ANALYZING HISTORICAL NYPD SHOOTING DATA

Anonymous

23/05/2021

IMPORTING THE DATA

We are looking to analyse every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year. Using the following source.

```
#import data
nypd <- read.csv("NYPD_Shooting_Incident_Data__Historic_.csv")
head(nypd)
```

Initially from a quick analysis of the data, we can use this data to look at the approximate likelihood of crime in a borough in NYC.

#Looking at some of the data: We are looking at a data.frame with 23568 observations of 19 variables:

```
summary(nypd)
head(nypd)
class(nypd)
names(nypd)
str(nypd)
```

```
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
describe(nypd$OCCUR_DATE)
```

```
## nypd$OCCUR_DATE
##        n  missing distinct
##    23568        0     5054
##
## lowest : 01/01/2006 01/01/2007 01/01/2008 01/01/2009 01/01/2010
## highest: 12/31/2016 12/31/2017 12/31/2018 12/31/2019 12/31/2020
```

```
describe(nypd$VIC_SEX)
```

```
## nypd$VIC_SEX
##        n  missing distinct
##    23568        0        3
##
## Value          F     M     U
## Frequency   2195 21353    20
## Proportion 0.093 0.906 0.001
```

```
describe(nypd$PERP_SEX)
```

```
## nypd$PERP_SEX
##        n  missing distinct
##    15143     8425        3
##
## Value          F     M     U
## Frequency    334 13305  1504
## Proportion 0.022 0.879 0.099
```

Changing up the format of Occurrence date so we can use it for analysis:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:Hmisc':
##
##     src, summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(magrittr)

nypd$DATE <- as.Date(nypd$OCCUR_DATE,
                     format = "%m/%d/%y")
head(nypd)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME          BORO PRECINCT JURISDICTION_CODE
## 1    201575314 08/23/2019   22:10:00        QUEENS      103                 0
## 2    205748546 11/27/2019   15:54:00         BRONX       40                 0
## 3    193118596 02/02/2019   19:40:00     MANHATTAN       23                 0
## 4    204192600 10/24/2019   00:52:00 STATEN ISLAND      121                 0
## 5    201483468 08/22/2019   18:03:00         BRONX       46                 0
## 6    198255460 06/07/2019   17:50:00      BROOKLYN       73                 0
##   LOCATION_DESC STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX      PERP_RACE
## 1                                false                                       
## 2                                false            <18        M          BLACK
## 3                                false          18-24        M WHITE HISPANIC
## 4     PVT HOUSE                   true          25-44        M          BLACK
## 5                                false          25-44        M BLACK HISPANIC
## 6                                false          45-64        M WHITE HISPANIC
##   VIC_AGE_GROUP VIC_SEX       VIC_RACE X_COORD_CD Y_COORD_CD Latitude Longitude
## 1         25-44       M          BLACK    1037451     193561 40.69781 -73.80814
## 2         25-44       F          BLACK    1006789     237559 40.81870 -73.91857
## 3         18-24       M BLACK HISPANIC     999347     227795 40.79192 -73.94548
## 4         25-44       F          BLACK     938149     171781 40.63806 -74.16611
## 5         18-24       M          BLACK    1008224     250621 40.85455 -73.91334
## 6         25-44       M          BLACK    1009650     186966 40.67983 -73.90843
##                                      Lon_Lat       DATE
## 1 POINT (-73.80814071699996 40.697805308000056) 2020-08-23
## 2  POINT (-73.91857061799993 40.81869973000005) 2020-11-27
## 3 POINT (-73.94547965999999 40.791916091000076) 2020-02-02
## 4  POINT (-74.16610830199996 40.63806398200006) 2020-10-24
## 5  POINT (-73.91333944399999 40.85454734900003) 2020-08-22
## 6  POINT (-73.90842523899994 40.67982701600005) 2020-06-07
```

#changing logical boolean into integer for STATISTICAL_MURDER_FLAG TO INDICATE WHETHER THE SHOOTING WAS FATAL OR NOT

```
library(dplyr)
library(ggplot2)
#changing logical boolean into integer
murder=nypd$STATISTICAL_MURDER_FLAG[nypd$STATISTICAL_MURDER_FLAG=="TRUE"]<-1#indicates fatality
shooting=nypd$STATISTICAL_MURDER_FLAG[nypd$STATISTICAL_MURDER_FLAG=="FALSE"]<-0#indicates non-fatality
```

There are some values noted in PERP_RACE and VIC_RACE as "UNKNOWN" which are missing so we want to remove those values from the dataset.

```
nypd$VIC_RACE[nypd$VIC_RACE == "UNKNOWN"] <- NA
nypd$PERP_RACE[nypd$PERP_RACE == "UNKNOWN"] <- NA
nypd$VIC_AGE_GROUP[nypd$VIC_AGE_GROUP == "UNKNOWN"] <- NA
nypd$PERP_AGE_GROUP[nypd$PERP_AGE_GROUP == "UNKNOWN"] <- NA
na.omit(nypd)
```

Now the data looks more complete.

```
print(nypd)
```

Now lets make some categoricals usable for the analysis:

```
nypd$BORO=factor(nypd$BORO,levels=c("MANHATTAN","BROOKLYN","QUEENS","BRONX","STATEN ISLAND"))
nypd$PERP_RACE= factor(nypd$PERP_RACE,levels=c("BLACK","ASIAN/PACIFIC ISLANDER","WHITE", "WHITE HISPANI
nypd$VIC_RACE=factor(nypd$VIC_RACE,levels=c("BLACK","ASIAN/PACIFIC ISLANDER","WHITE", "WHITE HISPANIC"))
nypd$PERP_AGE_GROUP=factor(nypd$PERP_AGE_GROUP,levels=c("<18", "18-24", "25-44","45-64","65+"))
nypd$VIC_AGE_GROUP=factor(nypd$VIC_AGE_GROUP,levels=c("<18", "18-24", "25-44","45-64","65+"))

head(nypd)
```

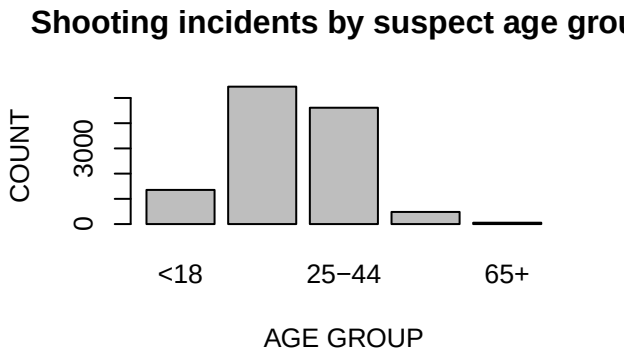
```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME          BORO PRECINCT JURISDICTION_CODE
## 1    201575314 08/23/2019   22:10:00        QUEENS      103                 0
## 2    205748546 11/27/2019   15:54:00         BRONX       40                 0
## 3    193118596 02/02/2019   19:40:00     MANHATTAN       23                 0
## 4    204192600 10/24/2019   00:52:00 STATEN ISLAND      121                 0
## 5    201483468 08/22/2019   18:03:00         BRONX       46                 0
## 6    198255460 06/07/2019   17:50:00      BROOKLYN       73                 0
##   LOCATION_DESC STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX      PERP_RACE
## 1                                false           <NA>                   <NA>
## 2                                false            <18        M          BLACK
## 3                                false          18-24        M WHITE HISPANIC
## 4     PVT HOUSE                   true          25-44        M          BLACK
## 5                                false          25-44        M           <NA>
## 6                                false          45-64        M WHITE HISPANIC
##   VIC_AGE_GROUP VIC_SEX VIC_RACE X_COORD_CD Y_COORD_CD Latitude Longitude
## 1         25-44       M    BLACK    1037451     193561 40.69781 -73.80814
## 2         25-44       F    BLACK    1006789     237559 40.81870 -73.91857
## 3         18-24       M     <NA>     999347     227795 40.79192 -73.94548
## 4         25-44       F    BLACK     938149     171781 40.63806 -74.16611
## 5         18-24       M    BLACK    1008224     250621 40.85455 -73.91334
## 6         25-44       M    BLACK    1009650     186966 40.67983 -73.90843
##                                      Lon_Lat       DATE
## 1 POINT (-73.80814071699996 40.697805308000056) 2020-08-23
## 2  POINT (-73.91857061799993 40.81869973000005) 2020-11-27
## 3 POINT (-73.94547965999999 40.791916091000076) 2020-02-02
## 4  POINT (-74.16610830199996 40.63806398200006) 2020-10-24
## 5  POINT (-73.91333944399999 40.85454734900003) 2020-08-22
## 6  POINT (-73.90842523899994 40.67982701600005) 2020-06-07
```

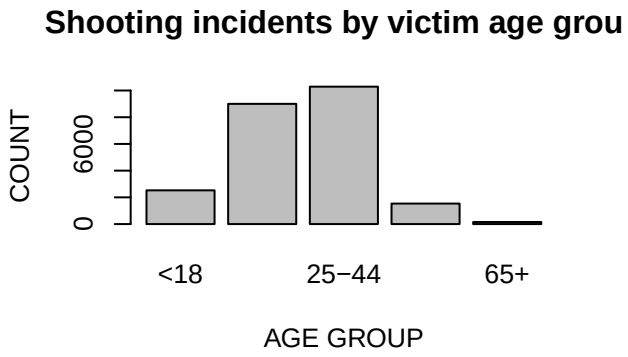
##Tidying and Transforming Data Now lets make remove some irrelvant columns for the analysis:

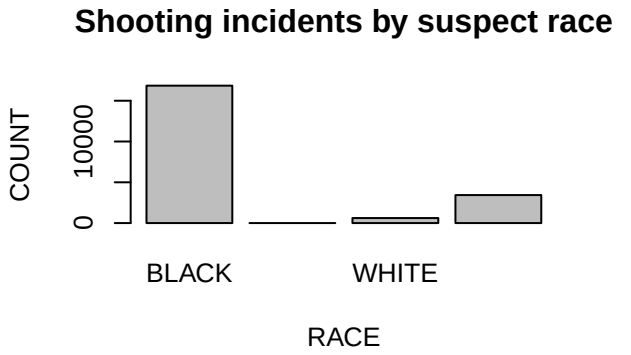
```
nypd$JURISDICTION_CODE<- NULL
nypd$count.2<- NULL
nypd$count<- NULL
print(nypd)
```

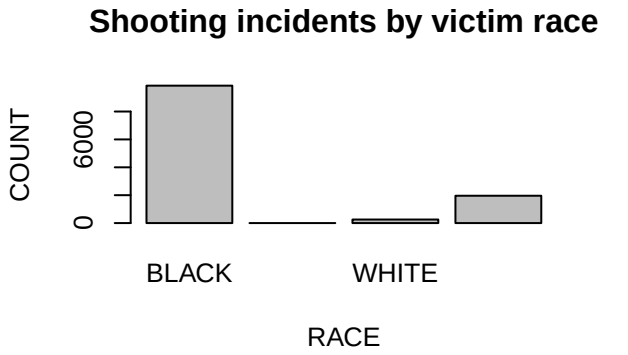
##Visualizing the data and modelling the data

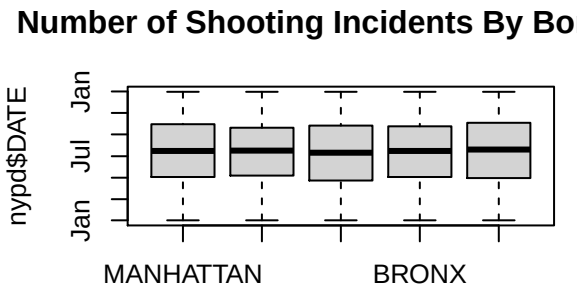
Looking at some of the data after being cleaned, we can come up with some visualizations as follows:

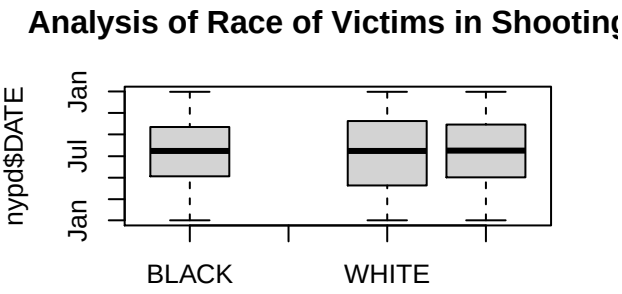

Shooting incidents by suspect age grou
COUNT
0
3000
<18
25−44
65+
AGE GROUP

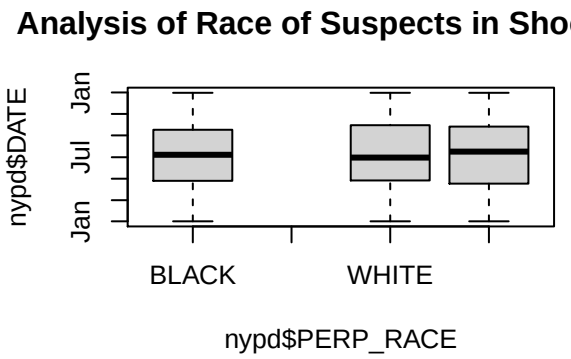

Shooting incidents by victim age grou
COUNT
0
6000
<18
25−44
65+
AGE GROUP


Shooting incidents by suspect race
COUNT
0
10000
BLACK
WHITE
RACE


Shooting incidents by victim race
COUNT
0
6000
BLACK
WHITE
RACE







##MODEL ANALYSIS

Now that we have a clear sense of the data. We can complete some analysis to determine how this data can be applied in real world applications. We can determine if the trend of shooting data has increased or decreased over time.

```
library(dbplyr)
```

```
##
## Attaching package: 'dbplyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
library(lubridate)
```

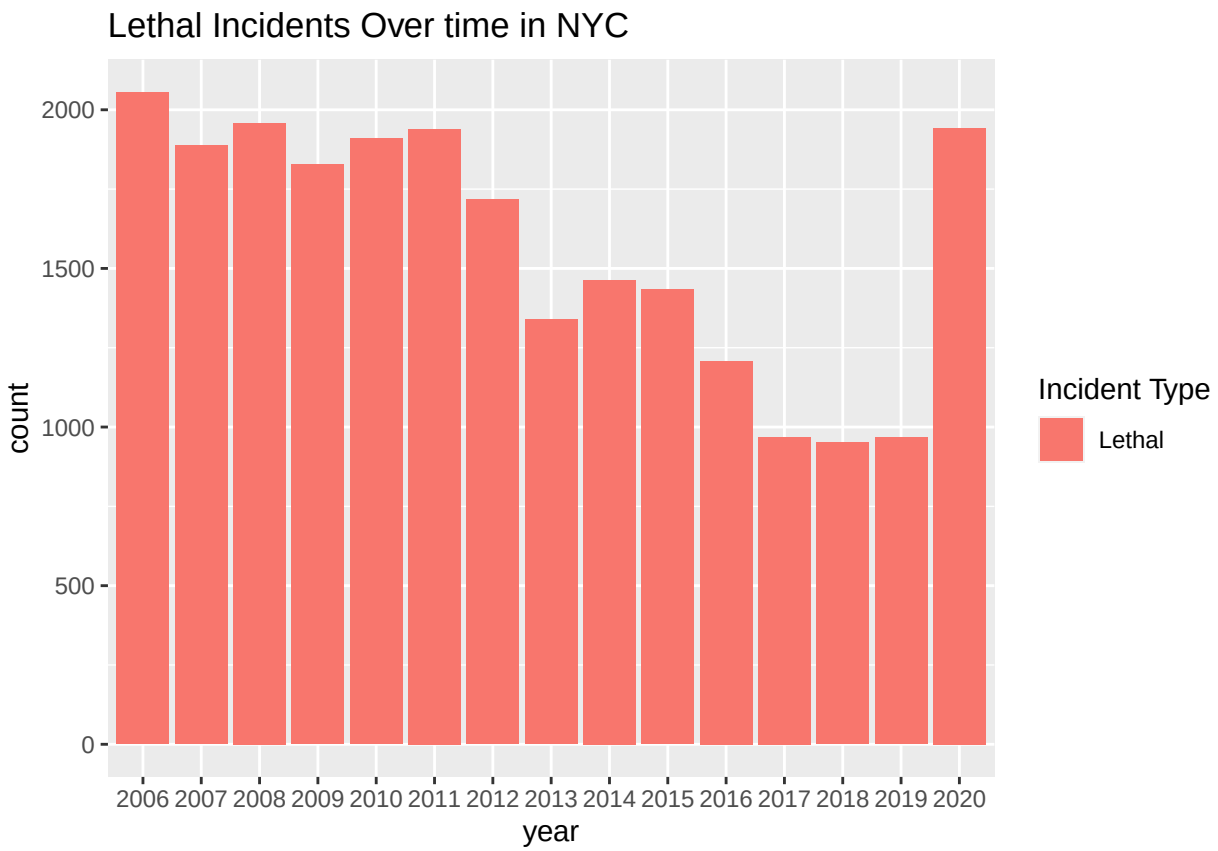
```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(tidyverse)
```

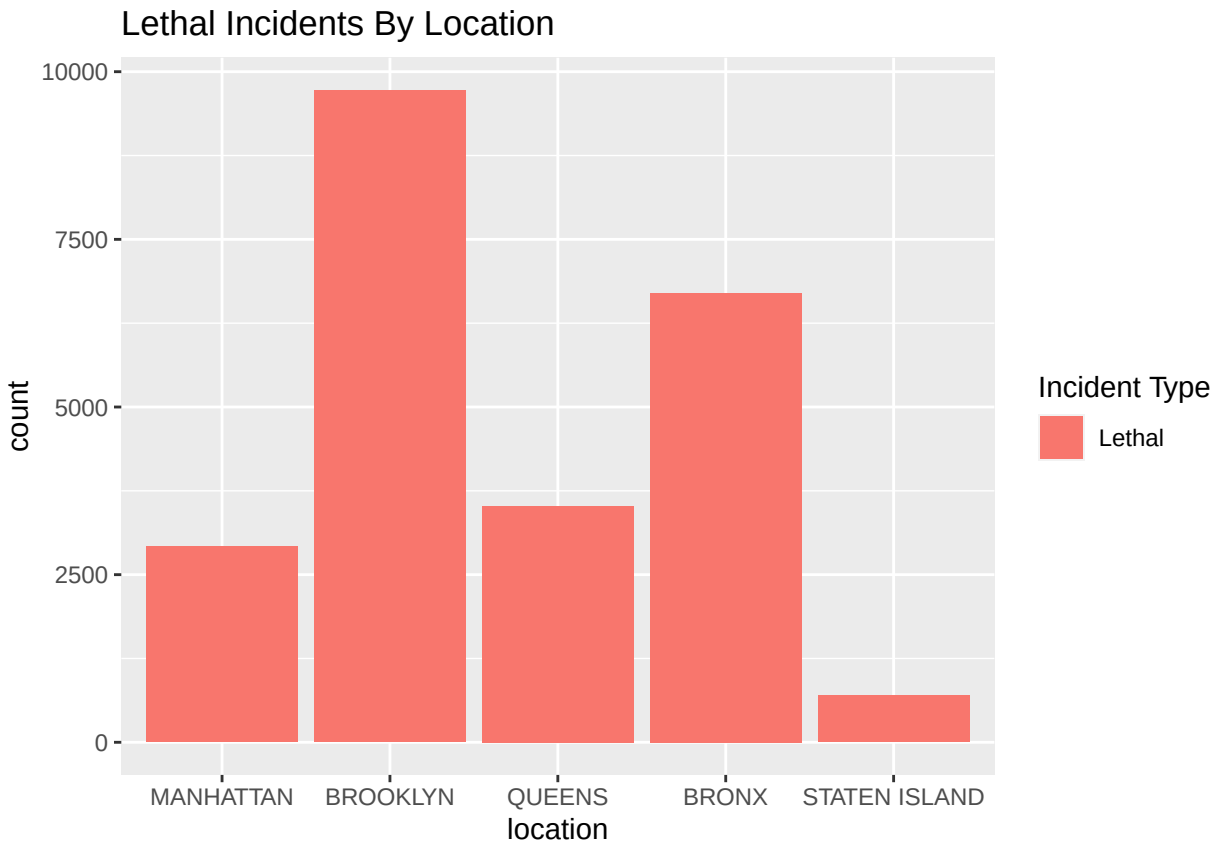
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.1     v purrr   0.3.4
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x tidyr::extract()         masks magrittr::extract()
## x dplyr::filter()          masks stats::filter()
## x dbplyr::ident()          masks dplyr::ident()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x purrr::set_names()       masks magrittr::set_names()
## x lubridate::setdiff()     masks base::setdiff()
## x dbplyr::sql()            masks dplyr::sql()
## x dplyr::src()             masks Hmisc::src()
## x dplyr::summarize()       masks Hmisc::summarize()
## x lubridate::union()       masks base::union()
```

```
df <- nypd %>% rename(incident_type = STATISTICAL_MURDER_FLAG) %>% mutate(year=substr(OCCUR_DATE,7,10))
  select(year, incident_type) %>%
  mutate(incident_type= ifelse(incident_type ==FALSE,"Non-Lethal","Lethal"))

ggplot(df, aes(x = year, fill = incident_type)) +
  geom_bar(position = position_dodge(preserve="single")) +
  labs(title="Lethal Incidents Over time in NYC", fill ="Incident Type")
```



In terms of real world applications, lets think of the real estate industry or tourism industry. If you were looking to own property in NYC and were researching the safety of each borough, the following data would be useful to analyse.

```
library(dbplyr)
library(lubridate)
library(tidyverse)
BORO=nypd$BORO=factor(nypd$BORO,levels=c("MANHATTAN","BROOKLYN","QUEENS","BRONX","STATEN ISLAND"))
df1 <- nypd %>% rename(incident_type= STATISTICAL_MURDER_FLAG) %>%
  mutate(location=BORO) %>%
  select(location, incident_type) %>%
  mutate(incident_type= ifelse(incident_type == FALSE,"Non-Lethal","Lethal"))

ggplot(df1, aes(x = location, fill = incident_type)) +
  geom_bar(position = position_dodge(preserve="single")) +
  labs(title="Lethal Incidents By Location", fill ="Incident Type")
```



From this analysis, we can see that Queens and Staten Island has a lesser likelihood of gun violence. Therefore we can use this data to determine the best place to own property or plan a visit for sightseeing.

```
summary(df1)
```

```
##           location    incident_type
##  MANHATTAN    :2921   Length:23568
##  BROOKLYN     :9722   Class :character
##  QUEENS       :3527   Mode  :character
##  BRONX        :6700
##  STATEN ISLAND: 698
```

```
describe(df1)
```

```
## df1
##
##  2  Variables      23568  Observations
## ---------------------------------------------------------------------------
## location
##        n  missing distinct
##    23568        0        5
##
## lowest : MANHATTAN     BROOKLYN      QUEENS        BRONX         STATEN ISLAND
## highest: MANHATTAN     BROOKLYN      QUEENS        BRONX         STATEN ISLAND
##
## Value          MANHATTAN      BROOKLYN        QUEENS         BRONX
```

```
## Frequency            2921          9722          3527          6700
## Proportion          0.124         0.413         0.150         0.284
##
## Value      STATEN ISLAND
## Frequency            698
## Proportion         0.030
## ---------------------------------------------------------------------------
## incident_type
##        n  missing distinct    value
##    23568        0        1   Lethal
##
## Value      Lethal
## Frequency   23568
## Proportion      1
## ---------------------------------------------------------------------------
```

```
percentage_fatality_Brooklyn=(2921/23568)*100
print(percentage_fatality_Brooklyn)
```

```
## [1] 12.39392
```

```
percentage_fatality_Brooklyn=(9722/23568)*100
print(percentage_fatality_Brooklyn)
```

```
## [1] 41.25085
```

```
percentage_fatality_QUEENS=(3527/23568)*100
print(percentage_fatality_QUEENS)
```

```
## [1] 14.96521
```

```
percentage_fatality_BRONX=(6700/23568)*100
print(percentage_fatality_BRONX)
```

```
## [1] 28.42838
```

```
percentage_fatality_STATEN_ISLAND = (698/23568)*100
print(percentage_fatality_STATEN_ISLAND)
```

```
## [1] 2.961643
```

BOROUGH	FREQUENCY	PERCENTAGE
MANHATTAN	2921	12.4%
BROOKYLN	9722	41.2%
QUEENS	3527	14.96%
BRONX	6700	28.43%
STATEN ISLAND	698	2.96%

##Bias Identification

The data included in the "HISTORICAL NYPD SHOOTING INCIDENTS" categorizes many factors including age, range, location etc. In real world analysis, it may create some biases when studying the data. Mainly, certain biases may arise when we ask how was the data extracted, what other information may have been left out. Were there environmental factors, how many fatalities were there? Was the victim unarmed? Does each Boro or Jursidiction have a history of shooting activity? These questions make incident based data is hard to analyze as we don't have information about each situation. There could be misreporting which leads to biased incident data collection. For example, many of the columns contain NA or "UNKNOWN" values so we dont have the complete data to work with.

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19592)
##
## 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] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.5.1    stringr_1.4.0    purrr_0.3.4      readr_1.4.0
##  [5] tidyr_1.1.3      tibble_3.1.1     tidyverse_1.3.1  lubridate_1.7.10
##  [9] dbplyr_2.1.1     magrittr_2.0.1   dplyr_1.0.6      Hmisc_4.5-0
## [13] ggplot2_3.3.3    Formula_1.2-4    survival_3.2-10  lattice_0.20-41
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.6          png_0.1-7           assertthat_0.2.1
##  [4] digest_0.6.27       utf8_1.2.1          cellranger_1.1.0
##  [7] R6_2.5.0            backports_1.2.1     reprex_2.0.0
## [10] evaluate_0.14       httr_1.4.2          highr_0.9
## [13] pillar_1.6.1        rlang_0.4.11        readxl_1.3.1
## [16] rstudioapi_0.13     data.table_1.14.0   rpart_4.1-15
## [19] Matrix_1.3-2        checkmate_2.0.0     rmarkdown_2.8
## [22] labeling_0.4.2      splines_4.0.5       foreign_0.8-81
## [25] htmlwidgets_1.5.3   munsell_0.5.0       broom_0.7.6
## [28] modelr_0.1.8        compiler_4.0.5      xfun_0.22
## [31] pkgconfig_2.0.3     base64enc_0.1-3     htmltools_0.5.1.1
## [34] nnet_7.3-15         tidyselect_1.1.1    gridExtra_2.3
## [37] htmlTable_2.2.1     fansi_0.4.2         crayon_1.4.1
## [40] withr_2.4.2         grid_4.0.5          jsonlite_1.7.2
## [43] gtable_0.3.0        lifecycle_1.0.0     DBI_1.1.1
## [46] scales_1.1.1        cli_2.5.0           stringi_1.5.3
## [49] farver_2.1.0        fs_1.5.0            latticeExtra_0.6-29
## [52] xml2_1.3.2          ellipsis_0.3.2      generics_0.1.0
## [55] vctrs_0.3.8         RColorBrewer_1.1-2  tools_4.0.5
## [58] glue_1.4.2          hms_1.1.0           jpeg_0.1-8.1
```

```
## [61] yaml_2.2.1          colorspace_2.0-1    cluster_2.1.1
## [64] rvest_1.0.0         knitr_1.33          haven_2.4.1
```