Analyzing NYPD Shooting Incidents

Punam Paul

2023-08-20

NYPD Shooting Incident Data (Historic)

Load the CSV data from *url* into *nypd_data* variable.

```
url <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv"
nypd_data <- read_csv(url)
```

Tidying and Transforming the Data

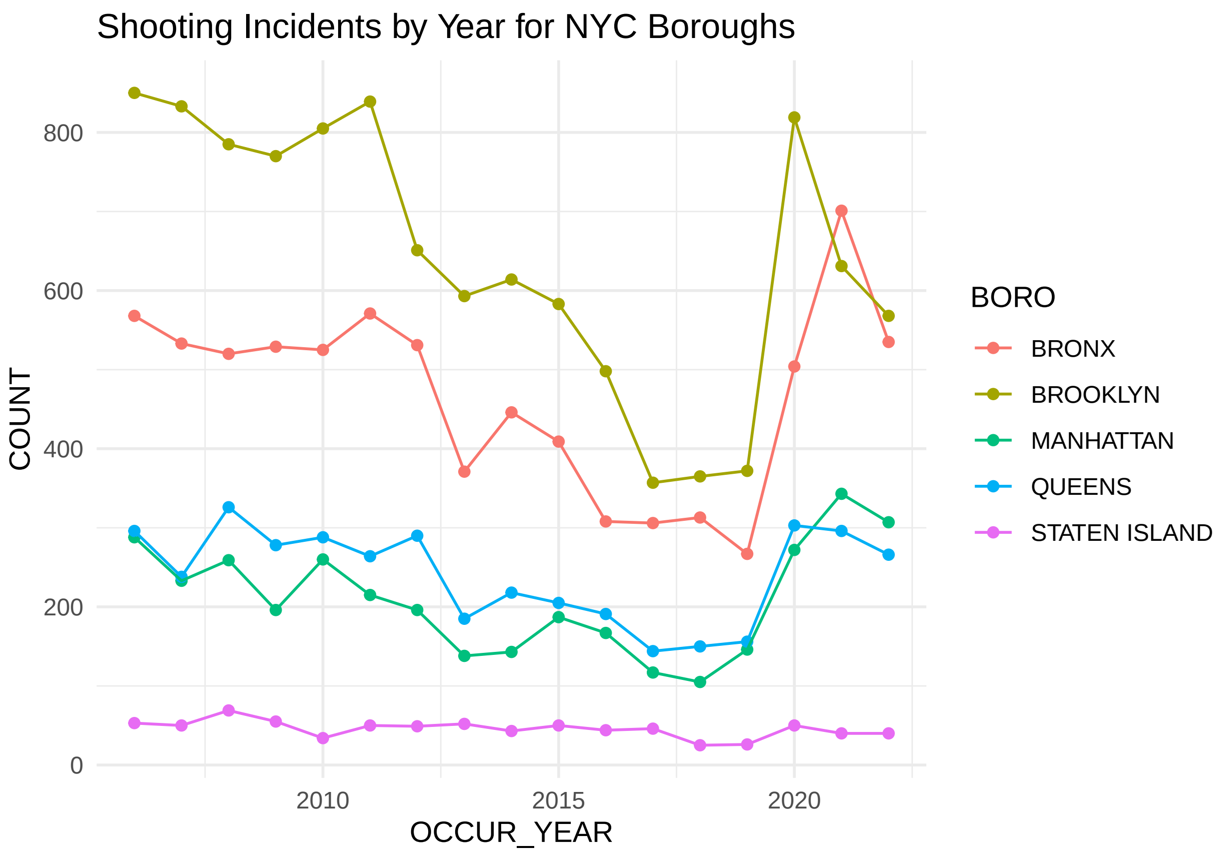
Tidy up the data. Remove x-y coordinates, latitude/longitude and convert *OCCUR_DATE* from *chr* to *date* datatype.

```
nypd_data <- nypd_data %>% select(-(X_COORD_CD:Lon_Lat)) %>% mutate(OCCUR_DATE = mdy(OCCUR_DATE))
```

Analyzing Shooting Incidents by Borough Over the Years

Extract *OCCUR_YEAR* from *OCCUR_DATE*, calculate counts of shooting incidents for each borough and year.

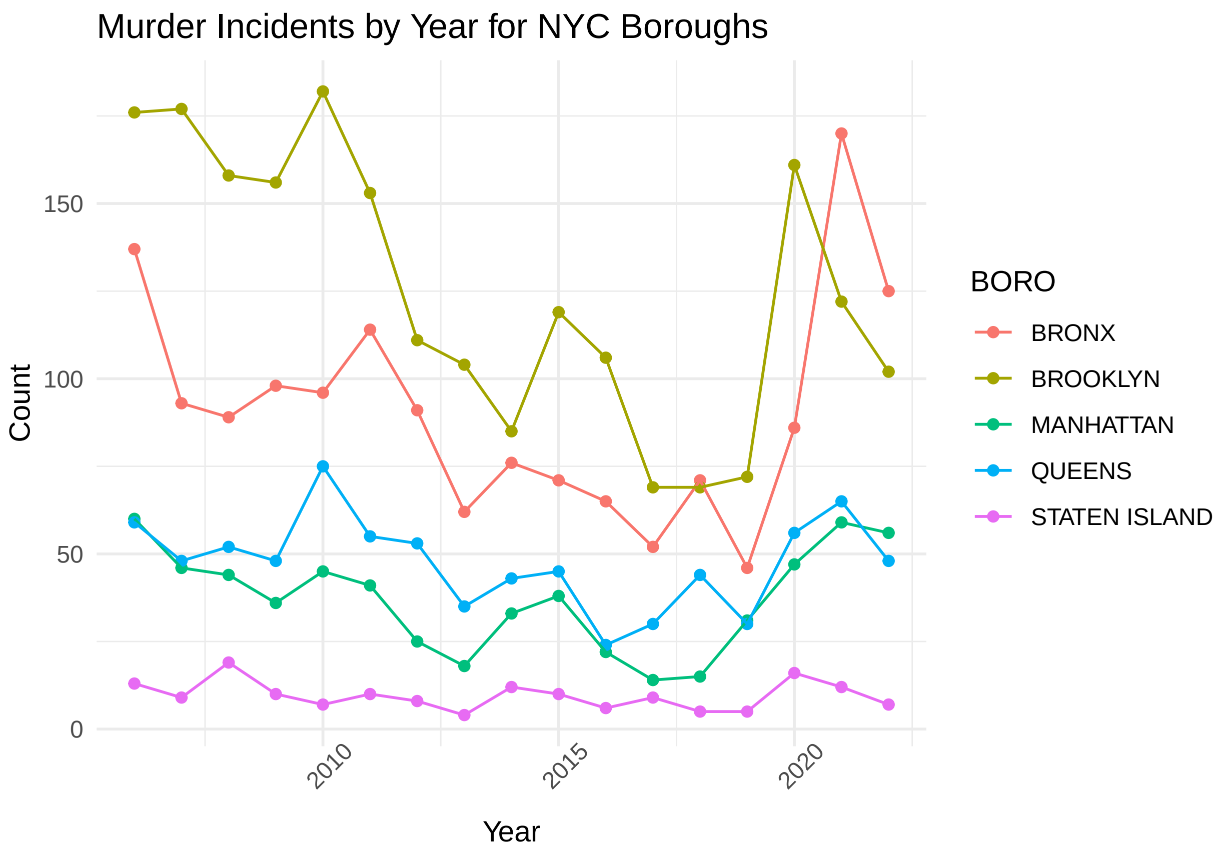
```
nypd_data_by_boro <- nypd_data %>% mutate(OCCUR_YEAR=year(OCCUR_DATE)) %>% group_by(BORO, OCCUR_YEAR) %

ggplot(nypd_data_by_boro, aes(x=OCCUR_YEAR, y=COUNT, colour=BORO)) + geom_point() + geom_line() + theme
```



Visualizing Murders for Every Borough Over the Years

Extract $OCCUR\_YEAR$ from $OCCUR\_DATE$. Then we filter incidents where $STATISTICAL\_MURDER\_FLAG$ is set to $TRUE$. Then calculate counts of murders for each borough and year.

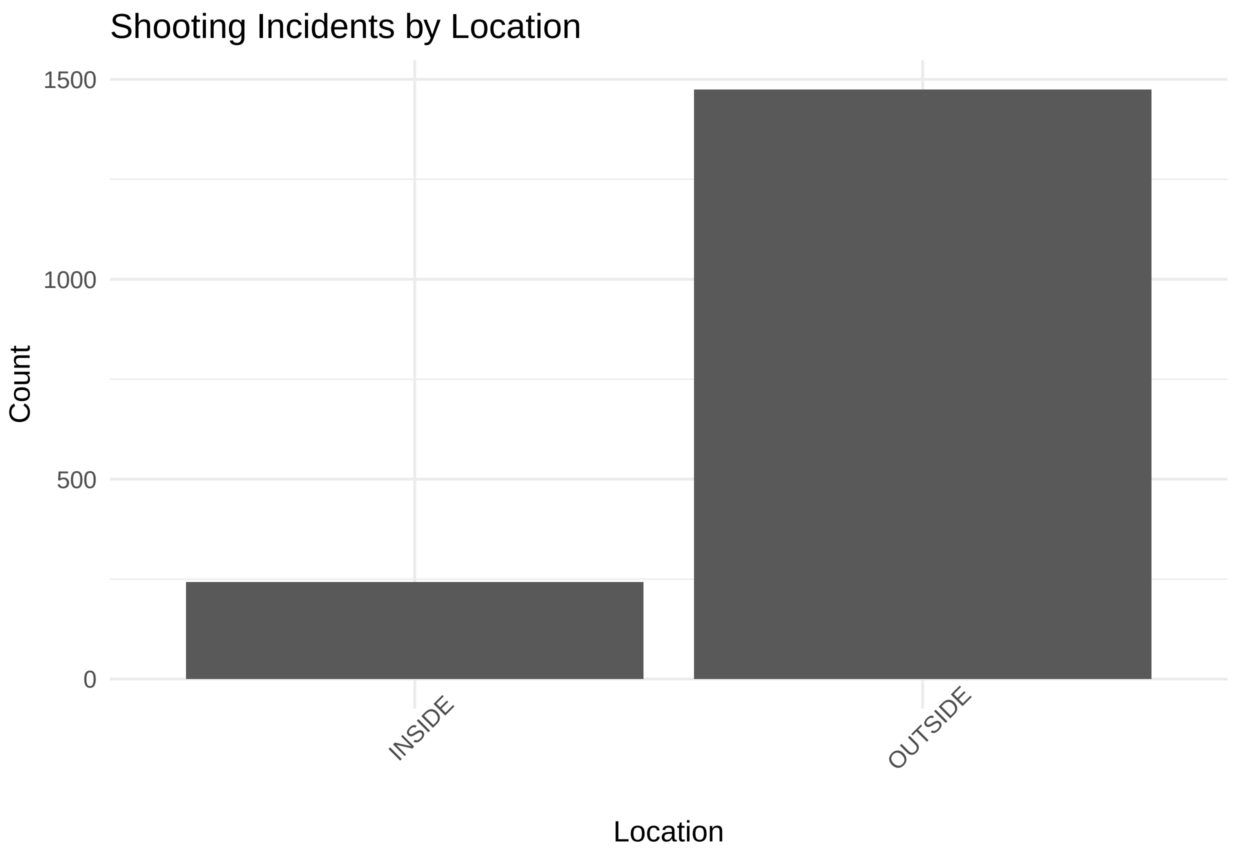
```
nypd_data_by_boro_murders <- nypd_data %>% filter(STATISTICAL_MURDER_FLAG) %>% mutate(OCCUR_YEAR=year(OC

ggplot(nypd_data_by_boro_murders, aes(x=OCCUR_YEAR, y=COUNT, colour=BORO)) + geom_point() + geom_line()
```



Visualizing Shooting Incidents by Location (Inside/Outside)

Drop NA values for $LOC\_OF\_OCCUR\_DESC$ column. Then do a bar chart on $LOC\_OF\_OCCUR\_DESC$ column.

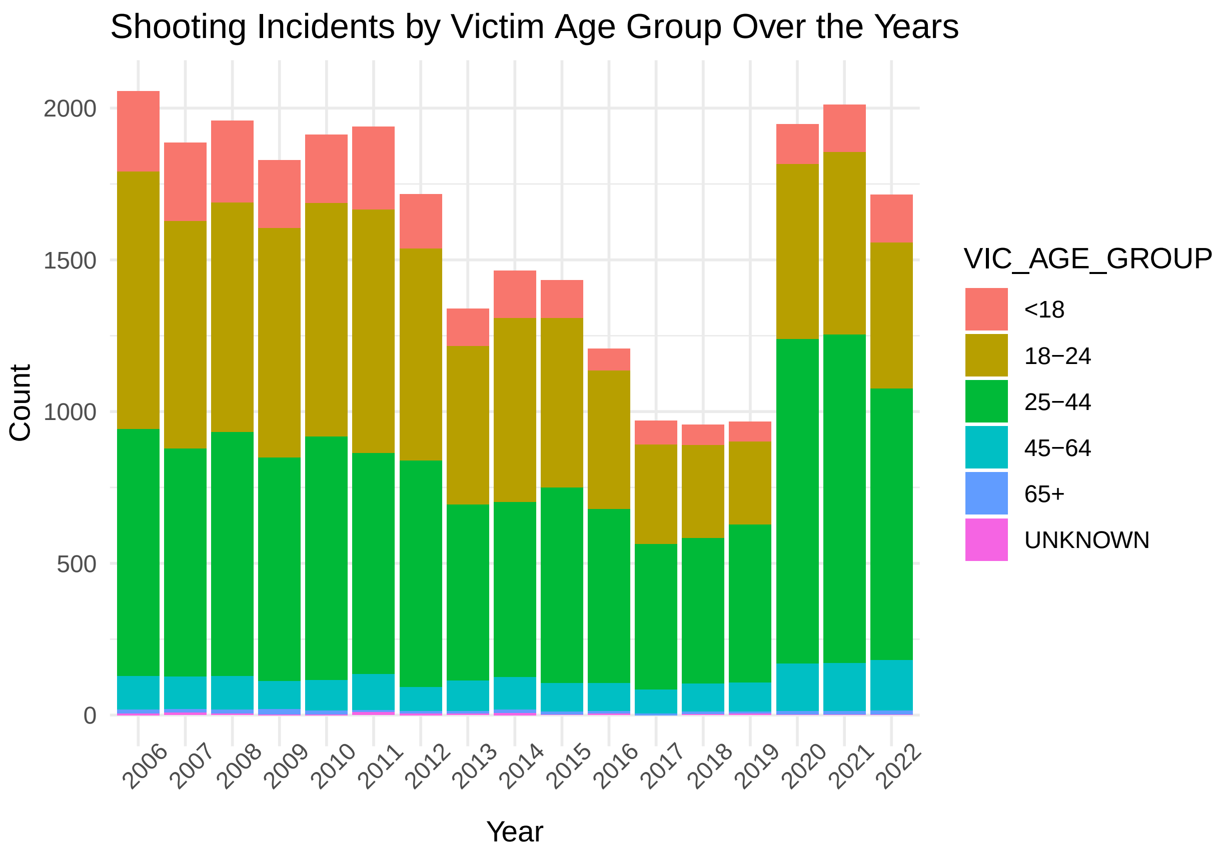
```r
nypd_data_by_location <- nypd_data %>% drop_na(LOC_OF_OCCUR_DESC)

ggplot(nypd_data_by_location, aes(LOC_OF_OCCUR_DESC)) + geom_bar() + theme_minimal() + ggtitle("Shootin
```



Visualizing Shooting Incidents by Victim Age Group

Drop NA values from *VIC_AGE_GROUP* column. Then do a bar chart on *VIC_AGE_GROUP* column. Then we filter out the invalid values for *VIC_AGE_GROUP*. Afterwards, we plot bar chart on *VIC_AGE_GROUP* over the years.

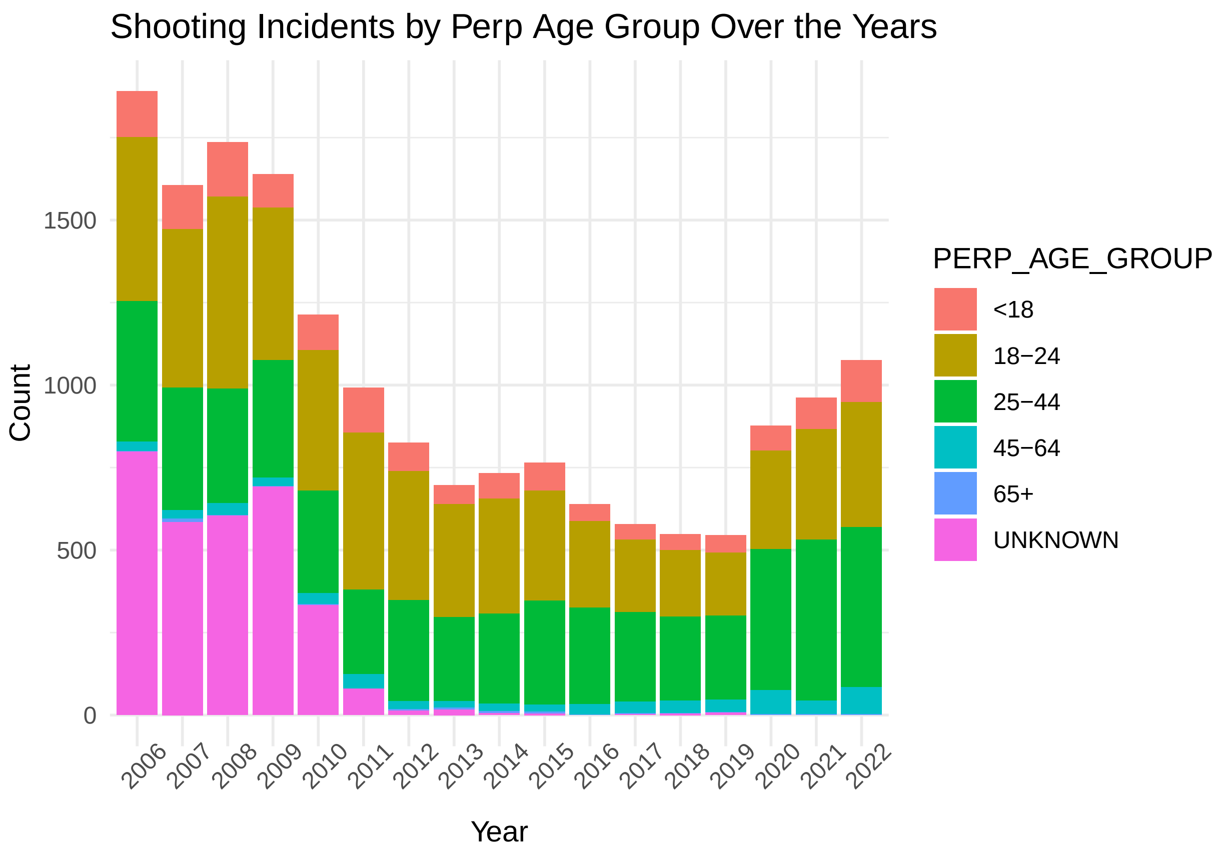
```
nypd_data_by_vic <- nypd_data %>% drop_na(VIC_AGE_GROUP) %>% filter(VIC_AGE_GROUP != 1022) %>% mutate(O

ggplot(nypd_data_by_vic, aes(x=as.factor(OCCUR_YEAR), fill=VIC_AGE_GROUP)) + geom_bar() + theme_minimal
```



Visualizing Shooting Incidents by Perp Age Group

Drop NA values from *PERP_AGE_GROUP* column. Then do a bar chart on *PERP_AGE_GROUP* column. Then we filter out the invalid values for *PERP_AGE_GROUP*. Afterwards, we plot bar chart on *PERP_AGE_GROUP* over the years.

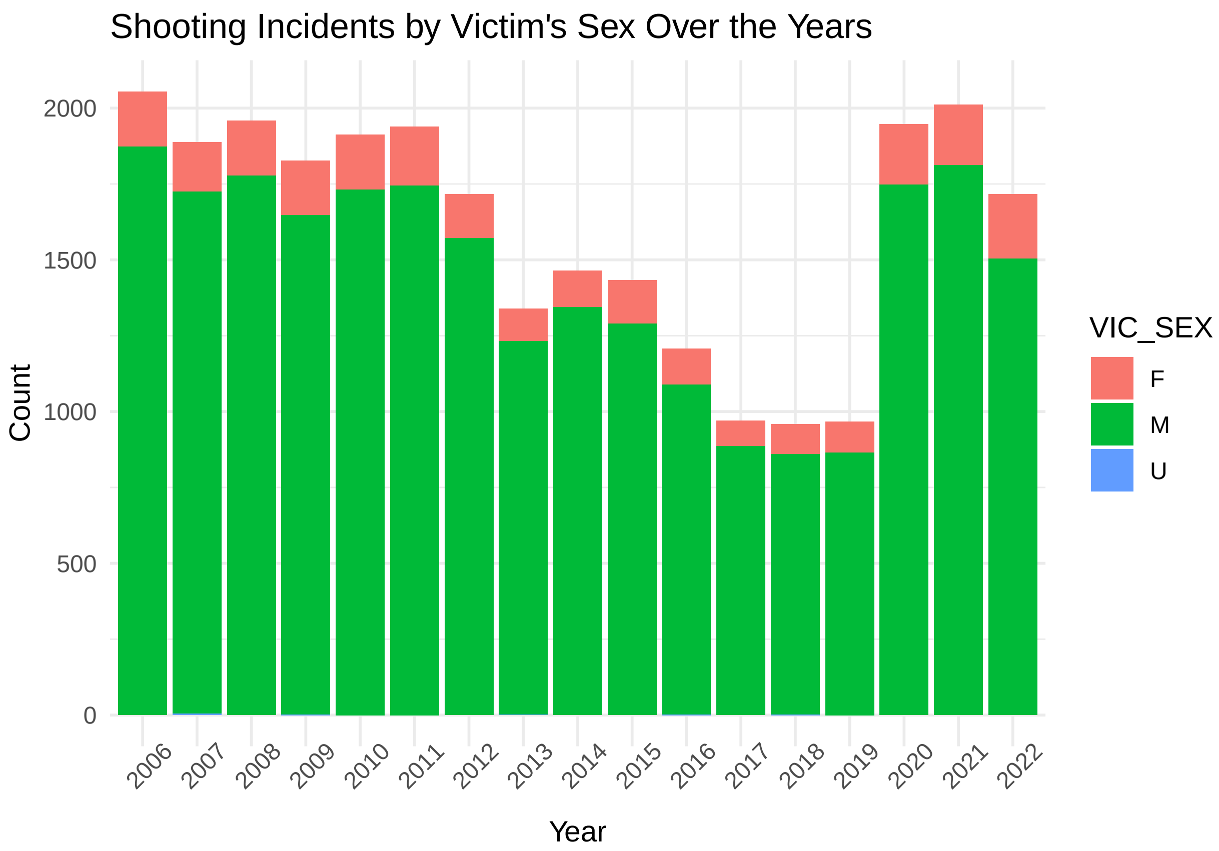
```
nypd_data_by_perp <- nypd_data %>% drop_na(PERP_AGE_GROUP) %>% filter(PERP_AGE_GROUP != 1020 & PERP_AGE_

ggplot(nypd_data_by_perp, aes(x=as.factor(OCCUR_YEAR), fill=PERP_AGE_GROUP)) + geom_bar() + theme_minima
```



Visualizing Sex of Victims

Drop NA values and plot a bar chart over the years.

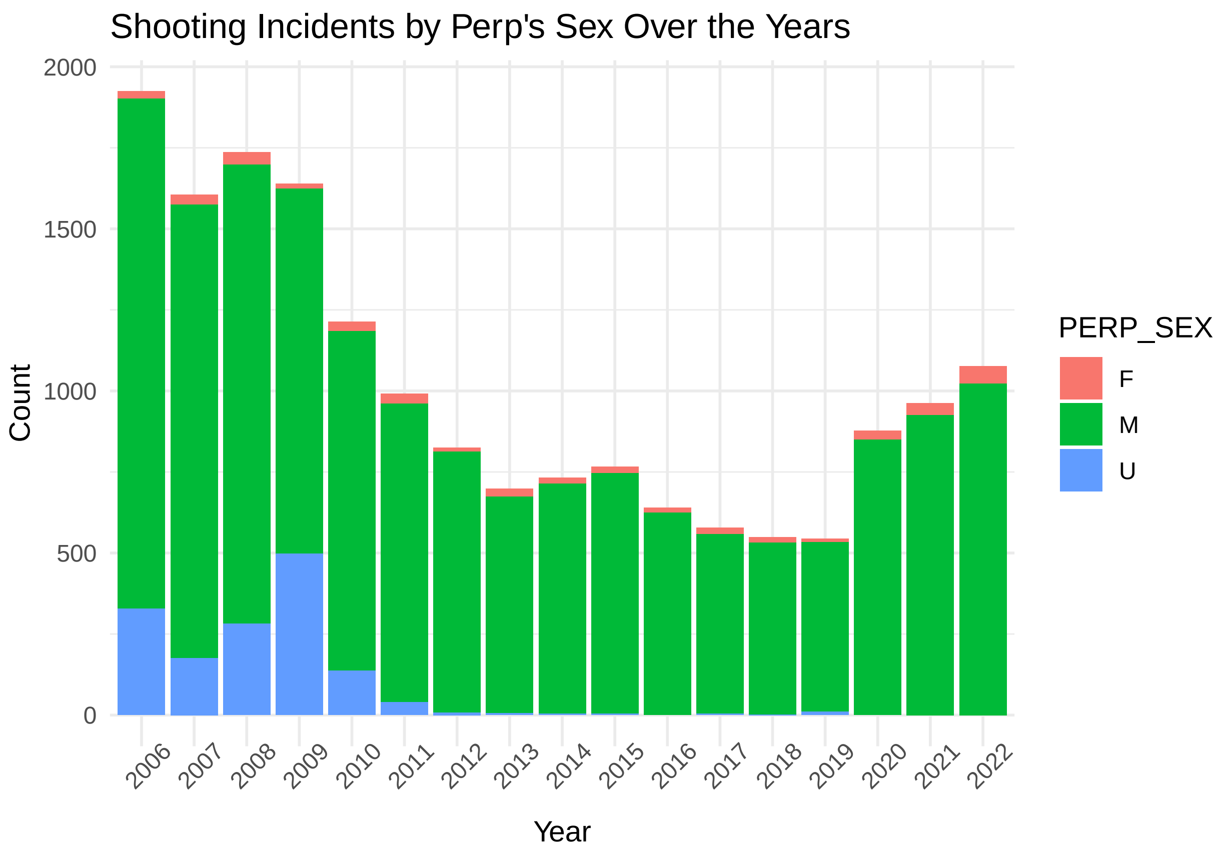
```
nypd_data_by_vic_sex <- nypd_data %>% drop_na(VIC_SEX) %>% mutate(OCCUR_YEAR=year(OCCUR_DATE))

ggplot(nypd_data_by_vic_sex, aes(x=as.factor(OCCUR_YEAR), fill=VIC_SEX)) + geom_bar() + theme_minimal()
```



Visualizing Sex of Perp

Drop NA values and plot a bar chart over the years.

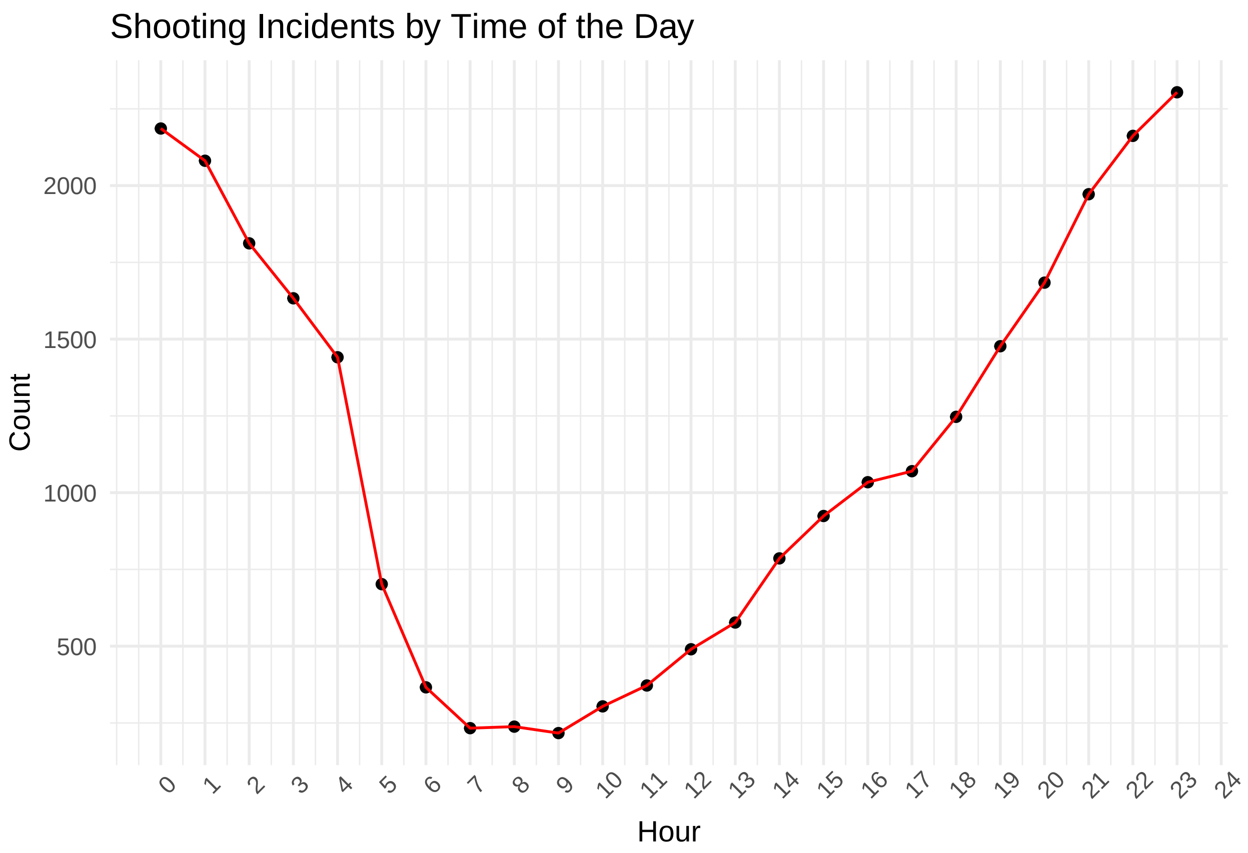
```
nypd_data_by_perp_sex <- nypd_data %>% drop_na(PERP_SEX) %>% mutate(OCCUR_YEAR=year(OCCUR_DATE)) %>% fil

ggplot(nypd_data_by_perp_sex, aes(x=as.factor(OCCUR_YEAR), fill=PERP_SEX)) + geom_bar() + theme_minimal
```



Visualizing Time of Shooting

Drop NA values from *OCCUR_TIME* column, extract the hour value and then plot a line chart.

```
nypd_data_by_time <- nypd_data %>% drop_na(OCCUR_TIME) %>% mutate(OCCUR_HOUR=hour(OCCUR_TIME)) %>% grou

ggplot(nypd_data_by_time, aes(OCCUR_HOUR, COUNT)) + geom_point() + geom_line(colour="RED") + theme_mini
```



Data Modeling: Predict Sex of the Victim Based on the Sex of the Perp

We are modeling the data using decision tree with *rpart* library. We are predicting *VIC_SEX* based on *PERP_SEX* and *VIC_AGE_GROUP* features. First, we will remove unknown and NA values from *PERP_SEX* and *VIC_SEX* columns, and drop NA values from *VIC_AGE_GROUP* column. Then we do a test-train split and create a decision tree. We predict the *VIC_SEX_PREDICTED* column for *nypd_test* dataset.

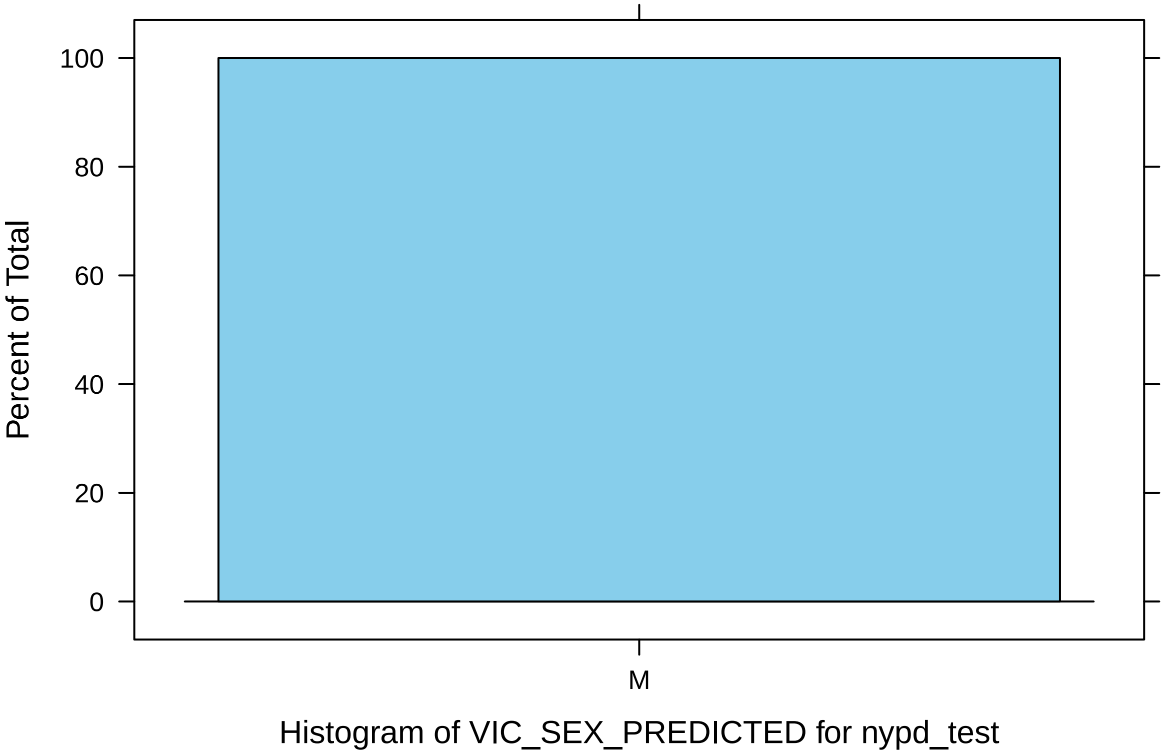
```
nypd_data_sex <- nypd_data %>% drop_na(VIC_SEX) %>% drop_na(PERP_SEX) %>% drop_na(VIC_AGE_GROUP) %>% fi

set.seed(4650)
train_index <- createDataPartition(nypd_data_sex$VIC_SEX,
                                  p = .80,
                                  list = FALSE,
                                  times = 1)

nypd_train <- nypd_data_sex[ train_index,]
nypd_test  <- nypd_data_sex[-train_index,]

mytree <- rpart(
  VIC_SEX ~ PERP_SEX + VIC_AGE_GROUP,
  data = nypd_train,
  method="class"
  )
```

```
nypd_test$VIC_SEX_PREDICTED <- predict(mytree, newdata = nypd_test, type = "class")
```

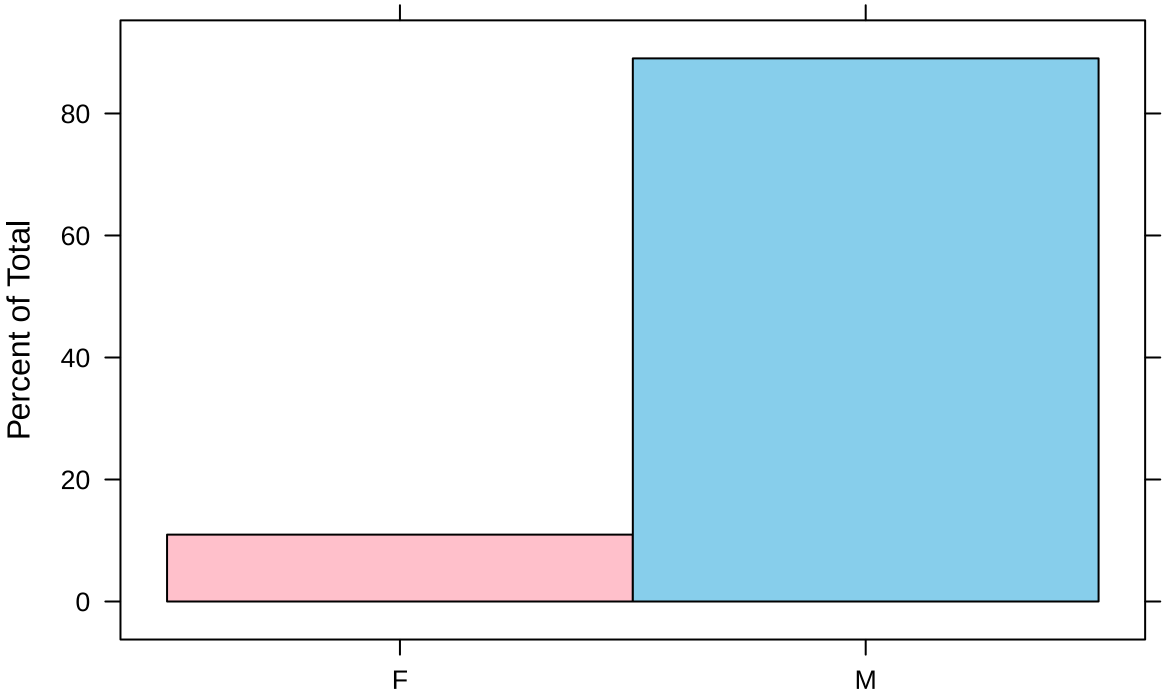
Model Bias

Based on the histogram, we can see that the model predicted all victims to be of male sex. We can see in the second histogram that in the *nypd_train* dataset, over 80% of the victims are of male sex. The decision tree model showed bias while predicting the *VIC_SEX* to show all the victims as men. We can either use more complex models or do model tuning to reduce the model bias.

```
histogram(as.factor(nypd_test$VIC_SEX_PREDICTED), col=c("skyblue"), xlab="Histogram of VIC_SEX_PREDICTEI
```



```
histogram(as.factor(nypd_train$VIC_SEX), col=c("pink", "skyblue"), xlab="Histogram of VIC_SEX for nypd_
```



Histogram of VIC_SEX for nypd_train