NYPD Shooting Analysis

R. Garcia

2024-08-09

Data Download

The first step in any analysis is to obtain the required data. Here, in this step, we perform the initial Data import from the City of New York site

```r
nypd_data_raw <- read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOA
```

```
## Rows: 28562 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (12): OCCUR_DATE, BORO, LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION...
## dbl   (7): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD...
## lgl   (1): STATISTICAL_MURDER_FLAG
## time  (1): OCCUR_TIME
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
summary(nypd_data_raw)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME           BORO
##  Min.   :  9953245   Length:28562       Length:28562      Length:28562
##  1st Qu.: 65439914   Class :character   Class1:hms        Class :character
##  Median : 92711254   Mode  :character   Class2:difftime   Mode  :character
##  Mean   :127405824                      Mode  :numeric
##  3rd Qu.:203131993
##  Max.   :279758069
##
##  LOC_OF_OCCUR_DESC     PRECINCT     JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:28562       Min.   :  1.0   Min.   :0.0000    Length:28562
##  Class :character   1st Qu.: 44.0   1st Qu.:0.0000    Class :character
##  Mode  :character   Median : 67.0   Median :0.0000    Mode  :character
##                     Mean   : 65.5   Mean   :0.3219
##                     3rd Qu.: 81.0   3rd Qu.:0.0000
##                     Max.   :123.0   Max.   :2.0000
##                                     NA's   :2
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  Length:28562       Mode :logical           Length:28562
##  Class :character   FALSE:23036             Class :character
##  Mode  :character   TRUE :5526              Mode  :character
```

```
##
##
##
##
##    PERP_SEX           PERP_RACE          VIC_AGE_GROUP        VIC_SEX
##  Length:28562       Length:28562       Length:28562       Length:28562
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_RACE           X_COORD_CD        Y_COORD_CD        Latitude
##  Length:28562       Min.   : 914928   Min.   :125757   Min.   :40.51
##  Class :character   1st Qu.:1000068   1st Qu.:182912   1st Qu.:40.67
##  Mode  :character   Median :1007772   Median :194901   Median :40.70
##                     Mean   :1009424   Mean   :208380   Mean   :40.74
##                     3rd Qu.:1016807   3rd Qu.:239814   3rd Qu.:40.82
##                     Max.   :1066815   Max.   :271128   Max.   :40.91
##                                                        NA's   :59
##    Longitude         Lon_Lat
##  Min.   :-74.25   Length:28562
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA's   :59
```

From the summary, we can see that we have a set of column names that we need to interpret. Some of the columns, such as `OCCUR_DATE` are fairly straightforward, but others, such as `BORO`, which is short for "borough", might require some knowledge of the specific municipality, as other areas use similar but distinct verbiage such as "Ward", "Parish", or "District" to denote zones in or around an urban area. Other columns, such as `LOC_OF_OCCUR_DESC` aren't very obvious, so we need to inspect the data manually to see how we might interpret what's in there.

Preliminary Data Inspection, Cleanup and Preparation

For our initial data cleanup, we're going to remove columns for GPS location, as any interesting geolocation analysis is probably a bit beyond the scope of this assignment. From inspection, we see that `JURISDICTION_CODE` is a column which only has 3 unique integer values which we can't easily interpret the meaning of. That column is unlikely to be of much utility, so that too can be removed. We will also convert the character string dates and times into native date/time data.

```r
# convert to data.table
nypd_data <- data.table(nypd_data_raw)

unique(nypd_data$JURISDICTION_CODE)
```

```
## [1]  0  2  1 NA
```

```r
# remove unused columns
nypd_data <- nypd_data %>% select(-c(X_COORD_CD:Lon_Lat))
nypd_data <- nypd_data %>% select(-c(JURISDICTION_CODE))

# change date/time strings to date/time values
nypd_data <- nypd_data %>% mutate(OCCUR_DATE = mdy(OCCUR_DATE))
nypd_data <- nypd_data %>% mutate(OCCUR_TIME = hms(OCCUR_TIME))
```

Secondary Data Inspection

We want to look into the data on some of the columns that we can't immediately determine the usefulness of by name. We're looking to see what kind of values we have in various fields that may be of interest for analysis.

```r
nypd_data[, .(count = .N), by = "LOCATION_DESC"]
```

```
##                  LOCATION_DESC count
##                         <char> <int>
##  1:                VIDEO STORE     8
##  2:                     (null)  1711
##  3:                       <NA> 14977
##  4: MULTI DWELL - PUBLIC HOUS  5007
##  5:   MULTI DWELL - APT BUILD  2964
##  6:             BAR/NIGHT CLUB   668
##  7:                  PVT HOUSE   983
##  8:                       NONE   175
##  9:                SUPERMARKET    21
## 10:             GROCERY/BODEGA   750
## 11:                GAS STATION    74
## 12:            COMMERCIAL BLDG   304
## 13:                   HOSPITAL    77
## 14:          RESTAURANT/DINER   212
## 15:          BEAUTY/NAIL SALON   119
## 16:                  FAST FOOD   130
## 17:             SMALL MERCHANT    44
## 18:         STORE UNCLASSIFIED    37
## 19:              VARIETY STORE    11
## 20:               LIQUOR STORE    42
## 21:          FACTORY/WAREHOUSE     8
## 22: SOCIAL CLUB/POLICY LOCATI    73
## 23:        DRY CLEANER/LAUNDRY    32
## 24:          CLOTHING BOUTIQUE    14
## 25:                 SHOE STORE    10
## 26:              JEWELRY STORE    14
## 27:       GYM/FITNESS FACILITY     4
## 28:                HOTEL/MOTEL    35
## 29:                CANDY STORE     7
## 30:                 DEPT STORE     9
## 31:                       BANK     3
## 32:            TELECOMM. STORE    11
## 33:                CHAIN STORE     7
## 34:                 DRUG STORE    14
```

```
## 35:                 LOAN COMPANY     1
## 36:                   CHECK CASH     1
## 37:                       SCHOOL     1
## 38:             STORAGE FACILITY     1
## 39:             PHOTO/COPY STORE     1
## 40:                          ATM     1
## 41:               DOCTOR/DENTIST     1
##                  LOCATION_DESC count
```

```
nypd_data[, .(count = .N), by = "BORO"]
```

```
##             BORO count
##           <char> <int>
## 1:     MANHATTAN  3762
## 2:         BRONX  8376
## 3:        QUEENS  4271
## 4:      BROOKLYN 11346
## 5: STATEN ISLAND   807
```

```
nypd_data[, .(count = .N), by = "LOC_CLASSFCTN_DESC"]
```

```
##     LOC_CLASSFCTN_DESC count
##                 <char> <int>
##  1:         COMMERCIAL   208
##  2:             STREET  1886
##  3:               <NA> 25596
##  4:            HOUSING   460
##  5:           DWELLING   243
##  6:              OTHER    59
##  7:         PLAYGROUND    41
##  8:            VEHICLE    29
##  9:            TRANSIT    23
## 10:        PARKING LOT    15
## 11:             (null)     2
```

```
nypd_data[, .(count = .N), by = "LOC_OF_OCCUR_DESC"]
```

```
##    LOC_OF_OCCUR_DESC count
##               <char> <int>
## 1:            INSIDE   460
## 2:           OUTSIDE  2506
## 3:              <NA> 25596
```

```
nypd_data[!is.na("LOC_CLASSFCTN_DESC"), .N, by=PRECINCT]
```

```
##     PRECINCT     N
##        <num> <int>
##   1:      14    61
##   2:      48   841
##   3:     103   605
##   4:      42   890
```

```
##  5:       83   520
##  6:       23   505
##  7:      113   834
##  8:       77   821
##  9:       49   368
## 10:       73  1500
## 11:      114   397
## 12:       28   353
## 13:       43   796
## 14:       71   595
## 15:      106   233
## 16:      105   488
## 17:        7   120
## 18:       41   519
## 19:       47  1006
## 20:       46   972
## 21:       32   663
## 22:      108    75
## 23:      100   178
## 24:      110   174
## 25:       75  1628
## 26:       67  1259
## 27:       44  1076
## 28:       84   131
## 29:       88   294
## 30:       79  1045
## 31:       50   162
## 32:       94    87
## 33:       40   947
## 34:       45   195
## 35:      101   502
## 36:       70   479
## 37:       60   383
## 38:       52   604
## 39:       63   292
## 40:       81   821
## 41:       69   484
## 42:      104   108
## 43:       34   335
## 44:       20    43
## 45:      115   185
## 46:      121   114
## 47:       61   157
## 48:        9   114
## 49:      107   105
## 50:      120   597
## 51:       68    36
## 52:       66    53
## 53:       24   113
## 54:        1    25
## 55:       25   494
## 56:       30   234
## 57:       62    72
## 58:       33   242
```

```
## 59:       26   157
## 60:       90   328
## 61:       76   179
## 62:       18    38
## 63:      123    33
## 64:       10    74
## 65:       19    24
## 66:      102   229
## 67:       78    65
## 68:      122    63
## 69:        6    28
## 70:      109   123
## 71:       72   117
## 72:        5    67
## 73:      112    23
## 74:       13    61
## 75:      111    12
## 76:       17    10
## 77:       22     1
##     PRECINCT     N
```

```
nypd_data[!is.na("LOC_OF_OCCURCLASSFCTN_DESC"), .N, by=PRECINCT]
```

```
##     PRECINCT     N
##        <num> <int>
##  1:       14    61
##  2:       48   841
##  3:      103   605
##  4:       42   890
##  5:       83   520
##  6:       23   505
##  7:      113   834
##  8:       77   821
##  9:       49   368
## 10:       73  1500
## 11:      114   397
## 12:       28   353
## 13:       43   796
## 14:       71   595
## 15:      106   233
## 16:      105   488
## 17:        7   120
## 18:       41   519
## 19:       47  1006
## 20:       46   972
## 21:       32   663
## 22:      108    75
## 23:      100   178
## 24:      110   174
## 25:       75  1628
## 26:       67  1259
## 27:       44  1076
## 28:       84   131
## 29:       88   294
```

```
## 30:       79  1045
## 31:       50   162
## 32:       94    87
## 33:       40   947
## 34:       45   195
## 35:      101   502
## 36:       70   479
## 37:       60   383
## 38:       52   604
## 39:       63   292
## 40:       81   821
## 41:       69   484
## 42:      104   108
## 43:       34   335
## 44:       20    43
## 45:      115   185
## 46:      121   114
## 47:       61   157
## 48:        9   114
## 49:      107   105
## 50:      120   597
## 51:       68    36
## 52:       66    53
## 53:       24   113
## 54:        1    25
## 55:       25   494
## 56:       30   234
## 57:       62    72
## 58:       33   242
## 59:       26   157
## 60:       90   328
## 61:       76   179
## 62:       18    38
## 63:      123    33
## 64:       10    74
## 65:       19    24
## 66:      102   229
## 67:       78    65
## 68:      122    63
## 69:        6    28
## 70:      109   123
## 71:       72   117
## 72:        5    67
## 73:      112    23
## 74:       13    61
## 75:      111    12
## 76:       17    10
## 77:       22     1
##     PRECINCT     N
```

From the initial data inspection, we can see that some of the columns offer limited utility. `LOC_OF_OCCUR_DESC`, for example, has only 3 distinct values, `INSIDE`, `OUTSIDE`, and `NA`. Further, the `NA` values make up over 90% of the entries, meaning that the non-empty values which we do have for that column are of limited meaning. Another potentially limited column is `LOC_CLASSFCTN_DESC`, which also has a high rate of `NA` values. Curiously, the number of NA values in the two columns matches exactly, so a future useful direction may be to see if any precincts have consistent reporting on this value, and may offer a potential insight into the rates at which these values occur in general. However, we see from inspection that reports that have both values are spread across precincts and boroughs, indicating that we do not have sufficient data to inspect those values, so we drop them from this analysis to tighten our scope. Additionally, we see that the column LOCATION_DESC has character strings of "(null)" values which are strings and not actually null and should be changed to NA for consistency.

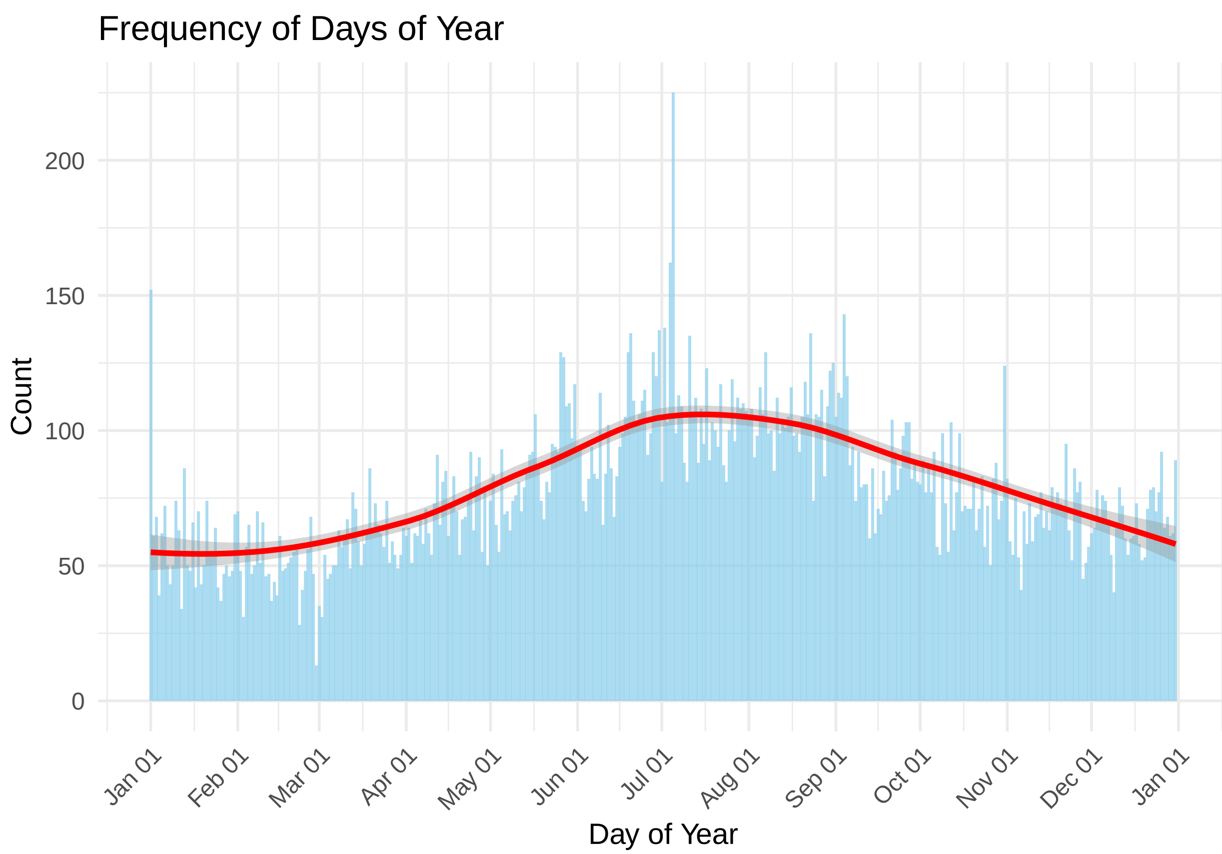
```
nypd_data[LOCATION_DESC == "(null)", LOCATION_DESC := NA]

nypd_data[VIC_RACE == "(null)", VIC_RACE := NA]
nypd_data[PERP_RACE == "(null)", PERP_RACE := NA]
nypd_data[VIC_AGE_GROUP == "(null)", VIC_AGE_GROUP := NA]
nypd_data[PERP_AGE_GROUP == "(null)", PERP_AGE_GROUP := NA]
nypd_data[VIC_SEX == "(null)", VIC_SEX := NA]
nypd_data[PERP_SEX == "(null)", PERP_SEX := NA]

nypd_data <- nypd_data %>% select(-c(LOC_OF_OCCUR_DESC))
nypd_data <- nypd_data %>% select(-c(LOC_CLASSFCTN_DESC))
```

Analysis and Visualization

Initial visualiation of potential areas of interest

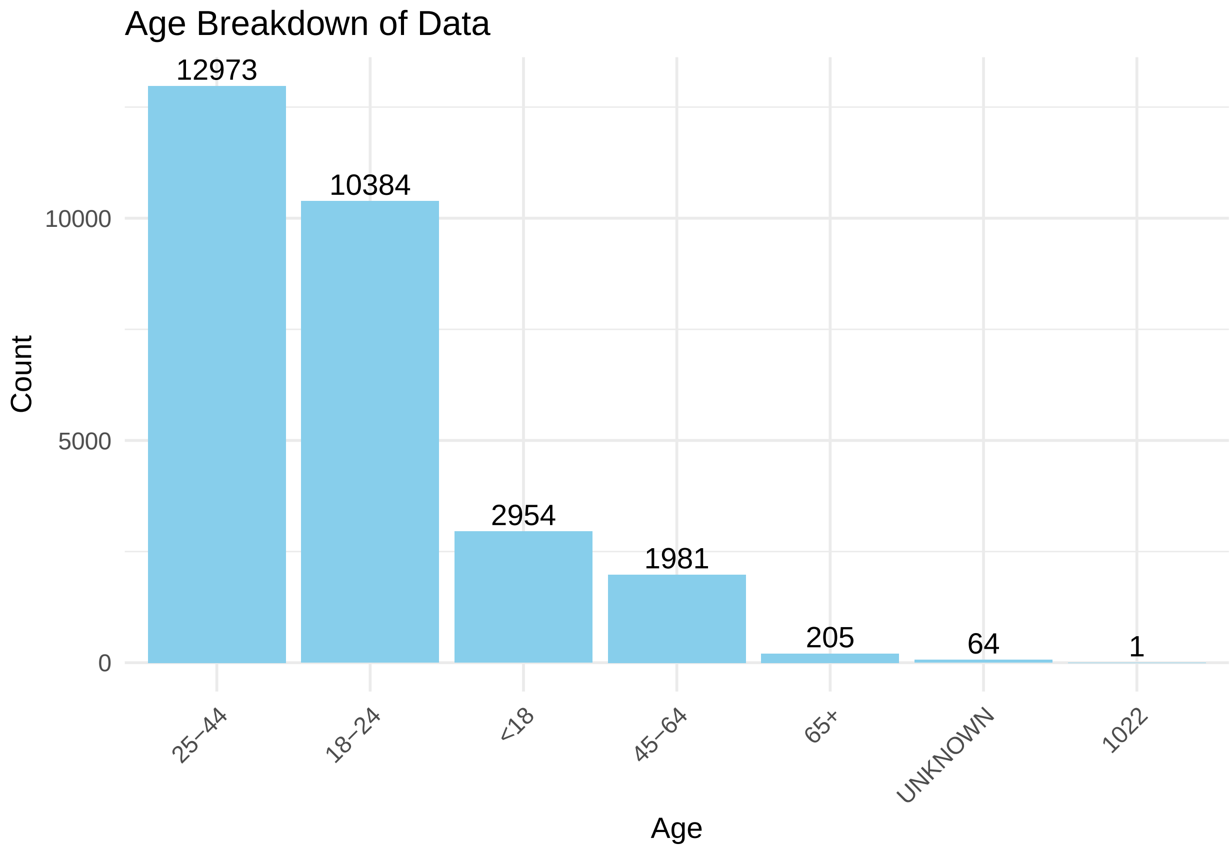
Create some initial visualizations to get a sense of how the data breaks down across various lines. On this initial graph, I'm breaking down the dates to strip off the years to see if we can identify any season trends in the data. I primarily chose this because I wanted to try adding in a smoothed line to show a curve for the seasonal trends.

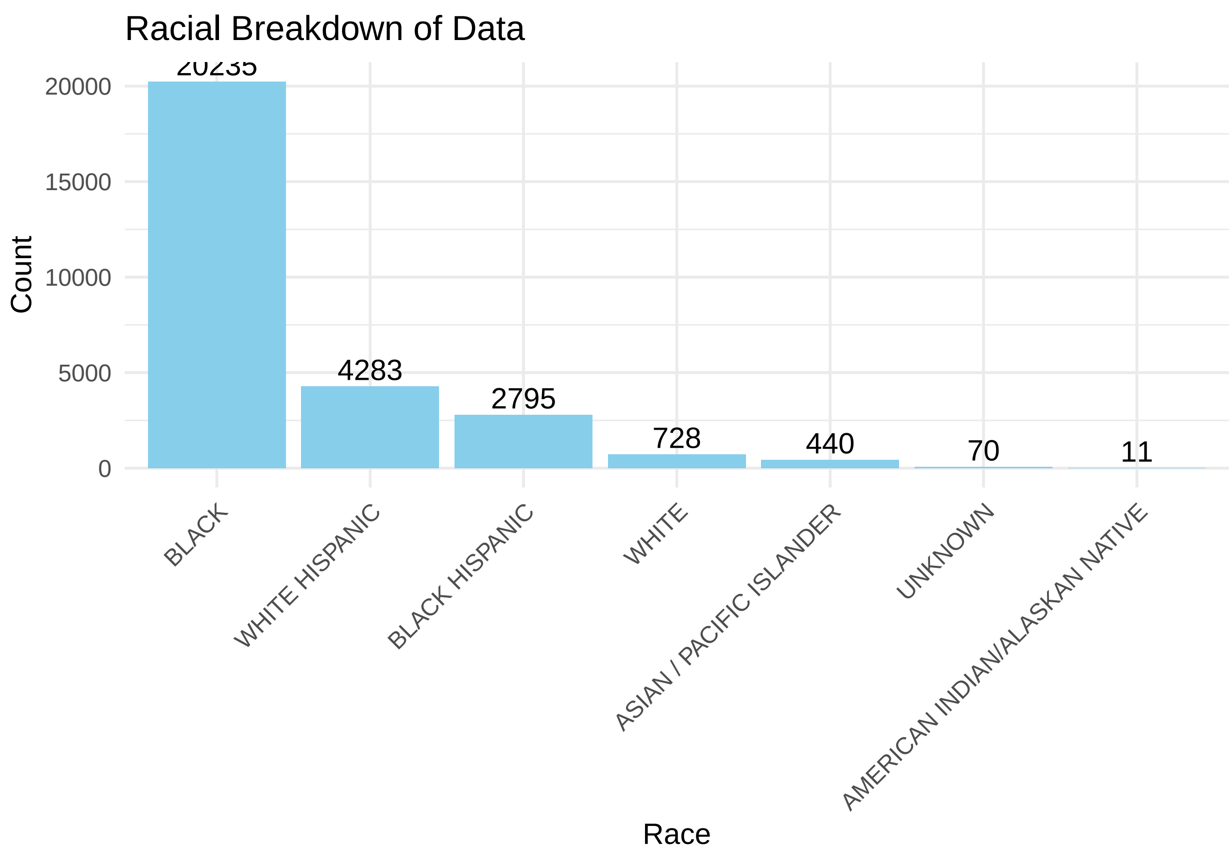
```
## `geom_smooth()` using formula = 'y ~ x'
```

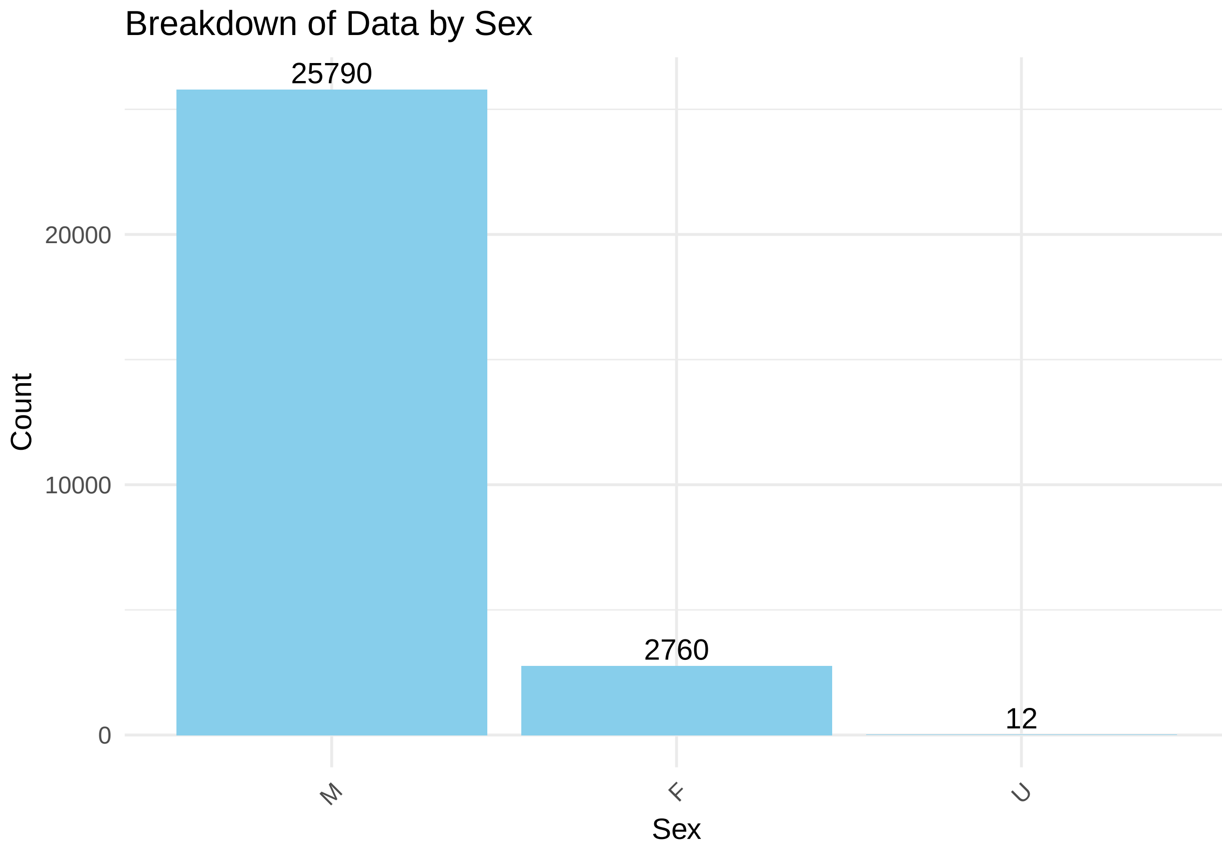



```
## Day with maximum shootings:

## Key: <DayOfYear>
##    DayOfYear
##       <char>
## 1:     07-05
```

On these next three, I create relatively simple bar graphs to create breakdowns by age, race and sex.


Age Breakdown of Data
12973
10384
2954
1981
205
64
1
10000
5000
0
Count
25–44
18–24
<18
45–64
65+
UNKNOWN
1022
Age

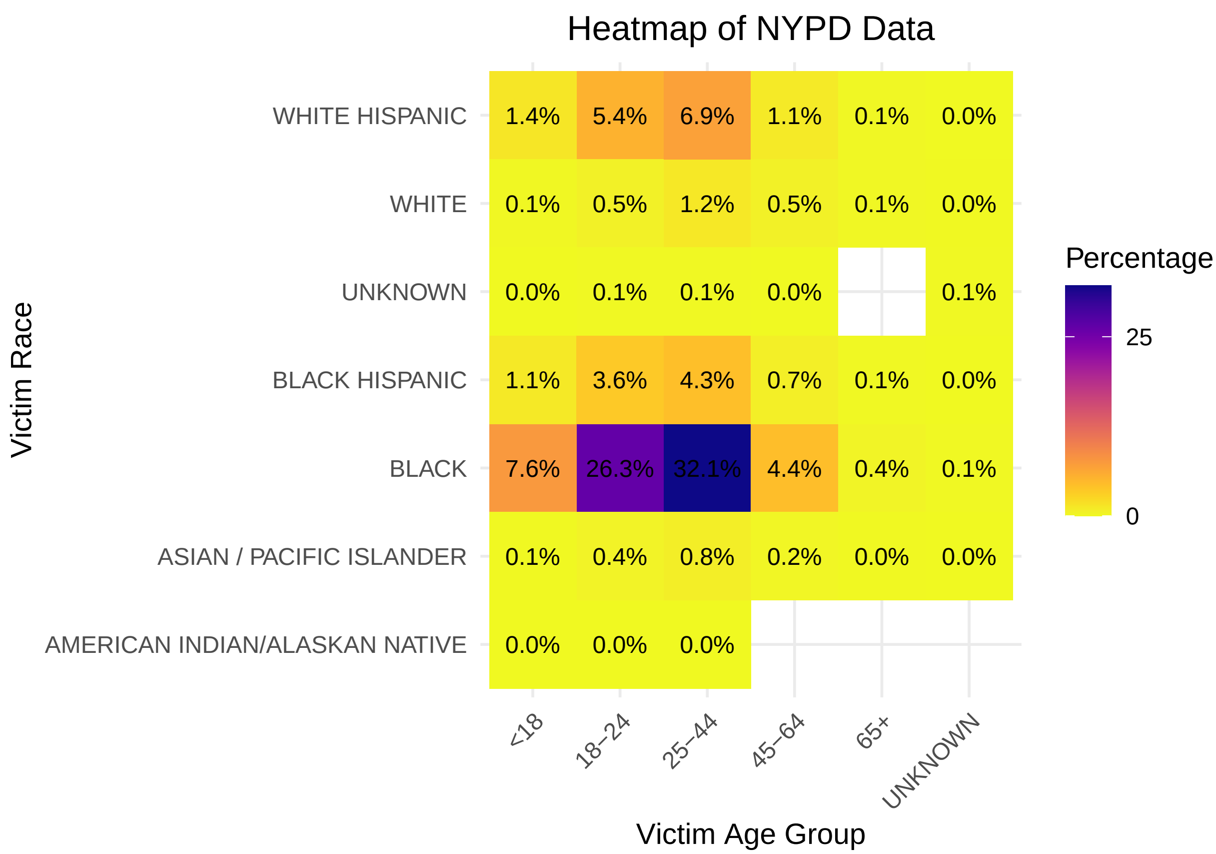

Racial Breakdown of Data
Count
Race
20000
15000
10000
5000
0
20235
4283
2795
728
440
70
11
BLACK
WHITE HISPANIC
BLACK HISPANIC
WHITE
ASIAN / PACIFIC ISLANDER
UNKNOWN
AMERICAN INDIAN/ALASKAN NATIVE



Investigate Correlation between Age and Race

From the initial visualizations, we can see that age and race are two factors that, when taken apart, seem highly correlated to shootings. Sex is also another factor, but it is so highly correlated to males that it might not be worth investigating nuances on that factor in this analysis. So next, we want to look deeper and see how age and race together are related to shootings, and how we can represent this data visually for both factors.

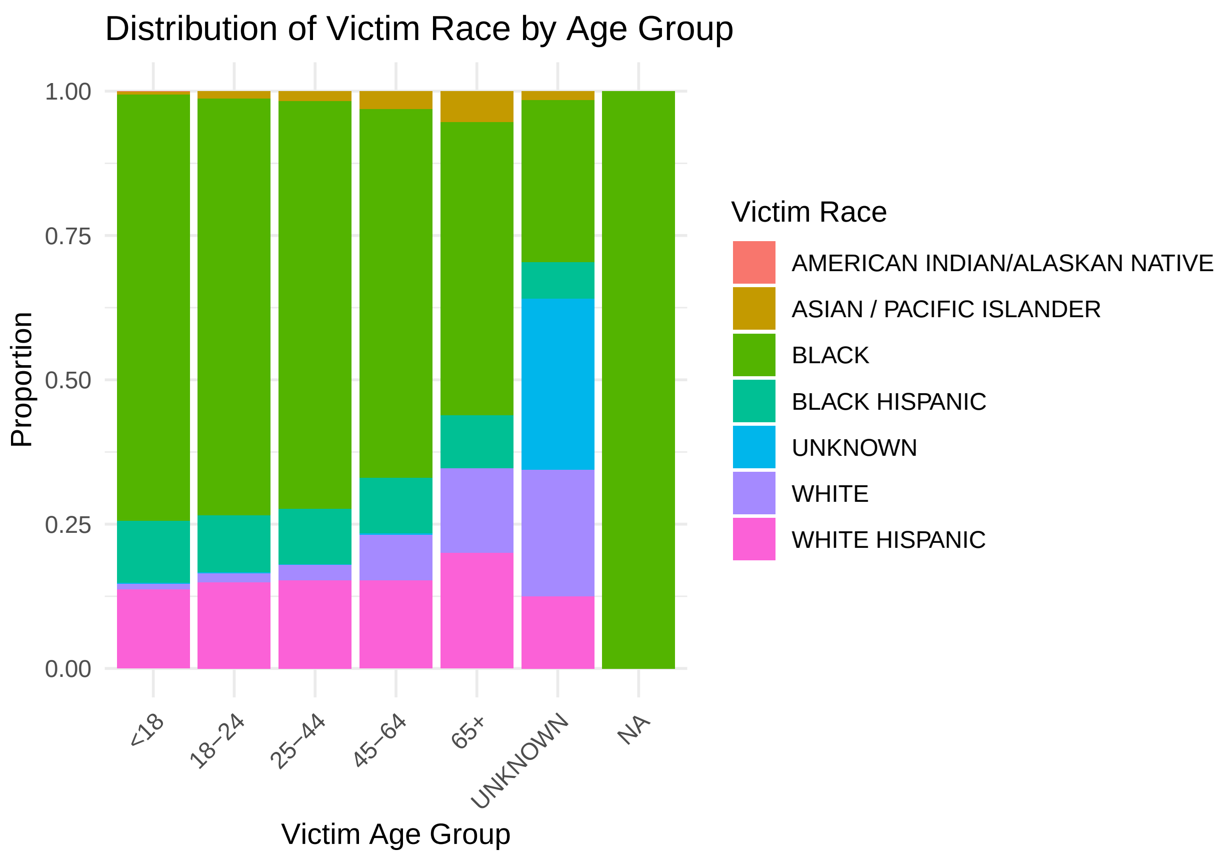
Heatmap Visualization To create a visualization for both age and race, we create a heatmap to try to visualize outliers of age and racial groupings.



From the heatmap, we can see that there are distinct areas where particular groups are significantly over-represented in the population. The biggest outlier is two age groups of "18-24" and "25-44" for blacks. There is a smaller but easily identifiable rise for hispanics, both black and white.

ChiSq Correlation Model and Distribution Chart Another part of the assignment was to create a model. So here, we create a model to give us a look at the distribution of the age of the victims within their race. In this model, I'm using chi square value and Cramer's V.

```
## Warning in chisq.test(cont_table): Chi-squared approximation may be incorrect

## Chi-square test:

##
##  Pearson's Chi-squared test
##
## data:  cont_table
## X-squared = 2919.7, df = 30, p-value < 2.2e-16

##
## Cramer's V:

## [1] 0.1429884
```

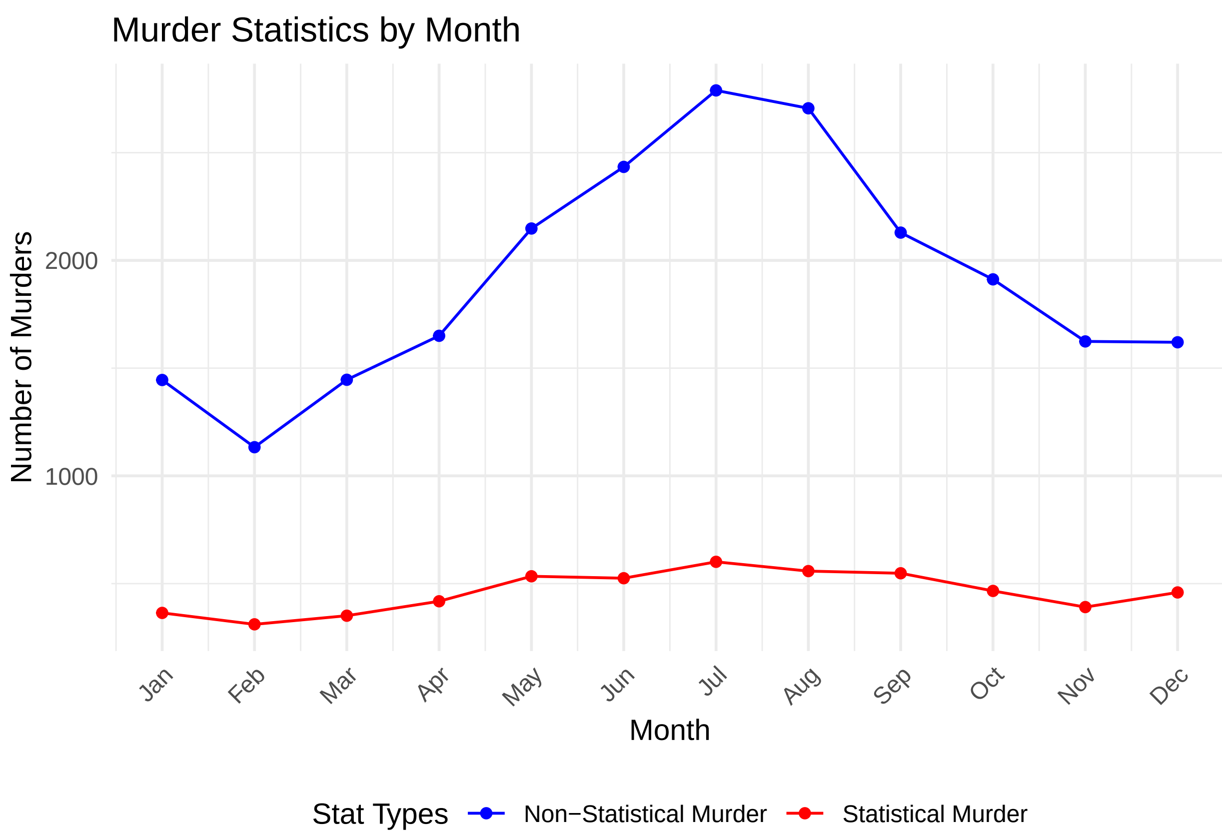


From this result, we see there is a very low p-value, indicating that the two factors, age and race, are highly likely to be correlated in shootings, and there is likely a meaningful association. However, the Cramer's V score is only 0.14, which denotes an association, but not a strong one. These two factors tell us together that while there is a significant association between age, race, and shootings, a significant portion of the data also falls outside of those two factors, meaning they alone do not significantly explain shooting frequency. On the graph, I created this primarily to try out a new style of plot, and it offers us a look at a breakdown of the racial representations across age groups, and here, we see how victim's demographics change as age increases, which can be an interesting trend.

Seasonal Shootings by Murder flag One of the other initial ideas I had to look at the data was on seasonal trends. We also had some data provided on statistical murders and non-statistical murders, although I don't know exactly what the distinction there is. So I'm going to break the two factors apart, graph them, and see if it tells me anything.

```
## # A tibble: 12 x 3
##    MONTH STAT_MURDER NON_STAT_MURDER
##    <int>       <dbl>           <dbl>
##  1     1         364            1445
##  2     2         311            1133
##  3     3         351            1446
##  4     4         418            1650
##  5     5         534            2148
##  6     6         525            2434
##  7     7         601            2789
##  8     8         558            2706
##  9     9         548            2129
```

```
## 10     10          466              1912
## 11     11          391              1624
## 12     12          459              1620
```



Here, while we can still see the seasonal trends reflected in both factors, the statistical murder make up a pretty small percentage of the shootings. It's not apparent if this tells us anything, but I do question if I am interpreting that column correctly. More research and lookups are required there.

Identification of Bias

Sources of bias in the data include: * Error and bias in the initial data collection and recording * Incomplete data and differences in data collection among precincts

Personal Bias: * Assumptions made by the researcher and analyst, including which data to trust and include. * Assumptions made about the meaning of the data and some of the field names.

Summary and Conclusion

From the data, we are able to conclude that there is a strong correlation between several factors in the data and shootings. Strongest correlations are race, age, and gender.

There are also significant indications that seasonal trends are involved as well, as there is significant increase in summer months. Differences in data collection and recording make it difficult to determine if there are signicant differences in shooting rates in various boroughs, or if the differences are due to variations in data recording.

R session information

```
## R version 4.4.1 (2024-06-14)
## Platform: aarch64-apple-darwin23.4.0
## Running under: macOS Sonoma 14.6.1
##
## Matrix products: default
## BLAS:   /opt/homebrew/Cellar/openblas/0.3.28/lib/libopenblasp-r0.3.28.dylib
## LAPACK: /opt/homebrew/Cellar/r/4.4.1/lib/R/lib/libRlapack.dylib;  LAPACK version 3.12.0
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Chicago
## tzcode source: internal
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] vcd_1.4-12        viridis_0.6.5     viridisLite_0.4.2 data.table_1.15.4
##  [5] lubridate_1.9.3   forcats_1.0.0     stringr_1.5.1     dplyr_1.1.4
##  [9] purrr_1.0.2       readr_2.1.5       tidyr_1.3.1       tibble_3.2.1
## [13] ggplot2_3.5.1     tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.3    stringi_1.8.4     lattice_0.22-6
##  [5] hms_1.1.3         digest_0.6.36     magrittr_2.0.3    evaluate_0.24.0
##  [9] timechange_0.3.0  fastmap_1.2.0     Matrix_1.7-0      gridExtra_2.3
## [13] mgcv_1.9-1        fansi_1.0.6       scales_1.3.0      cli_3.6.3
## [17] crayon_1.5.3      rlang_1.1.4       splines_4.4.1     bit64_4.0.5
## [21] munsell_0.5.1     withr_3.0.1       yaml_2.3.10       parallel_4.4.1
## [25] tools_4.4.1       tzdb_0.4.0        colorspace_2.1-1  curl_5.2.1
## [29] vctrs_0.6.5       R6_2.5.1          zoo_1.8-12        lifecycle_1.0.4
## [33] bit_4.0.5         vroom_1.6.5       MASS_7.3-60.2     pkgconfig_2.0.3
## [37] pillar_1.9.0      gtable_0.3.5      glue_1.7.0        highr_0.11
## [41] xfun_0.46         lmtest_0.9-40     tidyselect_1.2.1  rstudioapi_0.16.0
## [45] knitr_1.48        farver_2.1.2      nlme_3.1-164      htmltools_0.5.8.1
## [49] labeling_0.4.3    rmarkdown_2.27    compiler_4.4.1
```