Week3 NYPD Shooting Assignment

DS Student

5/14/2022

NYPD Shooting: Data wrangling, analysis, visualization and modeling

The NYPD shooting data covers shootings in the five boroughs of NY, that may or may not have resulted in death of the victim, from 2006 ro 2020. Geographical location; age, gender, and race info on the perpetrator and victim; time and date of occurrence are available in the data set.

In this assignment I will clean the data set and explore the data. As I am new to R I will also experiment with different visualizations.

Import data

Import data and print the first 6 entries

```
#df <- read.csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")
df <- read.csv("NYPD_Shooting_Incident_Data__Historic_.csv")
head(df)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO PRECINCT JURISDICTION_CODE
## 1     24050482 08/27/2006   05:35:00    BRONX       52                 0
## 2     77673979 03/11/2011   12:03:00   QUEENS      106                 0
## 3    203350417 10/06/2019   01:09:00 BROOKLYN       77                 0
## 4     80584527 09/04/2011   03:35:00    BRONX       40                 0
## 5     90843766 05/27/2013   21:16:00   QUEENS      100                 0
## 6     92393427 09/01/2013   04:17:00 BROOKLYN       67                 0
##   LOCATION_DESC STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX PERP_RACE
## 1                                 true
## 2                                false
## 3                                false
## 4                                false
## 5                                false
## 6                                false
##   VIC_AGE_GROUP VIC_SEX       VIC_RACE X_COORD_CD Y_COORD_CD Latitude Longitude
## 1         25-44       F BLACK HISPANIC    1017542   255918.9 40.86906 -73.87963
## 2           65+       M          WHITE    1027543   186095.0 40.67737 -73.84392
## 3         18-24       F          BLACK     995325   185155.0 40.67489 -73.96008
## 4           <18       M          BLACK    1007453   233952.0 40.80880 -73.91618
## 5         18-24       M          BLACK    1041267   157133.5 40.59780 -73.79469
## 6           <18       M          BLACK    1001694   170112.9 40.63359 -73.93715
```

```
##                                       Lon_Lat
## 1  POINT (-73.87963173099996 40.86905819000003)
## 2 POINT (-73.84392019199998 40.677366895000034)
## 3 POINT (-73.96007501899999 40.674885741000026)
## 4  POINT (-73.91618413199996 40.80879780500004)
## 5 POINT (-73.79468553799995 40.597796249000055)
## 6  POINT (-73.93715330699996 40.63358818100005)
```

Column names

Print the column names

```
colnames(df)
```

```
##  [1] "INCIDENT_KEY"            "OCCUR_DATE"
##  [3] "OCCUR_TIME"              "BORO"
##  [5] "PRECINCT"                "JURISDICTION_CODE"
##  [7] "LOCATION_DESC"           "STATISTICAL_MURDER_FLAG"
##  [9] "PERP_AGE_GROUP"          "PERP_SEX"
## [11] "PERP_RACE"               "VIC_AGE_GROUP"
## [13] "VIC_SEX"                 "VIC_RACE"
## [15] "X_COORD_CD"              "Y_COORD_CD"
## [17] "Latitude"                "Longitude"
## [19] "Lon_Lat"
```

Select columns

`select` columns to delete from data frame (mainly due to location of shooting and perpetrator or victim's sex). Print the first six rows of new data frame.

```
df <- select(df, - c(JURISDICTION_CODE,LOCATION_DESC,PERP_RACE,VIC_RACE, Latitude:Lon_Lat))
head(df)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO PRECINCT STATISTICAL_MURDER_FLAG
## 1     24050482 08/27/2006   05:35:00    BRONX       52                   true
## 2     77673979 03/11/2011   12:03:00   QUEENS      106                  false
## 3    203350417 10/06/2019   01:09:00 BROOKLYN       77                  false
## 4     80584527 09/04/2011   03:35:00    BRONX       40                  false
## 5     90843766 05/27/2013   21:16:00   QUEENS      100                  false
## 6     92393427 09/01/2013   04:17:00 BROOKLYN       67                  false
##   PERP_AGE_GROUP PERP_SEX VIC_AGE_GROUP VIC_SEX X_COORD_CD Y_COORD_CD
## 1                  25-44       F    1017542   255918.9
## 2                    65+       M    1027543   186095.0
## 3                  18-24       F     995325   185155.0
## 4                    <18       M    1007453   233952.0
## 5                  18-24       M    1041267   157133.5
## 6                    <18       M    1001694   170112.9
```

Summarize data set

Summarize the data in each column; this also prints out the data type for each column.

```
summary(df)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:23585       Length:23585       Length:23585
##  1st Qu.: 55322804   Class :character   Class :character   Class :character
##  Median : 83435362   Mode  :character   Mode  :character   Mode  :character
##  Mean   :102280741
##  3rd Qu.:150911774
##  Max.   :230611229
##     PRECINCT      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP       PERP_SEX
##  Min.   :  1.00   Length:23585            Length:23585       Length:23585
##  1st Qu.: 44.00   Class :character        Class :character   Class :character
##  Median : 69.00   Mode  :character        Mode  :character   Mode  :character
##  Mean   : 66.21
##  3rd Qu.: 81.00
##  Max.   :123.00
##  VIC_AGE_GROUP        VIC_SEX            X_COORD_CD        Y_COORD_CD
##  Length:23585       Length:23585       Min.   : 914928   Min.   :125757
##  Class :character   Class :character   1st Qu.: 999925   1st Qu.:182539
##  Mode  :character   Mode  :character   Median :1007654   Median :193470
##                                        Mean   :1009379   Mean   :207300
##                                        3rd Qu.:1016782   3rd Qu.:239163
##                                        Max.   :1066815   Max.   :271128
```

Change data types

Change the data types to factors, characters, or logical types. Change the column name for STATISTICAL_MURDER_FLAG to MURDER. Delete the duplicate STATISTICAL_MURDER_FLAG column. Print the first six entries.

```
df <- df %>%
  mutate(MURDER = as.logical(STATISTICAL_MURDER_FLAG),
         VIC_SEX = as.factor(VIC_SEX),
         VIC_AGE_GROUP = as.factor(VIC_AGE_GROUP),
         INCIDENT_KEY = as.character(INCIDENT_KEY),
         PERP_AGE_GROUP = as.factor(PERP_AGE_GROUP),
         PERP_SEX = as.factor(PERP_SEX),
         BORO = as.factor(BORO),
         PRECINCT = as.factor(PRECINCT)) %>%
  select(-c(STATISTICAL_MURDER_FLAG))


head(df)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO PRECINCT PERP_AGE_GROUP PERP_SEX
## 1     24050482 08/27/2006   05:35:00    BRONX       52
## 2     77673979 03/11/2011   12:03:00   QUEENS      106
## 3    203350417 10/06/2019   01:09:00 BROOKLYN       77
## 4     80584527 09/04/2011   03:35:00    BRONX       40
## 5     90843766 05/27/2013   21:16:00   QUEENS      100
## 6     92393427 09/01/2013   04:17:00 BROOKLYN       67
##   VIC_AGE_GROUP VIC_SEX X_COORD_CD Y_COORD_CD MURDER
```

```
## 1         25-44       F    1017542   255918.9   TRUE
## 2           65+       M    1027543   186095.0  FALSE
## 3         18-24       F     995325   185155.0  FALSE
## 4           <18       M    1007453   233952.0  FALSE
## 5         18-24       M    1041267   157133.5  FALSE
## 6           <18       M    1001694   170112.9  FALSE
```

Change the `OCCUR_DATE` variable to a date type.

```
df <- df %>%
  mutate(OCCUR_DATE = as.Date(OCCUR_DATE,
                              format= "%m/%d/%Y"))

head(df)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO PRECINCT PERP_AGE_GROUP PERP_SEX
## 1     24050482 2006-08-27   05:35:00    BRONX       52
## 2     77673979 2011-03-11   12:03:00   QUEENS      106
## 3    203350417 2019-10-06   01:09:00 BROOKLYN       77
## 4     80584527 2011-09-04   03:35:00    BRONX       40
## 5     90843766 2013-05-27   21:16:00   QUEENS      100
## 6     92393427 2013-09-01   04:17:00 BROOKLYN       67
##   VIC_AGE_GROUP VIC_SEX X_COORD_CD Y_COORD_CD MURDER
## 1         25-44       F    1017542   255918.9   TRUE
## 2           65+       M    1027543   186095.0  FALSE
## 3         18-24       F     995325   185155.0  FALSE
## 4           <18       M    1007453   233952.0  FALSE
## 5         18-24       M    1041267   157133.5  FALSE
## 6           <18       M    1001694   170112.9  FALSE
```

Separate into `year`, `month`, and `day` columns using the `year()`, `month()`, `date()` functions from the lubridate package. Print the first six entries.

```
df$year <- year(df$OCCUR_DATE)
df$month <- month(df$OCCUR_DATE, label = TRUE)
df$day <- day(df$OCCUR_DATE)

df <- df %>%
  select(INCIDENT_KEY, OCCUR_DATE, year, month, day, everything())

head(df)
```

```
##   INCIDENT_KEY OCCUR_DATE year month day OCCUR_TIME     BORO PRECINCT
## 1     24050482 2006-08-27 2006   Aug  27   05:35:00    BRONX       52
## 2     77673979 2011-03-11 2011   Mar  11   12:03:00   QUEENS      106
## 3    203350417 2019-10-06 2019   Oct   6   01:09:00 BROOKLYN       77
## 4     80584527 2011-09-04 2011   Sep   4   03:35:00    BRONX       40
## 5     90843766 2013-05-27 2013   May  27   21:16:00   QUEENS      100
## 6     92393427 2013-09-01 2013   Sep   1   04:17:00 BROOKLYN       67
##   PERP_AGE_GROUP PERP_SEX VIC_AGE_GROUP VIC_SEX X_COORD_CD Y_COORD_CD MURDER
## 1                          25-44       F    1017542   255918.9   TRUE
## 2                            65+       M    1027543   186095.0  FALSE
## 3                          18-24       F     995325   185155.0  FALSE
```

```
## 4                                    <18       M    1007453   233952.0  FALSE
## 5                                  18-24       M    1041267   157133.5  FALSE
## 6                                    <18       M    1001694   170112.9  FALSE
```

Add an additional column of `hour` that contains the hour of the day the shooting occurred.

```
# make a second column containing the OCCUR_TIME values
df$OCCUR_HOUR <- df$OCCUR_TIME

df <- df %>%
  separate(OCCUR_HOUR,c('hour', 'minute', 'second'), ':') %>%
  mutate(hour = as.double(hour)) %>%
  select(-c(minute, second)) %>%
  select(INCIDENT_KEY, OCCUR_DATE, year, month, day,hour, everything())

head(df)
```

```
##   INCIDENT_KEY OCCUR_DATE year month day hour OCCUR_TIME     BORO PRECINCT
## 1     24050482 2006-08-27 2006   Aug  27    5   05:35:00    BRONX       52
## 2     77673979 2011-03-11 2011   Mar  11   12   12:03:00   QUEENS      106
## 3    203350417 2019-10-06 2019   Oct   6    1   01:09:00 BROOKLYN       77
## 4     80584527 2011-09-04 2011   Sep   4    3   03:35:00    BRONX       40
## 5     90843766 2013-05-27 2013   May  27   21   21:16:00   QUEENS      100
## 6     92393427 2013-09-01 2013   Sep   1    4   04:17:00 BROOKLYN       67
##   PERP_AGE_GROUP PERP_SEX VIC_AGE_GROUP VIC_SEX X_COORD_CD Y_COORD_CD MURDER
## 1                              25-44       F    1017542   255918.9   TRUE
## 2                                65+       M    1027543   186095.0  FALSE
## 3                              18-24       F     995325   185155.0  FALSE
## 4                                <18       M    1007453   233952.0  FALSE
## 5                              18-24       M    1041267   157133.5  FALSE
## 6                                <18       M    1001694   170112.9  FALSE
```

Change the `OCCUR_TIME` to the hour, minute, second format (class: Period) using using the `hms()` function from the lubridate package .

```
df$OCCUR_TIME <- hms(df$OCCUR_TIME)

head(df)
```

```
##   INCIDENT_KEY OCCUR_DATE year month day hour OCCUR_TIME     BORO PRECINCT
## 1     24050482 2006-08-27 2006   Aug  27    5  5H 35M 0S    BRONX       52
## 2     77673979 2011-03-11 2011   Mar  11   12  12H 3M 0S   QUEENS      106
## 3    203350417 2019-10-06 2019   Oct   6    1   1H 9M 0S BROOKLYN       77
## 4     80584527 2011-09-04 2011   Sep   4    3  3H 35M 0S    BRONX       40
## 5     90843766 2013-05-27 2013   May  27   21 21H 16M 0S   QUEENS      100
## 6     92393427 2013-09-01 2013   Sep   1    4  4H 17M 0S BROOKLYN       67
##   PERP_AGE_GROUP PERP_SEX VIC_AGE_GROUP VIC_SEX X_COORD_CD Y_COORD_CD MURDER
## 1                              25-44       F    1017542   255918.9   TRUE
## 2                                65+       M    1027543   186095.0  FALSE
## 3                              18-24       F     995325   185155.0  FALSE
## 4                                <18       M    1007453   233952.0  FALSE
## 5                              18-24       M    1041267   157133.5  FALSE
## 6                                <18       M    1001694   170112.9  FALSE
```

Summarize the new data set

The summary demonstrates that the data is cleaner. However, some entries are missing (for example in the PERP_AGE_GROUP and PERP_SEX). In the following sections when we analyze and visualize variables with missing entries, we will filter them out for the analysis.

```
summary(df)
```

```
##  INCIDENT_KEY         OCCUR_DATE              year          month
##  Length:23585       Min.   :2006-01-01   Min.   :2006   Jul    :2805
##  Class :character   1st Qu.:2008-12-31   1st Qu.:2008   Aug    :2774
##  Mode  :character   Median :2012-02-27   Median :2012   Jun    :2458
##                     Mean   :2012-10-05   Mean   :2012   Sep    :2224
##                     3rd Qu.:2016-03-02   3rd Qu.:2016   May    :2174
##                     Max.   :2020-12-31   Max.   :2020   Oct    :2017
##                                                         (Other):9133
##       day              hour         OCCUR_TIME
##  Min.   : 1.00   Min.   : 0.00   Min.   :0S
##  1st Qu.: 8.00   1st Qu.: 3.00   1st Qu.:3H 20M 0S
##  Median :16.00   Median :15.00   Median :15H 0M 0S
##  Mean   :15.99   Mean   :12.08   Mean   :12H 33M 7.48187407250225S
##  3rd Qu.:24.00   3rd Qu.:20.00   3rd Qu.:20H 45M 0S
##  Max.   :31.00   Max.   :23.00   Max.   :23H 59M 0S
##
##             BORO          PRECINCT     PERP_AGE_GROUP PERP_SEX  VIC_AGE_GROUP
##  BRONX        :6701   75     : 1375          :8295    : 8261   <18    : 2525
##  BROOKLYN     :9734   73     : 1284   18-24  :5508   F:  335   18-24  : 9003
##  MANHATTAN    :2922   67     : 1101   25-44  :4714   M:13490   25-44  :10303
##  QUEENS       :3532   79     :  921   UNKNOWN:3148   U: 1499   45-64  : 1541
##  STATEN ISLAND: 696   44     :  841   <18    :1368             65+    :  154
##                       47     :  818   45-64  : 495             UNKNOWN:   59
##                       (Other):17245   (Other):  57
##  VIC_SEX     X_COORD_CD        Y_COORD_CD       MURDER
##  F: 2204   Min.   : 914928   Min.   :125757   Mode :logical
##  M:21370   1st Qu.: 999925   1st Qu.:182539   FALSE:19085
##  U:   11   Median :1007654   Median :193470   TRUE :4500
##            Mean   :1009379   Mean   :207300
##            3rd Qu.:1016782   3rd Qu.:239163
##            Max.   :1066815   Max.   :271128
##
```

Exploring the dataframe

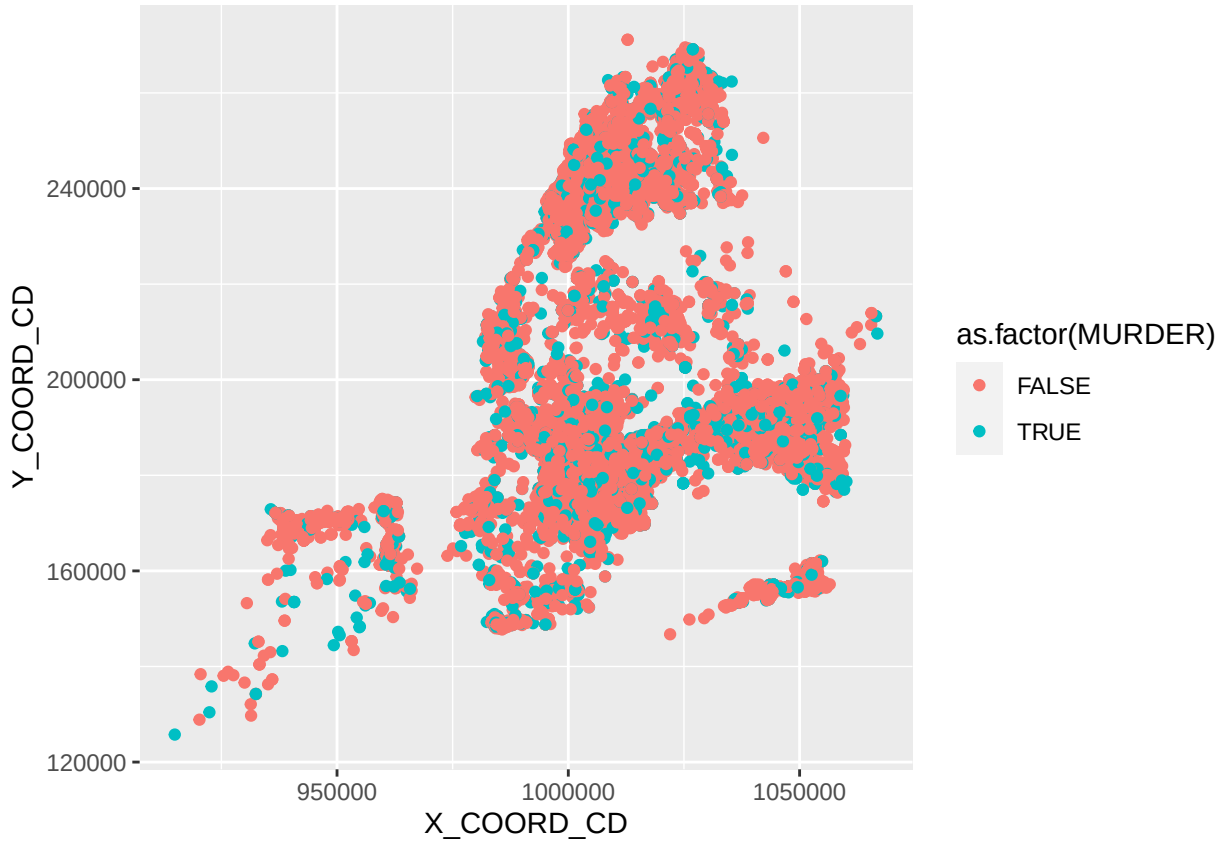
1) how do the x and y coordinates and boroughs correlate with one another?
2) what age groups are most commonly represented in the perpetrator and victim cohorts?
3) what is the gender distribution of the victims?
4) what has been the trend of number of shootings per year?
5) are shootings more common in specific months or specific hours of the day?
6) which boroughs have the highest number of shootings? What is the relationship between number of shootings and murders?

I have intentionally chosen not to evaluate the race of the victim and perpetrator on this analysis - this may be a source of bias for this analysis.

Scatterplot of x and y coordinates

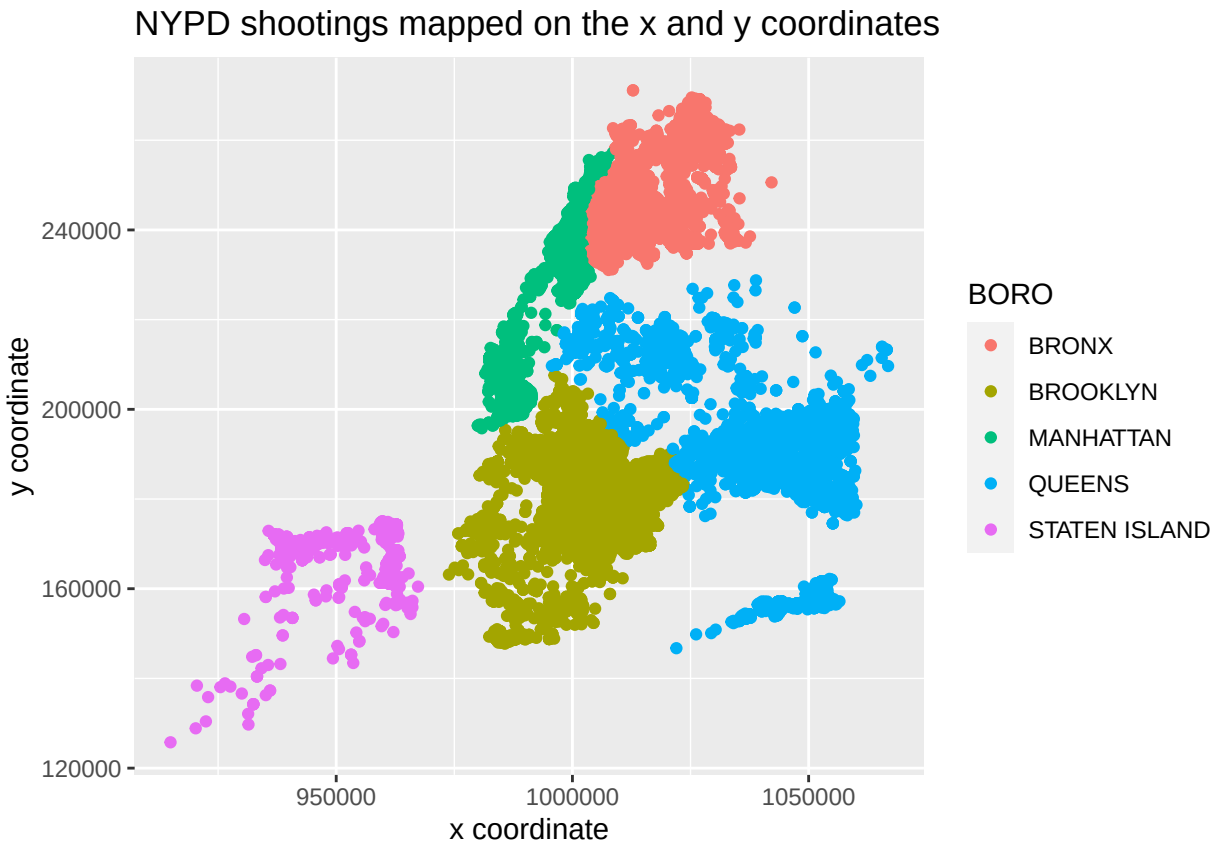
First visualization in the assignment - part 1; demonstrates the geographical coordinates and whether the shooting resulted in a murder or not.

```
coord_xy_1 <- ggplot(df, aes(x = X_COORD_CD, y = Y_COORD_CD, color = as.factor(MURDER))) +
  geom_point()

coord_xy_1
```



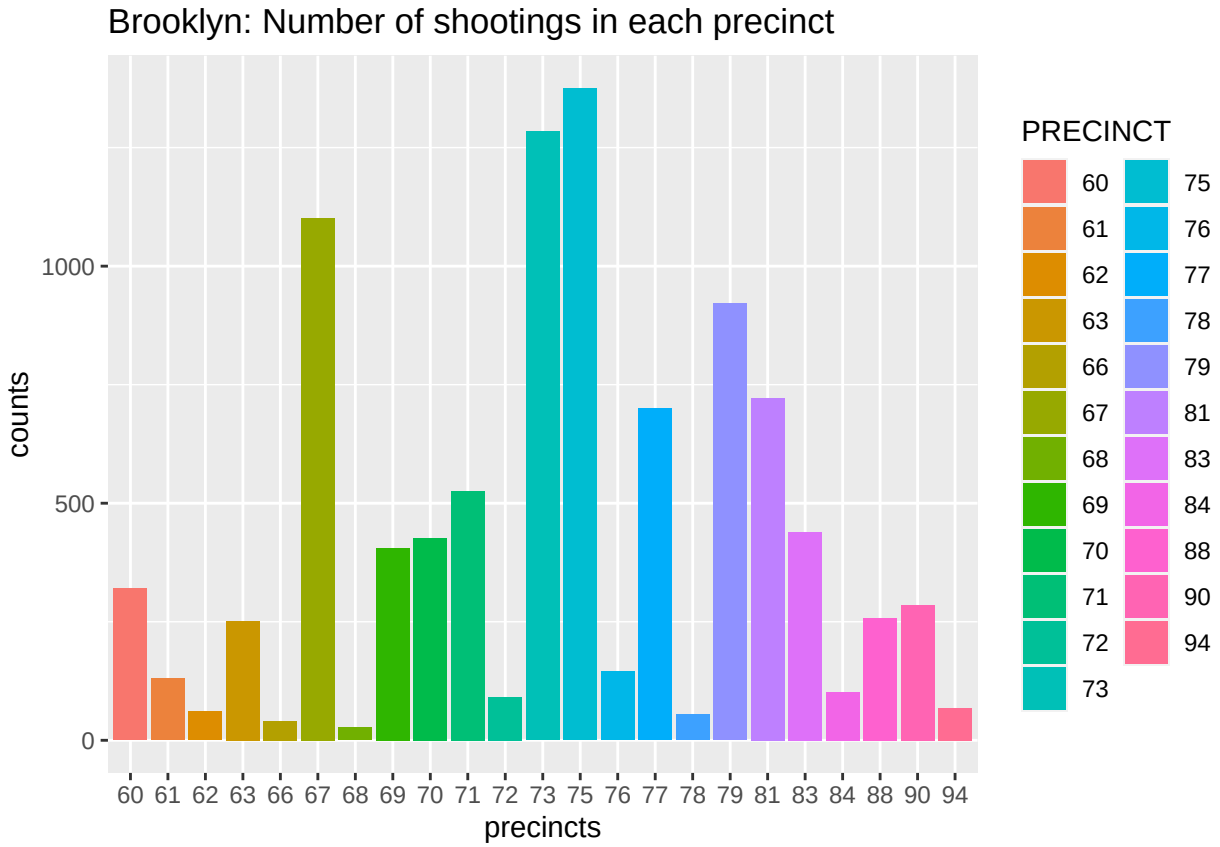
First visualization in the assignment - part 2; demonstrates the geographical coordinates and the color-coded boroughs.

```
coord_xy_2 <- ggplot(df, aes(x = X_COORD_CD, y = Y_COORD_CD, color = BORO)) +
  geom_point() +
  labs(title = "NYPD shootings mapped on the x and y coordinates",
       x = "x coordinate",
       y = "y coordinate")



coord_xy_2
```



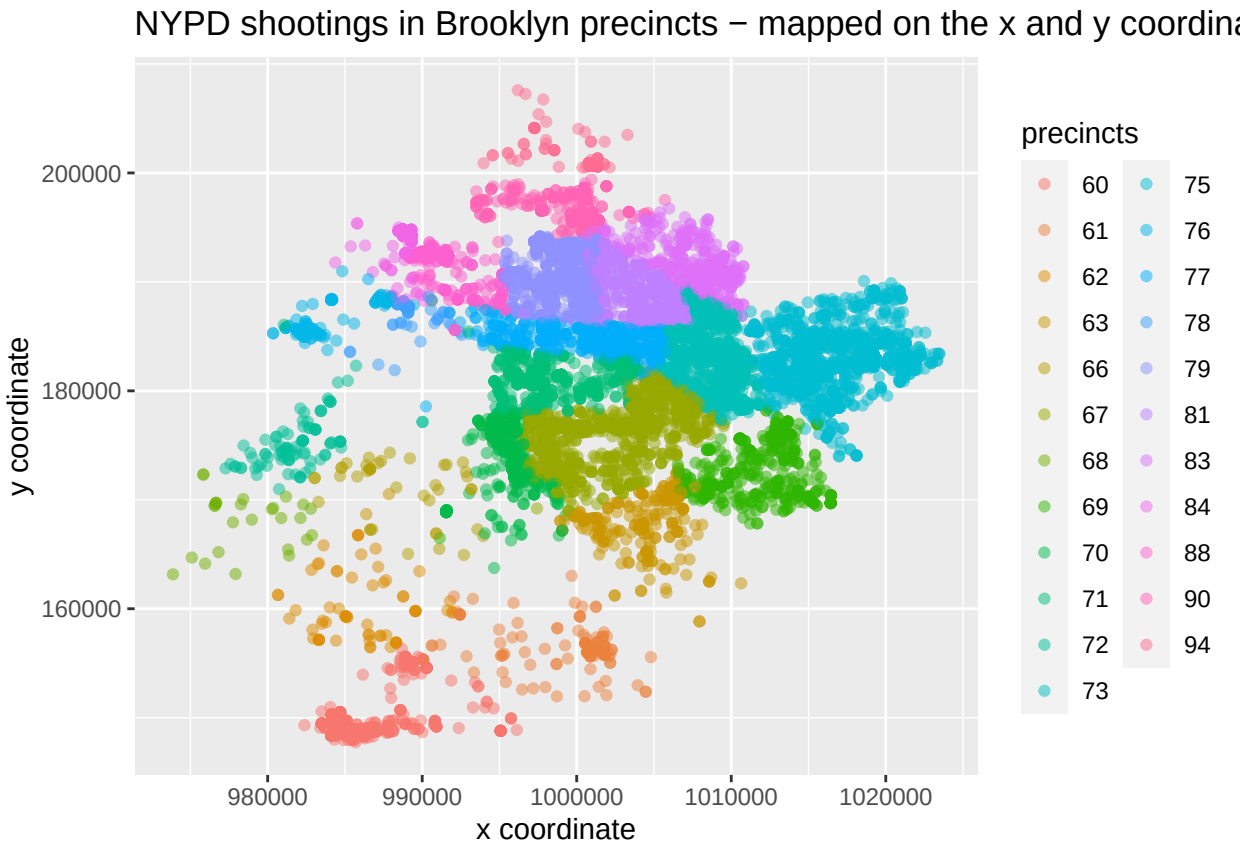
By precincts

```
brooklyn <- df %>%
  filter(BORO == "BROOKLYN") %>%
  group_by(PRECINCT) %>%
  ggplot(aes(x= PRECINCT, fill = PRECINCT)) +
  geom_bar() +
  labs(title = "Brooklyn: Number of shootings in each precinct",
       x = "precincts",
       y = "counts")

brooklyn
```



Map the x and y coordinates to the precincts in Brooklyn.

```
coord_brooklyn <- df %>%
  filter(BORO == "BROOKLYN") %>%
  ggplot(aes(x = X_COORD_CD, y = Y_COORD_CD, color = factor(PRECINCT))) +
  geom_point(alpha = 0.5) +
  labs(title = "NYPD shootings in Brooklyn precincts - mapped on the x and y coordinates",
       x = "x coordinate",
       y = "y coordinate",
       color = "precincts")


coord_brooklyn
```



Vertical bar graph of perpetrators' age groups

Print out the different categories in the perpetrators age group.

```
unique(df$PERP_AGE_GROUP)
```

```
##  [1]         18-24   UNKNOWN 25-44   <18     45-64   65+     1020    940
## [10] 224
## Levels:  <18 1020 18-24 224 25-44 45-64 65+ 940 UNKNOWN
```

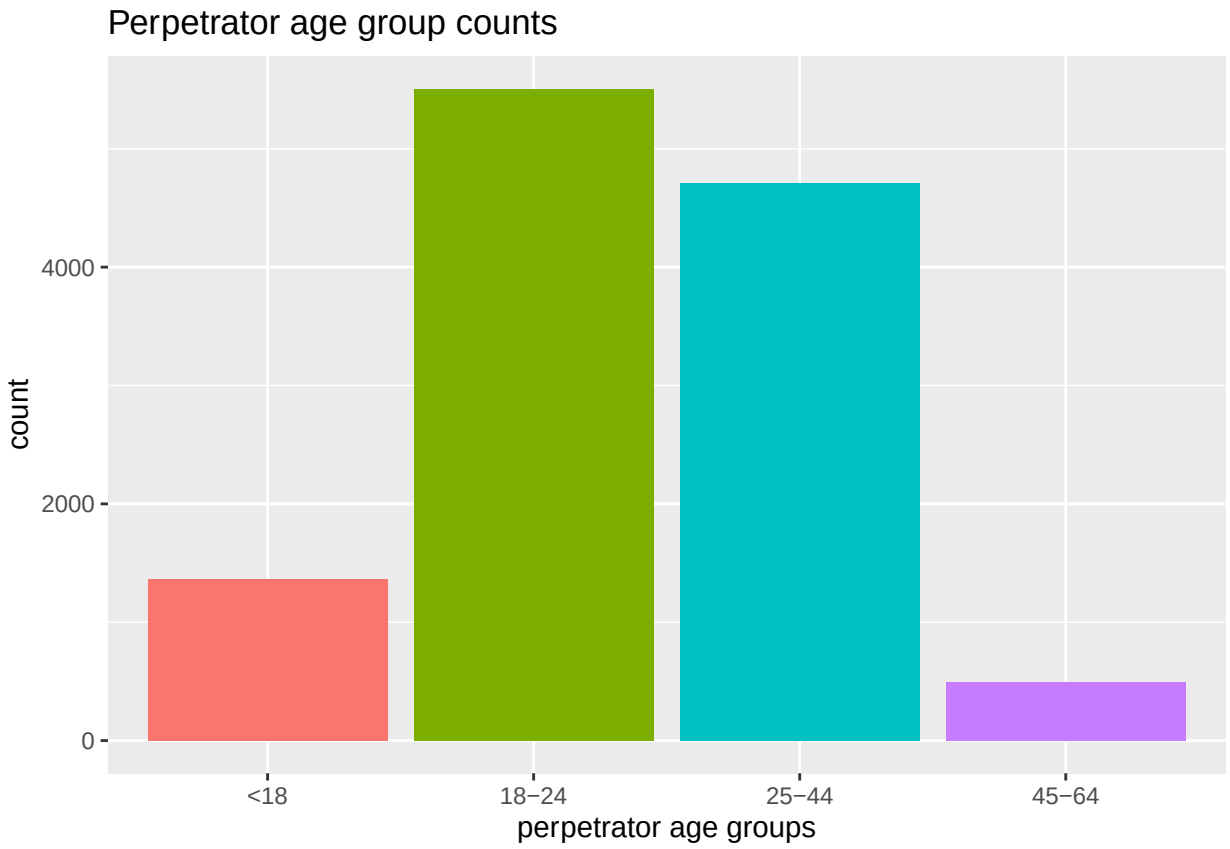
The data in this column is not tidy. If the levels are 1020, 224, or 940 filter them out. Print table of counts of known perpetrator age groups.

```
df_perp_age_known <- df %>%
  filter(PERP_AGE_GROUP %in% c('<18','18-24', '25-44', '45-64'))

table(df_perp_age_known$PERP_AGE_GROUP)
```

```
##
##             <18    1020   18-24     224   25-44   45-64     65+     940 UNKNOWN
##       0    1368       0    5508       0    4714     495       0       0       0
```

Second visualization in the assignment - part 1. Vertical bar graph of perpetrators with a known age group.

```
perp_age <- ggplot(df_perp_age_known, aes(x = PERP_AGE_GROUP, fill = factor(PERP_AGE_GROUP))) +
  geom_bar(show.legend = FALSE) +
  labs(title = "Perpetrator age group counts",
       x = "perpetrator age groups",
       y = "count")

perp_age
```



Vertical bar graph of victims' age groups

Print out the different categories in the victims age . The data in this column is tidy.

```
unique(df$VIC_AGE_GROUP)
```

```
## [1] 25-44   65+     18-24   <18     45-64   UNKNOWN
## Levels: <18 18-24 25-44 45-64 65+ UNKNOWN
```

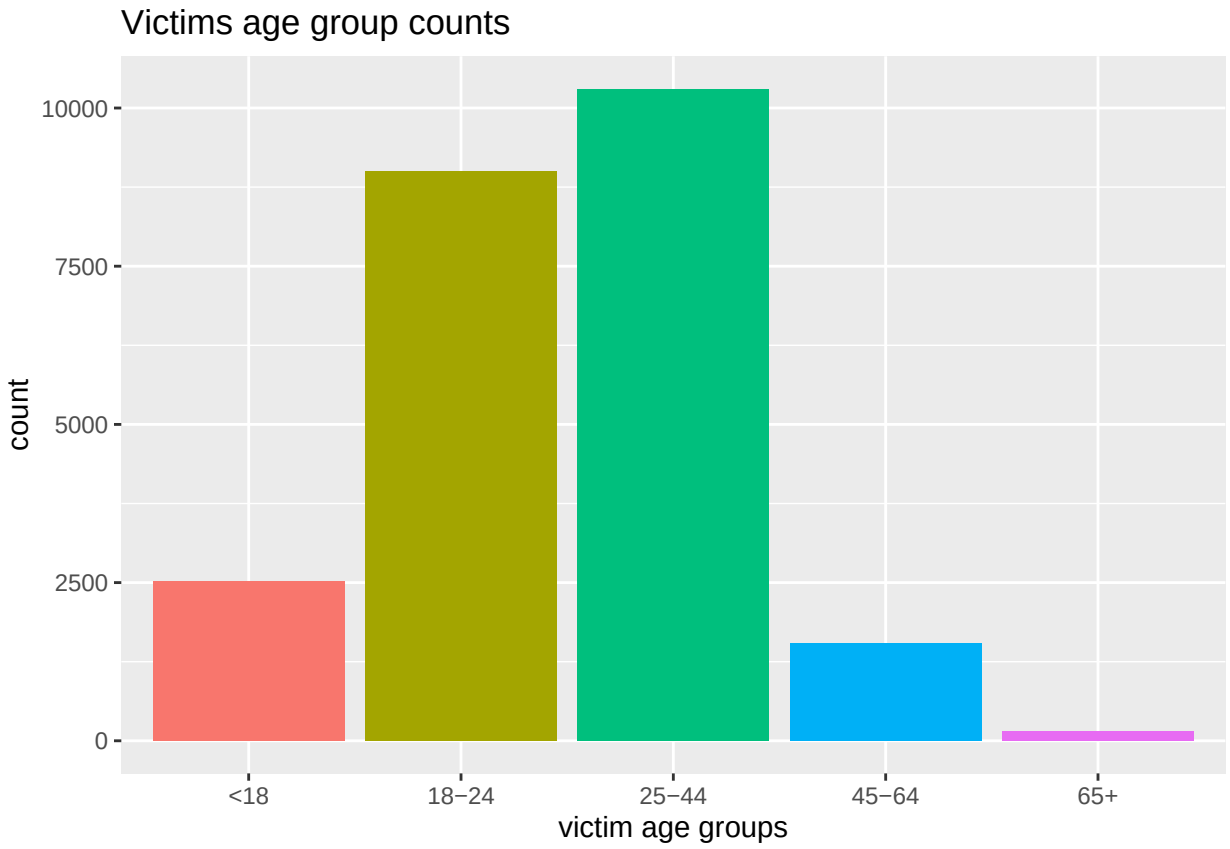
Print out the count in each category

```
table(df$VIC_AGE_GROUP)
```

```
##
##     <18   18-24   25-44   45-64     65+ UNKNOWN
##    2525    9003   10303    1541     154      59
```

Second visualization in the assignment - part 2. Plot victim age groups as a vertical bar graph

```
vic_age <- df %>%
  filter(VIC_AGE_GROUP %in% c('<18','18-24', '25-44', '45-64', '65+')) %>%
  ggplot(aes(x = VIC_AGE_GROUP, fill = factor(VIC_AGE_GROUP))) +
  geom_bar(show.legend = FALSE) +
  labs(title = "Victims age group counts",
       x = "victim age groups",
       y = "count")

vic_age
```



Horizontal bar graph of victims' gender

Table of counts of victims' gender

```
table(df$VIC_SEX)
```

```
##
##     F     M     U
##  2204 21370    11
```

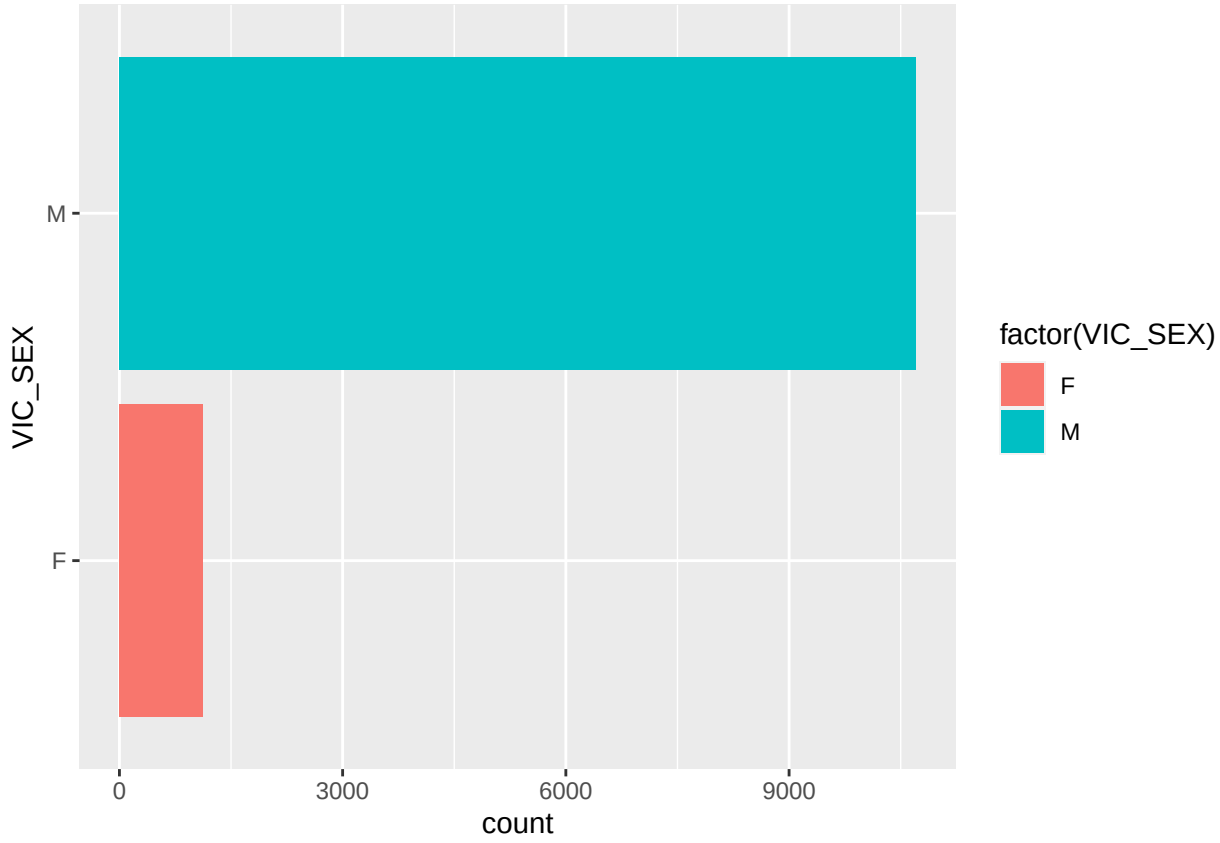
Second visualization in the assignment - part 3. Plot victim age groups as a vertical bar graph. `filter` out the unknown (U) values.

```
vic_sex <- df %>%
  filter(VIC_SEX == c("F", "M")) %>%
  ggplot(aes(x=VIC_SEX, fill = factor(VIC_SEX))) +
  geom_bar() +
  coord_flip()
```

```
## Warning in `==.default`(VIC_SEX, c("F", "M")): longer object length is not a
## multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
vic_sex
```



Number of shootings each year

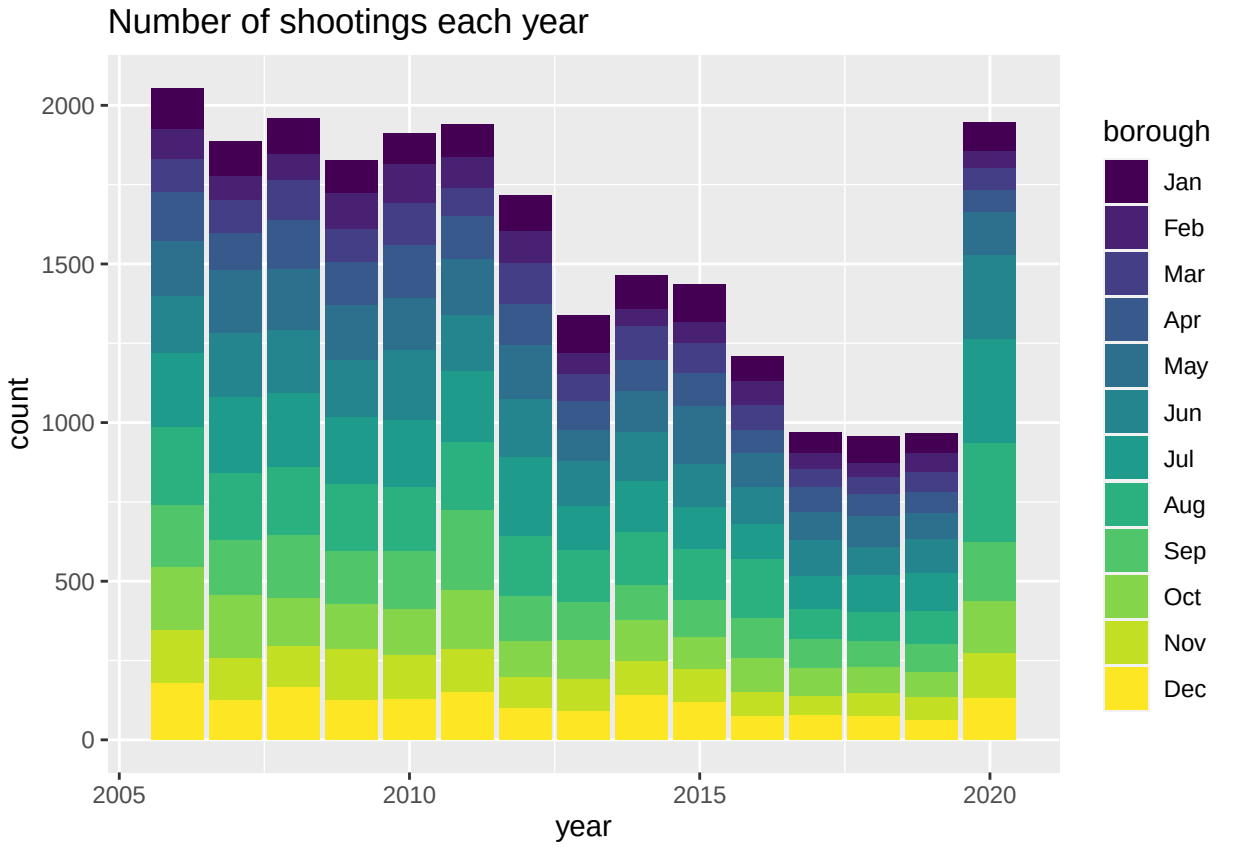
Table of counts of shootings each year.

```
table(df$year)
```

```
##
## 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020
## 2055 1887 1959 1828 1912 1939 1717 1339 1464 1434 1208  970  958  967 1948
```

Third visualization in the assignment - part 1. Plot the number of shootings in each year, further grouped by month

```
shootings_yearly <- ggplot(df, aes(x=year, fill=factor(month))) +
  geom_bar()+
  labs(title = "Number of shootings each year",
       fill = "borough")

shootings_yearly
```

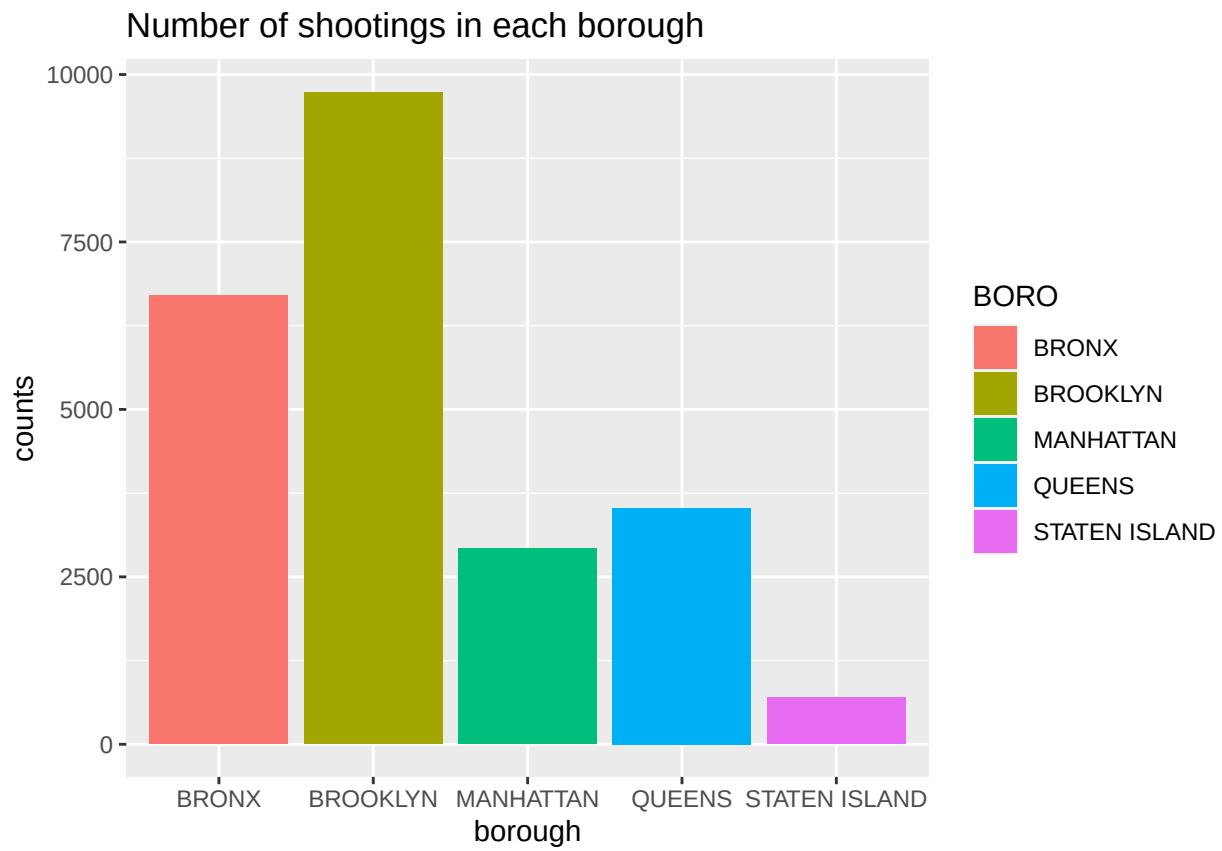


Observation: The bar graph demonstrates and overall decreasing trend in the number of shootings between 2012 and 2019 and an increase to 2011 numbers in 2020.

Number of shootings in each boroough

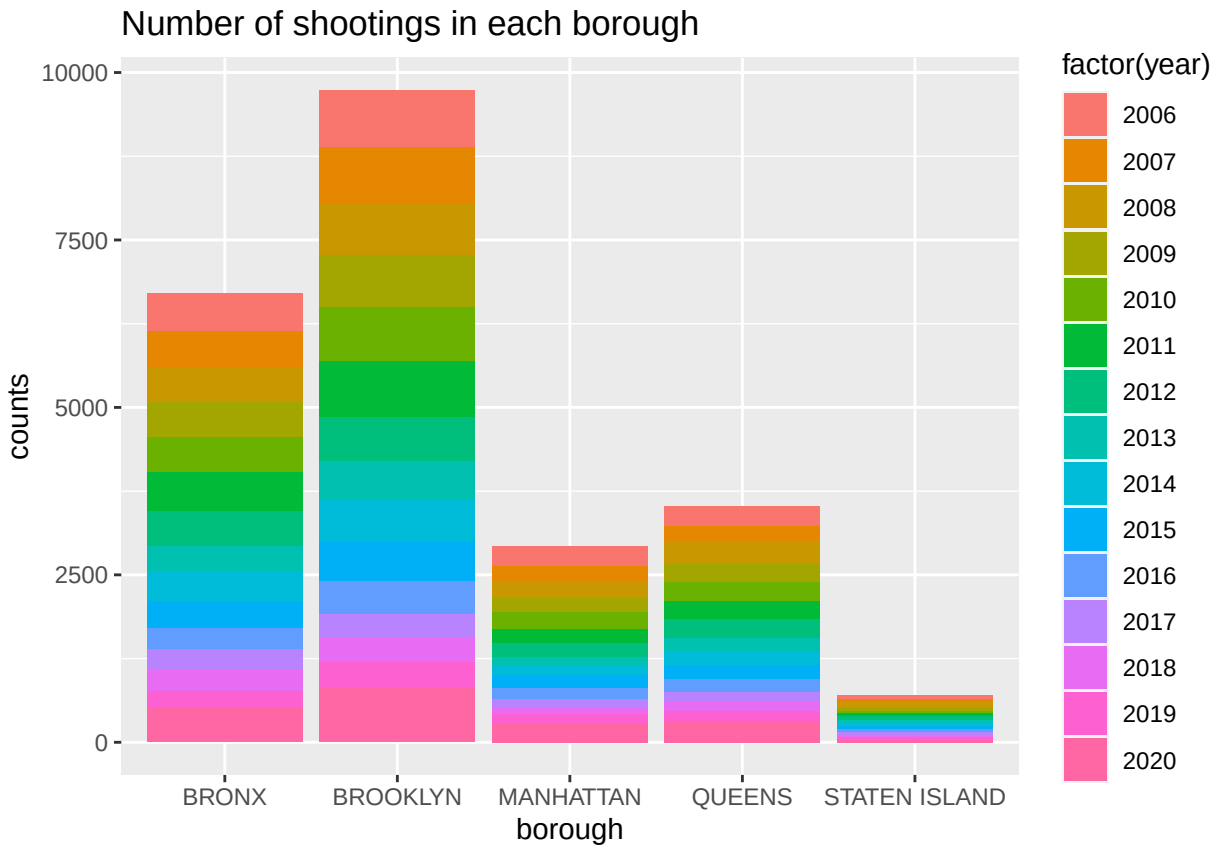
Third visualization in the assignment - part 2. Vertical bar graph of shootings in each borough grouped by year.

```
shootings_boro_yr <- ggplot(df, aes(x=BORO, fill = BORO)) +
  geom_bar() +
  labs(title = "Number of shootings in each borough",
       x = "borough",
       y = "counts")

shootings_boro_yr
```



Shootings in each borough, factored by year.

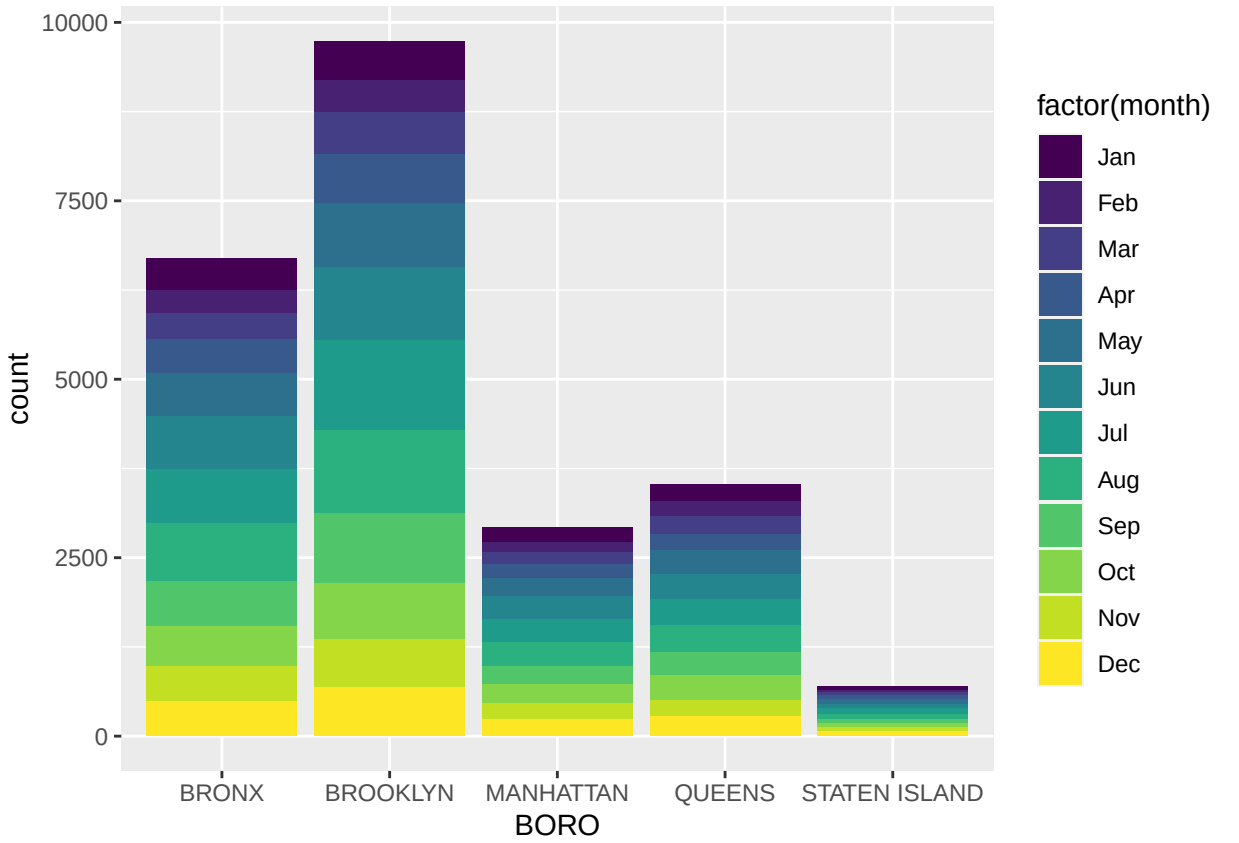
```
shootings_boro_yr <- ggplot(df, aes(x=BORO, fill=factor(year))) +
  geom_bar() +
  labs(title = "Number of shootings in each borough",
       x = "borough",
       y = "counts")

shootings_boro_yr
```



Observation: Graph demonstrates that Brooklyn and Bronx have the highest number of shootings. Height of the different colored bars in these two boroughs also corroborates with the decreased number of shootings between 2012 and 2019 compared to prior years and 2020.

Third visualization in the assignment - part 3. Vertical bar graph of shootings in each borough grouped by month.

```
shootings_boro_mo <- ggplot(df, aes(x=BORO, fill=factor(month))) +
  geom_bar()

shootings_boro_mo
```



Observation: Graph demonstrates that Brooklyn and Bronx have the highest number of shootings. Height of the different colored bars in these two boroughs also corroborates with the decreased number of shootings between 2012 and 2019 compared to prior years and 2020.

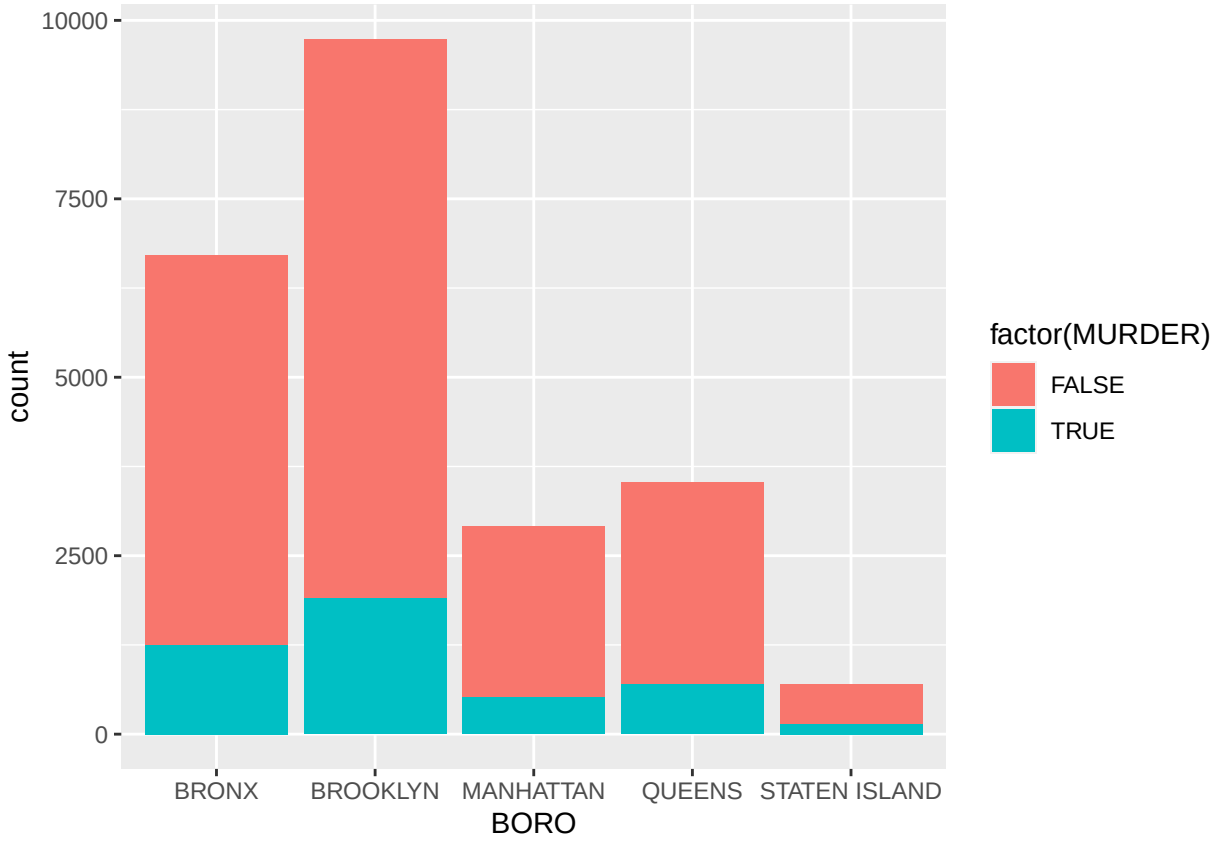
This observation is corroborated when we summarize the total shooting counts per month as a table.

```
table(df$month)
```

```
##
##  Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1505 1149 1402 1647 2174 2458 2805 2774 2224 2017 1692 1738
```

Third visualization in the assignment - part 4. Vertical bar graph of shootings in each borough grouped by whether or not it resulted in demise of the victim.

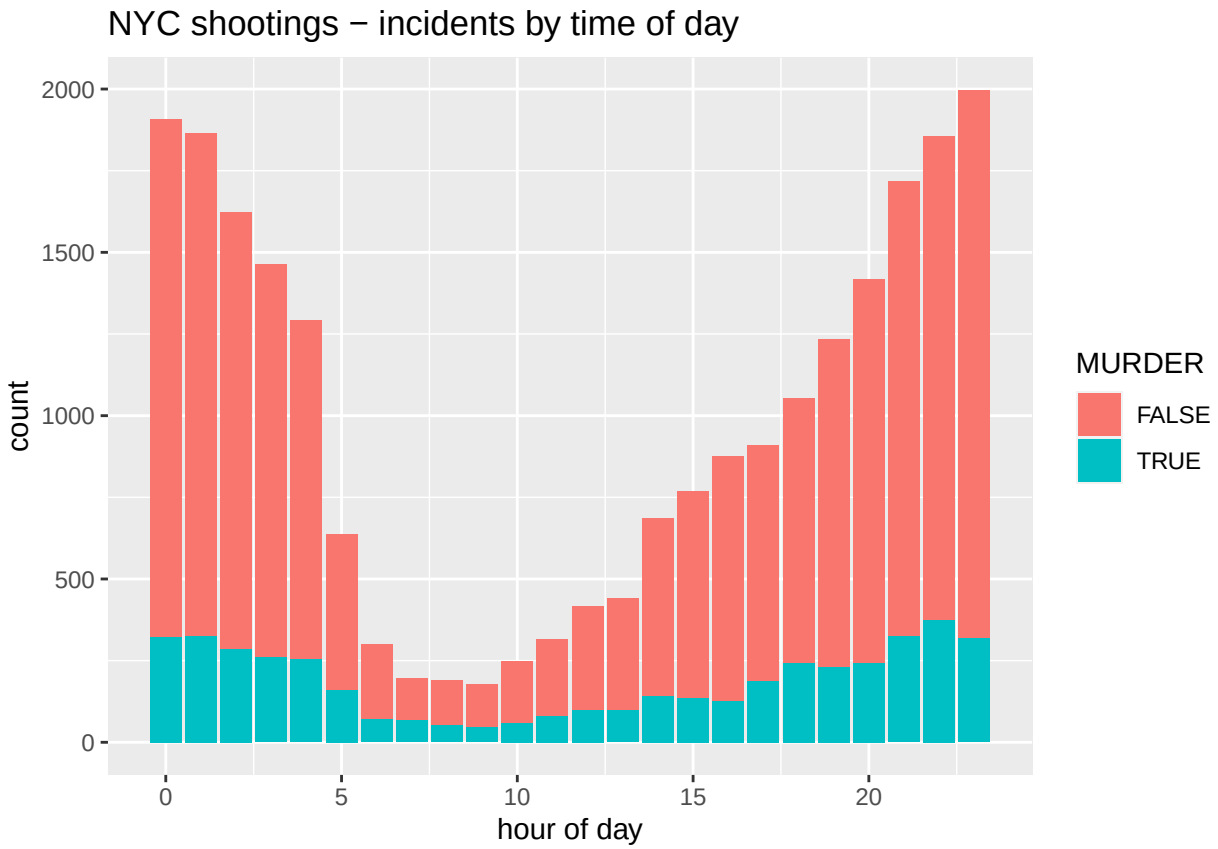
```
shootings_boro_mur <- ggplot(df, aes(x=BORO, fill=factor(MURDER))) +
  geom_bar()

shootings_boro_mur
```



Number of shootings at different times of day

Third visualization in the assignment - part 5. Number of shootings in each hour of the day subgrouped by MURDER.

```
shootings_hour<- ggplot(df, aes(hour, fill = MURDER)) +
  geom_bar() +
  labs(title = "NYC shootings - incidents by time of day",
  x = "hour of day", y = "count")

shootings_hour
```

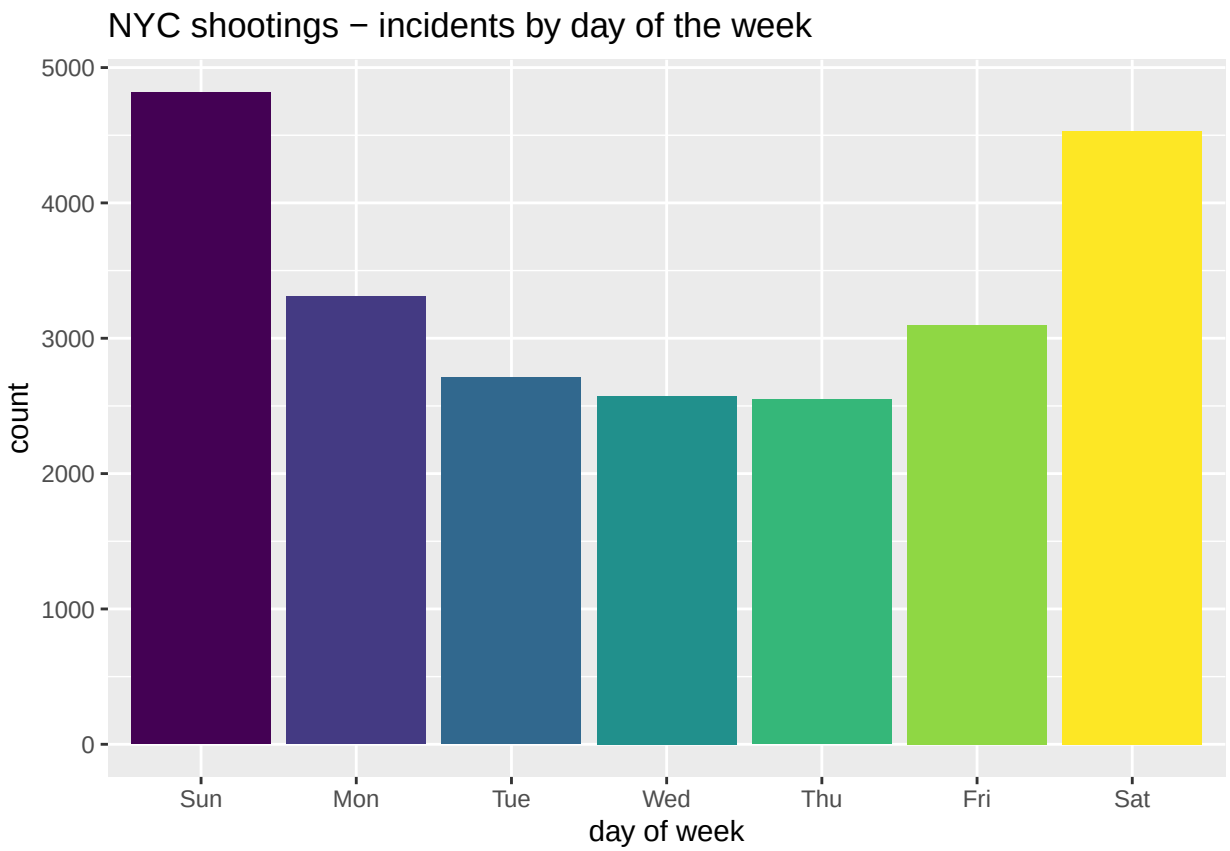


Observation: Graph demonstrates that most shootings occur between 7 pm and 4 am.

Time series - from another student's work

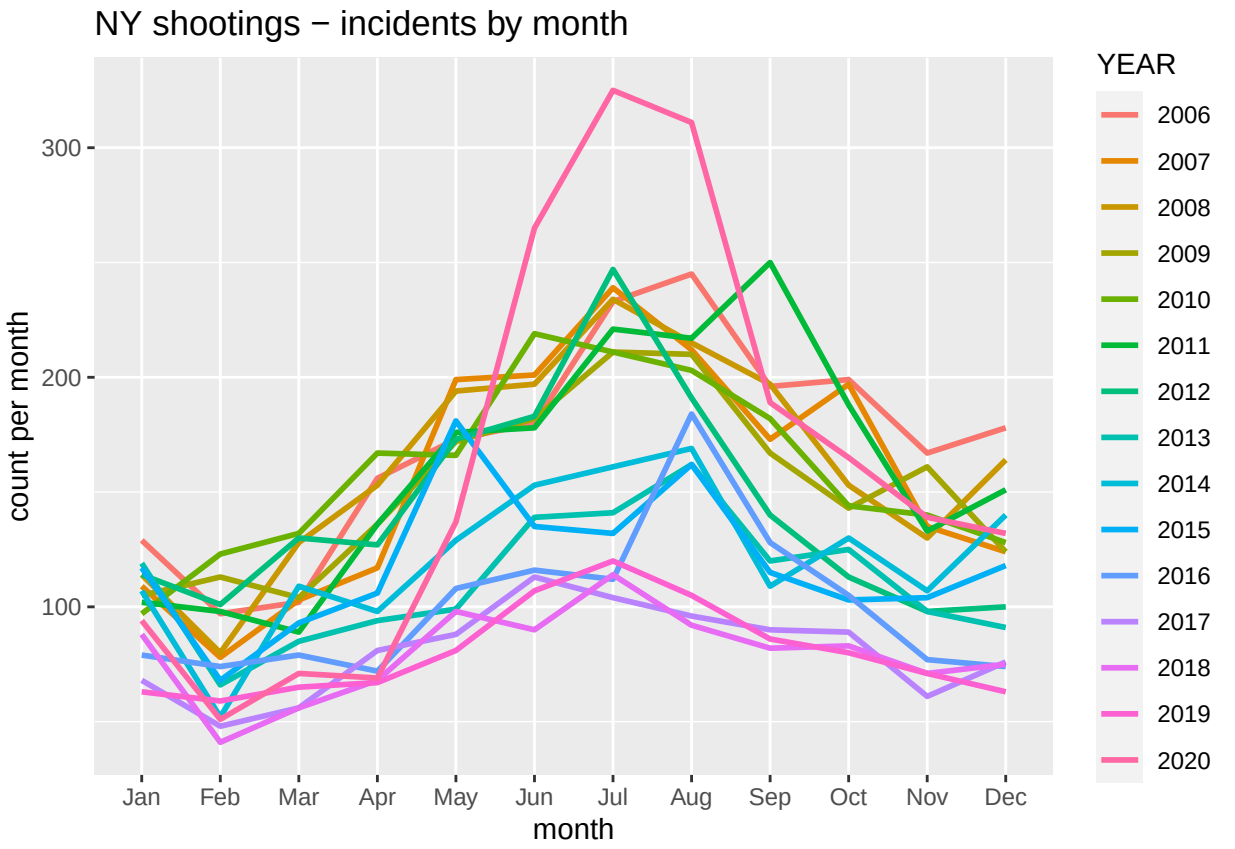
Number if shootings in different days of the week. I saw this visualization in another students' work and replicated it (with some minor changes) for the final presentation - originally I had analyzed the count of shooting for each year and time of day and visualized the months as a factor in the yearly shooting bar chart.

```
df_wd <- df %>%
  mutate(DAY_OF_WEEK=factor(wday(OCCUR_DATE, label = TRUE, locale="English_United States"))) %>%
  group_by(DAY_OF_WEEK) %>%
  count()

df_wd %>% ggplot(aes(x = DAY_OF_WEEK, y = n, fill = DAY_OF_WEEK )) +
  geom_col(show.legend = FALSE) +
  labs(title = "NYC shootings - incidents by day of the week",
  x = "day of week", y = "count")
```



In general, there were more shooting cases in summer months. I saw this visualization in another students' work and replicated it (with some changes) for the final presentation - originally I had looked analyzed and visualized the months as a factor in the yearly shooting bar chart, but I found this visualization much more informative. It also taught me how to effectively use group_by and ponder more carefully on my choice of variables for the x and y axes.

```
df_m <- df %>%
  mutate(YEAR=factor(year), MONTH=factor(month)) %>%
  group_by(YEAR, MONTH) %>%
  count() %>%
  ungroup()

df_m %>% ggplot(aes(x = MONTH, y = n, color = YEAR, group = YEAR)) +
  geom_line(size = 1) +
  labs(title = "NY shootings - incidents by month",
  x = "month", y = "count per month")
```



Linear model

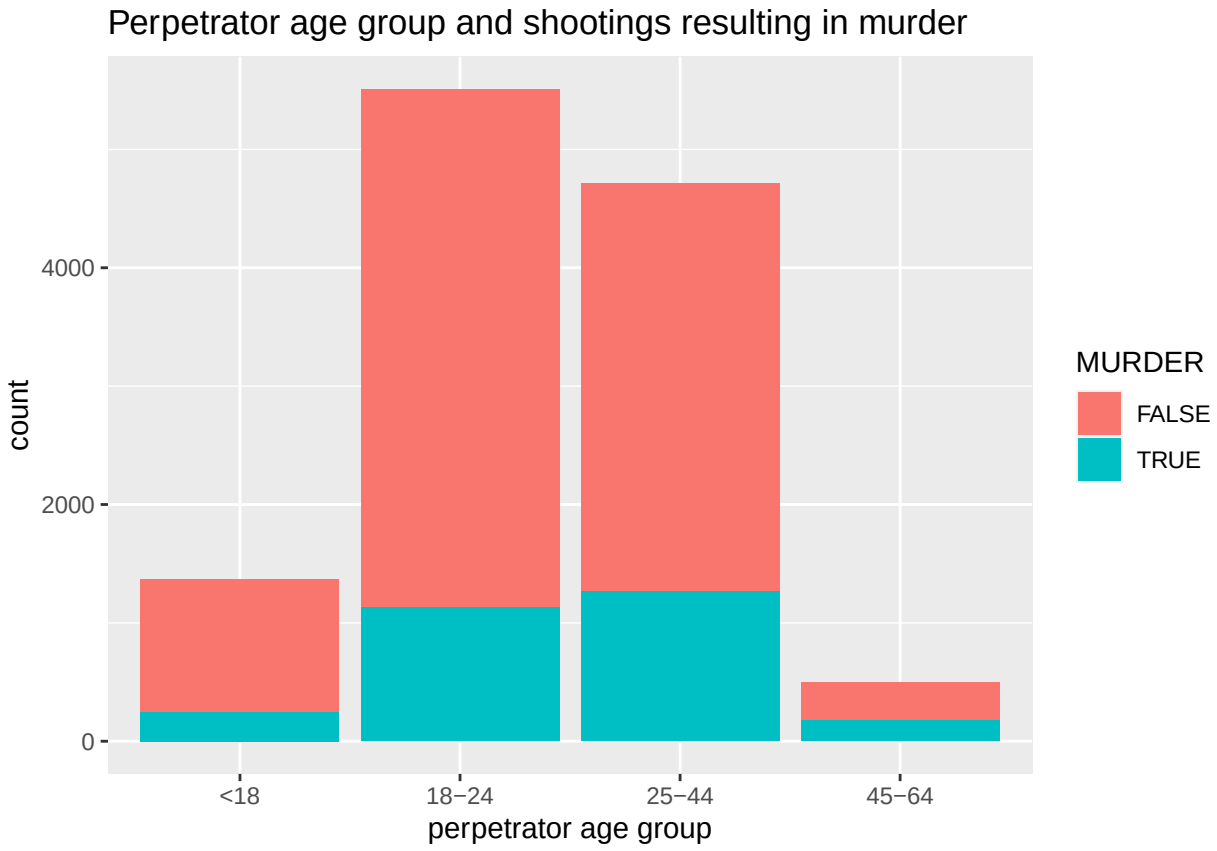
I could not figure out appropriate variables to evaluate with a linear model. I chose to evaluate 'MURDER' as a response variable and 'PERP_AGE_GROUP' as a predictor.

```
mod_1 = lm(MURDER ~ c(PERP_AGE_GROUP), data = df_perp_age_known)

summary(mod_1)
```

```
##
## Call:
## lm(formula = MURDER ~ c(PERP_AGE_GROUP), data = df_perp_age_known)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3495 -0.2686 -0.2052 -0.1806  0.8194
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.18056    0.01138  15.870  < 2e-16 ***
## c(PERP_AGE_GROUP)18-24    0.02460    0.01271   1.935    0.053 .
## c(PERP_AGE_GROUP)25-44    0.08801    0.01292   6.810 1.02e-11 ***
## c(PERP_AGE_GROUP)45-64    0.16894    0.02207   7.654 2.09e-14 ***
## ---
```

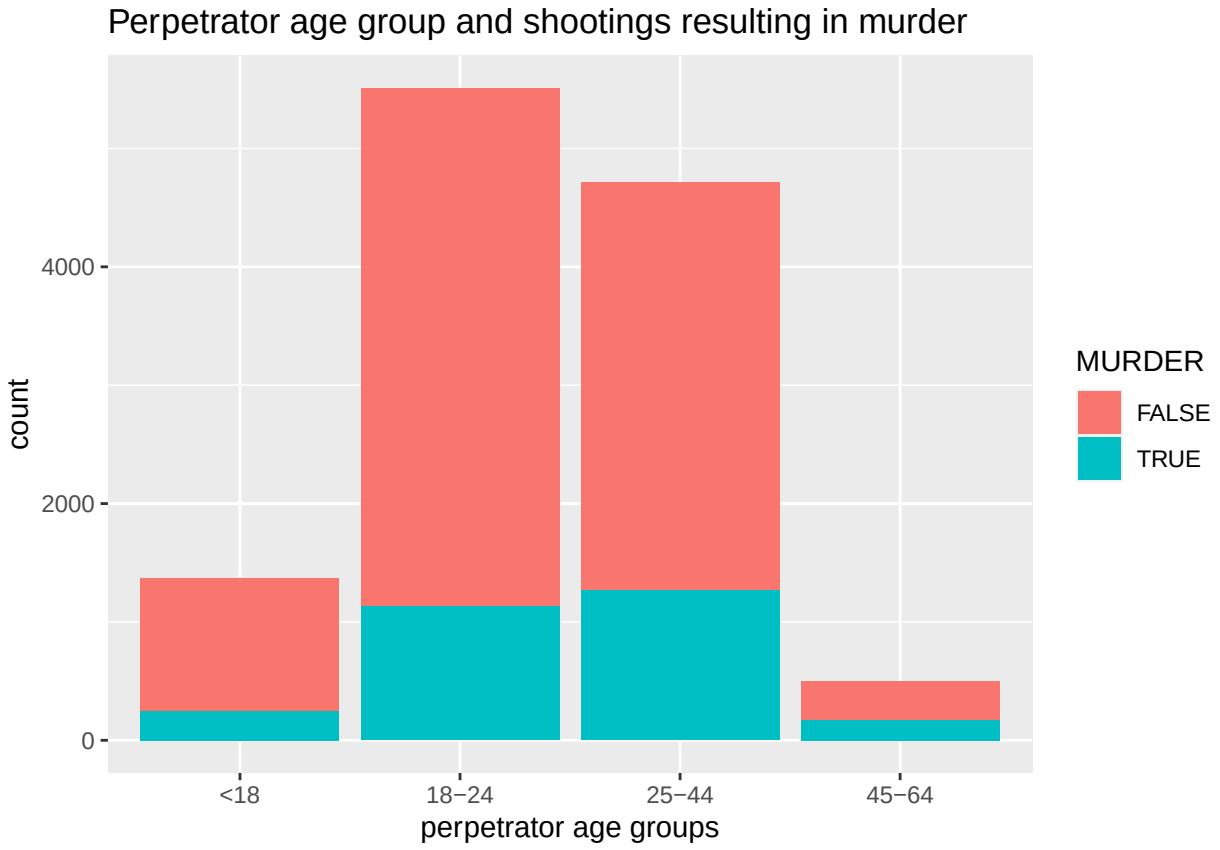
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4208 on 12081 degrees of freedom
## Multiple R-squared:  0.00959,    Adjusted R-squared:  0.009344
## F-statistic: 38.99 on 3 and 12081 DF,  p-value: < 2.2e-16
```

```
perp_age_murder <- df_perp_age_known %>%
  ggplot(aes(PERP_AGE_GROUP, fill = MURDER)) +
  geom_bar() +
  labs(title = "Perpetrator age group and shootings resulting in murder",
       x = "perpetrator age group",
       y = "count"
  )

perp_age_murder
```



Visualizing the relationship between perpetrator's age group and whether the shooting results in a murder.

```
perp_age_2 <- ggplot(df_perp_age_known, aes(x = PERP_AGE_GROUP, fill = MURDER)) +
  geom_bar() +
  labs(title = "Perpetrator age group and shootings resulting in murder",
       x = "perpetrator age groups",
       y = "count",
       fill = "MURDER")

perp_age_2
```



Sources of potential bias

As covered in week three's lecture bias may arise from multiple sources including the data scientist chose the particular topic to analyze, the questionnaire (for example, choice and wording of questions, and the multiple choice responses provided as options), the sample that is surveyed, the way missing or unknown data is handled, and how the results of the analysis are presented.

As I am working with a data set that I was already available online, many on these sources of bias are not applicable. However, the variables that I chose to analyze and the relationships that I investigated (or did not investigate - such as race) may be clouded by my biases of expecting to find some associations between variables prior to visualizing and analyzing what the data actually demonstrate. Another source of bias are the missing data.

This data set should be evaluated in the context of the other data for the NY population, such as income and education level. In addition, based on researching this data set online, each of the randomly generated INCIDENT_KEYs may be associated with more than one victim - I did not explore this aspect of the data set.

Summary

The visualizations in this assignment demonstrate the NY shootings plotted for multiple variables including borough, year, month, perpetrators' and victims' age groups, and victims' gender. In order to generate a more meaningful model, other data sets (such as population, income, education) need to be joined to this data set to determine which predictive variables can foretell shooting rates and thus facilitate strategies for decreasing shootings in NY.

```
sessionInfo()
```

Session Info

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.8.0 scales_1.2.0    forcats_0.5.1   stringr_1.4.0
##  [5] dplyr_1.0.7     purrr_0.3.4     readr_2.1.1     tidyr_1.1.4
##  [9] tibble_3.1.6    ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.30         haven_2.4.3       colorspace_2.0-2
##  [5] vctrs_0.3.8       generics_0.1.1    viridisLite_0.4.0 htmltools_0.5.2
##  [9] yaml_2.2.1        utf8_1.2.2        rlang_1.0.2       pillar_1.6.4
## [13] withr_2.4.3       glue_1.6.0        DBI_1.1.2         dbplyr_2.1.1
## [17] modelr_0.1.8      readxl_1.3.1      lifecycle_1.0.1   munsell_0.5.0
## [21] gtable_0.3.0      cellranger_1.1.0  rvest_1.0.2       evaluate_0.14
## [25] labeling_0.4.2    knitr_1.37        tzdb_0.2.0        fastmap_1.1.0
## [29] fansi_1.0.2       highr_0.9         broom_0.7.11      Rcpp_1.0.8
## [33] backports_1.4.1   jsonlite_1.7.3    farver_2.1.0      fs_1.5.2
## [37] hms_1.1.1         digest_0.6.29     stringi_1.7.6     grid_4.1.2
## [41] cli_3.1.0         tools_4.1.2       magrittr_2.0.1    crayon_1.4.2
## [45] pkgconfig_2.0.3   ellipsis_0.3.2    xml2_1.3.3        reprex_2.0.1
## [49] assertthat_0.2.1  rmarkdown_2.14    httr_1.4.2        rstudioapi_0.13
## [53] R6_2.5.1          compiler_4.1.2
```