) %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Total Incidents	Total Unique Incidents

Total Incidents	Total Unique Incidents
23568	18562

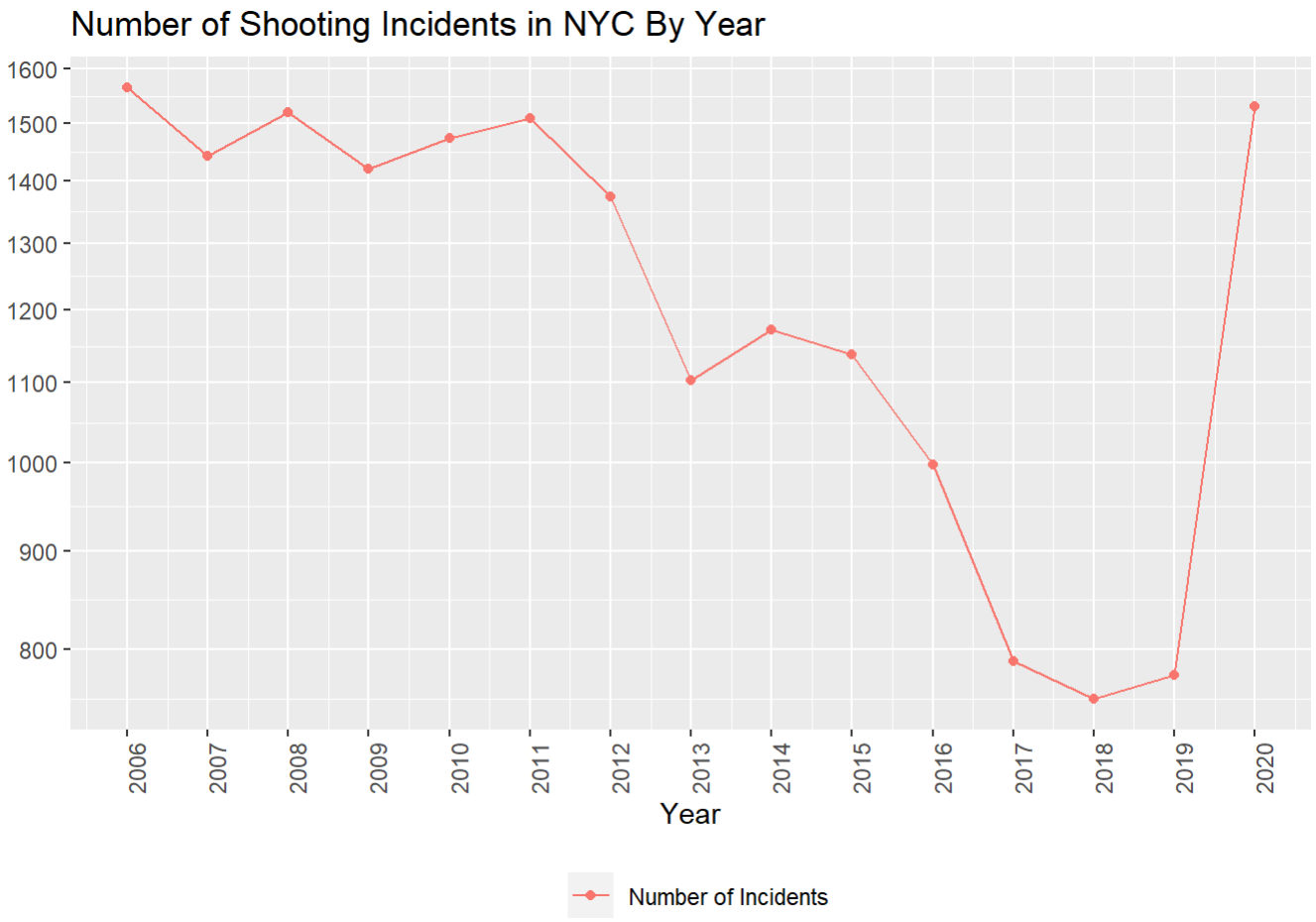
- As observed above, there are many incidents in which multiple victims were involved. Some of the analysis will be desirable to be performed on the dataframe of unique incidents otherwise it could skew the statistics.

6 - Data Visualization and Analysis

6.1 - Number of NYC Shootings By Year

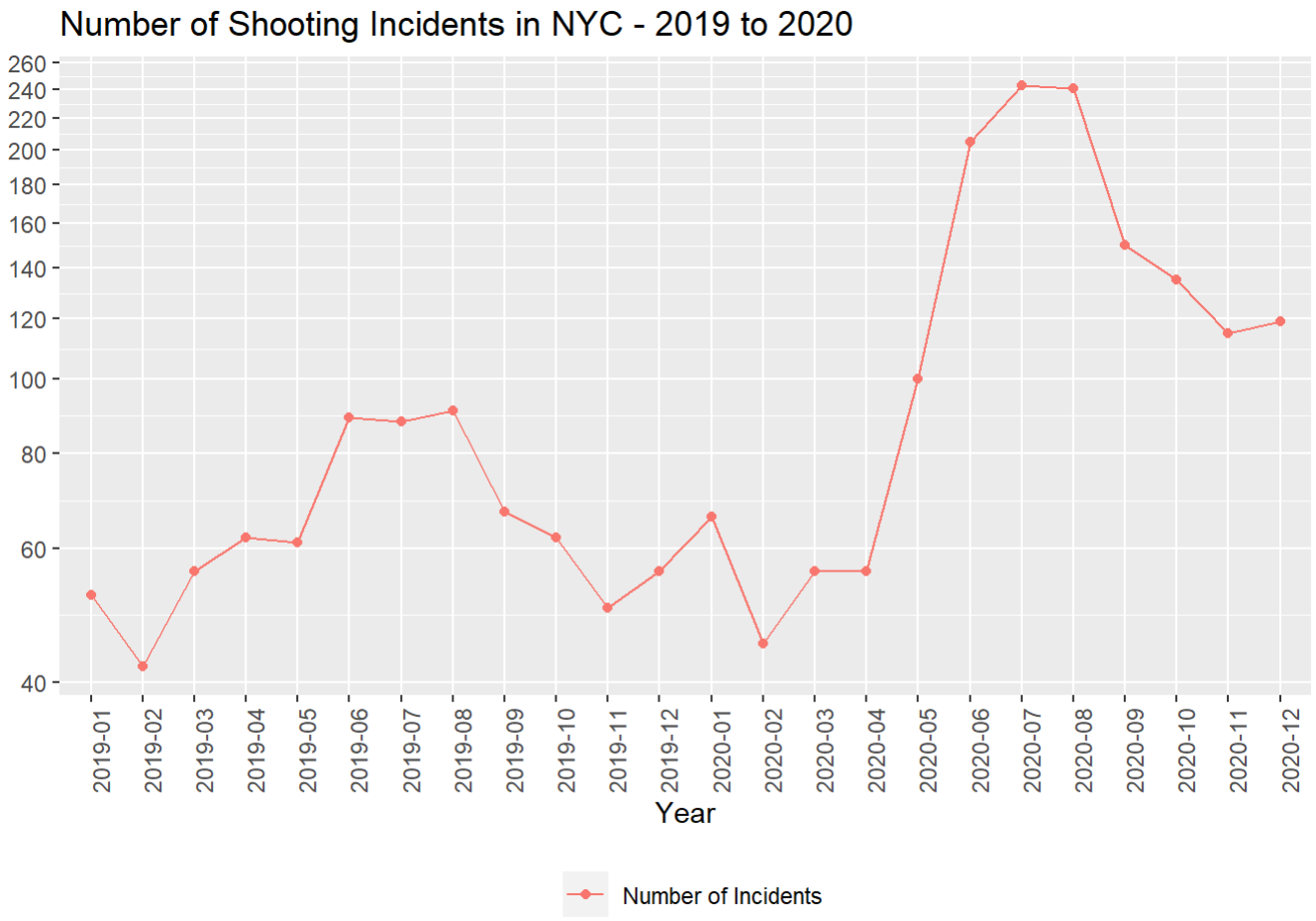
- The first analysis will focus on answering the simple question, “is the number of shootings in NYC increasing over time?”
- Notice that the dataframe of unique incidents will be used for the analysis rather than the original dataframe otherwise incidents with multiple victims would be overcounted (please see 5.2 for details).
- In addition, the Y-axis scale in all charts will be logarithmic in order to have a better sense of the rate of change over time.

```
# Shootings per year
ny_shootings_unique %>% group_by(year(OCCUR_DATE)) %>% count() %>%
    setNames(c('YEAR', 'NUM_INC')) %>%
    ggplot(aes(x = YEAR, y = NUM_INC, group = 1)) +
    geom_line(aes(color = 'NUM_INC')) +
    geom_point(aes(color = 'NUM_INC')) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
    scale_y_log10(breaks = scales::pretty_breaks(n = 10)) +
    scale_colour_discrete(name = '', labels = c('Number of Incidents')) +
    theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
    labs(title = "Number of Shooting Incidents in NYC By Year",
         x= 'Year', y = NULL)
```



- Interesting to notice that the number of shootings/year had been steadily declining over time but there was a significant starting in 2019. One could speculate that the reason might be explained by the COVID-19 lockdown effects, which led to higher unemployment and potentially more criminality.
- Let us zoom in to 2019 - 2020 and check if there's any correlation between higher shooting incidents and the COVID-19 lockdown due to the global pandemic.

```
# Shootings per year - zoomed in to 2019-2020
ny_shootings_unique %>% group_by(format(as.Date(OCCUR_DATE), "%Y-%m")) %>% count() %>%
    setNames(c('YEAR', 'NUM_INC')) %>%
    filter(YEAR >= '2019-01' & YEAR <= '2020-12') %>%
    ggplot(aes(x = YEAR, y = NUM_INC, group = 1)) +
    geom_line(aes(color = 'NUM_INC')) +
    geom_point(aes(color = 'NUM_INC')) +
    scale_y_log10(breaks = scales::pretty_breaks(n = 10)) +
    scale_colour_discrete(name = '', labels = c('Number of Incidents')) +
    theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
    labs(title = "Number of Shooting Incidents in NYC - 2019 to 2020", x= 'Year', y = NULL)
```

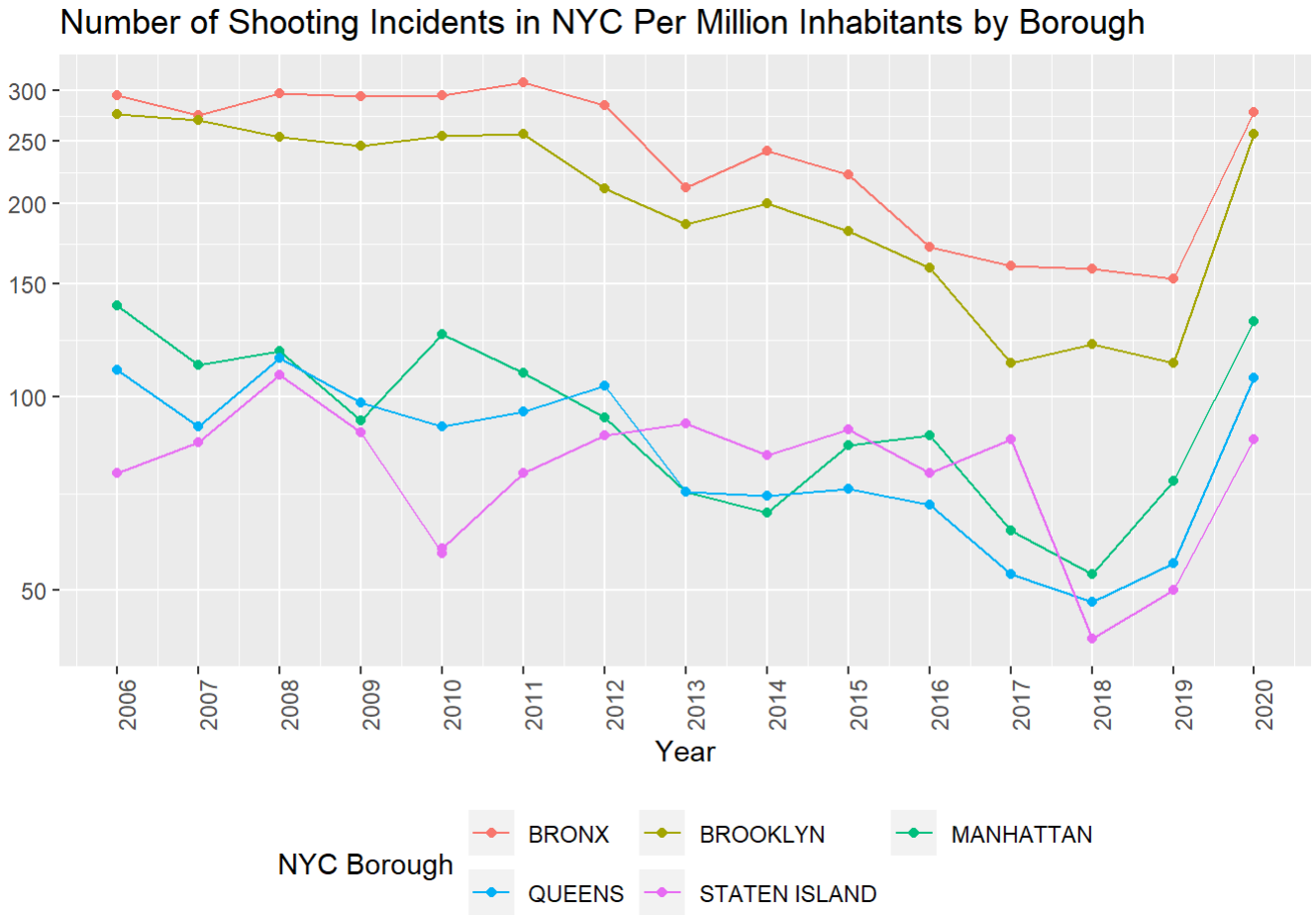


- Indeed, after April 2020 when the general population really started to experience the effects of the COVID-19 lockdown and the consequent job losses, we can identify a clear spike in the number of shootings, which strongly suggests a correlation between the two events.

6.2 - Number of NYC Shootings Per Million Inhabitants - By Year and Borough

- Another interesting analysis would be to explore the number of shooting incidents by year and as a function of NYC boroughs population, i.e. is the change in shooting incidents proportional to the population growth rate?
- This data point could identify which NYC boroughs tend to experience relative more shootings incidents per capita, which in turn could be used as a proxy to determine boroughs with potential higher criminality.

```
# Incidents per million/borough
ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    group_by(YEAR, BORO) %>% count() %>%
    setNames(c('YEAR', 'BORO', 'NUM_INC')) %>%
    left_join(ny_population, by = c('YEAR', 'BORO')) %>%
    mutate(NUM_INC_PER_1MM = round(1000000 * NUM_INC / POPULATION)) %>%
    ggplot(aes(x = YEAR, y = NUM_INC_PER_1MM, group = BORO, color = BORO)) +
    geom_line() + geom_point() +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
    scale_y_log10(breaks = scales::pretty_breaks(n = 10)) +
    scale_colour_discrete(name = 'NYC Borough') +
    theme(legend.position = 'bottom', axis.text.x = element_text(angle = 90)) +
    guides(color = guide_legend(nrow = 2, byrow = TRUE)) +
    labs(title = 'Number of Shooting Incidents in NYC Per Million Inhabitants by Borough', x= 'Yea
r', y = NULL)
```



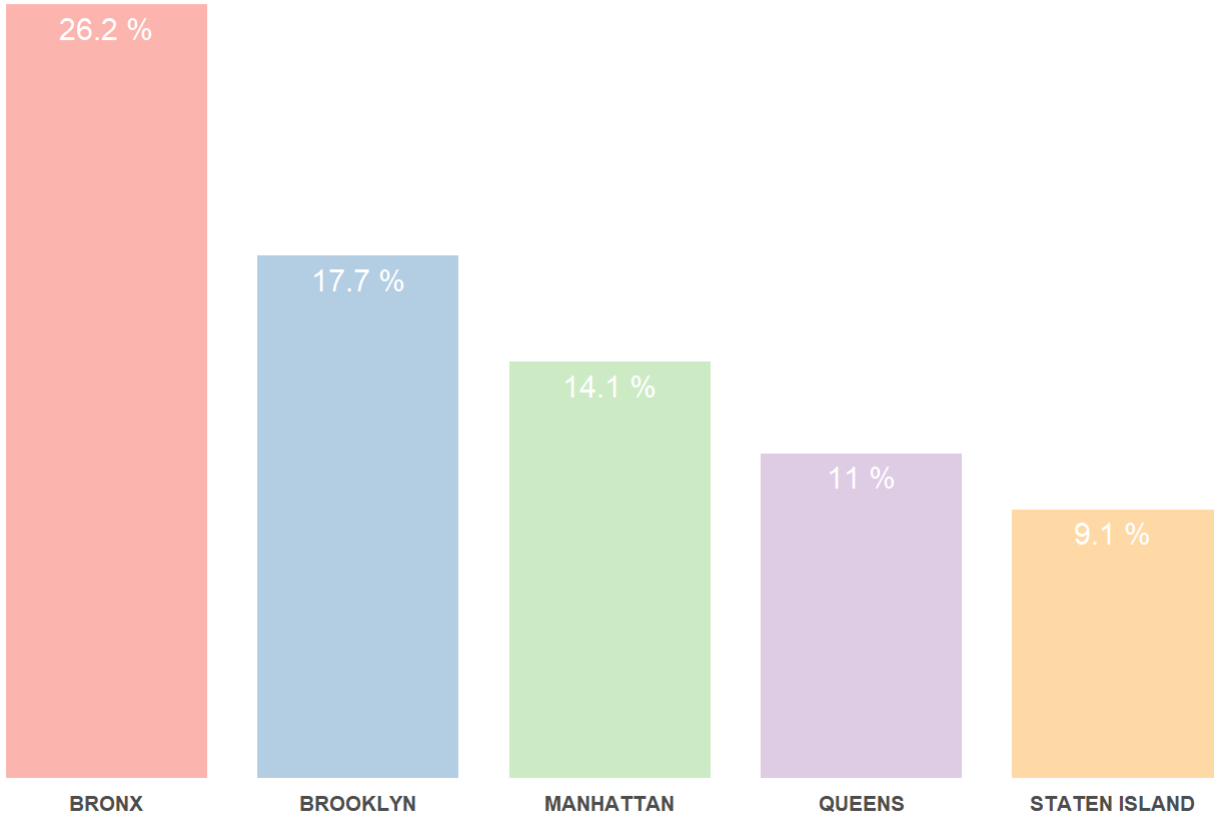

```
# Table - Poverty Rates
ny_poverty %>%
  kbl() %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

BORO	2019_POVERTY_RATES
BRONX	26.2%

BORO	2019_POVERTY_RATES
BROOKLYN	17.7%
MANHATTAN	14.1%
QUEENS	11.0%
STATEN ISLAND	9.1%

```
# Bar chart - Poverty Rates
ny_poverty %>% setNames(c('BORO', 'POVERTY_RATES')) %>%
    mutate(POVERTY_RATES =
             as.numeric(str_replace_all(POVERTY_RATES, '%', ''))) %>%
    ggplot(aes(x = BORO, y = POVERTY_RATES, fill = BORO)) +
    geom_bar(stat = "identity", width= 0.8) +
    scale_fill_brewer(palette = "Pastel1",
                      name = 'Fatality Outcome') +
    geom_text(aes(label = paste(POVERTY_RATES, '%')),
              color = "white", size = 4,
              vjust = 1.5) +
    labs(title = 'Poverty Rates in NYC Boroughs', x= NULL, y = NULL) +
    theme_minimal() +
    theme(
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_text(size = 8, face = 'bold', vjust = 10),
        axis.text.y = element_blank(),
        legend.position = 'none',
    )
```

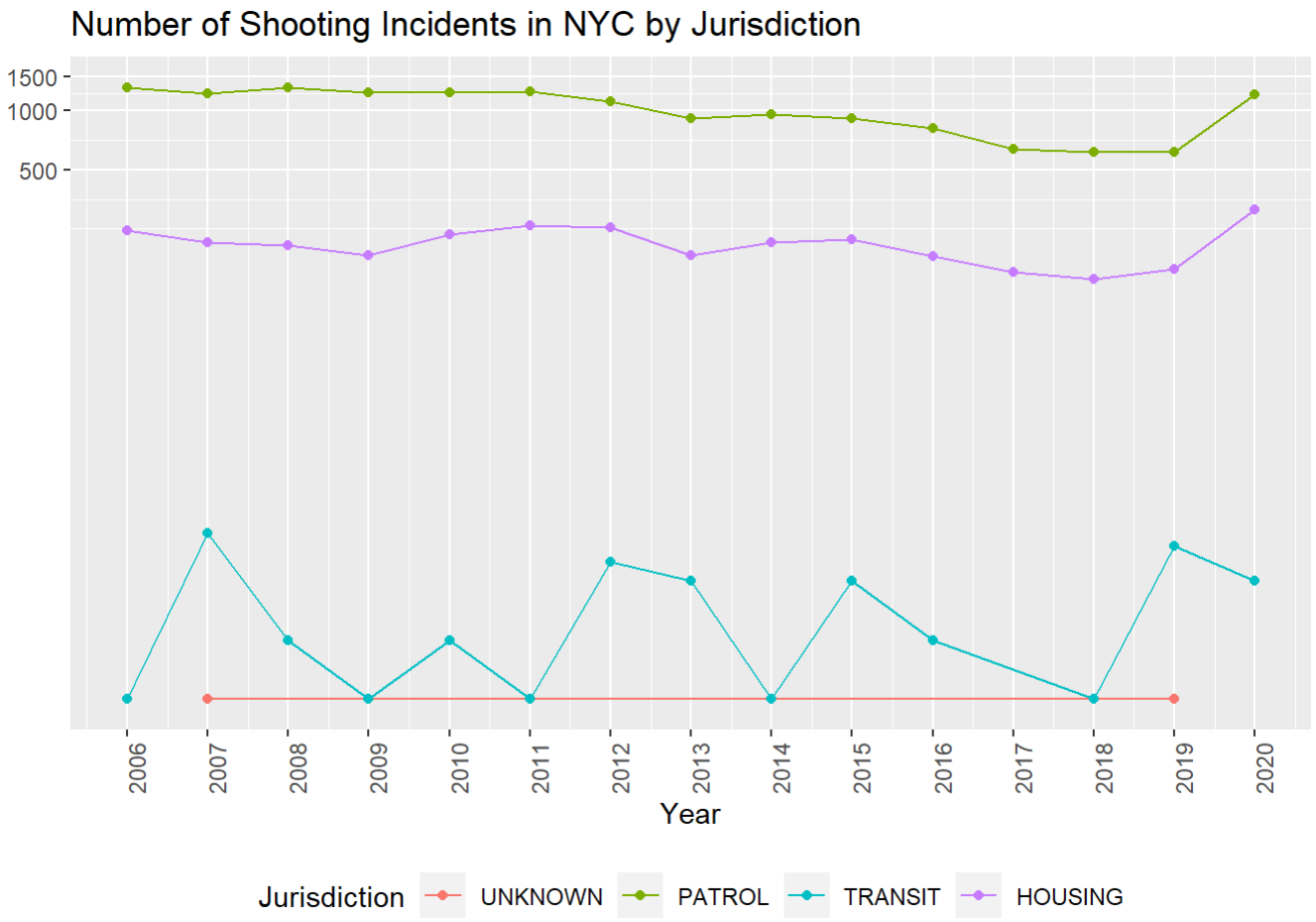
Poverty Rates in NYC Boroughs



- As it can be observed in the graph, the boroughs of Bronx and Brooklyn have a higher number of shootings per million than all other boroughs. Interesting to observe that those two boroughs have the highest poverty rates as of 2019, which can suggest some correlation between shooting incidents and higher poverty rates.
- Another point to note is that all boroughs experienced a sharp increase in the number of shooting incidents per million during the COVID-19 lockdown, perhaps as a result of higher crime due to higher unemployment caused by the lockdown.

6.3 - Number of NYC Shootings By Jurisdiction

```
# Incidents per jurisdiction
ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    group_by(YEAR, JURISDICTION_CODE) %>% count() %>%
    setNames(c('YEAR', 'JURISDICTION', 'NUM_INC')) %>%
    ggplot(aes(x = YEAR, y = NUM_INC, group = JURISDICTION, color = JURISDICTION)) +
    geom_line() + geom_point() +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
    scale_y_log10(breaks = scales::pretty_breaks(n = 6)) +
    scale_colour_discrete(name = 'Jurisdiction') +
    theme(legend.position = 'bottom', axis.text.x = element_text(angle = 90)) +
    guides(color = guide_legend(nrow = 1, byrow = TRUE)) +
    labs(title = 'Number of Shooting Incidents in NYC by Jurisdiction', x= 'Year', y = NULL)
```



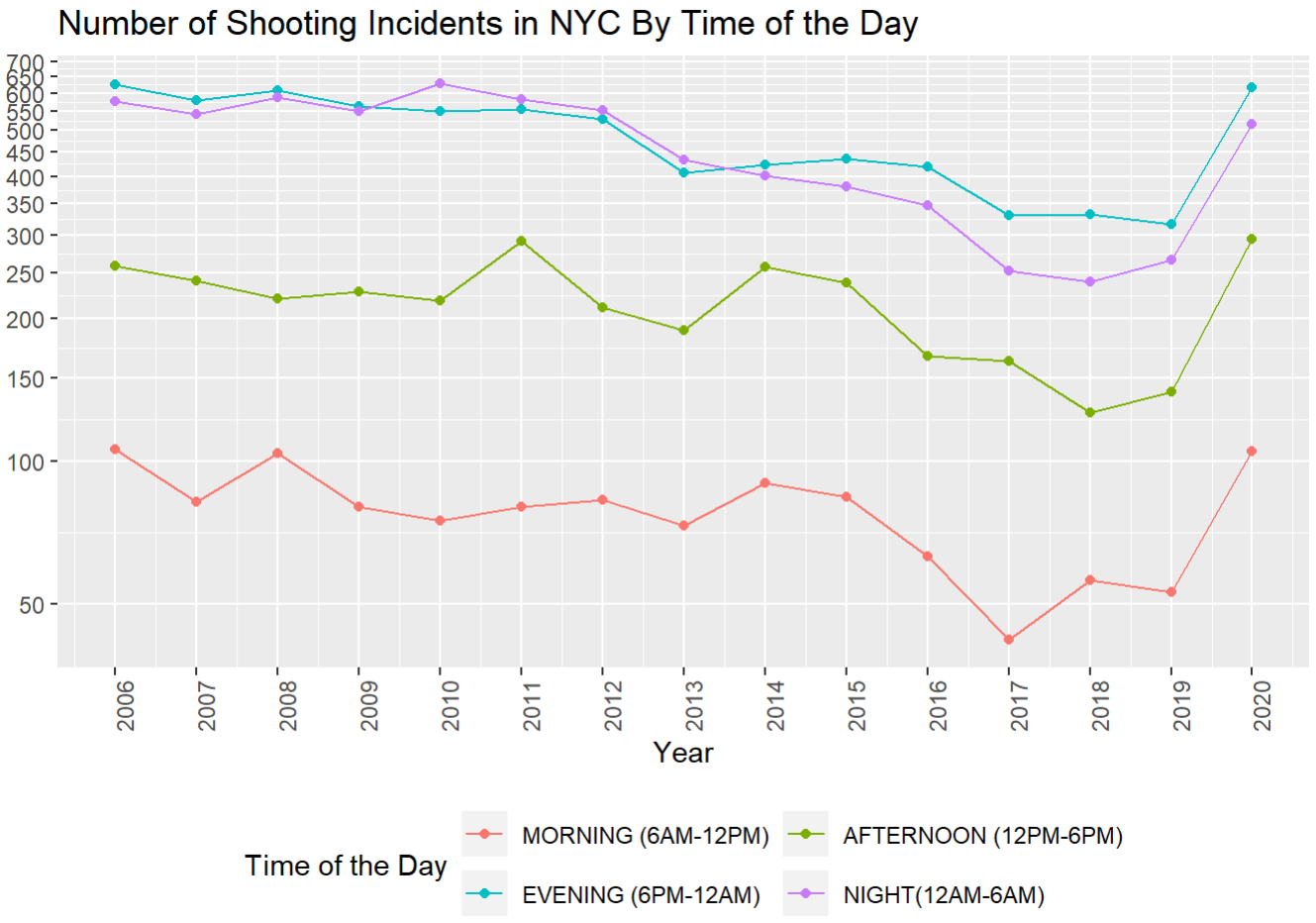
- Not surprisingly, most of the shooting incidents in NYC occur at the “Patrol” jurisdiction, meaning, anywhere other than in housings and in transit.
- Also, interesting to note that the number of shooting incidents in transit during the COVID-19 lockdown actually decreased, likely as a result of the fact that schools and workplaces were closed during the lockdown period.

6.4 - Number of Shooting Incidents in NYC By Time of the Day

- Another interesting question to answer is “which time of the day is more prone to shootings in NYC?”. The general perception would suggest that it would likely be late at night but is it really the case?

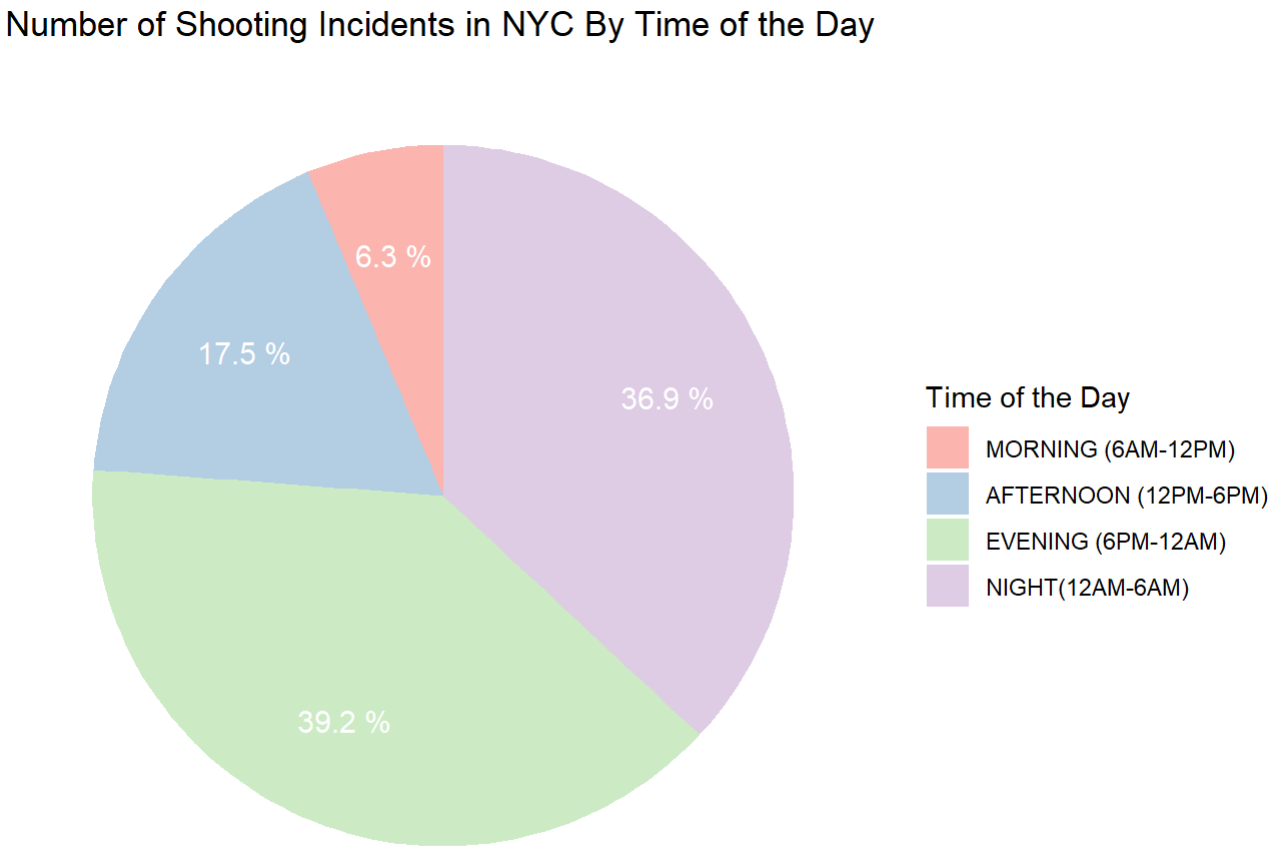
```r
# Incidents by time of the day
df = ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    group_by(YEAR, OCCUR_TIMEDAY) %>% count() %>%
    setNames(c('YEAR', 'OCCUR_TIMEDAY', 'NUM_INC')) %>%
    ungroup() %>%
    mutate(OCCUR_TIMEDAY = factor(OCCUR_TIMEDAY, levels = c('MORNING', 'AFTERNOON', 'EVENING', 'NIGHT')))

# Line chart - Incidents by time of the day
df %>%
    ggplot(aes(x = YEAR, y = NUM_INC, group = OCCUR_TIMEDAY,
               color = OCCUR_TIMEDAY)) +
    geom_line() + geom_point() +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
    scale_y_log10(breaks = scales::pretty_breaks(n = 10)) +
    scale_colour_discrete(name = 'Time of the Day',
                          labels = c('MORNING (6AM-12PM)',
                                     'AFTERNOON (12PM-6PM)',
                                     'EVENING (6PM-12AM)',
                                     'NIGHT(12AM-6AM)')) +
    theme(legend.position = 'bottom', axis.text.x = element_text(angle = 90)) +
    guides(color = guide_legend(nrow = 2, byrow = TRUE)) +
    labs(title = 'Number of Shooting Incidents in NYC By Time of the Day',
         x= 'Year', y = NULL)
```



```
# Summarize incidents by time of the day and compute the avg %
df = df %>% group_by(OCCUR_TIMEDAY) %>%
    summarise(NUM_INC = sum(NUM_INC)) %>%
    mutate(AVG_PCT = round(NUM_INC / sum(NUM_INC) * 100, 1))

# Pie chart - Incidents by time of the day
df %>%
    ggplot(aes(x = '', y = AVG_PCT, fill = OCCUR_TIMEDAY)) +
    geom_bar(width = 1, stat = "identity") +
    coord_polar("y", start = 0) +
    scale_fill_brewer(palette = "Pastel1",
                      name = 'Time of the Day',
                      labels = c('MORNING (6AM-12PM)',
                                 'AFTERNOON (12PM-6PM)',
                                 'EVENING (6PM-12AM)',
                                 'NIGHT(12AM-6AM)')) +
    geom_text(aes(label = paste(AVG_PCT, '%'), x = 1.2),
              color = "white", size = 4,
              position = position_stack(vjust = 0.5)) +
    labs(title = 'Number of Shooting Incidents in NYC By Time of the Day',
         x= NULL, y = NULL) +
    theme_minimal() +
    theme(
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        legend.position = 'right'
    )
```



- As expected, there are significantly more shooting incidents in NYC between the hours of 6PM and 6AM, especially after 12AM.

6.5 - Expected Outcome of Shooting Incidents in NYC

- The dataset documentation states that it only includes shooting incidents in which there were victims involved.
- It would be interesting to investigate the typical number of victims and the likelihood of resulting in a murder in order to determine the expected outcome of a shooting incident in NYC

```
# Dataframe with number of incidents per number of victims
df = ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    mutate(NUM_VIC = cut(NUM_VIC, breaks = c(0, 1, 2, max(NUM_VIC)),
        ordered_result = TRUE, labels = c('1', '2', '> 2'))) %>%
    select(c('YEAR', 'NUM_VIC'))

# Convert to a table and compute percentages
df = as.data.frame.matrix(
  round(prop.table(table(df$YEAR, df$NUM_VIC), 1) * 100, 2))
df %>%
  kbl(caption = 'Number of Victims in Shooting Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Number of Victims in Shooting Incidents in NYC (%)

	1	2	> 2
2006	81.61	13.28	5.11
2007	81.48	13.11	5.41
2008	81.96	12.97	5.07
2009	83.31	11.13	5.56
2010	82.35	12.22	5.43
2011	83.63	11.33	5.04
2012	85.36	10.12	4.52
2013	85.40	11.15	3.45
2014	85.32	11.18	3.50
2015	84.27	10.81	4.92
2016	85.16	10.93	3.91
2017	85.17	10.39	4.44
2018	84.35	11.67	3.98
2019	84.66	11.60	3.74
2020	83.87	11.95	4.18

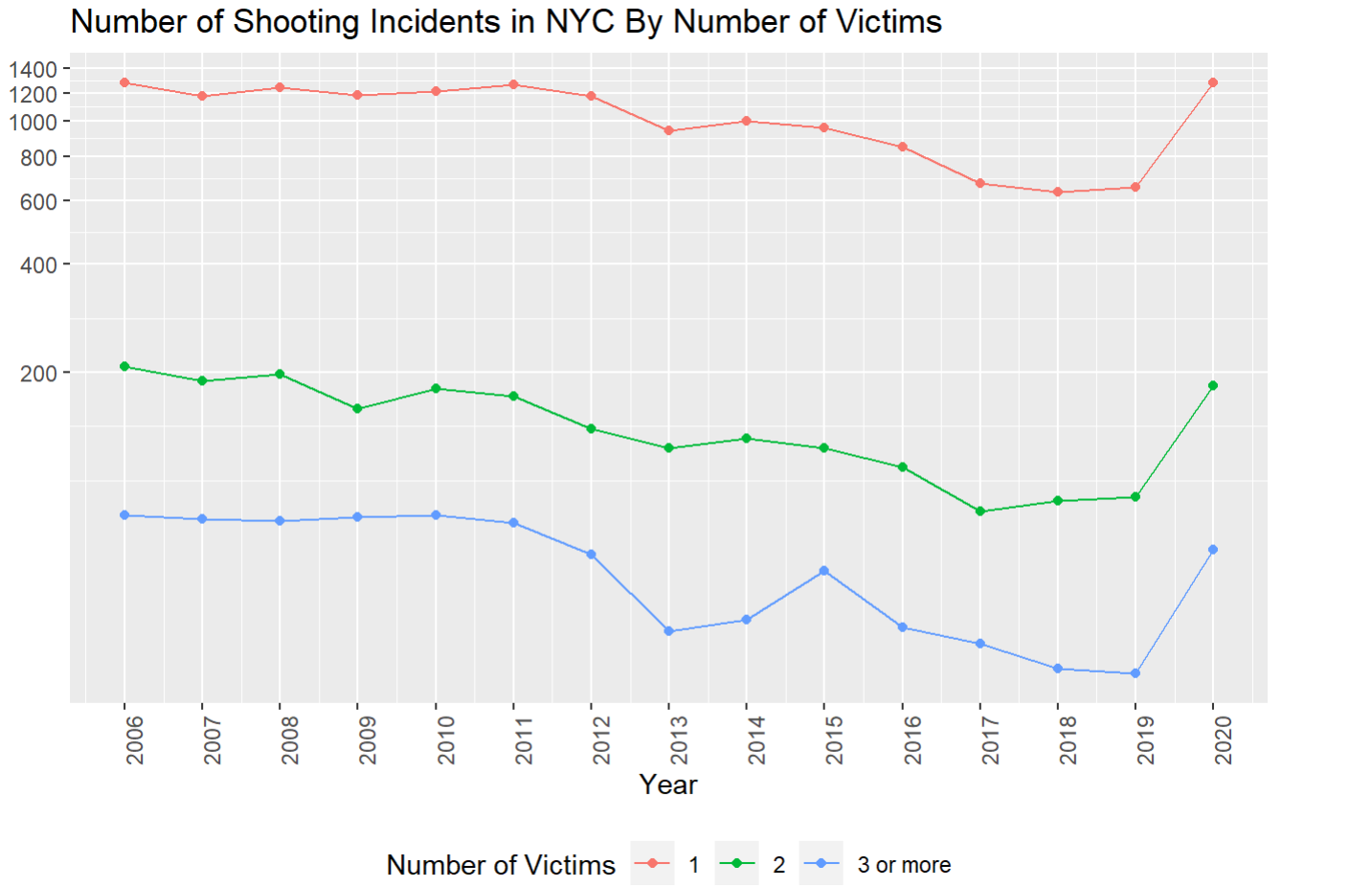
```
# Compute the mean
as.data.frame(round(colMeans(df), 2)) %>%
  setNames(c('Average (%)')) %>%
  kbl(caption = 'Average Number of Victims in Shootings Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Average Number of Victims in Shootings Incidents in NYC (%)

	Average (%)
1	83.86
2	11.59
> 2	4.55

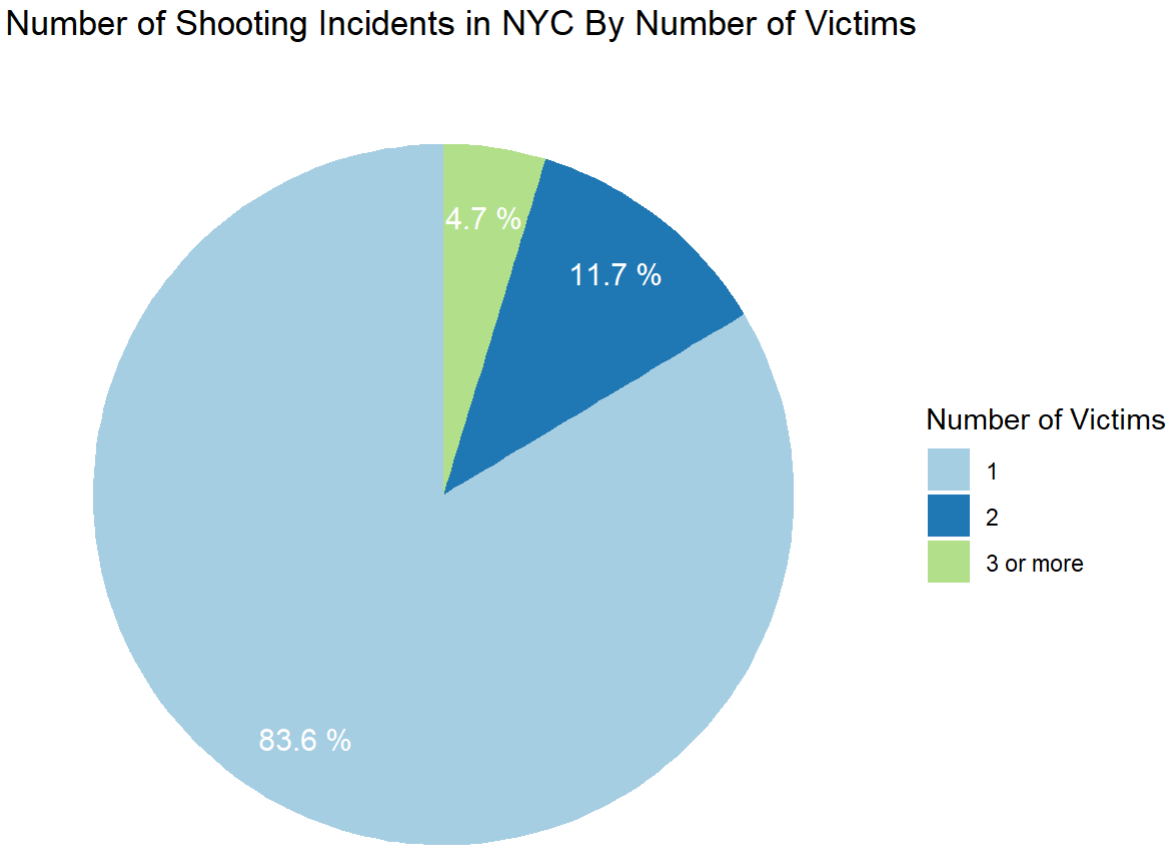
```r
# Incidents by number of victims
df = ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    mutate(NUM_VIC = cut(NUM_VIC, breaks = c(0, 1, 2, max(NUM_VIC)),
        ordered_result = TRUE, labels = c('1', '2', '3 or more'))) %>%
    group_by(YEAR, NUM_VIC) %>% count() %>%
    setNames(c('YEAR', 'NUM_VIC', 'NUM_INC'))

# Line chart
df %>%
    ggplot(aes(x = YEAR, y = NUM_INC, group = NUM_VIC, color = NUM_VIC)) +
    geom_line() + geom_point() +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
    scale_y_log10(breaks = scales::pretty_breaks(n = 10)) +
    scale_colour_discrete(name = 'Number of Victims') +
    theme(legend.position = 'bottom', axis.text.x = element_text(angle = 90)) +
    labs(title = 'Number of Shooting Incidents in NYC By Number of Victims', x= 'Year', y = NULL)
```



```r
# Summarize incidents by number of victims and compute the avg %
df = df %>% group_by(NUM_VIC) %>%
    summarise(NUM_INC = sum(NUM_INC)) %>%
    mutate(AVG_PCT = round(NUM_INC / sum(NUM_INC) * 100, 1))

# Pie chart
df %>%
    ggplot(aes(x = '', y = AVG_PCT, fill = NUM_VIC)) +
    geom_bar(width = 1, stat = "identity") +
    coord_polar("y", start = 0) +
    scale_fill_brewer(palette = "Paired",
                      name = 'Number of Victims') +
    geom_text(aes(label = paste(AVG_PCT, '%'), x = 1.3), color = "white", size = 4,
              position = position_stack(vjust = 0.5)) +
    labs(title = 'Number of Shooting Incidents in NYC By Number of Victims', x= NULL, y = NULL) +
    theme_minimal() +
    theme(
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        legend.position = 'right',
    )
```



```
# Dataframe with fatality outcome of shooting incidents
df = ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    select(c('YEAR', 'STATISTICAL_MURDER_FLAG'))

# Convert to a table and compute percentages
df = as.data.frame.matrix(
  round(prop.table(
    table(df$YEAR, df$STATISTICAL_MURDER_FLAG), 1) * 100, 2))
df %>%
  kbl(caption =
        'Fatality Outcome of Shooting Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Fatality Outcome of Shooting Incidents in NYC (%)

	FALSE	TRUE
2006	80.01	19.99
2007	80.37	19.63
2008	82.42	17.58
2009	82.18	17.82
2010	81.19	18.81
2011	82.77	17.23
2012	84.63	15.37
2013	84.86	15.14
2014	86.01	13.99
2015	82.43	17.57
2016	82.15	17.85
2017	83.90	16.10
2018	81.70	18.30
2019	83.12	16.88
2020	83.08	16.92

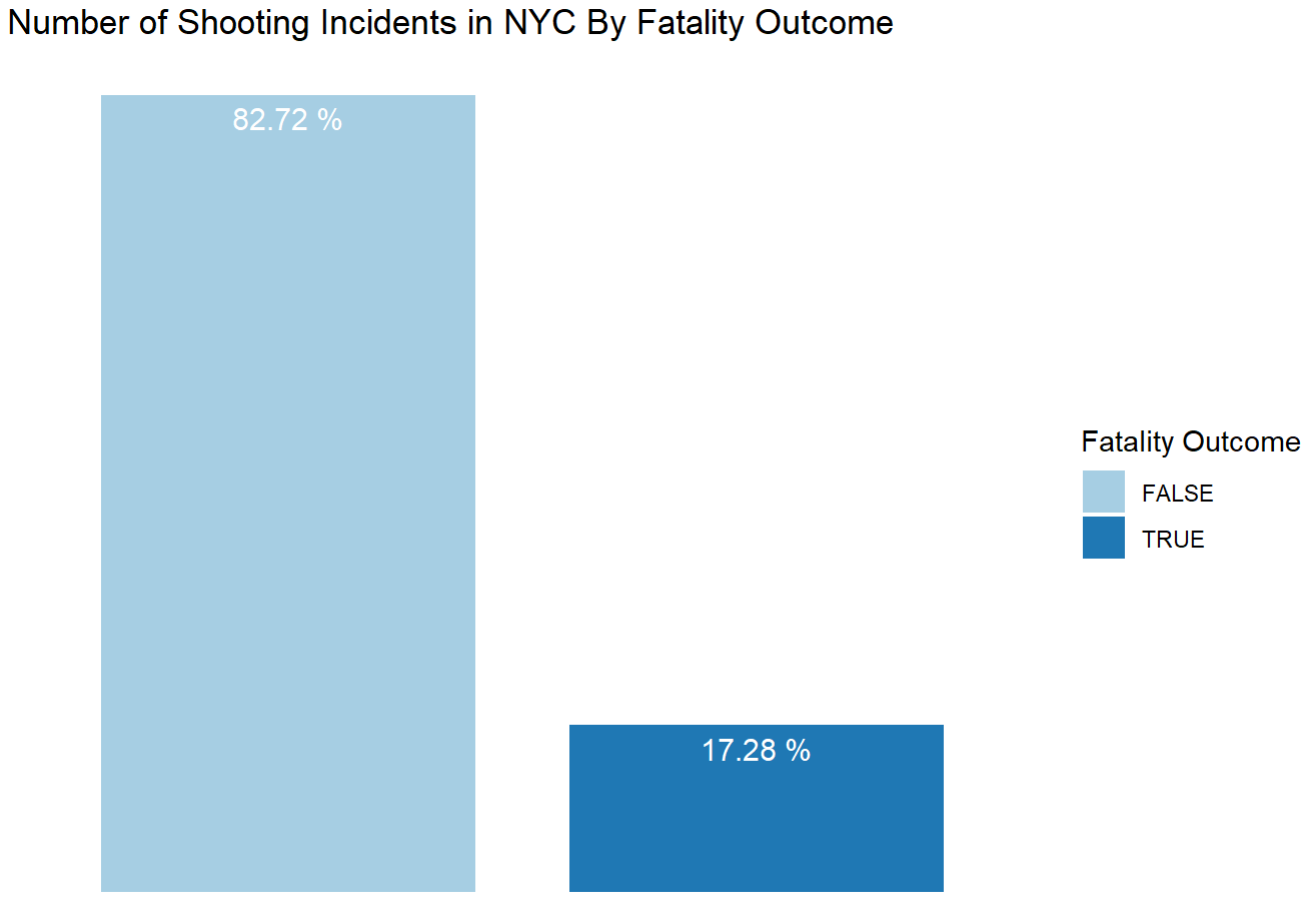
```
# Compute the mean
df = as.data.frame(round(colMeans(df), 2)) %>%
  setNames(c('Average (%)'))
df %>%
  kbl(caption =
        'Average Fatality Outcome of Shootings Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Average Fatality Outcome of Shootings Incidents in NYC (%)

	Average (%)
FALSE	82.72
TRUE	17.28

```
# Reorder columns
df = df %>% setNames(c('AVG_PCT')) %>%
    mutate(MURDER = rownames(df))
df = df[, c('MURDER', 'AVG_PCT')]

# Bar chart - fatality outcome of shooting incidents
df %>%
    ggplot(aes(x = MURDER, y = AVG_PCT, fill = MURDER)) +
    geom_bar(stat = "identity", width= 0.8) +
    scale_fill_brewer(palette = "Paired",
                      name = 'Fatality Outcome') +
    geom_text(aes(label = paste(AVG_PCT, '%')), color = "white", size = 4,
              vjust = 1.5) +
    labs(title = 'Number of Shooting Incidents in NYC By Fatality Outcome', x= NULL, y = NULL) +
    theme_minimal() +
    theme(
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        legend.position = 'right',
    )
```



- As observed in the data, the majority of shooting incidents that involve victims in NYC, results in only one victim (83.86%) and generally the victim survives the incident (82.72%).

6.6 - Typical Profile of the Perpetrator of Shooting Incidents in NYC

- The data also allows to explore the age and sex of a typical perpetrators of shooting incidents in NYC.

```
# Dataframe with age group of the perpetrator
df = ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    select(c('YEAR', 'PERP_AGE_GROUP'))

# Convert to a table and compute percentages
df = as.data.frame.matrix(
  round(prop.table(table(df$YEAR, df$PERP_AGE_GROUP), 1) * 100, 2))
df %>%
  kbl(caption =
        'Age of Perpetrator of Shooting Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Age of Perpetrator of Shooting Incidents in NYC (%)

	<18	18-24	25-44	45-64	65+	UNKNOWN
2006	5.68	21.46	18.97	1.21	0.13	52.55

	<18	18-24	25-44	45-64	65+	UNKNOWN
2007	6.38	23.09	17.34	1.25	0.21	51.73
2008	7.50	26.99	16.13	1.78	0.20	47.40
2009	4.72	22.61	17.54	1.41	0.14	53.59
2010	5.16	20.30	13.51	1.49	0.20	59.33
2011	5.70	20.87	12.39	1.79	0.07	59.18
2012	4.30	19.52	15.80	1.38	0.44	58.56
2013	3.81	23.03	17.14	1.63	0.18	54.22
2014	4.86	21.08	17.24	1.45	0.26	55.12
2015	5.01	20.74	20.39	1.49	0.35	52.02
2016	4.11	20.16	22.37	2.51	0.10	50.75
2017	4.69	21.29	25.98	2.66	0.25	45.12
2018	4.64	17.11	23.34	2.52	0.13	52.25
2019	4.90	17.53	23.32	2.84	0.39	51.03
2020	2.61	12.02	18.42	3.20	0.20	63.55

```
# Compute the mean
df = as.data.frame(round(colMeans(df), 2)) %>% setNames(c('Average (%)'))
df %>%
  kbl(caption =
        'Average Age of Perpetrator of Shootings Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

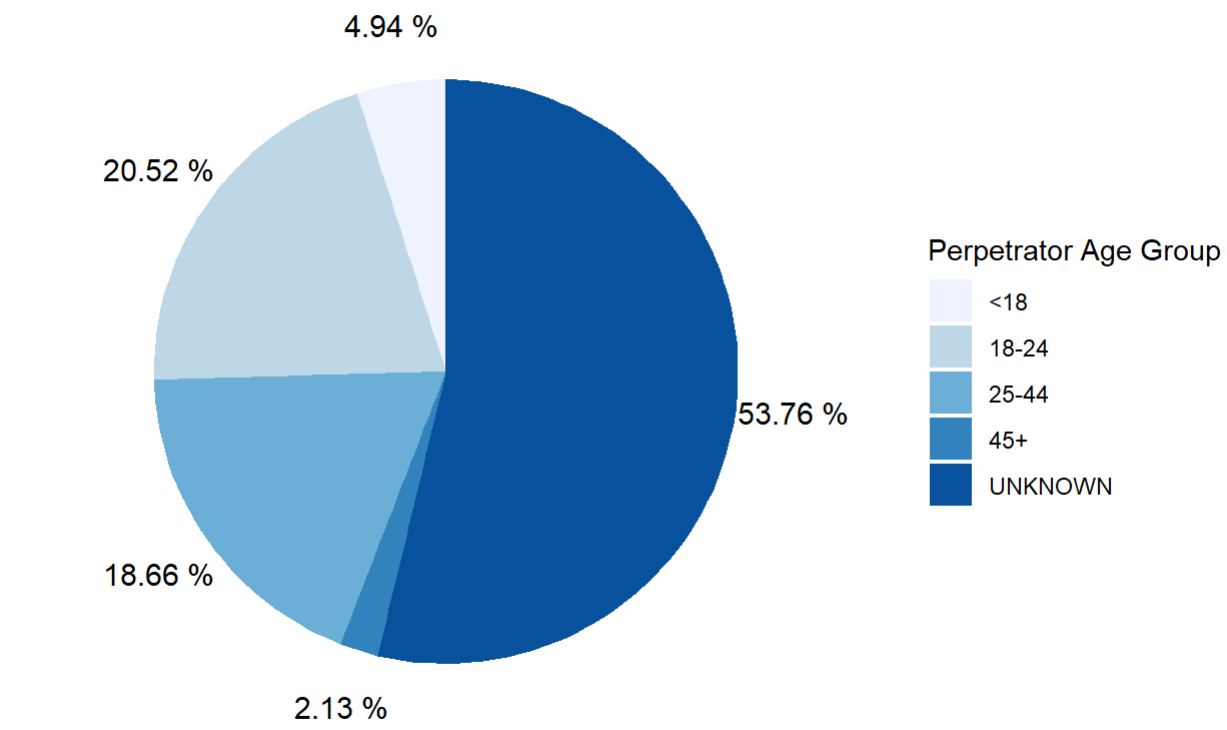
Average Age of Perpetrator of Shootings Incidents in NYC (%)

	Average (%)
<18	4.94
18-24	20.52
25-44	18.66
45-64	1.91
65+	0.22
UNKNOWN	53.76

```
# Consolidate 65+ age group with 45+ for cleanliness
df['45-64', ] = df['45-64', ] + df['65+', ]
df['65+', ] = NA
rownames(df)[rownames(df) == '45-64'] = '45+'
df = na.omit(df)

# Reorder columns
df = df %>% setNames(c('AVG_PCT')) %>%
    mutate(PERP_AGE_GROUP = rownames(df))
df = df[, c('PERP_AGE_GROUP', 'AVG_PCT')]

# Pie chart - age group of the perpetrator
df %>%
    ggplot(aes(x = '', y = AVG_PCT, fill = PERP_AGE_GROUP)) +
    geom_bar(width = 1, stat = "identity") +
    coord_polar("y", start = 0) +
    scale_fill_brewer(palette = "Blues",
                      name = 'Perpetrator Age Group') +
    geom_text(aes(label = paste(AVG_PCT, '%'), x = 1.7),
              color = "black", size = 4,
              position = position_stack(vjust = 0.5)) +
    labs(title =
           'Number of Shooting Incidents in NYC By Age Group of the Perpertrator', x= NULL, y = NU
LL) +
    theme_minimal() +
    theme(
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        legend.position = 'right'
    )
```




```
# Dataframe with sex of the perpetrator
df = ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    select(c('YEAR', 'PERP_SEX'))

# Convert to a table and compute percentages
df = as.data.frame.matrix(
  round(prop.table(table(df$YEAR, df$PERP_SEX), 1) * 100, 2))
df %>%
  kbl(caption =
        'Sex of Perpetrator of Shooting Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Sex of Perpetrator of Shooting Incidents in NYC (%)

	F	M	UNKNOWN
2006	0.89	73.82	25.29
2007	0.83	71.64	27.53
2008	1.25	69.12	29.62
2009	0.77	57.11	42.11
2010	0.81	51.32	47.86

	F	M	UNKNOWN
2011	0.99	41.75	57.26
2012	0.36	41.44	58.19
2013	1.45	45.06	53.49
2014	0.85	44.03	55.12
2015	0.97	47.10	51.93
2016	0.80	48.55	50.65
2017	1.39	53.49	45.12
2018	1.33	46.68	51.99
2019	0.90	47.55	51.55
2020	1.18	35.21	63.62

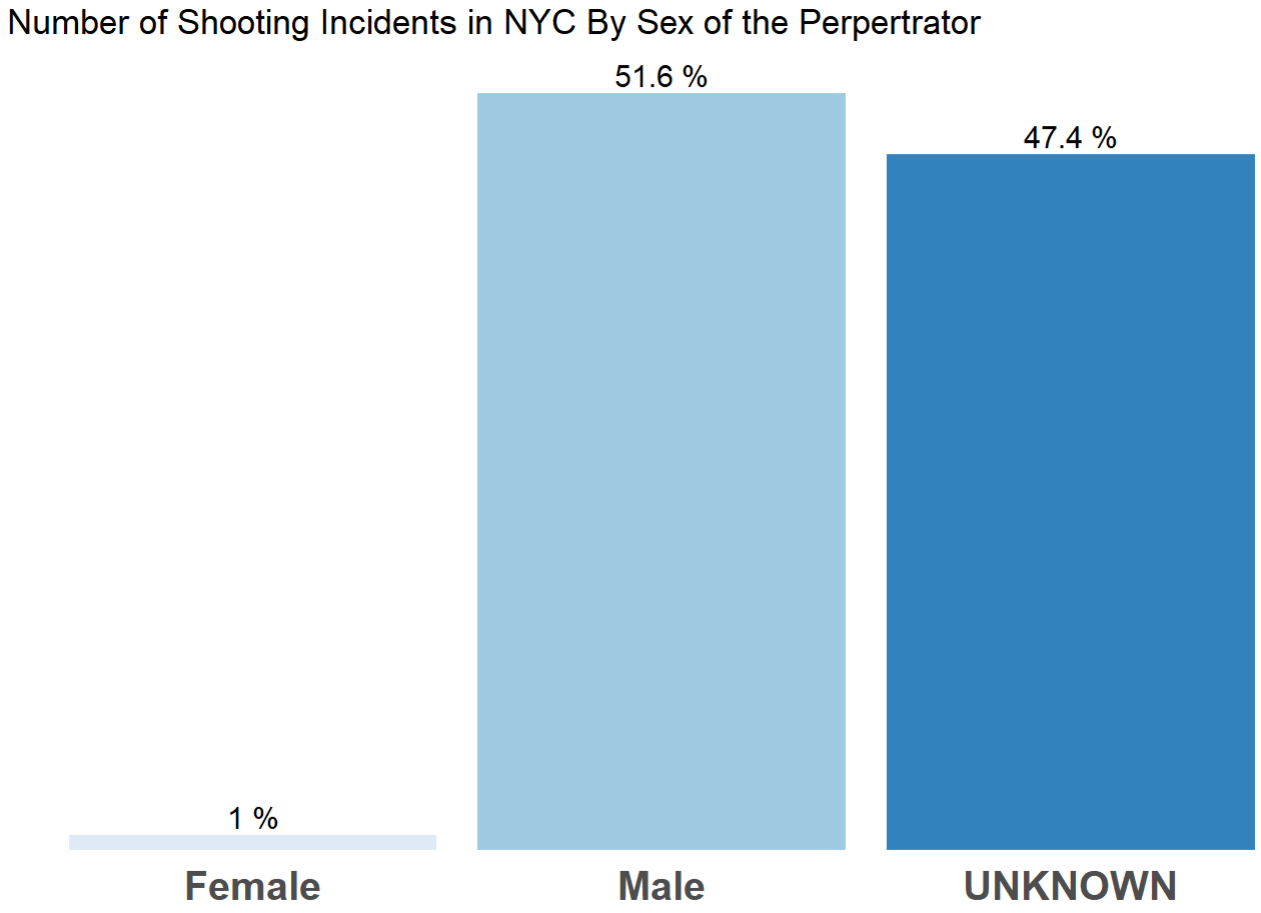
```
# Compute the mean
df = as.data.frame(round(colMeans(df), 1)) %>%
  setNames(c('Average (%)'))
df %>%
  kbl(caption =
        'Sex of Perpetrator of Shootings Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Sex of Perpetrator of Shootings Incidents in NYC (%)

	Average (%)
F	1.0
M	51.6
UNKNOWN	47.4

```r
# Reorder columns
df = df %>% setNames(c('AVG_PCT')) %>%
    mutate(PERP_SEX = rownames(df))
df = df[, c('PERP_SEX', 'AVG_PCT')]

# Bar chart
df %>%
    ggplot(aes(x = c('Female', 'Male', 'UNKNOWN'), y = AVG_PCT, fill = PERP_SEX)) +
    geom_bar(stat = "identity", width= 0.9) +
    scale_fill_brewer(palette = "Blues",
                      name = 'Sex of the Perpetrator') +
    geom_text(aes(label = paste(AVG_PCT, '%')),
              color = "black", size = 4,
              vjust = -0.3) +
    labs(title =
           'Number of Shooting Incidents in NYC By Sex of the Perpertrator',
         x = NULL, y = NULL) +
    theme_minimal() +
    theme(
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_text(size = 15, face = 'bold', vjust = 5),
        axis.text.y = element_blank(),
        legend.position = 'none',
    )
```



- As observed in the data, in the majority of the shooting incidents in NYC the age group of the perpetrator is unknown (53.76%), which suggests that most of the incidents go unresolved, i.e. the perpetrator is not apprehended.
- Out of known cases, the majority of perpetrators are in the age group of 18 to 44 years old (39.18%) and male (47.42%).

6.7 - Typical Profile of the Victims of Shooting Incidents in NYC

- Finally, the data also allows to explore the age and sex of a typical victims of shooting incidents in NYC.

```
# Dataframe with age group of the victim
df = ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    select(c('YEAR', 'VIC_AGE_GROUP'))

# Convert to a table and compute percentages
df = as.data.frame.matrix(
  round(prop.table(table(df$YEAR, df$VIC_AGE_GROUP), 1) * 100, 2))
df %>%
  kbl(caption = 'Age of Victims of Shooting Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Age of Victims of Shooting Incidents in NYC (%)

	<18	18-24	25-44	45-64	65+	UNKNOWN

	<18	18-24	25-44	45-64	65+	UNKNOWN
2006	11.88	42.34	39.72	5.30	0.57	0.19
2007	13.31	39.25	40.78	5.69	0.62	0.35
2008	13.23	38.51	42.20	5.33	0.53	0.20
2009	11.41	40.99	41.55	4.93	0.99	0.14
2010	11.88	39.99	42.09	5.16	0.75	0.14
2011	12.79	42.28	38.44	5.70	0.53	0.27
2012	9.54	40.28	44.57	4.95	0.44	0.22
2013	8.61	38.89	44.15	7.34	0.73	0.27
2014	9.30	42.06	40.10	7.42	0.77	0.34
2015	7.82	39.02	45.96	6.68	0.44	0.09
2016	5.22	38.62	48.24	7.32	0.50	0.10
2017	7.60	33.84	50.19	7.86	0.51	0.00
2018	6.90	31.43	51.86	8.62	0.93	0.27
2019	6.96	28.35	54.64	9.15	0.64	0.26
2020	6.14	29.52	55.91	7.45	0.59	0.39

```
# Compute the mean
df = as.data.frame(round(colMeans(df), 2)) %>%
  setNames(c('Average (%)'))
df %>%
  kbl(caption = 'Average Age of Victims of Shootings Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

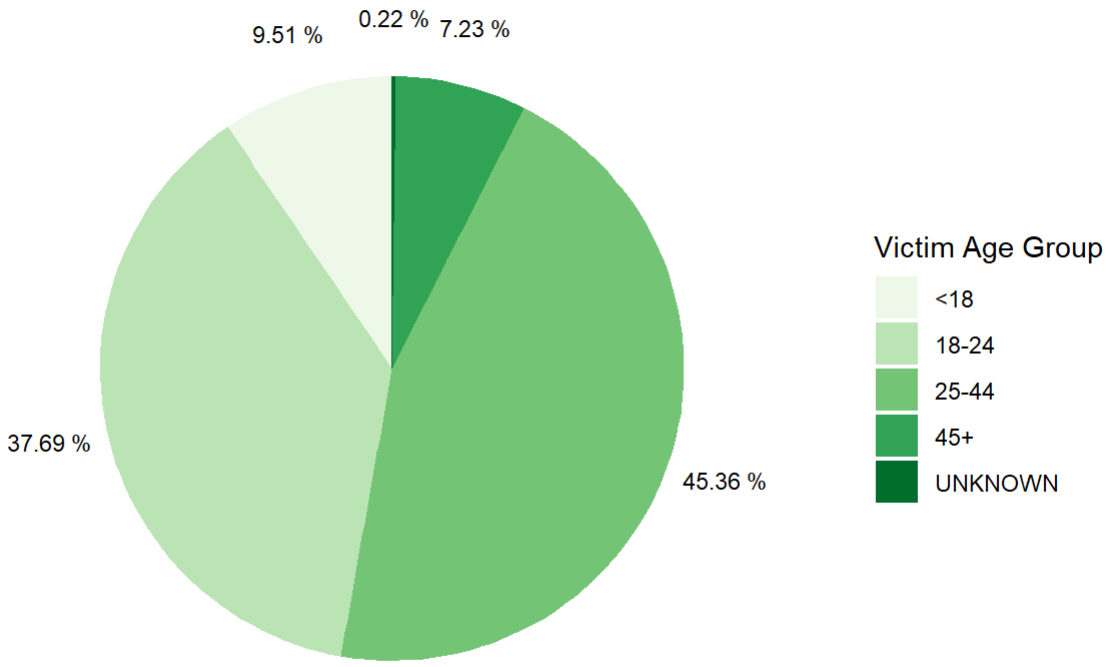
Average Age of Victims of Shootings Incidents in NYC (%)

	Average (%)
<18	9.51
18-24	37.69
25-44	45.36
45-64	6.59
65+	0.64
UNKNOWN	0.22

```
# Consolidate 65+ age group with 45+
df['45-64', ] = df['45-64', ] + df['65+', ]
df['65+', ] = NA
rownames(df)[rownames(df) == '45-64'] = '45+'
df = na.omit(df)

# Reorder columns
df = df %>% setNames(c('AVG_PCT')) %>%
    mutate(VIC_AGE_GROUP = rownames(df))
df = df[, c('VIC_AGE_GROUP', 'AVG_PCT')]

# Pie chart
df %>%
    ggplot(aes(x = '', y = AVG_PCT, fill = VIC_AGE_GROUP)) +
    geom_bar(width = 1, stat = "identity") +
    coord_polar("y", start = 0) +
    scale_fill_brewer(palette = "Greens",
                      name = 'Victim Age Group') +
    geom_text(aes(label = paste(AVG_PCT, '%'), x = 1.7), color = "black", size = 3,
              position = position_stack(vjust = 0.5)) +
    labs(title = 'Number of Shooting Incidents in NYC By Age Group of the Victim', x= NULL, y = NU
LL) +
    theme_minimal() +
    theme(
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        legend.position = 'right'
    )
```




```
# Dataframe with sex of the victims
df = ny_shootings_unique %>% mutate(YEAR = year(OCCUR_DATE)) %>%
    select(c('YEAR', 'VIC_SEX'))

# Convert to a table and compute percentages
df = as.data.frame.matrix(
  round(prop.table(table(df$YEAR, df$VIC_SEX), 1) * 100, 2))
df %>%
  kbl(caption = 'Age of Victims of Shooting Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Age of Victims of Shooting Incidents in NYC (%)

	F	M	UNKNOWN
2006	7.28	92.72	0.00
2007	7.21	92.58	0.21
2008	6.78	93.22	0.00
2009	8.38	91.48	0.14
2010	7.06	92.94	0.00

	F	M	UNKNOWN
2011	8.28	91.72	0.00
2012	7.28	92.72	0.00
2013	5.80	94.11	0.09
2014	6.31	93.69	0.00
2015	7.56	92.44	0.00
2016	8.53	91.37	0.10
2017	7.48	92.52	0.00
2018	7.16	92.71	0.13
2019	9.15	90.72	0.13
2020	8.82	90.73	0.46

```
# Compute the mean
as.data.frame(round(colMeans(df), 2)) %>%
  setNames(c('Average (%)')) %>%
  kbl(caption = 'Average Age of Victims of Shootings Incidents in NYC (%)') %>%
    kable_styling(bootstrap_options = c("striped", "condensed"),
                  full_width = FALSE, position = "left", font_size = 12)
```

Average Age of Victims of Shootings Incidents in NYC (%)

	Average (%)
F	7.54
M	92.38
UNKNOWN	0.08

- In the vast majority of the shooting incidents in NYC, the victims are in the age group of 18 to 44 years old (83.35%) and male (92.38%).

7 - Conclusions

- We can now summarize the conclusions of all the analysis performed.

  1 - The number of shooting incidents in NYC had been decreasing steadily since 2006 until April of 2020, when a significant spike was observed.

  2 - The spike observed in April 2020, coincides with the COVID-19 lockdown situation, which probably suggests that the increase in number of shootings was the result of the higher number of unemployment caused by the economic impact of the lockdown.

  3 - The boroughs of Bronx and Brooklyn are the areas with the highest number of incidents per million inhabitants. These two areas are also the boroughs that experience higher rates of poverty, which suggests a possible correlation between the two data points.

4 - Most of the shooting incidents in NYC occur anywhere outside housings and transit. During the COVID-19 lockdown, there was an observable decrease in the number of incidents in transit, likely as a result of the fact that schools and workplaces were closed during that period and thus less commuting.

5 - Consistently with the general perception, most of the shooting incidents happen between the hours of 6PM and 6AM, especially after 12AM.

6 - Most of the shooting incidents with victims result in only one victim (approx 83%) and generally the victim is expected to survive (approx 82% of the time).

7 - The perpetrator of shooting incidents in NYC is generally not apprehended since their age and sex is usually unknown (>50%). Out of the known cases, the majority of perpetrators are in the age group of 18 to 44 years old (approx 39%) and male (approx 47%).

8 - In the vast majority of the shooting incidents in NYC, the victims are in the age group of 18 to 44 years old (approx 83%) and male (approx 92%).

Session info

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.1252  LC_CTYPE=English_Canada.1252
## [3] LC_MONETARY=English_Canada.1252 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.1252
##
## attached base packages:
## [1] graphics  grDevices datasets  utils     stats     methods   base
##
## other attached packages:
##  [1] kableExtra_1.3.4 lubridate_1.7.10 forcats_0.5.1    stringr_1.4.0
##  [5] dplyr_1.0.7      purrr_0.3.4      readr_2.0.0      tidyr_1.1.3
##  [9] tibble_3.1.2     ggplot2_3.3.5    tidyverse_1.3.1  gtools_3.9.2
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7         svglite_2.0.0      assertthat_0.2.1   digest_0.6.27
##  [5] utf8_1.2.1         R6_2.5.0           cellranger_1.1.0   backports_1.2.1
##  [9] reprex_2.0.0       evaluate_0.14      highr_0.9          httr_1.4.2
## [13] pillar_1.6.1       rlang_0.4.11       curl_4.3.2         readxl_1.3.1
## [17] rstudioapi_0.13    jquerylib_0.1.4    rmarkdown_2.9      labeling_0.4.2
## [21] webshot_0.5.2      bit_4.0.4          munsell_0.5.0      broom_0.7.8
## [25] compiler_4.1.0     modelr_0.1.8       xfun_0.24          pkgconfig_2.0.3
## [29] systemfonts_1.0.2  htmltools_0.5.1.1  tidyselect_1.1.1   fansi_0.5.0
## [33] viridisLite_0.4.0  crayon_1.4.1       tzdb_0.1.2         dbplyr_2.1.1
## [37] withr_2.4.2        grid_4.1.0         jsonlite_1.7.2     gtable_0.3.0
## [41] lifecycle_1.0.0    DBI_1.1.1          magrittr_2.0.1     scales_1.1.1
## [45] cli_3.0.1          stringi_1.6.2      vroom_1.5.3        farver_2.1.0
## [49] fs_1.5.0           xml2_1.3.2         bslib_0.2.5.1      ellipsis_0.3.2
## [53] generics_0.1.0     vctrs_0.3.8        RColorBrewer_1.1-2 tools_4.1.0
## [57] bit64_4.0.5        glue_1.4.2         hms_1.1.0          parallel_4.1.0
## [61] yaml_2.2.1         colorspace_2.0-2   rvest_1.0.0        knitr_1.33
## [65] haven_2.4.1        sass_0.4.0
```