Week 3 Project

2023-02-12

Required libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.0
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Importing data

To obtain the NYPD Shooting Incident Data (Historic), the following URL can be utilized for importing:

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

Tidying and transforming data

In order to facilitate the analysis of imported data, it is necessary to perform the process of tidying and transforming. This involves reshaping the data into a structured format that is suitable for analysis, and creating new columns that can provide relevant information for the subsequent analytical tasks.

```r
## Define a helper function to categorize time of the day
get_time_day <- function(hour) {
  time_day <- cut(hour,c(-Inf, 6, 12, 18, Inf),
                  labels = c("midnight", "morning", "afternoon", "evening"))
  return(as.character(time_day))
}
```

```r
incident <- incident %>%
  # Remove duplicate data
  distinct() %>%
  # Convert date to date object
  mutate(DATE = mdy(OCCUR_DATE), .before = 1) %>%
  # Extract year component
  mutate(YEAR = year(DATE), .after = DATE) %>%
  # Extract month component
  mutate(MONTH = month(DATE), .after = YEAR) %>%
  # Create a column for weekday
  mutate(WEEKDAY = wday(DATE, label = TRUE), .after = MONTH) %>%
  # Create a column for time of the day
  mutate(TIME_DAY = get_time_day(hour(hms(OCCUR_TIME))), .after = OCCUR_TIME) %>%
  # change blank data to NA
  mutate_all(~na_if(., '')) %>%
  # Deselect unwanted columns
  select(-c(Latitude, Longitude, X_COORD_CD, LOCATION_DESC, JURISDICTION_CODE,
          Y_COORD_CD, Lon_Lat, INCIDENT_KEY, OCCUR_DATE))

# Convert character to numeric value
incident$STATISTICAL_MURDER_FLAG <- as.integer(as.logical(incident$STATISTICAL_MURDER_FLAG))

# show summary of the data
summary(incident)
```

```
##       DATE                 YEAR          MONTH         WEEKDAY
##  Min.   :2006-01-01   Min.   :2006   Min.   : 1.000   Sun:5156
##  1st Qu.:2009-05-10   1st Qu.:2009   1st Qu.: 5.000   Mon:3597
##  Median :2012-08-26   Median :2012   Median : 7.000   Tue:2945
##  Mean   :2013-06-13   Mean   :2013   Mean   : 6.857   Wed:2818
##  3rd Qu.:2017-07-01   3rd Qu.:2017   3rd Qu.: 9.000   Thu:2809
##  Max.   :2021-12-31   Max.   :2021   Max.   :12.000   Fri:3384
##                                                       Sat:4887
##   OCCUR_TIME          TIME_DAY             BORO              PRECINCT
##  Length:25596       Length:25596       Length:25596       Min.   :  1.00
##  Class :character   Class :character   Class :character   1st Qu.: 44.00
##  Mode  :character   Mode  :character   Mode  :character   Median : 69.00
##                                                           Mean   : 65.87
##                                                           3rd Qu.: 81.00
##                                                           Max.   :123.00
##
##  STATISTICAL_MURDER_FLAG PERP_AGE_GROUP       PERP_SEX
##  Min.   :0.0000          Length:25596       Length:25596
##  1st Qu.:0.0000          Class :character   Class :character
##  Median :0.0000          Mode  :character   Mode  :character
##  Mean   :0.1925
```

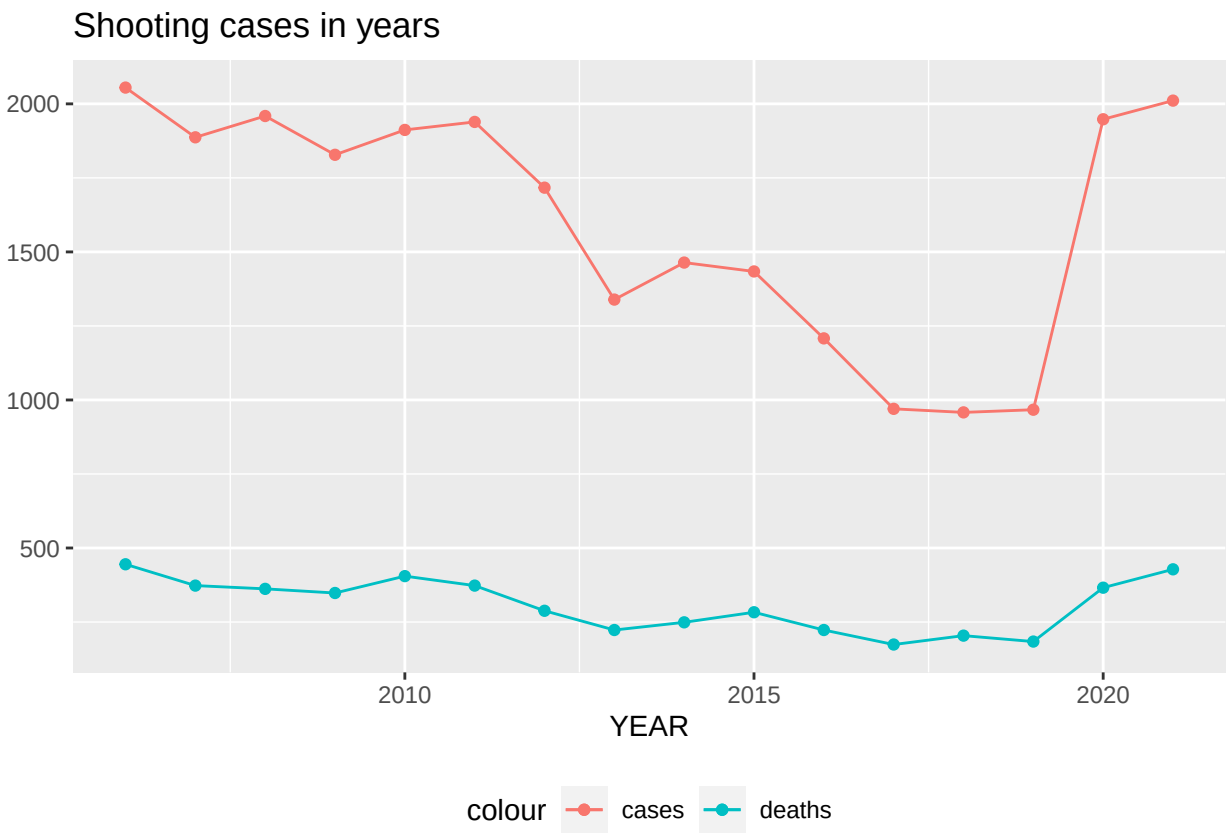
```
##  3rd Qu.:0.0000
##  Max.   :1.0000
##
##   PERP_RACE         VIC_AGE_GROUP        VIC_SEX            VIC_RACE
##  Length:25596       Length:25596       Length:25596       Length:25596
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

Data Analysis

General Analysis

To begin the analysis, the data may be stratified by year, and the total number of cases and deaths per year can be computed.

```
incident_by_year <- incident %>%
  # Group data by year
  group_by(YEAR) %>%
  summarize(CASES = n(), DEATHS = sum(STATISTICAL_MURDER_FLAG)) %>%
  mutate(PERCENT_DEATH = DEATHS/CASES*100)

# Plot a graph with cases and deaths
incident_by_year %>%
  ggplot(aes(x = YEAR, y = CASES)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = DEATHS, color = "deaths")) +
  geom_point(aes(y = DEATHS, color = "deaths")) +
  theme(legend.position = "bottom") +
  labs(title = "Shooting cases in years", y = NULL)
```

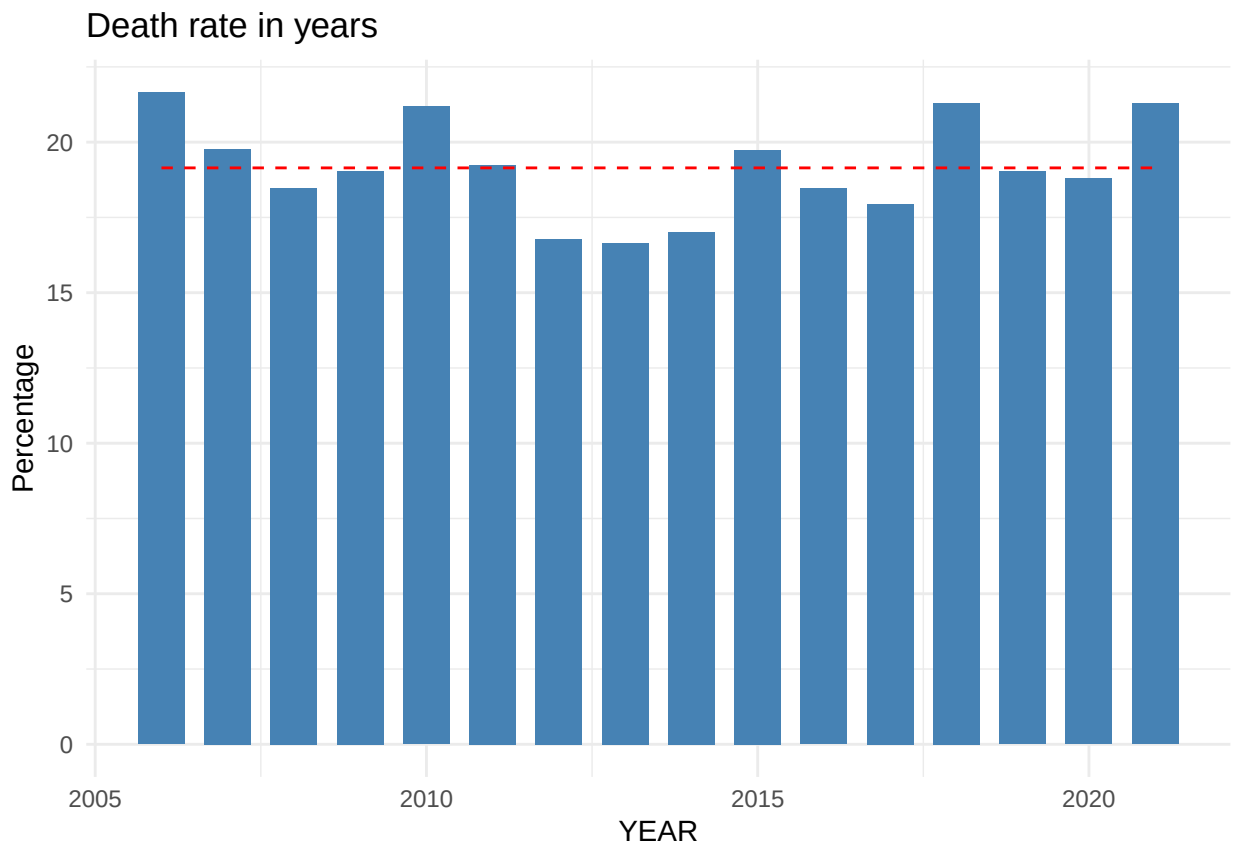


The chart in the 'Shooting cases in years' tab indicates that the total number of shooting cases gradually decreased from 2006 to 2019. However, there was a sharp increase in the number of cases after 2019. The total number of deaths followed a similar pattern to the total number of shooting cases.

Death analysis

Based on the earlier analysis, the death rate can be visualized through a graph by plotting the relevant data in the following way:

```
# Plot bar graph
incident_by_year %>%
  ggplot(aes(x = YEAR, y = PERCENT_DEATH)) +
  # Draw the bar
  geom_bar(width = 0.7, stat = "identity", fill="steelblue") +
  # Draw a dashed line showing the average death rate
  geom_line(aes(y = mean(PERCENT_DEATH)), color="red", linetype = "dashed") +
  labs(title = "Death rate in years", y = "Percentage") +
  theme_minimal()
```



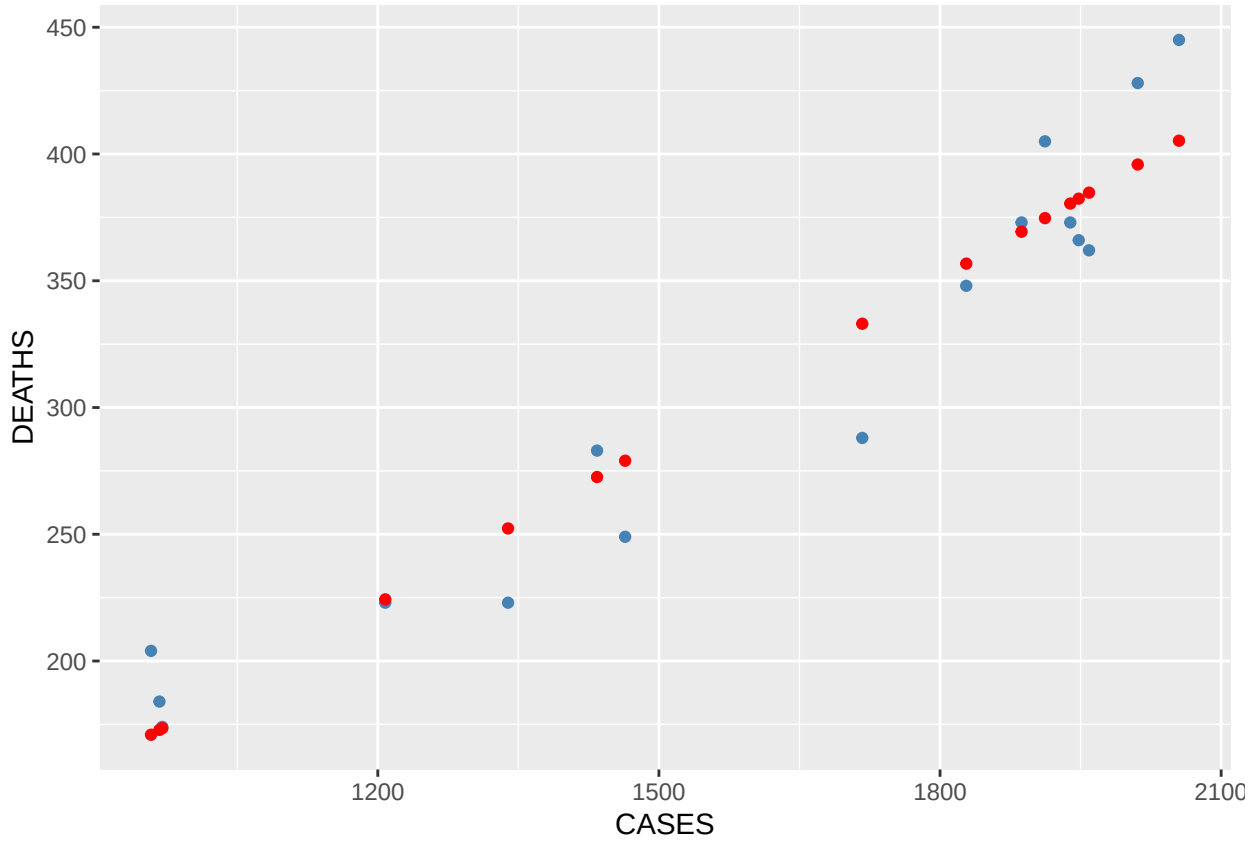
The bar chart displaying the death rate changes in time series demonstrates that the death rate in the past 15 years remained relatively stable at around 19% (red dashed line), which implies that there was approximately one death reported in every five shooting cases. This suggests that a linear relationship between the number of deaths and the number of shooting cases should be considered.

```
# Create a linear model
mod_year <- lm(DEATHS ~ CASES, data = incident_by_year)
# Show the summary of the model
summary(mod_year)
```

```
##
## Call:
## lm(formula = DEATHS ~ CASES, data = incident_by_year)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.047 -17.980  -0.395  15.950  39.750
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -33.73553   27.45209  -1.229    0.239
## CASES         0.21362    0.01667  12.817    4e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.16 on 14 degrees of freedom
```

```
## Multiple R-squared:  0.9215, Adjusted R-squared:  0.9159
## F-statistic: 164.3 on 1 and 14 DF,  p-value: 3.996e-09
```

```
# Plot a graph to show the relationship and prediction
incident_by_year_w_pred <- incident_by_year %>% mutate(pred = predict(mod_year))
incident_by_year_w_pred %>%
  ggplot() +
  geom_point(aes(x = CASES, y = DEATHS), color ="steelblue") +
   geom_point(aes(x = CASES, y = pred), color ="red")
```



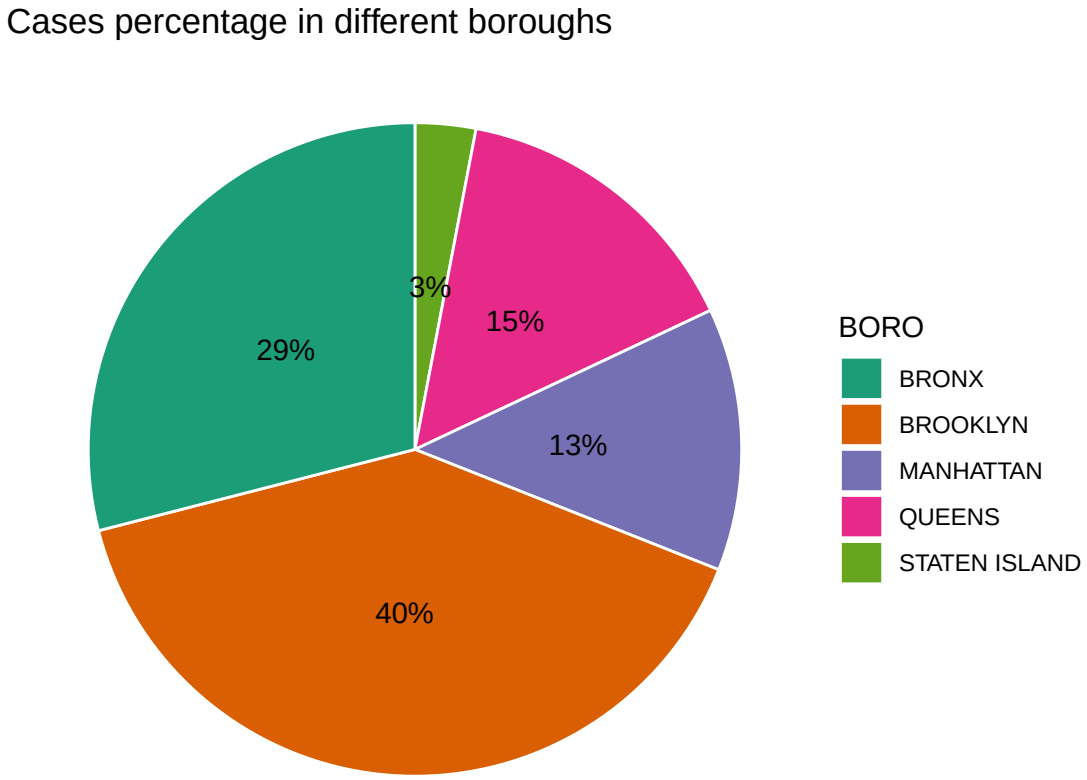
The scatter plot reveals a linear relationship between the number of deaths and the number of cases, with an estimated coefficient rate of around 0.21362.

Borough analysis

For a more detailed analysis, the data can be grouped by boroughs. The first pie chart illustrates the distribution of cases among the different boroughs, while the second pie chart depicts the distribution of deaths.

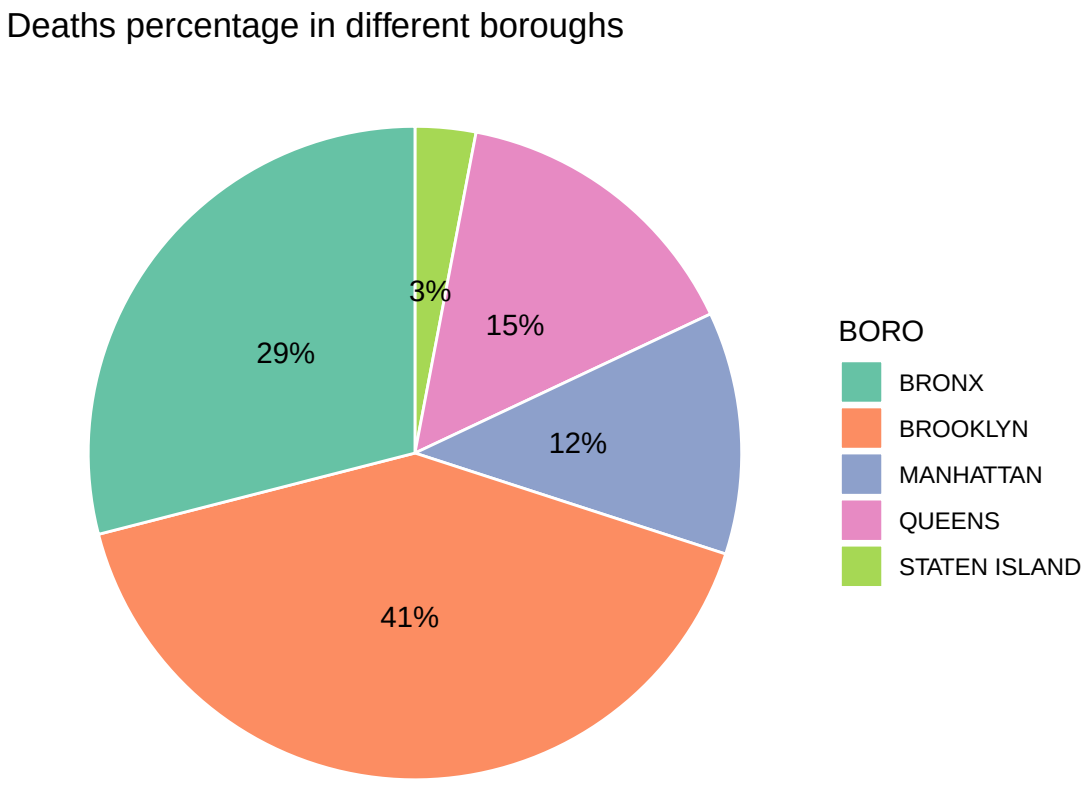
```
incident_by_boro <- incident %>%
  # Group data by borough
  group_by(BORO) %>%
  summarize(CASES = n(), DEATHS = sum(STATISTICAL_MURDER_FLAG)) %>%
  mutate(PERCENT_DEATH = DEATHS/CASES*100) %>%
  # Calculate the percentage
```

```
  mutate(prop = round(CASES/sum(CASES)*100)) %>%
  # Arrange it in descending order
  arrange(desc(BORO)) %>%
  # Find the label position
  mutate(lab.ypos = cumsum(prop) - 0.5*prop) %>%
  # Create label
  mutate(lab = paste(as.character(prop),"%", sep = ""))

# Plot case data
incident_by_boro %>%
  # Create pie chart
  ggplot(aes(x="", y = prop, fill = BORO)) +
  geom_bar(width = 1, stat = "identity",color = "white") +
  coord_polar("y", start = 0) +
  # Add label
  geom_text(aes(y = lab.ypos, label = lab)) +
  # Add color
  scale_fill_brewer(palette = "Dark2") +
  # Add title
  labs(title = "Cases percentage in different boroughs") +
  # Hide axis
  theme_void()
```




```
# Create death data
death_by_boro <- incident %>%
```

```
  group_by(BORO) %>%
  summarize(CASES = n(), DEATHS = sum(STATISTICAL_MURDER_FLAG)) %>%
  mutate(PERCENT_DEATH = DEATHS/DEATHS*100) %>%
  mutate(prop = round(DEATHS/sum(DEATHS)*100)) %>%
  arrange(desc(BORO)) %>%
  mutate(lab.ypos = cumsum(prop) - 0.5*prop) %>%
  mutate(lab = paste(as.character(prop),"%", sep = ""))

# Plot death data
death_by_boro %>%
  ggplot(aes(x="", y = prop, fill = BORO))+
  geom_bar(width = 1, stat = "identity",color = "white") +
  coord_polar("y", start = 0) +
  geom_text(aes(y = lab.ypos, label = lab)) +
  scale_fill_brewer(palette = "Set2") +
  labs(title = "Deaths percentage in different boroughs") +
  theme_void()
```



The pie chart illustrates that the majority of shooting cases occurred in Brooklyn and Bronx, which requires more attention from the police. Brooklyn and Bronx also reported the most deaths in the past 15 years. 41% of the death cases were reported from Brooklyn, and 29% were reported from the Bronx.

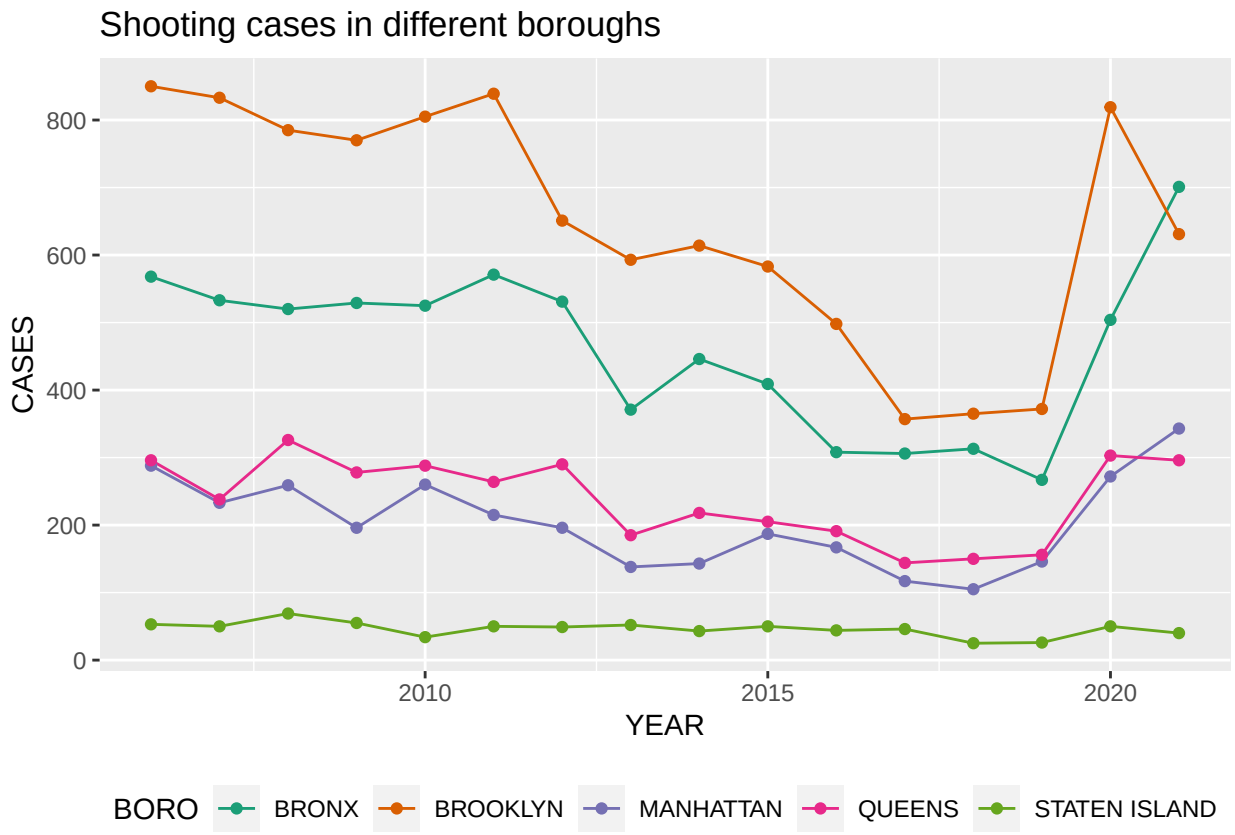
To conduct a more comprehensive analysis, a line graph is generated as follows:

```
incident_by_year_boro <- incident %>%
  # Group data by year and borough
```

```
  group_by(YEAR, BORO) %>%
  summarize(CASES = n(), DEATHS = sum(STATISTICAL_MURDER_FLAG))
```

```
## `summarise()` has grouped output by 'YEAR'. You can override using the
## `.groups` argument.
```

```
incident_by_year_boro %>%
  ggplot(aes(x = YEAR, y = CASES)) +
  # Draw lines and points for each borough
  geom_line(aes(color = BORO)) +
  geom_point(aes(color = BORO)) +
  theme(legend.position = "bottom") +
  labs(title = "Shooting cases in different boroughs") +
  scale_color_brewer(palette="Dark2")
```

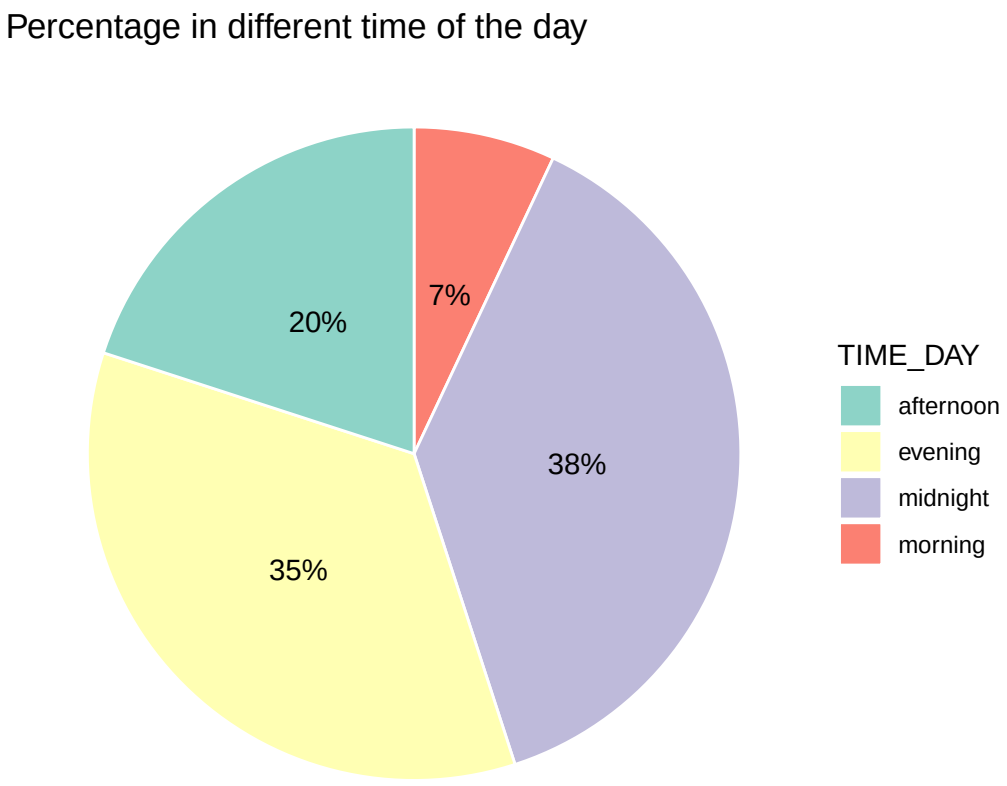


The line graphs of the total number of shooting cases in different boroughs during the past 15 years showed similar trends. However, the number of shooting cases in Brooklyn increased more rapidly after 2019 compared to other boroughs and dropped off after 2022. In contrast, the number of shooting cases in the Bronx continued to increase after 2019.

Time Analysis

For further examination, the data can be classified based on the time of day. The time can be divided into four categories: 0-6 (midnight), 6-12 (morning), 12-18 (afternoon), and 18-0 (evening). A pie chart can be plotted to display the corresponding data.

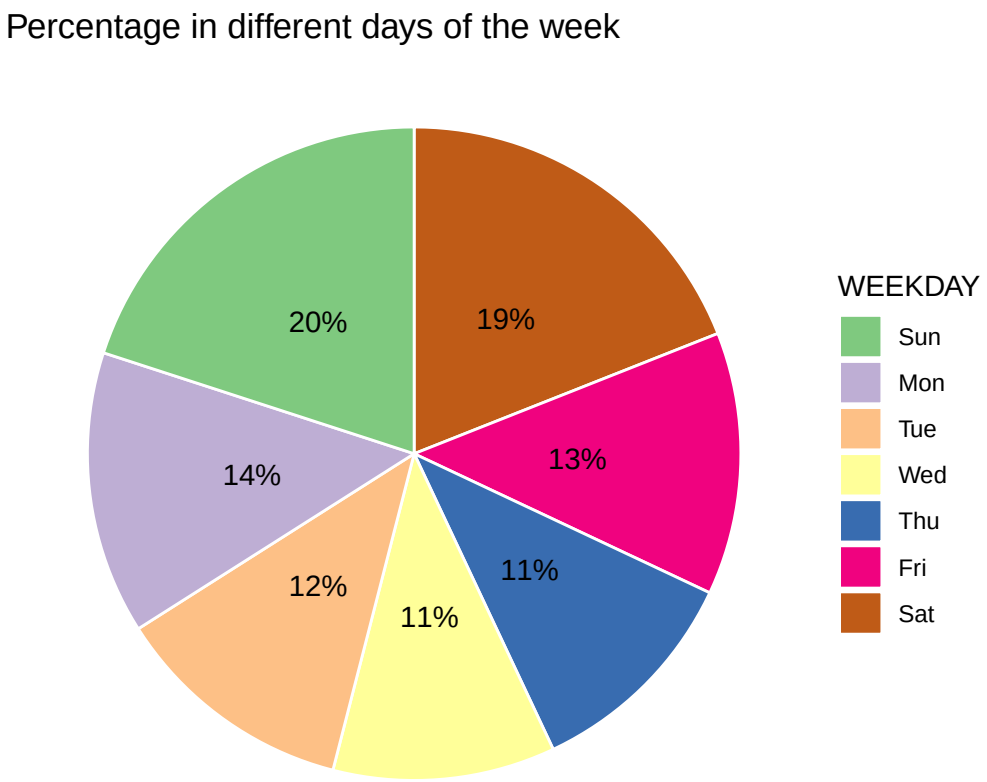
```r
incident_by_time <- incident %>%
  # Group data by time of the day
  group_by(TIME_DAY) %>%
  summarize(CASES = n(), DEATHS = sum(STATISTICAL_MURDER_FLAG)) %>%
  mutate(PERCENT_DEATH = DEATHS/CASES*100) %>%
  mutate(prop = round(CASES/sum(CASES)*100)) %>%
  arrange(desc(TIME_DAY)) %>%
  mutate(lab.ypos = cumsum(prop) - 0.5*prop) %>%
  mutate(lab = paste(as.character(prop),"%", sep = ""))

# Create a pie chart for different times of the day
incident_by_time %>%
  ggplot(aes(x="", y = prop, fill = TIME_DAY))+
  geom_bar(width = 1, stat = "identity",color = "white") +
  coord_polar("y", start = 0) +
  geom_text(aes(y = lab.ypos, label = lab)) +
  scale_fill_brewer(palette = "Set3") +
  labs(title = "Percentage in different time of the day") +
  theme_void()
```



The pie chart showing the distribution of the number of cases during the day indicates that most shootings occurred in the evening and at midnight (from 6 pm to 6 am).

As a second approach to analyze the data, we may consider exploring the distribution of shooting cases by weekdays. To visualize this, we can create a pie chart that displays the percentage of shooting cases for each day of the week.

```r
incident_by_wday <- incident %>%
  # Group data by day of the week
  group_by(WEEKDAY) %>%
  summarize(CASES = n(), DEATHS = sum(STATISTICAL_MURDER_FLAG)) %>%
  mutate(PERCENT_DEATH = DEATHS/CASES*100) %>%
  mutate(prop = round(CASES/sum(CASES)*100)) %>%
  arrange(desc(WEEKDAY)) %>%
  mutate(lab.ypos = cumsum(prop) - 0.5*prop) %>%
  mutate(lab = paste(as.character(prop),"%", sep = ""))

# Create a pie chart for each day of the week
incident_by_wday %>%
  ggplot(aes(x="", y = prop, fill = WEEKDAY))+
  geom_bar(width = 1, stat = "identity",color = "white") +
  coord_polar("y", start = 0) +
  geom_text(aes(y = lab.ypos, label = lab)) +
  scale_fill_brewer(palette = "Accent") +
  labs(title = "Percentage in different days of the week") +
  theme_void()
```

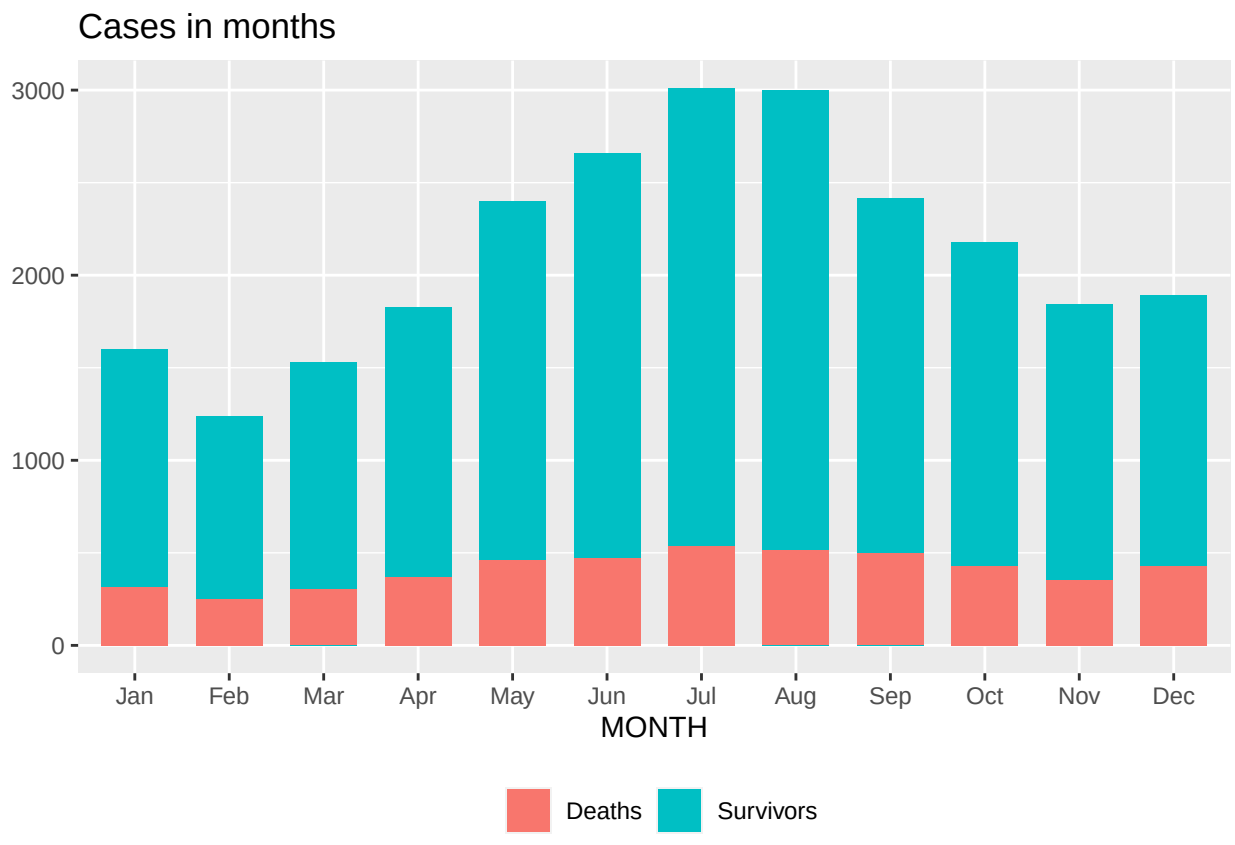


The pie chart displaying the distribution of the number of cases during a week shows that shootings were more likely to occur on weekends than on weekdays.

Month analysis

Another interesting approach would be to conduct a monthly analysis of the data to examine whether there are any seasonal patterns in the occurrence of shooting incidents, thus providing an additional perspective for analysis.

```
incident_by_month <- incident %>%
  # Group data by time of the day
  group_by(MONTH) %>%
  summarize(CASES = n(), DEATHS = sum(STATISTICAL_MURDER_FLAG)) %>%
  mutate(PERCENT_DEATH = DEATHS/CASES*100) %>%
  mutate(MONTH_ABBR = as.character(month(MONTH ,label=TRUE,abbr=TRUE)), .after = MONTH) %>%
  arrange(MONTH)

incident_by_month %>%
  ggplot(aes(x = MONTH, y = CASES)) +
  # Create two different bars
  geom_bar(aes(fill = "Survivors"), width = 0.7, stat = "identity") +
  geom_bar(aes(y = DEATHS, fill = "Deaths"), width = 0.7, stat = "identity") +
  scale_x_discrete(limits = incident_by_month$MONTH_ABBR) +
  labs(title = "Cases in months", fill = NULL, y = NULL) +
  theme(legend.position = "bottom")
```



An examination of the stacked bar graph reveals that the frequency of shooting incidents is noticeably higher during the summer season and experiences a sharp decline during the winter months.

Summary

This data analysis delves into various facets of NYPD shooting incident data, including general analysis, death analysis, borough analysis, time analysis, and weekday analysis. The report is accompanied by charts and graphs that depict trends in the data, such as the gradual decline in the number of shooting cases and deaths from 2006 to 2019, a sharp increase in the number of cases after 2019, the correlation between the number of deaths and shooting cases, the distribution of shooting cases and deaths across different boroughs, and the distribution of shooting cases by time of day and day of the week.

Bias Identification

Although the current analysis provides some insights, several other areas could be investigated for a more comprehensive understanding of the data. For instance, examining the demographic information of those involved in the shootings could reveal any patterns or disparities.

Additionally, plotting the geographic location of the shootings on a map could highlight hotspots or patterns in the distribution of incidents. Another interesting area to explore is the relationship between the location of the shootings and the socioeconomic status of the surrounding area.

Furthermore, comparing the NYPD Shooting Incident Data to similar data from other cities could help identify unique patterns or trends. Meanwhile, conducting a sentiment analysis of media coverage of the incidents could offer insights into public perception and potential biases.

By exploring these areas and others, a more thorough and nuanced understanding of the NYPD Shooting Incident Data could be gained. Such insights could lead to more effective policy decisions and interventions to address the root causes of gun violence in New York City.

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_Canada.utf8  LC_CTYPE=English_Canada.utf8
## [3] LC_MONETARY=English_Canada.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_Canada.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.0  timechange_0.1.1 forcats_0.5.2    stringr_1.5.0
##  [5] dplyr_1.0.10     purrr_1.0.0      readr_2.1.3      tidyr_1.2.1
##  [9] tibble_3.1.8     ggplot2_3.4.0    tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0    xfun_0.36           haven_2.5.1
##  [4] gargle_1.2.1        colorspace_2.0-3    vctrs_0.5.1
##  [7] generics_0.1.3      htmltools_0.5.4     yaml_2.3.6
## [10] utf8_1.2.2          rlang_1.0.6         pillar_1.8.1
## [13] withr_2.5.0         glue_1.6.2          DBI_1.1.3
```

```
## [16] RColorBrewer_1.1-3  dbplyr_2.2.1        modelr_0.1.10
## [19] readxl_1.4.1        lifecycle_1.0.3     munsell_0.5.0
## [22] gtable_0.3.1        cellranger_1.1.0    rvest_1.0.3
## [25] evaluate_0.19       labeling_0.4.2      knitr_1.41
## [28] tzdb_0.3.0          fastmap_1.1.0       fansi_1.0.3
## [31] highr_0.10          broom_1.0.2         scales_1.2.1
## [34] backports_1.4.1     googlesheets4_1.0.1 jsonlite_1.8.4
## [37] farver_2.1.1        fs_1.5.2            hms_1.1.2
## [40] digest_0.6.31       stringi_1.7.8       grid_4.2.2
## [43] cli_3.4.1           tools_4.2.2         magrittr_2.0.3
## [46] crayon_1.5.2        pkgconfig_2.0.3     ellipsis_0.3.2
## [49] xml2_1.3.3          reprex_2.0.2        googledrive_2.0.0
## [52] assertthat_0.2.1    rmarkdown_2.19      httr_1.4.4
## [55] rstudioapi_0.14     R6_2.5.1            compiler_4.2.2
```