Chicago Crimes Final

Kristin Fesmire

2023-07-27

```
rm(list = ls())

library(stringr)
library(EnvStats)
library(ggpubr)
library(ggplot2)
library(reshape2)

# df <- read.csv('C:/Users/krtfe/Downloads/Crimes_-_2023-Updated.csv')
df <- read.csv("C:/Users/krtfe/Downloads/Crimes_-_2023 (ret. 082023).csv")

cat('Pre-cleaning summary:\n\n')
```

```
## Pre-cleaning summary:
```

```
df %>% summary %>% print
```

```
##        ID           Case.Number           Date              Block
##  Min.   :   27279   Length:150712      Length:150712      Length:150712
##  1st Qu.:12997210   Class :character   Class :character   Class :character
##  Median :13054382   Mode  :character   Mode  :character   Mode  :character
##  Mean   :13021795
##  3rd Qu.:13111126
##  Max.   :13171025
##
##      IUCR           Primary.Type       Description        Location.Description
##  Length:150712      Length:150712      Length:150712      Length:150712
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    Arrest            Domestic             Beat         District
##  Length:150712      Length:150712      Min.   : 111   Min.   : 1.00
##  Class :character   Class :character   1st Qu.: 532   1st Qu.: 5.00
##  Mode  :character   Mode  :character   Median :1031   Median :10.00
##                                        Mean   :1148   Mean   :11.25
##                                        3rd Qu.:1724   3rd Qu.:17.00
##                                        Max.   :2535   Max.   :31.00
##
```

```
##       Ward       Community.Area    FBI.Code          X.Coordinate
##  Min.   : 1.00   Min.   : 1.00   Length:150712      Min.   :1091242
##  1st Qu.: 9.00   1st Qu.:22.00   Class :character   1st Qu.:1153761
##  Median :23.00   Median :32.00   Mode  :character   Median :1167054
##  Mean   :23.03   Mean   :36.53                      Mean   :1165353
##  3rd Qu.:34.00   3rd Qu.:53.00                      3rd Qu.:1176910
##  Max.   :50.00   Max.   :77.00                      Max.   :1205114
##  NA's   :3                                          NA's   :7489
##   Y.Coordinate          Year       Updated.On          Latitude
##  Min.   :1813897   Min.   :2023   Length:150712      Min.   :41.65
##  1st Qu.:1859491   1st Qu.:2023   Class :character   1st Qu.:41.77
##  Median :1892280   Median :2023   Mode  :character   Median :41.86
##  Mean   :1886838   Mean   :2023                      Mean   :41.84
##  3rd Qu.:1910139   3rd Qu.:2023                      3rd Qu.:41.91
##  Max.   :1951503   Max.   :2023                      Max.   :42.02
##  NA's   :7489                                        NA's   :7489
##    Longitude        Location
##  Min.   :-87.94   Length:150712
##  1st Qu.:-87.71   Class :character
##  Median :-87.66   Mode  :character
##  Mean   :-87.67
##  3rd Qu.:-87.63
##  Max.   :-87.53
##  NA's   :7489
```

```r
# remove duplicate rows, removed 0 rows
df <- dplyr::distinct(df)

# remove duplicate rows by case number, removed 12 rows, from 150712 to 150700
df <- df[!duplicated(df$Case.Number),]

# simplify data, remove columns that aren't useful for current project
df <- df[c(3, 6, 7:10, 12:14)]

# removed columns  so the data could be imported to github
# write.csv(df, 'C:/Users/krtfe/Downloads/Crimes_-_2023-8-20.csv')

# remove rows with NA values, removed 3 rows, 150700 rows -> 150679 rows
df <- na.omit(df)

# adding useful columns, dates, times, time of day
dates <- str_split(df$Date, pattern = ' ', simplify = TRUE)[,1]
times <- str_split(df$Date, pattern = ' ', simplify = TRUE)[,2]
time_of_day <- str_split(df$Date, pattern = ' ', simplify = TRUE)[,3]

# add the useful columns and transform data types
df['Date'] <- as.Date(dates, format = '%m/%d/%Y')
df['Time'] <- times
df['Time of Day'] <- time_of_day

# set dataframe such that it only includes months from january to july
# from 150679 rows -> 147596
df <- df[df$Date < lubridate::ymd("2023-08-01"),]
```

```r
cat('\n\nPost-cleaning summary:\n\n')
```

```
##
##
## Post-cleaning summary:
```

```r
df %>% summary %>% print
```

```
##       Date             Primary.Type       Description
##  Min.   :2023-01-01   Length:147596      Length:147596
##  1st Qu.:2023-02-25   Class :character   Class :character
##  Median :2023-04-21   Mode  :character   Mode  :character
##  Mean   :2023-04-19
##  3rd Qu.:2023-06-12
##  Max.   :2023-07-31
##  Location.Description    Arrest            Domestic            District
##  Length:147596        Length:147596      Length:147596      Min.   : 1.00
##  Class :character     Class :character   Class :character   1st Qu.: 5.00
##  Mode  :character     Mode  :character   Mode  :character   Median :10.00
##                                                             Mean   :11.25
##                                                             3rd Qu.:17.00
##                                                             Max.   :31.00
##       Ward       Community.Area      Time           Time of Day
##  Min.   : 1.00   Min.   : 1.00   Length:147596      Length:147596
##  1st Qu.: 9.00   1st Qu.:22.00   Class :character   Class :character
##  Median :23.00   Median :32.00   Mode  :character   Mode  :character
##  Mean   :23.03   Mean   :36.54
##  3rd Qu.:34.00   3rd Qu.:53.00
##  Max.   :50.00   Max.   :77.00
```

```r
# data frame representation
df %>% head
```

```
##         Date    Primary.Type         Description Location.Description Arrest
## 1 2023-06-28        HOMICIDE FIRST DEGREE MURDER                ALLEY   true
## 2 2023-06-29        HOMICIDE FIRST DEGREE MURDER               STREET  false
## 3 2023-03-30 CRIMINAL DAMAGE         TO PROPERTY          GAS STATION  false
## 4 2023-03-07           THEFT       FROM BUILDING            RESIDENCE  false
## 5 2023-06-29        HOMICIDE FIRST DEGREE MURDER               STREET  false
## 6 2023-06-29        HOMICIDE FIRST DEGREE MURDER               STREET  false
##   Domestic District Ward Community.Area     Time Time of Day
## 1    false       17   33             16 11:04:00          PM
## 2    false        7    6             68 07:40:00          PM
## 3    false        1    4             32 02:16:00          PM
## 4    false        3   20             42 10:57:00          AM
## 5    false        8   14             57 07:00:00          AM
## 6    false        7   16             67 04:39:00          PM
```

Separate data frame for (specific variable) counts by dates:

```
# second data frame, number of crimes

# start with the unique dates and their counts
numCrimes <- table(df$Date)
dfCounts <- data.frame(numCrimes)
colnames(dfCounts) <- c('Date', 'Number of Crimes')

# make row names the dates, for convenience
row.names(dfCounts) <- dfCounts$Date

# add a count by each date for domestic crimes
for (i in dfCounts$Date) {
  dfCounts[i, 'Domestic'] <- sum((df$Date == i & df$Domestic == 'true'))
}

# add a count by each date for crimes with arrests
for (i in dfCounts$Date) {
  dfCounts[i, 'Arrest'] <- sum((df$Date == i & df$Arrest == 'true'))
}

# table of main types of crimes
tabType <- table(df$Primary.Type)

# top types of primary types of crimes
topTypes = sort(tabType, decreasing = TRUE)[c(1:6, 8:11)]
ttLabels = labels(topTypes)[[1]]

# columns for the counts for the top types of primary types of crimes
for (j in ttLabels) {
  for (i in dfCounts$Date) {
    dfCounts[i, j] <- sum((df$Date == i & df$Primary.Type == j))
  }
}

# renaming column names for consistency
colnames(dfCounts) <- str_to_title(colnames(dfCounts))
ttLabels = str_to_title(ttLabels)

# printing the counts dataset
dfCounts %>% head
```

```
##                  Date Number Of Crimes Domestic Arrest Theft Battery
## 2023-01-01 2023-01-01              970      237    115   124     206
## 2023-01-02 2023-01-02              649      134     77   110     103
## 2023-01-03 2023-01-03              733       97     67   144      92
## 2023-01-04 2023-01-04              680      107     84   148      81
## 2023-01-05 2023-01-05              654      110     83   141      92
## 2023-01-06 2023-01-06              722      113     88   136      87
##            Criminal Damage Motor Vehicle Theft Assault Deceptive Practice
## 2023-01-01             159                  87      91                  63
## 2023-01-02              99                  87      45                  33
```

```
## 2023-01-03             131                  98      52                  54
## 2023-01-04              66                 111      53                  46
## 2023-01-05              75                  89      37                  48
## 2023-01-06              90                  88      51                  62
##            Robbery Weapons Violation Burglary Narcotics
## 2023-01-01      25                63       20        11
## 2023-01-02      28                32       15        17
## 2023-01-03      32                24       31        16
## 2023-01-04      37                26       13        19
## 2023-01-05      36                26       12        15
## 2023-01-06      41                26       33        17
```

EDA by arrests, domestic crimes, and general crimes:

```
dateCounts <- table(df$Date)
meanD <- sum(dateCounts)/length(dateCounts)
variance <- sum((meanD - dateCounts)^2)/length(dateCounts)

# output variance and mean information about the crime counts
cat(paste('Number of Crimes by Day:',
          '\n\tMean = ', round(meanD, 5),
          '\n\tVariance = ', round(var(dateCounts), 5)))
```

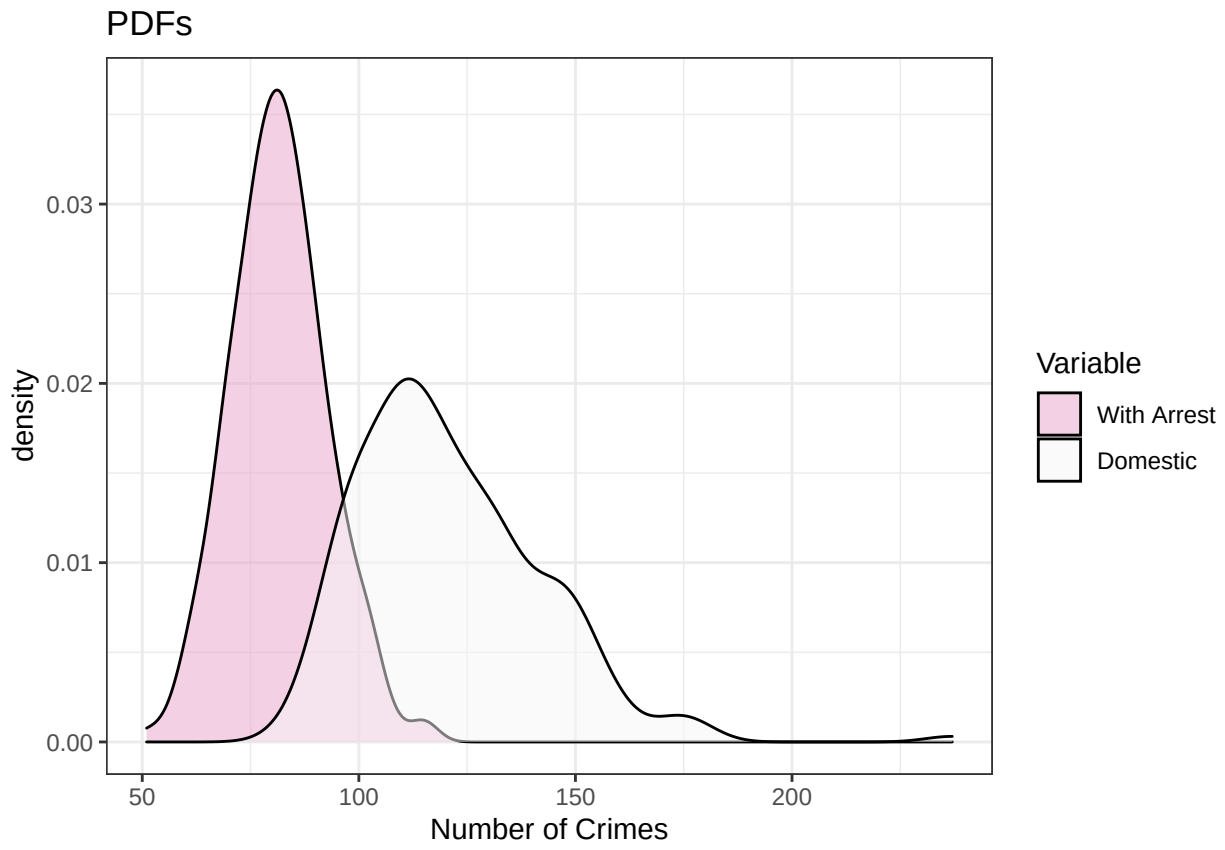
```
## Number of Crimes by Day:
##  Mean =  696.20755
##  Variance =  3505.71027
```

```
# outputting variance and mean information
for (i in c(colnames(dfCounts)[3:length(colnames(dfCounts))])) {
  meanCounts <- round(mean(unlist(dfCounts[i])), 5)
  varCounts <- round(var(unlist(dfCounts[i])), 5)

  if (i %in% c('Domestic', 'Criminal Damage', 'Battery')) {
    cat(paste('\n\nNumber of ', i, ' Crimes by Day:',
              '\n\tMean = ', meanCounts,
              '\n\tVariance = ', varCounts,
              sep = ''))
  }
  else {
    cat(paste('\n\nNumber of ', i, 's by Day:',
              '\n\tMean = ', meanCounts,
              '\n\tVariance = ', varCounts,
              sep = ''))
  }

}
```

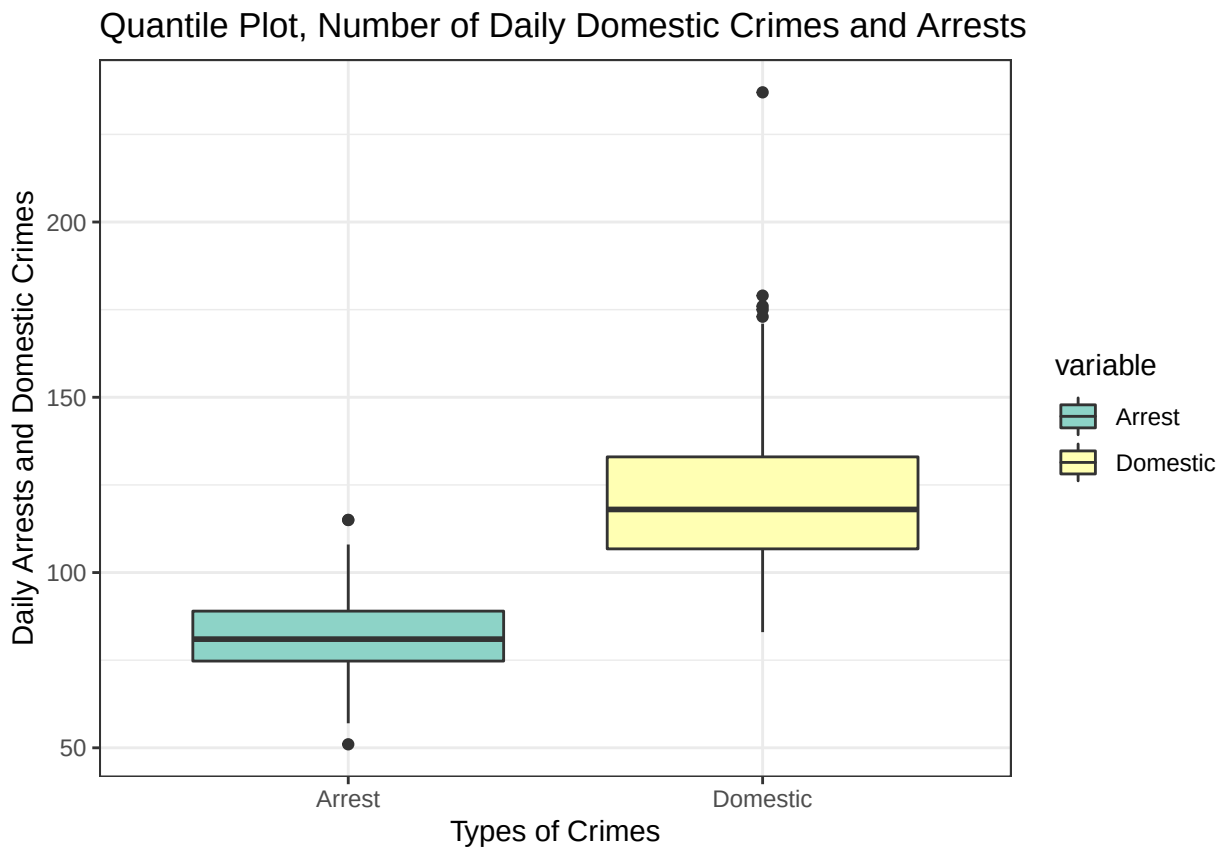
```
##
##
## Number of Domestic Crimes by Day:
##  Mean = 121.0283
##  Variance = 454.54896
##
## Number of Arrests by Day:
##  Mean = 81.60377
##  Variance = 121.77591
##
## Number of Thefts by Day:
##  Mean = 148.75472
##  Variance = 405.17178
##
## Number of Battery Crimes by Day:
##  Mean = 118.88208
##  Variance = 486.08555
##
## Number of Criminal Damage Crimes by Day:
##  Mean = 81.23585
##  Variance = 286.98677
##
## Number of Motor Vehicle Thefts by Day:
##  Mean = 80.92925
##  Variance = 192.6727
##
## Number of Assaults by Day:
##  Mean = 60.41509
##  Variance = 104.18707
##
## Number of Deceptive Practices by Day:
##  Mean = 43.42925
##  Variance = 163.90492
##
## Number of Robberys by Day:
##  Mean = 25.81604
##  Variance = 72.42571
##
## Number of Weapons Violations by Day:
##  Mean = 24.43868
##  Variance = 58.93935
##
## Number of Burglarys by Day:
##  Mean = 19.84906
##  Variance = 28.52687
##
## Number of Narcoticss by Day:
##  Mean = 13.23113
##  Variance = 20.16907
```

```
meltedDens <- melt(dfCounts[c('Arrest',
                              'Domestic')])

# pdf for the arrest and domestic count columns
```

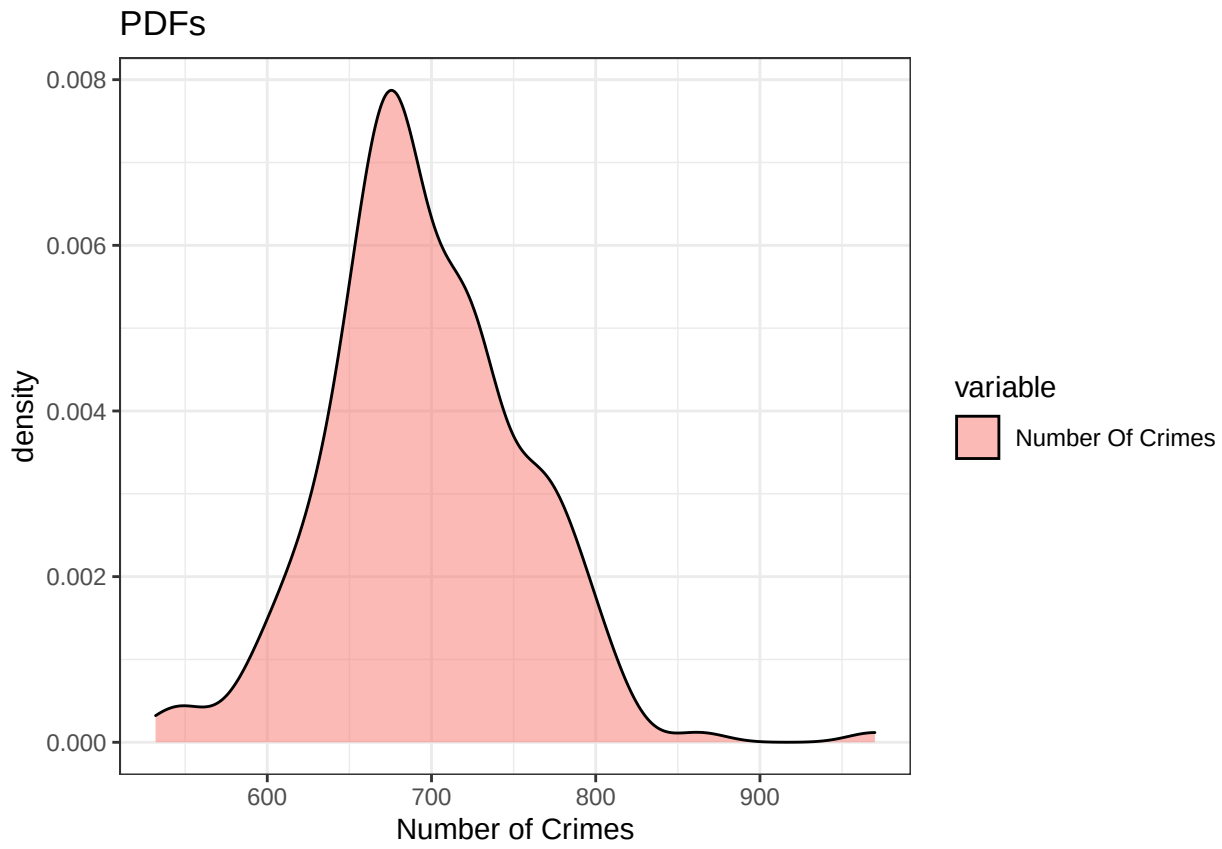
```
ggplot(meltedDens, aes(x = value, fill = variable)) +
  geom_density(alpha = 0.5, adjust = 1) +
  # xlim(c(0, 50)) +
  xlab('Number of Crimes') +
  scale_fill_brewer('Variable', palette = 'PiYG',
                    labels = c('With Arrest',
                               'Domestic')) +
  theme_bw() +
  ggtitle('PDFs')
```




```
# boxplot for the arrest and domestic count columns
aplot <- ggplot(meltedDens,
       aes(x = variable, y = value, fill = variable),
       ) + geom_boxplot()

aplot +
  scale_fill_brewer(palette="Set3") +
  ylab('Daily Arrests and Domestic Crimes') +
  ggtitle('Quantile Plot, Number of Daily Domestic Crimes and Arrests') +
  scale_x_discrete(name = 'Types of Crimes',
                   limits = c('Arrest', 'Domestic') ) +
  theme_bw()
```

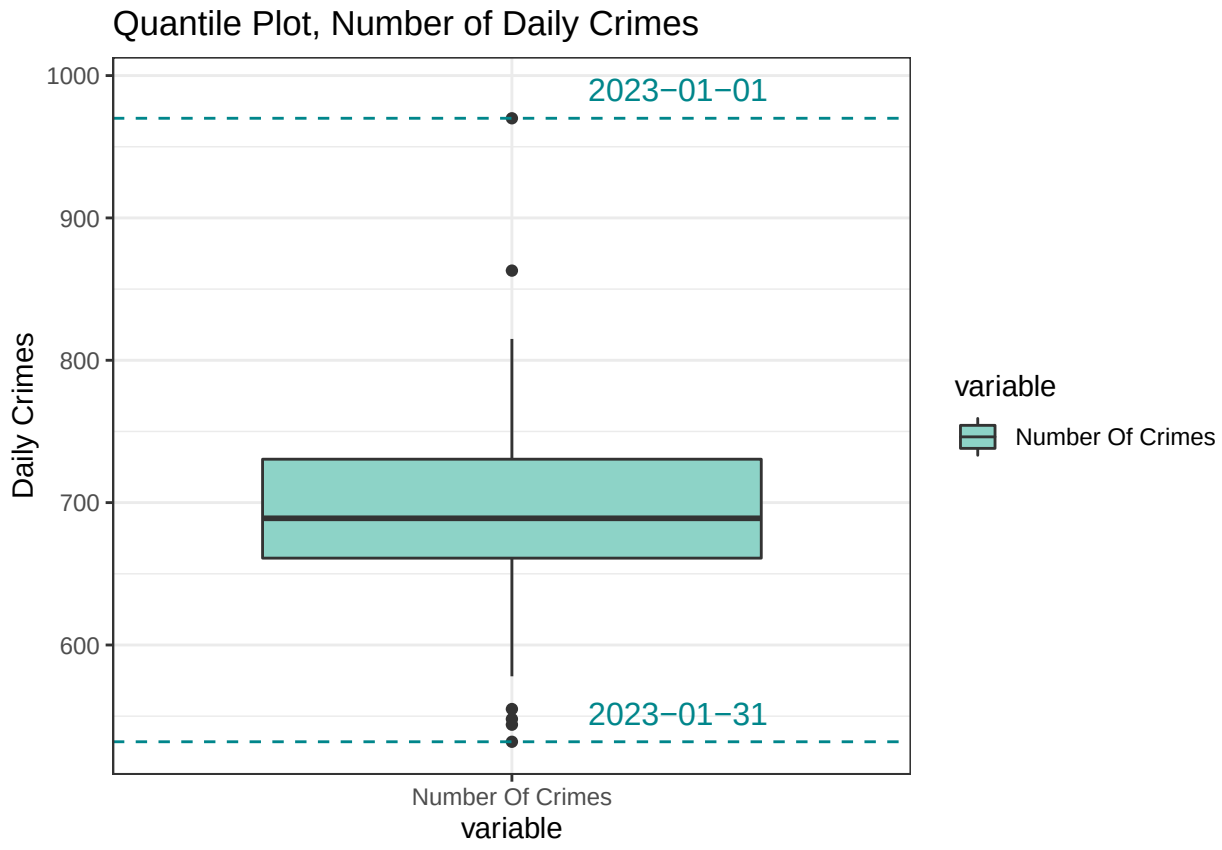



```
# pdf for the number of crimes counts
meltedDens <- melt(dfCounts[c('Number Of Crimes')])
ggplot(meltedDens, aes(x = value, fill = variable)) +
  geom_density(alpha = 0.5, adjust = 1) +
  xlab('Number of Crimes') +
  theme_bw() +
  ggtitle('PDFs')
```




```
# boxplot for the number of crimes counts
aplot <- ggplot(meltedDens,
       aes(x = variable, y = value, fill = variable),
       ) + geom_boxplot()

aplot +
  scale_fill_brewer(palette="Set3") +
  ylab('Daily Crimes') +
  ggtitle('Quantile Plot, Number of Daily Crimes') +
  geom_hline(yintercept = max(dfCounts$`Number Of Crimes`),
             linetype = 'dashed',
             color = 'turquoise4') +
  annotate(geom = 'text',
           label = dfCounts[dfCounts$`Number Of Crimes` ==
                              max(dfCounts$`Number Of Crimes`), ]$Date,
           size = 4.2,
           color = 'turquoise4',
           x = 1.25,
           y = max(dfCounts$`Number Of Crimes`) + 20) +
  geom_hline(yintercept = min(dfCounts$`Number Of Crimes`),
             linetype = 'dashed',
             color = 'turquoise4') +
  annotate(geom = 'text',
           label = dfCounts[dfCounts$`Number Of Crimes` ==
                              min(dfCounts$`Number Of Crimes`), ]$Date,
           size = 4.2,
```

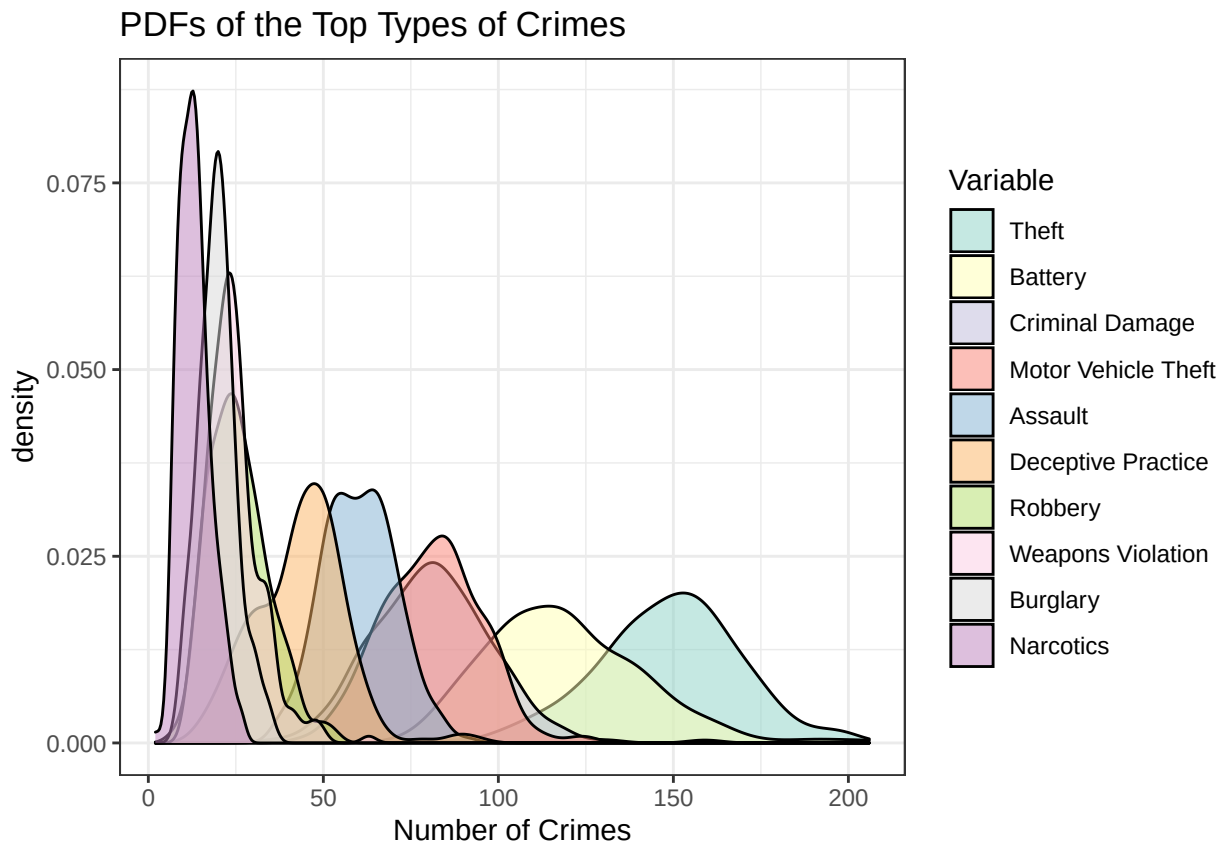
```
          color = 'turquoise4',
          x = 1.25,
          y = min(dfCounts$`Number Of Crimes`) + 20) +
  # scale_x_discrete(name = 'Types of Crimes',
  #                  limits = c('Domestic', 'Arrest') ) +
  theme_bw()
```



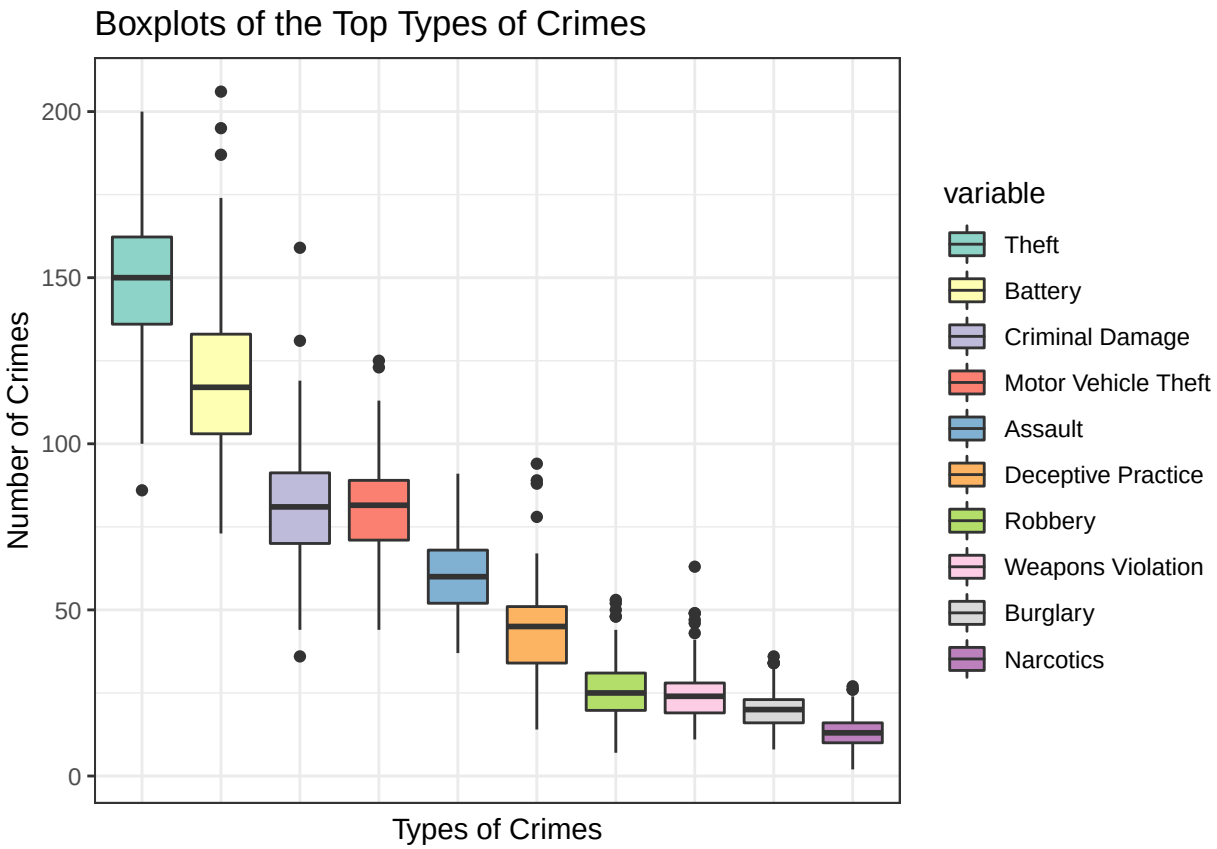
Visualizations for the most common types of crimes in the dataset:

```
meltedDens <- melt(dfCounts[ttLabels])

# pdf for the most common types of crimes
ggplot(meltedDens, aes(x = value, fill = variable)) +
  geom_density(alpha = 0.5, adjust = 1) +
  # xlim(c(0, 50)) +
  xlab('Number of Crimes') +
  scale_fill_brewer('Variable',
                    palette = 'Set3',
                    labels = c(ttLabels)) +
```

```
  theme_bw() +
  ggtitle('PDFs of the Top Types of Crimes')
```

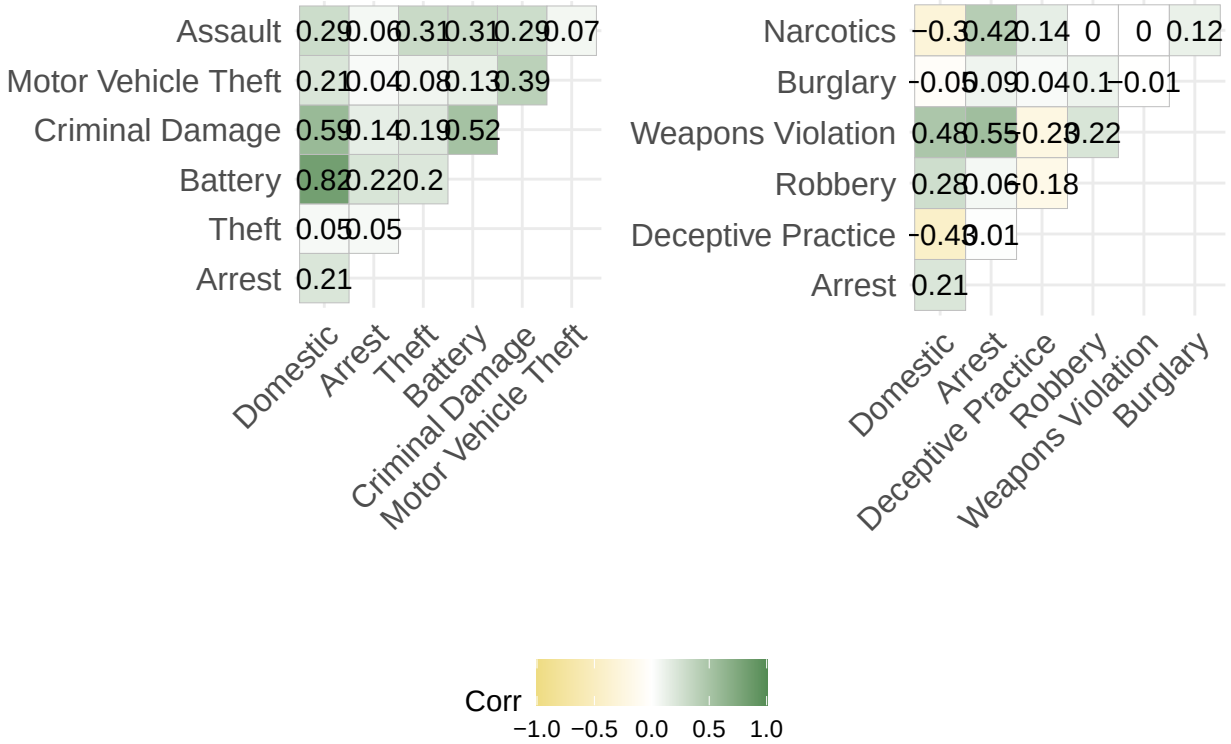



```
# boxplot for the most common types of crimes
aplot <- ggplot(meltedDens,
       aes(x = variable, y = value, fill = variable),
       ) + geom_boxplot()

aplot +
  scale_fill_brewer(palette="Set3") +
  ylab('Number of Crimes') +
  ggtitle('Boxplots of the Top Types of Crimes') +
  scale_x_discrete(name = 'Types of Crimes',
                   limits = c(ttLabels) ) +
  theme_bw() +
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank())
```



Correlation plot for the counts dataset

```
# split correlation plots for better visualization
corrCounts <- dfCounts[3:9] %>% cor(method = 'pearson')
corr1 <- corrCounts %>% ggcorrplot::ggcorrplot(lab = TRUE, type = 'upper',
                                    colors = c('lightgoldenrod2', 'white',
                                               'palegreen4'))

corrCounts2 <- dfCounts[c(3:4, 10:14)] %>% cor(method = 'pearson')
corr2 <- corrCounts2 %>% ggcorrplot::ggcorrplot(lab = TRUE, type = 'upper',
                                    colors = c('lightgoldenrod2', 'white',
                                               'palegreen4'))

ggarrange(corr1, corr2, common.legend = TRUE, legend = 'bottom')
```



Correlation plots were mostly created to investigate different crimes relationships with domestic crimes and crimes with arrest.

All crimes had positive correlations with arrests. Motor vehicle theft and deceptive practice had the lowest correlations with arrests while weapons violations and narcotics had the highest correlations with arrests.

Whether or not crimes were correlated with domestic crimes depended on the crimes. Battery and criminal damage had the highest correlations with domestic crimes, while narcotics and deceptive practice had the lowest correlations with domestic crimes.

Cleaning for data visualization

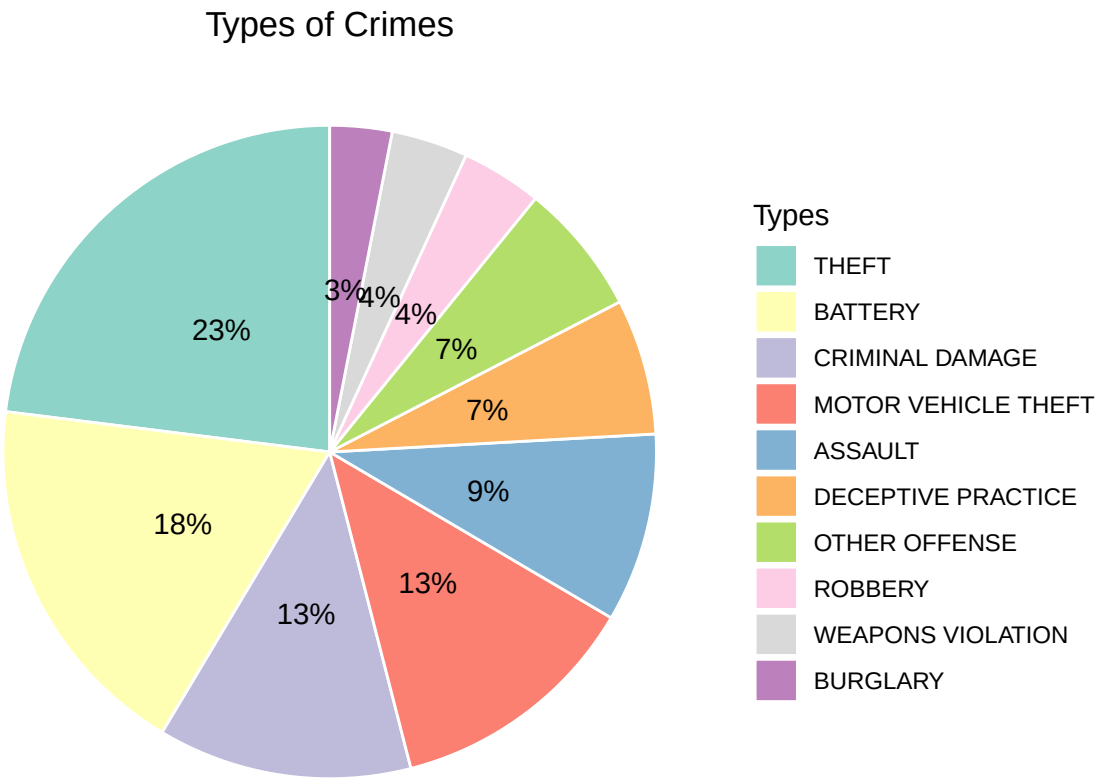
```
# # dataset for each type of primary type of crime, with their frequency counts
stabType <- head(sort(tabType,
                      decreasing = TRUE), 10)
tabTypeDF <- data.frame(stabType)
# tabTypeDF <- data.frame(tabType)
colnames(tabTypeDF) <- c('Types', 'Frequency')

# convert types of crimes from factor to string
tabTypeDF$Types <- sapply(tabTypeDF$Types, toString)
```

```r
# sum other crimes that won't be included in visualization
otherCrimesSum <- sum(tabTypeDF[tabTypeDF$Frequency < stabType[length(stabType)],]$Frequency)

# other crimes variable that is already set
otherCrimesVal <- tabTypeDF[tabTypeDF$Types == 'OTHER OFFENSE',]$Frequency

# Move specific crime types to the "other offense" column for better visualization
tabTypeDF[tabTypeDF$Types == 'OTHER OFFENSE', 2] <- sum(otherCrimesSum,
                                                        otherCrimesVal)

# # sorting the dataset
# tabTypeDF <- tabTypeDF[order(tabTypeDF$Frequency,
#                              decreasing = TRUE),]

# factor strings so that it sorts properly
tabTypeDF$Types <- factor(tabTypeDF$Types, levels = tabTypeDF$Types)
```

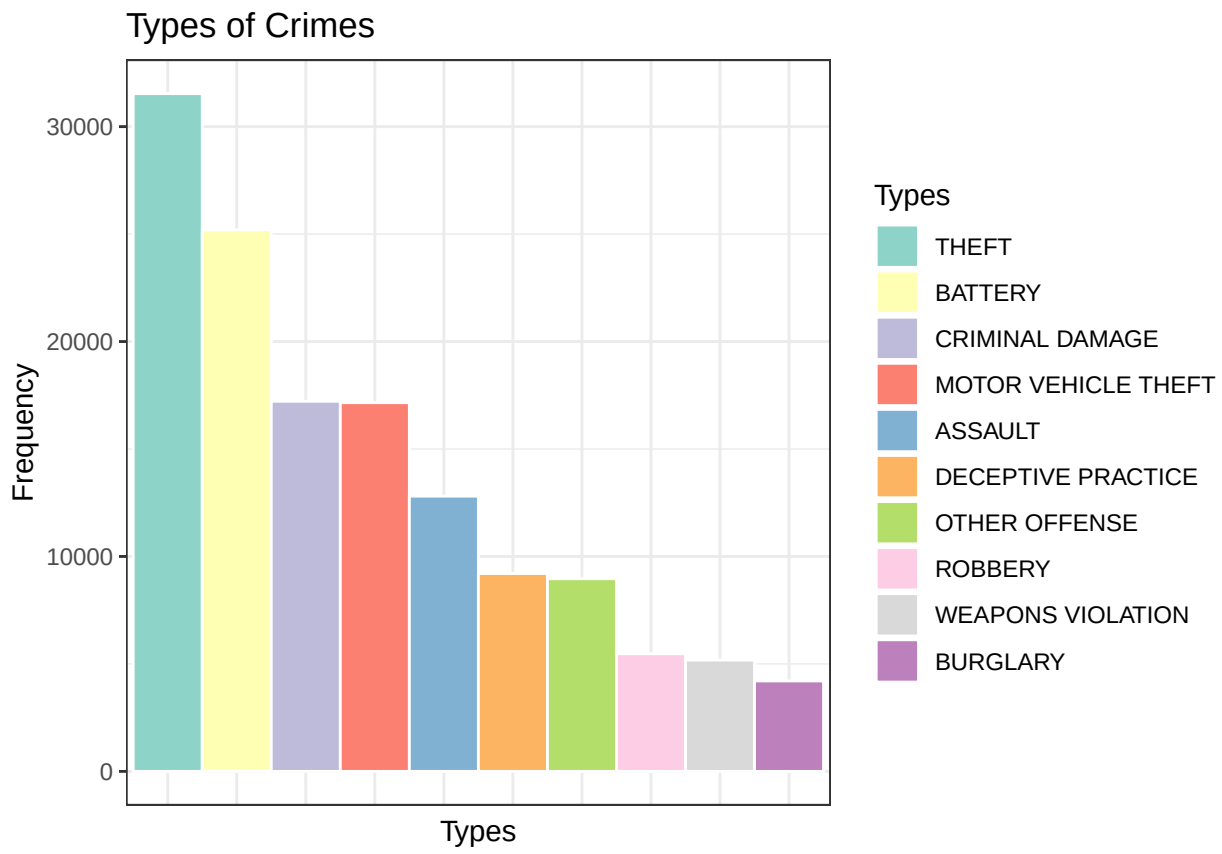
Pie and bar plots

```r
# Pie plot for most common types of crimes commit
pieP <- ggplot(tabTypeDF,
               aes(x="",
                   y=100*Frequency/sum(Frequency),
                   fill=Types)) +
        geom_bar(stat="identity", width=1, color = 'white') +
        coord_polar("y", start=0) +
        theme_void() +
        ggtitle(paste('Types of Crimes')) +
        theme(plot.title = element_text(hjust = 0.5)) +
        geom_text(aes(label = paste0(round(100*Frequency/sum(Frequency)),
                                     "%")),
                  position = position_stack(vjust = 0.5)) +
        scale_fill_brewer('Types', palette = 'Set3')

# Bar plot for the most common types of crimes commit
barP <- ggplot(tabTypeDF,
               aes(x=Types, y=Frequency, fill = Types)) +
        geom_bar(stat="identity", width=1, color = 'white') +
        theme_bw()  +
        theme(axis.text.x = element_blank(),
              axis.ticks.x = element_blank()) +
        ggtitle(paste('Types of Crimes')) +
        scale_fill_brewer('Types', palette = 'Set3',
                          labels = c(tabTypeDF$Types))

# Calculations for below bar plot
numDays <- length(unique(df$Date))
```

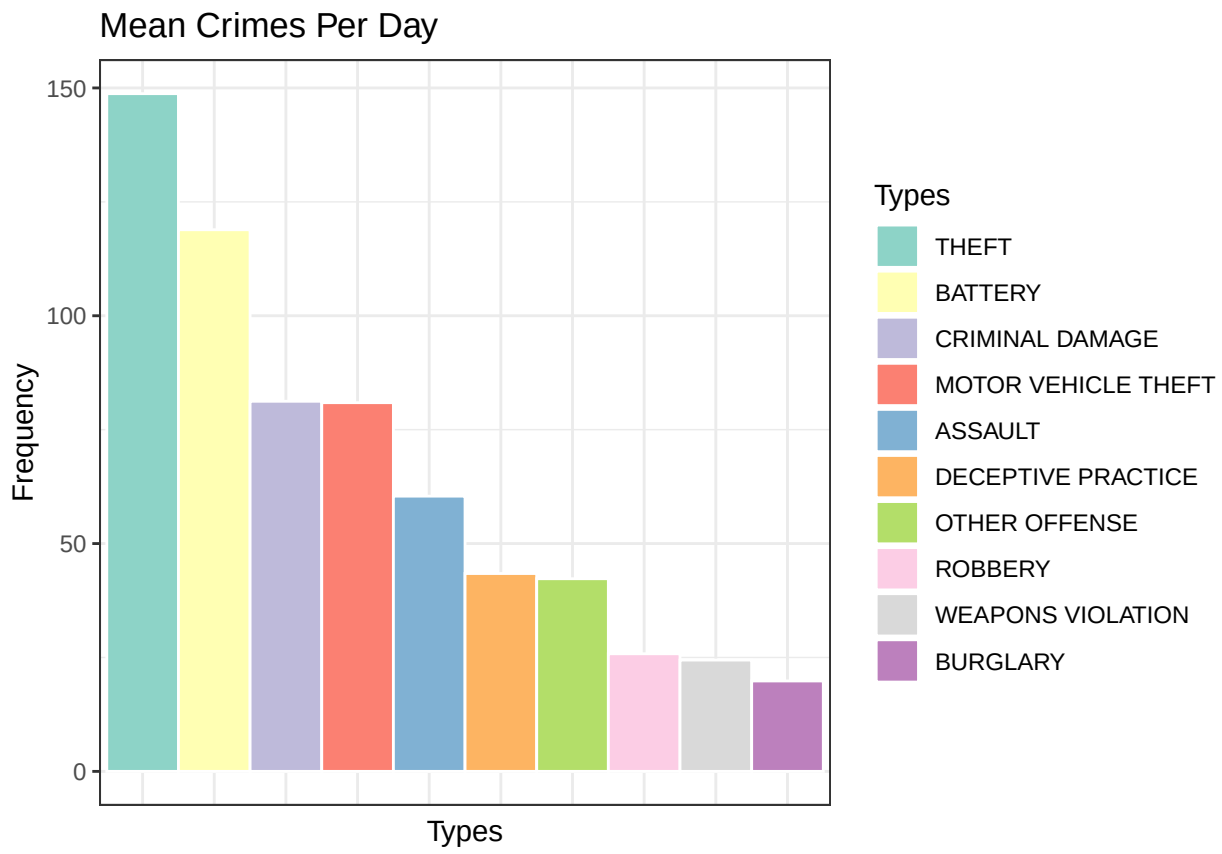
```r
tabTypeDFAvg <- tabTypeDF
tabTypeDFAvg$Frequency <- tabTypeDFAvg$Frequency / numDays

# Bar plot for the most common types of crimes commit, as mean values
barPAvg <- ggplot(tabTypeDFAvg,
                  aes(x=Types, y=Frequency, fill = Types)) +
                  geom_bar(stat="identity", width=1, color = 'white') +
                  theme_bw()  +
                  theme(axis.text.x = element_blank(),
                        axis.ticks.x = element_blank()) +
                  ggtitle(paste('Mean Crimes Per Day')) +
          scale_fill_brewer('Types', palette = 'Set3',
                            labels = c(tabTypeDF$Types))

# visualization of the plots
pieP
```




```r
barP
```


Types of Crimes
30000
20000
10000
0
Frequency
Types
Types
THEFT
BATTERY
CRIMINAL DAMAGE
MOTOR VEHICLE THEFT
ASSAULT
DECEPTIVE PRACTICE
OTHER OFFENSE
ROBBERY
WEAPONS VIOLATION
BURGLARY

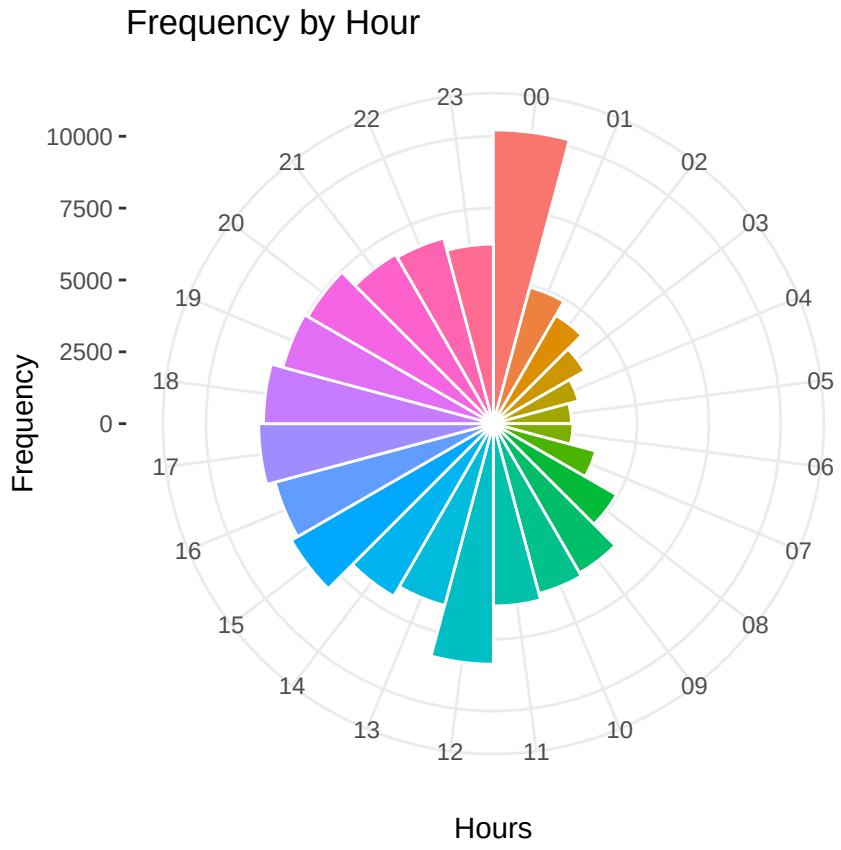
```
barPAvg
```



Clock of when most crimes happened during the day

```
# Getting hour integers for when the time of day is AM
hourTimesAM <- df[df$`Time of Day` == 'AM',]$Time %>%
               substr(start = 1, stop = 2)
# turn 12 AM to 00 AM, for easier viz
hourTimes <- ifelse(hourTimesAM == '12', '00', hourTimesAM)

# Getting hour integers for when the time of day is PM
hourTimesPM <- df[df$`Time of Day` == 'PM',]$Time %>%
                            substr(start = 1, stop = 2) %>%
                            c
# add 12 to all except 12 PM for PM times for easier viz
hourTimes <- c(hourTimes, ifelse(as.integer(hourTimesPM) != 12,
                                 as.integer(hourTimesPM) + 12,
                                 hourTimesPM))

# calculating frequencies for specific hour integers
hourTimesFreq <- table(hourTimes)
hourTimesDF <- data.frame(hourTimesFreq)
colnames(hourTimesDF) <- c('Hours', 'Frequency')
```

```
# visualizing a clock for when crimes most frequently occur
barP <- ggplot(hourTimesDF,
               aes(x=Hours, y=Frequency, fill = Hours)) +
        geom_bar(stat="identity", width=1, color = 'white') +
        coord_polar() +
        theme_bw()  +
        theme(legend.position = 'none',
              panel.border = element_blank()) +
        ggtitle(paste('Frequency by Hour'))
barP
```

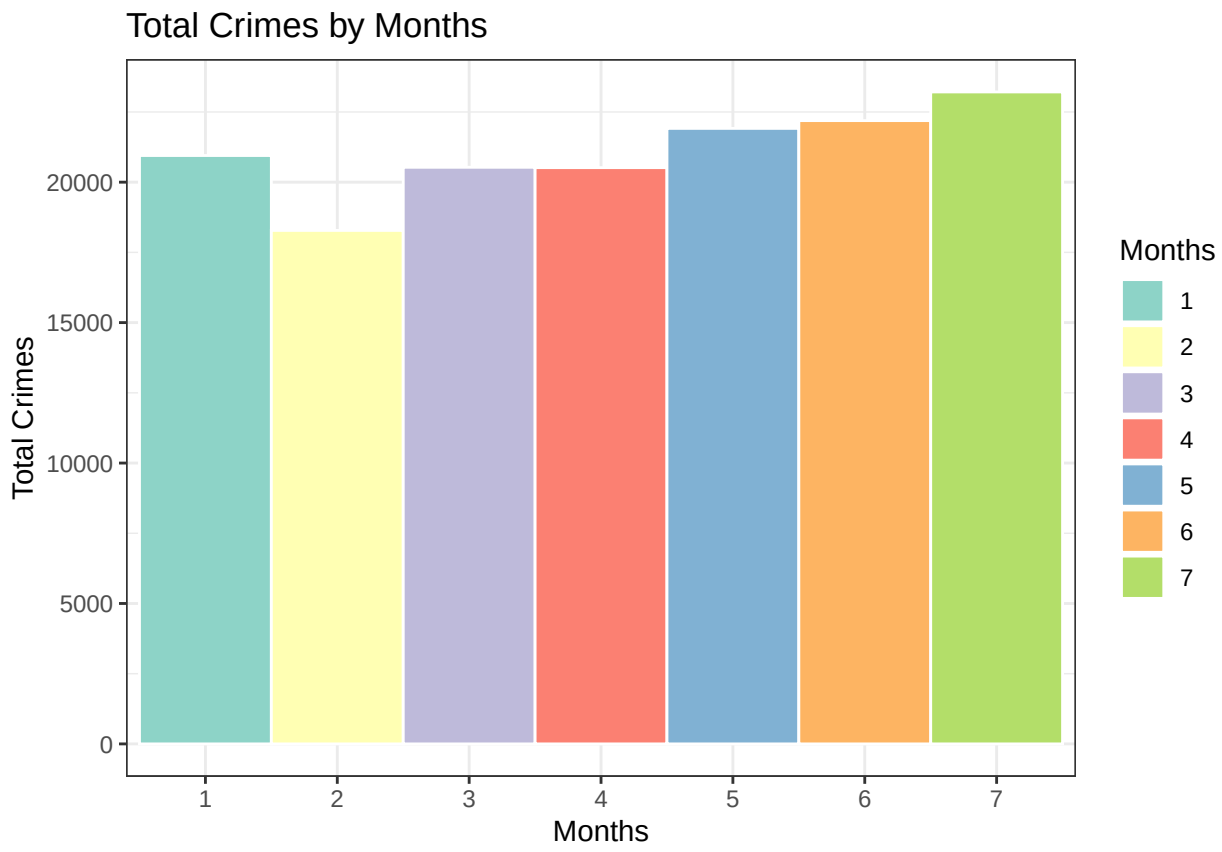


Crimes happened the most at 12 AM. The occurrence of crimes decreased throughout the day when not including 12 AM.

Total crimes by each month

```
# add column just to specify months in the dataframe
dfCounts['Month'] <- dplyr::case_when(grepl('-01-', dfCounts$Date) ~ '1',
                                      grepl('-02-', dfCounts$Date) ~ '2',
                                      grepl('-03-', dfCounts$Date) ~ '3',
                                      grepl('-04-', dfCounts$Date) ~ '4',
                                      grepl('-05-', dfCounts$Date) ~ '5',
```

```
                                    grepl('-06-', dfCounts$Date) ~ '6',
                                    grepl('-07-', dfCounts$Date) ~ '7')

# create dataframe specifically for months, for data viz
dfCountsMonths <- data.frame('Months' = unique(dfCounts$Month),
                             row.names = c(unique(dfCounts$Month)))

# sum total crimes for each month
for (i in unique(dfCounts$Month)) {
    monthsSum <- sum(dfCounts[dfCounts$Month == i,]$`Number Of Crimes`)
    dfCountsMonths[i, 'Total Crimes'] <- monthsSum
}

# bar plot for unique months
dfCountsMonths %>% ggplot(aes(x=Months, y=`Total Crimes`, fill = Months)) +
                           scale_fill_brewer(palette="Set3") +
                           geom_bar(stat="identity", width=1, color = 'white') +
                           theme_bw()  +
                           ggtitle(paste('Total Crimes by Months'))
```



The occurrence of crimes increased throughout the year month by month, except for January.