Shootings in NYC

3/30/2023

Abstract

This report aims to study the change in relationships and trends of shooting incidents in NYC over time.

Is there a relationship between time of day and shooting frequency? Has that relationship changed over time? Does that relationship change based on distance?

You hear it in the news and are probably told by authority figures that after a certain time of the day, you probably want to be at home. Has that changed over the years? Has NYC seen more gun violence in the later hours of the day?

Acknowledging my personal bias that there has not been a significant change in shooting frequency at each hour of the day, I will scrutinize the data from the view point of someone who wants to showcase a change in shooting frequency.

To study the trends over time I will compare the frequency of shooting from the most recent 5 years with the preceding 5 years (5 years before the last 5 years).

Libraries Used: - library(dplyr) - library(BSDA) - library(tidyverse) - library(lubridate) - library(ggplot2)

Importing Data

Data is NYPD Shooting Incident Data (Historic), published by the City of New York and hosted on data.gov.

Each row of data is a seperate shooting incident.

The last update to the data was on September 2, 2023.

```
library(tidyverse)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

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

```
## Rows: 27312 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(df)
```

```
##   INCIDENT_KEY        OCCUR_DATE          OCCUR_TIME           BORO          
##  Min.   :  9953245   Length:27312       Length:27312      Length:27312      
##  1st Qu.: 63860880   Class :character   Class1:hms        Class :character  
##  Median : 90372218   Mode  :character   Class2:difftime   Mode  :character  
##  Mean   :120860536                      Mode  :numeric                      
##  3rd Qu.:188810230                                                          
##  Max.   :261190187                                                          
##                                                                             
##  LOC_OF_OCCUR_DESC     PRECINCT      JURISDICTION_CODE LOC_CLASSFCTN_DESC
##  Length:27312       Min.   :  1.00   Min.   :0.0000    Length:27312      
##  Class :character   1st Qu.: 44.00   1st Qu.:0.0000    Class :character  
##  Mode  :character   Median : 68.00   Median :0.0000    Mode  :character  
##                     Mean   : 65.64   Mean   :0.3269                      
##                     3rd Qu.: 81.00   3rd Qu.:0.0000                      
##                     Max.   :123.00   Max.   :2.0000                      
##                                      NA's   :2                           
##  LOCATION_DESC      STATISTICAL_MURDER_FLAG PERP_AGE_GROUP    
##  Length:27312       Mode :logical           Length:27312      
##  Class :character   FALSE:22046             Class :character  
##  Mode  :character   TRUE :5266              Mode  :character  
##                                                               
##                                                               
##                                                               
##                                                               
##    PERP_SEX          PERP_RACE         VIC_AGE_GROUP        VIC_SEX         
##  Length:27312       Length:27312       Length:27312       Length:27312      
##  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:27312       Min.   : 914928   Min.   :125757   Min.   :40.51  
##  Class :character   1st Qu.:1000028   1st Qu.:182834   1st Qu.:40.67  
##  Mode  :character   Median :1007731   Median :194487   Median :40.70  
##                     Mean   :1009449   Mean   :208127   Mean   :40.74  
##                     3rd Qu.:1016838   3rd Qu.:239518   3rd Qu.:40.82  
##                     Max.   :1066815   Max.   :271128   Max.   :40.91  
##                                                        NA's   :10     
##    Longitude         Lon_Lat         
```

```
##  Min.   :-74.25   Length:27312
##  1st Qu.:-73.94   Class :character
##  Median :-73.92   Mode  :character
##  Mean   :-73.91
##  3rd Qu.:-73.88
##  Max.   :-73.70
##  NA's   :10
```

Tidying Data

Transforming data prior to further analysis.

Number of unique values in each column:

- This is helpful for determining which columns are important categorical variables vs unique identifiers.

```r
library(tidyverse)
library(dplyr)

df %>%
  summarise_all(n_distinct)
```

```
## # A tibble: 1 x 21
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME  BORO LOC_OF_OCCUR_DESC PRECINCT
##          <int>      <int>      <int> <int>             <int>    <int>
## 1        21420       5761       1421     5                 3       77
## # i 15 more variables: JURISDICTION_CODE <int>, LOC_CLASSFCTN_DESC <int>,
## #   LOCATION_DESC <int>, STATISTICAL_MURDER_FLAG <int>, PERP_AGE_GROUP <int>,
## #   PERP_SEX <int>, PERP_RACE <int>, VIC_AGE_GROUP <int>, VIC_SEX <int>,
## #   VIC_RACE <int>, X_COORD_CD <int>, Y_COORD_CD <int>, Latitude <int>,
## #   Longitude <int>, Lon_Lat <int>
```

Number of null values in each column:

```r
df %>%
  summarise_all(~sum(is.na(.))) %>%
  t() %>%
  as.data.frame() %>%
  rename(Null_Count = V1) %>%
  arrange(Null_Count)
```

```
##                         Null_Count
## INCIDENT_KEY                     0
## OCCUR_DATE                       0
## OCCUR_TIME                       0
## BORO                             0
## PRECINCT                         0
## STATISTICAL_MURDER_FLAG          0
## VIC_AGE_GROUP                    0
## VIC_SEX                          0
## VIC_RACE                         0
## X_COORD_CD                       0
```

```
## Y_COORD_CD                    0
## JURISDICTION_CODE             2
## Latitude                     10
## Longitude                    10
## Lon_Lat                      10
## PERP_SEX                   9310
## PERP_RACE                  9310
## PERP_AGE_GROUP             9344
## LOCATION_DESC             14977
## LOC_OF_OCCUR_DESC         25596
## LOC_CLASSFCTN_DESC        25596
```

Dropping Out-of-Scope Columns

For this project, the only columns we will start our analysis with are the date, time, longitude, and latitude.

Please see further information regarding each column.

- Location and other coordinate based columns are missing a lot of data and will not be used in this project. Longitude and Latitude will be kept for further spatial analysis.
- Boro, jurisdiction code, and precinct are all columns that provide spatial information, however, I will be enriching the data using miles to dowwntown Manhattan using the longitude and latitude coordinate points.
- Perpetrator columns are missing 9,310 values. For this project, perpetrator information is out of scope, however, this is a potential exploration path for further investigation regarding perpetrator profiling after dropping missing values.
- The victim columns appear complete, however, out of scope for this project.
- The incident key column appears to be an unique identifier used to enrich the data through another data set. That is out of scope for this project.
- The statistical murder flag is another interesting column that invites further exploration of analysis. For this project, however, it is out of scope.

```r
df = df %>%
  select(OCCUR_DATE,OCCUR_TIME,Latitude,Longitude)
```

10 Shooting Cases missing Longitude/Latitude

- These will have to be dropped since we do not have a method of filling in implied/assumed data points.
- 10 Cases is a fraction of a percent and will not have impact on the analysis.

```r
print(sum(is.na(df$Longitude)) / nrow(df))
```

```
## [1] 0.0003661394
```

```r
df = df %>%
  filter(!is.na(Longitude))
```

Feature Engineering

Modifying the Population of Data

- Create a new column called RECENT_FLAG. This will have a 1 if the case was from the last 5 years and a 0 if it was from the 5 years before the last 5 years. Any cases older than 10 years will be dropped.

```
library(lubridate)

df = df %>%
  mutate(
    OCCUR_DATE = mdy(OCCUR_DATE),
    RECENT_FLAG = case_when(
      year(OCCUR_DATE) >= 2018 ~ 1,
      between(year(OCCUR_DATE), 2013, 2017) ~ 0,
      TRUE ~ NA_real_
    )
  ) %>%
  filter(!is.na(RECENT_FLAG))
```

Modifying Date column

- While further analysis at a more granular level may require the full date, my project will analyze the shootings at a monthly scope.

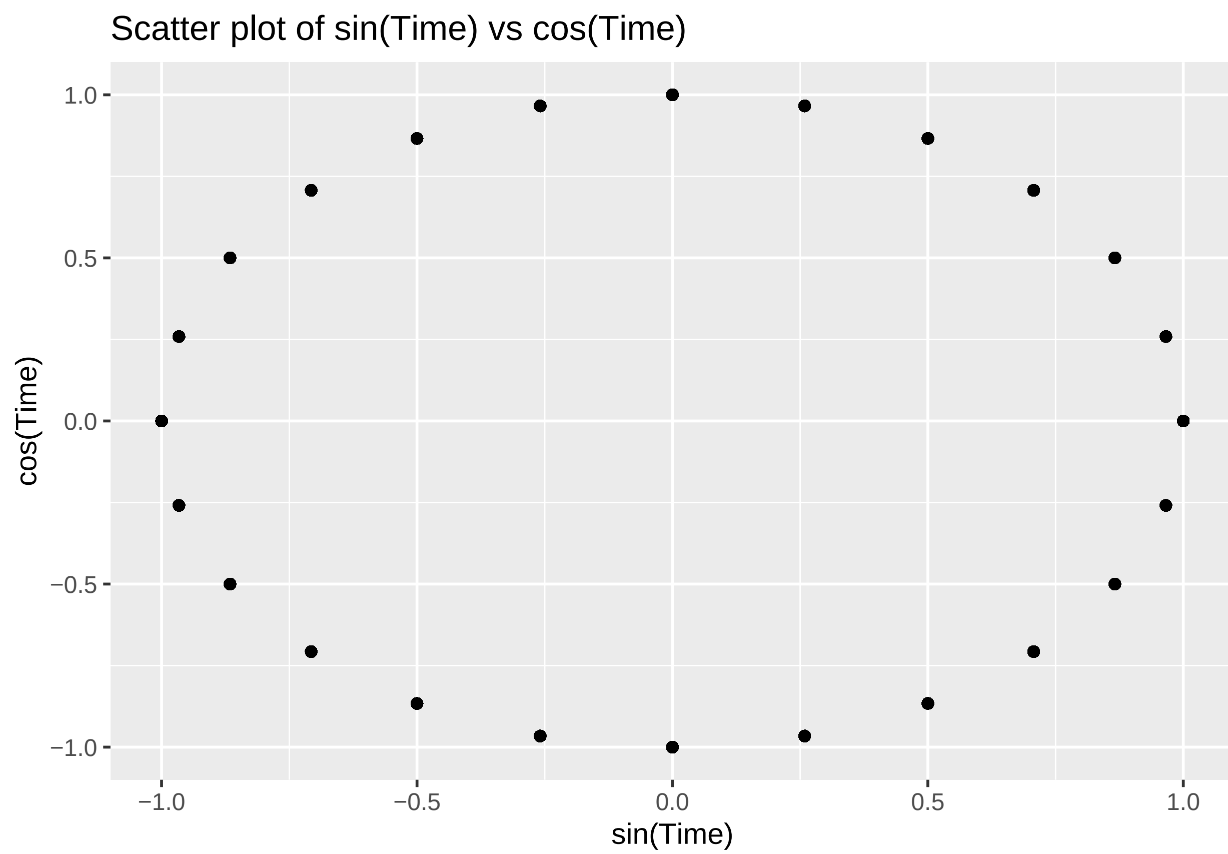
```
df = df %>%
  mutate(
    OCCUR_YEAR = year(OCCUR_DATE),
    OCCUR_MONTH = month(OCCUR_DATE)
  )

df = df %>%
  mutate(
    OCCUR_DATE = OCCUR_YEAR * 100 + OCCUR_MONTH
  ) %>%
  select(-OCCUR_YEAR, -OCCUR_MONTH)
```

Cyclical Encoding Time

- Time columns must be cyclically encoded as the time 24 is close to the time 1, but will not be treated that way in any regression analysis. Without proper encoding any analysis will assume those times are far from each other.
- When visualizing volume per time, non cyclical Time is okay as it is a categorical value. This project will analyze shootings at an hourly basis, and therefore, drop the minutes and seconds.

```
library(ggplot2)

df = df %>%
  mutate(OCCUR_TIME = as.integer(substr(OCCUR_TIME, 1, 2)))

df = df %>%
  mutate(
    sin_Time = sin(2 * pi * OCCUR_TIME / 24),
    cos_Time = cos(2 * pi * OCCUR_TIME / 24)
  )
```

```r
ggplot(df, aes(x = sin_Time, y = cos_Time)) +
  geom_point() +
  xlab("sin(Time)") +
  ylab("cos(Time)") +
  ggtitle("Scatter plot of sin(Time) vs cos(Time)")
```



Creating a TIME Group

- While the cyclical encoded time can be used to examine the regression based relationship between time and frequency of shooting, I will create 4 groups of 6 hours as a categorical variable to examine the differences in mean and proportions.

```r
df$TIME = ifelse((df$OCCUR_TIME >= 22 | (df$OCCUR_TIME >= 0 & df$OCCUR_TIME < 4)), 'Night',
                  ifelse((df$OCCUR_TIME >= 4 & df$OCCUR_TIME < 10), 'Morning',
                         ifelse((df$OCCUR_TIME >= 10 & df$OCCUR_TIME < 16), 'Noon',
                                ifelse((df$OCCUR_TIME >= 16 & df$OCCUR_TIME < 22), 'Evening',
                                       'ERROR'))))

table(df$TIME)
```

```
##
## Evening Morning   Night    Noon
##    4583    1635    5858    1929
```

Distance to downtown Manhattan

- Leveraging the Haversine distance formula, we will explore the distance between the shooting event and downtown Manhattan.

```
deg2rad = function(degrees) {
  radians = degrees * (pi / 180)
  return(radians)
}

haversine_distance = function(lat1, lon1, lat2, lon2) {
  lat1 = deg2rad(lat1)
  lon1 = deg2rad(lon1)
  lat2 = deg2rad(lat2)
  lon2 = deg2rad(lon2)

  dlon = lon2 - lon1
  dlat = lat2 - lat1
  a = sin(dlat/2)^2 + cos(lat1) * cos(lat2) * sin(dlon/2)^2
  c = 2 * asin(sqrt(a))
  distance = 3963.0 * c

  return(distance)
}

manhattan_lat = 40.720259
manhattan_lon = -74.000772

df = df %>%
  mutate(Distance_Downtown = haversine_distance(manhattan_lat, manhattan_lon, Latitude, Longitude)) %>%
  select(-Latitude, -Longitude)
```

Creating a DISTANCE Group

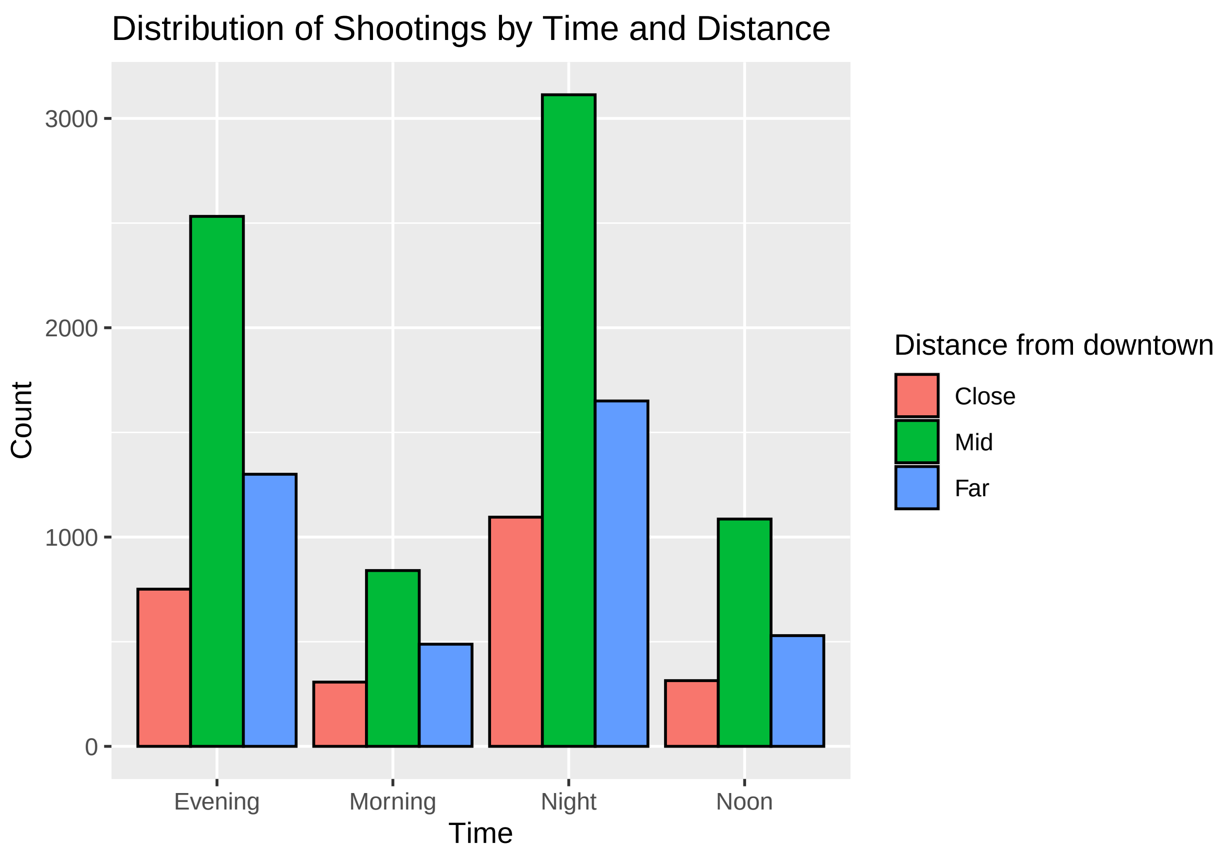
- While the miles away from downtown can be used to examine the regression based relationship between distance and frequency of shooting, I will create 3 groups of distances as a categorical variable to examine the differences in mean and proportions.

```
dist_bin = c(0, 5, 10, Inf)
dist_labels = c('Close', 'Mid', 'Far')

df$DISTANCE = cut(df$Distance_Downtown, breaks = dist_bin, labels = dist_labels, right = FALSE)
```

Visualyzing and Analyzing Data

Starting off, lets explore with a simple "heatmap" to examine the most popular time of day and distance grouping for shooting frequency. It appears that the Night and Evening medium distance away from downtown Manhattan are the most frequent. Is that distinction between medium distance and the other distances visibile in other visualizations?

```
ggplot(df, aes(x = TIME, fill = DISTANCE)) +
  geom_bar(position = "dodge", color = "black") +
  labs(x = "Time", y = "Count", fill = "Distance from downtown") +
  ggtitle("Distribution of Shootings by Time and Distance")
```

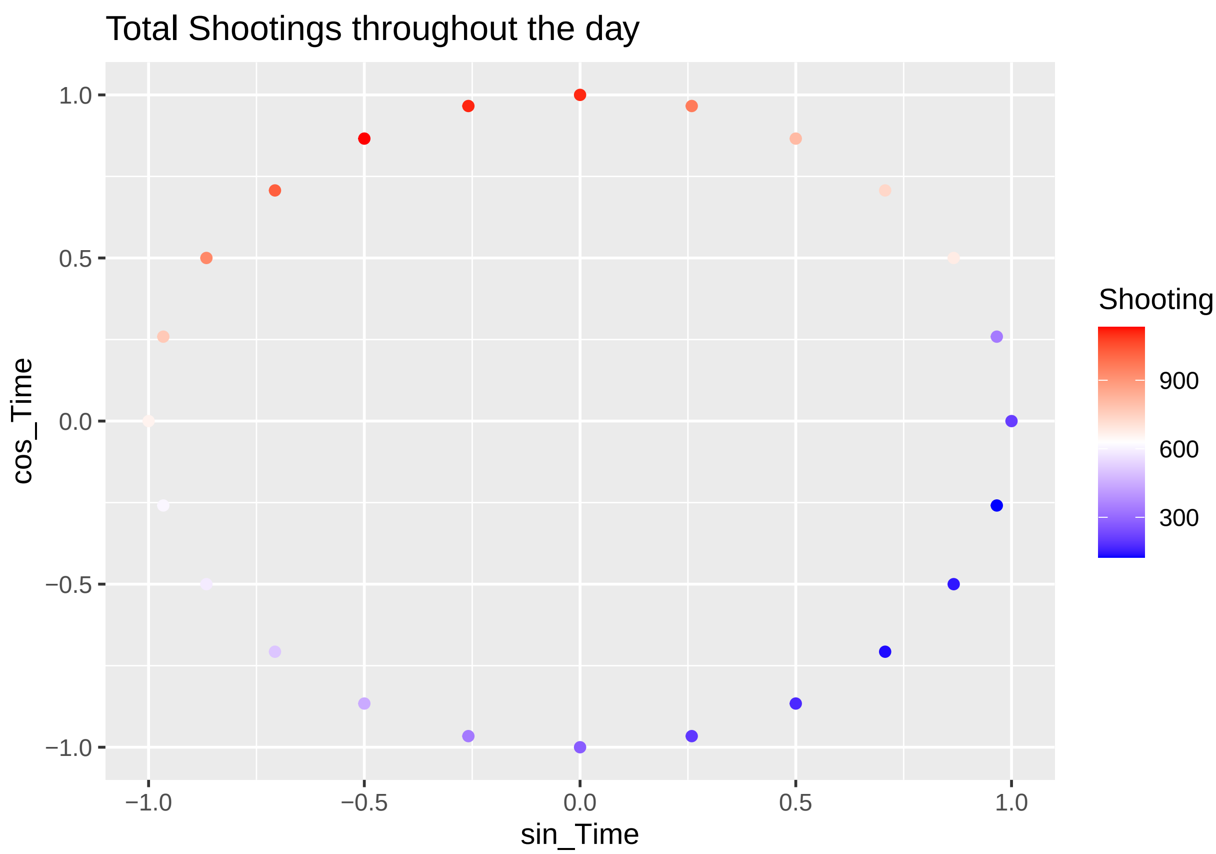


Volume of shooting at each time of the day

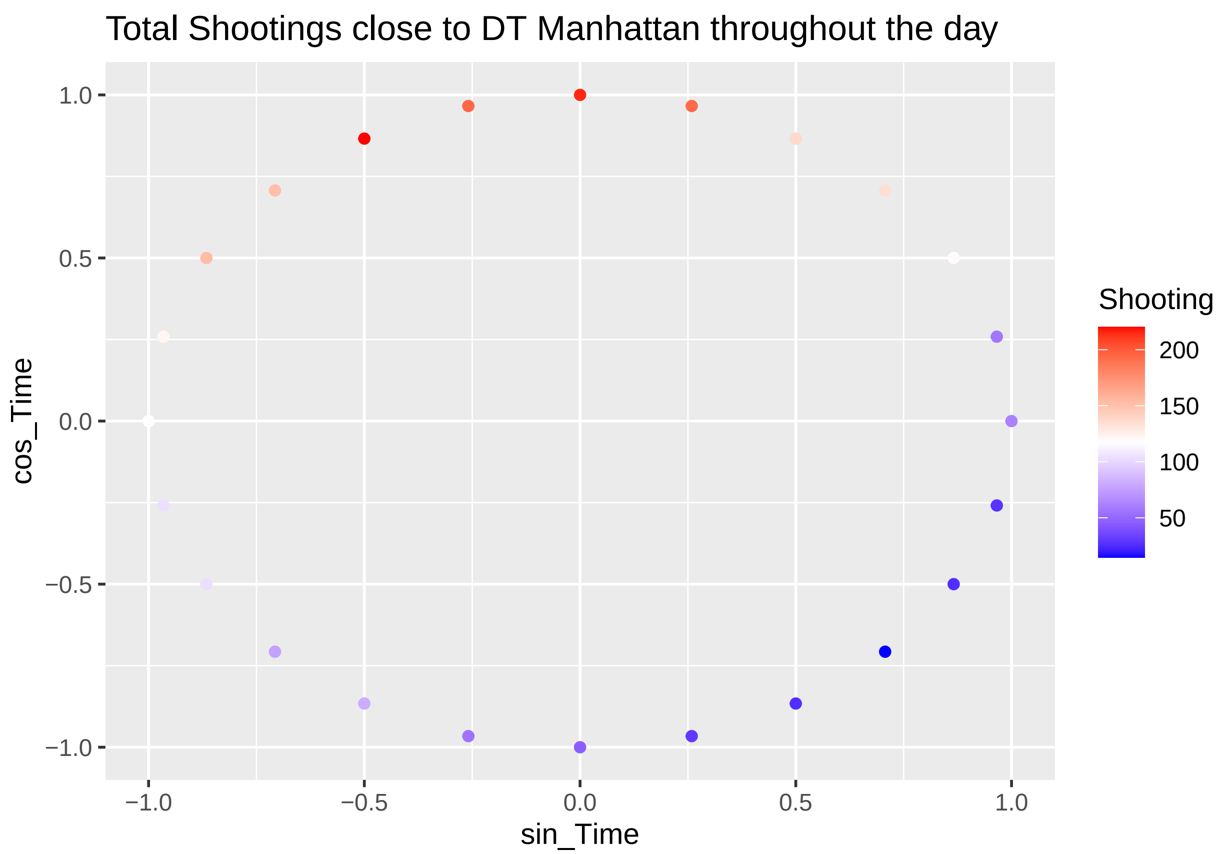
These plots are read like a 24 hour clock. Imagine if a normal clock had 24 indice markers instead of 12. The top center of the clock is midnight and the bottom center of the clock is noon.

- Additional Question for Further Investigation: Are there differences in the proportions of shootings at each time sector of the day? This will be further explored in the modeling section.

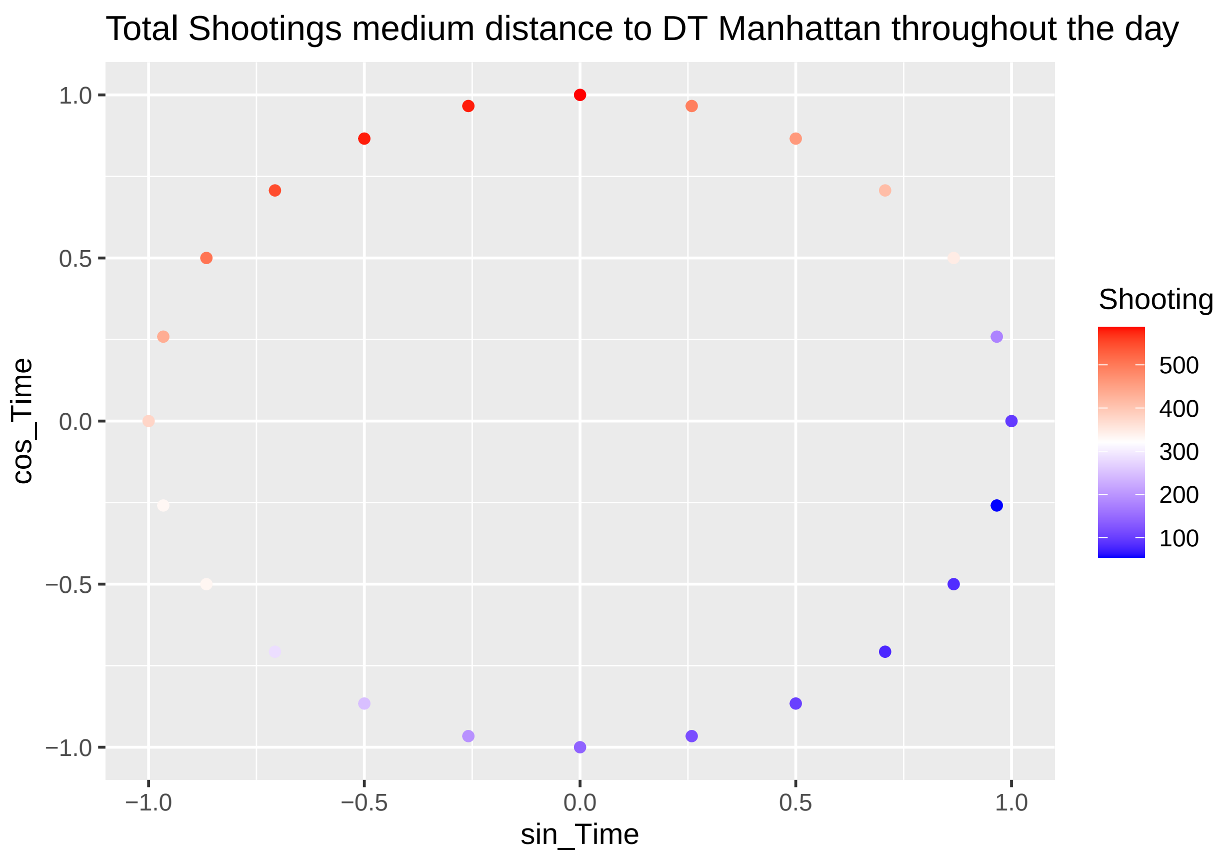
```
all_dist = df %>%
  group_by(OCCUR_TIME, sin_Time, cos_Time) %>%
  summarise(Shooting = n(),.groups = 'drop') %>%
  ungroup()
ggplot(all_dist, aes(x = sin_Time, y = cos_Time, color = Shooting)) +
  geom_point() +
  scale_color_gradientn(colors = c("blue", "white", "red")) +
  labs(x = "sin_Time", y = "cos_Time", color = "Shooting") +
  ggtitle("Total Shootings throughout the day")
```



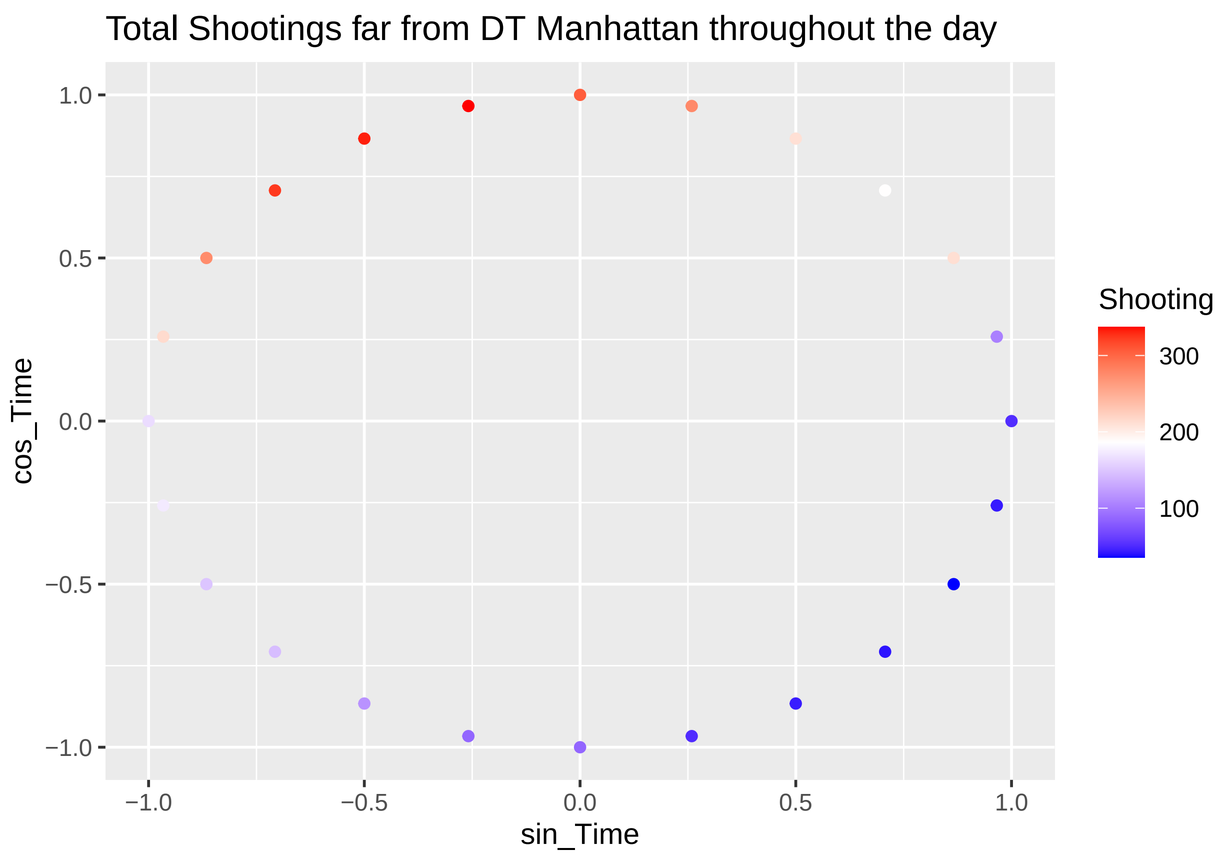

```r
close_dist = df %>%
  filter(DISTANCE == "Close") %>%
  group_by(OCCUR_TIME, sin_Time, cos_Time) %>%
  summarise(Shooting = n(),.groups = 'drop') %>%
  ungroup()
ggplot(close_dist, aes(x = sin_Time, y = cos_Time, color = Shooting)) +
  geom_point() +
  scale_color_gradientn(colors = c("blue", "white", "red")) +
  labs(x = "sin_Time", y = "cos_Time", color = "Shooting") +
  ggtitle("Total Shootings close to DT Manhattan throughout the day")
```




```r
mid_dist = df %>%
  filter(DISTANCE == "Mid") %>%
  group_by(OCCUR_TIME, sin_Time, cos_Time) %>%
  summarise(Shooting = n(),.groups = 'drop') %>%
  ungroup()
ggplot(mid_dist, aes(x = sin_Time, y = cos_Time, color = Shooting)) +
  geom_point() +
  scale_color_gradientn(colors = c("blue", "white", "red")) +
  labs(x = "sin_Time", y = "cos_Time", color = "Shooting") +
  ggtitle("Total Shootings medium distance to DT Manhattan throughout the day")
```




```r
far_dist = df %>%
  filter(DISTANCE == "Far") %>%
  group_by(OCCUR_TIME, sin_Time, cos_Time) %>%
  summarise(Shooting = n(),.groups = 'drop') %>%
  ungroup()
ggplot(far_dist, aes(x = sin_Time, y = cos_Time, color = Shooting)) +
  geom_point() +
  scale_color_gradientn(colors = c("blue", "white", "red")) +
  labs(x = "sin_Time", y = "cos_Time", color = "Shooting") +
  ggtitle("Total Shootings far from DT Manhattan throughout the day")
```

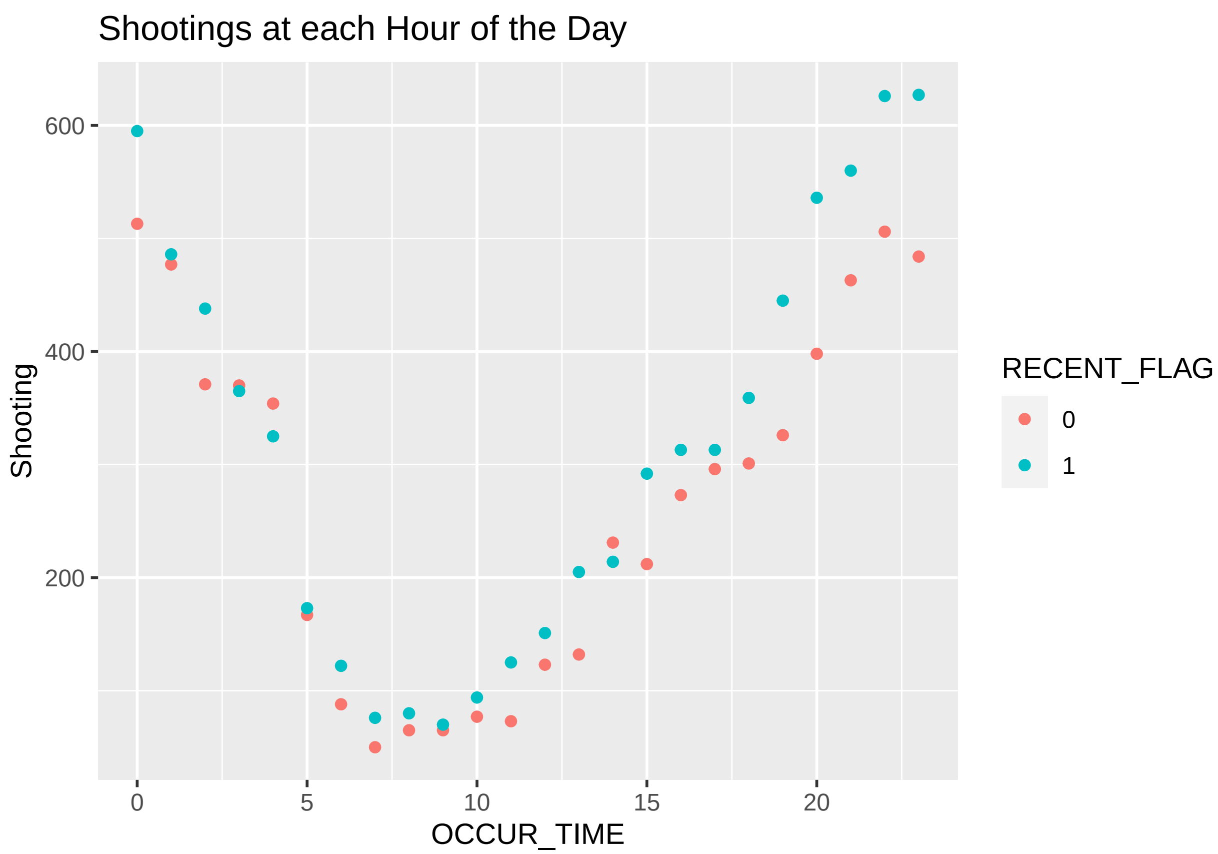


Volume of shooting at each time of the day (Recent 5Y vs Preceding 5Y)

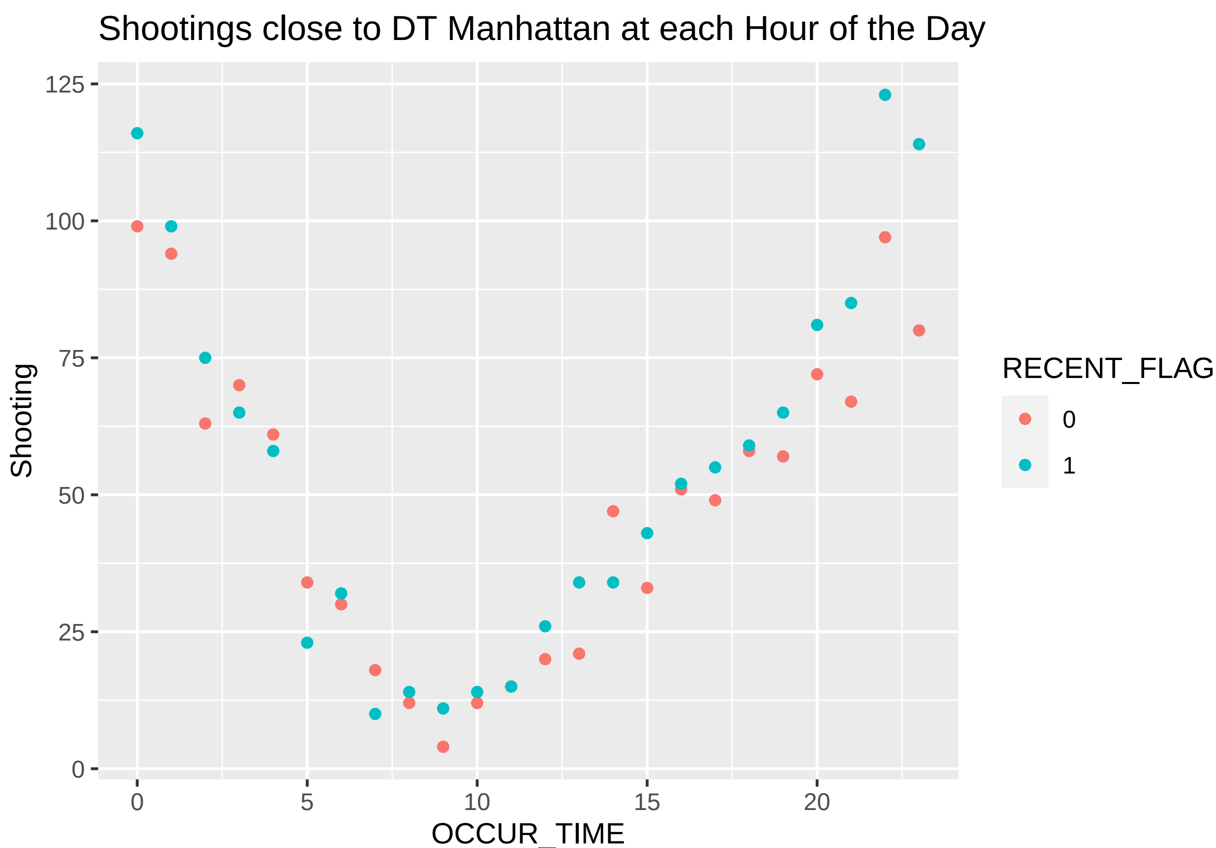
Are there more shootings at certain times of the day in the most recent 5 years compared to the preceding 5 years?

- Additional Question for Further Investigation: Each distance looks to have a similar pattern where there are more shootings recently at night than there were in the preceding 5 years. The monthly mean shootings at each time and distance to downtown will be compared between recent 5 years and preceding 5 years to test for differences.

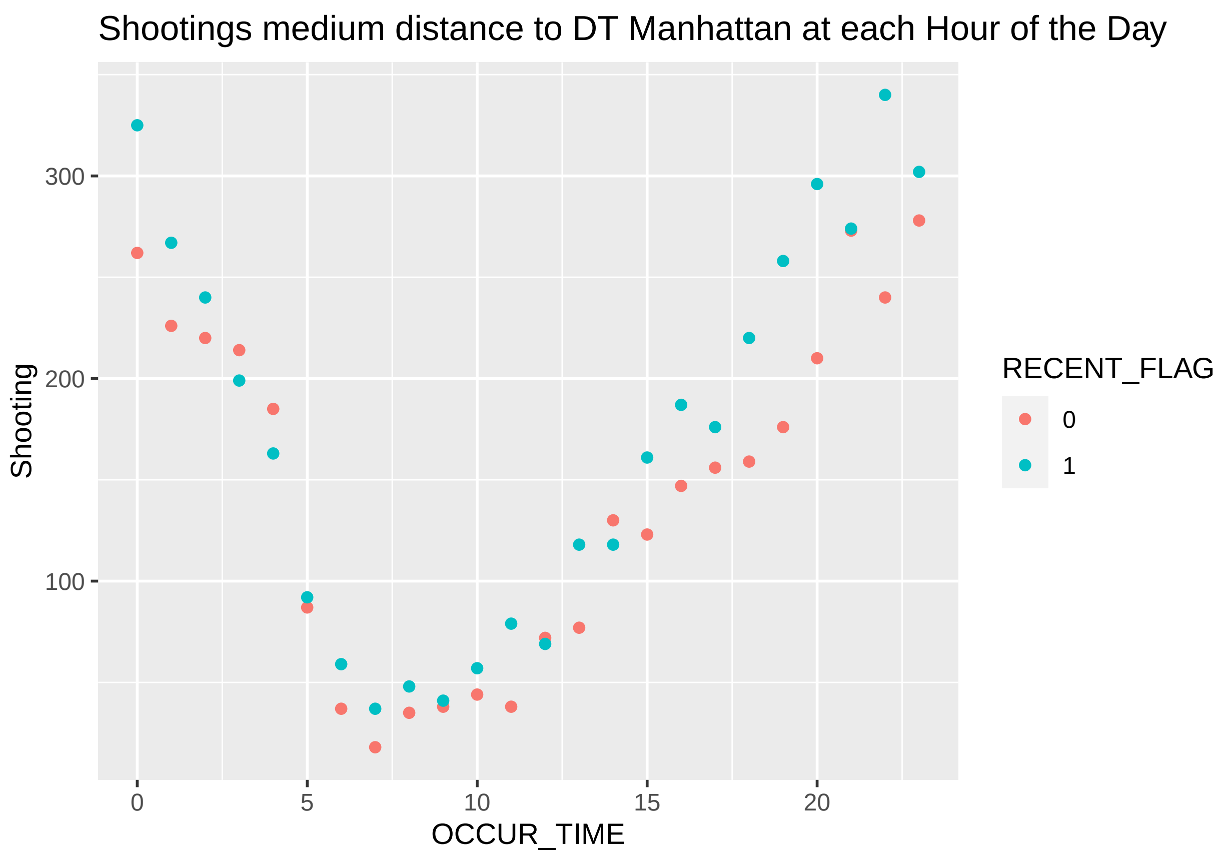
```
all_dist_t = df %>%
  group_by(OCCUR_TIME, RECENT_FLAG) %>%
  summarise(Shooting = n(),.groups = 'drop') %>%
  ungroup()
ggplot(all_dist_t, aes(x = OCCUR_TIME, y = Shooting, color = factor(RECENT_FLAG))) +
  geom_point() +
  labs(x = "OCCUR_TIME", y = "Shooting", color = "RECENT_FLAG") +
  ggtitle("Shootings at each Hour of the Day")
```



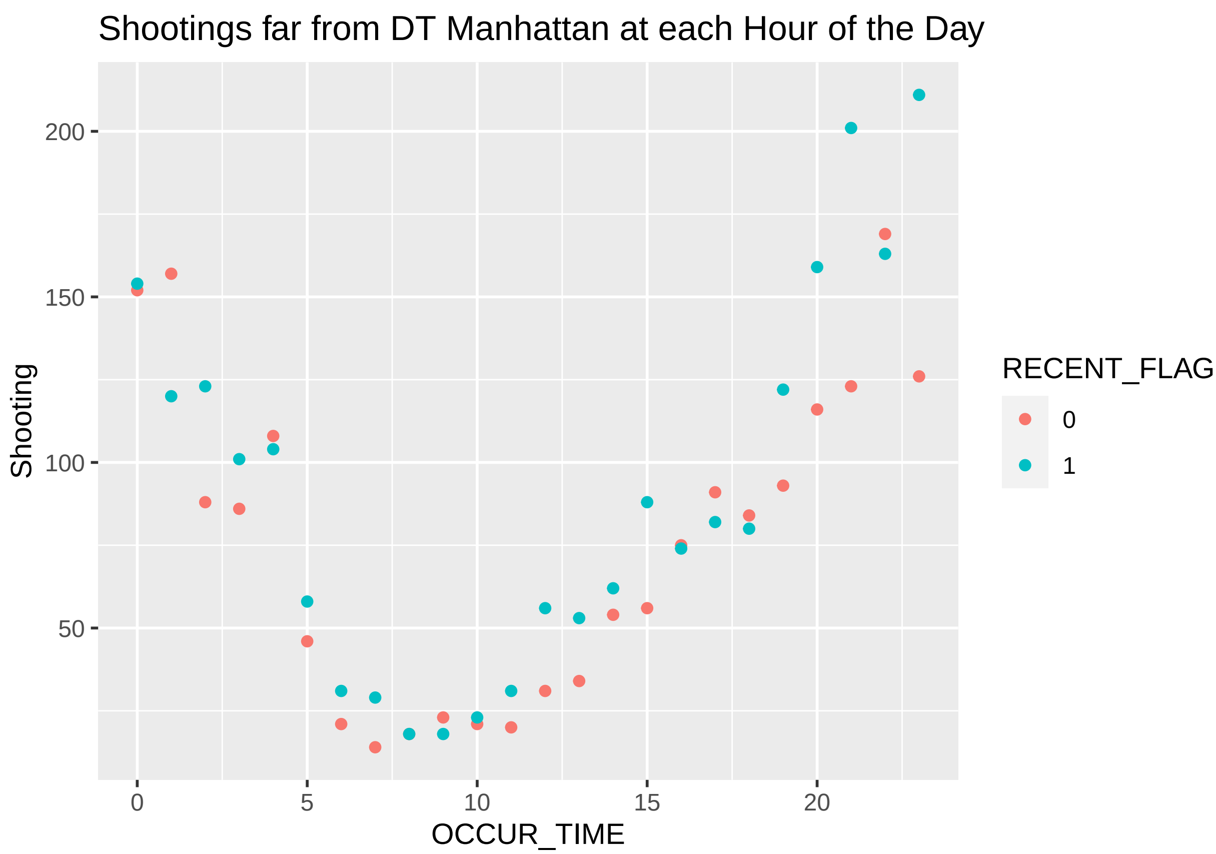

```r
close_dist_t = df %>%
  filter(DISTANCE == "Close") %>%
  group_by(OCCUR_TIME, RECENT_FLAG) %>%
  summarise(Shooting = n(),.groups = 'drop') %>%
  ungroup()
ggplot(close_dist_t, aes(x = OCCUR_TIME, y = Shooting, color = factor(RECENT_FLAG))) +
  geom_point() +
  labs(x = "OCCUR_TIME", y = "Shooting", color = "RECENT_FLAG") +
  ggtitle("Shootings close to DT Manhattan at each Hour of the Day")
```




```
mid_dist_t = df %>%
  filter(DISTANCE == "Mid") %>%
  group_by(OCCUR_TIME, RECENT_FLAG) %>%
  summarise(Shooting = n(),.groups = 'drop') %>%
  ungroup()
ggplot(mid_dist_t, aes(x = OCCUR_TIME, y = Shooting, color = factor(RECENT_FLAG))) +
  geom_point() +
  labs(x = "OCCUR_TIME", y = "Shooting", color = "RECENT_FLAG") +
  ggtitle("Shootings medium distance to DT Manhattan at each Hour of the Day")
```



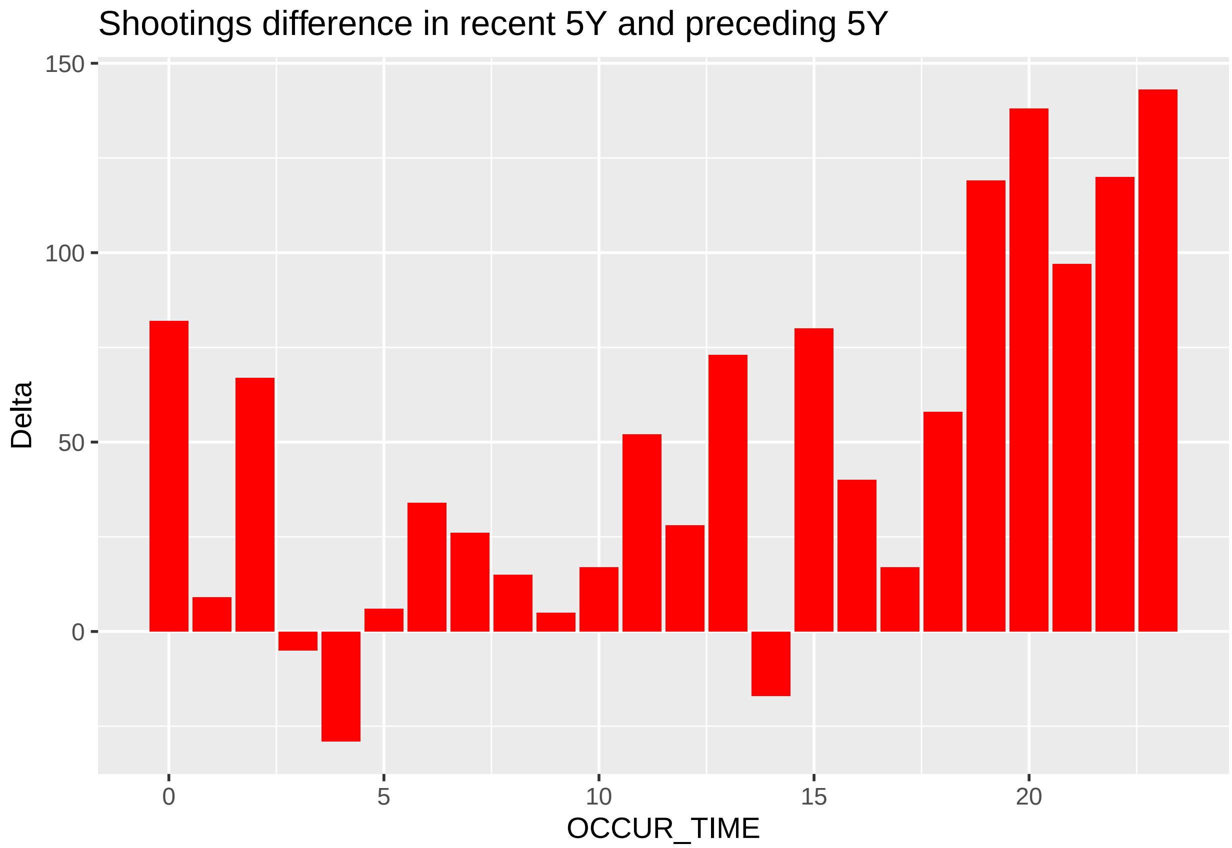

```r
far_dist_t = df %>%
  filter(DISTANCE == "Far") %>%
  group_by(OCCUR_TIME, RECENT_FLAG) %>%
  summarise(Shooting = n(),.groups = 'drop') %>%
  ungroup()
ggplot(far_dist_t, aes(x = OCCUR_TIME, y = Shooting, color = factor(RECENT_FLAG))) +
  geom_point() +
  labs(x = "OCCUR_TIME", y = "Shooting", color = "RECENT_FLAG") +
  ggtitle("Shootings far from DT Manhattan at each Hour of the Day")
```



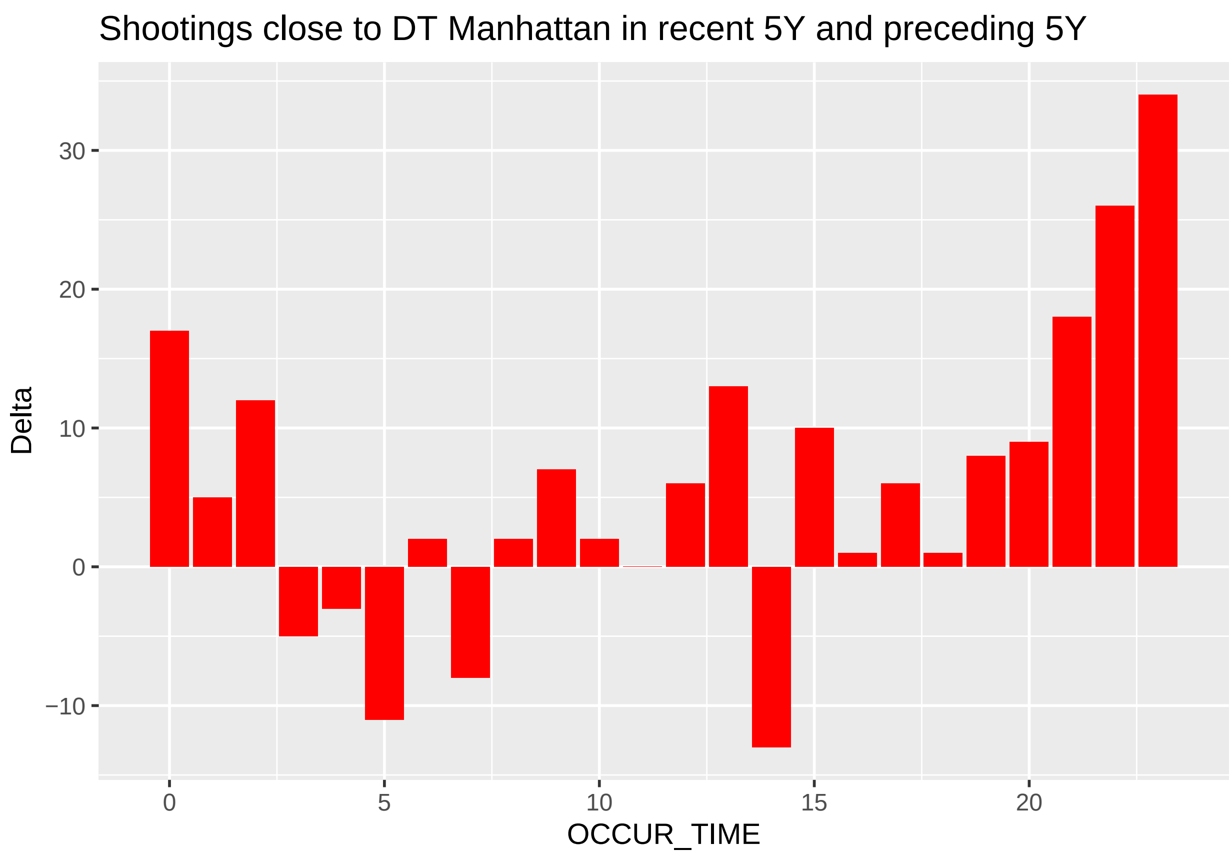
Delta of the volume of shootings at each time of the day (Recent 5Y vs Preceding 5Y)

This graph explores the differences in delta of shooting volume at each time of the day across the various distances to downtown. We will further explore if these differences are greater than 0 in the modeling section.

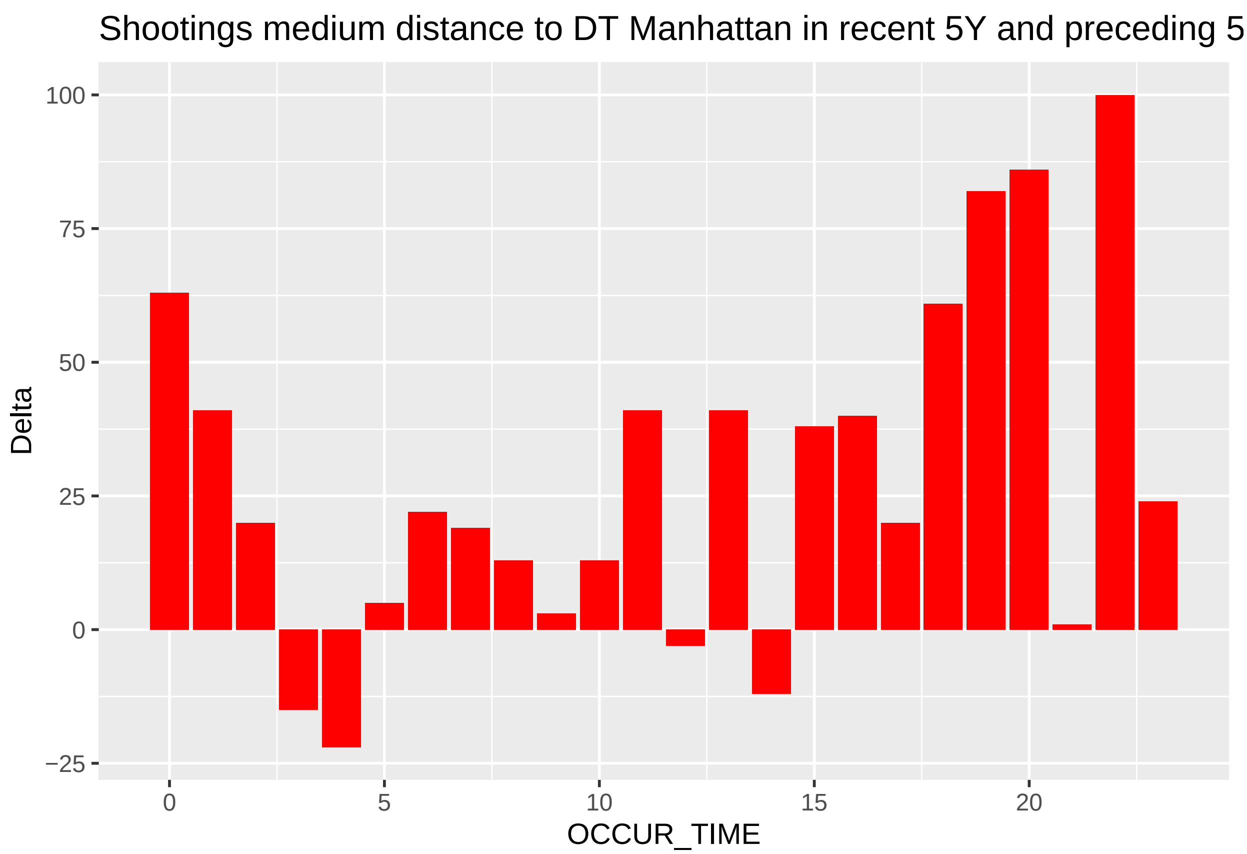
```
delta_all = df %>%
  group_by(OCCUR_TIME, RECENT_FLAG) %>%
  summarise(index_count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = RECENT_FLAG, values_from = index_count) %>%
  mutate(Delta = `1` - `0`) %>%
  ungroup()
ggplot(delta_all, aes(x = OCCUR_TIME, y = Delta)) +
  geom_bar(stat = "identity", fill = "red") +
  labs(x = "OCCUR_TIME", y = "Delta", title = "Shootings difference in recent 5Y and preceding 5Y")
```



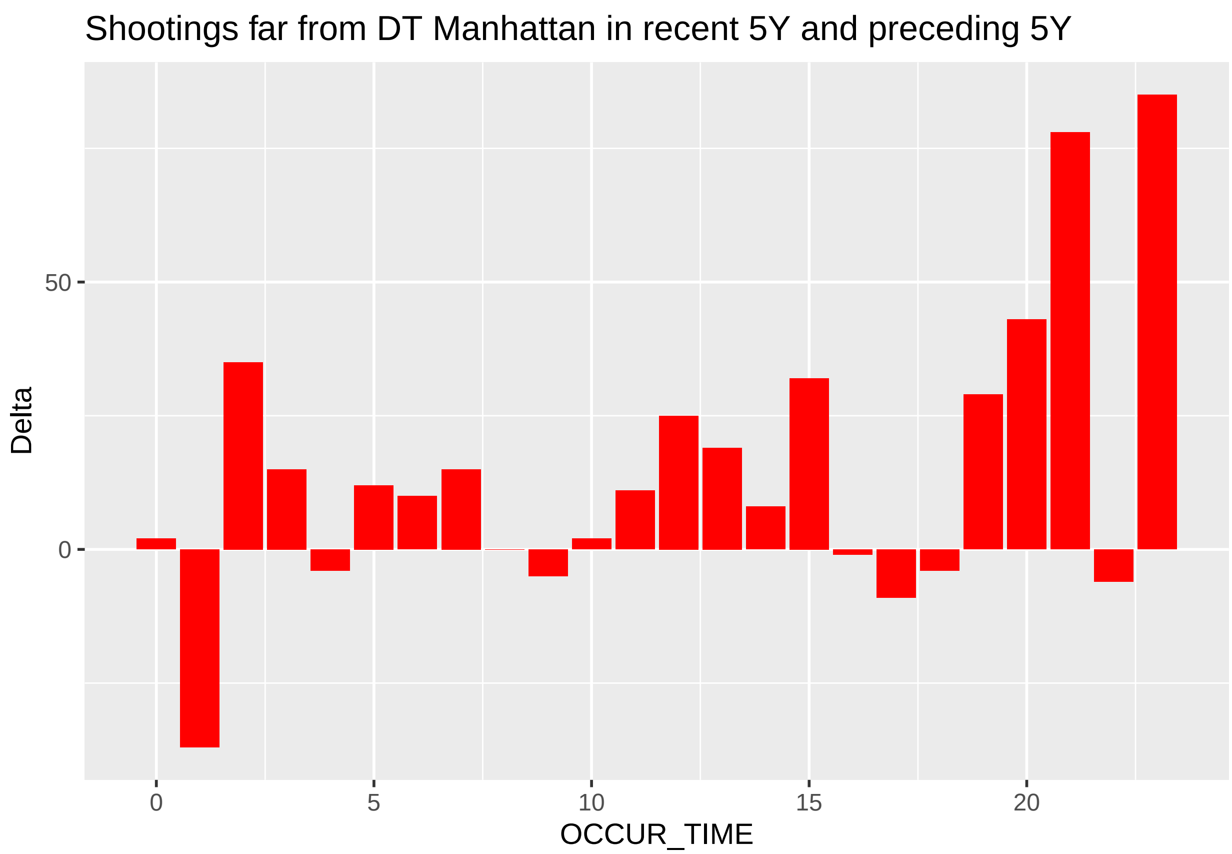

```
delta_close = df %>%
  filter(DISTANCE == "Close") %>%
  group_by(OCCUR_TIME, RECENT_FLAG) %>%
  summarise(index_count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = RECENT_FLAG, values_from = index_count) %>%
  mutate(Delta = `1` - `0`) %>%
  ungroup()
ggplot(delta_close, aes(x = OCCUR_TIME, y = Delta)) +
  geom_bar(stat = "identity", fill = "red") +
  labs(x = "OCCUR_TIME", y = "Delta", title = "Shootings close to DT Manhattan in recent 5Y and precedi
```




```
delta_mid = df %>%
  filter(DISTANCE == "Mid") %>%
  group_by(OCCUR_TIME, RECENT_FLAG) %>%
  summarise(index_count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = RECENT_FLAG, values_from = index_count) %>%
  mutate(Delta = `1` - `0`) %>%
  ungroup()
ggplot(delta_mid, aes(x = OCCUR_TIME, y = Delta)) +
  geom_bar(stat = "identity", fill = "red") +
  labs(x = "OCCUR_TIME", y = "Delta", title = "Shootings medium distance to DT Manhattan in recent 5Y ar
```




```
delta_far = df %>%
  filter(DISTANCE == "Far") %>%
  group_by(OCCUR_TIME, RECENT_FLAG) %>%
  summarise(index_count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = RECENT_FLAG, values_from = index_count) %>%
  mutate(Delta = `1` - `0`) %>%
  ungroup()
ggplot(delta_far, aes(x = OCCUR_TIME, y = Delta)) +
  geom_bar(stat = "identity", fill = "red") +
  labs(x = "OCCUR_TIME", y = "Delta", title = "Shootings far from DT Manhattan in recent 5Y and precedi
```



Modeling Data

Linear Regression: Distance to Downtown Manhattan and Shooting Volume Multivariate Linear Regression model fit for cos and sin Time of Day and Shooting Volume using data from the recent 5 years.

```
recent_data = df %>%
  filter(RECENT_FLAG == 1) %>%
  group_by(sin_Time, cos_Time) %>%
  summarise(Shootings = n(), .groups = 'drop') %>%
  ungroup()

linear_model = lm(Shootings ~ sin_Time + cos_Time, data = recent_data)

summary(linear_model)
```

```
##
## Call:
## lm(formula = Shootings ~ sin_Time + cos_Time, data = recent_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -92.186 -24.253  -0.542  34.526  70.856
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   316.25      10.23  30.922  < 2e-16 ***
## sin_Time     -134.94      14.46  -9.329 6.44e-09 ***
## cos_Time      212.20      14.46  14.671 1.65e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.1 on 21 degrees of freedom
## Multiple R-squared:  0.935,  Adjusted R-squared:  0.9289
## F-statistic: 151.1 on 2 and 21 DF,  p-value: 3.409e-13
```

Multivariate Linear Regression model fit for cos and sin Time of Day and Shooting Volume using data from the preceding 5 years.

```r
preceding_data = df %>%
  filter(RECENT_FLAG == 0) %>%
  group_by(sin_Time, cos_Time) %>%
  summarise(Shootings = n(), .groups = 'drop') %>%
  ungroup()

linear_model_2 = lm(Shootings ~ sin_Time + cos_Time, data = preceding_data)

summary(linear_model_2)
```

```
##
## Call:
## lm(formula = Shootings ~ sin_Time + cos_Time, data = preceding_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -83.148 -30.510   9.997  30.637  78.103
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   267.29      10.29  25.964  < 2e-16 ***
## sin_Time      -96.14      14.56  -6.604 1.54e-06 ***
## cos_Time      187.60      14.56  12.886 1.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.43 on 21 degrees of freedom
## Multiple R-squared:  0.909,  Adjusted R-squared:  0.9003
## F-statistic: 104.8 on 2 and 21 DF,  p-value: 1.181e-11
```

Now we will explore the difference in means and proportions in shooting frequency at the time of the day between the last 5 years and the preceding 5 years. The chi-squared goodness of fit test will be used to compare the underlying distribution of shooting frequency between each Time Groups. We will compare the proportions for shootings all distance to DT Manhattan, Close, Medium, and Far to see if there is variation between shooting distribution over distance and time of day.

Then we will perform Z tests to identify if the mean monthly shooting average has changed from the preceding 5 years. Again, we will compare the mean monthly shootings for all distances, close, medium, and far from downtown.

Starting with difference in proportions and mean monthly shootings for ALL distances.

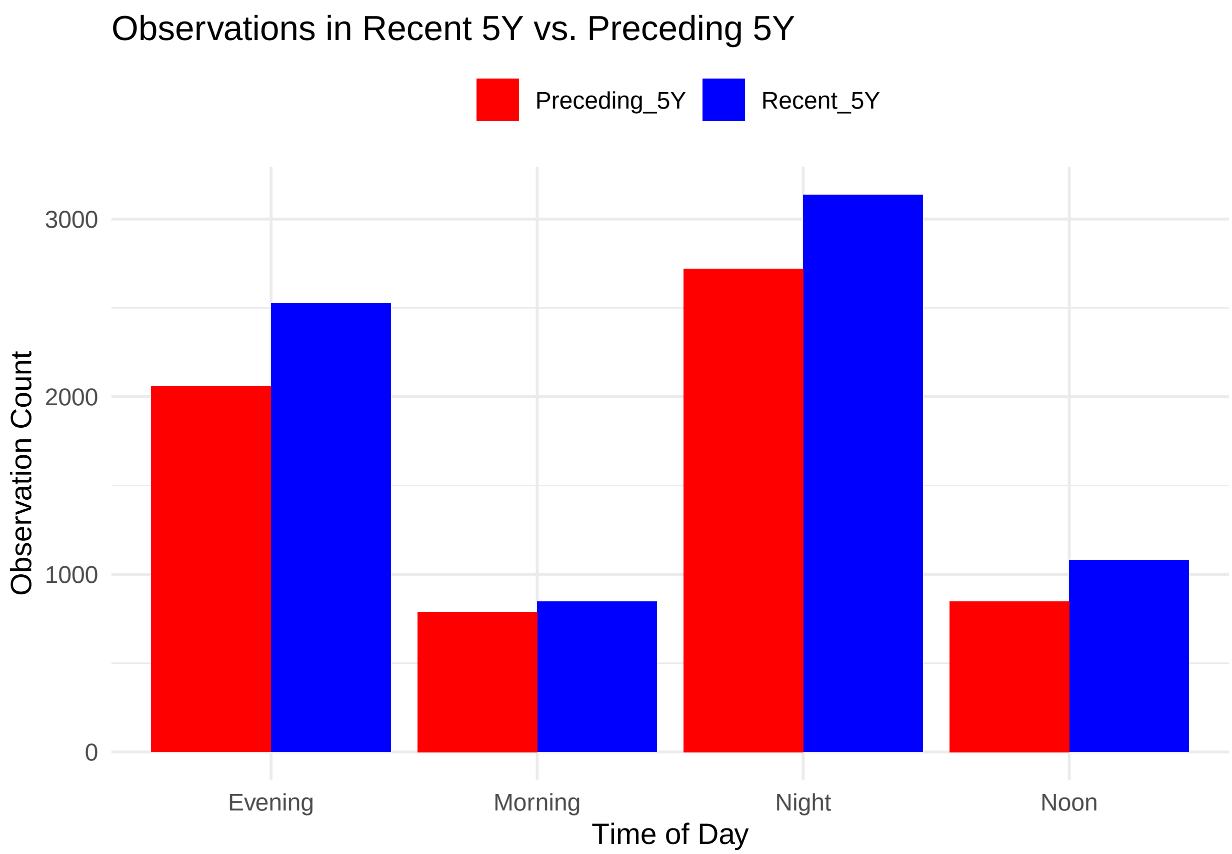
```r
library(BSDA)
```

```
## Loading required package: lattice

##
## Attaching package: 'BSDA'

## The following object is masked from 'package:datasets':
##
##     Orange
```

```r
recent_group = df %>%
  filter(RECENT_FLAG == 1) %>%
  group_by(OCCUR_DATE, TIME) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = TIME, values_from = count, values_fill = 0)

preced_group = df %>%
  filter(RECENT_FLAG == 0) %>%
  group_by(OCCUR_DATE, TIME) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = TIME, values_from = count, values_fill = 0)

observed_freq_1 = colSums(recent_group[, -1])
observed_freq_0 = colSums(preced_group[, -1])
bind = merge(data.frame(observed_freq_1),data.frame(observed_freq_0), by = "row.names")[,-1]
result = chisq.test(bind, correct = FALSE)

print(result)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bind
## X-squared = 9.1511, df = 3, p-value = 0.02735
```

```r
merged_df <- merge(data.frame(observed_freq_1),data.frame(observed_freq_0), by = "row.names")
names(merged_df) <- c("Time_of_Day", "Recent_5Y", "Preceding_5Y")
merged_df = tidyr::pivot_longer(merged_df, cols = c(`Recent_5Y`, `Preceding_5Y`), names_to = "Category"
merged_df$Time_of_Day <- as.character(merged_df$Time_of_Day)
ggplot(merged_df, aes(x = Time_of_Day, y = Observation, fill = Category)) +
  geom_col(position = "dodge") +
  labs(x = "Time of Day", y = "Observation Count", fill = NULL) +
  ggtitle("Observations in Recent 5Y vs. Preceding 5Y") +
  theme_minimal() +
  theme(legend.position = "top") +
  scale_fill_manual(values = c("Recent_5Y" = "blue", "Preceding_5Y" = "red"))
```




```
for (time in colnames(recent_group[, -1])) {
  z_stat = z.test(recent_group[[time]], preced_group[[time]], sigma.x = sd(recent_group[[time]]), sigma
  p_value = z.test(recent_group[[time]], preced_group[[time]], sigma.x = sd(recent_group[[time]]), sigma

  cat("\nZ-test for mean shootings per month in the", time, "between recent 5Y and Preceding 5Y:\n")
  cat("Recent 5Y Mean:", mean(recent_group[[time]]), "\n")
  cat("Preceding 5Y Mean:", mean(preced_group[[time]]), "\n")
  cat("Z-statistic:", z_stat, "\n")
  cat("P-value:", p_value, "\n")
}
```

```
##
## Z-test for mean shootings per month in the Evening between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 42.1
## Preceding 5Y Mean: 34.28333
## Z-statistic: 2.55168
## P-value: 0.01072051
##
## Z-test for mean shootings per month in the Morning between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 14.1
## Preceding 5Y Mean: 13.15
## Z-statistic: 0.8174545
## P-value: 0.4136687
##
## Z-test for mean shootings per month in the Night between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 52.28333
```

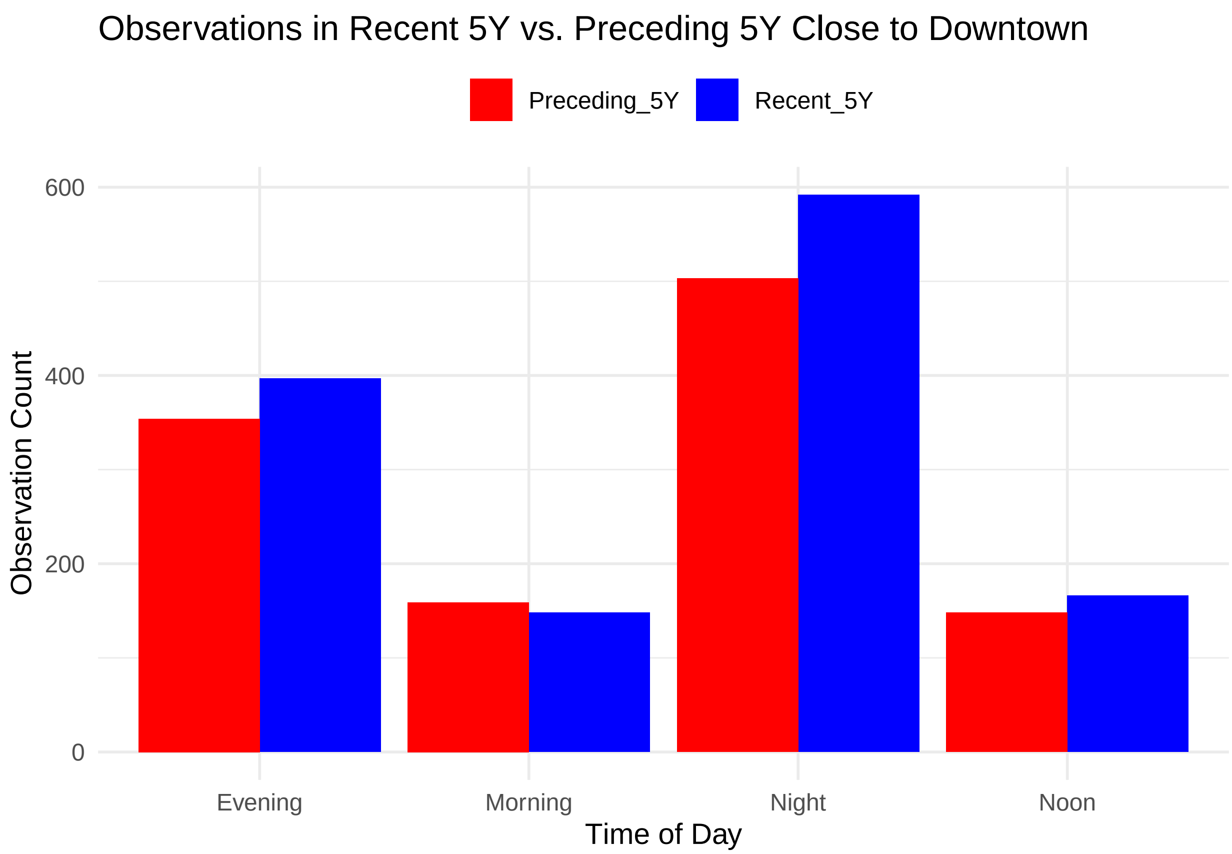
```
## Preceding 5Y Mean: 45.35
## Z-statistic: 1.352384
## P-value: 0.1762526
##
## Z-test for mean shootings per month in the Noon between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 18.01667
## Preceding 5Y Mean: 14.13333
## Z-statistic: 2.941962
## P-value: 0.0032614
```

Next, lets explore the difference in proportions and mean monthly shootings for close to DT.

```r
recent_close = df %>%
  filter(RECENT_FLAG == 1, DISTANCE == 'Close') %>%
  group_by(OCCUR_DATE, TIME) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = TIME, values_from = count, values_fill = 0)

preced_close = df %>%
  filter(RECENT_FLAG == 0, DISTANCE == 'Close') %>%
  group_by(OCCUR_DATE, TIME) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = TIME, values_from = count, values_fill = 0)

observed_freq_1 = colSums(recent_close[, -1])
observed_freq_0 = colSums(preced_close[, -1])
bind = merge(data.frame(observed_freq_1),data.frame(observed_freq_0), by = "row.names")[,-1]
result2 = chisq.test(bind, correct = FALSE)

print(result2)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bind
## X-squared = 3.3005, df = 3, p-value = 0.3476
```

```r
merged_df <- merge(data.frame(observed_freq_1),data.frame(observed_freq_0), by = "row.names")
names(merged_df) <- c("Time_of_Day", "Recent_5Y", "Preceding_5Y")
merged_df = tidyr::pivot_longer(merged_df, cols = c(`Recent_5Y`, `Preceding_5Y`), names_to = "Category"
merged_df$Time_of_Day <- as.character(merged_df$Time_of_Day)
ggplot(merged_df, aes(x = Time_of_Day, y = Observation, fill = Category)) +
  geom_col(position = "dodge") +
  labs(x = "Time of Day", y = "Observation Count", fill = NULL) +
  ggtitle("Observations in Recent 5Y vs. Preceding 5Y Close to Downtown") +
  theme_minimal() +
  theme(legend.position = "top") +
  scale_fill_manual(values = c("Recent_5Y" = "blue", "Preceding_5Y" = "red"))
```




```r
for (time in colnames(recent_close[, -1])) {
  z_stat = z.test(recent_close[[time]], preced_close[[time]], sigma.x = sd(recent_close[[time]]), sigma
  p_value = z.test(recent_close[[time]], preced_close[[time]], sigma.x = sd(recent_close[[time]]), sigma

  cat("\nZ-test for mean shootings Close to DT per month in the", time, "between recent 5Y and Precedin
  cat("Recent 5Y Mean:", mean(recent_close[[time]]), "\n")
  cat("Preceding 5Y Mean:", mean(preced_close[[time]]), "\n")
  cat("Z-statistic:", z_stat, "\n")
  cat("P-value:", p_value, "\n")
}
```

```
##
## Z-test for mean shootings Close to DT per month in the Evening between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 6.616667
## Preceding 5Y Mean: 5.9
## Z-statistic: 0.971208
## P-value: 0.3314447
##
## Z-test for mean shootings Close to DT per month in the Morning between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 2.466667
## Preceding 5Y Mean: 2.65
## Z-statistic: -0.3845542
## P-value: 0.7005677
##
## Z-test for mean shootings Close to DT per month in the Night between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 9.866667
```

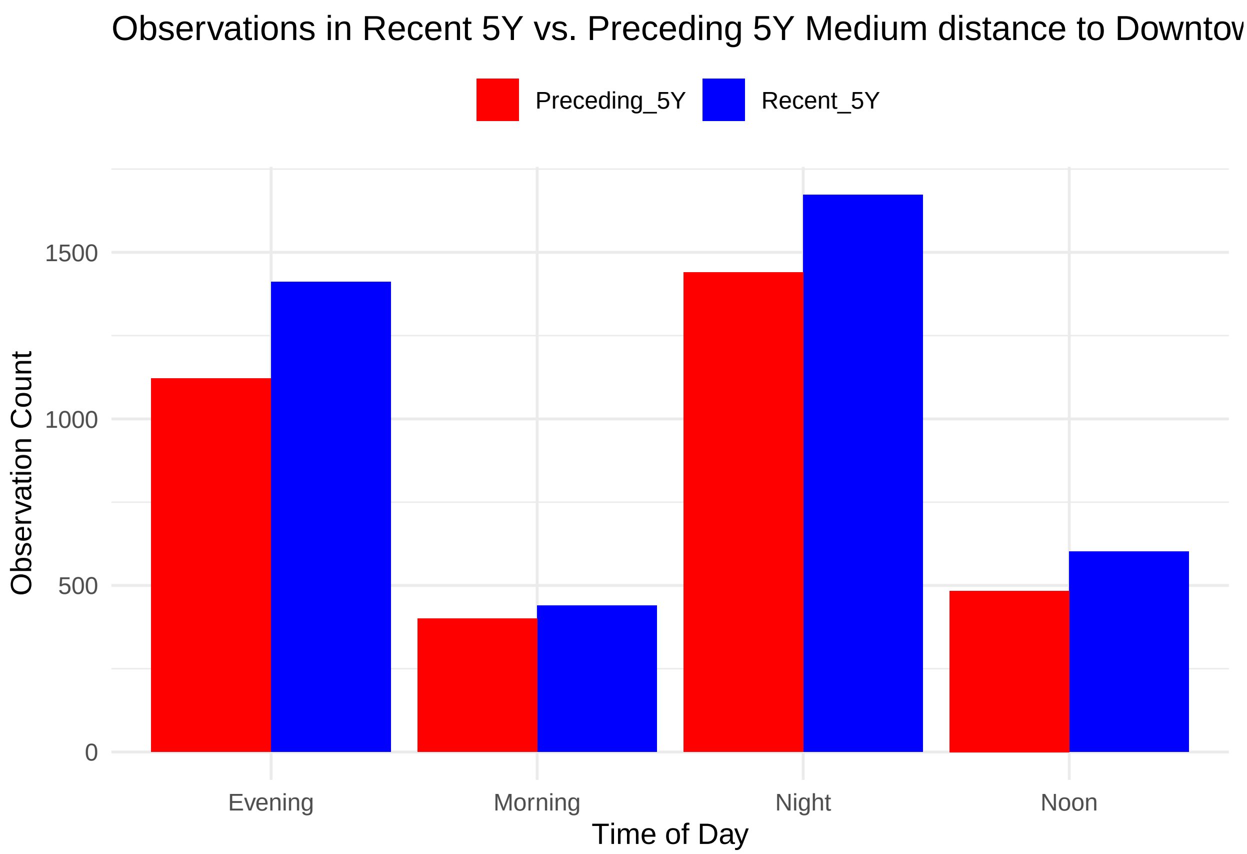
```
## Preceding 5Y Mean: 8.383333
## Z-statistic: 1.297153
## P-value: 0.1945784
##
## Z-test for mean shootings Close to DT per month in the Noon between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 2.766667
## Preceding 5Y Mean: 2.466667
## Z-statistic: 0.7544981
## P-value: 0.4505502
```

Next lets explore the difference in proportions and mean monthly shootings for medium distances to DT.

```r
recent_mid = df %>%
  filter(RECENT_FLAG == 1, DISTANCE == 'Mid') %>%
  group_by(OCCUR_DATE, TIME) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = TIME, values_from = count, values_fill = 0)

preced_mid = df %>%
  filter(RECENT_FLAG == 0, DISTANCE == 'Mid') %>%
  group_by(OCCUR_DATE, TIME) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = TIME, values_from = count, values_fill = 0)
observed_freq_1 = colSums(recent_mid[, -1])
observed_freq_0 = colSums(preced_mid[, -1])
bind = merge(data.frame(observed_freq_1),data.frame(observed_freq_0), by = "row.names")[,-1]
result3 = chisq.test(bind, correct = FALSE)

print(result3)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bind
## X-squared = 4.1592, df = 3, p-value = 0.2448
```

```r
merged_df <- merge(data.frame(observed_freq_1),data.frame(observed_freq_0), by = "row.names")
names(merged_df) <- c("Time_of_Day", "Recent_5Y", "Preceding_5Y")
merged_df = tidyr::pivot_longer(merged_df, cols = c(`Recent_5Y`, `Preceding_5Y`), names_to = "Category"
merged_df$Time_of_Day <- as.character(merged_df$Time_of_Day)
ggplot(merged_df, aes(x = Time_of_Day, y = Observation, fill = Category)) +
  geom_col(position = "dodge") +
  labs(x = "Time of Day", y = "Observation Count", fill = NULL) +
  ggtitle("Observations in Recent 5Y vs. Preceding 5Y Medium distance to Downtown") +
  theme_minimal() +
  theme(legend.position = "top") +
  scale_fill_manual(values = c("Recent_5Y" = "blue", "Preceding_5Y" = "red"))
```




```
for (time in colnames(recent_mid[, -1])) {
  z_stat = z.test(recent_mid[[time]], preced_mid[[time]], sigma.x = sd(recent_mid[[time]]), sigma.y = sd
  p_value = z.test(recent_mid[[time]], preced_mid[[time]], sigma.x = sd(recent_mid[[time]]), sigma.y = s

  cat("\nZ-test for mean shootings Medium distances to DT per month in the", time, "between recent 5Y an
  cat("Recent 5Y Mean:", mean(recent_close[[time]]), "\n")
  cat("Preceding 5Y Mean:", mean(preced_close[[time]]), "\n")
  cat("Z-statistic:", z_stat, "\n")
  cat("P-value:", p_value, "\n")
}
```

```
##
## Z-test for mean shootings Medium distances to DT per month in the Evening between recent 5Y and Prec
## Recent 5Y Mean: 6.616667
## Preceding 5Y Mean: 5.9
## Z-statistic: 2.57296
## P-value: 0.0100833
##
## Z-test for mean shootings Medium distances to DT per month in the Morning between recent 5Y and Prec
## Recent 5Y Mean: 2.466667
## Preceding 5Y Mean: 2.65
## Z-statistic: 0.8767723
## P-value: 0.3806103
##
## Z-test for mean shootings Medium distances to DT per month in the Night between recent 5Y and Preced
## Recent 5Y Mean: 9.866667
```

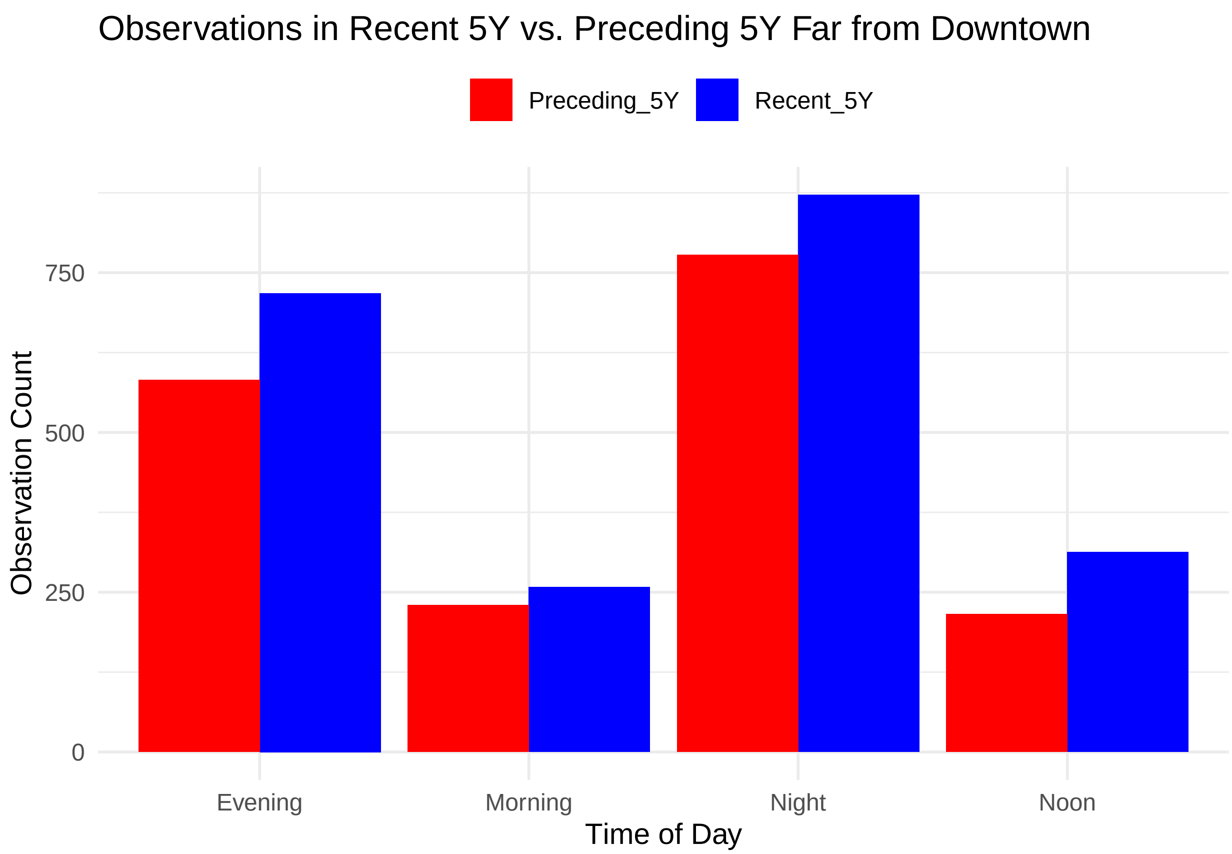
```
## Preceding 5Y Mean: 8.383333
## Z-statistic: 1.296314
## P-value: 0.1948674
##
## Z-test for mean shootings Medium distances to DT per month in the Noon between recent 5Y and Precedin
## Recent 5Y Mean: 2.766667
## Preceding 5Y Mean: 2.466667
## Z-statistic: 2.318254
## P-value: 0.0204355
```

Finally, lets explore the difference in proportions and mean monthly shootings far from DT.

```r
recent_far = df %>%
  filter(RECENT_FLAG == 1, DISTANCE == 'Far') %>%
  group_by(OCCUR_DATE, TIME) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = TIME, values_from = count, values_fill = 0)

preced_far = df %>%
  filter(RECENT_FLAG == 0, DISTANCE == 'Far') %>%
  group_by(OCCUR_DATE, TIME) %>%
  summarise(count = n(), .groups = 'drop') %>%
  pivot_wider(names_from = TIME, values_from = count, values_fill = 0)
observed_freq_1 = colSums(recent_far[, -1])
observed_freq_0 = colSums(preced_far[, -1])
bind = merge(data.frame(observed_freq_1),data.frame(observed_freq_0), by = "row.names")[,-1]
result4 = chisq.test(bind, correct = FALSE)

print(result4)
```

```
##
##  Pearson's Chi-squared test
##
## data:  bind
## X-squared = 7.2656, df = 3, p-value = 0.0639
```

```r
merged_df <- merge(data.frame(observed_freq_1),data.frame(observed_freq_0), by = "row.names")
names(merged_df) <- c("Time_of_Day", "Recent_5Y", "Preceding_5Y")
merged_df = tidyr::pivot_longer(merged_df, cols = c(`Recent_5Y`, `Preceding_5Y`), names_to = "Category"
merged_df$Time_of_Day <- as.character(merged_df$Time_of_Day)
ggplot(merged_df, aes(x = Time_of_Day, y = Observation, fill = Category)) +
  geom_col(position = "dodge") +
  labs(x = "Time of Day", y = "Observation Count", fill = NULL) +
  ggtitle("Observations in Recent 5Y vs. Preceding 5Y Far from Downtown") +
  theme_minimal() +
  theme(legend.position = "top") +
  scale_fill_manual(values = c("Recent_5Y" = "blue", "Preceding_5Y" = "red"))
```




```
for (time in colnames(recent_far[, -1])) {
  z_stat = z.test(recent_far[[time]], preced_far[[time]], sigma.x = sd(recent_far[[time]]), sigma.y = sd
  p_value = z.test(recent_far[[time]], preced_far[[time]], sigma.x = sd(recent_far[[time]]), sigma.y = s

  cat("\nZ-test for mean shootings Far from DT per month in the", time, "between recent 5Y and Preceding
  cat("Recent 5Y Mean:", mean(recent_far[[time]]), "\n")
  cat("Preceding 5Y Mean:", mean(preced_far[[time]]), "\n")
  cat("Z-statistic:", z_stat, "\n")
  cat("P-value:", p_value, "\n")
}
```

```
##
## Z-test for mean shootings Far from DT per month in the Evening between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 11.96667
## Preceding 5Y Mean: 9.7
## Z-statistic: 2.037632
## P-value: 0.04158673
##
## Z-test for mean shootings Far from DT per month in the Morning between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 4.3
## Preceding 5Y Mean: 3.833333
## Z-statistic: 0.8508386
## P-value: 0.394859
##
## Z-test for mean shootings Far from DT per month in the Night between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 14.53333
```

```
## Preceding 5Y Mean: 12.96667
## Z-statistic: 1.027562
## P-value: 0.3041557
##
## Z-test for mean shootings Far from DT per month in the Noon between recent 5Y and Preceding 5Y:
## Recent 5Y Mean: 5.216667
## Preceding 5Y Mean: 3.6
## Z-statistic: 2.63102
## P-value: 0.008512911
```

Conclusion & Bias

It's crucial to acknowledge that biases, including recency bias influenced by media reporting, and potential biases in data collection from sources such as the NYPD, may impact this analysis. My personal bias was that there were no significant changes in NYC shootings over recent periods compared to the past. Due to the volume of recent reporting and the impact of social media, I believed that the trends in shooting remained the same, with recency bias being the main driver. I took deliberate steps to mitigate these biases by examining trends over a 10-year period, split into two 5-year blocks, and adopting a structured analytic approach.

The observed association between the time of day and shooting volume underscores the presence of a strong cyclical pattern, indicating temporal dynamics in gun violence. The OLS multivariate linear regression model had an R score of 93.5% over the last 5 years and 90.5% over the preceding 5 years, showing stronger temporal relationships in recent years. The larger intercept of the model using recent data shows a higher number of shootings overall.

Additionally, statistically significant shifts in the proportions of shooting victims looking at NYC as a whole, suggests the need for further investigation into the temporal and spatial relationships between frequency of shootings. However, when studying the proportions within close, medium, and far shootings, the statistical significance disappears. Aggregating shootings from all geographies could leads to a larger sample size, which in turn may increase the statistical power to detect differences in the proportions. It is important to keep in mind that if the data is not correctly geo-coded with longitude and latitude, it could mislead the results and lead to inconclusive analysis.

Finally, the mean monthly shootings close to DT Manhattan had no statistically significant differences between the recent 5 year and the preceding 5 years. However, shootings medium distances and far away from DT Manhattan saw a statistically significant increase in evening and noon shootings. This raises the potential for further analysis regarding noon shootings further from downtown Manhattan. The differences in shooting volume could be driven by the varying policing and community characteristics in the various geographic area.

Despite the rigorous analysis conducted, it's crucial to recognize the limitations and potential biases inherent in the data and analytic methods employed.

```r
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: x86_64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.6.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;  LAPACK
##
## 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] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] BSDA_1.2.2      lattice_0.21-8  lubridate_1.9.3 forcats_1.0.0
##  [5] stringr_1.5.1   dplyr_1.1.3     purrr_1.0.2     readr_2.1.5
##  [9] tidyr_1.3.1     tibble_3.2.1    ggplot2_3.4.3   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.3        generics_0.1.3    class_7.3-22      stringi_1.8.3
##  [5] hms_1.1.3         digest_0.6.35     magrittr_2.0.3    evaluate_0.23
##  [9] grid_4.3.1        timechange_0.3.0  fastmap_1.1.1     e1071_1.7-14
## [13] fansi_1.0.4       scales_1.2.1      cli_3.6.1         rlang_1.1.1
## [17] crayon_1.5.2      bit64_4.0.5       munsell_0.5.0     withr_2.5.0
## [21] yaml_2.3.8        tools_4.3.1       parallel_4.3.1    tzdb_0.4.0
## [25] colorspace_2.1-0  curl_5.2.1        vctrs_0.6.3       R6_2.5.1
## [29] proxy_0.4-27      lifecycle_1.0.3   bit_4.0.5         vroom_1.6.5
## [33] pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.4      glue_1.6.2
## [37] xfun_0.42         tidyselect_1.2.0  highr_0.10        rstudioapi_0.15.0
## [41] knitr_1.45        farver_2.1.1      htmltools_0.5.7   rmarkdown_2.26
## [45] labeling_0.4.3    compiler_4.3.1
```