JosiahBall_NYPD_DataAnalysis

Josiah Ball

2025-05-30

Overview

The purpose of this document is to glorify Jesus Christ by learning proper data analysis in R.

In this R Markdown file, I will:

1. Overview the problem
2. Overview and describe the dataset
3. Import and tidying the data
4. Perform exploratory data analysis
5. Train and test a predictive model

The Problem

In the show "Person of interest" the main characters Harold Finch and John Reese use a massive data-driven artificial intelligence model called "the Machine" to predict which Social Security Number is either in danger of either doing a violent crime or having a violent crime done to them. This workbook is meant to be a mini-"The Machine". We will look at the crime data from the New York police Department (NYPD), explore and tidy the data, and run a logistic analysis to see if we can predict where crimes are more likely to be murders.

The Dataset

The dataset used in this analysis is the "NYPD Shooting Incident Data (Historic)" public dataset from the data.gov data catalog and may be found here: https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic.[1] As the source explains, this dataset contains "every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year."[2] More information about the dataset may be seen in the exploratory data analysis section below. It is important to note much of the R code in this document was informed by the help of ChatGPT.[3]

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

```
## Rows: 29744 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (12): OCCUR_DATE, BORO, LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION...
## dbl   (5): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, Latitude, Longitude
## num   (2): 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
head(df)
```

```
## # A tibble: 6 x 21
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME BORO     LOC_OF_OCCUR_DESC PRECINCT
##          <dbl> <chr>      <time>     <chr>    <chr>                <dbl>
## 1    231974218 08/09/2021 01:06      BRONX    <NA>                    40
## 2    177934247 04/07/2018 19:48      BROOKLYN <NA>                    79
## 3    255028563 12/02/2022 22:57      BRONX    OUTSIDE                 47
## 4     25384540 11/19/2006 01:50      BROOKLYN <NA>                    66
## 5     72616285 05/09/2010 01:58      BRONX    <NA>                    46
## 6     85875439 07/22/2012 21:35      BRONX    <NA>                    42
## # i 15 more variables: JURISDICTION_CODE <dbl>, LOC_CLASSFCTN_DESC <chr>,
## #   LOCATION_DESC <chr>, STATISTICAL_MURDER_FLAG <lgl>, PERP_AGE_GROUP <chr>,
## #   PERP_SEX <chr>, PERP_RACE <chr>, VIC_AGE_GROUP <chr>, VIC_SEX <chr>,
## #   VIC_RACE <chr>, X_COORD_CD <dbl>, Y_COORD_CD <dbl>, Latitude <dbl>,
## #   Longitude <dbl>, Lon_Lat <chr>
```

Tidy Data

Now that the data is uploaded, we want to tidy the data. Specifically, we will examine:

1. Examine the structure of the dataset/columns
2. Drop unnecessary columns
3. Handle NA values column-by-column

```r
glimpse(df)
```

```
## Rows: 29,744
## Columns: 21
## $ INCIDENT_KEY            <dbl> 231974218, 177934247, 255028563, 25384540, 726~
## $ OCCUR_DATE              <chr> "08/09/2021", "04/07/2018", "12/02/2022", "11/~
## $ OCCUR_TIME              <time> 01:06:00, 19:48:00, 22:57:00, 01:50:00, 01:58~
## $ BORO                    <chr> "BRONX", "BROOKLYN", "BRONX", "BROOKLYN", "BRO~
## $ LOC_OF_OCCUR_DESC       <chr> NA, NA, "OUTSIDE", NA, NA, NA, NA, NA, NA, NA,~
## $ PRECINCT                <dbl> 40, 79, 47, 66, 46, 42, 71, 69, 75, 69, 40, 42~
## $ JURISDICTION_CODE       <dbl> 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 2, 0, 0, 2, 0~
## $ LOC_CLASSFCTN_DESC      <chr> NA, NA, "STREET", NA, NA, NA, NA, NA, NA, NA, ~
## $ LOCATION_DESC           <chr> NA, NA, "GROCERY/BODEGA", "PVT HOUSE", "MULTI ~
## $ STATISTICAL_MURDER_FLAG <lgl> FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, F~
## $ PERP_AGE_GROUP          <chr> NA, "25-44", "(null)", "UNKNOWN", "25-44", "18~
## $ PERP_SEX                <chr> NA, "M", "(null)", "U", "M", "M", NA, NA, "M",~
## $ PERP_RACE               <chr> NA, "WHITE HISPANIC", "(null)", "UNKNOWN", "BL~
## $ VIC_AGE_GROUP           <chr> "18-24", "25-44", "25-44", "18-24", "<18", "18~
## $ VIC_SEX                 <chr> "M", "M", "M", "M", "F", "M", "M", "M", "M", "~
## $ VIC_RACE                <chr> "BLACK", "BLACK", "BLACK", "BLACK", "BLACK", "~
```

```
## $ X_COORD_CD              <dbl> 1006343.0, 1000082.9, 1020691.0, 985107.3, 100~
## $ Y_COORD_CD              <dbl> 234270.0, 189064.7, 257125.0, 173349.8, 247502~
## $ Latitude                <dbl> 40.80967, 40.68561, 40.87235, 40.64249, 40.845~
## $ Longitude               <dbl> -73.92019, -73.94291, -73.86823, -73.99691, -7~
## $ Lon_Lat                 <chr> "POINT (-73.92019278899994 40.80967347200004)"~
```

We can see from glimpsing the data that many columns are not in the correct format. We will first correct the column data types.

```
# Change columns which should be factors into factors
df <- df %>%
  mutate(across(c(BORO,
                LOC_OF_OCCUR_DESC,
                LOC_CLASSFCTN_DESC,
                LOCATION_DESC,
                PERP_AGE_GROUP,
                PERP_SEX,
                PERP_RACE,
                VIC_AGE_GROUP,
                VIC_SEX,
                VIC_RACE), as.factor))

# Change columns which should be dates into dates
df <- df %>%
  mutate(across(c(OCCUR_DATE), ~ as.Date(., format = "%m/%d/%y")))

# Change columns which should be boolean into boolean
df <- df %>%
  mutate(across(c(STATISTICAL_MURDER_FLAG), as.logical))

# Review data types
glimpse(df)
```

```
## Rows: 29,744
## Columns: 21
## $ INCIDENT_KEY            <dbl> 231974218, 177934247, 255028563, 25384540, 726~
## $ OCCUR_DATE              <date> 2020-08-09, 2020-04-07, 2020-12-02, 2020-11-1~
## $ OCCUR_TIME              <time> 01:06:00, 19:48:00, 22:57:00, 01:50:00, 01:58~
## $ BORO                    <fct> BRONX, BROOKLYN, BRONX, BROOKLYN, BRONX, BRONX~
## $ LOC_OF_OCCUR_DESC       <fct> NA, NA, OUTSIDE, NA, NA, NA, NA, NA, NA, NA, N~
## $ PRECINCT                <dbl> 40, 79, 47, 66, 46, 42, 71, 69, 75, 69, 40, 42~
## $ JURISDICTION_CODE       <dbl> 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0, 2, 0, 0, 2, 0~
## $ LOC_CLASSFCTN_DESC      <fct> NA, NA, STREET, NA, NA, NA, NA, NA, NA, NA, NA~
## $ LOCATION_DESC           <fct> NA, NA, GROCERY/BODEGA, PVT HOUSE, MULTI DWELL~
## $ STATISTICAL_MURDER_FLAG <lgl> FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, F~
## $ PERP_AGE_GROUP          <fct> NA, 25-44, (null), UNKNOWN, 25-44, 18-24, NA, ~
## $ PERP_SEX                <fct> NA, M, (null), U, M, M, NA, NA, M, M, M, M, M,~
## $ PERP_RACE               <fct> NA, WHITE HISPANIC, (null), UNKNOWN, BLACK, BL~
## $ VIC_AGE_GROUP           <fct> 18-24, 25-44, 25-44, 18-24, <18, 18-24, 25-44,~
## $ VIC_SEX                 <fct> M, M, M, M, F, M, M, M, M, M, M, M, M, M, M, M~
## $ VIC_RACE                <fct> BLACK, BLACK, BLACK, BLACK, BLACK, BLACK, BLAC~
## $ X_COORD_CD              <dbl> 1006343.0, 1000082.9, 1020691.0, 985107.3, 100~
## $ Y_COORD_CD              <dbl> 234270.0, 189064.7, 257125.0, 173349.8, 247502~
```

```
## $ Latitude               <dbl> 40.80967, 40.68561, 40.87235, 40.64249, 40.845~
## $ Longitude              <dbl> -73.92019, -73.94291, -73.86823, -73.99691, -7~
## $ Lon_Lat                <chr> "POINT (-73.92019278899994 40.80967347200004)"~
```

Now we can drop the columns which we will not be utilizing in our analysis.

```r
# Drop unnecessary columns
df <- subset(df, select =  -c(PRECINCT,
                    JURISDICTION_CODE,
                    X_COORD_CD,
                    Y_COORD_CD,
                    Latitude,
                    Longitude,
                    Lon_Lat))

glimpse(df)
```

```
## Rows: 29,744
## Columns: 14
## $ INCIDENT_KEY            <dbl> 231974218, 177934247, 255028563, 25384540, 726~
## $ OCCUR_DATE              <date> 2020-08-09, 2020-04-07, 2020-12-02, 2020-11-1~
## $ OCCUR_TIME              <time> 01:06:00, 19:48:00, 22:57:00, 01:50:00, 01:58~
## $ BORO                    <fct> BRONX, BROOKLYN, BRONX, BROOKLYN, BRONX, BRONX~
## $ LOC_OF_OCCUR_DESC       <fct> NA, NA, OUTSIDE, NA, NA, NA, NA, NA, NA, NA, N~
## $ LOC_CLASSFCTN_DESC      <fct> NA, NA, STREET, NA, NA, NA, NA, NA, NA, NA, NA~
## $ LOCATION_DESC           <fct> NA, NA, GROCERY/BODEGA, PVT HOUSE, MULTI DWELL~
## $ STATISTICAL_MURDER_FLAG <lgl> FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, F~
## $ PERP_AGE_GROUP          <fct> NA, 25-44, (null), UNKNOWN, 25-44, 18-24, NA, ~
## $ PERP_SEX                <fct> NA, M, (null), U, M, M, NA, NA, M, M, M, M, M,~
## $ PERP_RACE               <fct> NA, WHITE HISPANIC, (null), UNKNOWN, BLACK, BL~
## $ VIC_AGE_GROUP           <fct> 18-24, 25-44, 25-44, 18-24, <18, 18-24, 25-44,~
## $ VIC_SEX                 <fct> M, M, M, M, F, M, M, M, M, M, M, M, M, M, M, M~
## $ VIC_RACE                <fct> BLACK, BLACK, BLACK, BLACK, BLACK, BLACK, BLAC~
```

Now we may begin NULL handling column by column. We will begin by examining the whole table using skim.

```r
skim(df)
```

Table 1: Data summary

Name	df
Number of rows	29744
Number of columns	14
Column type frequency:	
Date	1
difftime	1
factor	10
logical	1
numeric	1

Group variables	None

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
OCCUR_DATE	0	1	2020-01-01	2020-12-31	2020-07-13	366

Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
OCCUR_TIME	0	1	0 secs	86340 secs	54900 secs	1424

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
BORO	0	1.00	FALSE	5	BRO: 11685, BRO: 8834, QUE: 4426, MAN: 3977
LOC_OF_OCCUR_DESC	25596	0.14	FALSE	2	OUT: 3466, INS: 682
LOC_CLASSFCTN_DESC	25596	0.14	FALSE	10	STR: 2639, HOU: 643, DWE: 341, COM: 276
LOCATION_DESC	14977	0.50	FALSE	40	MUL: 5188, MUL: 3042, (nu: 2526, PVT: 1010
PERP_AGE_GROUP	9344	0.69	FALSE	12	18-: 6630, 25-: 6342, UNK: 3148, <18: 1805
PERP_SEX	9310	0.69	FALSE	4	M: 16845, (nu: 1628, U: 1500, F: 461
PERP_RACE	9310	0.69	FALSE	8	BLA: 12323, WHI: 2667, UNK: 1838, (nu: 1628
VIC_AGE_GROUP	0	1.00	FALSE	7	25-: 13563, 18-: 10677, <18: 3081, 45-: 2118
VIC_SEX	0	1.00	FALSE	3	M: 26841, F: 2891, U: 12
VIC_RACE	0	1.00	FALSE	7	BLA: 20999, WHI: 4511, BLA: 2930, WHI: 741

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
STATISTICAL_MURDER_FLAG	0	1	0.19	FAL: 23979, TRU: 5765

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
INCIDENT_KEY	0	1	133850951	82786370	9953245	67321141	109291972	214741917	299462478	

From this, we can see the columns OCCUR_DATE, OCCUR_TIME, BORO, VIC_AGE_GROUP, VIC_SEX, VIC_RACE, STATISTICAL_MURDER_FLAG, and INCIDENT_KEY all have no missing values, and thus we do not need to perform any NULL handling.

Next, we will examine each column that does have missing values and come up with a strategy on how to handle them.

```
df %>%
  group_by(PERP_RACE) %>%
  summarise(count = n())
```

```
## # A tibble: 9 x 2
##   PERP_RACE                      count
##   <fct>                          <int>
## 1 (null)                          1628
## 2 AMERICAN INDIAN/ALASKAN NATIVE     2
## 3 ASIAN / PACIFIC ISLANDER         184
## 4 BLACK                          12323
## 5 BLACK HISPANIC                  1487
## 6 UNKNOWN                         1838
## 7 WHITE                            305
## 8 WHITE HISPANIC                  2667
## 9 <NA>                            9310
```

```
df <- df %>%
  mutate(PERP_RACE = case_when(
    is.na(PERP_RACE) ~ "UNKNOWN",
    PERP_RACE == "(null)" ~ "UNKNOWN",
    TRUE ~ PERP_RACE
  ))

df %>%
  group_by(PERP_RACE) %>%
  summarise(count = n())
```

```
## # A tibble: 7 x 2
##   PERP_RACE                      count
##   <chr>                          <int>
## 1 AMERICAN INDIAN/ALASKAN NATIVE     2
## 2 ASIAN / PACIFIC ISLANDER         184
## 3 BLACK                          12323
## 4 BLACK HISPANIC                  1487
## 5 UNKNOWN                        12776
## 6 WHITE                            305
## 7 WHITE HISPANIC                  2667
```

For the column PERP_RACE, we see that there were three different values all meaning "UNKNOWN": 1) (null), 2) NA, and 3) UNKNOWN. So we replaced all values as (null) or NA as UNKNOWN.

```
df %>%
  group_by(PERP_SEX) %>%
  summarise(count = n())
```

```
## # A tibble: 5 x 2
##   PERP_SEX count
##   <fct>    <int>
```

```
## 1 (null)     1628
## 2 F           461
## 3 M         16845
## 4 U          1500
## 5 <NA>       9310
```

```
df <- df %>%
  mutate(PERP_SEX = case_when(
    is.na(PERP_SEX) ~ "U",
    PERP_SEX == "(null)" ~ "U",
    TRUE ~ PERP_SEX
  ))

df %>%
  group_by(PERP_SEX) %>%
  summarise(count = n())
```

```
## # A tibble: 3 x 2
##   PERP_SEX count
##   <chr>    <int>
## 1 F          461
## 2 M        16845
## 3 U        12438
```

For the column PERP_SEX, we see that there were three different values all meaning "UNKNOWN": 1) (null), 2) NA, and 3) U. So we replaced all values as (null) or NA as U.

```
df %>%
  group_by(PERP_AGE_GROUP) %>%
  summarise(count = n())
```

```
## # A tibble: 13 x 2
##    PERP_AGE_GROUP count
##    <fct>          <int>
##  1 (null)          1628
##  2 <18             1805
##  3 1020               1
##  4 1028               1
##  5 18-24           6630
##  6 2021               1
##  7 224                1
##  8 25-44           6342
##  9 45-64            775
## 10 65+               67
## 11 940                1
## 12 UNKNOWN         3148
## 13 <NA>            9344
```

```
df <- df %>%
  mutate(PERP_AGE_GROUP = case_when(
    is.na(PERP_AGE_GROUP) ~ "UNKNOWN",
    PERP_AGE_GROUP == "1020" ~ "UNKNOWN",
```

```
    PERP_AGE_GROUP == "1022" ~ "UNKNOWN",
    PERP_AGE_GROUP == "1028" ~ "UNKNOWN",
    PERP_AGE_GROUP == "2021" ~ "UNKNOWN",
    PERP_AGE_GROUP == "224" ~ "UNKNOWN",
    PERP_AGE_GROUP == "940" ~ "UNKNOWN",
    PERP_AGE_GROUP == "(null)" ~ "UNKNOWN",
    TRUE ~ PERP_AGE_GROUP
  ))

df %>%
  group_by(PERP_AGE_GROUP) %>%
  summarise(count = n())
```

```
## # A tibble: 6 x 2
##   PERP_AGE_GROUP count
##   <chr>          <int>
## 1 18-24           6630
## 2 25-44           6342
## 3 45-64            775
## 4 65+               67
## 5 <18             1805
## 6 UNKNOWN        14125
```

For the column PERP_AGE_GROUP, we see that there were three different values all meaning "UNKNOWN": 1) (null), 2) NA, and 3) UNKNOWN. So wereplaced all values as (null) or NA as UNKNOWN. Additionally, there were a number of errant values such as 1020, 1028, 2021, 224, and 940. We will also change these to be UNKNOWN.

```
print(df %>%
  group_by(LOCATION_DESC) %>%
  summarise(count = n()),
  n=41)
```

```
## # A tibble: 41 x 2
##    LOCATION_DESC            count
##    <fct>                    <int>
##  1 (null)                    2526
##  2 ATM                          1
##  3 BANK                         3
##  4 BAR/NIGHT CLUB             695
##  5 BEAUTY/NAIL SALON          120
##  6 CANDY STORE                 10
##  7 CHAIN STORE                  9
##  8 CHECK CASH                   1
##  9 CLOTHING BOUTIQUE           14
## 10 COMMERCIAL BLDG            306
## 11 DEPT STORE                   9
## 12 DOCTOR/DENTIST               1
## 13 DRUG STORE                  14
## 14 DRY CLEANER/LAUNDRY         32
## 15 FACTORY/WAREHOUSE            8
## 16 FAST FOOD                  131
## 17 GAS STATION                 76
```

```
## 18 GROCERY/BODEGA            775
## 19 GYM/FITNESS FACILITY        4
## 20 HOSPITAL                   84
## 21 HOTEL/MOTEL                38
## 22 JEWELRY STORE              14
## 23 LIQUOR STORE               42
## 24 LOAN COMPANY                1
## 25 MULTI DWELL - APT BUILD  3042
## 26 MULTI DWELL - PUBLIC HOUS 5188
## 27 NONE                      175
## 28 PHOTO/COPY STORE            2
## 29 PVT HOUSE                1010
## 30 RESTAURANT/DINER          216
## 31 SCHOOL                      1
## 32 SHOE STORE                 10
## 33 SMALL MERCHANT             46
## 34 SOCIAL CLUB/POLICY LOCATI  74
## 35 STORAGE FACILITY            1
## 36 STORE UNCLASSIFIED         37
## 37 SUPERMARKET                21
## 38 TELECOMM. STORE            11
## 39 VARIETY STORE              11
## 40 VIDEO STORE                 8
## 41 <NA>                    14977
```

```
df <- df %>%
  mutate(LOCATION_DESC = case_when(
    is.na(LOCATION_DESC) ~ "UNKNOWN",
    LOCATION_DESC == "(null)" ~ "UNKNOWN",
    LOCATION_DESC == "NONE" ~ "UNKNOWN",
    TRUE ~ LOCATION_DESC
  ))

df %>%
  group_by(LOCATION_DESC) %>%
  summarise(count = n())
```

```
## # A tibble: 39 x 2
##    LOCATION_DESC     count
##    <chr>             <int>
##  1 ATM                   1
##  2 BANK                  3
##  3 BAR/NIGHT CLUB      695
##  4 BEAUTY/NAIL SALON   120
##  5 CANDY STORE          10
##  6 CHAIN STORE           9
##  7 CHECK CASH            1
##  8 CLOTHING BOUTIQUE    14
##  9 COMMERCIAL BLDG     306
## 10 DEPT STORE            9
## # i 29 more rows
```

For the column LOCATION_DESC, we see that there were three different values all meaning "UNKNOWN": 1) (null), 2) NA, and 3) NONE. So we replaced all values as (null), NA, or NONE as UNKNOWN.

```r
print(df %>%
  group_by(LOC_CLASSFCTN_DESC) %>%
  summarise(count = n()),
  n=11)
```

```
## # A tibble: 11 x 2
##    LOC_CLASSFCTN_DESC count
##    <fct>              <int>
##  1 (null)                 7
##  2 COMMERCIAL           276
##  3 DWELLING             341
##  4 HOUSING              643
##  5 OTHER                 74
##  6 PARKING LOT           16
##  7 PLAYGROUND            67
##  8 STREET              2639
##  9 TRANSIT               52
## 10 VEHICLE               33
## 11 <NA>               25596
```

```r
df <- df %>%
  mutate(LOC_CLASSFCTN_DESC = case_when(
    is.na(LOC_CLASSFCTN_DESC) ~ "UNKNOWN",
    LOC_CLASSFCTN_DESC == "(null)" ~ "UNKNOWN",
    TRUE ~ LOC_CLASSFCTN_DESC
  ))

df %>%
  group_by(LOC_CLASSFCTN_DESC) %>%
  summarise(count = n())
```

```
## # A tibble: 10 x 2
##    LOC_CLASSFCTN_DESC count
##    <chr>              <int>
##  1 COMMERCIAL           276
##  2 DWELLING             341
##  3 HOUSING              643
##  4 OTHER                 74
##  5 PARKING LOT           16
##  6 PLAYGROUND            67
##  7 STREET              2639
##  8 TRANSIT               52
##  9 UNKNOWN            25603
## 10 VEHICLE               33
```

For the column LOC_CLASSFCTN_DESC, we see that there were two different values all meaning "UNKNOWN": 1) (null), and 2) NA. So we replaced all values as (null) or NA as UNKNOWN.

```r
df %>% group_by(LOC_OF_OCCUR_DESC) %>%
        summarise(count = n())
```

```
## # A tibble: 3 x 2
```

```
##   LOC_OF_OCCUR_DESC count
##   <fct>             <int>
## 1 INSIDE              682
## 2 OUTSIDE            3466
## 3 <NA>              25596
```

```
df <- df %>%
  mutate(LOC_OF_OCCUR_DESC = case_when(
    is.na(LOC_OF_OCCUR_DESC) ~ "UNKNOWN",
    TRUE ~ LOC_OF_OCCUR_DESC
  ))

df %>%
  group_by(LOC_OF_OCCUR_DESC) %>%
  summarise(count = n())
```

```
## # A tibble: 3 x 2
##   LOC_OF_OCCUR_DESC count
##   <chr>             <int>
## 1 INSIDE              682
## 2 OUTSIDE            3466
## 3 UNKNOWN           25596
```

For the column LOC_OF_OCCUR_DESC, we see that the rows listed as NA were meant to be UNKNOWN. So we replaced all NA values as UNKNOWN.

Now we will view the cleaned dataset one last time to ensure we caught everything.

```
skim(df)
```

Table 7: Data summary

Name	df
Number of rows	29744
Number of columns	14
Column type frequency:	
character	6
Date	1
difftime	1
factor	4
logical	1
numeric	1
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
LOC_OF_OCCUR_DESC	0	1	6	7	0	3	0
LOC_CLASSFCTN_DESC	0	1	5	11	0	10	0

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
LOCATION_DESC	0	1	3	25	0	39	0
PERP_AGE_GROUP	0	1	3	7	0	6	0
PERP_SEX	0	1	1	1	0	3	0
PERP_RACE	0	1	5	30	0	7	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
OCCUR_DATE	0	1	2020-01-01	2020-12-31	2020-07-13	366

Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
OCCUR_TIME	0	1	0 secs	86340 secs	54900 secs	1424

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
BORO	0	1	FALSE	5	BRO: 11685, BRO: 8834, QUE: 4426, MAN: 3977
VIC_AGE_GROUP	0	1	FALSE	7	25-: 13563, 18-: 10677, <18: 3081, 45-: 2118
VIC_SEX	0	1	FALSE	3	M: 26841, F: 2891, U: 12
VIC_RACE	0	1	FALSE	7	BLA: 20999, WHI: 4511, BLA: 2930, WHI: 741

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
STATISTICAL_MURDER_FLAG	0	1	0.19	FAL: 23979, TRU: 5765

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
INCIDENT_KEY	0	1	133850951	82786370	9953245	67321141	109291972	214741917	299462478	

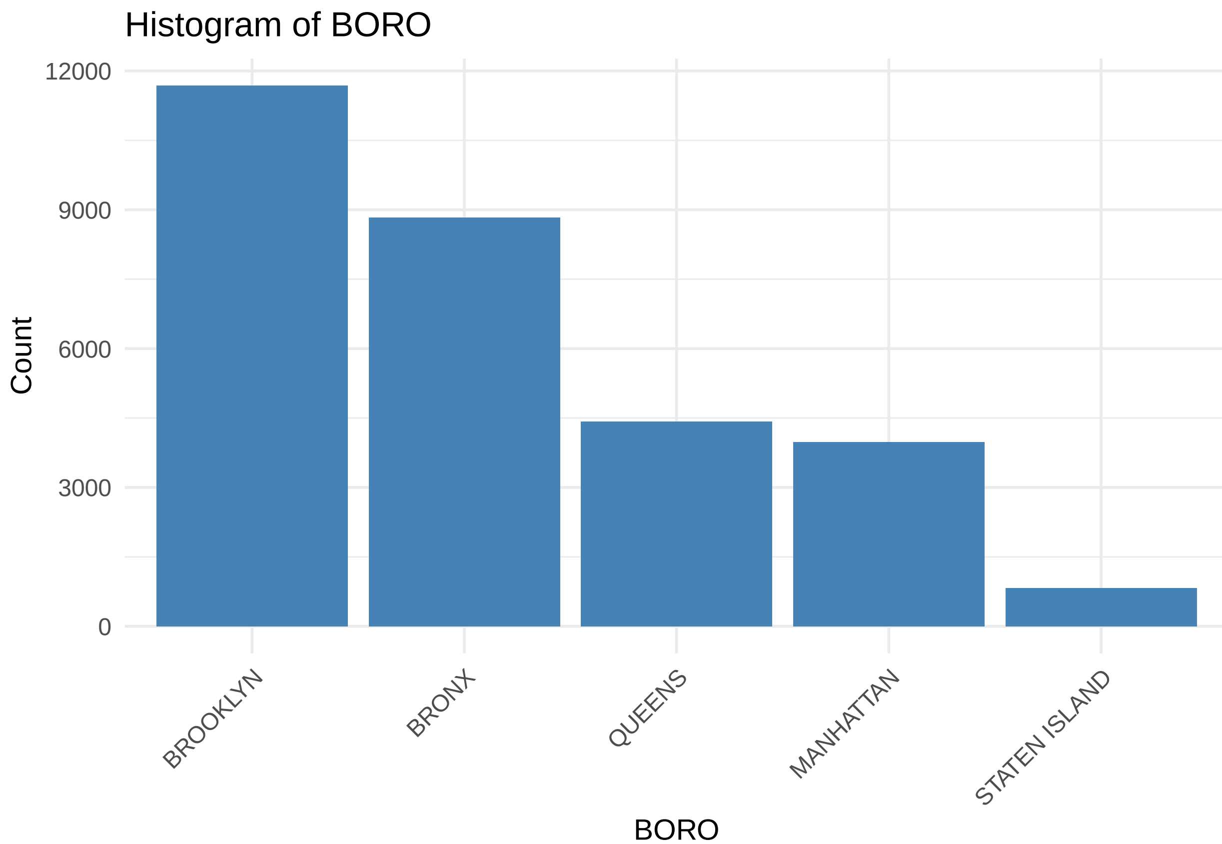
Note that some columns got changed back to character data types, so we will transform factor columns back into factors.

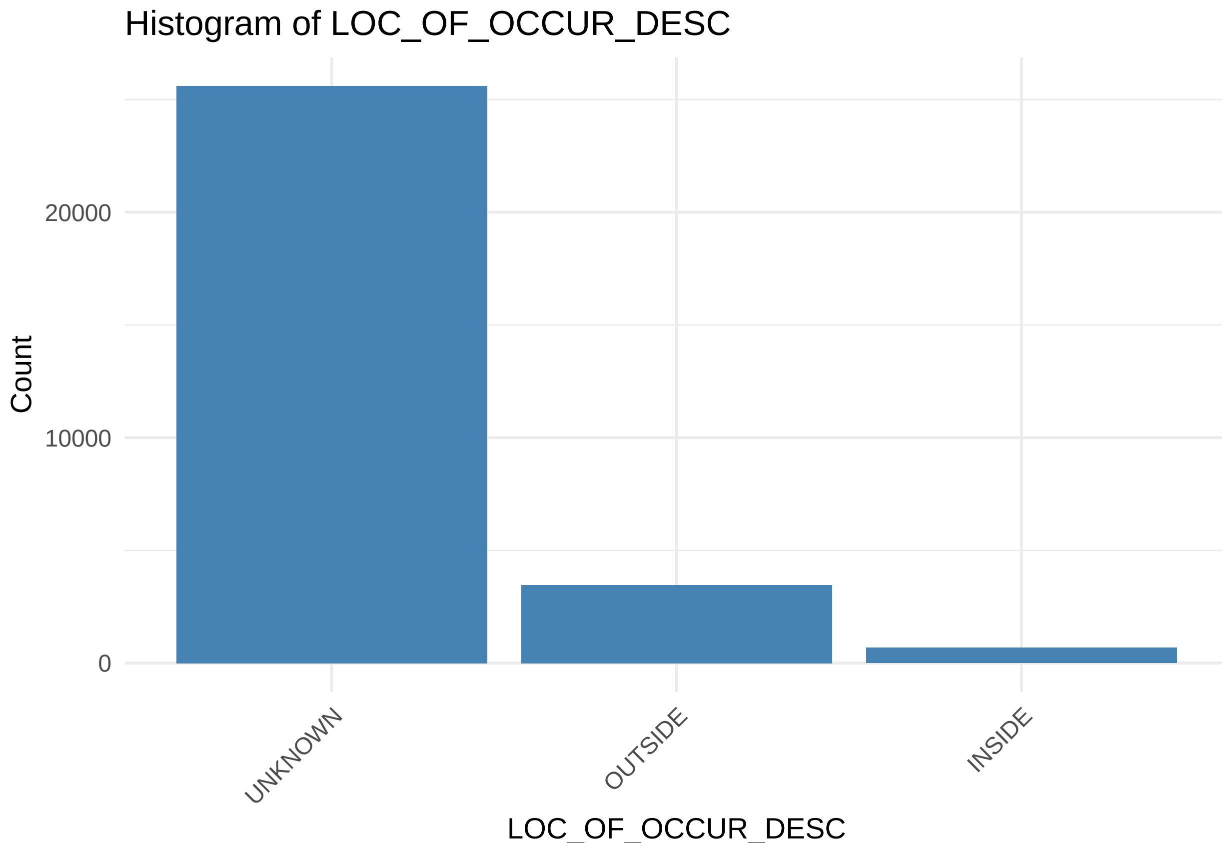
```r
# Change columns which should be factors into factors
df <- df %>%
  mutate(across(c(BORO,
                  LOC_OF_OCCUR_DESC,
                  LOC_CLASSFCTN_DESC,
```

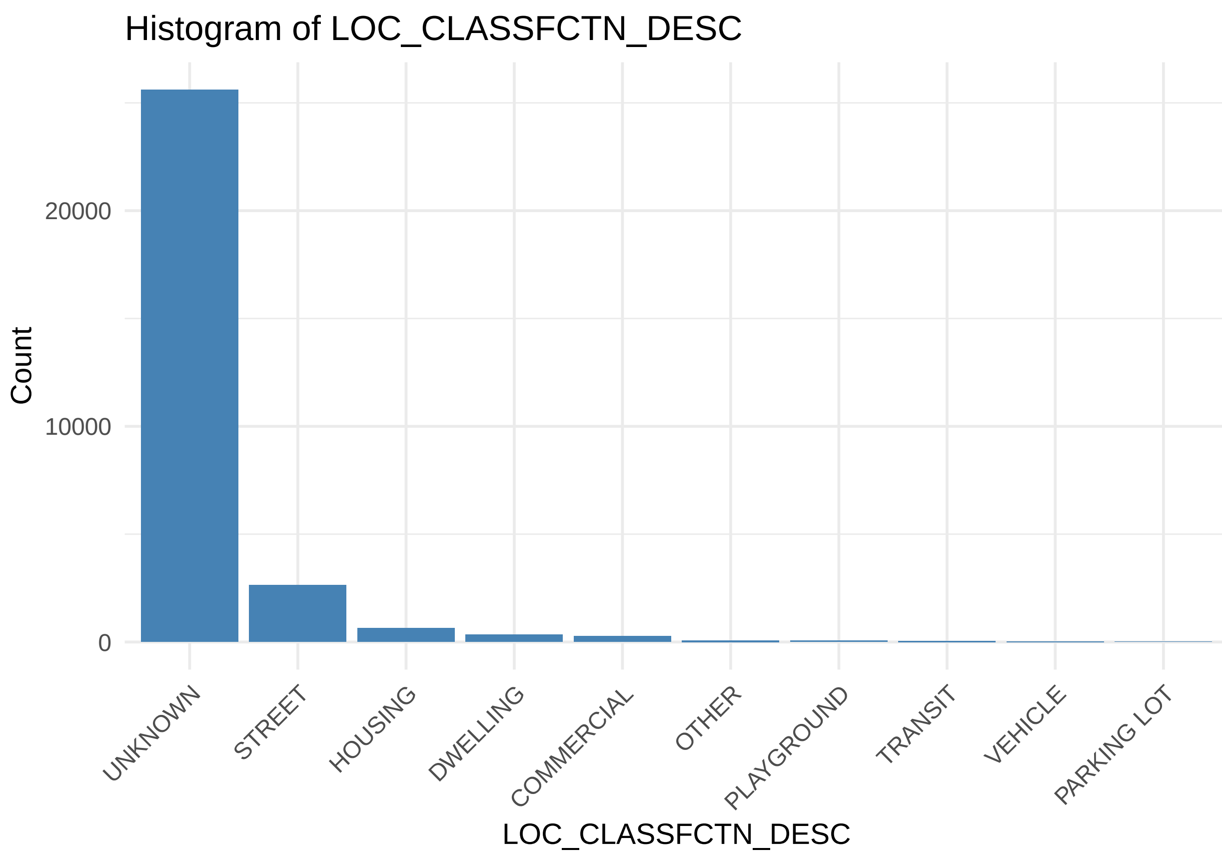
```
                  LOCATION_DESC,
                  PERP_AGE_GROUP,
                  PERP_SEX,
                  PERP_RACE,
                  VIC_AGE_GROUP,
                  VIC_SEX,
                  VIC_RACE), as.factor))
```

Exploratory Data Analysis

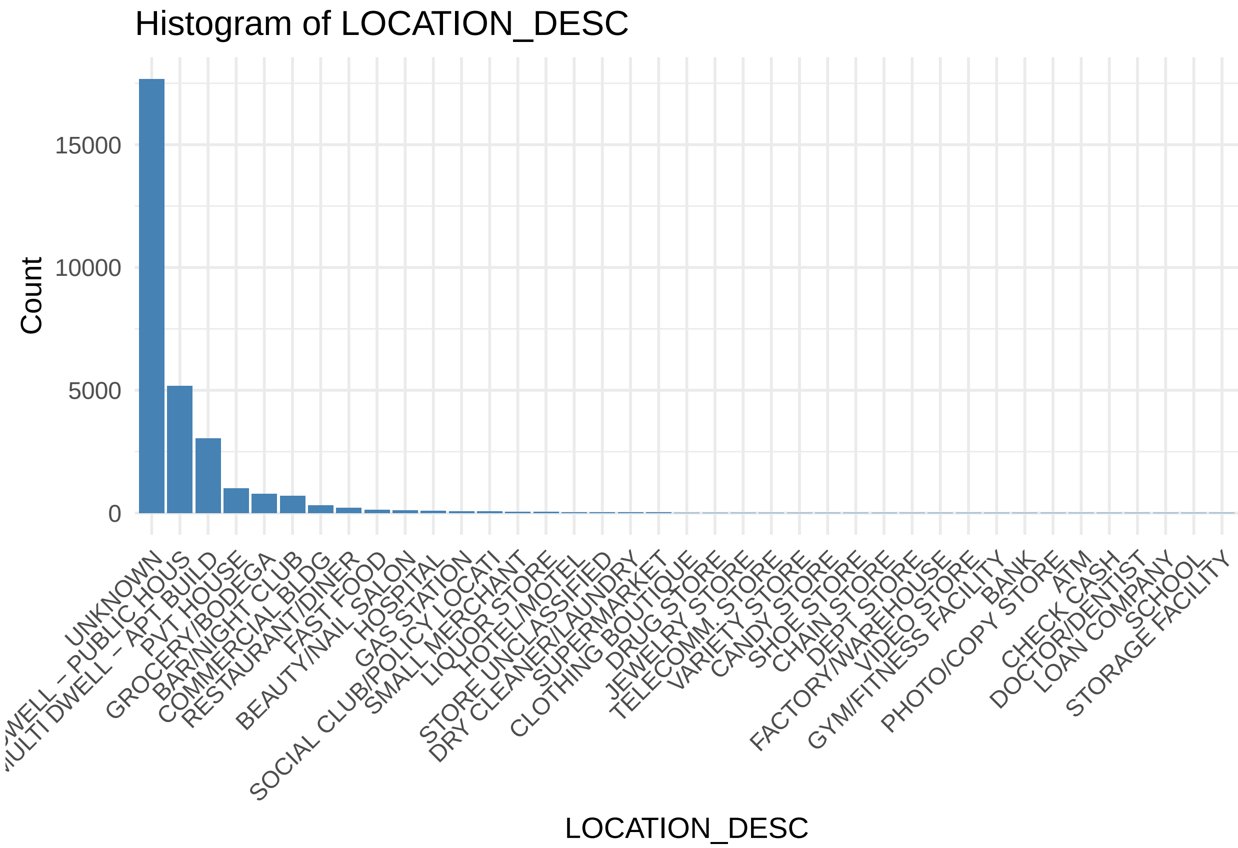
Now that the data is tidy, we will begin exploring and understanding the data.



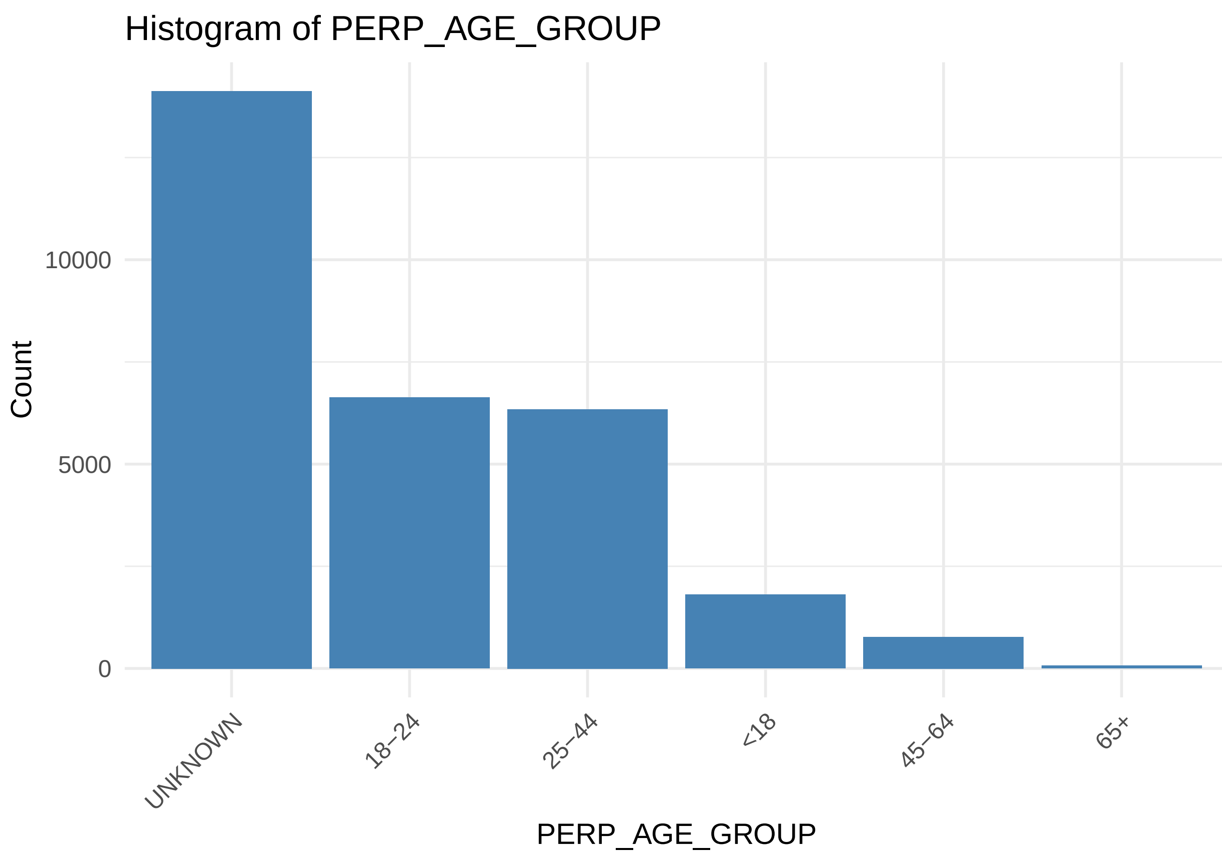

Histogram of LOC_OF_OCCUR_DESC
Count
20000
10000
0
UNKNOWN
OUTSIDE
INSIDE
LOC_OF_OCCUR_DESC

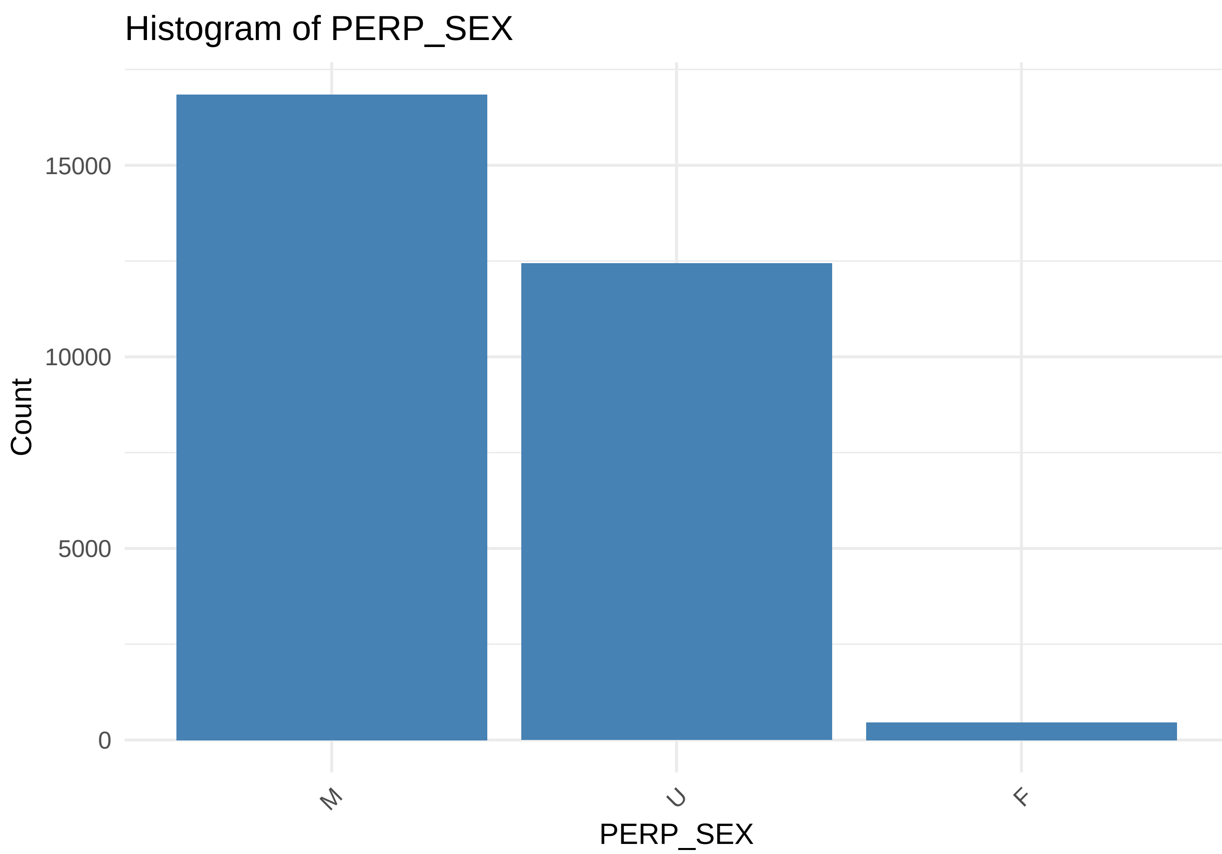


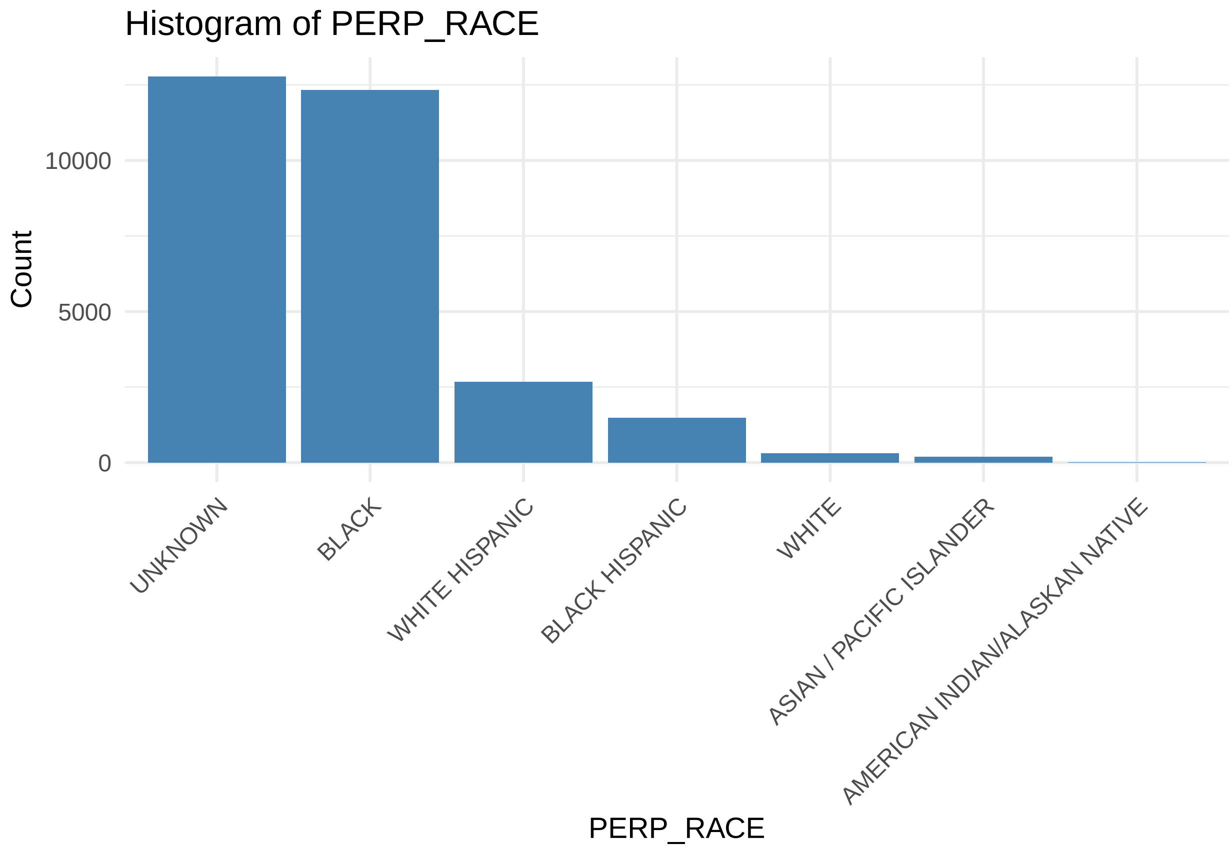
Histogram of LOC_CLASSFCTN_DESC
Count
20000
10000
0
UNKNOWN
STREET
HOUSING
DWELLING
COMMERCIAL
OTHER
PLAYGROUND
TRANSIT
VEHICLE
PARKING LOT
LOC_CLASSFCTN_DESC

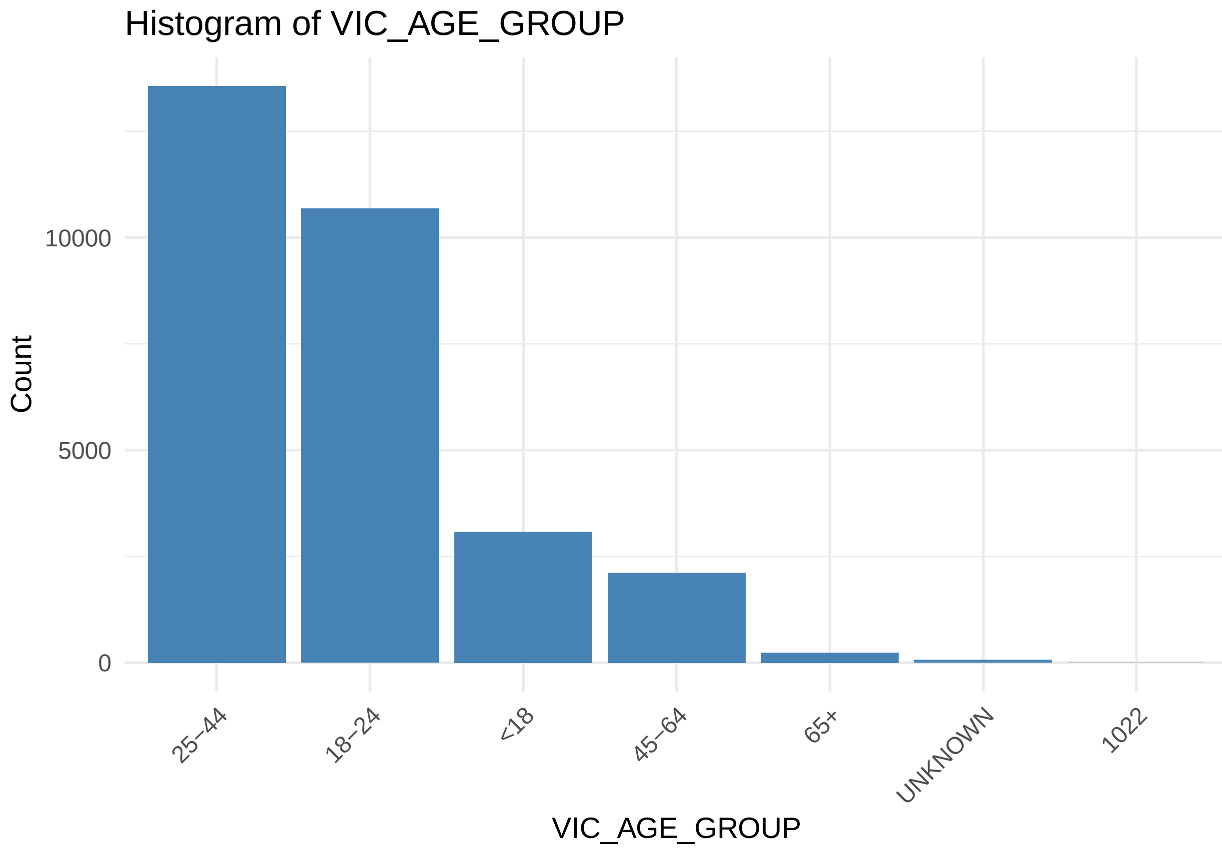


Histogram of LOCATION_DESC
Count
15000
10000
5000
0
UNKNOWN
MULTI DWELL – PUBLIC HOUS
MULTI DWELL – APT BUILD
PVT HOUSE
GROCERY/BODEGA
BAR/NIGHT CLUB
COMMERCIAL BLDG
RESTAURANT/DINER
FAST FOOD
BEAUTY/NAIL SALON
HOSPITAL
GAS STATION
SOCIAL CLUB/POLICY LOCATI
SMALL MERCHANT
LIQUOR STORE
HOTEL/MOTEL
STORE UNCLASSIFIED
DRY CLEANER/LAUNDRY
SUPERMARKET
CLOTHING BOUTIQUE
DRUG STORE
JEWELRY STORE
TELECOMM. STORE
VARIETY STORE
CANDY STORE
SHOE STORE
CHAIN STORE
DEPT STORE
FACTORY/WAREHOUSE
VIDEO STORE
GYM/FITNESS FACILITY
BANK
PHOTO/COPY STORE
ATM
CHECK CASH
DOCTOR/DENTIST
LOAN COMPANY
SCHOOL
STORAGE FACILITY
LOCATION_DESC

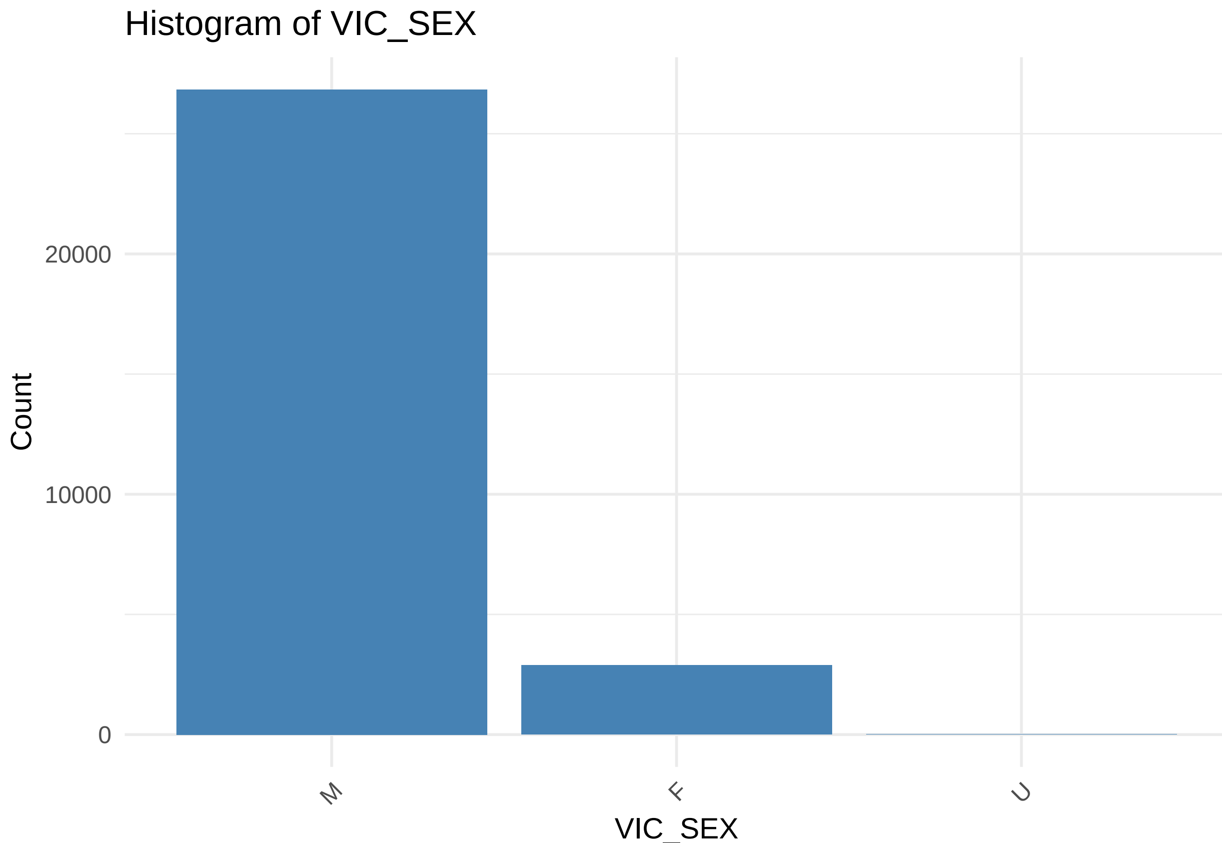

Histogram of PERP_AGE_GROUP
Count
10000
5000
0
UNKNOWN
18–24
25–44
<18
45–64
65+
PERP_AGE_GROUP


Histogram of PERP_SEX
Count
15000
10000
5000
0
M
U
F
PERP_SEX

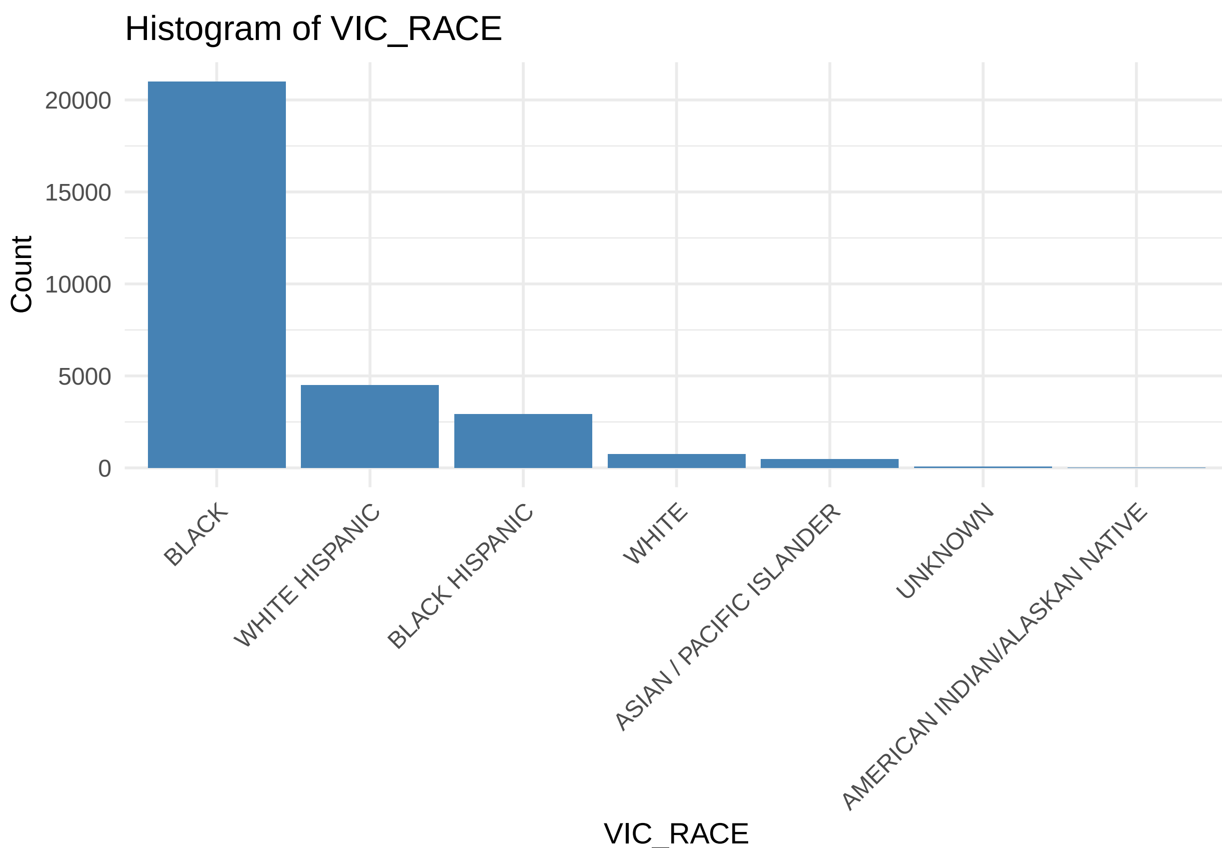

Histogram of PERP_RACE
Count
10000
5000
0
UNKNOWN
BLACK
WHITE HISPANIC
BLACK HISPANIC
WHITE
ASIAN / PACIFIC ISLANDER
AMERICAN INDIAN/ALASKAN NATIVE
PERP_RACE

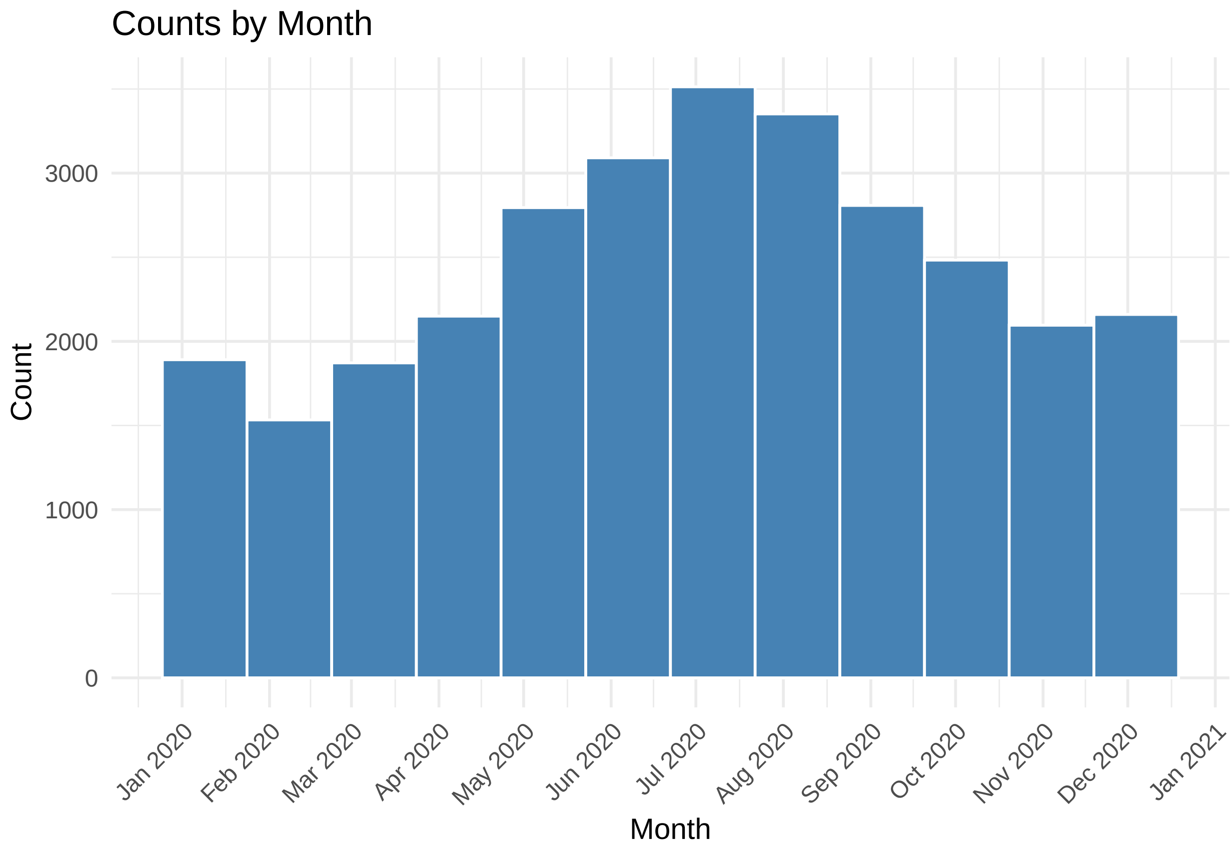


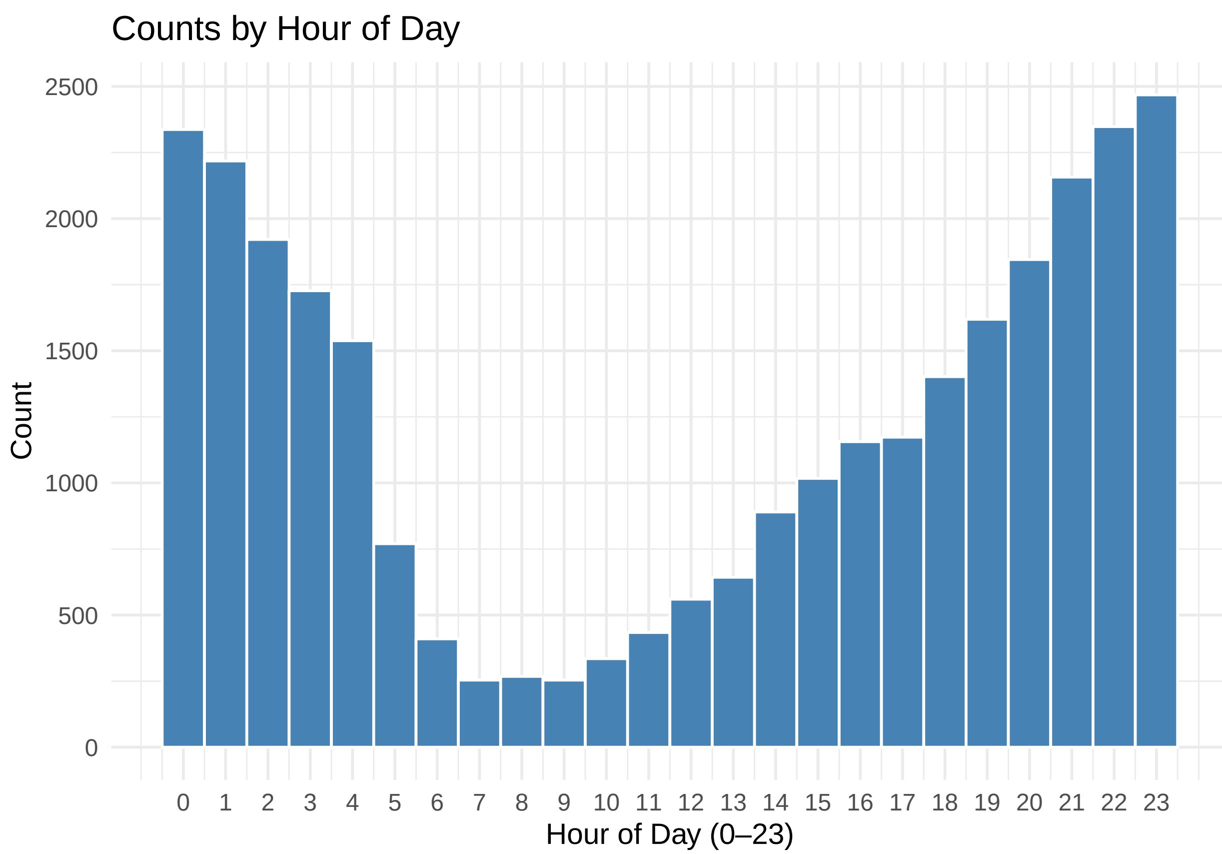
Histogram of VIC_AGE_GROUP
Count
10000
5000
0
25–44
18–24
<18
45–64
65+
UNKNOWN
1022
VIC_AGE_GROUP

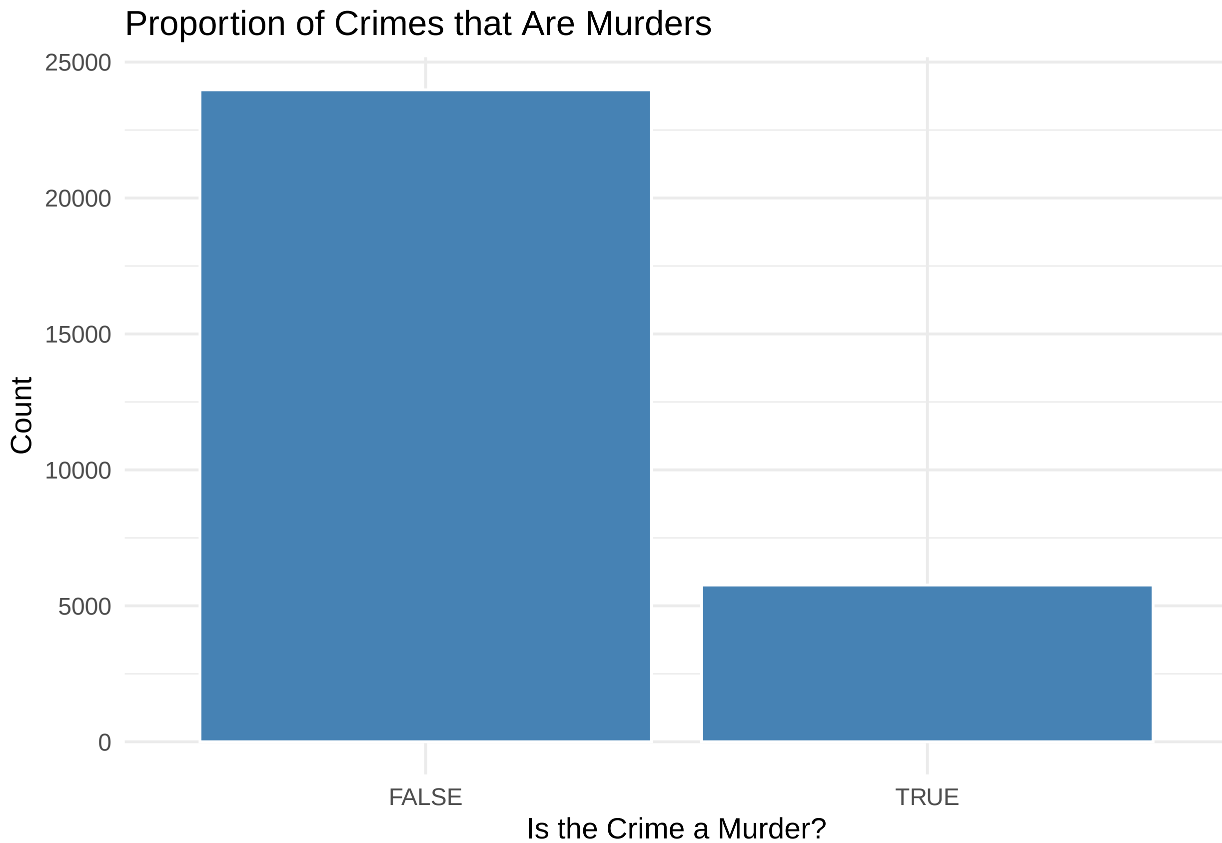


Histogram of VIC_SEX
Count
20000
10000
0
M
F
U
VIC_SEX


Histogram of VIC_RACE
Count
20000
15000
10000
5000
0
BLACK
WHITE HISPANIC
BLACK HISPANIC
WHITE
ASIAN / PACIFIC ISLANDER
UNKNOWN
AMERICAN INDIAN/ALASKAN NATIVE
VIC_RACE


Counts by Month
Count
3000
2000
1000
0
Jan 2020
Feb 2020
Mar 2020
Apr 2020
May 2020
Jun 2020
Jul 2020
Aug 2020
Sep 2020
Oct 2020
Nov 2020
Dec 2020
Jan 2021
Month


Counts by Hour of Day
2500
2000
1500
1000
500
0
Count
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Hour of Day (0–23)



From these graphs, we can begin gathering some initial questions for us to further research:

1. Why are Brooklyn and the Bronx the top two areas in regards to the number of crimes committed?
2. What factors influence geographic increases in crime?
3. In what ways can we use the information about geographic crime rates to try and prevent crime in ways that do not further historical injustices?
4. Why do crime rates go up in the summer months?
5. Why is 12AM-1AM so much higher than all other hour slots? Is this a data entry bug or accurate information?

Analysis

With the data now understood, we can begin analyzing it. In this analysis, we will run a logistic regression on the STATISTICAL_MURDER_FLAG field to see what variables are the strongest predictor of if a crime will be a murder so that we may better plan to eliminate these cases.

First, we will need to remove any rows where the variables are "unknown" so we do not muddy the data. We will shrink the dataset into a "golden" subset containing only rows with all the needed information and then run a logistic regression on it.

```
# Remove rows where any column has the value "UNKNOWN"
df_clean <- df %>%
  mutate(across(c(BORO,
                  LOC_OF_OCCUR_DESC,
                  LOC_CLASSFCTN_DESC,
                  PERP_AGE_GROUP,
```

```r
                  PERP_SEX,
                  PERP_RACE,
                  LOCATION_DESC,
                  VIC_AGE_GROUP,
                  VIC_SEX,
                  VIC_RACE), as.character))
df_clean <- df_clean %>%
  filter(!if_any(c(BORO,
                   LOC_OF_OCCUR_DESC,
                   LOC_CLASSFCTN_DESC,
                   PERP_AGE_GROUP,
                   PERP_SEX,
                   PERP_RACE,
                   LOCATION_DESC,
                   VIC_AGE_GROUP,
                   VIC_SEX,
                   VIC_RACE), ~ .x == "UNKNOWN"))

# Change columns which should be factors into factors
df_clean <- df_clean %>%
  mutate(across(c(BORO,
                  LOC_OF_OCCUR_DESC,
                  LOC_CLASSFCTN_DESC,
                  PERP_AGE_GROUP,
                  PERP_SEX,
                  PERP_RACE,
                  LOCATION_DESC,
                  VIC_AGE_GROUP,
                  VIC_SEX,
                  VIC_RACE), as.factor))

# Change columns which should be dates into dates
df_clean <- df_clean %>%
  mutate(across(c(OCCUR_DATE), ~ as.Date(., format = "%m/%d/%y")))

# Change columns which should be boolean into boolean
df_clean <- df_clean %>%
  mutate(across(c(STATISTICAL_MURDER_FLAG), as.logical))

# Perform Logistic regression on STATISTICAL_MURDER_FLAG field
model <- glm(STATISTICAL_MURDER_FLAG ~ ., data = df_clean, family = binomial)
summary(model)
```

```
##
## Call:
## glm(formula = STATISTICAL_MURDER_FLAG ~ ., family = binomial,
##     data = df_clean)
##
## Coefficients:
##                                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)                             1.312e+01  2.400e+03   0.005  0.99564
## INCIDENT_KEY                            5.304e-09  4.966e-09   1.068  0.28546
## OCCUR_DATE                              4.283e-03  9.272e-03   0.462  0.64414
```

```
## OCCUR_TIME                               1.350e-04  8.030e-05   1.681  0.09279
## BOROBROOKLYN                            -2.339e-01  2.206e-01  -1.060  0.28895
## BOROMANHATTAN                           -3.844e-02  2.426e-01  -0.158  0.87407
## BOROQUEENS                              -4.770e-01  2.820e-01  -1.692  0.09071
## BOROSTATEN ISLAND                       -4.501e-01  5.119e-01  -0.879  0.37926
## LOC_OF_OCCUR_DESCOUTSIDE                -6.273e-01  2.177e-01  -2.882  0.00395
## LOC_CLASSFCTN_DESCDWELLING               1.150e+00  4.140e-01   2.777  0.00548
## LOC_CLASSFCTN_DESCHOUSING                1.166e+00  1.254e+00   0.930  0.35225
## LOC_CLASSFCTN_DESCOTHER                 -1.300e+01  7.757e+02  -0.017  0.98663
## LOC_CLASSFCTN_DESCPARKING LOT           -1.193e-01  1.307e+00  -0.091  0.92730
## LOC_CLASSFCTN_DESCSTREET                 8.634e-01  3.444e-01   2.507  0.01217
## LOCATION_DESCBEAUTY/NAIL SALON           1.208e+00  8.037e-01   1.503  0.13293
## LOCATION_DESCCANDY STORE                -1.561e+01  1.356e+03  -0.012  0.99082
## LOCATION_DESCCHAIN STORE                -1.477e+01  1.697e+03  -0.009  0.99305
## LOCATION_DESCCOMMERCIAL BLDG             5.238e-01  6.584e-01   0.796  0.42622
## LOCATION_DESCDRUG STORE                  1.746e+01  1.385e+03   0.013  0.98994
## LOCATION_DESCFACTORY/WAREHOUSE           1.923e+01  1.697e+03   0.011  0.99096
## LOCATION_DESCFAST FOOD                   2.349e-02  6.124e-01   0.038  0.96940
## LOCATION_DESCGAS STATION                -1.167e+00  1.124e+00  -1.038  0.29916
## LOCATION_DESCGROCERY/BODEGA              4.651e-01  4.410e-01   1.055  0.29160
## LOCATION_DESCGYM/FITNESS FACILITY       -2.072e+00  2.522e+03  -0.001  0.99934
## LOCATION_DESCHOSPITAL                   -1.411e+01  7.817e+02  -0.018  0.98560
## LOCATION_DESCHOTEL/MOTEL                -6.967e-01  1.235e+00  -0.564  0.57260
## LOCATION_DESCJEWELRY STORE              -1.598e+01  2.400e+03  -0.007  0.99469
## LOCATION_DESCLIQUOR STORE                1.773e+00  9.962e-01   1.780  0.07515
## LOCATION_DESCMULTI DWELL - APT BUILD    -4.066e-01  4.880e-01  -0.833  0.40481
## LOCATION_DESCMULTI DWELL - PUBLIC HOUS -9.858e-01  1.281e+00  -0.770  0.44142
## LOCATION_DESCPVT HOUSE                  -1.307e-01  5.277e-01  -0.248  0.80439
## LOCATION_DESCRESTAURANT/DINER            4.504e-01  6.895e-01   0.653  0.51360
## LOCATION_DESCSHOE STORE                 -1.559e+01  2.400e+03  -0.006  0.99482
## LOCATION_DESCSMALL MERCHANT              8.321e-01  7.752e-01   1.073  0.28313
## LOCATION_DESCSOCIAL CLUB/POLICY LOCATI -1.543e+01  1.693e+03  -0.009  0.99273
## LOCATION_DESCSTORE UNCLASSIFIED          1.431e+00  1.585e+00   0.903  0.36664
## LOCATION_DESCSUPERMARKET                -1.566e+01  1.694e+03  -0.009  0.99262
## LOCATION_DESCTELECOMM. STORE             2.436e+00  9.994e-01   2.437  0.01480
## LOCATION_DESCVIDEO STORE                 1.803e+01  9.760e+02   0.018  0.98526
## PERP_AGE_GROUP18-24                     -2.140e-01  3.051e-01  -0.702  0.48299
## PERP_AGE_GROUP25-44                     -1.776e-01  3.038e-01  -0.584  0.55895
## PERP_AGE_GROUP45-64                      5.316e-01  3.688e-01   1.441  0.14949
## PERP_AGE_GROUP65+                        1.903e-01  1.342e+00   0.142  0.88722
## PERP_SEXM                                1.065e-01  4.241e-01   0.251  0.80171
## PERP_RACEBLACK                           9.685e-01  7.176e-01   1.350  0.17713
## PERP_RACEBLACK HISPANIC                  1.099e+00  7.587e-01   1.449  0.14746
## PERP_RACEWHITE                           1.185e+00  1.003e+00   1.181  0.23759
## PERP_RACEWHITE HISPANIC                  1.194e+00  7.354e-01   1.624  0.10442
## VIC_AGE_GROUP18-24                      -3.806e-02  3.336e-01  -0.114  0.90916
## VIC_AGE_GROUP25-44                       2.431e-01  3.167e-01   0.767  0.44281
## VIC_AGE_GROUP45-64                       5.267e-01  3.639e-01   1.447  0.14786
## VIC_AGE_GROUP65+                         5.881e-01  6.504e-01   0.904  0.36585
## VIC_SEXM                                -3.426e-02  2.232e-01  -0.154  0.87799
## VIC_RACEASIAN / PACIFIC ISLANDER         1.693e+01  2.400e+03   0.007  0.99437
## VIC_RACEBLACK                            1.614e+01  2.400e+03   0.007  0.99463
## VIC_RACEBLACK HISPANIC                   1.576e+01  2.400e+03   0.007  0.99476
## VIC_RACEWHITE                            1.631e+01  2.400e+03   0.007  0.99458
```

```
## VIC_RACEWHITE HISPANIC                  1.596e+01  2.400e+03   0.007  0.99469
## month                                  -6.083e-03  9.318e-03  -0.653  0.51389
## hour                                   -4.880e-01  2.905e-01  -1.680  0.09295
##
## (Intercept)
## INCIDENT_KEY
## OCCUR_DATE
## OCCUR_TIME                             .
## BOROBROOKLYN
## BOROMANHATTAN
## BOROQUEENS                             .
## BOROSTATEN ISLAND
## LOC_OF_OCCUR_DESCOUTSIDE               **
## LOC_CLASSFCTN_DESCDWELLING             **
## LOC_CLASSFCTN_DESCHOUSING
## LOC_CLASSFCTN_DESCOTHER
## LOC_CLASSFCTN_DESCPARKING LOT
## LOC_CLASSFCTN_DESCSTREET               *
## LOCATION_DESCBEAUTY/NAIL SALON
## LOCATION_DESCCANDY STORE
## LOCATION_DESCCHAIN STORE
## LOCATION_DESCCOMMERCIAL BLDG
## LOCATION_DESCDRUG STORE
## LOCATION_DESCFACTORY/WAREHOUSE
## LOCATION_DESCFAST FOOD
## LOCATION_DESCGAS STATION
## LOCATION_DESCGROCERY/BODEGA
## LOCATION_DESCGYM/FITNESS FACILITY
## LOCATION_DESCHOSPITAL
## LOCATION_DESCHOTEL/MOTEL
## LOCATION_DESCJEWELRY STORE
## LOCATION_DESCLIQUOR STORE              .
## LOCATION_DESCMULTI DWELL - APT BUILD
## LOCATION_DESCMULTI DWELL - PUBLIC HOUS
## LOCATION_DESCPVT HOUSE
## LOCATION_DESCRESTAURANT/DINER
## LOCATION_DESCSHOE STORE
## LOCATION_DESCSMALL MERCHANT
## LOCATION_DESCSOCIAL CLUB/POLICY LOCATI
## LOCATION_DESCSTORE UNCLASSIFIED
## LOCATION_DESCSUPERMARKET
## LOCATION_DESCTELECOMM. STORE           *
## LOCATION_DESCVIDEO STORE
## PERP_AGE_GROUP18-24
## PERP_AGE_GROUP25-44
## PERP_AGE_GROUP45-64
## PERP_AGE_GROUP65+
## PERP_SEXM
## PERP_RACEBLACK
## PERP_RACEBLACK HISPANIC
## PERP_RACEWHITE
## PERP_RACEWHITE HISPANIC
## VIC_AGE_GROUP18-24
## VIC_AGE_GROUP25-44
```

```
## VIC_AGE_GROUP45-64
## VIC_AGE_GROUP65+
## VIC_SEXM
## VIC_RACEASIAN / PACIFIC ISLANDER
## VIC_RACEBLACK
## VIC_RACEBLACK HISPANIC
## VIC_RACEWHITE
## VIC_RACEWHITE HISPANIC
## month
## hour                                   .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1159.6  on 985  degrees of freedom
## Residual deviance: 1017.9  on 926  degrees of freedom
## AIC: 1137.9
##
## Number of Fisher Scoring iterations: 15
```

The output of the logistic regression is showing the following variables are statistically significant when predicting if the crime will be a murder or not:

1. LOC_OF_OCCUR_DESCOUTSIDE has effect -0.6273 with p-value 0.00395 –> This means if a crime happens outdoors, it is less likely to be murder.
2. LOC_CLASSFCTN_DESCDWELLING has effect 1.150 with p-value 0.00548 –> This means if a crime happens at a dwelling, it is more likely to be murder.
3. LOCATION_DESCTELECOMM. STORE has effect 2.436 with p-value 0.01480 –> This means if a crime happens at a telecommunication store, it is more likely to be murder.

These results make intuitive sense, as most murders are committed inside homes. However, the third most significant variable is a bit surprising: that if a crime happens at a telecommunication store, then it is more likely to be a murder. This is especially surprising given we would expect theft to be common at these locations. This is worth looking into and validating further. We will add this to our list of further questions to investigate:

6. Is the initial observation that a crime happening at a telecommunication store meaning it is more likely to be a murder accurate? If so, why might this be?

Conclusion

Biases

As in all cases, I come into this investigation with biases. Some possible biases include:

1. My own background. I grew up in a very affirming and supportive Christian home. Because of this, I am very sheltered from many historical structures of injustice that have perpetuated inequalities. When discussing things like whether police should more heavily patrol areas with higher crime rates, I need to listen well to others who raise concerns about perpetuating cycles of crime and poverty.
2. I tend to trust law enforcement authorities. Some people in my circles are very distrusting of law enforcement because of their own negative experiences. I need to be aware of my bias in conversations where I do strongly believe law enforcement is a net good and necessary institution.

Questions for further investigation:

1. Why are Brooklyn and the Bronx the top two areas in regards to the number of crimes committed?
2. What factors influence geographic increases in crime?
3. In what ways can we use the information about geographic crime rates to try and prevent crime in ways that do not further historical injustices?
4. Why do crime rates go up in the summer months?
5. Why is 12AM-1AM so much higher than all other hour slots? Is this a data entry bug or accurate information?
6. Is the initial observation that a crime happening at a telecommunication store meaning it is more likely to be a murder accurate? If so, why might this be?

Works Cited

[1] "NYPD Shooting Incident Data (Historic)" Data Catalog, Data.gov, last updated Apirl 19, 2025, accessed May 16, 2025. Link: https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic
[2] Ibid.
[3] Workbook was created with the assistance of ChatGPT.