Chicago_Crimes_Final

Kristin Fesmire

2023-07-27

```r
rm(list = ls())

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

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

# 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 685 rows 132001 -> 131999
df <- na.omit(df)

# remove duplicate rows
df <- dplyr::distinct(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
df <- df[df$Date < lubridate::ymd("2023-08-01"),]


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
```

```r
# 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)[1:5]
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
```

Separate data frame for counts by dates and specific variables

```
##                  Date Number Of Crimes Domestic Arrest Theft Battery
## 2023-01-01 2023-01-01              969      237    115   124     206
## 2023-01-02 2023-01-02              648      134     77   110     103
## 2023-01-03 2023-01-03              730       97     67   143      91
## 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
## 2023-01-01             159                  87      91
## 2023-01-02              99                  87      45
## 2023-01-03             130                  98      52
## 2023-01-04              66                 111      53
## 2023-01-05              75                  89      37
## 2023-01-06              90                  88      51
```

```


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)))
```

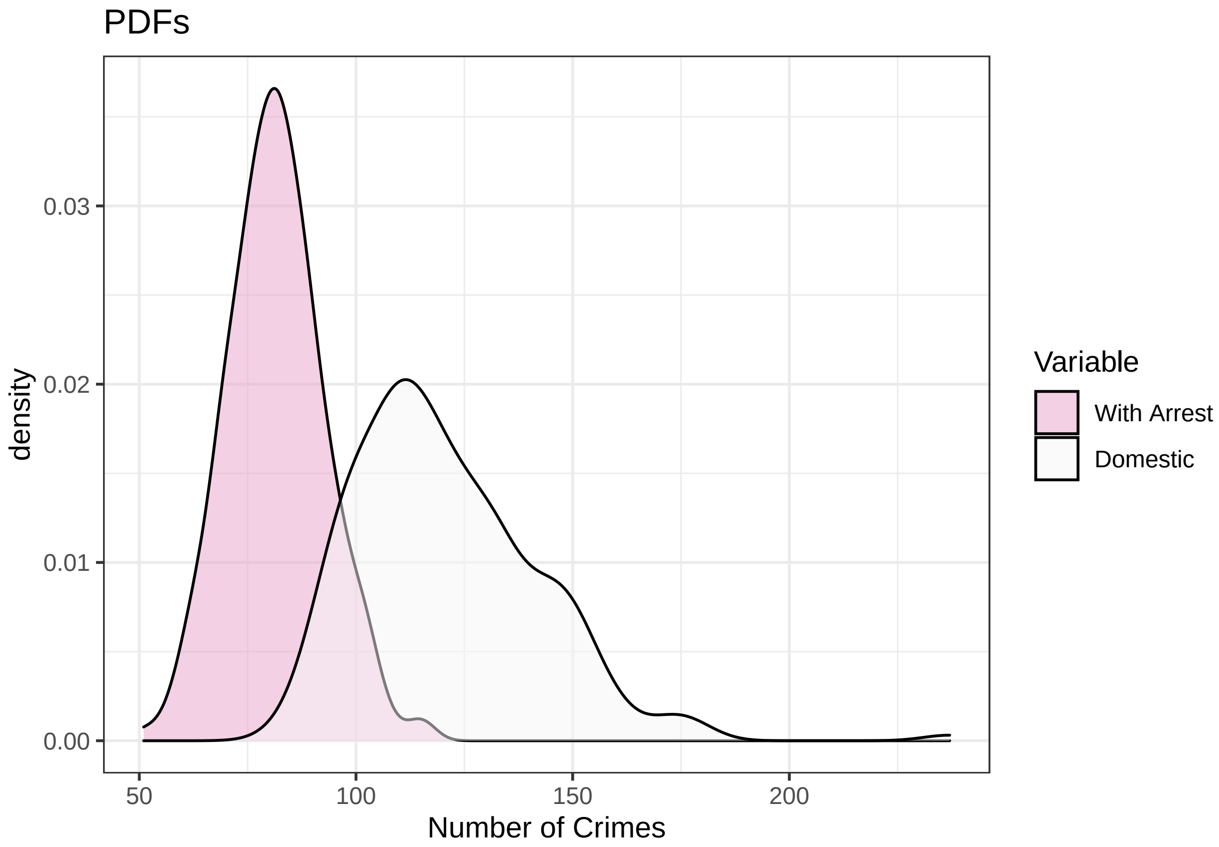
EDA by arrests, domestic, crime, ward, community area

```
## Number of Crimes by Day:
##  Mean =  694.24528
##  Variance =  3436.94429
```

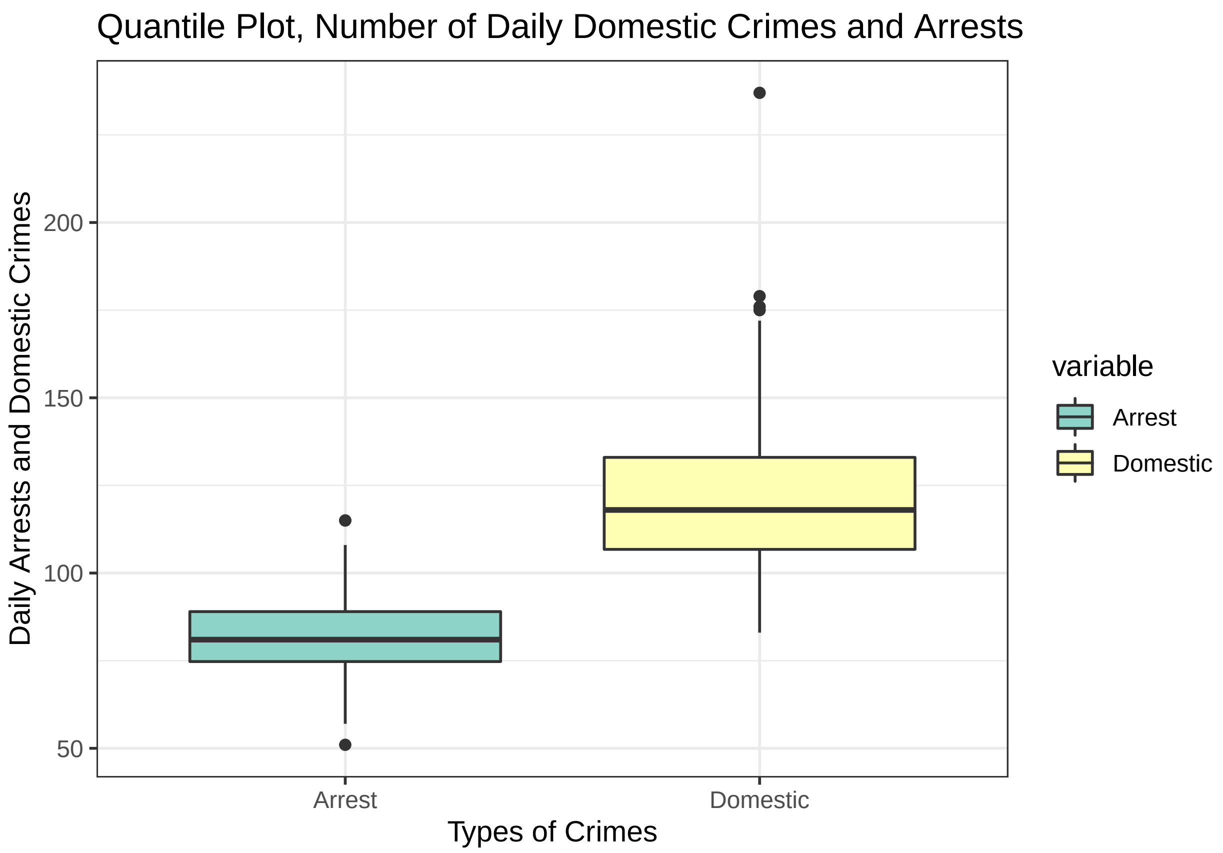
```
# outputting variance and mean information
for (i in c(colnames(dfCounts)[3:length(colnames(dfCounts))])) {
  meanCounts <- round(mean(unlist(dfCounts[i])), 5)
  varCounts <- round(var(unlist(dfCounts[i])), 5)

  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 = 120.97642
##  Variance = 454.16532
##
## Number of Arrests by Day:
##  Mean = 81.58019
##  Variance = 121.25894
##
## Number of Thefts by Day:
##  Mean = 148.21226
##  Variance = 399.88364
##
## Number of Battery Crimes by Day:
##  Mean = 118.77358
##  Variance = 484.67835
##
## Number of Criminal Damage Crimes by Day:
##  Mean = 80.80189
##  Variance = 279.06957
##
## Number of Motor Vehicle Thefts by Day:
##  Mean = 80.56604
##  Variance = 189.95296
##
## Number of Assaults by Day:
##  Mean = 60.40566
##  Variance = 104.17589
```

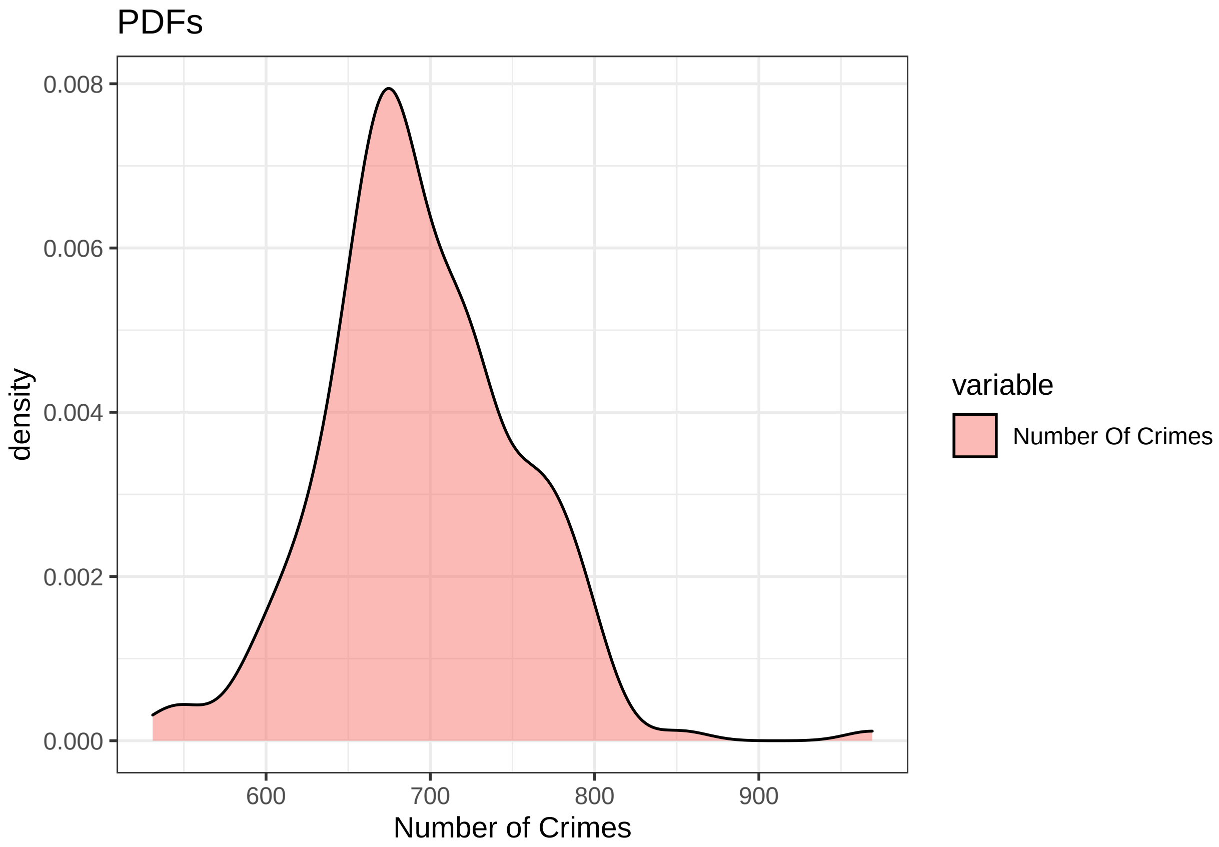
```
meltedDens <- melt(dfCounts[c('Arrest',
                              'Domestic')])

# 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')
```



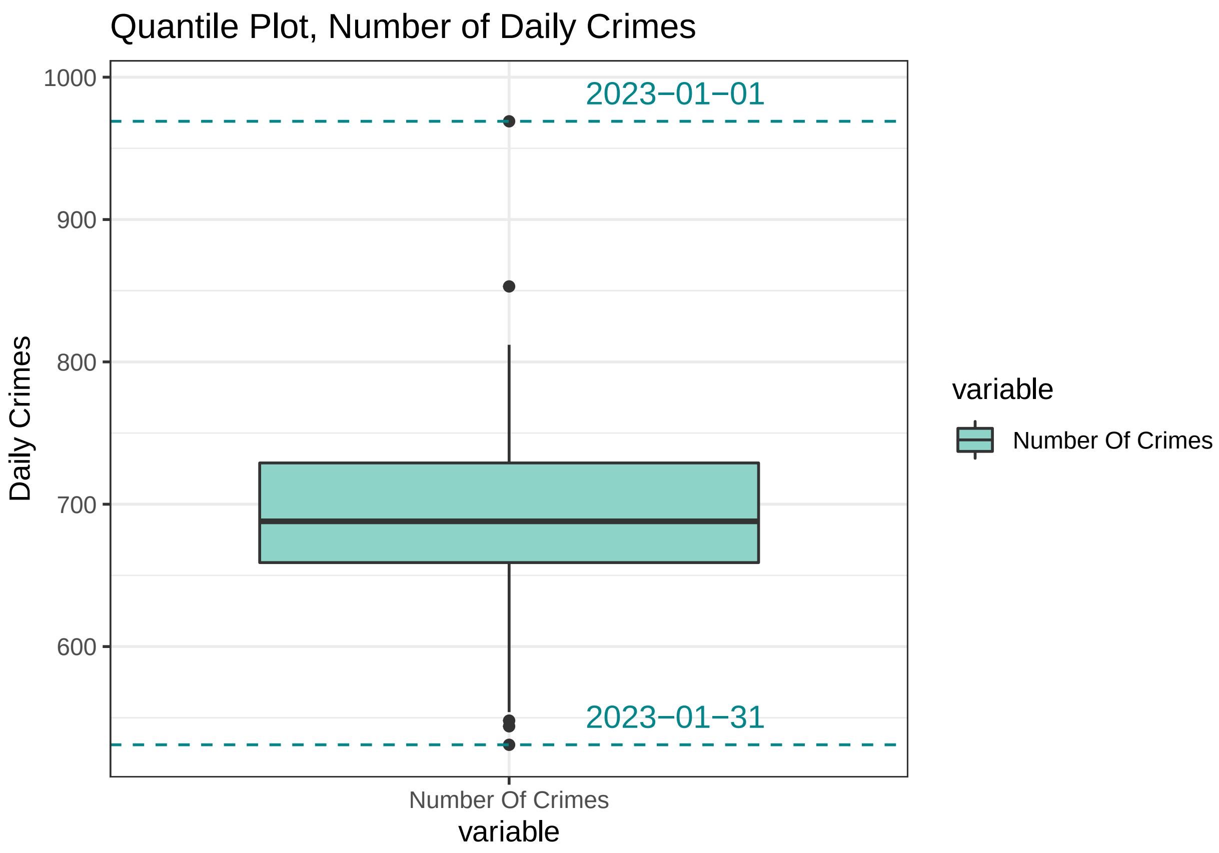
```
# boxplot for the arrest and domestic count columns
aplot <- ggplot(meltedDens,
       aes(x = variable, y = value, fill = variable),
       ) + geom_boxplot()

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()
```



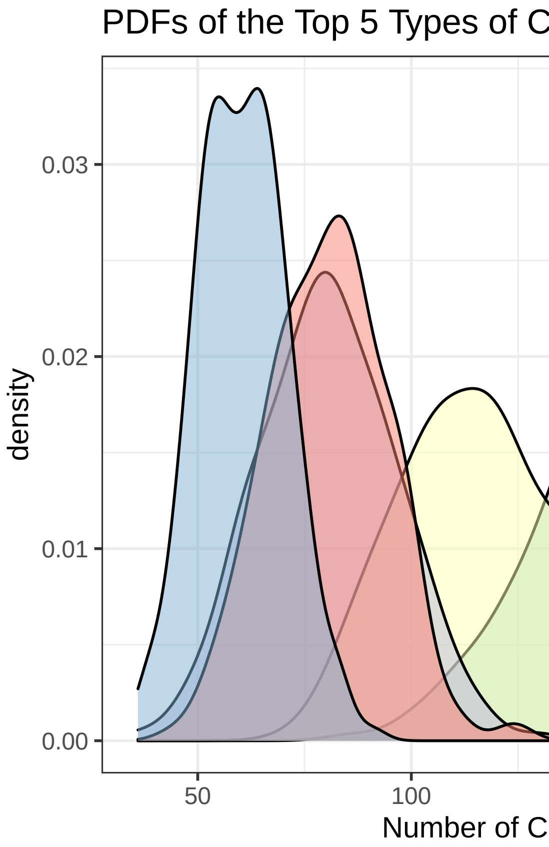
```r
# pdf for the number of crimes counts
meltedDens <- melt(dfCounts[c('Number Of Crimes')])
ggplot(meltedDens, aes(x = value, fill = variable)) +
  geom_density(alpha = 0.5, adjust = 1) +
  xlab('Number of Crimes') +
  theme_bw() +
  ggtitle('PDFs')
```



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

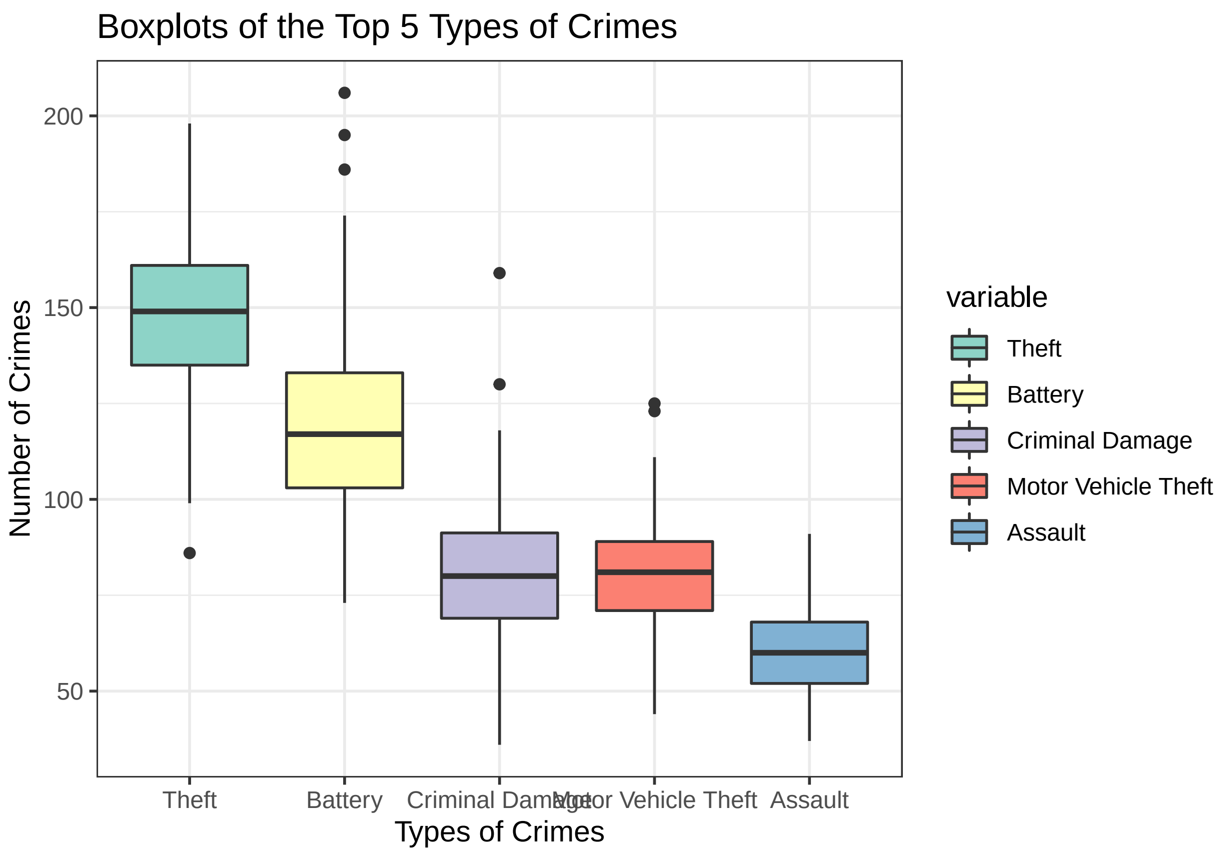


```
meltedDens <- melt(dfCounts[ttLabels])

# 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 5 Types of Crimes')
```

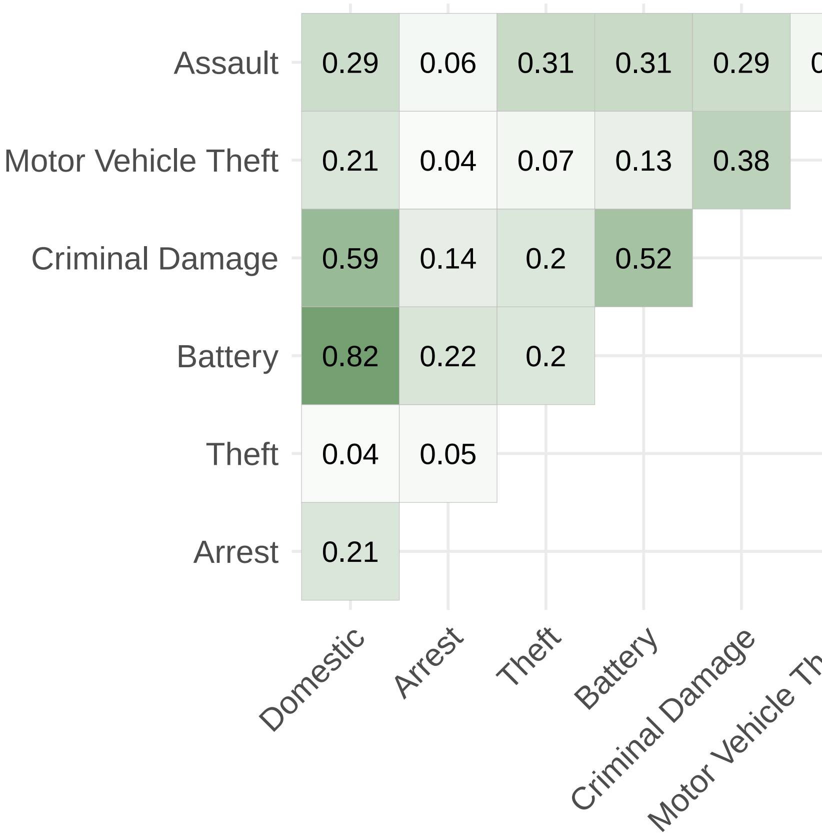


Visualizations for the most common types of crimes in the dataset

```
# 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 5 Types of Crimes') +
  scale_x_discrete(name = 'Types of Crimes',
                   limits = c(ttLabels) ) +
  theme_bw()
```



```
corrCounts <- dfCounts[3:9] %>% cor(method = 'pearson')
corrCounts %>% ggcorrplot::ggcorrplot(lab = TRUE, type = 'upper',
                                      colors = c('lightgoldenrod2', 'white',
                                                 'palegreen4'))
```

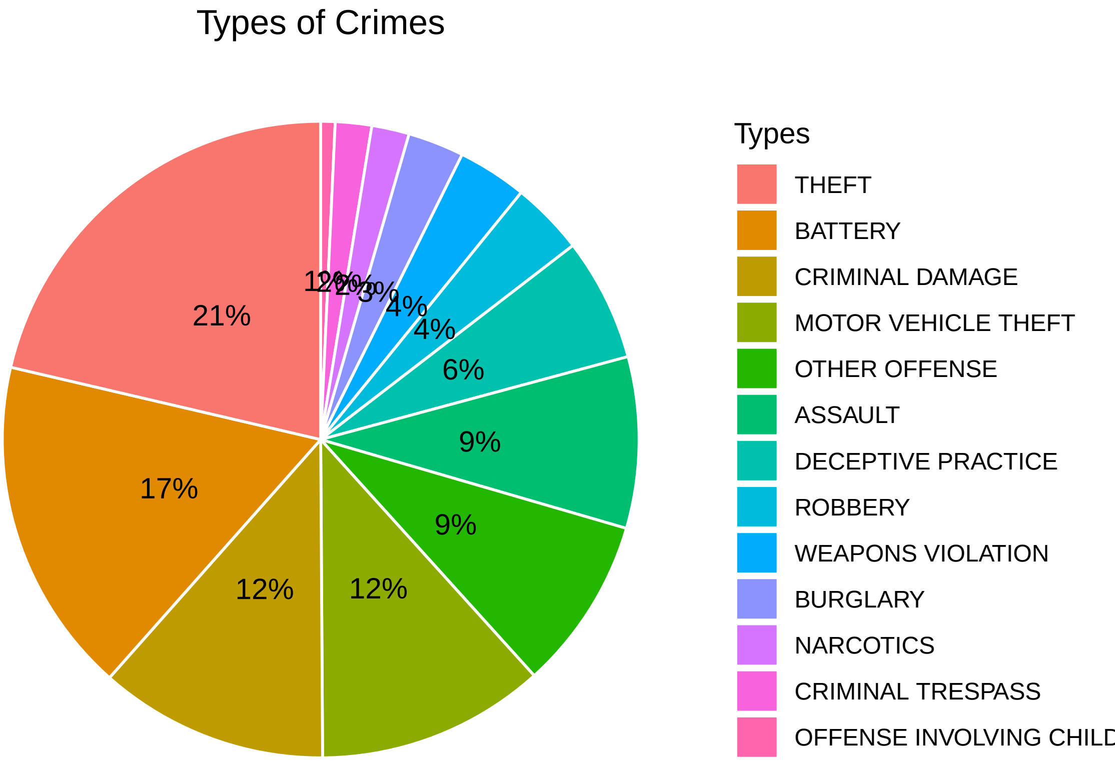


Correlation plot for the counts dataset

```r
# dataset for each type of primary type of crime, with their frequency counts
tabTypeDF <- data.frame(tabType)
colnames(tabTypeDF) <- c('Types', 'Frequency')

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

# Move specific crime types to the "other offense" column for better visualization
tabTypeDF[tabTypeDF$Types == 'OTHER OFFENSE', 2] <- sum(sum(tabTypeDF[tabTypeDF$Frequency < 1000,]$Frequ
                                                        tabTypeDF[tabTypeDF$Types == 'OTHER OFFENSE',]$F

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


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

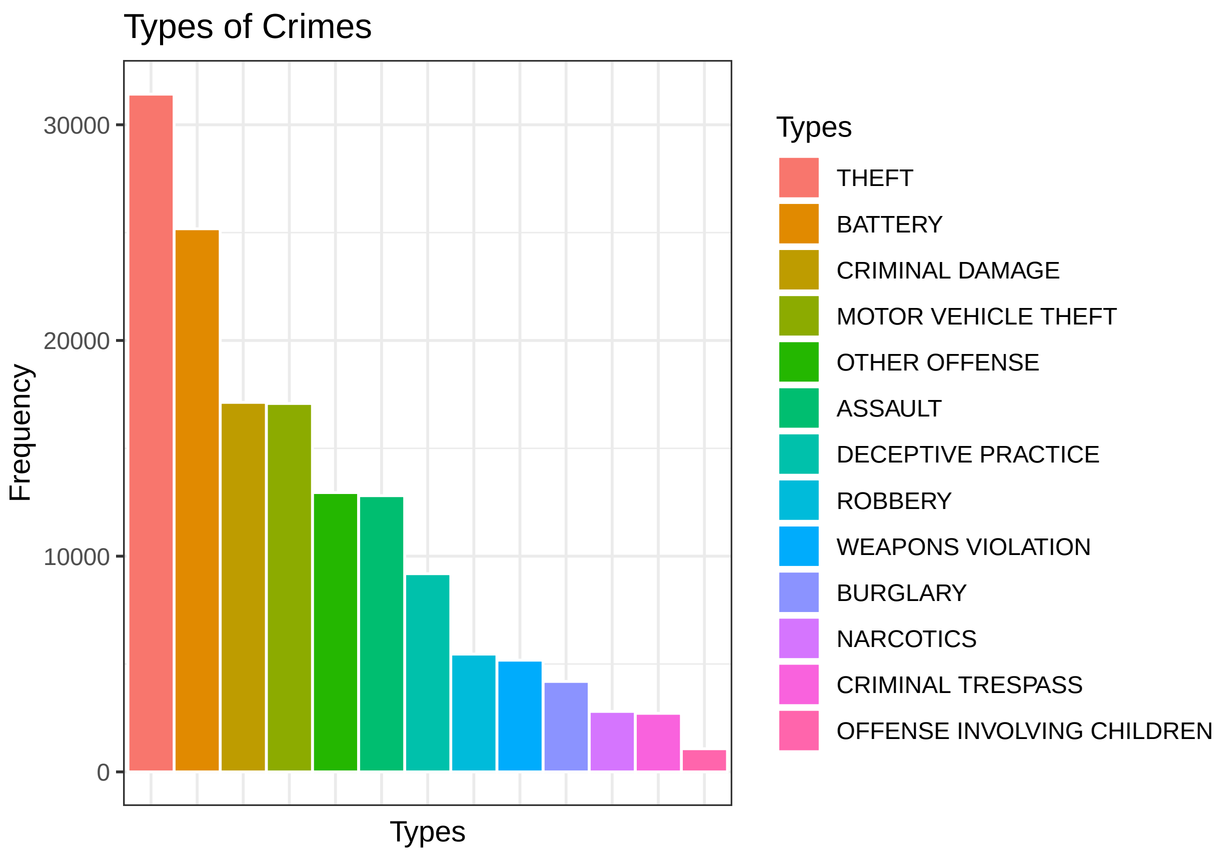
Cleaning for data visualization

```r
# Pie plot for most common types of crimes commit
pieP <- ggplot(tabTypeDF[tabTypeDF$Frequency > 1000,],
               aes(x="", y=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))

# Bar plot for the most common types of crimes commit
barP <- ggplot(tabTypeDF[tabTypeDF$Frequency > 1000,],
               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'))

# Calculations for below bar plot
numDays <- length(unique(df$Date))
tabTypeDFAvg <- tabTypeDF
tabTypeDFAvg$Frequency <- tabTypeDFAvg$Frequency / numDays

# Bar plot for the most common types of crimes commit, as mean values
barPAvg <- ggplot(tabTypeDFAvg[tabTypeDFAvg$Frequency > 1,],
               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'))

# visualization of the plots
pieP
```

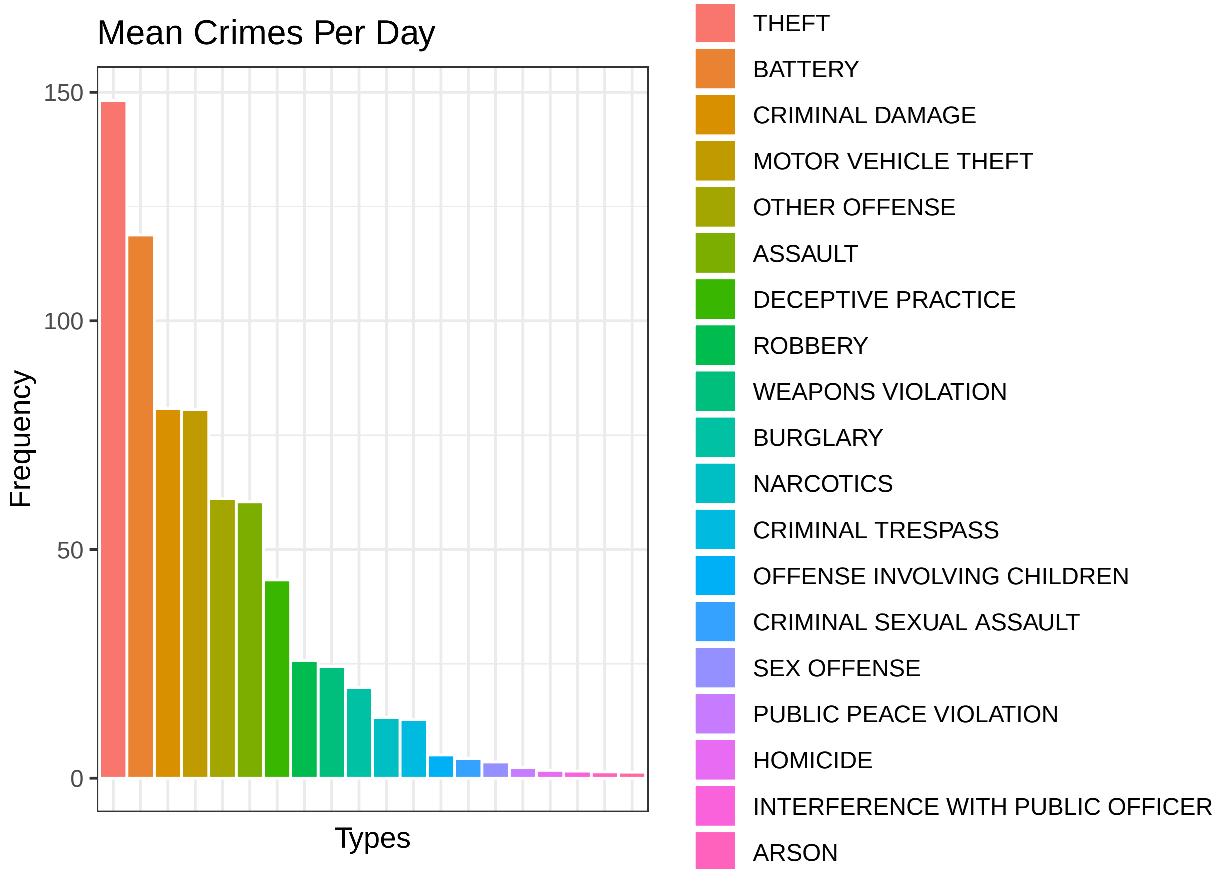


Pie and bar plots

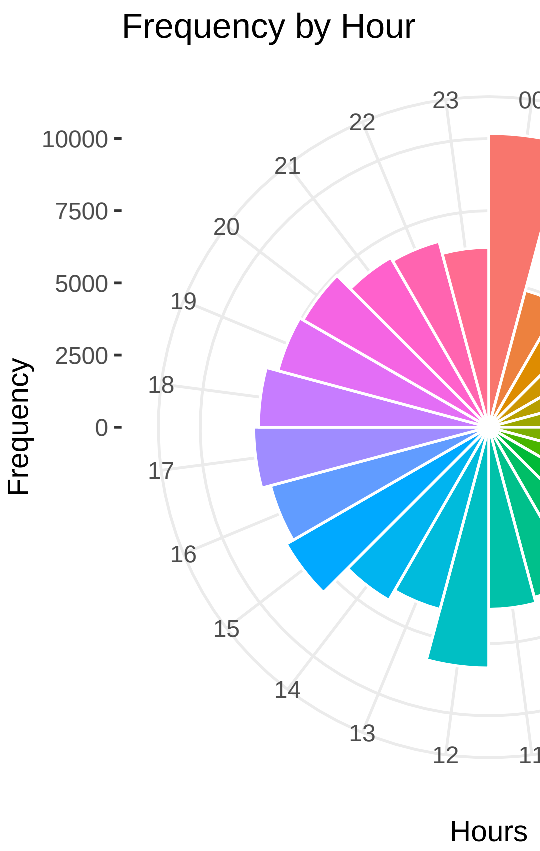
```
barP
```


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

```
barPAvg
```

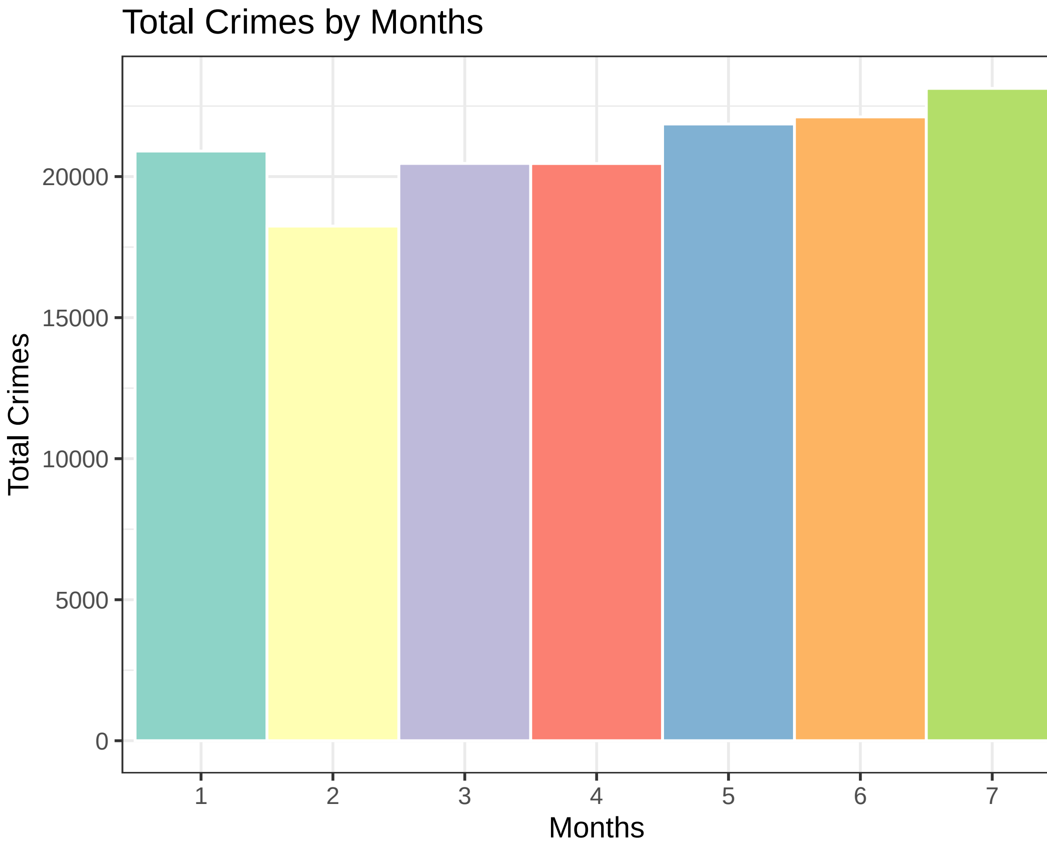


```r
#
hourTimesAM <- substr(df[df$`Time of Day` == 'AM',]$Time, start = 1, stop = 2)
# turn 12 AM to 00 AM
hourTimes <- ifelse(hourTimesAM == '12', '00', hourTimesAM)
hourTimesPM <- c(substr(df[df$`Time of Day` == 'PM',]$Time, start = 1, stop = 2))
# add 12 to all except 12 PM
hourTimes <- c(hourTimes, ifelse(as.integer(hourTimesPM) != 12,
                                 as.integer(hourTimesPM) + 12,
                                 hourTimesPM))

hourTimesFreq <- table(hourTimes)
hourTimesDF <- data.frame(hourTimesFreq)
colnames(hourTimesDF) <- c('Hours', 'Frequency')

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
```



Clock of when most crimes happened during the day

```
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')

dfCountsMonths <- data.frame('Months' = unique(dfCounts$Month), row.names = c(unique(dfCounts$Month)))

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


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'))
```



Total crimes by each month