CU-DTSA-530 - NYPD Shooting Incident Report

Rajesh Kutti, Student - University of Colorado, Boulder

2024-02-27

Important Note

- **Libraries Required**: Please plan to install the libraries **(ltidyverse, llubridate, lusmap, lstringr)**
- **File Downloads**: If you happen to run into data file download issues due to network issues, please retry, if not please manually download the file and save it to folder: cached_data under your current directory.
- **PDF Document**: A PDF version of this report is available in GitHub.

Executive Summary

This project is to analyze the **NYPD Shooting Incident data** and explore the nature of the criminal activity in NYC from 2006 to 2022. Gun violence has been one of the most sensitive and debated topics in USA for years since there are differences in opinion between people - sometimes even in the same household. Our goal is to help reduce crime in NYC, so for analyzing and preparing this report, we will be mindful, and we will keep our biases aside and try to look at the real facts in the shooting incident data presented to us by the **catalog.data.gov** website. We will also build a **Quantitative machine learning model** to predict the future shooting incidents which could hopefully help the NYPD.

- We will follow a step by step and iterative Data Science process to analyze and generate insights.
- **Goal** - Analyze, share insights, provide actionable plan to reduce crime in NYC and build a Quantitative model to predict gun violence in NYC Boro's.
- **Benefits** - By exploring the nature of shooting/criminal activity, we can help the Law Enforcement Agencies to minimize crime. We can help the public to stay informed and We can help to Government to create a safe neighborhood.
- **Objectives** - To achieve our goal, we must continue to ask questions (as follows) to gain more insights into the problem until all our objectives are met.
  - How many incidents are happening across New York City?
    - What location and at what time are these incidents more common?
    - How fatal are these incidents?
    - Can we understand the demographics of the perpetrators?
      - What is their race, age group and gender?
    - Can we understand the demographics of the victims?
      - What is their race, age group and gender?
  - Can we narrow down into the top 5 precinct and get some insights?

- Finally, we want to train and test a predictive model to predict future shooting incidents to help law enforcement.
- For those who want a summarized view of this report, please start from **Section 3: Visualizations**. The observations and insights are articulated in that section.

Data Science Process - An iterative approach. Reference:*DTSA-530 by Prof. Jane Wall)*

- We will follow a well-defined process that we learned in DTSA-5301 for doing our analysis. During this process we will document our observations and share insights that will help the team and the key stakeholders.
- **Import data** from catalog.data.gov
- **Tidy data** by removing unwanted rows, then formatting the data so that each row is an observation and finally null imputation or null value replacements.
- **Transform data** - Here we change numbers to categorical variables, rename columns or we add new columns as needed.
- **Analyze and Visualize** - We then go through iterative process of analyzing, visualizing, asking questions then going back to prior steps and repeating the process until we have good understanding of the answers to the questions we started with, (until our goal is met).
- For analyzing, we will do Univariate analysis by looking at distributions for single variable, Bivariate analysis by looking at correlations between two variables and Multivariate analysis by checking relationship between multiple predictors and our target variables.
- **Model** - We will build a quantitative model for our data and observations. During this phase we will try to check if our assumptions are accurate by running the predictions.
- **Communicate** - Finally, we will communicate our results in a reproducible way so that the report can be improved in future or can be leveraged by others to do more analysis.

References:

- University of Colorado, Boulder - DTSA-530
- CSSE Johns Hopkins University
- Coursera, CU Boulder

Section 1A: Common Functions:

- In this section we will add all the common functions used.
- Each function is well-documented.
  - Data can come from various sources, and it could be various types, size and shapes.
  - There could be null values, unknown values, NA, etc. To do good analysis we need to tidy/clean the data.
  - Common and reusable functions will help all the team members (like Airbnb approach for Data Science).

```
# function to format large numbers
formatNumber <- function(x) {
  format( x, digits = 2, scientific=FALSE, big.mark = ","
  )
}
```

```r
# Tidy dataset: Remove unwanted columns - select only the columns needed for analysis.
tdRemoveCols <- function(ds_nyc_sh) {
  cols <- c("INCIDENT_KEY", "OCCUR_DATE", "OCCUR_TIME", "BORO", "LOC_OF_OCCUR_DESC", "PRECINCT", "JURISI
  result <- ds_nyc_sh %>% select(all_of(cols))
  result
}

# Tidy dataset - Fix/substitute nulls with appropriate values
# Lump bad PERP_AGE_GROUP into one category - unknown
# Lump bad VIC_AGE_GROUP into one category - unknown
# Furthermore, move bad/unknown values for PERP_SEX, VIC_SEX, PERP_RACE, LOCATION_DESC, under one categ

tdDatasetFixColValues <- function(ds_nyc_sh) {

  # We will replace all null and unknown into one bucket called unknown
  ds_nyc_sh$PERP_AGE_GROUP[ds_nyc_sh$PERP_AGE_GROUP %in% c("(null)", "1020", "224", "940")] <- "UNKNOWN

  ds_nyc_sh$VIC_AGE_GROUP[ds_nyc_sh$VIC_AGE_GROUP %in% c("1022")] <- "UNKNOWN"

  # Fix PERP_SEX cols
  ds_nyc_sh$PERP_SEX[ds_nyc_sh$PERP_SEX %in% c("(null)", "NA")] <- "U"
  ds_nyc_sh <- ds_nyc_sh  %>% replace_na(list(PERP_SEX="U"))

  # Fix VIC_SEX cols
  ds_nyc_sh$VIC_SEX[ds_nyc_sh$VIC_SEX %in% c("(null)", "NA")] <- "U"
  ds_nyc_sh <- ds_nyc_sh  %>% replace_na(list(VIC_SEX="U"))

  # Fix Race cols
  ds_nyc_sh$PERP_RACE[ds_nyc_sh$PERP_RACE %in% c("(null)", "NA")] <- "UNKNOWN"
  ds_nyc_sh <- ds_nyc_sh  %>% replace_na(list(PERP_RACE="UNKNOWN"))

  # Fix LOCATION_DESC cols
  ds_nyc_sh$LOCATION_DESC[ds_nyc_sh$LOCATION_DESC %in% c("(null)", "NA")] <- "UNKNOWN"
  ds_nyc_sh <- ds_nyc_sh  %>% replace_na(list(LOCATION_DESC="UNKNOWN"))

  # replace all nulls with "unknown
  ds_nyc_sh <- ds_nyc_sh  %>% replace_na(list(PERP_AGE_GROUP="UNKNOWN"))
  ds_nyc_sh
}

# Tidy dataset:
# In this method we will create date type, then year, month, dayofweek columns for our analysis using m
tdAugmentDates <- function(df_nyc_sh) {
  result <- df_nyc_sh %>%
    mutate(
      OCCUR_DATE = mdy(OCCUR_DATE),
      syear = year(OCCUR_DATE),
      smonth = month(OCCUR_DATE, label = TRUE),
      sday_of_week = wday(OCCUR_DATE, label = TRUE),
      shour = hour(hms(as.character(OCCUR_TIME)))
  ) %>%
  mutate(hour_bucket = case_when(
    (shour >= 5 & shour < 12) ~ "Morning 5AM-12PM",
```

```
    (shour >= 12 & shour < 16) ~ "Afternoon 12PM-4PM",
    (shour >= 16 & shour < 20) ~ "Evening 4PM-8PM",
    (shour >= 20 & shour <= 24) ~ "Night 8PM-12AM",
    (shour >= 0 & shour < 5) ~ "Midnight 12AM-5AM",
    ))
  result
}

# Create Categorical columns - Categorical functions are very useful while doing grouping or drawing gr
# In our use case, here are all columns that need to be transformed to categorical.
tdChangeColTypesForCategoricalCols <- function(ds_nyc_sh) {
  ds_nyc_sh$BORO = as.factor(ds_nyc_sh$BORO)
  # replace PERP_AGE_GROUP with factors
  ds_nyc_sh$PERP_AGE_GROUP = as.factor(ds_nyc_sh$PERP_AGE_GROUP)
  ds_nyc_sh$PRECINCT = as.factor(ds_nyc_sh$PRECINCT)

  ds_nyc_sh$PERP_AGE_GROUP = as.factor(ds_nyc_sh$PERP_AGE_GROUP)
  ds_nyc_sh$VIC_AGE_GROUP = as.factor(ds_nyc_sh$VIC_AGE_GROUP)
  ds_nyc_sh$PERP_SEX = as.factor(ds_nyc_sh$PERP_SEX)
  ds_nyc_sh$VIC_SEX = as.factor(ds_nyc_sh$VIC_SEX)
  ds_nyc_sh$PERP_RACE = as.factor(ds_nyc_sh$PERP_RACE)
  ds_nyc_sh$VIC_RACE = as.factor(ds_nyc_sh$VIC_RACE)
  ds_nyc_sh
}

# Plot US Map
# This function uses the map api to plot some cool graphs.
# https://cran.r-project.org/web/packages/usmap/vignettes/usmap3.html
plotMapForUSACounties <- function(df, region_type, state_code, column_name,
                          color1, color_low, color_high, plot_name, plot_title) {
  plt <- plot_usmap(regions = region_type, include=state_code,data=df,
             values=column_name, color = color1, labels=TRUE) +
    scale_fill_continuous(low=color_low, high=color_high, na.value="white", name=plot_name) +
    theme(legend.position = "right")
  plt$layers[[2]]$aes_params$size <- 2.5
  plt <- plt + ggtitle(plot_title)
  plt
}
```

Section 1B: Import the datasets and cache locally:

- We will first load the data from its SOR :
  - URL : https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD
- We will cache the data locally for future analysis.

```
DOWNLOAD_URL <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
LOCAL_FILE = "cached_data/NYPD_Shooting_Incident_Data__Historic_.csv"
if (!file.exists("cached_data")) {
  dir.create("cached_data")
}

if (!file.exists(LOCAL_FILE)) {
```

```
  print(str_c("Please wait .. Downloading the data from :", DOWNLOAD_URL))
  download.file(DOWNLOAD_URL, destfile=LOCAL_FILE)
}
nyc_raw_data <- read_csv(LOCAL_FILE)
records <- nyc_raw_data %>% summarize(ct = n())
columns <- length(colnames(nyc_raw_data))
print(str_c("Datafile loaded. Rows:", records, ", Columns:", columns))
```

```
## [1] "Datafile loaded. Rows:27312, Columns:21"
```

```
# We will create a fips dataset to create some cool maps. Please review fips code from:-
# https://en.wikipedia.org/wiki/List_of_counties_in_New_York
# New York County (Manhattan), Kings County (Brooklyn), Bronx County (The Bronx),
# Richmond County (Staten Island), and Queens County (Queens)
fips_df <- df <- data.frame(BORO=c("BRONX", "BROOKLYN", "MANHATTAN", "QUEENS", "STATEN ISLAND"),
                fips=c("36005", "36047", "36061", "36081", "36085"))
```

Section 1C: Prepare the NYC incident dataset

- Tidy the NYC incident datasets.
- Join with the FIPS for NYC Boros

```
# Total incidents
total_no_of_incidents_fl <- (nyc_raw_data %>% summarize(ct = n()))$ct
total_no_of_incidents <- formatNumber(total_no_of_incidents_fl)

# Get the incidents by Boro and join with FIPS dataset
nyc_inc_by_boro_df <- nyc_raw_data %>%
  group_by(BORO) %>%
  summarise(inc_ct = n()) %>%
  ungroup() %>%
  full_join(fips_df, by = c("BORO")) %>%
  mutate(BORO = factor(BORO))

# remove unwanted cols
nyc_dataset_df <- tdRemoveCols(nyc_raw_data)

# fix field values
nyc_dataset_df <- tdDatasetFixColValues(nyc_dataset_df)

# Augment Date time
nyc_dataset_df <- tdAugmentDates(nyc_dataset_df)

# Prepare categorical columns
nyc_dataset_df <- tdChangeColTypesForCategoricalCols(nyc_dataset_df)
```

Section 1D: Summarize: Summarize the NYC incident datasets

- Two datasets were prepared:
  - nyc_dataset_df : Tidy'd data for detail analysis.

  - nyc_inc_by_boro_df : For high level stats of the incidents.

- From the summary, we noticed the following
  - OCCUR_DATE is a date column.
  - Factor variables are PRECINCT, STATISTICAL_MURDER_FLAG, PERP_AGE_GROUP, PERP_SEX.
  - PERP_RACE, VIC_AGE_GROUP, VIC_SEX, VIC_RACE.
  - Transformations: We added columns:- syear, smonth, sday_of_week, shour.

```
# Display summary for the nyc_dataset_df dataset.
summary(nyc_dataset_df)
```

```
##   INCIDENT_KEY         OCCUR_DATE          OCCUR_TIME
##  Min.   :  9953245   Min.   :2006-01-01   Length:27312
##  1st Qu.: 63860880   1st Qu.:2009-07-18   Class1:hms
##  Median : 90372218   Median :2013-04-29   Class2:difftime
##  Mean   :120860536   Mean   :2014-01-06   Mode  :numeric
##  3rd Qu.:188810230   3rd Qu.:2018-10-15
##  Max.   :261190187   Max.   :2022-12-31
##
##             BORO       LOC_OF_OCCUR_DESC     PRECINCT     JURISDICTION_CODE
##  BRONX        : 7937   Length:27312       75     : 1557   Min.   :0.0000
##  BROOKLYN     :10933   Class :character   73     : 1452   1st Qu.:0.0000
##  MANHATTAN    : 3572   Mode  :character   67     : 1216   Median :0.0000
##  QUEENS       : 4094                      44     : 1020   Mean   :0.3269
##  STATEN ISLAND:  776                      79     : 1012   3rd Qu.:0.0000
##                                           47     :  953   Max.   :2.0000
##                                           (Other):20102   NA's   :2
##  LOC_CLASSFCTN_DESC LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:27312       Length:27312       Mode :logical           <18    : 1591
##  Class :character   Class :character   FALSE:22046             18-24  : 6222
##  Mode  :character   Mode  :character   TRUE :5266              25-44  : 5687
##                                                               45-64  :  617
##                                                               65+    :   60
##                                                               UNKNOWN:13135
##
##  PERP_SEX                            PERP_RACE     VIC_AGE_GROUP   VIC_SEX
##  F:  424   AMERICAN INDIAN/ALASKAN NATIVE:    2   <18    : 2839   F: 2615
##  M:15439   ASIAN / PACIFIC ISLANDER      :  154   18-24  :10086   M:24686
##  U:11449   BLACK                        :11432   25-44  :12281   U:   11
##            BLACK HISPANIC               : 1314   45-64  : 1863
##            UNKNOWN                      :11786   65+    :  181
##            WHITE                        :  283   UNKNOWN:   62
##            WHITE HISPANIC               : 2341
##                              VIC_RACE         syear          smonth
##  AMERICAN INDIAN/ALASKAN NATIVE:   10   Min.   :2006   Jul    : 3238
##  ASIAN / PACIFIC ISLANDER      :  404   1st Qu.:2009   Aug    : 3156
##  BLACK                         :19439   Median :2013   Jun    : 2829
##  BLACK HISPANIC                : 2646   Mean   :2013   Sep    : 2572
##  UNKNOWN                       :   66   3rd Qu.:2018   May    : 2571
##  WHITE                         :  698   Max.   :2022   Oct    : 2279
##  WHITE HISPANIC                : 4049                  (Other):10667
##  sday_of_week     shour       hour_bucket
##  Sun:5452     Min.   : 0.00   Length:27312
```

```
##  Mon:3883     1st Qu.: 3.00   Class :character
##  Tue:3163     Median :15.00   Mode  :character
##  Wed:3000     Mean   :12.22
##  Thu:3034     3rd Qu.:20.00
##  Fri:3585     Max.   :23.00
##  Sat:5195
```

```
# Display summary for the nyc_inc_by_boro_df dataset.
summary(nyc_inc_by_boro_df)
```

```
##             BORO       inc_ct          fips
##  BRONX        :1   Min.   :  776   Length:5
##  BROOKLYN     :1   1st Qu.: 3572   Class :character
##  MANHATTAN    :1   Median : 4094   Mode  :character
##  QUEENS       :1   Mean   : 5462
##  STATEN ISLAND:1   3rd Qu.: 7937
##                    Max.   :10933
```

- Display few rows.

```
# Display two rows for the nyc_dataset_df dataset.
nyc_dataset_df %>% head(n=2) %>% print(width=Inf)
```

```
## # A tibble: 2 x 21
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO   LOC_OF_OCCUR_DESC PRECINCT
##          <dbl> <date>     <time>     <fct>  <chr>             <fct>
## 1    228798151 2021-05-27 21:30      QUEENS <NA>              105
## 2    137471050 2014-06-27 17:40      BRONX  <NA>              40
##   JURISDICTION_CODE LOC_CLASSFCTN_DESC LOCATION_DESC STATISTICAL_MURDER_FLAG
##               <dbl> <chr>              <chr>         <lgl>
## 1                 0 <NA>               UNKNOWN       FALSE
## 2                 0 <NA>               UNKNOWN       FALSE
##   PERP_AGE_GROUP PERP_SEX PERP_RACE VIC_AGE_GROUP VIC_SEX VIC_RACE syear smonth
##   <fct>          <fct>    <fct>     <fct>         <fct>   <fct>    <dbl> <ord>
## 1 UNKNOWN        U        UNKNOWN   18-24         M       BLACK     2021 May
## 2 UNKNOWN        U        UNKNOWN   18-24         M       BLACK     2014 Jun
##   sday_of_week shour hour_bucket
##   <ord>        <dbl> <chr>
## 1 Thu             21 Night 8PM-12AM
## 2 Fri             17 Evening 4PM-8PM
```

```
# Display two rows for the nyc_inc_by_boro_df dataset.
nyc_inc_by_boro_df %>% head(n=2) %>% print(width=Inf)
```

```
## # A tibble: 2 x 3
##   BORO     inc_ct fips
##   <fct>     <int> <chr>
## 1 BRONX      7937 36005
## 2 BROOKLYN  10933 36047
```

Section 2A: Analyze and Prepare the graphs for the NYC incident datasets Analyze and prepare stats for **State, Counties & Counts** for Visualization

```
# ----------------- NYC Counties
# County label and fips code for NYC as found from wiki page. https://en.wikipedia.org/wiki/List_of_cou
counties_lbl <- "Shooting incidents in NYC BORO's: \nKings County (Brooklyn), \nBronx County (The Bronx
NYC_BORO_CODES <- c("36005", "36047", "36061", "36081", "36085");

# Display the map of NYC
nyc_inc_by_boro_vz1 <- plotMapForUSACounties(
  nyc_inc_by_boro_df, "county", c("NY", NYC_BORO_CODES), "inc_ct",
  "coral", "yellow", "coral3", "Shooting Incidents",
  str_c("New York City - Shooting Incidents. Total incidents:", total_no_of_incidents))

# Display only the Boros and concentration of incidents
nyc_inc_by_boro_vz2 <- plotMapForUSACounties(
  nyc_inc_by_boro_df, "county", NYC_BORO_CODES, "inc_ct",
  "coral", "yellow", "coral3", "Shooting Incidents",
  counties_lbl)

# Create a new dataset for display labels with counts
# We want the legends to display labels, hence we need to create a dataframe and join the counts for th
nyc_dataset_df1 <- nyc_dataset_df %>%
  group_by(syear, BORO) %>% summarize(ct = n()) %>%
  ungroup() %>%
  full_join(nyc_inc_by_boro_df) %>%
  # reorder the data so that it can appear sequentially on the facet grid
  mutate(boro_with_inc = paste0(BORO, '-', formatNumber(inc_ct))) %>%
  mutate(BORO = reorder(BORO, -inc_ct),
    boro_with_inc = reorder(boro_with_inc, -inc_ct)) %>%
  select(syear, BORO, ct, boro_with_inc)

# Create the visualization for cases per boro.
# The legends will include the counts
# the facet wrap will be by Boro. It helps to see the data for each Boro independantly.
nyc_incidents_ts_vz <- nyc_dataset_df1 %>%
  ggplot(aes(x=syear, y=ct, fill=boro_with_inc)) +
  geom_bar(stat='identity', aes(color=boro_with_inc), alpha=0.8, width = 0.5) +
  facet_wrap(~BORO, scales="free_y", nrow=3) +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  xlab("Shooting incidents per year") + ylab("Incident count") +
  labs(
    title = str_c("Shooting incidents in NYC Boros. Total incidents:", total_no_of_incidents)
  )
```

Analyze and prepare stats for **Location and Hours** for Visualization

```r
# ----------------- NYC incidents by Location
# Prepare the location stats dataframe by grouping by LOCATION_DESC
nyc_incidents_by_loc_df <- nyc_dataset_df %>%
  group_by(LOCATION_DESC) %>% summarize(inc_ct = n()) %>%
  filter(inc_ct > 70)

# How many other location exists -- todo try fct_lump
other_locations_ct <- (nyc_dataset_df %>%
  group_by(LOCATION_DESC) %>% summarize(inc_ct = n()) %>%
  filter(inc_ct <= 70) %>%
  summarize(inc_ct = n())
  )$inc_ct

# All other locations since fct_lump is not working
# Any locations with fewer counts will be lumped together with locations
other_locations_df <- data.frame(LOCATION_DESC=c("Other-Known-Locations-Grouped"),
                inc_ct=c(other_locations_ct))

# Merge two datasets - append operation :- nyc_incidents_by_loc_df, other_locations_df
nyc_incidents_by_loc_df <- rbind(nyc_incidents_by_loc_df, other_locations_df)

# Sort the dataframe and add more informative labels
nyc_incidents_by_loc_df <- nyc_incidents_by_loc_df %>%
  mutate(LOCATION_DESC = paste0(LOCATION_DESC, ' - ', formatNumber(inc_ct))) %>%
  mutate(LOCATION_DESC = reorder(LOCATION_DESC, -inc_ct)) %>%
  ungroup()

# Prepare the graph for location display
nyc_incidents_by_loc_vz <- nyc_incidents_by_loc_df  %>%
  ggplot(aes(x="", y=inc_ct, fill=LOCATION_DESC)) +
  geom_bar(stat="identity", width=1)  +
  coord_polar("y", start=0) +
  xlab("") + ylab("") +
  theme_light() +
  ggtitle(str_c("S-Inc by Location. Total incidents:", total_no_of_incidents)) +
  theme(legend.direction="vertical")



# ----------------- NYC incidents by Day of Week and Hours
# Compute Weekly and Hourly incidents
nyc_incidents_by_week_hour_df <- nyc_dataset_df %>%
  group_by(sday_of_week, hour_bucket) %>% count()

# get the bucket wise count
nyc_incidents_hour_buc_df <- nyc_incidents_by_week_hour_df %>%
  group_by(hour_bucket) %>%
  summarize(bucket_ct = sum(n)) %>%
  ungroup()

# join two datasets to get count and labels together
nyc_incidents_by_week_hour_df <- nyc_incidents_by_week_hour_df %>%
  full_join(nyc_incidents_hour_buc_df, by = c("hour_bucket"))
```

```
nyc_incidents_by_week_hour_df <- nyc_incidents_by_week_hour_df %>%
  mutate(hour_bucket = paste0(hour_bucket, '.......', formatNumber(bucket_ct)))

# Prepare graph for incidents by week.
nyc_incidents_by_week_hour_vz <- nyc_incidents_by_week_hour_df %>%
  ggplot(aes(x=sday_of_week, y=n, fill=hour_bucket), label=hour_bucket) +
  geom_bar(stat="identity") +
  scale_fill_manual(values=c("tomato" ,"lightcyan", "lightsteelblue", "khaki1", "gray49", "snow2")) +
  #scale_fill_viridis_c(option="H") +
  theme_light() +
  xlab("Weekday, hour") + ylab("Incident count") +
  ggtitle(str_c("S-Inc by week and hour. Total incidents:", total_no_of_incidents)) +
  theme_classic()
```

Analyze and prepare stats for **Boro wise fatal flags** for Visualization

```
colors_fatal <- c("orange", "red")

# Create new dataset by aggregating data by Boro and STATISTICAL_MURDER_FLAG
nyc_dataset_byfatal_df <- nyc_dataset_df %>%
  group_by(BORO, STATISTICAL_MURDER_FLAG) %>%
  summarise(inc_ct = n()) %>%
  ungroup()

# Create new dataset by aggregating data by STATISTICAL_MURDER_FLAG
stat_flag_df <- nyc_dataset_df %>%
  group_by(STATISTICAL_MURDER_FLAG) %>%
  summarise(flag_ct = n()) %>%
  ungroup()

# Create new dataset by aggregating data by BORO, STATISTICAL_MURDER_FLAG and creating new columns for
nyc_dataset_byfatal_df2 <- nyc_dataset_df %>%
  group_by(BORO) %>%
  summarise(boro_ct = n(),
            true_ct = sum(STATISTICAL_MURDER_FLAG == TRUE),
            false_ct = sum(STATISTICAL_MURDER_FLAG == FALSE)
            ) %>%
  ungroup()

# Join the two datasets to create new dataset with good labels.
nyc_dataset_byfatal_df <- nyc_dataset_byfatal_df %>%
  full_join(nyc_dataset_byfatal_df2, by = c("BORO")) %>%
  full_join(stat_flag_df, by = c("STATISTICAL_MURDER_FLAG")) %>%
  mutate(pct = paste0(round(100 * (inc_ct / boro_ct), 2), "%"),
         fatal_flag = paste0(STATISTICAL_MURDER_FLAG, " - ", formatNumber(flag_ct)))


# Prepare graph for murder flag view.
nyc_dataset_byfatal_vz <- nyc_dataset_byfatal_df %>%
  ggplot(aes(x=BORO, y=inc_ct, fill=fatal_flag)) +
  scale_fill_manual(values=colors_fatal) +
  geom_bar(stat="identity", position=position_dodge()) +
  geom_text(aes(label=pct), color="navy", hjust=-0.06,
```

```
            position = position_dodge(0.9), size=3)+
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  xlab("STATISTICAL_MURDER_FLAG") + ylab("Incident count") +
  labs(
    title = str_c("S-Inc by Fatal flag. Total incidents:", total_no_of_incidents)
  ) +
  coord_flip()
```

Analyze and prepare stats for **Age Groups** for Visualization

```
# ----------------- NYC incidents by perpetrators age group
# Timeseries data : NYC incidents by perpetrators age group
colors_age_grp1 <- c("red" ,"orange", "purple", "limegreen", "blue", "snow2")

# Create dataset for age-group, aggregating by syear, BORO, PERP_AGE_GROUP
nyc_incidents_by_perp_age_group_df <- nyc_dataset_df %>%
  group_by(syear, BORO, PERP_AGE_GROUP) %>% summarize(ct = n()) %>%
  ungroup()

# Create dataset for age-group, aggregating by PERP_AGE_GROUP
nyc_incidents_age_grp_ct_df1 <- nyc_incidents_by_perp_age_group_df %>%
  group_by(PERP_AGE_GROUP) %>% summarize(bucket_ct = sum(ct)) %>%
  ungroup()

# Merge the two datasets to get all the columns from both.
nyc_incidents_by_perp_age_group_df <- nyc_incidents_by_perp_age_group_df %>%
  full_join(nyc_incidents_age_grp_ct_df1, by = c("PERP_AGE_GROUP")) %>%
  mutate(age_grp_ct = paste0(PERP_AGE_GROUP, '.......', formatNumber(bucket_ct)))

# Prepare graph using merged dataset. The graph will be wrapped by Boro
nyc_incidents_by_perp_age_group_ts_vz <- nyc_incidents_by_perp_age_group_df %>%
  ggplot(aes(x=syear, y=ct, fill=age_grp_ct)) +
  geom_bar(stat='identity', alpha=0.8, width = 0.5) +

  facet_wrap(~PERP_AGE_GROUP, scales="free_y") +
  scale_fill_manual(values=colors_age_grp1) +
  facet_wrap(~BORO, scales="free_y") +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  xlab("Year and age group") + ylab("Incident Count") +
```

```
  labs(
    title = str_c("S-Inc by Perpetrator age group. Total incidents:", total_no_of_incidents)
  )



# ----------------- NYC incidents by Victims age group
colors_age_grp2 <- c("red" ,"orange", "purple", "limegreen", "blue", "snow2")
# Timeseries data : NYC incidents by Victims Age group

# Create dataset for age-group, aggregating by syear, BORO, VIC_AGE_GROUP
nyc_incidents_by_vic_age_group_df <- nyc_dataset_df %>%
  group_by(syear, BORO, VIC_AGE_GROUP) %>% summarize(ct = n()) %>%
  ungroup()

# Create dataset for age-group, aggregating by VIC_AGE_GROUP
nyc_incidents_age_grp_ct_df2 <- nyc_incidents_by_vic_age_group_df %>%
  group_by(VIC_AGE_GROUP) %>% summarize(bucket_ct = sum(ct)) %>%
  ungroup()

# merge wo datasets
nyc_incidents_by_vic_age_group_df <- nyc_incidents_by_vic_age_group_df %>%
  full_join(nyc_incidents_age_grp_ct_df2, by = c("VIC_AGE_GROUP")) %>%
  mutate(age_grp_ct = paste0(VIC_AGE_GROUP, '.......', formatNumber(bucket_ct)))

# Prepare graph using merged dataset. The graph will be wrapped by Boro
nyc_incidents_by_vic_age_group_ts_vz <- nyc_incidents_by_vic_age_group_df %>%
  ggplot(aes(x=syear, y=ct, fill=age_grp_ct)) +
  geom_bar(stat='identity', alpha=0.8, width = 0.5) +

  facet_wrap(~VIC_AGE_GROUP, scales="free_y") +
  scale_fill_manual(values = colors_age_grp2) +
  facet_wrap(~BORO, scales = "free_y") +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  xlab("Year and age group") + ylab("Incident Count") +
  labs(
    title = str_c("S-Inc by Victims age group. Total incidents:", total_no_of_incidents)
  )
```

Analyze and prepare stats for **Age Groups and Race** for Visualization

```
# ----------------- NYC incidents by Perpetrator age group and race

rc_colors1 <- c("blue" ,"darkslategrey", "rosybrown3", "snow2", "orchid", "skyblue", "red" )
rc_colors2 <- c("blue", "orchid", "darkslategrey", "rosybrown3", "snow2", "gold", "skyblue",  "cyan", "
```

```r
# Create dataset for age-group and race, aggregating by syear, BORO, PERP_AGE_GROUP, PERP_RACE
nyc_incidents_by_perp_age_group_race_ts_df <- nyc_dataset_df %>%
  group_by(syear, BORO, PERP_AGE_GROUP, PERP_RACE) %>%
  summarize(ct_inc = n()) %>% filter(PERP_AGE_GROUP != "(null)" & ct_inc > 1) %>%
  ungroup()

# Create dataset for age-group and race, aggregating by PERP_RACE
nyc_race_ct_df <- nyc_incidents_by_perp_age_group_race_ts_df %>%
  group_by(PERP_RACE) %>%
  summarize(ct_rc = sum(ct_inc)) %>%
  ungroup()

# Merge two datasets for providing label ct views.
nyc_incidents_by_perp_age_group_race_ts_df <- nyc_incidents_by_perp_age_group_race_ts_df %>%
  full_join(nyc_race_ct_df) %>%
  mutate(race_with_inc = paste0(PERP_RACE, '-', formatNumber(ct_rc)))

# Prepare the graph for age group and race
nyc_incidents_by_perp_age_group_race_ts_vz <- nyc_incidents_by_perp_age_group_race_ts_df %>%
  ggplot(aes(x=PERP_AGE_GROUP, y=ct_inc, fill=race_with_inc)) +
  geom_bar(stat="identity") +
  facet_wrap(~PERP_AGE_GROUP, scales="free_y") +
  scale_fill_manual(values=rc_colors1) +
  facet_wrap(~BORO, scales="free_y") + # nrow=3
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  xlab("Perpetrator age group and race") + ylab("Incident Count") +
  labs(
    title = str_c("S-Inc by Perpetrator age group and race. \nTotal incidents:", total_no_of_incidents)
  )


# ----------------- NYC incidents by Victims age group and race

# Create dataset for age-group and race, aggregating by syear, BORO, VIC_AGE_GROUP, VIC_AGE_GROUP
nyc_incidents_by_vic_age_group_race_ts_df <- nyc_dataset_df %>%
  group_by(syear, BORO, VIC_AGE_GROUP, VIC_RACE) %>%
  summarize(ct_inc = n()) %>% filter(VIC_AGE_GROUP != "(null)" & ct_inc > 1) %>%
  ungroup()

# Create dataset for age-group and race, aggregating by VIC_RACE
nyc_vic_race_ct_df <- nyc_incidents_by_vic_age_group_race_ts_df %>%
  group_by(VIC_RACE) %>%
  summarize(ct_rc = sum(ct_inc)) %>%
  ungroup()

# join both datasets just above.
```

```
nyc_incidents_by_vic_age_group_race_ts_df <- nyc_incidents_by_vic_age_group_race_ts_df %>%
  full_join(nyc_vic_race_ct_df) %>%
  mutate(race_with_inc = paste0(VIC_RACE, '-', formatNumber(ct_rc)))

# Prepare graphs for displaying info by age-group and race
# use facet wrap to show each Boro separately with the info above.
nyc_incidents_by_vic_age_group_race_ts_vz <- nyc_incidents_by_vic_age_group_race_ts_df %>%
  ggplot(aes(x=VIC_AGE_GROUP, y=ct_inc, fill=race_with_inc)) +
  geom_bar(stat="identity") +
  facet_wrap(~VIC_AGE_GROUP, scales="free_y") +
  scale_fill_manual(values=rc_colors2) +
  facet_wrap(~BORO, scales="free_y") + # nrow=3
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  xlab("Victims age group and race") + ylab("Incident Count") +
  labs(
    title = str_c("S-Inc by Victims age group and race. \nTotal incidents:", total_no_of_incidents)
  )
```

Analyze and prepare stats for **Sex** for Visualization

```
# ----------------- NYC incidents by perpetrators gender
# Prepare visualization by Sex
x_colors <- c("deeppink" ,"slategray3", "snow2")
# Timeseries data : NYC incidents by Perpetrators Age group
# Create dataset for perpetrators gender, aggregating by syear, BORO, PERP_SEX
nyc_dataset_by_sex_df <-  nyc_dataset_df %>%
  group_by(syear, BORO, PERP_SEX) %>% summarize(ct = n()) %>%
  ungroup()

# Create dataset for perpetrators gender, aggregating by PERP_SEX
nyc_dataset_by_sex_df2 <- nyc_dataset_by_sex_df %>%
  group_by(PERP_SEX) %>%
  summarize(sex_ct = sum(ct)) %>%
  ungroup()

# join the two datasets to get all cols
nyc_dataset_by_sex_df <- nyc_dataset_by_sex_df %>%
  full_join(nyc_dataset_by_sex_df2, by = c("PERP_SEX")) %>%
  mutate(sex_ct = str_c(PERP_SEX, '-', formatNumber(sex_ct)))

# Prepare graph for perpetrators gender stats display
nyc_incidents_by_perp_sex_ts_vz <- nyc_dataset_by_sex_df %>%
  ggplot(aes(x=syear, y=ct, fill=sex_ct)) +
  geom_bar(stat='identity', aes(fill=sex_ct), alpha=0.8, width = 0.5) +

  facet_wrap(~PERP_SEX, scales="free_y") +
```

```
  scale_fill_manual(values=x_colors) +
  facet_wrap(~BORO, scales="free_y") +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  xlab("Perpetrators gender") + ylab("Incident Count") +
  labs(
    title = str_c("S-Inc by Perpetrator sex: Total incidents:", total_no_of_incidents)
  )

# ----------------- NYC incidents by victims sex
# Timeseries data
# Create dataset for victims gender, aggregating by syear, BORO, VIC_SEX
nyc_dataset_by_vic_sex_df <-  nyc_dataset_df %>%
  group_by(syear, BORO, VIC_SEX) %>% summarize(ct = n()) %>%
  ungroup()

# Create dataset for victims gender, aggregating by VIC_SEX
nyc_dataset_by_vic_sex_df2 <- nyc_dataset_by_vic_sex_df %>%
  group_by(VIC_SEX) %>%
  summarize(sex_ct = sum(ct)) %>%
  ungroup()

# Merge the two datasets
nyc_dataset_by_vic_sex_df <- nyc_dataset_by_vic_sex_df %>%
  full_join(nyc_dataset_by_vic_sex_df2, by = c("VIC_SEX")) %>%
  mutate(sex_ct = str_c(VIC_SEX, '-', formatNumber(sex_ct)))

# Prepare graph for viictoms gender stats display
nyc_incidents_by_vic_sex_ts_vz <- nyc_dataset_by_vic_sex_df %>%
  ggplot(aes(x=syear, y=ct, fill=sex_ct)) +
  geom_bar(stat='identity', aes(fill=sex_ct), alpha=0.8, width = 0.5) +

  facet_wrap(~VIC_SEX, scales="free_y") +
  scale_fill_manual(values=x_colors) +
  facet_wrap(~BORO, scales="free_y") +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  xlab("Victims gender") + ylab("Incident Count") +
  labs(
    title = str_c("S-Inc by Victims sex: Total incidents:", total_no_of_incidents)
  )
```

Analyze and prepare stats for **Precinct** for Visualization

```
# ----------------- NYC incidents for Top N Precincts
# Group by Boro and precinct
nyc_top_n_precinct_df <- nyc_dataset_df %>%
  group_by(BORO, PRECINCT) %>% summarize(ct = n()) %>%
  ungroup()

# Group by Boro and arrange/sort by ct and fetch the TOP 5 per Boro
nyc_top_n_precinct_df <- nyc_top_n_precinct_df %>%
  group_by(BORO) %>%
  arrange(desc(ct)) %>%
  slice(1:5)

# Group by Boro and get the sum of incidents
nyc_top_n_precinct_df2 <- nyc_top_n_precinct_df %>%
  group_by(BORO) %>%
  summarise(tot=sum(ct))

# Join with precinct and get the labels
nyc_top_n_precinct_df <- nyc_top_n_precinct_df %>%
  full_join(nyc_top_n_precinct_df2) %>%
  mutate(boro_pct = paste0(BORO, '\n', formatNumber(tot)))

# Get the total for the precinct for displaying in the label.
nyc_top_n_precinct_total <- (nyc_top_n_precinct_df2 %>% summarise(s = sum(tot)))$s

# Get the precent for the precinct for displaying in the label.
nyc_top_n_precinct_pct <- round(100 * (nyc_top_n_precinct_total/total_no_of_incidents_fl),2)

# label for graph
lbl_prec <- str_c("S-Inc Top 5 precincts by each Boro: \nIncidents", formatNumber(nyc_top_n_precinct_to

# Prepare he graph for Top 5 precinct
nyc_top_n_precinct_vz <- nyc_top_n_precinct_df %>%
  ggplot(aes(x=PRECINCT, y=ct, fill=BORO)) +
  geom_bar(stat='identity', aes(fill=BORO), alpha=0.8, width = 0.4) +
  geom_text(aes(label=ct), color="navy", vjust=-0.08,
            position = position_dodge(0.9), size=3)+
  theme_light() +
  theme(
    legend.position="top",
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 0, hjust = 1)
  ) +
  xlab("Top 5 Precincts per Boro") + ylab("Incident Count") +
  labs(
    title = lbl_prec
  )
```

```r
# ----------------- NYC incidents for Top 5 Precinct - Timeseries data
# Prepare the timeseries data
nyc_top_n_precinct_ts_df <- nyc_dataset_df %>%
  filter(PRECINCT %in% nyc_top_n_precinct_df$PRECINCT) %>%
  group_by(syear, BORO) %>%
  summarise(inc_ct = n())

# Prepare the timeseries graph for Top5 precincts
nyc_top_n_precinct_ts_vz <- nyc_top_n_precinct_ts_df %>%
  ggplot(aes(x=syear, y=inc_ct, fill=BORO)) +
  geom_text(aes(label=inc_ct), color="navy", vjust=-0.08, position = position_dodge(0.9), size=2)+
  geom_line(aes(x=syear, y=inc_ct, color=BORO), stat="identity", size=1.2) +
  theme_light() +
  theme(
    legend.position="top",
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 0, hjust = 1)
  ) +
  xlab("Incident Year") + ylab("Incident Count") +
  scale_x_continuous(n.breaks = 12) +
  labs(
    title = str_c("S-Inc Top 5 precincts by each Boro: \nIncidents", formatNumber(nyc_top_n_precinct_to
                  " (", nyc_top_n_precinct_pct, "%)")
  )


# ----------------- NYC incidents for Brooklyn Precinct
# Create dataset for Brooklyn precinct, aggregating by PRECINCT
nyc_dataset_brooklyn_df <- nyc_dataset_df %>% filter(BORO == "BROOKLYN") %>%
  group_by(PRECINCT) %>% summarize(ct = n()) %>% arrange(desc(ct))

total_no_of_incidents_in_brooklyn <- (nyc_dataset_brooklyn_df %>% summarize(total=sum(ct)))$total
brooklyn_pct <- round((total_no_of_incidents_in_brooklyn/total_no_of_incidents_fl) * 100,2)
total_no_of_incidents_in_brooklyn <- formatNumber(total_no_of_incidents_in_brooklyn)

# ------------
# Slice and dice Top 10
# Get the top 10 precinct in Brooklyn using the rank function and desc order of shooting counts
bk_top10_precinct_df <- nyc_dataset_brooklyn_df %>%
  mutate(rnk = min_rank(desc(ct))) %>%
  filter(rnk <=10)

# Get total no of shooting incidents in Brooklyn
bk_total_inc <- (nyc_dataset_brooklyn_df %>% summarize(tot = sum(ct)))$tot

# Which were the top 10 precints in terms of no of shootings
top10_precinct <- paste(bk_top10_precinct_df$PRECINCT, collapse=', ')

# What was the share of top 10 precints
bk_top10_share <- (bk_top10_precinct_df %>% summarize(tot = sum(ct)))$tot
```

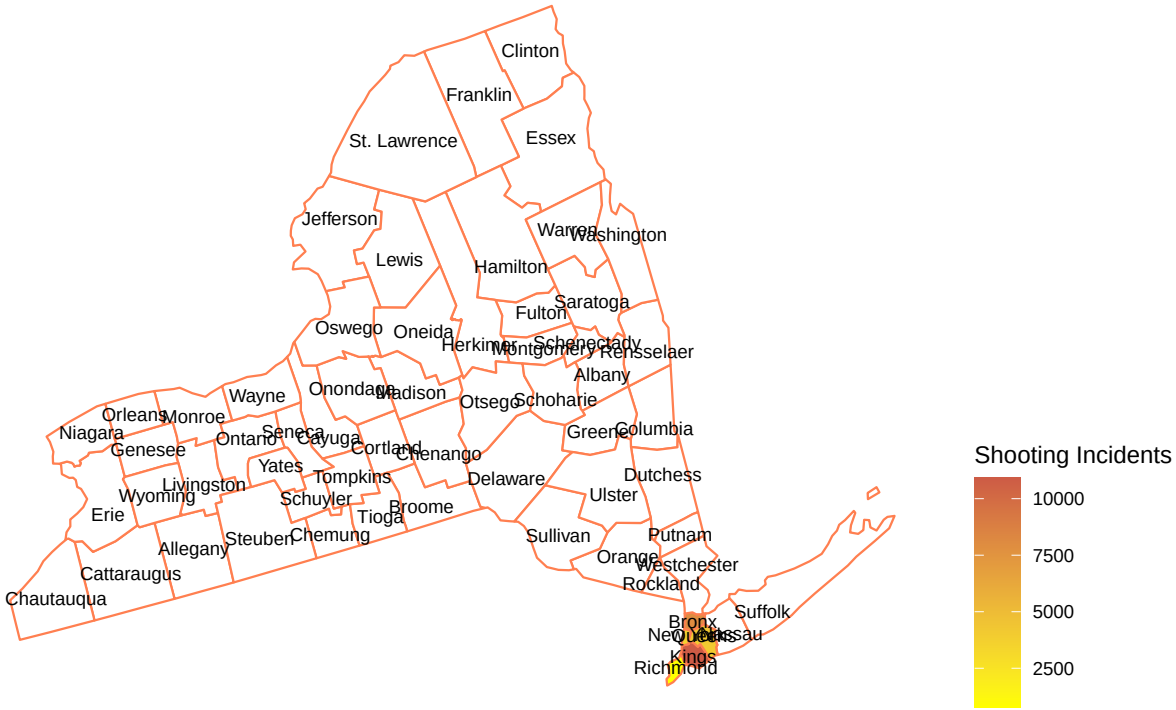
```
bk_top10_global_pct <- str_c(round((bk_top10_share/total_no_of_incidents_fl) * 100,2), "%")

# What was the percentage of top 10 precints
bk_top10_share_pct <- str_c(round((bk_top10_share/bk_total_inc) * 100,2), "%")

bk_inference1 <- str_c("\n NYC: ", total_no_of_incidents, ", Brooklyn: ", total_no_of_incidents_in_brook
bk_top10_inference1 <- str_c("\nTop 10 Brooklyn precincts with the most incidents were :\n", top10_preci
bk_top10_inference2 <- str_c("\nIncidents in the Top 10 precincts was ", formatNumber(bk_top10_share), "
bk_top10_inference3 <- str_c("\nTop 10 precincts overall share: ", bk_top10_global_pct)


display_label <- str_c("Shooting incidents in Brooklyn precincts. ", bk_inference1, bk_top10_inference3

# --------------

# Prepare graph for Brooklyn precinct
nyc_dataset_brooklyn_vz <- ggplot(nyc_dataset_brooklyn_df, aes(x=PRECINCT, y=ct)) +
  geom_bar(stat='identity', fill="coral") +
  xlab("Brooklyn precincts") + ylab("Brooklyn shooting counts") + ggtitle(display_label) +
  theme_light() +
  coord_flip() +
  theme(
    plot.title = element_text(hjust = 0.5, colour = "tomato", size=11),
    plot.subtitle = element_text(hjust = 0.97)
  )
```

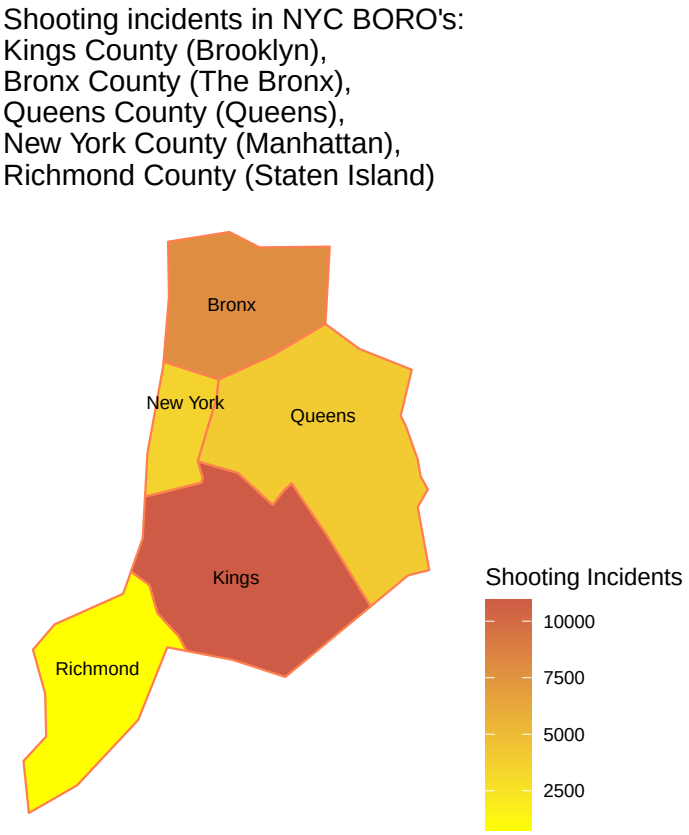
Section 3: Visualizations

Section 3A: Big picture: - Shooting Incidents across New York City

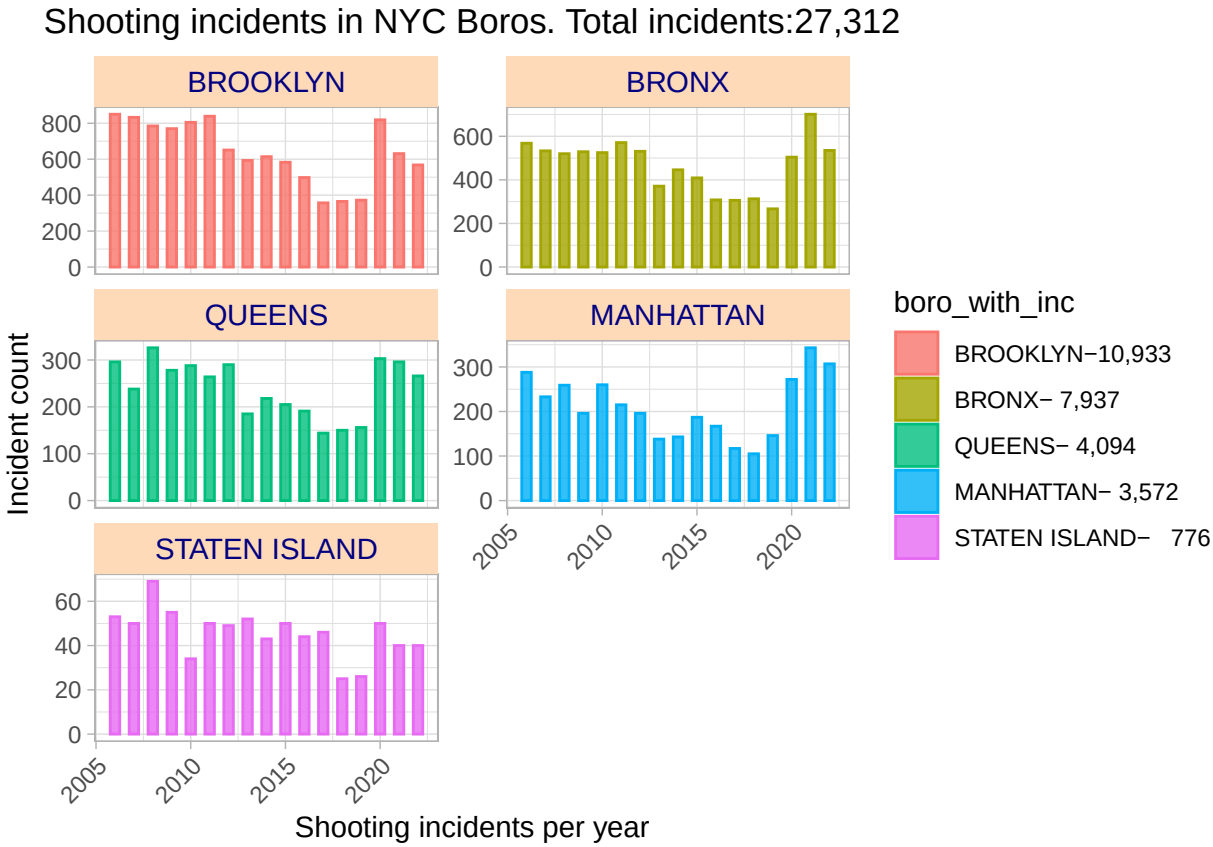
- How many incidents are happening across New York City?
  - The total no of incidents was: **27,312‘**.
  - The incidents only impacted the 5 Boro’s of NYC
  - NYC Boro names and FIPS codes can be found here - List_of_counties_in_New_York

New York City − Shooting Incidents. Total incidents:27,312



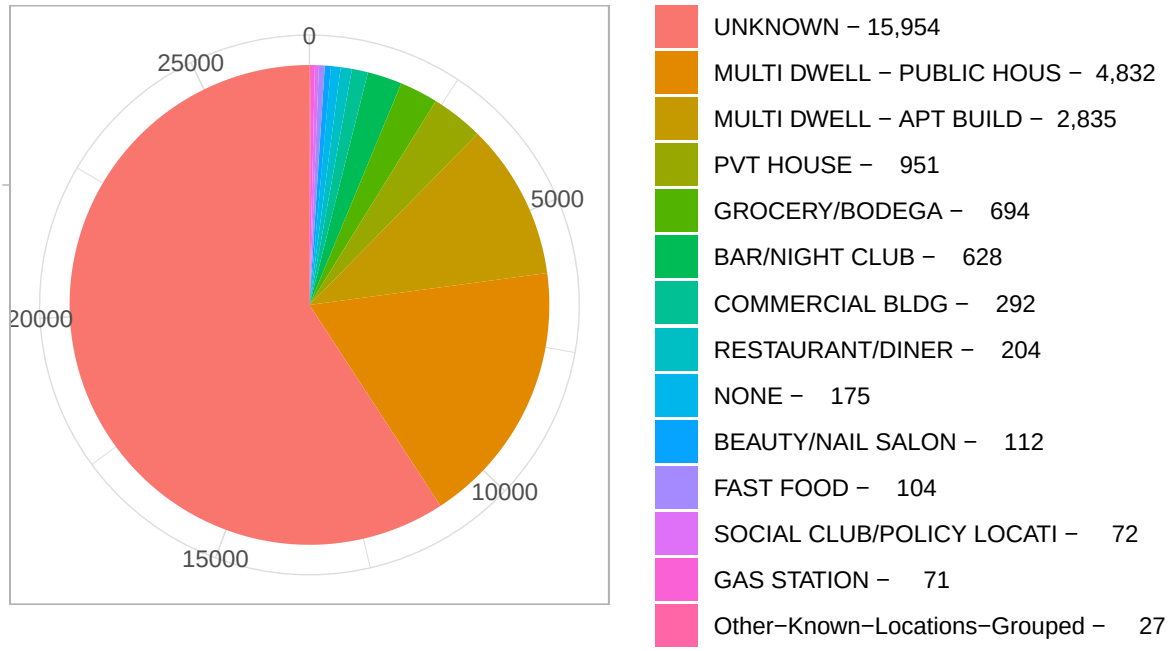


- There were more incidents in Brooklyn and Bronx when compared with others.
  - Brooklyn started with 800+ incidents in 2005, then the incidents slowed down until *2019*.
  - After 2019, the incidents are again on rise in Brooklyn.
  - We observed a similar pattern in cases in Bronx where incidents are on the rise after 2019.

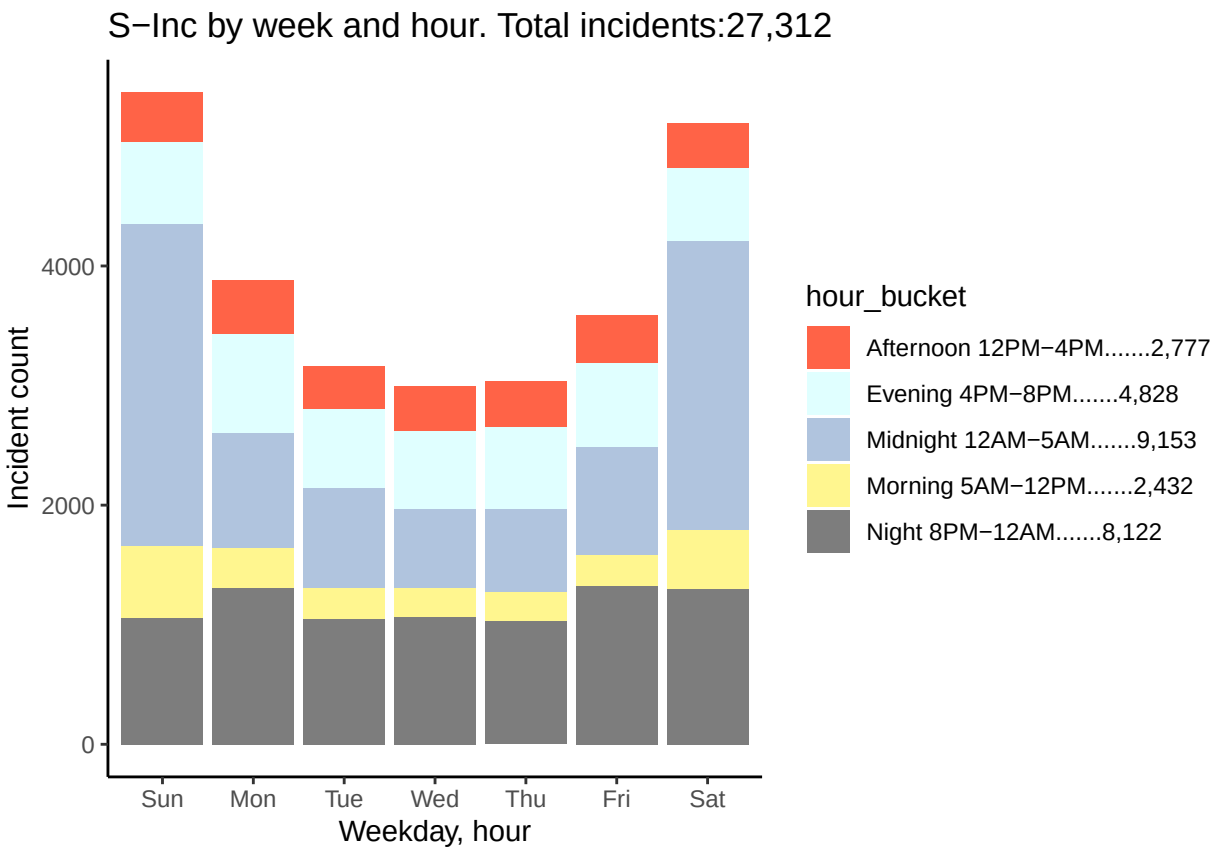


Section 3B: Location & Time: Shooting Incidents across New York City

- **Can we look at the location and time of the incident? What can we conclude?**
  - Incidents were more frequent in the following locations. There were many unknowns too due to data collection issues.
  - We lumped few locations that had fewer than 70 incidents into **Other-known-locations**.
  - From these insights, we can warn people in the top 5 locations to be careful.
  - We can also ask Government to put more law enforcement surveillance in the top 3 areas.

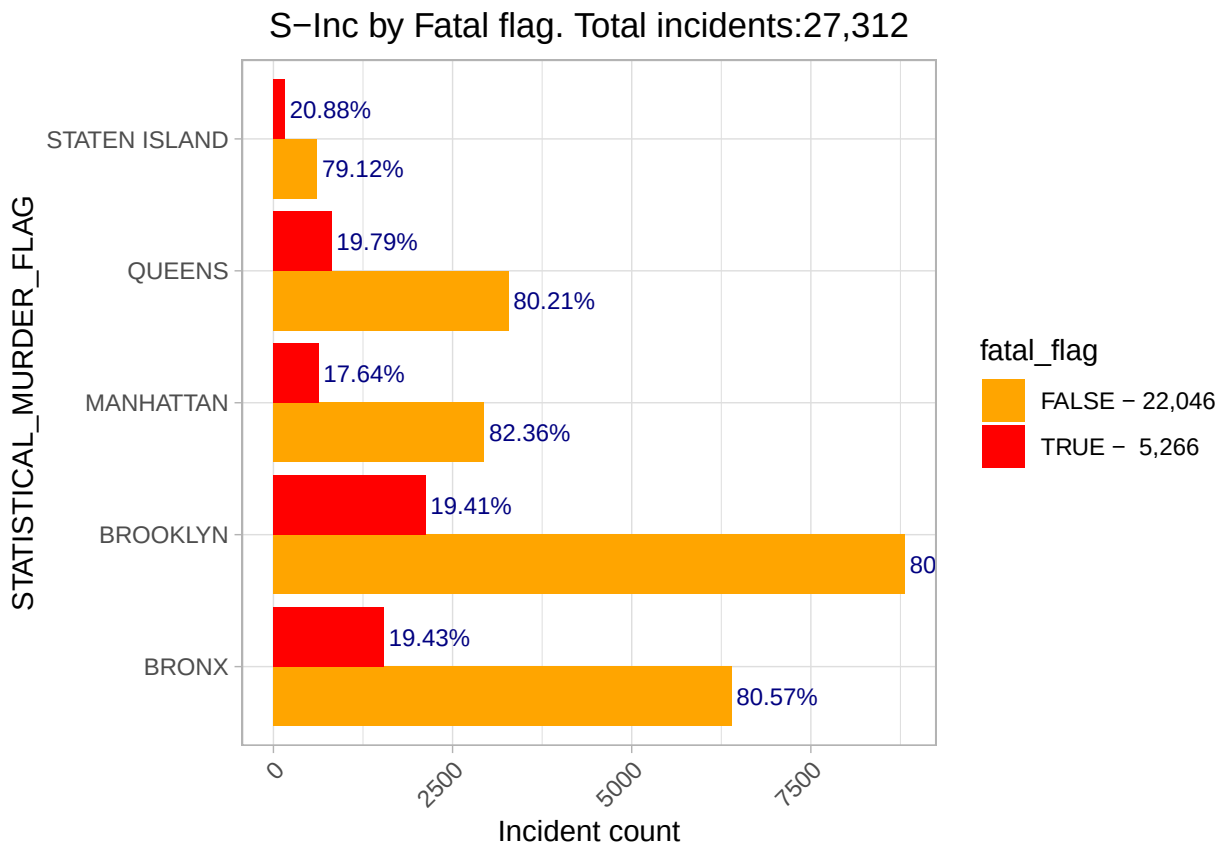


- **What time during the week do these incidents happen?**
  - On a Weekday basis, we get to know when these incidents are happening.
  - We can infer that most of the incidents are happening during the Night and Midnight time in general.
  - This info should help Law Enforcement and the public for safety purposes.



Section 3C: Incident Fatality: Shooting Incidents across New York City

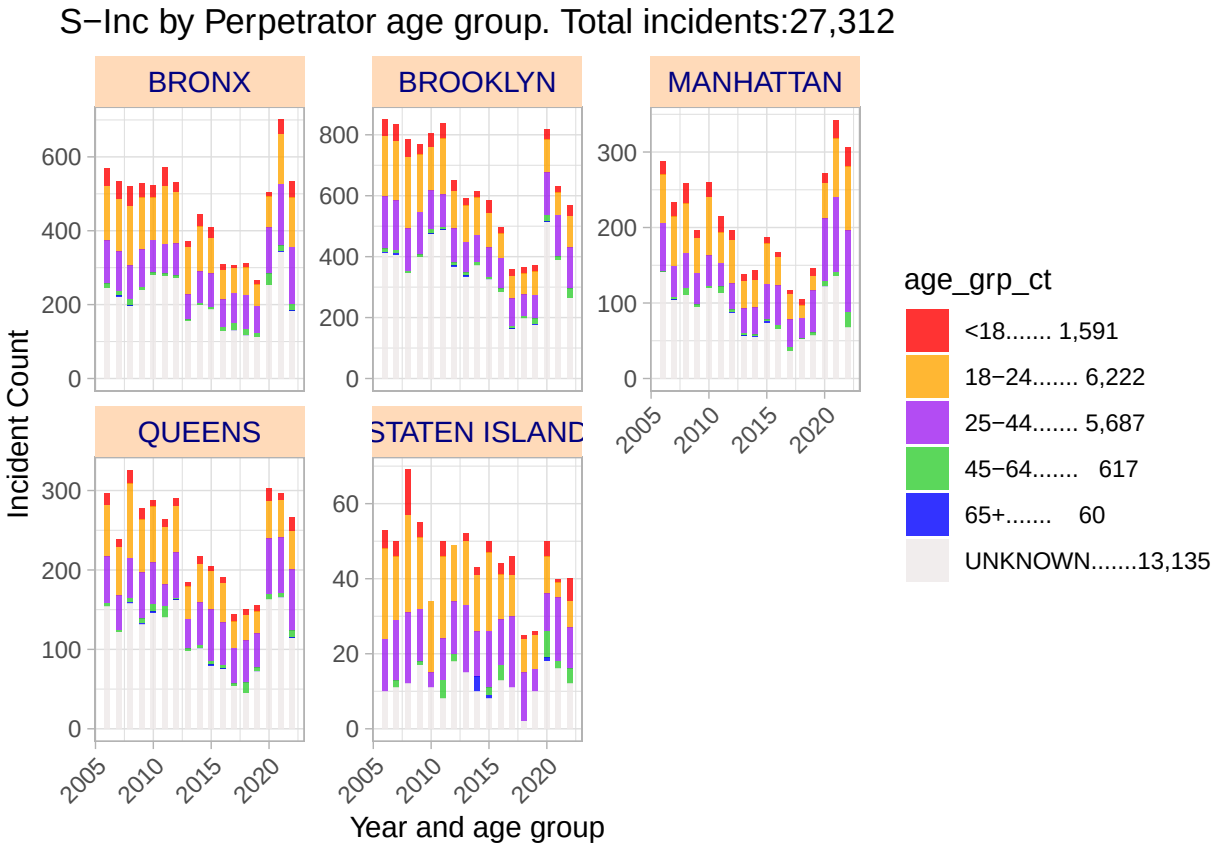
- **Were the incidents Fatal?**
  - Between **17% to 20%** of the incidents resulted in a murder.



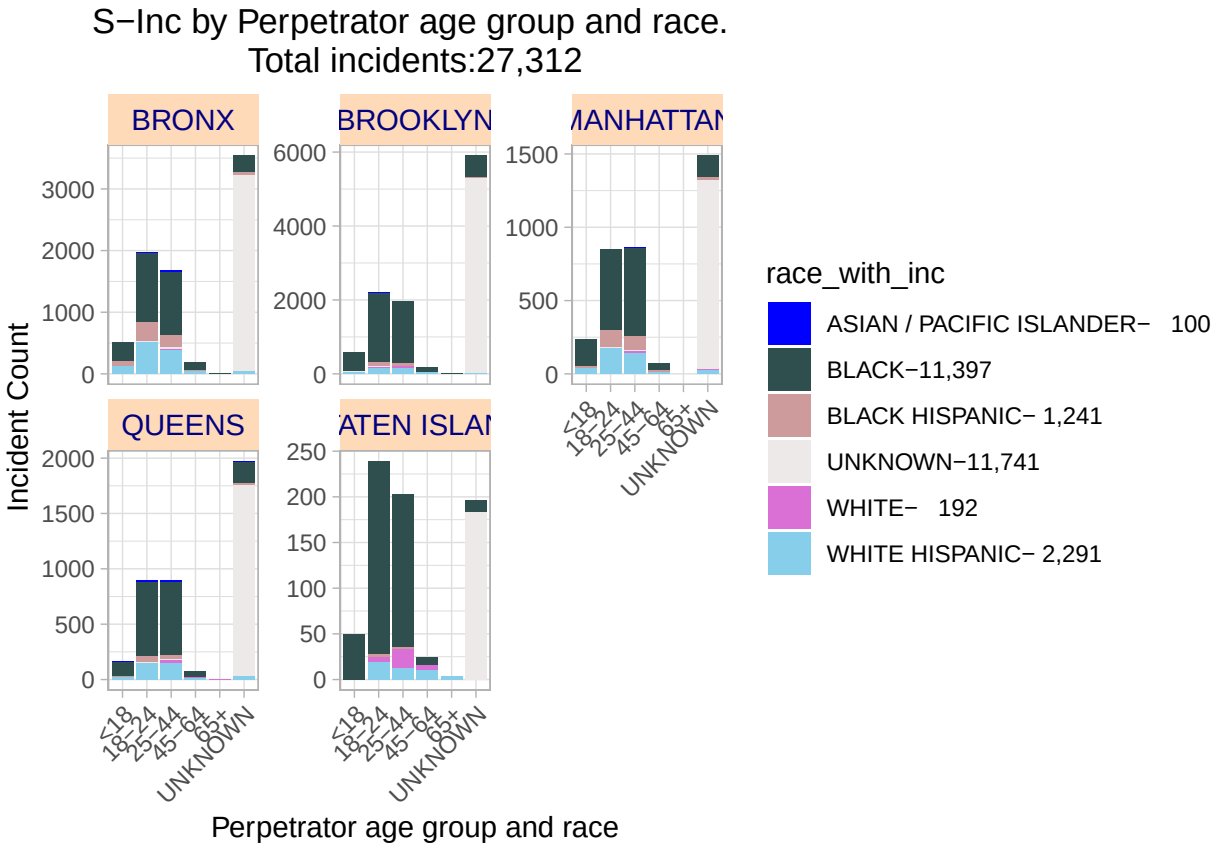
Demographics - Perpetrators

Section 3D: Perpetrators Demographics: Shooting Incidents across New York City

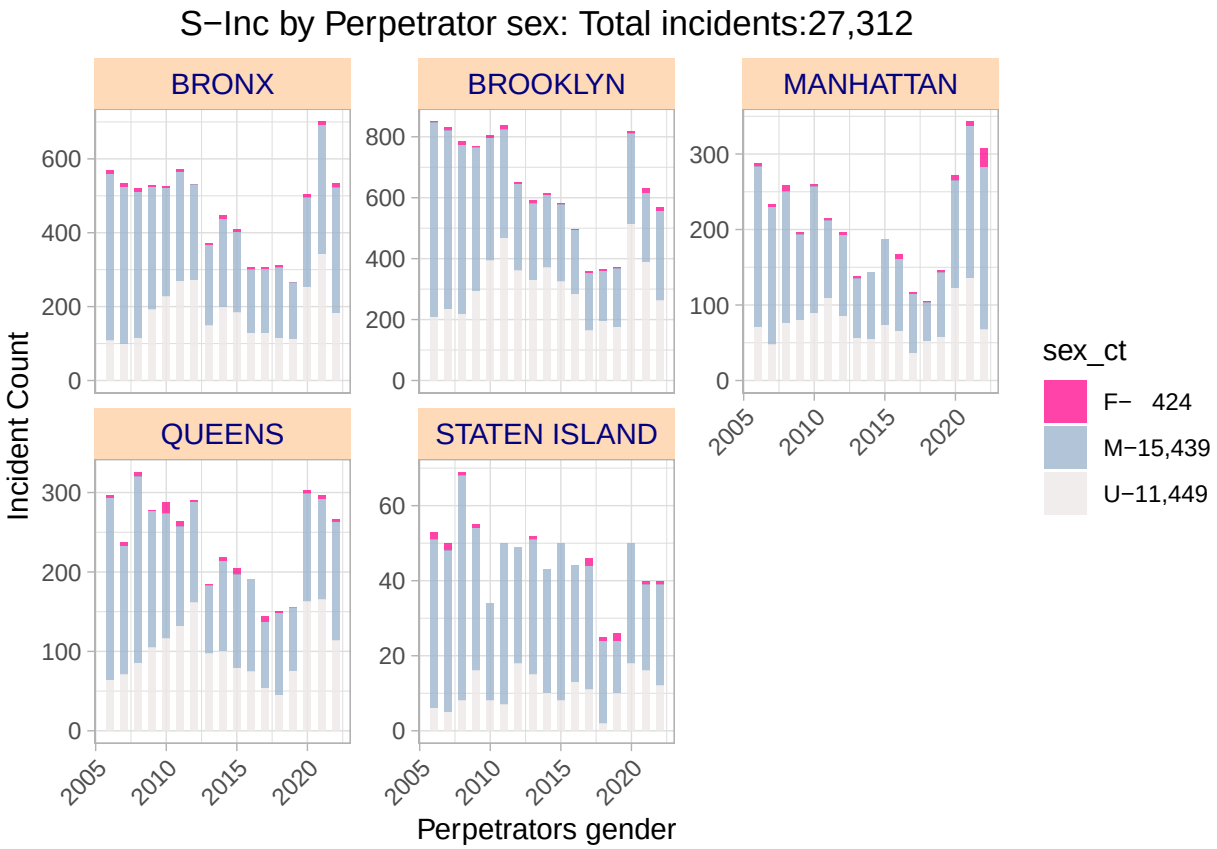
- **Can we understand the demographics of the perpetrators?**
- **What were the age-group of the perpetrators?**
  - Throughout the years, we noticed that the perpetrators of the age groups 18 to 44 committed the most crimes.
  - A large section of the age group is unknown to us, hence it is lightly shaded.
  - The cops couldn't figure out the ages for some reasons.



- **What are the race and age-group of the perpetrators in each Boro?**
  - Based on our data, Black and Black Hispanic between ages 18-44 committed most crimes.



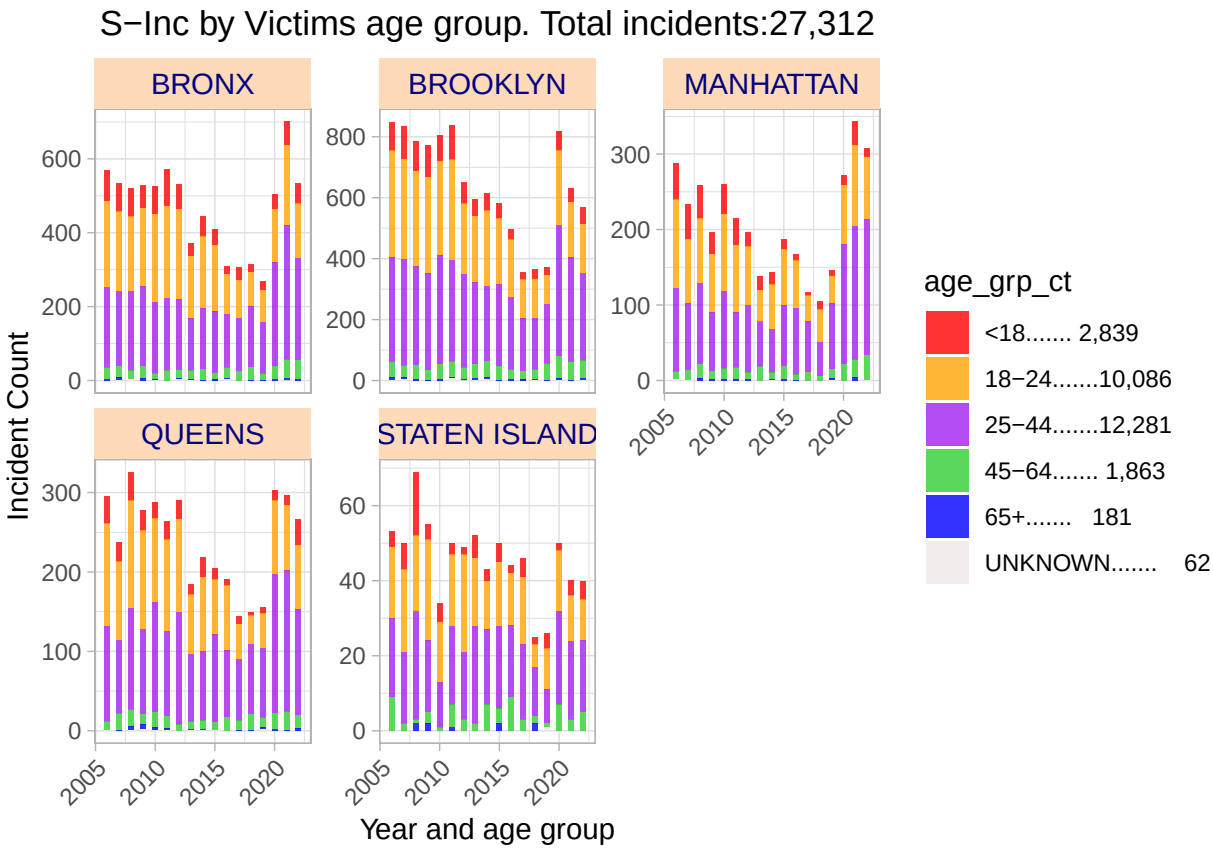
- **What were the gender of the perpetrators?**
  - We can conclude that amongst this group, Men are more involved in crimes than women.
  - There are many unknowns too, since data is missing.
  - Also, men shop for firearms more than women.



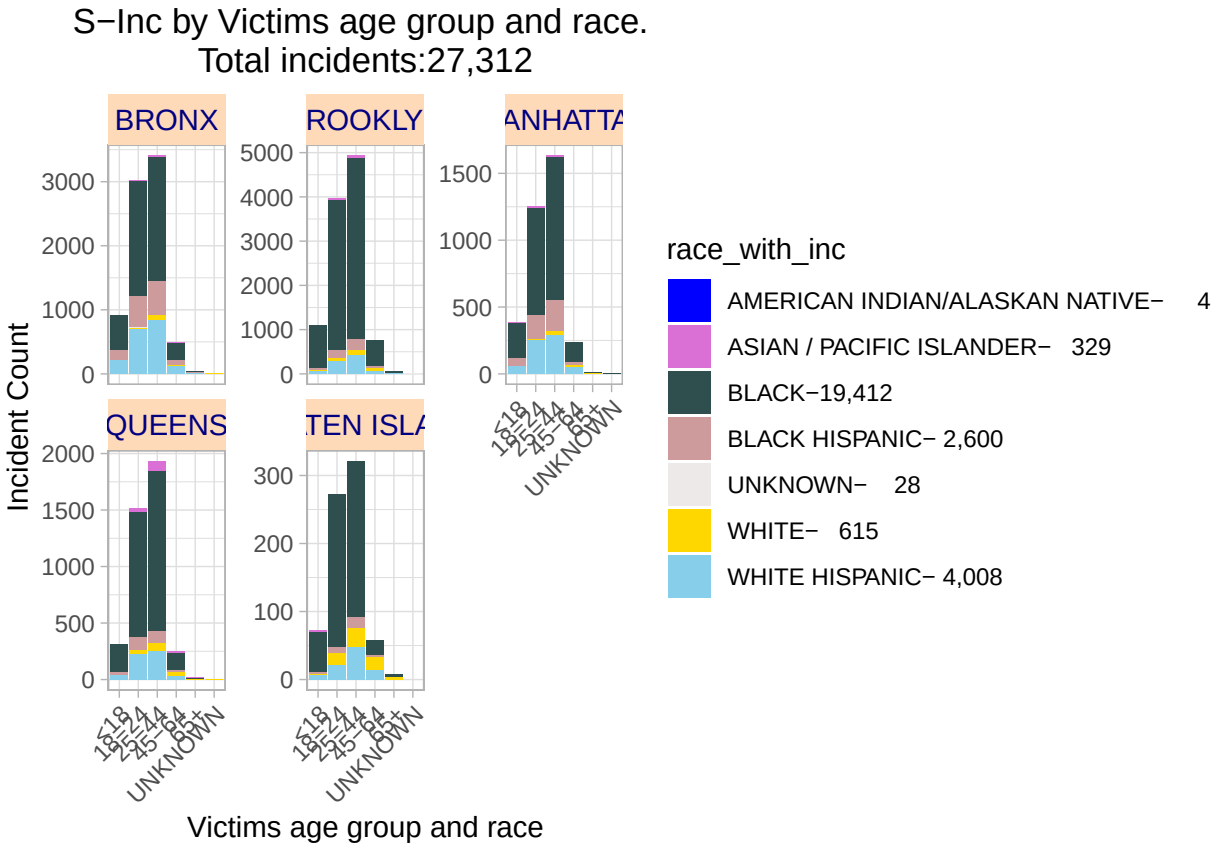
Demographics - Victims

Section 3E: Victims Demographics: Shooting Incidents across New York City

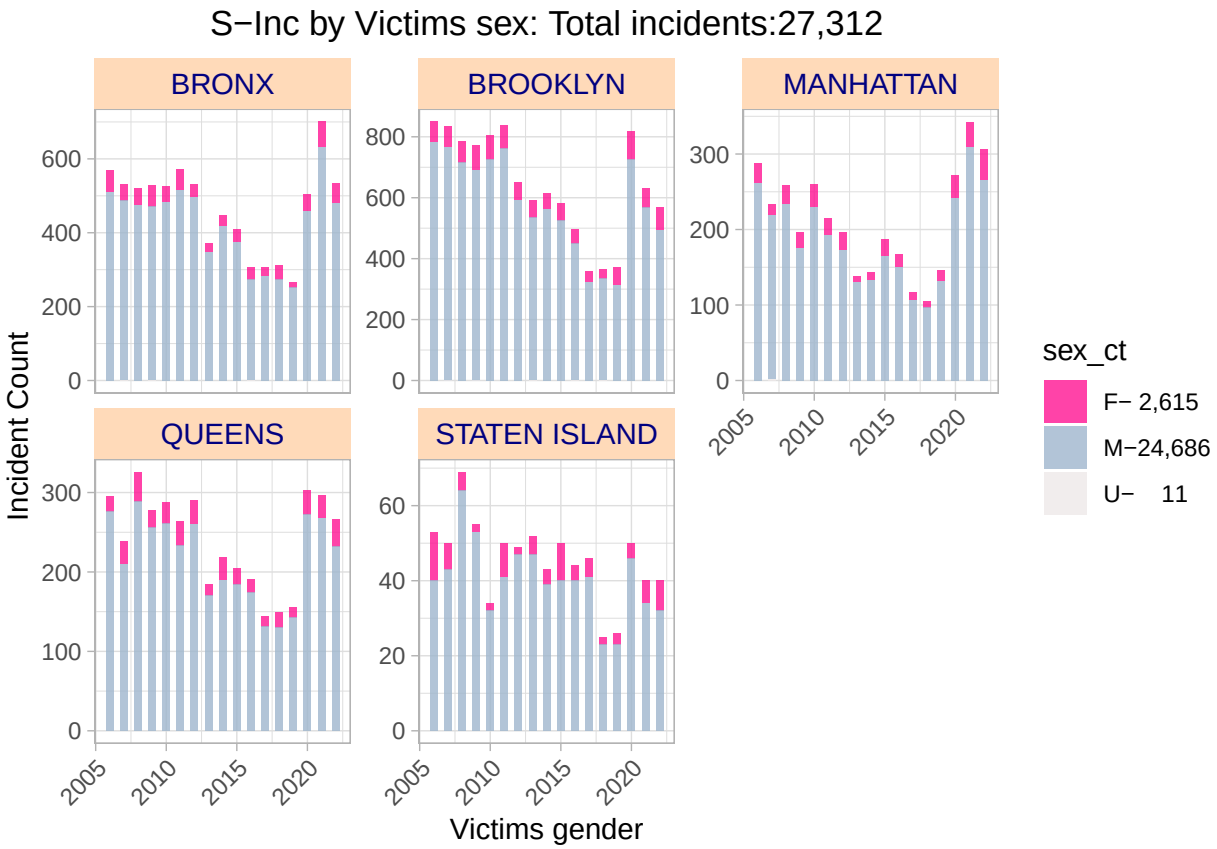
- Can we understand the demographics of the victims?
- **What was the age-group of the victims?**
  - Based on our data, ages between 18-44 were victims.



- **What are the race and age-group of the victims in each Boro?**
  - Based on our data, Black and Black Hispanic between ages 18-44 were the victims.

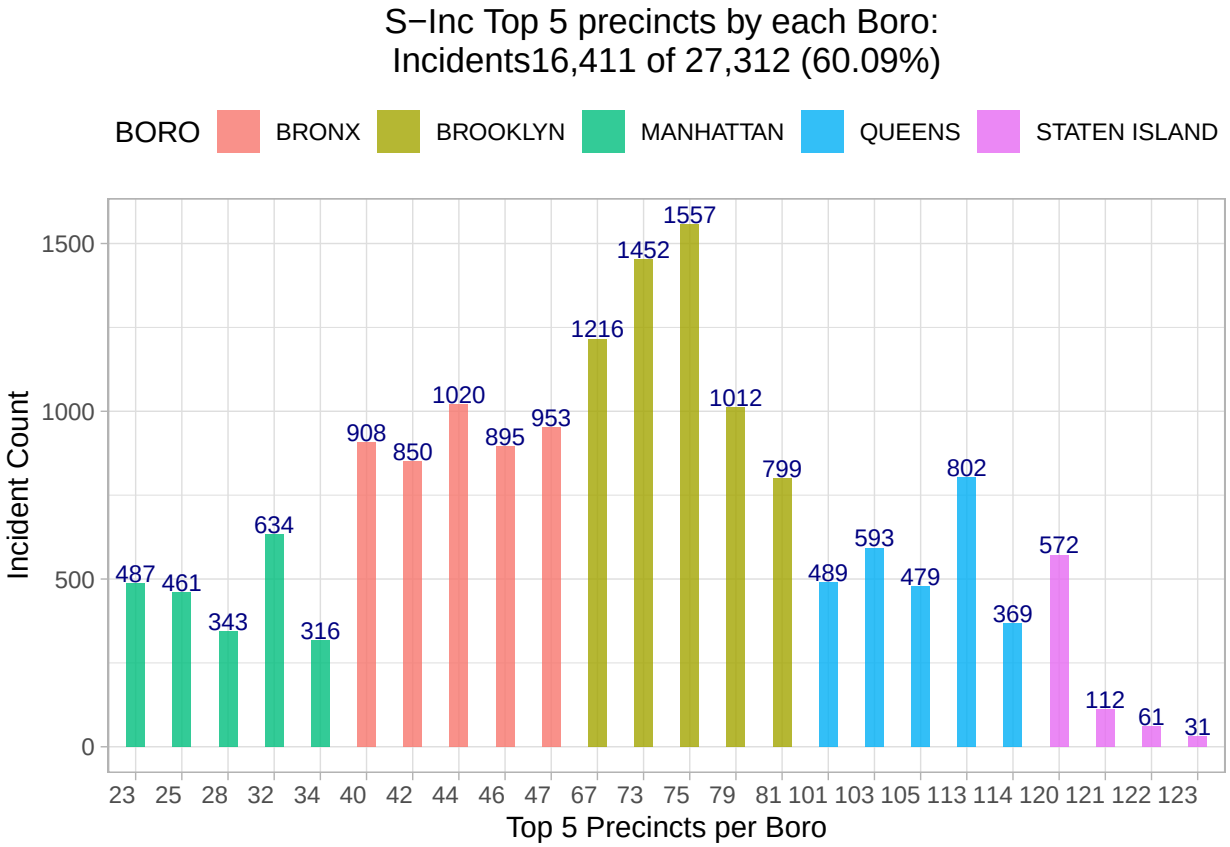


- **What were the gender of the Victims?**
  - We can conclude that amongst this group, Men are more targeted in crimes than women.
  - It is possible that Men are out most of the time than women during night hours in these areas.

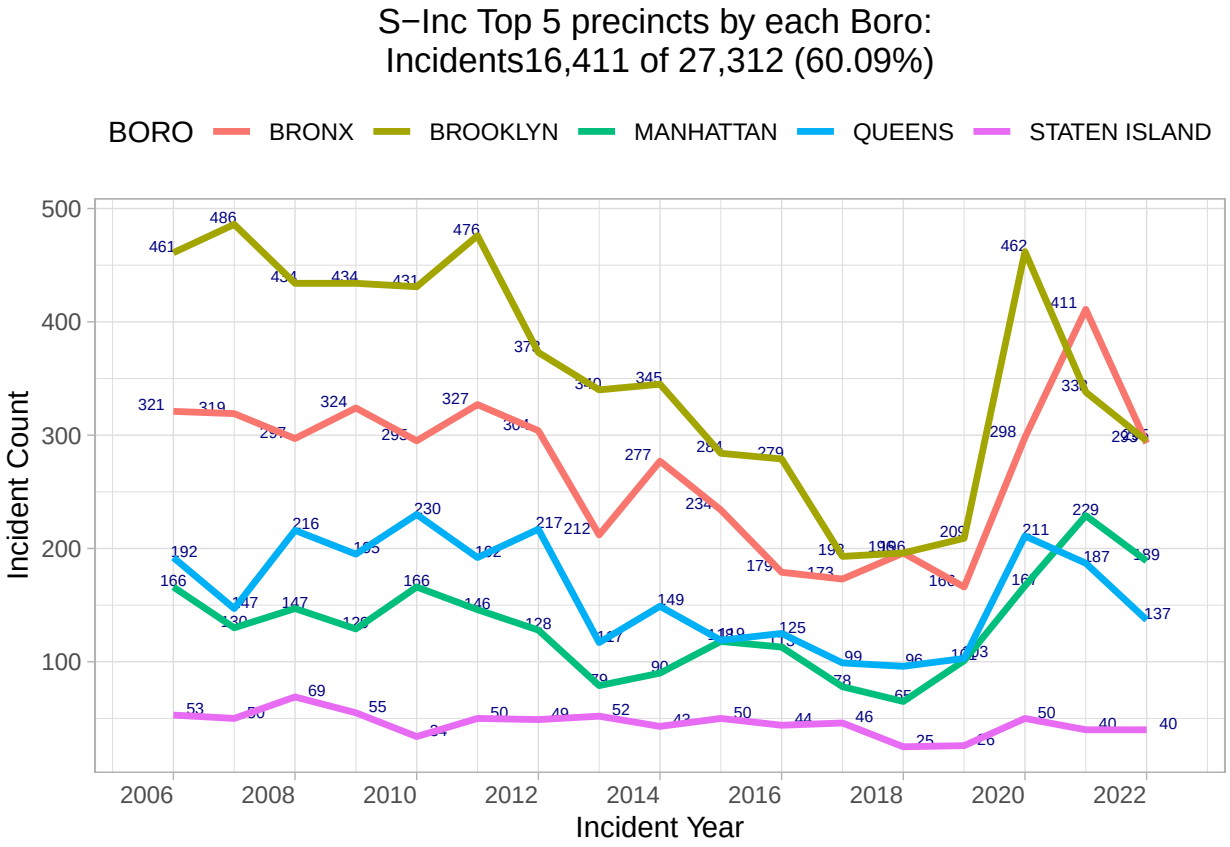


Section 3F: Top 5 precincts: Boro wise shooting incidents.

- **Which are the top 5 precincts(with the most incidents) in each NYC Boro?**
  - Approx 60 % of the incidents happened in these Precincts.

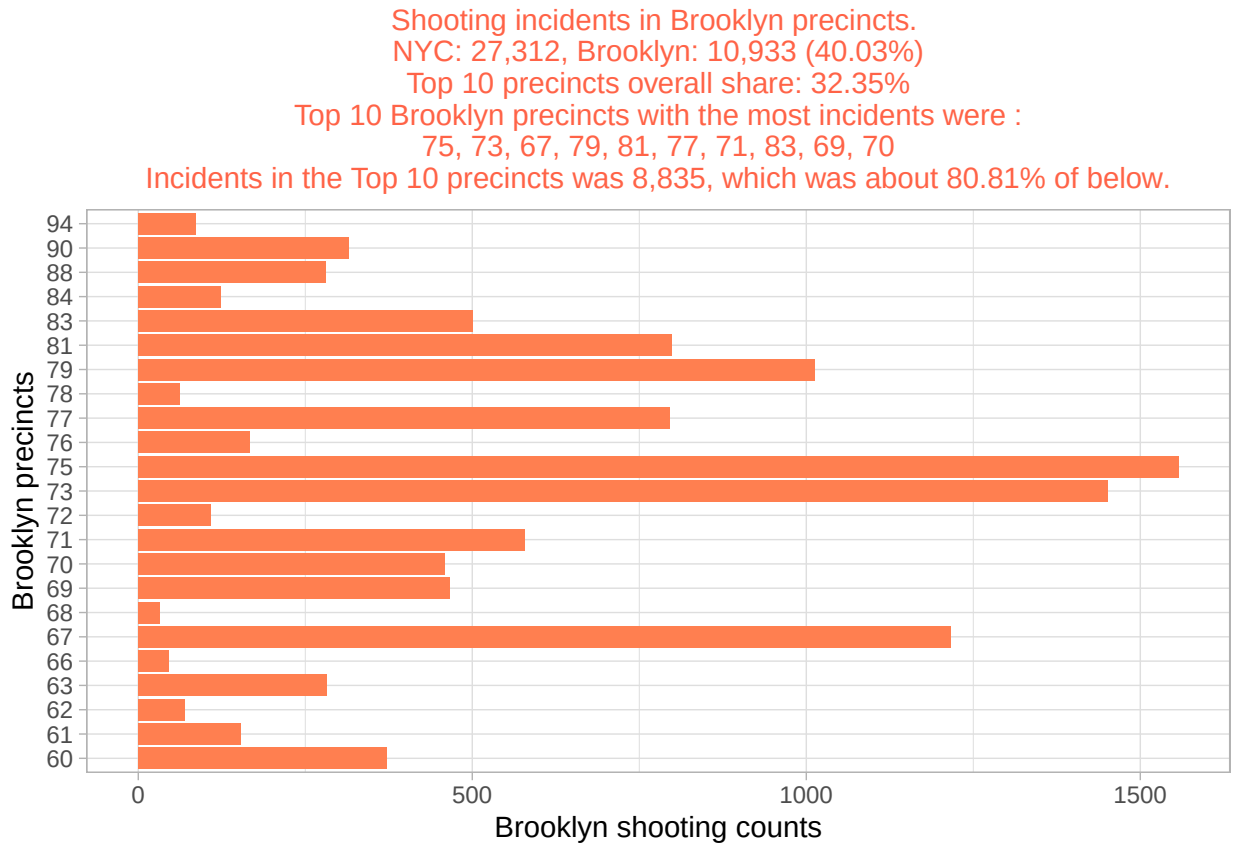


- **Display the yearly trends for the top 5 precincts(with the most incidents) in each NYC Boro?**
  - Clearly, these are **low hanging fruits** for NYPD. If they partner with other Government and Health Agencies, they can reduce the crime by more than 60%.



Section 3G: Closer look: Shooting Incidents in Brooklyn, New York City

- **Which precinct in Brooklyn had the most incidents? What could be the reasons?**
  - There were a few precincts that need more attention from Law Enforcement
  - With limited data we cannot do the root cause analysis for the shootings in this location.
  - **Government needs to partner with other agencies to reduce crime in these precincts**.



Section 4: Model Training and Prediction

Section 4: Model Building: Build Quantitative model for shooting incidents in New York City

- Train and test a model for NYC Shooting incident

Section 4A: Prepare model data and Train a linear model:

- Prepare the data and train model

```
nyc_ct_by_year_mdl_df <- nyc_dataset_df %>%
  group_by(BORO, syear) %>%
  summarise(inc_ct = n()) %>%
  ungroup()

nyc_ct_by_year_mdl = lm(inc_ct ~ syear + BORO, data = nyc_ct_by_year_mdl_df)
```

Section 4B: Run predictions on the model and prepare the Visualizations:

- We ran the predictions or the model and created nyc_ct_by_year_pred_df
- Prepare the visualizations for the predicted model.

```
nyc_ct_by_year_pred_df = nyc_ct_by_year_mdl_df %>% mutate(pred=predict(nyc_ct_by_year_mdl)) %>%
  mutate(BORO = factor(BORO))

nyc_ct_by_year_pred_vz <- ggplot(nyc_ct_by_year_pred_df) +
  geom_point(aes(x = syear, y = inc_ct, color = "Actual")) +
  geom_point(aes(x = syear, y = pred, color = "Predicted")) +
  facet_wrap(~BORO, scales="free_y") +
  theme_light() +
  theme(
    strip.background = element_rect(fill = "peachpuff"),
    strip.text = element_text(colour = "navy", size = rel(1.0)),
    plot.title = element_text(hjust = 0.5),
    plot.subtitle = element_text(hjust = 0.97),
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  labs(
    title = "Predictions by NYC Boros. "
  )
```

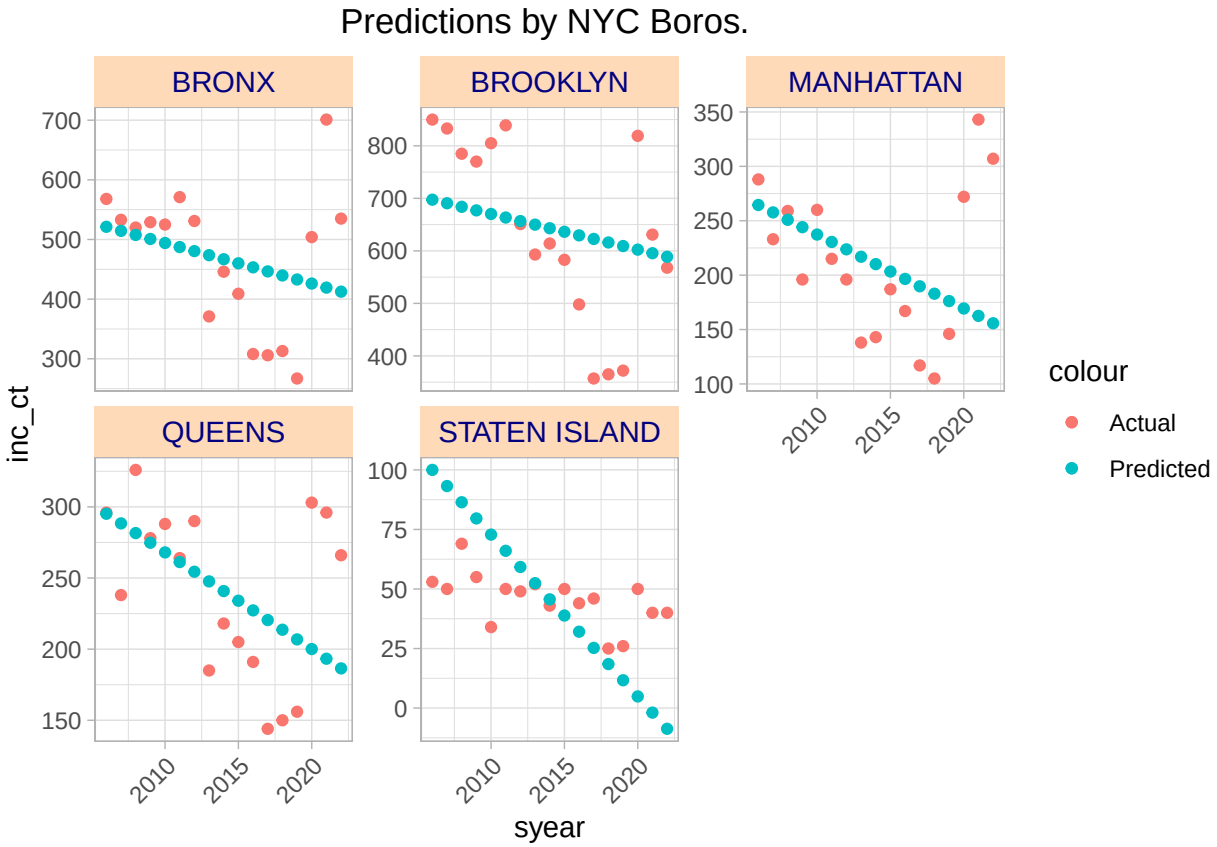
```
# Formula is
summary(nyc_ct_by_year_mdl)[1]
```

```
## $call
## lm(formula = inc_ct ~ syear + BORO, data = nyc_ct_by_year_mdl_df)
```

```
# summarize the champion model
summary(nyc_ct_by_year_mdl$coefficients)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -421.2  -249.1  -116.4  2236.1   130.5 14151.2
```

- **How close are we with respect to Boro wise yearly predictions? will this help NYPD or others?**
  - The future predictions, if accurate, will help the law enforcement to stop the crime before it happens.
  - In the prediction's dataset, we noticed that the incidents are decreasing over time in our model. This is a good starting point.
  - However, this model is not accurate. There are more improvements needed to gain accuracy.
  - Perhaps some other model or new variables may yield better results. We must continue doing the experiments



Section 5A: Bias Identification & Elimination: Professor Jane Wall rightly mentioned that Biases are natural. Everyone falls into their trap. While looking at the initial data I might have some implicit bias such as stereotypes.

Implicit bias refers to the attitudes or stereotypes that affect our understanding, actions, and decisions in an unconscious manner. An implicit bias can make us susceptible to unintentionally acting in ways that are inconsistent with our values. Implicit bias can be based on stereotypes.

A **stereotype** is a generalized belief about a particular category of people. It is an expectation that people might have about every person of a particular group. The type of expectation can vary; it can be, for example, an expectation about the group's personality, preferences, appearance or ability. Stereotypes are sometimes overgeneralized, inaccurate, and resistant to new information, but can sometimes be accurate.

There are other biases that could be introduced while working on this case study: - Emotional bias: due to the nature of crime - Selection bias: Who and how the data was collected. - Anchoring bias: We react to the first piece of information. - Confirmation bias: We might think we are experts. - Salience - Focusing on the simple things to get work done.

Biases - How to overcome: To overcome implicit biases, we need to follow a disciplined and mindful approach of thoroughly analyzing the data, drawing inferences, asking questions, collecting more data and validating our assumptions using some quantitative models. Our report must be based on facts in the data and not based on our emotions and biases.

Section 5B: Conclusion:

- After downloading and cleaning up the data for NYPD shooting, we went through and iterative Data science process of Asking questions, preparing data, analyzing, visualizing the data and drawing conclusions. We eliminated the Biases - stereotype, emotional and other biases - by gleaning valuable information from the data to make inferences.
- We reported our observations at each step. Our key observations were:
  - Between **2005** and **2020**, there were **27,312** shooting incidents in the **five** Boro's and **Brooklyn 40%** and **Bronx** were the areas with most crimes. **Approx. 17% to 20%** of the incidents resulted in a murder. We also noticed that the incident was concentrated in few Precincts within these Boro's.
  - The incidents mostly occurred in the multi-dwell, apt, grocery, bar/night club, and commercial buildings during the night and the midnight hours and mostly men were involved in crime.
  - The race, sex and age-group of most of the perpetrators were - Black Male between **ages 18 to 44** years. The race, sex and age-group of most of the victims were - Black and Black Hispanic and white Hispanic male between **ages 18 to 44** years.
  - The top 5 precincts in each Boro accounted for 60% of the incidents.
  - The top 10 precincts in Brooklyn resulted in a crime rate of **32.35%** of the NYC shootings. These precincts: **75, 73, 67, 79, 81, 77, 71, 83, 69, 70** had total **8,835** shooting incidents and hence they really need attention.
- We built a model to predict the shootings in each Boro for the future. We noticed that our model is good, but not 100% accurate. A linear model in this case might not yield good accuracy. We need to run various experiments - try new variables, try new models, etc. to fine tune our model.
- With the **limited amount** of data, we cannot do provide justifications to the causes of these crimes. It could be late night street fights, drug abuse, etc. It could be related to unemployment and theft. Government should investigate the root cause of crime in the key precincts.

Actionable plan

- The crime rate in NYC can be reduced by more than 1/2 if NYPD can partner with other Government and private agencies to improve the quality of life in the **Top five precincts in each BORO** shared in section 3F.
- We **recommend** that Law Enforcement should deploy more cops in the high-risk region that was highlighted in the report for the safety of the public during night hours. The public should be extra careful during the high-risk hours.

```
sessionInfo()
```

Session Information

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22621)
##
## Matrix products: default
##
##
## locale:
```

```
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] usmap_0.6.2     lubridate_1.9.2 forcats_1.0.0   stringr_1.5.0
##  [5] dplyr_1.1.2     purrr_1.0.2     readr_2.1.4     tidyr_1.3.0
##  [9] tibble_3.2.1    ggplot2_3.4.3   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.3        generics_0.1.3    stringi_1.7.12    hms_1.1.3
##  [5] digest_0.6.33     magrittr_2.0.3    evaluate_0.21     grid_4.3.1
##  [9] timechange_0.2.0  fastmap_1.1.1     fansi_1.0.4       scales_1.2.1
## [13] cli_3.6.1         rlang_1.1.1       crayon_1.5.2      bit64_4.0.5
## [17] munsell_0.5.0     withr_2.5.0       yaml_2.3.7        tools_4.3.1
## [21] parallel_4.3.1    tzdb_0.4.0        usmapdata_0.1.0   colorspace_2.1-0
## [25] vctrs_0.6.3       R6_2.5.1          lifecycle_1.0.3   bit_4.0.5
## [29] vroom_1.6.3       pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.3
## [33] glue_1.6.2        xfun_0.40         tidyselect_1.2.0  highr_0.10
## [37] rstudioapi_0.15.0 knitr_1.43        farver_2.1.1      htmltools_0.5.6
## [41] rmarkdown_2.24    labeling_0.4.2    compiler_4.3.1
```