In [ ]:
```
# imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# read in data
raw_data = pd.read_csv('../data/raw/Bos_crime_2023.csv')
```

Week2 EXPLORE DATASET

In [ ]:
```
# print head of data
print(raw_data.head())
```

```
   _id INCIDENT_NUMBER  OFFENSE_CODE  OFFENSE_CODE_GROUP  \
0    1       232006462          1107                 NaN
1    2       232000093          1402                 NaN
2    3       232003336          3115                 NaN
3    4       232011306          3115                 NaN
4    5       232000104          3831                 NaN

                    OFFENSE_DESCRIPTION DISTRICT REPORTING_AREA  SHOOTING  \
0                 FRAUD - IMPERSONATION      E13            574         0
1                             VANDALISM       C6            200         0
2                   INVESTIGATE PERSON       B3            466         0
3                   INVESTIGATE PERSON       B2            282         0
4  M/V - LEAVING SCENE - PROPERTY DAMAGE      E13            579         0

         OCCURRED_ON_DATE  YEAR  MONTH DAY_OF_WEEK  HOUR  UCR_PART  \
0  2023-01-01 00:00:00+00  2023      1      Sunday     0       NaN
1  2023-01-01 00:01:00+00  2023      1      Sunday     0       NaN
2  2023-01-13 14:00:00+00  2023      1      Friday    14       NaN
3  2023-02-11 00:00:00+00  2023      2    Saturday     0       NaN
4  2023-01-01 00:00:00+00  2023      1      Sunday     0       NaN

              STREET        Lat       Long  \
0      WASHINGTON ST  42.309718 -71.104295
1         W BROADWAY  42.341287 -71.054680
2       DEERING ROAD        NaN        NaN
3  WASHINGTON STREET        NaN        NaN
4        ESTRELLA ST  42.322432 -71.102849

                                  Location
0   (42.30971815419562, -71.10429497557632)
1   (42.34128702104515, -71.05467980204799)
2                                       NaN
3                                       NaN
4  (42.322431629155794, -71.10284879123009)
```

In [ ]:
```
# explore data
print(raw_data.describe())
print(raw_data.info())
```

```
                _id  OFFENSE_CODE  OFFENSE_CODE_GROUP      SHOOTING  \
count  81133.000000  81133.000000                 0.0  81133.000000
mean   40567.000000   2340.595861                 NaN      0.008073
std    23421.224032   1175.333353                 NaN      0.089488
min        1.000000    111.000000                 NaN      0.000000
25%    20284.000000   1106.000000                 NaN      0.000000
50%    40567.000000   2907.000000                 NaN      0.000000
75%    60850.000000   3201.000000                 NaN      0.000000
max    81133.000000   3831.000000                 NaN      1.000000

               YEAR         MONTH          HOUR  UCR_PART           Lat  \
count  81133.000000  81133.000000  81133.000000       0.0  7.528700e+04
mean    2023.044261      6.361481     12.486411       NaN  4.232312e+01
std        0.205675      3.530441      6.564979       NaN  1.576223e-01
min     2023.000000      1.000000      0.000000       NaN  1.327335e-07
25%     2023.000000      3.000000      8.000000       NaN  4.229755e+01
50%     2023.000000      6.000000     13.000000       NaN  4.232866e+01
75%     2023.000000      9.000000     18.000000       NaN  4.234906e+01
max     2024.000000     12.000000     23.000000       NaN  4.239504e+01

               Long
count  7.528700e+04
mean  -7.108282e+01
std    2.610060e-01
min   -7.120251e+01
25%   -7.109943e+01
50%   -7.107775e+01
75%   -7.106090e+01
max    5.249645e-08
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 81133 entries, 0 to 81132
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   _id                  81133 non-null  int64
 1   INCIDENT_NUMBER      81133 non-null  object
 2   OFFENSE_CODE         81133 non-null  int64
 3   OFFENSE_CODE_GROUP   0 non-null      float64
 4   OFFENSE_DESCRIPTION  81133 non-null  object
 5   DISTRICT             80929 non-null  object
 6   REPORTING_AREA       81133 non-null  object
 7   SHOOTING             81133 non-null  int64
 8   OCCURRED_ON_DATE     81133 non-null  object
 9   YEAR                 81133 non-null  int64
 10  MONTH                81133 non-null  int64
 11  DAY_OF_WEEK          81133 non-null  object
 12  HOUR                 81133 non-null  int64
 13  UCR_PART             0 non-null      float64
 14  STREET               81133 non-null  object
 15  Lat                  75287 non-null  float64
 16  Long                 75287 non-null  float64
 17  Location             75287 non-null  object
dtypes: float64(4), int64(6), object(8)
memory usage: 11.1+ MB
None
```

In [ ]:
```
# print all types of OFFENSE_DESCRIPTION
print(raw_data['OFFENSE_DESCRIPTION'].unique())
```

```
['FRAUD - IMPERSONATION' 'VANDALISM' 'INVESTIGATE PERSON'
 'M/V - LEAVING SCENE - PROPERTY DAMAGE'
 'LARCENY THEFT FROM MV - NON-ACCESSORY' 'M/V ACCIDENT - PERSONAL INJURY'
 'THREATS TO DO BODILY HARM' 'SICK ASSIST - DRUG RELATED ILLNESS'
 'FRAUD - WELFARE' 'M/V ACCIDENT - OTHER' 'SICK ASSIST' 'VERBAL DISPUTE'
 'INVESTIGATE PROPERTY' 'ASSAULT - SIMPLE' 'BURGLARY - RESIDENTIAL'
 'TOWED MOTOR VEHICLE' 'ASSAULT - AGGRAVATED'
 'PROPERTY - LOST THEN LOCATED' 'HARASSMENT/ CRIMINAL HARASSMENT'
 'PROPERTY - LOST/ MISSING' 'OPERATING UNDER THE INFLUENCE (OUI) ALCOHOL'
 'VAL - OPERATING W/O AUTHORIZATION LAWFUL' 'LICENSE PREMISE VIOLATION'
 'SICK/INJURED/MEDICAL - PERSON' 'M/V ACCIDENT - PROPERTY DAMAGE'
 'LARCENY THEFT FROM BUILDING' 'AUTO THEFT'
 'M/V ACCIDENT - INVOLVING PEDESTRIAN - INJURY'
 'MURDER, NON-NEGLIGENT MANSLAUGHTER' 'PROPERTY - FOUND'
 'LIQUOR/ALCOHOL - DRINKING IN PUBLIC' 'VAL - VIOLATION OF AUTO LAW'
 'ROBBERY' 'LARCENY THEFT OF MV PARTS & ACCESSORIES' 'MISSING PERSON'
 'FIRE REPORT' 'FRAUD - CREDIT CARD / ATM FRAUD' 'LARCENY ALL OTHERS'
 'FIREARM/WEAPON - FOUND OR CONFISCATED'
 'DANGEROUS OR HAZARDOUS CONDITION' 'LARCENY SHOPLIFTING'
 'VIOLATION - CITY ORDINANCE' 'BALLISTICS EVIDENCE/FOUND'
 'DRUGS - POSSESSION/ SALE/ MANUFACTURING/ USE' 'TRESPASSING'
 'MISSING PERSON - LOCATED'
 'STOLEN PROPERTY - BUYING / RECEIVING / POSSESSING'
 'SERVICE TO OTHER AGENCY' 'FORGERY / COUNTERFEITING' 'M/V PLATES - LOST'
 'SUDDEN DEATH' 'DEATH INVESTIGATION' 'LARCENY THEFT OF BICYCLE'
 'SICK/INJURED/MEDICAL - POLICE' 'M/V ACCIDENT - POLICE VEHICLE'
 'VAL - OPERATING AFTER REV/SUSP.' 'PROPERTY - STOLEN THEN RECOVERED'
 'WEAPON VIOLATION - CARRY/ POSSESSING/ SALE/ TRAFFICKING/ OTHER'
 'ANIMAL INCIDENTS (DOG BITES, LOST DOG, ETC)'
 'M/V - LEAVING SCENE - PERSONAL INJURY'
 'CHILD REQUIRING ASSISTANCE (FOMERLY CHINS)'
 'PRISONER - SUICIDE / SUICIDE ATTEMPT' 'FRAUD - FALSE PRETENSE / SCHEME'
 'EVADING FARE' 'BURGLARY - COMMERICAL'
 'MISSING PERSON - NOT REPORTED - LOCATED'
 'M/V ACCIDENT - INVOLVING BICYCLE - NO INJURY'
 'M/V ACCIDENT - OTHER CITY VEHICLE'
 'WARRANT ARREST - OUTSIDE OF BOSTON WARRANT'
 'RECOVERED - MV RECOVERED IN BOSTON (STOLEN OUTSIDE BOSTON)'
 'FRAUD - WIRE' 'LANDLORD - TENANT' 'EXTORTION OR BLACKMAIL'
 'M/V ACCIDENT - INVOLVING BICYCLE - INJURY' 'SEARCH WARRANT'
 'DISTURBING THE PEACE/ DISORDERLY CONDUCT/ GATHERING CAUSING ANNOYANCE/ NOISY PA
R'
 'PROPERTY - ACCIDENTAL DAMAGE' 'NOISY PARTY/RADIO-NO ARREST' 'GRAFFITI'
 'LARCENY PICK-POCKET' 'OTHER OFFENSE'
 'DRUGS - POSSESSION OF DRUG PARAPHANALIA'
 'WARRANT ARREST - BOSTON WARRANT (MUST BE SUPPLEMENTAL)'
 'FUGITIVE FROM JUSTICE' 'HARBOR INCIDENT / VIOLATION'
 'INJURY BICYCLE NO M/V INVOLVED'
 'M/V ACCIDENT - INVOLVING PEDESTRIAN - NO INJURY'
 'FIRE REPORT/ALARM - FALSE' 'AUTO THEFT - MOTORCYCLE / SCOOTER' 'ARSON'
 'SUICIDE / SUICIDE ATTEMPT' 'EMBEZZLEMENT'
 'AUTO THEFT - LEASED/RENTED VEHICLE'
 'RECOVERED - MV RECOVERED IN BOSTON (STOLEN IN BOSTON) MUST BE SUPPLEMENTAL'
 'BOMB THREAT' 'TRUANCY / RUNAWAY' 'MANSLAUGHTER - VEHICLE - NEGLIGENCE'
 'AIRCRAFT INCIDENTS' 'AFFRAY' 'DRUNKENNESS' 'INTIMIDATING WITNESS'
 'LARCENY PURSE SNATCH - NO FORCE' 'LIQUOR LAW VIOLATION'
 'BREAKING AND ENTERING (B&E) MOTOR VEHICLE' 'ANIMAL ABUSE'
 'FIREARM/WEAPON - LOST' 'MURDER, NON-NEGLIGIENT MANSLAUGHTER'
 'LARCENY THEFT FROM COIN-OP MACHINE'
 'KIDNAPPING/CUSTODIAL KIDNAPPING/ ABDUCTION'
```

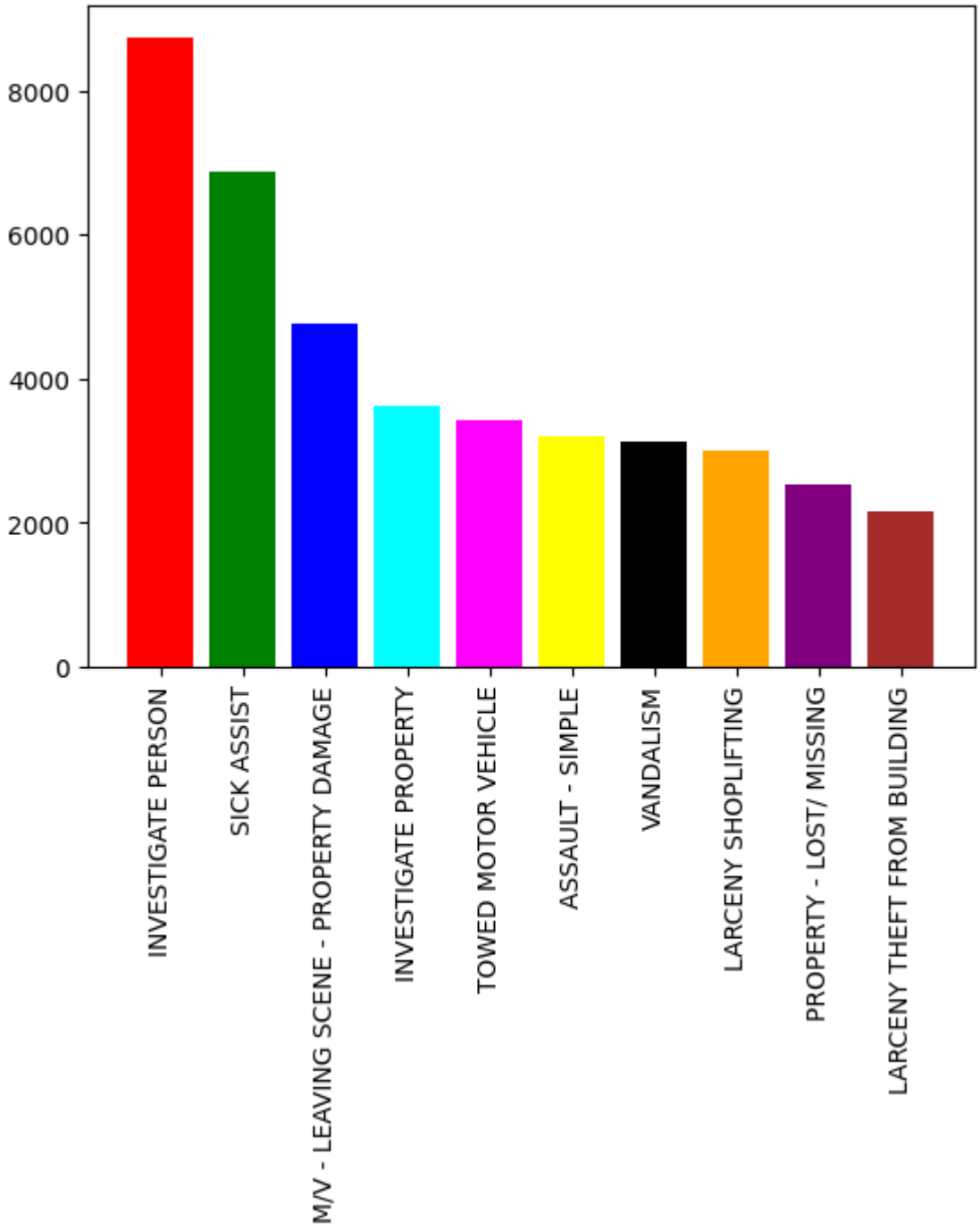
```
'PROTECTIVE CUSTODY / SAFEKEEPING'
'FIREARM/WEAPON - ACCIDENTAL INJURY / DEATH' 'PROSTITUTION - SOLICITING'
'EXPLOSIVES - POSSESSION OR USE' 'OBSCENE PHONE CALLS' 'PROSTITUTION'
'BREAKING AND ENTERING (B&E) MOTOR VEHICLE (NO PROPERTY STOLEN)'
'OPERATING UNDER THE INFLUENCE (OUI) DRUGS'
'POSSESSION OF BURGLARIOUS TOOLS']
```

In [ ]:

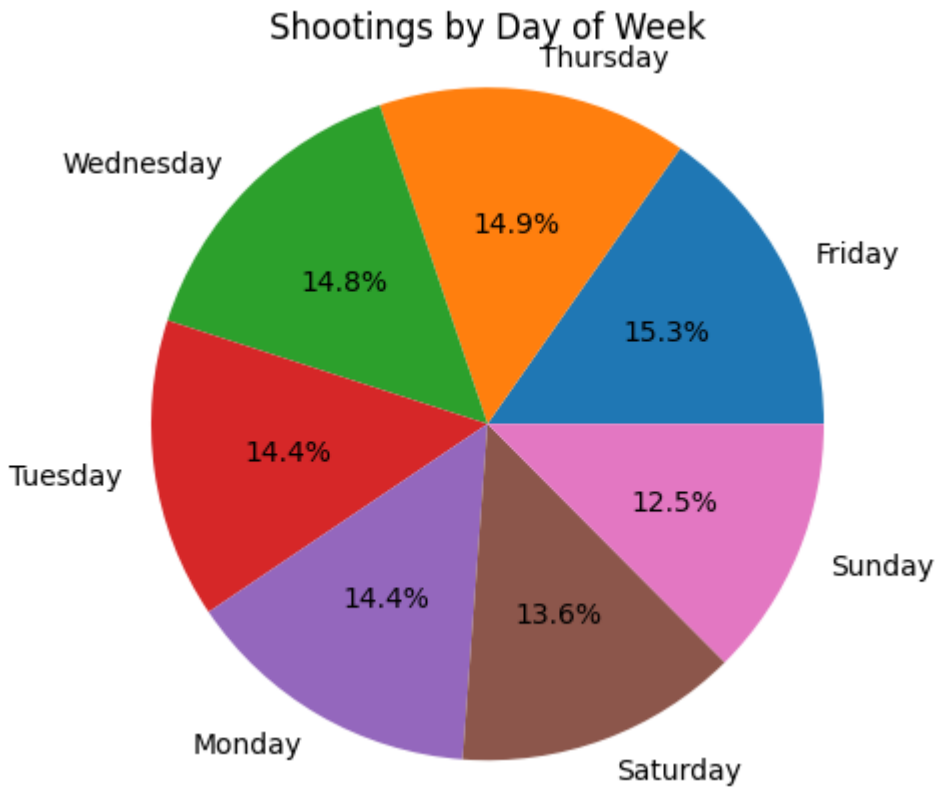
```python
# create a list of all crime descriptions and their counts
crime_list = raw_data['OFFENSE_DESCRIPTION'].value_counts().reset_index().values

# print crime_list
print(crime_list)

# save crime_list to csv
crime_df = pd.DataFrame(crime_list)
crime_df.to_csv('../data/processed/crime_list.csv', index=False)

print('Done!')
```

```
[['INVESTIGATE PERSON', 8741], ['SICK ASSIST', 6869], ['M/V - LEAVING SCENE - PRO
PERTY DAMAGE', 4759], ['INVESTIGATE PROPERTY', 3610], ['TOWED MOTOR VEHICLE', 342
4], ['ASSAULT - SIMPLE', 3210], ['VANDALISM', 3132], ['LARCENY SHOPLIFTING', 298
8], ['PROPERTY - LOST/ MISSING', 2519], ['LARCENY THEFT FROM BUILDING', 2164],
['LARCENY THEFT FROM MV - NON-ACCESSORY', 2026], ['M/V ACCIDENT - PROPERTY DAMAG
E', 1910], ['THREATS TO DO BODILY HARM', 1864], ['VERBAL DISPUTE', 1854], ['DRUGS
- POSSESSION/ SALE/ MANUFACTURING/ USE', 1810], ['LARCENY ALL OTHERS', 1713],
['M/V ACCIDENT - OTHER', 1648], ['ASSAULT - AGGRAVATED', 1585], ['MISSING PERSON
- LOCATED', 1283], ['HARASSMENT/ CRIMINAL HARASSMENT', 1273], ['FRAUD - FALSE PRE
TENSE / SCHEME', 1262], ['SICK/INJURED/MEDICAL - PERSON', 1236], ['VAL - VIOLATIO
N OF AUTO LAW', 1138], ['AUTO THEFT', 1121], ['M/V ACCIDENT - PERSONAL INJURY', 1
041], ['PROPERTY - FOUND', 985], ['SICK ASSIST - DRUG RELATED ILLNESS', 812], ['R
OBBERY', 794], ['LARCENY THEFT OF BICYCLE', 679], ['BURGLARY - RESIDENTIAL', 67
8], ['M/V ACCIDENT - INVOLVING PEDESTRIAN - INJURY', 572], ['FRAUD - CREDIT CARD
/ ATM FRAUD', 542], ['WARRANT ARREST - OUTSIDE OF BOSTON WARRANT', 497], ['DEATH
INVESTIGATION', 447], ['FORGERY / COUNTERFEITING', 444], ['FIRE REPORT', 439],
['SUDDEN DEATH', 437], ['LANDLORD - TENANT', 407], ['TRESPASSING', 400], ['VAL -
OPERATING AFTER REV/SUSP.', 397], ['LICENSE PREMISE VIOLATION', 396], ['SICK/INJU
RED/MEDICAL - POLICE', 385], ['MISSING PERSON', 376], ['MISSING PERSON - NOT REPO
RTED - LOCATED', 367], ['LARCENY THEFT OF MV PARTS & ACCESSORIES', 340], ['SERVIC
E TO OTHER AGENCY', 333], ['M/V - LEAVING SCENE - PERSONAL INJURY', 332], ['BURGL
ARY - COMMERICAL', 323], ['WEAPON VIOLATION - CARRY/ POSSESSING/ SALE/ TRAFFICKIN
G/ OTHER', 307], ['FRAUD - IMPERSONATION', 297], ['RECOVERED - MV RECOVERED IN BO
STON (STOLEN OUTSIDE BOSTON)', 292], ['BALLISTICS EVIDENCE/FOUND', 276], ['FRAUD
- WIRE', 266], ['M/V ACCIDENT - POLICE VEHICLE', 261], ['PROPERTY - ACCIDENTAL DA
MAGE', 261], ['M/V ACCIDENT - OTHER CITY VEHICLE', 259], ['ANIMAL INCIDENTS (DOG
BITES, LOST DOG, ETC)', 230], ['M/V ACCIDENT - INVOLVING BICYCLE - INJURY', 226],
['AUTO THEFT - MOTORCYCLE / SCOOTER', 222], ['FIREARM/WEAPON - FOUND OR CONFISCAT
ED', 193], ['STOLEN PROPERTY - BUYING / RECEIVING / POSSESSING', 174], ['M/V PLAT
ES - LOST', 151], ['DISTURBING THE PEACE/ DISORDERLY CONDUCT/ GATHERING CAUSING A
NNOYANCE/ NOISY PAR', 148], ['SEARCH WARRANT', 121], ['M/V ACCIDENT - INVOLVING P
EDESTRIAN - NO INJURY', 118], ['VIOLATION - CITY ORDINANCE', 111], ['M/V ACCIDENT
- INVOLVING BICYCLE - NO INJURY', 103], ['OPERATING UNDER THE INFLUENCE (OUI) ALC
OHOL', 95], ['EXTORTION OR BLACKMAIL', 95], ['FRAUD - WELFARE', 82], ['WARRANT AR
REST - BOSTON WARRANT (MUST BE SUPPLEMENTAL)', 79], ['LIQUOR/ALCOHOL - DRINKING I
N PUBLIC', 75], ['LIQUOR LAW VIOLATION', 71], ['AUTO THEFT - LEASED/RENTED VEHICL
E', 67], ['LARCENY PICK-POCKET', 63], ['NOISY PARTY/RADIO-NO ARREST', 63], ['HARB
OR INCIDENT / VIOLATION', 63], ['VAL - OPERATING W/O AUTHORIZATION LAWFUL', 57],
['FUGITIVE FROM JUSTICE', 46], ['GRAFFITI', 46], ['EMBEZZLEMENT', 46], ['DANGEROU
S OR HAZARDOUS CONDITION', 44], ['PROPERTY - LOST THEN LOCATED', 43], ['FIRE REPO
RT/ALARM - FALSE', 43], ['AFFRAY', 40], ['PROPERTY - STOLEN THEN RECOVERED', 34],
['SUICIDE / SUICIDE ATTEMPT', 32], ['OTHER OFFENSE', 31], ['MURDER, NON-NEGLIGENT
MANSLAUGHTER', 30], ['BOMB THREAT', 30], ['AIRCRAFT INCIDENTS', 26], ['RECOVERED
- MV RECOVERED IN BOSTON (STOLEN IN BOSTON) MUST BE SUPPLEMENTAL', 25], ['EVADING
FARE', 25], ['INJURY BICYCLE NO M/V INVOLVED', 23], ['ARSON', 20], ['DRUGS - POSS
ESSION OF DRUG PARAPHANALIA', 19], ['LARCENY PURSE SNATCH - NO FORCE', 17], ['BRE
AKING AND ENTERING (B&E) MOTOR VEHICLE', 16], ['TRUANCY / RUNAWAY', 15], ['INTIMI
DATING WITNESS', 14], ['CHILD REQUIRING ASSISTANCE (FOMERLY CHINS)', 10], ['PRISO
NER - SUICIDE / SUICIDE ATTEMPT', 10], ['DRUNKENNESS', 8], ['FIREARM/WEAPON - LOS
T', 7], ['ANIMAL ABUSE', 7], ['LARCENY THEFT FROM COIN-OP MACHINE', 6], ['KIDNAPP
ING/CUSTODIAL KIDNAPPING/ ABDUCTION', 6], ['BREAKING AND ENTERING (B&E) MOTOR VEH
ICLE (NO PROPERTY STOLEN)', 4], ['OPERATING UNDER THE INFLUENCE (OUI) DRUGS', 4],
['MANSLAUGHTER - VEHICLE - NEGLIGENCE', 3], ['PROSTITUTION - SOLICITING', 3], ['O
BSCENE PHONE CALLS', 3], ['PROTECTIVE CUSTODY / SAFEKEEPING', 2], ['FIREARM/WEAPO
N - ACCIDENTAL INJURY / DEATH', 2], ['EXPLOSIVES - POSSESSION OR USE', 2], ['MURD
ER, NON-NEGLIGIENT MANSLAUGHTER', 1], ['PROSTITUTION', 1], ['POSSESSION OF BURGLA
RIOUS TOOLS', 1]]
Done!
```

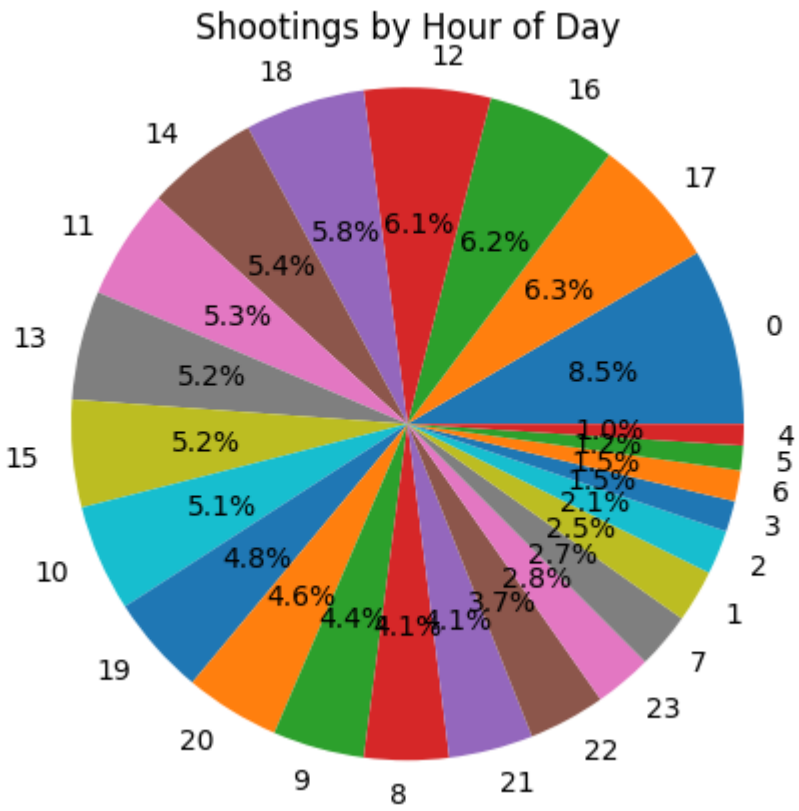
In [ ]:
```
# draw a bar chart of the top 10 crimes using different colors for each crime

# create a list of the top 10 crimes
top_10_crimes = raw_data['OFFENSE_DESCRIPTION'].value_counts().head(10).reset_in

# draw a bar chart of the top 10 crimes
plt.bar(range(len(top_10_crimes)), [val[1] for val in top_10_crimes], align='cen
plt.xticks(range(len(top_10_crimes)), [val[0] for val in top_10_crimes])
plt.xticks(rotation=90)

# save bar chart to file

plt.savefig('../reports/figures/top_10_crimes.png') # need to call this before p


plt.show()
```



In [ ]:
```
# count shootings by day of week
shootings_by_day = raw_data['DAY_OF_WEEK'].value_counts().reset_index().values.t

# draw a pie chart of shootings by day of week
plt.pie([val[1] for val in shootings_by_day], labels=[val[0] for val in shooting
plt.title('Shootings by Day of Week')
plt.axis('equal')

# save pie chart to file
plt.savefig('../reports/figures/shootings_by_day.png')

plt.show()
```



In [ ]:
```
# count shootings by hour of day
shootings_by_hour = raw_data['HOUR'].value_counts().reset_index().values.tolist(

# draw a pie chart of shootings by hour of day
plt.pie([val[1] for val in shootings_by_hour], labels=[val[0] for val in shootin
plt.title('Shootings by Hour of Day')
plt.axis('equal')

# save pie chart to file
plt.savefig('../reports/figures/shootings_by_hour.png')

plt.show()
```



In [ ]:
```
# show datatype of all columns
print(raw_data.dtypes)

# show dataframe size
print(raw_data.shape)
```

```
_id                     int64
INCIDENT_NUMBER        object
OFFENSE_CODE            int64
OFFENSE_CODE_GROUP    float64
OFFENSE_DESCRIPTION    object
DISTRICT               object
REPORTING_AREA         object
SHOOTING                int64
OCCURRED_ON_DATE       object
YEAR                    int64
MONTH                   int64
DAY_OF_WEEK            object
HOUR                    int64
UCR_PART              float64
STREET                 object
Lat                   float64
Long                  float64
Location               object
dtype: object
(81133, 18)
```

WEEK2 DATA CLEANING

In [ ]:
```
# drop Offense Code group column
raw_data.drop('OFFENSE_CODE_GROUP', axis=1, inplace=True)

# drio UCR_PART column
raw_data.drop('UCR_PART', axis=1, inplace=True)
```

```
# drop lat and long and location columns
raw_data.drop('Lat', axis=1, inplace=True)
raw_data.drop('Long', axis=1, inplace=True)
raw_data.drop('Location', axis=1, inplace=True)

# drop incident number column
raw_data.drop('INCIDENT_NUMBER', axis=1, inplace=True)

print(raw_data.shape)


# never run this repeatedly!!!!!!!!!!!!!!!
```

```
(81133, 12)
```

In [ ]:
```
# drop rows with null values]
raw_data.dropna(inplace=True)
raw_data.reset_index(drop=True, inplace=True)
print(raw_data.shape)
```

```
(80929, 12)
```

In [ ]:
```
# save cleaned data to csv
raw_data.to_csv('../data/processed/cleaned_data.csv', index=False)

print('Done!')
```

```
Done!
```