National University
of computer and emerging sciences

DATA SCIENCE PROJECT

Crime Data from 2010 to 2019
(City of Los Angeles)

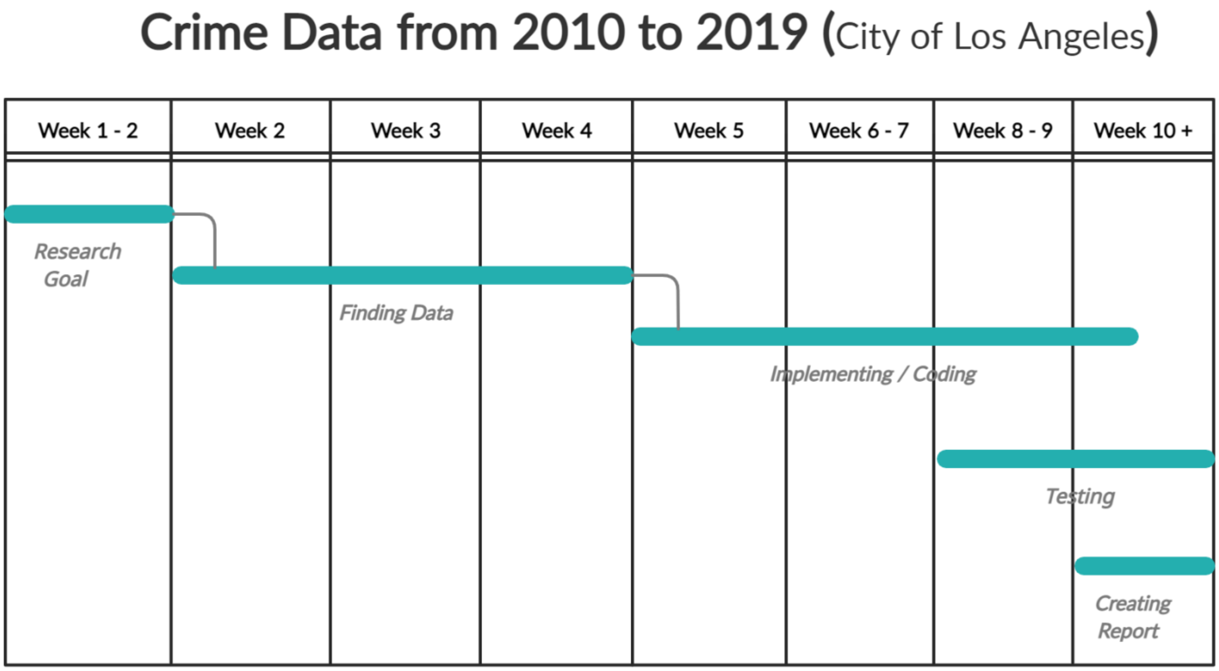
https://catalog.data.gov/dataset/crime-data-from-2010-to-2019

ID: K17-3613
Name: Sameer Khowaja
Sec: 9A

Professor
Dr. Muhammad Atif Tahir

1. Setting the research goal

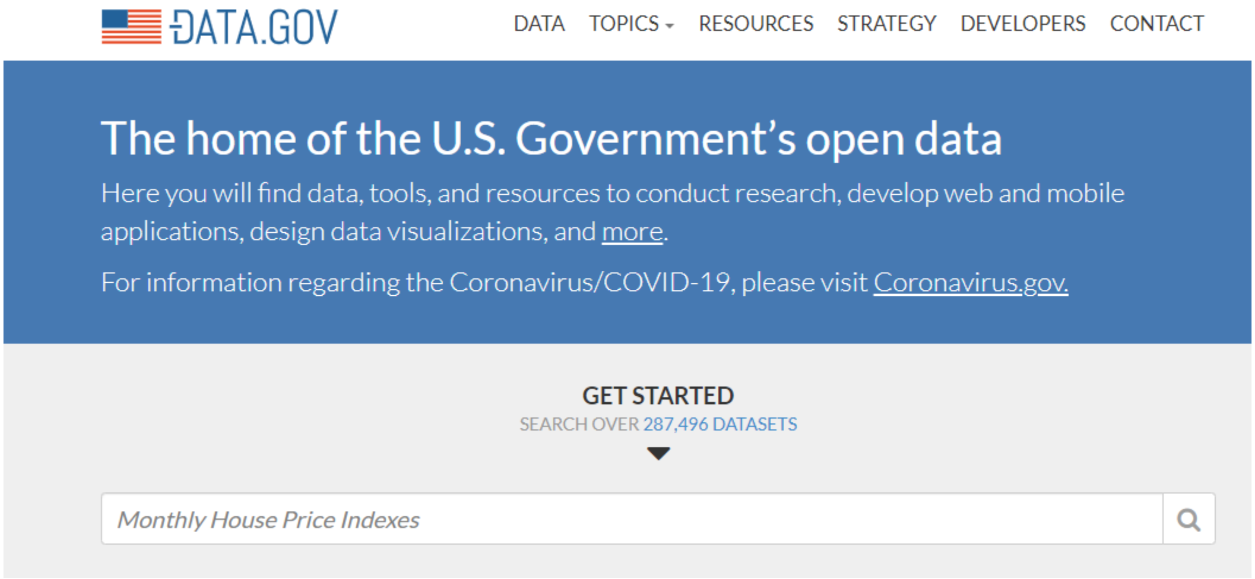
For this course I think why not research on criminal activities happening in the world. So, I choose America as my country and Los Angeles as City to research about crime there. My goal to search for top most criminal activities, targeted gender and age group, find what kind of crime and weapon used and which descent is targeted and most important timing of crime.



Crime Data from 2010 to 2019 (City of Los Angeles)

2. Retrieving data

The Data retrieved as Open data from DATA.GOV website contain 287,496 Datasets. This website contains U.S. Government's open data. My main focus is on criminal data. https://www.data.gov/



The home of the U.S. Government's open data

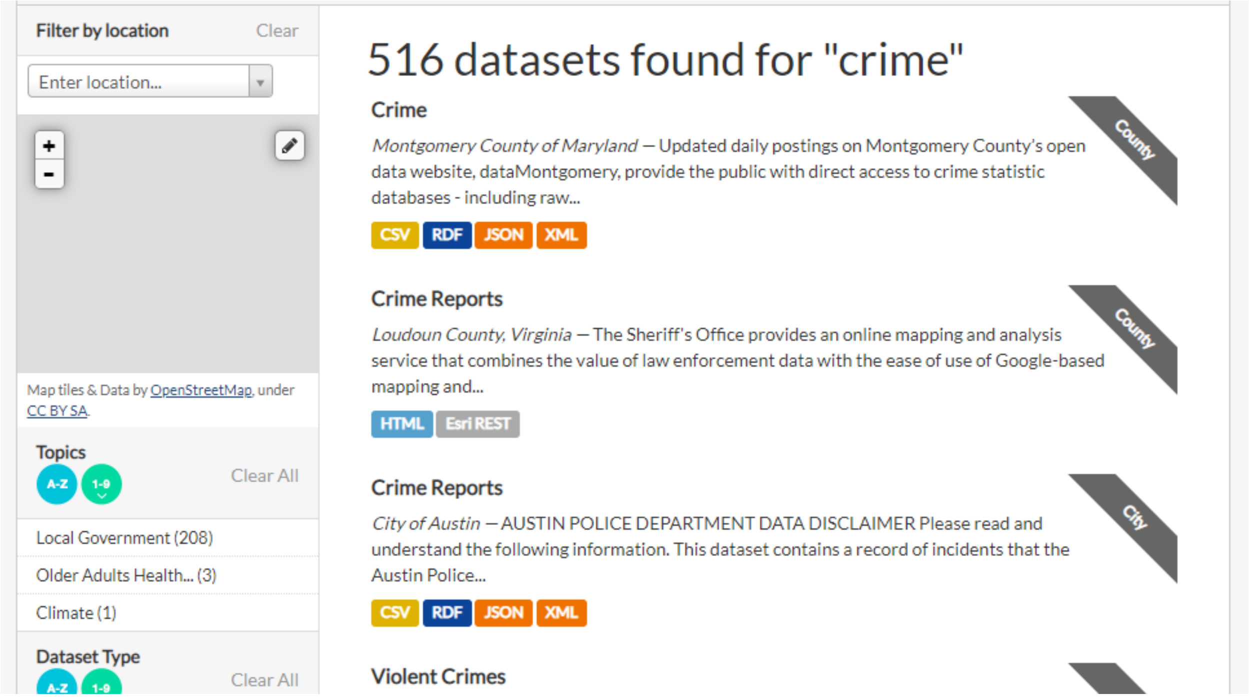
Here you will find data, tools, and resources to conduct research, develop web and mobile applications, design data visualizations, and more.

For information regarding the Coronavirus/COVID-19, please visit Coronavirus.gov.

GET STARTED

SEARCH OVER 287,496 DATASETS

Monthly House Price Indexes



Map tiles & Data by OpenStreetMap, under CC BY SA.

Local Government (208)

Older Adults Health... (3)

Climate (1)

516 datasets found for "crime"

Crime

Montgomery County of Maryland — Updated daily postings on Montgomery County's open data website, dataMontgomery, provide the public with direct access to crime statistic databases - including raw...

CSV RDF JSON XML

Crime Reports

Loudoun County, Virginia — The Sheriff's Office provides an online mapping and analysis service that combines the value of law enforcement data with the ease of use of Google-based mapping and...

HTML Esri REST

Crime Reports

City of Austin — AUSTIN POLICE DEPARTMENT DATA DISCLAIMER Please read and understand the following information. This dataset contains a record of incidents that the Austin Police...

CSV RDF JSON XML

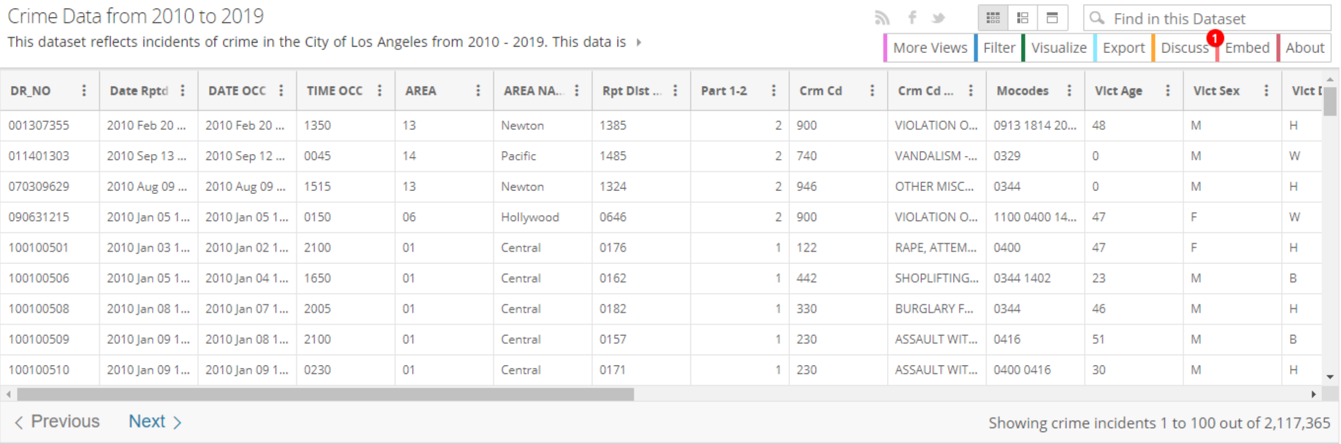
Violent Crimes

I choose **Crime Data from 2010 to 2019 (City of Los Angeles).** This dataset reflects incidents of crime in the City of Los Angeles from 2010 - 2019. This data is transcribed from original crime reports that are typed on paper and therefore there may be some inaccuracies within the data. Some location fields with missing data are noted as (0°, 0°). Address fields are only provided to the nearest hundred blocks in order to maintain privacy. This data is as accurate as the data in the database.

Dataset Download Link: https://catalog.data.gov/dataset/crime-data-from-2010-to-2019

Comma Separated Values File downloaded (510 MB file size)

Contain 28 columns and 2116843 Rows.

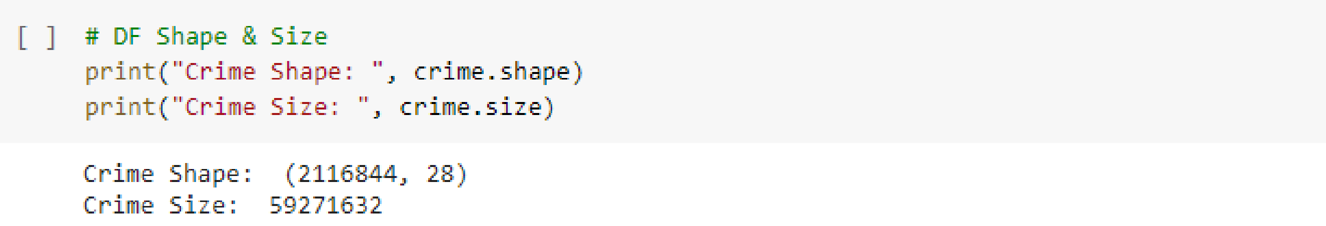


Crime Data from 2010 to 2019

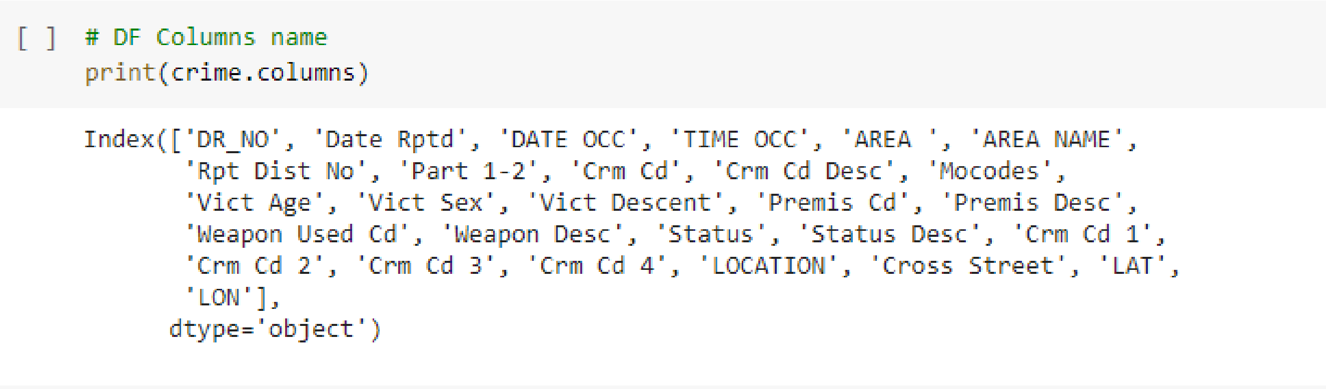
This dataset reflects incidents of crime in the City of Los Angeles from 2010 - 2019. This data is

DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NA...	Rpt Dist ...	Part 1-2	Crm Cd	Crm Cd ...	Mocodes	Vict Age	Vict Sex	Vict D
001307355	2010 Feb 20 ...	2010 Feb 20 ...	1350	13	Newton	1385	2	900	VIOLATION O...	0913 1814 20...	48	M	H
011401303	2010 Sep 13 ...	2010 Sep 12 ...	0045	14	Pacific	1485	2	740	VANDALISM -...	0329	0	M	W
070309629	2010 Aug 09 ...	2010 Aug 09 ...	1515	13	Newton	1324	2	946	OTHER MISC...	0344	0	M	H
090631215	2010 Jan 05 1...	2010 Jan 05 1...	0150	06	Hollywood	0646	2	900	VIOLATION O...	1100 0400 14...	47	F	W
100100501	2010 Jan 03 1...	2010 Jan 02 1...	2100	01	Central	0176	1	122	RAPE, ATTEM...	0400	47	F	H
100100506	2010 Jan 05 1...	2010 Jan 04 1...	1650	01	Central	0162	1	442	SHOPLIFTING...	0344 1402	23	M	B
100100508	2010 Jan 08 1...	2010 Jan 07 1...	2005	01	Central	0182	1	330	BURGLARY F...	0344	46	M	H
100100509	2010 Jan 09 1...	2010 Jan 08 1...	2100	01	Central	0157	1	230	ASSAULT WIT...	0416	51	M	B
100100510	2010 Jan 09 1...	2010 Jan 09 1...	0230	01	Central	0171	1	230	ASSAULT WIT...	0400 0416	30	M	H

Showing crime incidents 1 to 100 out of 2,117,365

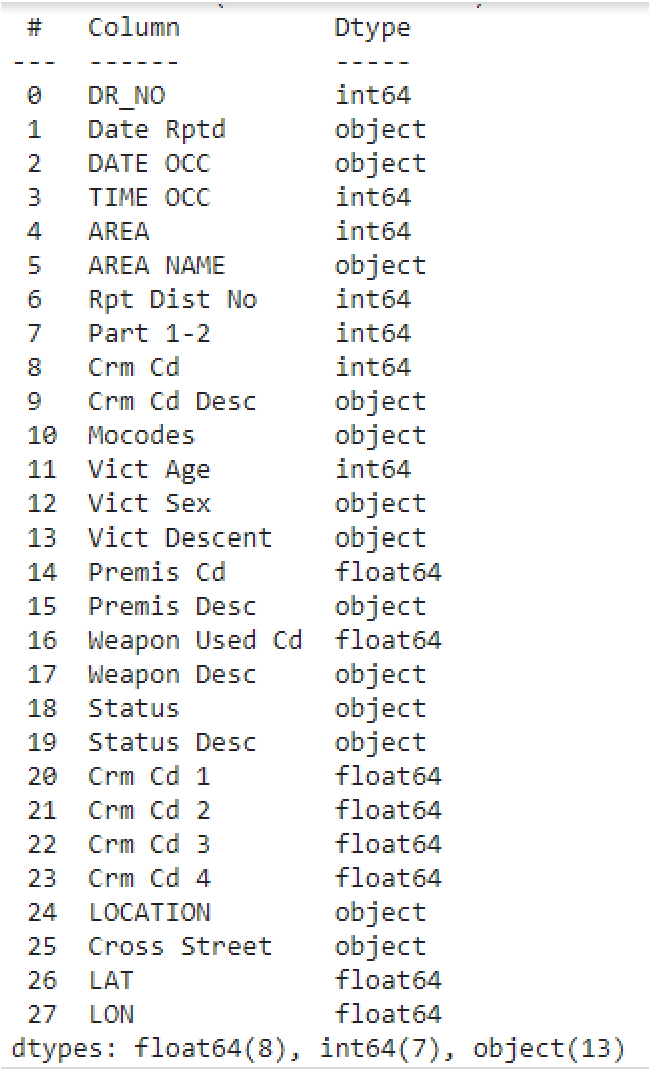


```
[ ]  # DF Shape & Size
     print("Crime Shape: ", crime.shape)
     print("Crime Size: ", crime.size)

     Crime Shape:  (2116844, 28)
     Crime Size:  59271632
```

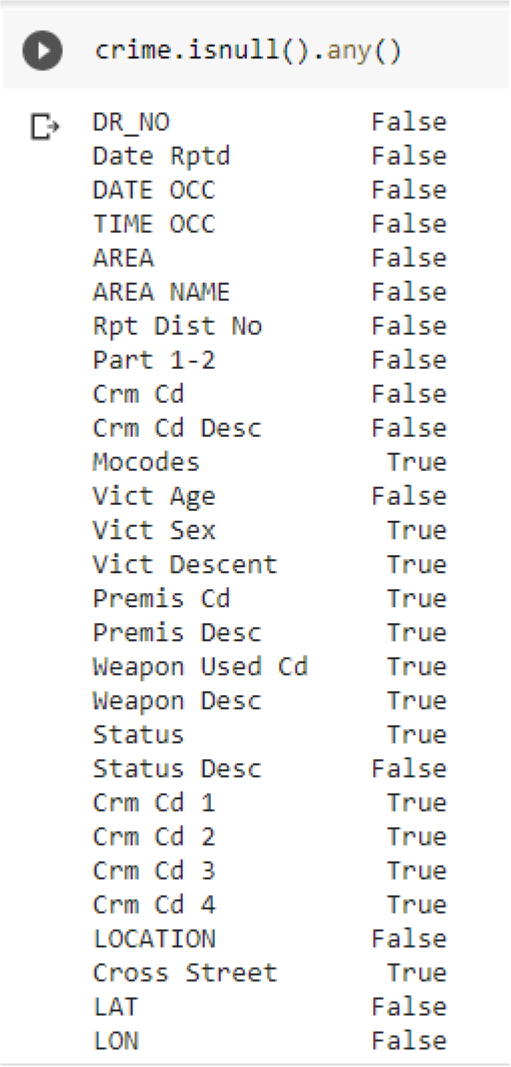


```
[ ]  # DF Columns name
     print(crime.columns)

     Index(['DR_NO', 'Date Rptd', 'DATE OCC', 'TIME OCC', 'AREA ', 'AREA NAME',
            'Rpt Dist No', 'Part 1-2', 'Crm Cd', 'Crm Cd Desc', 'Mocodes',
            'Vict Age', 'Vict Sex', 'Vict Descent', 'Premis Cd', 'Premis Desc',
            'Weapon Used Cd', 'Weapon Desc', 'Status', 'Status Desc', 'Crm Cd 1',
            'Crm Cd 2', 'Crm Cd 3', 'Crm Cd 4', 'LOCATION', 'Cross Street', 'LAT',
            'LON'],
           dtype='object')
```

3. Data preparation

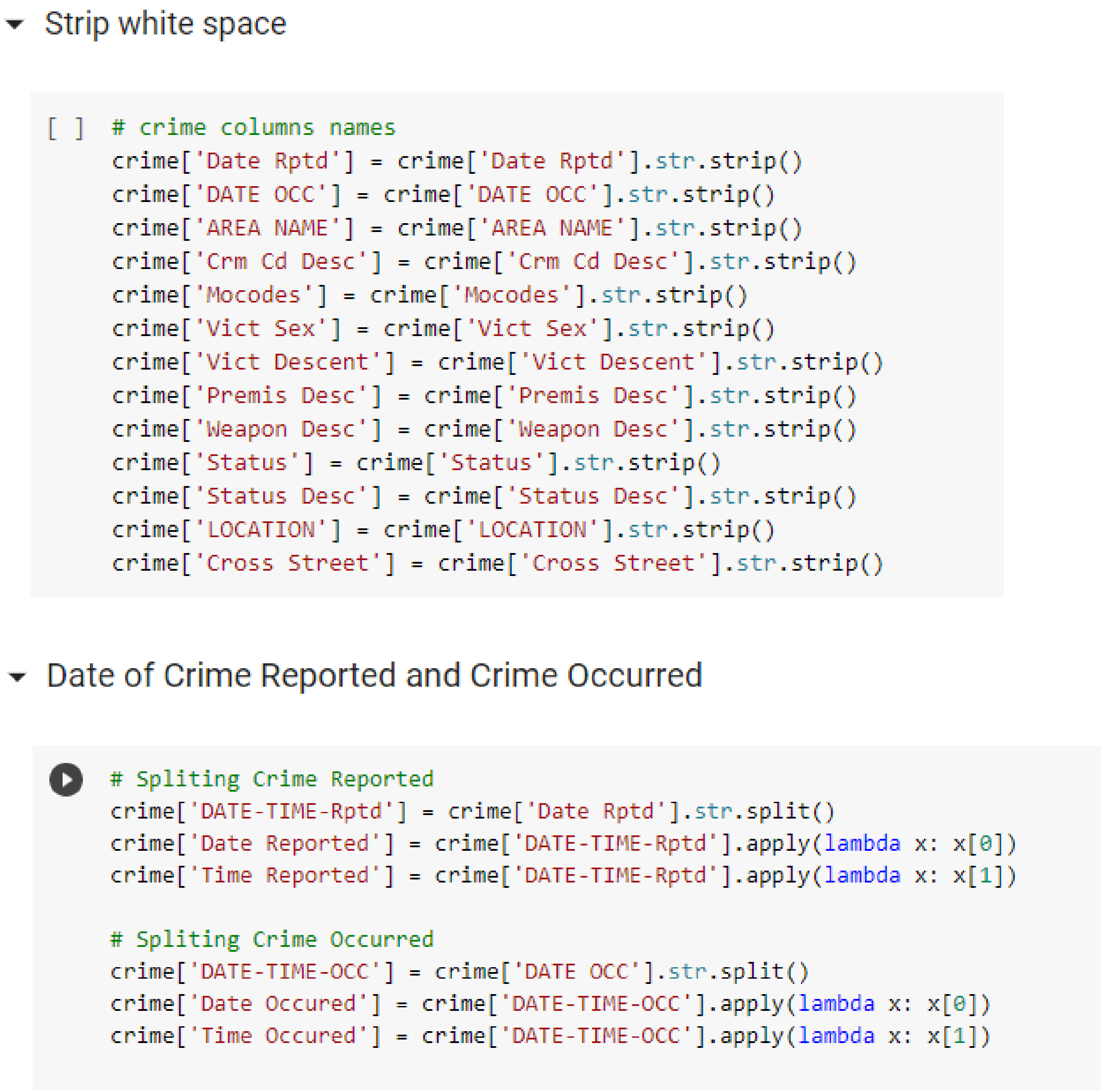


```
 #   Column          Dtype  
---  ------          -----  
 0   DR_NO           int64  
 1   Date Rptd       object 
 2   DATE OCC        object 
 3   TIME OCC        int64  
 4   AREA            int64  
 5   AREA NAME       object 
 6   Rpt Dist No     int64  
 7   Part 1-2        int64  
 8   Crm Cd          int64  
 9   Crm Cd Desc     object 
 10  Mocodes         object 
 11  Vict Age        int64  
 12  Vict Sex        object 
 13  Vict Descent    object 
 14  Premis Cd       float64
 15  Premis Desc     object 
 16  Weapon Used Cd  float64
 17  Weapon Desc     object 
 18  Status          object 
 19  Status Desc     object 
 20  Crm Cd 1        float64
 21  Crm Cd 2        float64
 22  Crm Cd 3        float64
 23  Crm Cd 4        float64
 24  LOCATION        object 
 25  Cross Street    object 
 26  LAT             float64
 27  LON             float64
dtypes: float64(8), int64(7), object(13)
```



```
crime.isnull().any()
```

```
DR_NO             False
Date Rptd         False
DATE OCC          False
TIME OCC          False
AREA              False
AREA NAME         False
Rpt Dist No       False
Part 1-2          False
Crm Cd            False
Crm Cd Desc       False
Mocodes            True
Vict Age          False
Vict Sex           True
Vict Descent       True
Premis Cd          True
Premis Desc        True
Weapon Used Cd     True
Weapon Desc        True
Status             True
Status Desc       False
Crm Cd 1           True
Crm Cd 2           True
Crm Cd 3           True
Crm Cd 4           True
LOCATION          False
Cross Street       True
LAT               False
LON               False
```

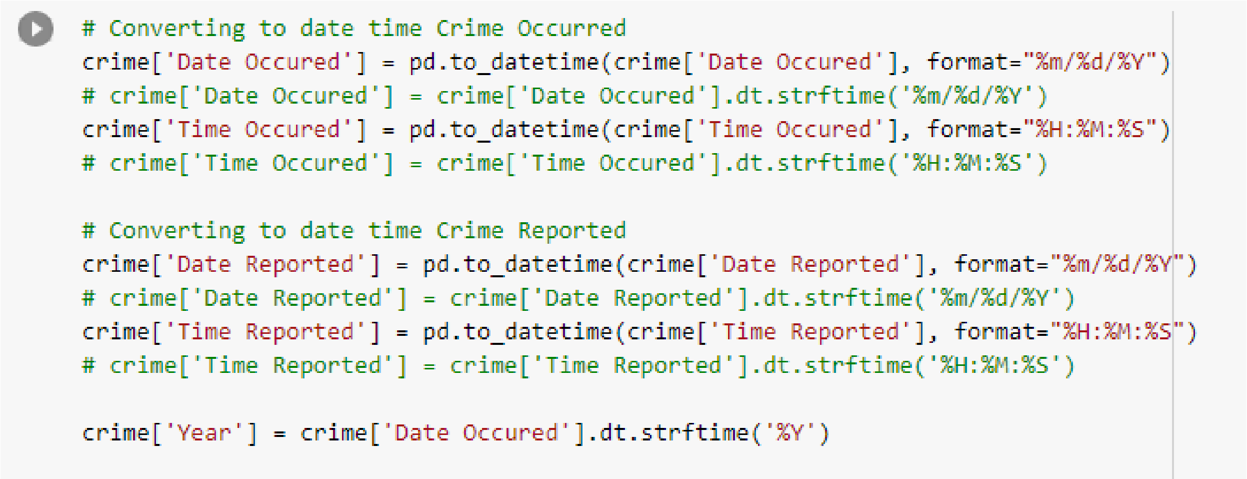


Strip white space

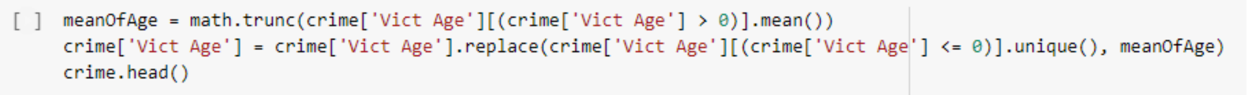
```python
# crime columns names
crime['Date Rptd'] = crime['Date Rptd'].str.strip()
crime['DATE OCC'] = crime['DATE OCC'].str.strip()
crime['AREA NAME'] = crime['AREA NAME'].str.strip()
crime['Crm Cd Desc'] = crime['Crm Cd Desc'].str.strip()
crime['Mocodes'] = crime['Mocodes'].str.strip()
crime['Vict Sex'] = crime['Vict Sex'].str.strip()
crime['Vict Descent'] = crime['Vict Descent'].str.strip()
crime['Premis Desc'] = crime['Premis Desc'].str.strip()
crime['Weapon Desc'] = crime['Weapon Desc'].str.strip()
crime['Status'] = crime['Status'].str.strip()
crime['Status Desc'] = crime['Status Desc'].str.strip()
crime['LOCATION'] = crime['LOCATION'].str.strip()
crime['Cross Street'] = crime['Cross Street'].str.strip()
```

Date of Crime Reported and Crime Occurred

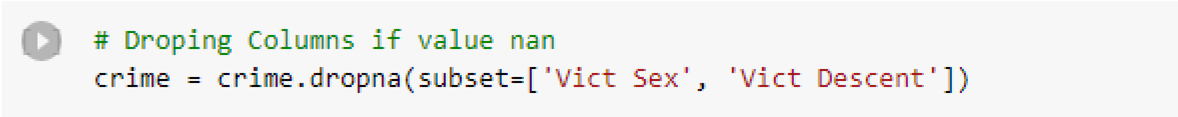
```python
# Spliting Crime Reported
crime['DATE-TIME-Rptd'] = crime['Date Rptd'].str.split()
crime['Date Reported'] = crime['DATE-TIME-Rptd'].apply(lambda x: x[0])
crime['Time Reported'] = crime['DATE-TIME-Rptd'].apply(lambda x: x[1])

# Spliting Crime Occurred
crime['DATE-TIME-OCC'] = crime['DATE OCC'].str.split()
crime['Date Occured'] = crime['DATE-TIME-OCC'].apply(lambda x: x[0])
crime['Time Occured'] = crime['DATE-TIME-OCC'].apply(lambda x: x[1])
```



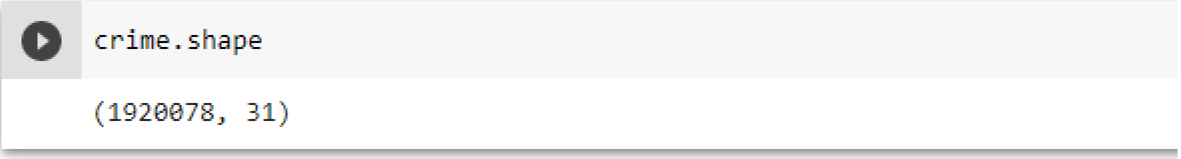
```
# Converting to date time Crime Occurred
crime['Date Occured'] = pd.to_datetime(crime['Date Occured'], format="%m/%d/%Y")
# crime['Date Occured'] = crime['Date Occured'].dt.strftime('%m/%d/%Y')
crime['Time Occured'] = pd.to_datetime(crime['Time Occured'], format="%H:%M:%S")
# crime['Time Occured'] = crime['Time Occured'].dt.strftime('%H:%M:%S')

# Converting to date time Crime Reported
crime['Date Reported'] = pd.to_datetime(crime['Date Reported'], format="%m/%d/%Y")
# crime['Date Reported'] = crime['Date Reported'].dt.strftime('%m/%d/%Y')
crime['Time Reported'] = pd.to_datetime(crime['Time Reported'], format="%H:%M:%S")
# crime['Time Reported'] = crime['Time Reported'].dt.strftime('%H:%M:%S')

crime['Year'] = crime['Date Occured'].dt.strftime('%Y')
```



```
meanOfAge = math.trunc(crime['Vict Age'][(crime['Vict Age'] > 0)].mean())
crime['Vict Age'] = crime['Vict Age'].replace(crime['Vict Age'][(crime['Vict Age'] <= 0)].unique(), meanOfAge)
crime.head()
```

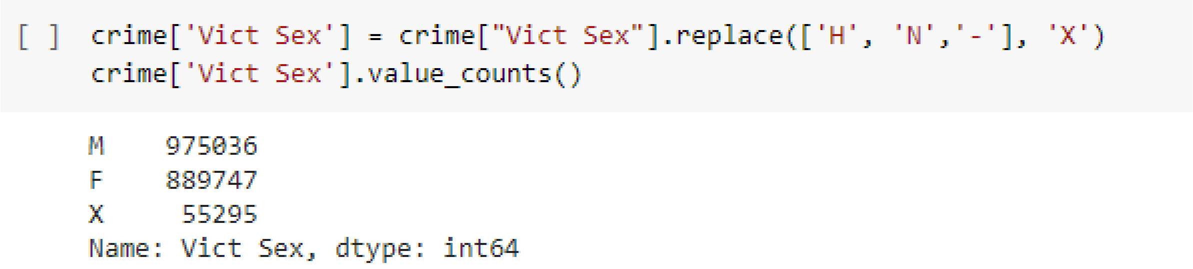


```
# Droping Columns if value nan
crime = crime.dropna(subset=['Vict Sex', 'Vict Descent'])
```



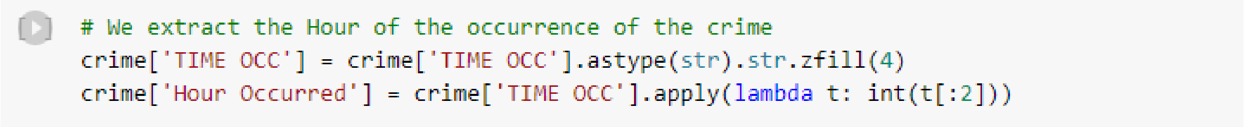
```
crime.shape
```

```
(1920078, 31)
```

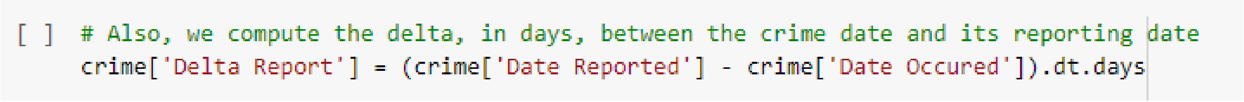


```
crime['Vict Sex'] = crime["Vict Sex"].replace(['H', 'N','-'], 'X')
crime['Vict Sex'].value_counts()
```

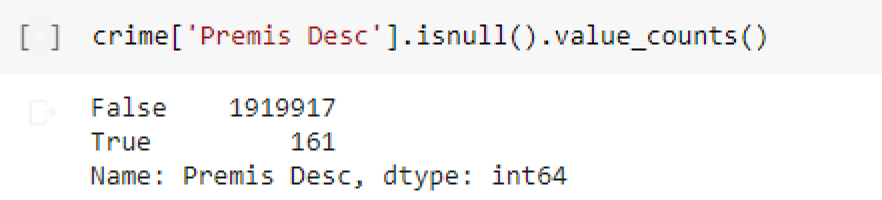
```
M    975036
F    889747
X     55295
Name: Vict Sex, dtype: int64
```



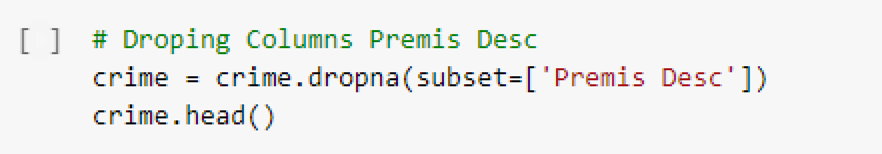
```
# We extract the Hour of the occurrence of the crime
crime['TIME OCC'] = crime['TIME OCC'].astype(str).str.zfill(4)
crime['Hour Occurred'] = crime['TIME OCC'].apply(lambda t: int(t[:2]))
```



```
# Also, we compute the delta, in days, between the crime date and its reporting date
crime['Delta Report'] = (crime['Date Reported'] - crime['Date Occured']).dt.days
```

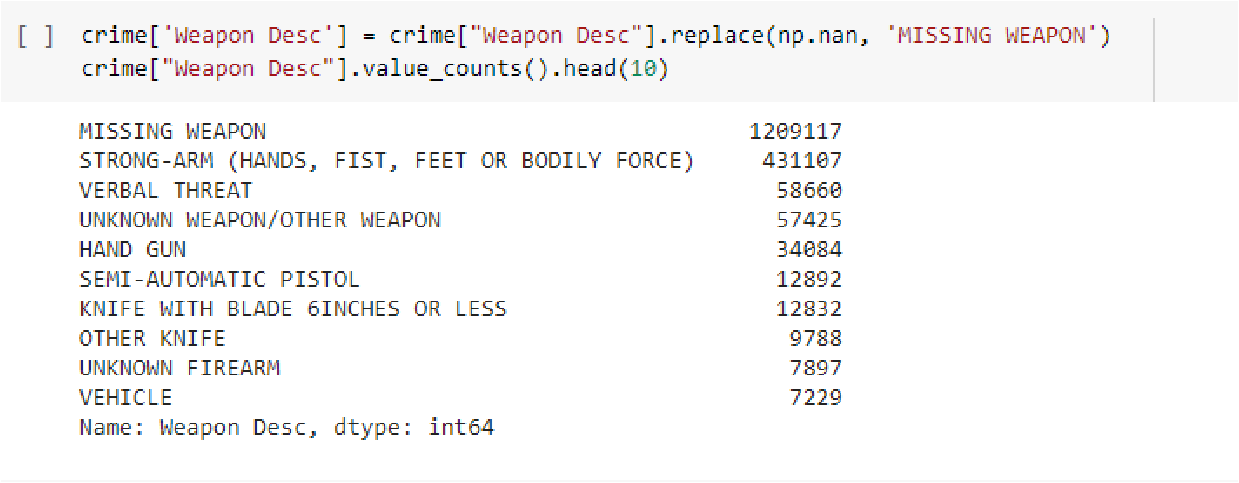


```
[ ]  crime['Premis Desc'].isnull().value_counts()

     False    1919917
     True         161
     Name: Premis Desc, dtype: int64
```



```
[ ]  # Droping Columns Premis Desc
     crime = crime.dropna(subset=['Premis Desc'])
     crime.head()
```

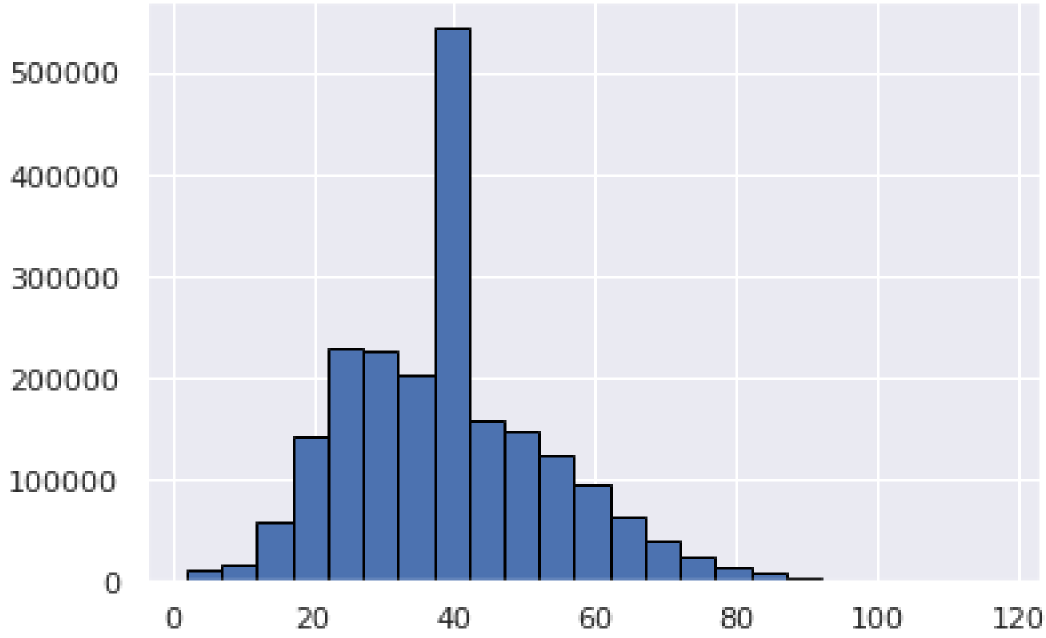
We assume that missing values are Weapon Desc is Missing Weapon.

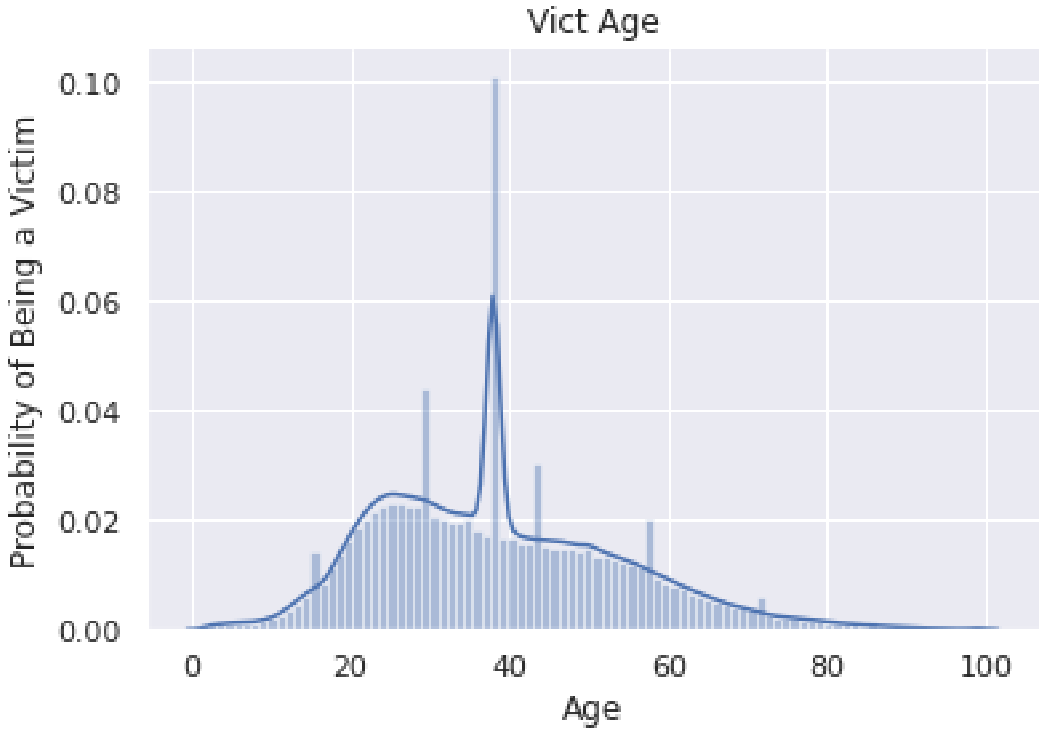


```
[ ]  crime['Weapon Desc'] = crime["Weapon Desc"].replace(np.nan, 'MISSING WEAPON')
     crime["Weapon Desc"].value_counts().head(10)

     MISSING WEAPON                                    1209117
     STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)     431107
     VERBAL THREAT                                       58660
     UNKNOWN WEAPON/OTHER WEAPON                         57425
     HAND GUN                                            34084
     SEMI-AUTOMATIC PISTOL                               12892
     KNIFE WITH BLADE 6INCHES OR LESS                    12832
     OTHER KNIFE                                          9788
     UNKNOWN FIREARM                                      7897
     VEHICLE                                              7229
     Name: Weapon Desc, dtype: int64
```

4. Data exploration

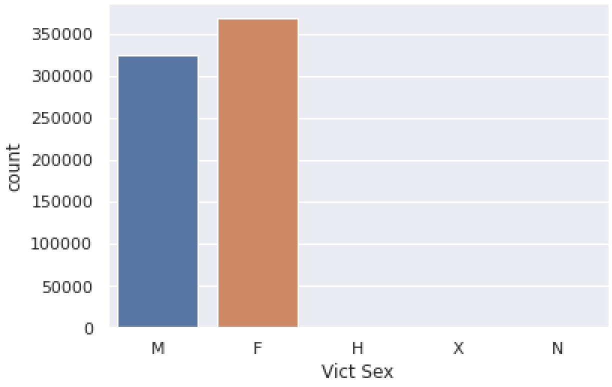
Age of Victim when they encountered crimes





```
# Number of Victims according to Sex of the Victim b/w the age group 20 and 35 years
sns.countplot('Vict Sex', data=victim[(victim['Vict Age'] >= 20) & (victim['Vict Age'] <= 35)])
```

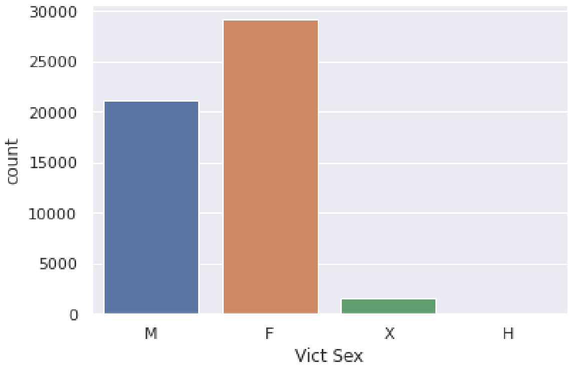
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following va
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f1dbfca58d0>
```



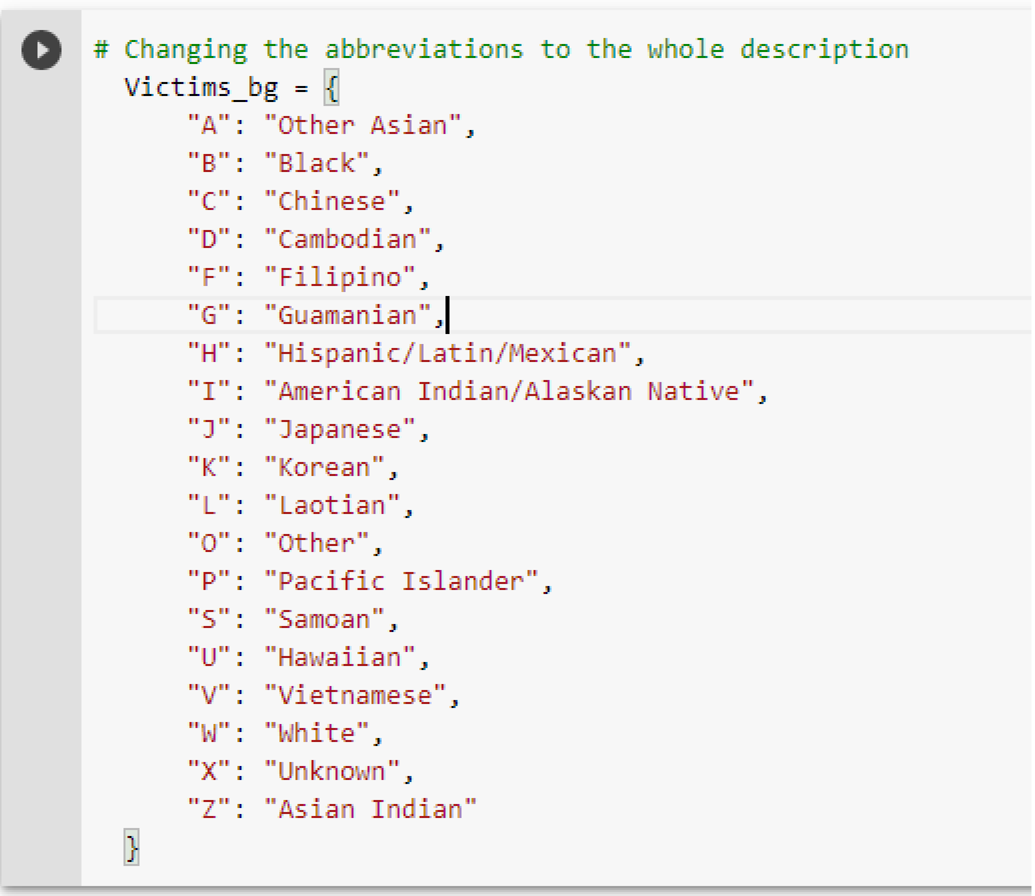
As we can see, Females in the age group 20-35 Years are being targeted more.

```
# Number of Victims according to Sex of the Victim b/w the age group 10 and 15 years
sns.countplot('Vict Sex', data=victim[(victim['Vict Age'] >= 10) & (victim['Vict Age'] <= 15)])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variab
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f1dbfca2b50>
```

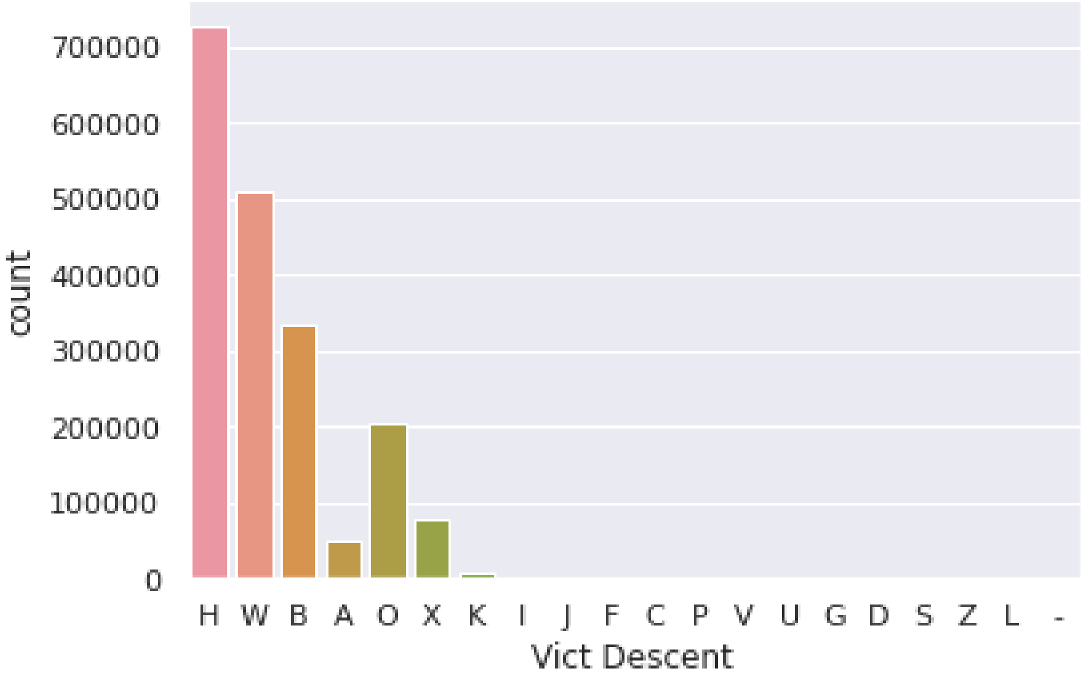


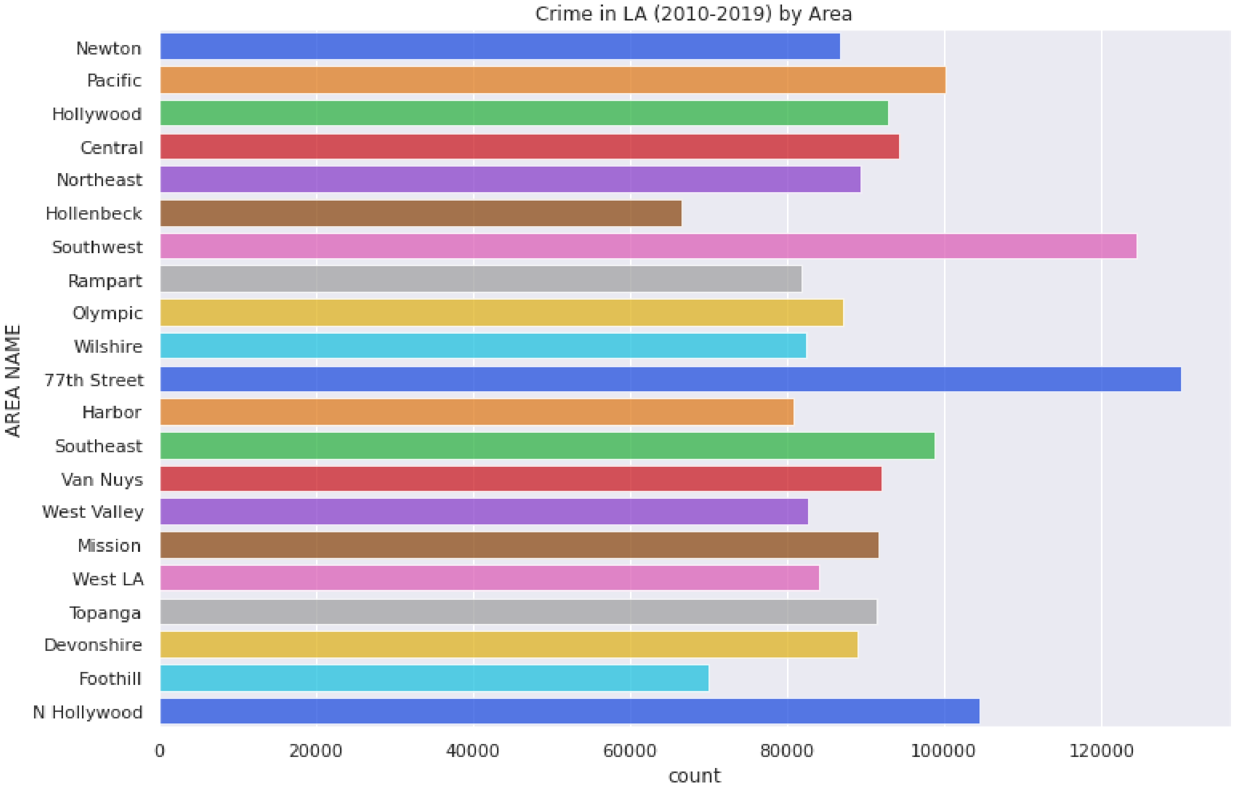
Boys in their have been target more much more

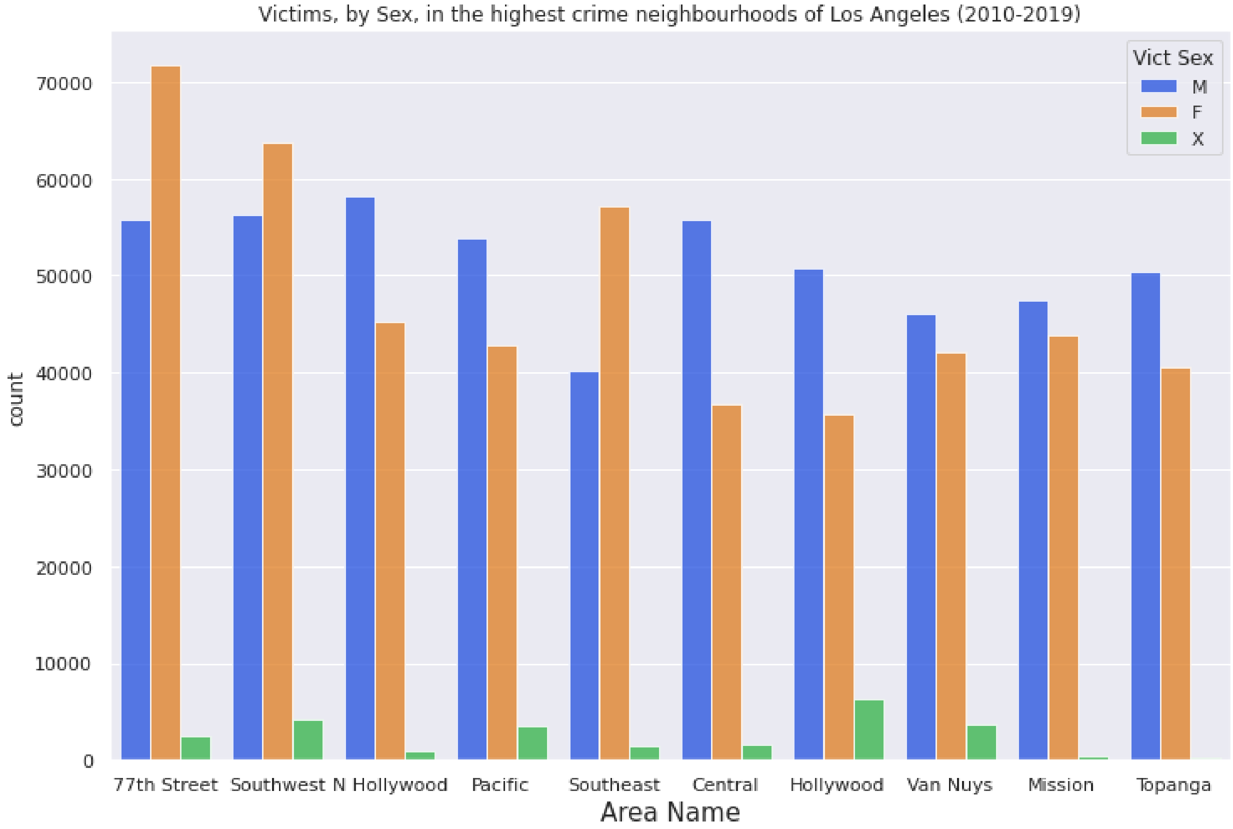


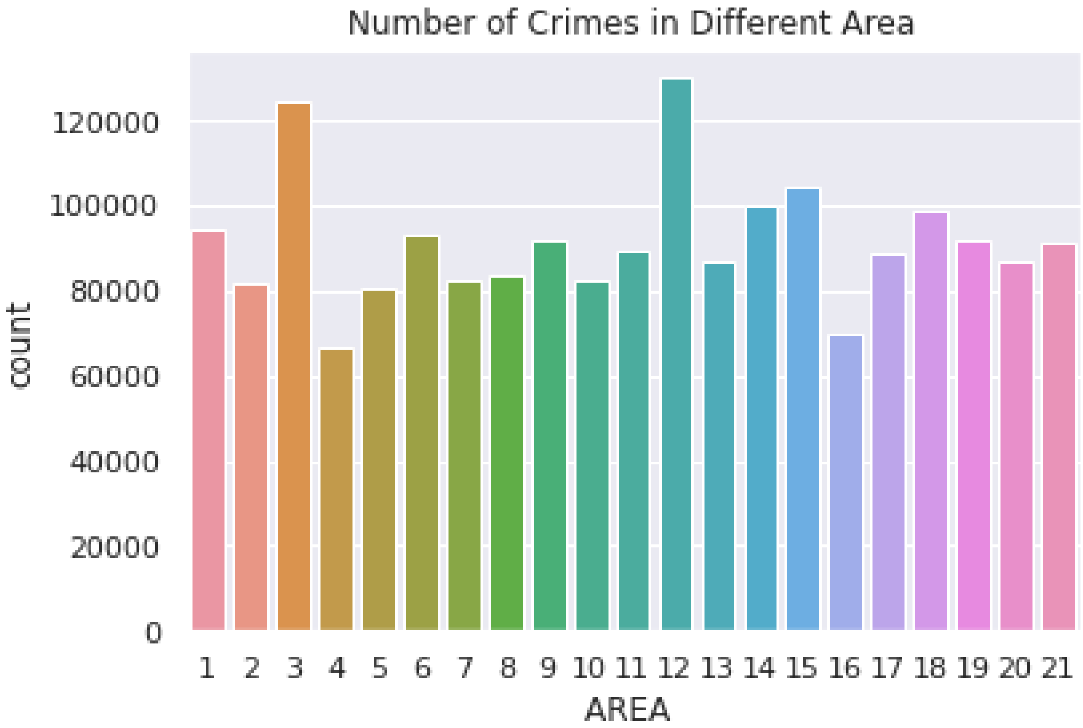
```
# Changing the abbreviations to the whole description
  Victims_bg = {
      "A": "Other Asian",
      "B": "Black",
      "C": "Chinese",
      "D": "Cambodian",
      "F": "Filipino",
      "G": "Guamanian",
      "H": "Hispanic/Latin/Mexican",
      "I": "American Indian/Alaskan Native",
      "J": "Japanese",
      "K": "Korean",
      "L": "Laotian",
      "O": "Other",
      "P": "Pacific Islander",
      "S": "Samoan",
      "U": "Hawaiian",
      "V": "Vietnamese",
      "W": "White",
      "X": "Unknown",
      "Z": "Asian Indian"
  }
```

People who were from the Descent H has been targeted the most followed by W and B.

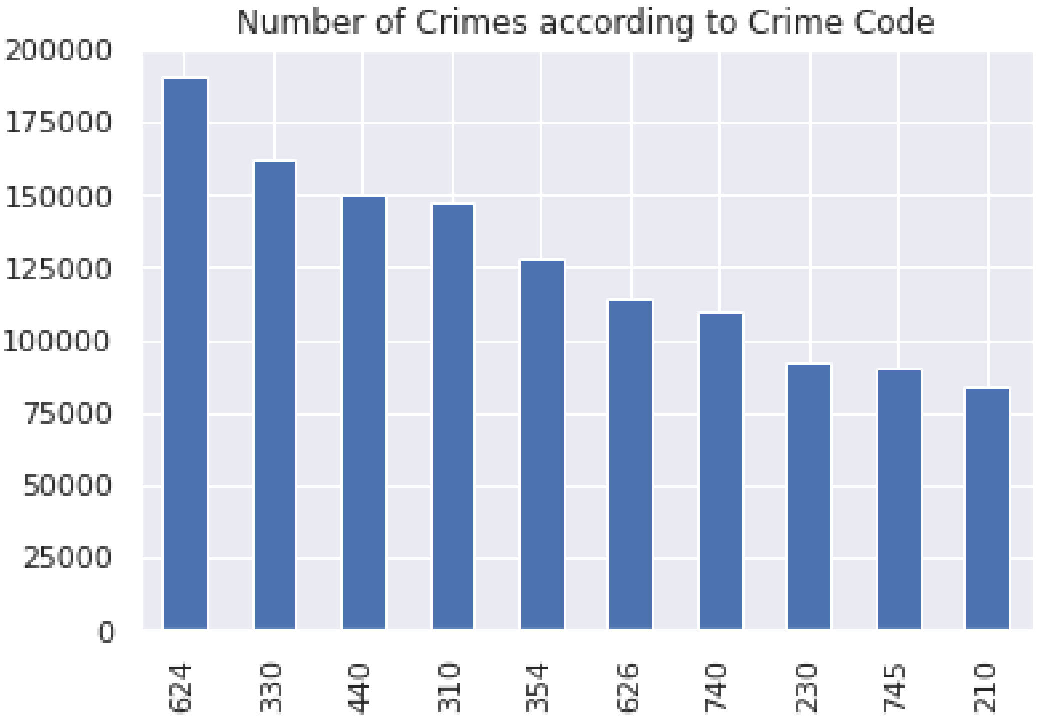


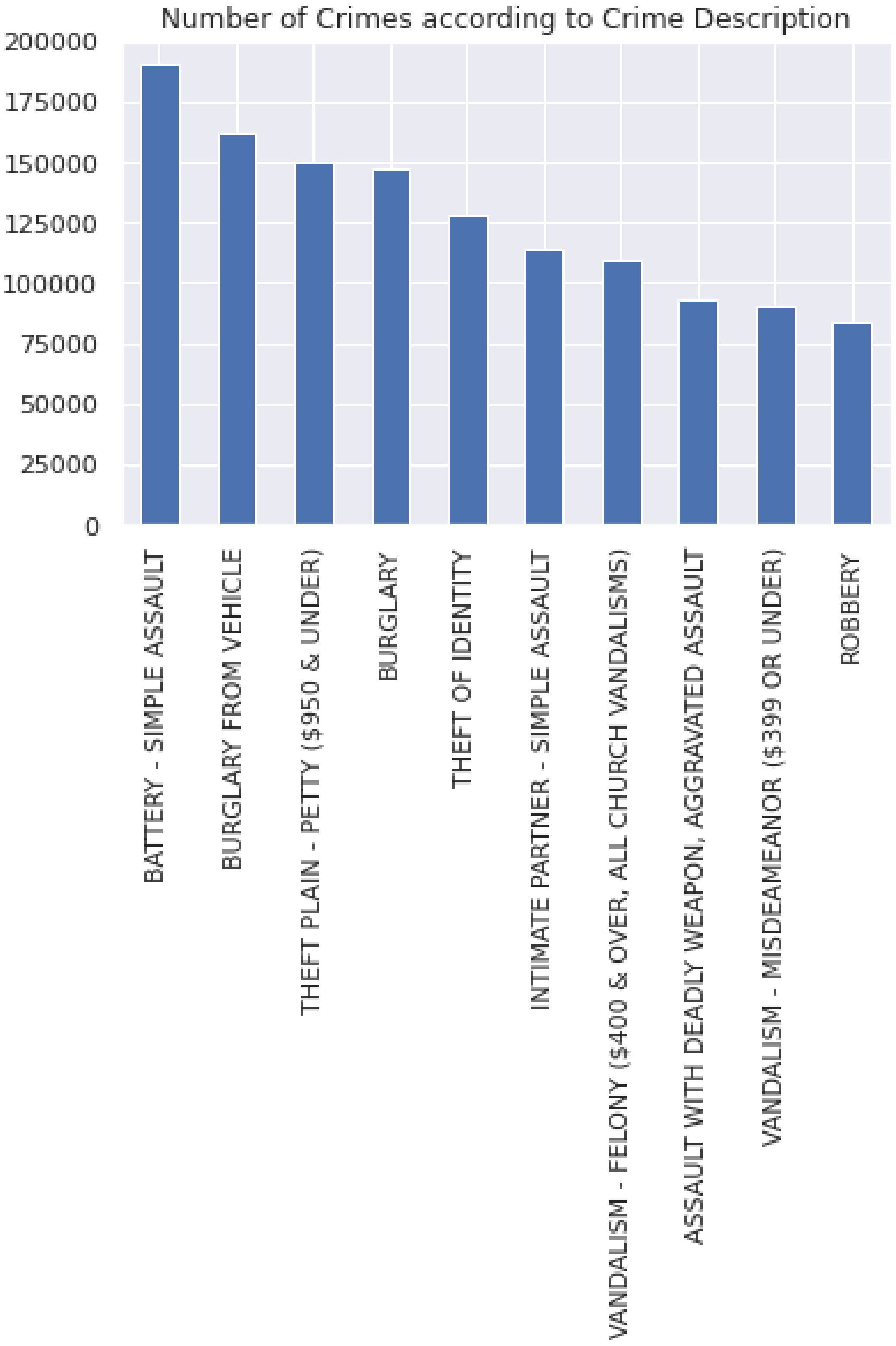

Crime in LA (2010-2019) by Area
AREA NAME
Newton
Pacific
Hollywood
Central
Northeast
Hollenbeck
Southwest
Rampart
Olympic
Wilshire
77th Street
Harbor
Southeast
Van Nuys
West Valley
Mission
West LA
Topanga
Devonshire
Foothill
N Hollywood
0
20000
40000
60000
80000
100000
120000
count


Victims, by Sex, in the highest crime neighbourhoods of Los Angeles (2010-2019)
Vict Sex
M
F
X
count
0
10000
20000
30000
40000
50000
60000
70000
77th Street
Southwest
N Hollywood
Pacific
Southeast
Central
Hollywood
Van Nuys
Mission
Topanga
Area Name

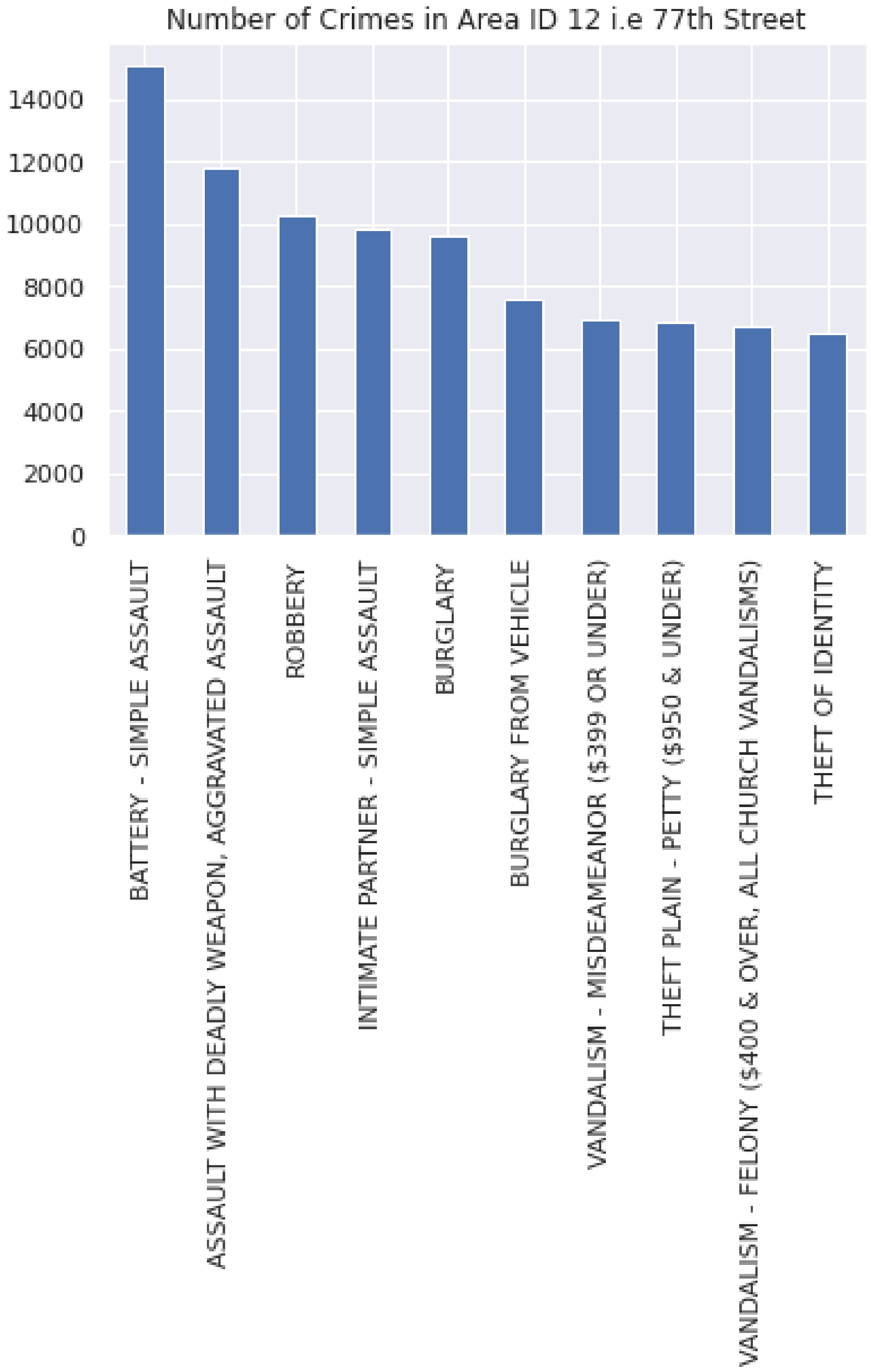


Area ID 12 is 77th Street Area ID 3 is Southwest

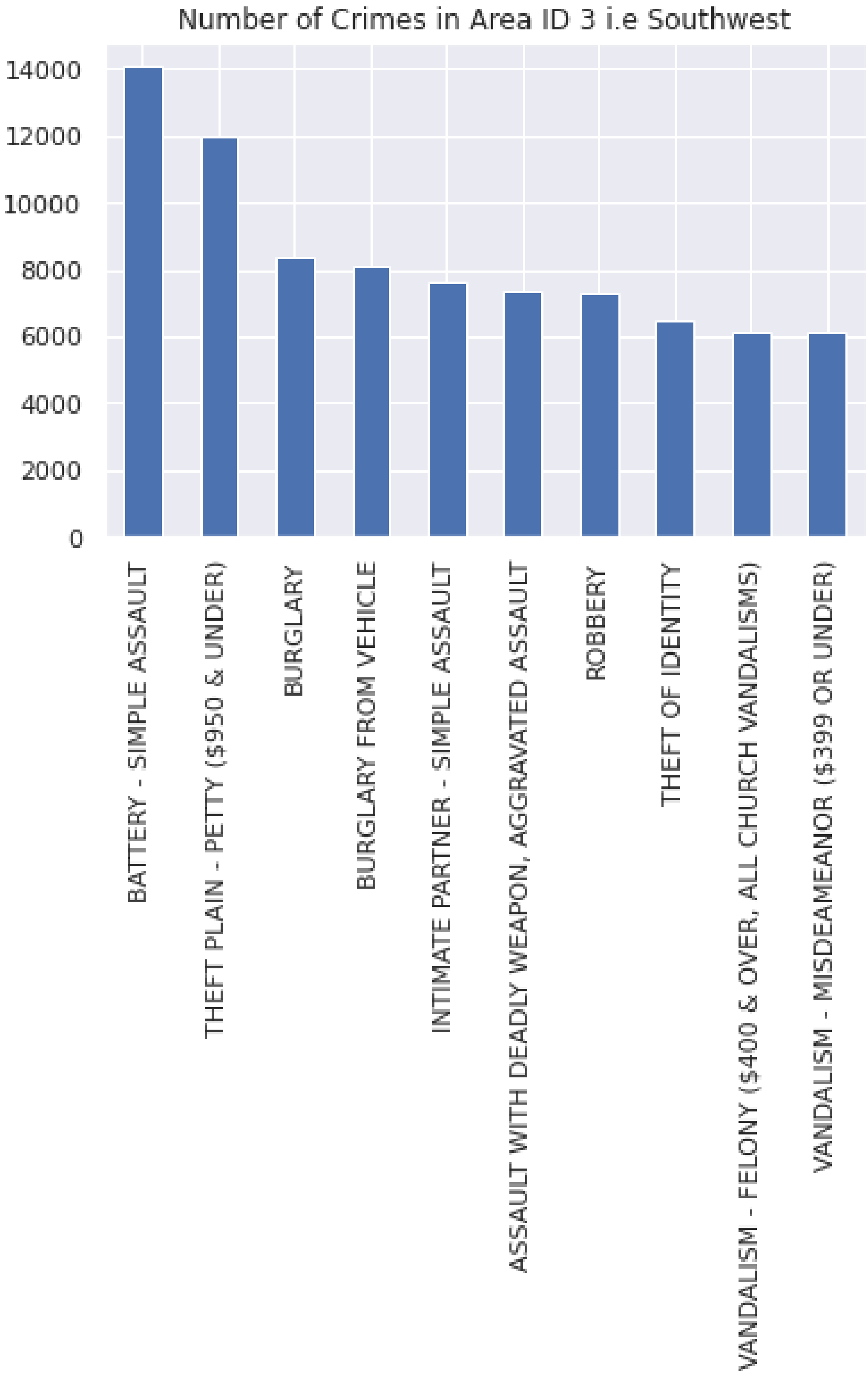


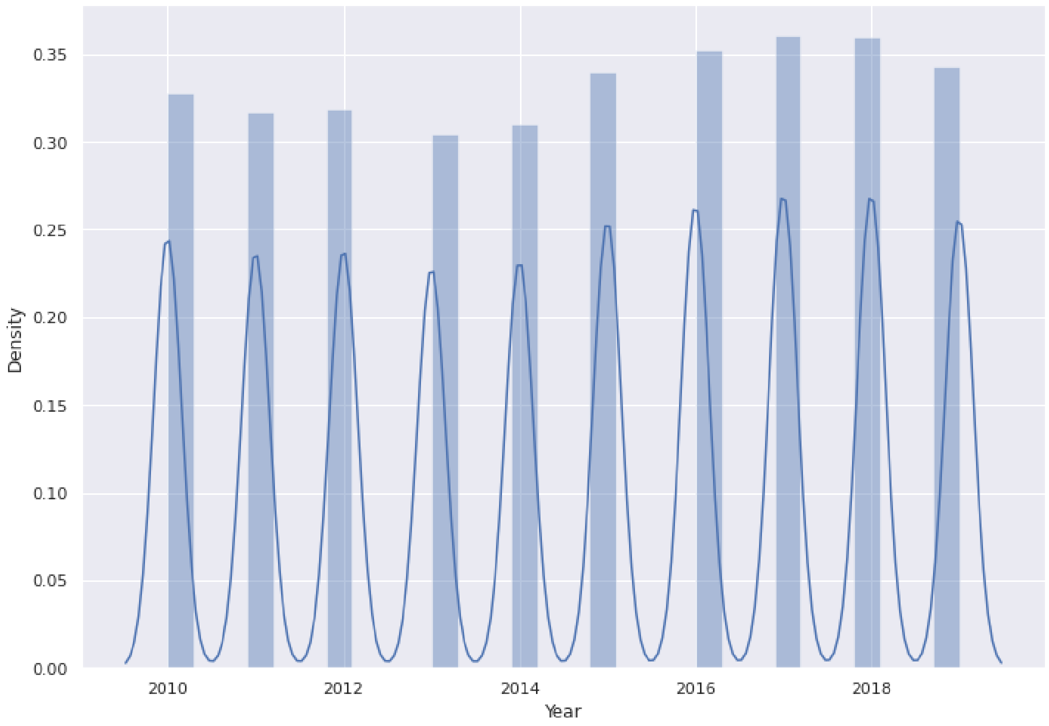


According to the Graph, BATTERY - SIMPLE ASSAULT has the Highest Crime Rate followed by BURGLARY FROM VEHICLE

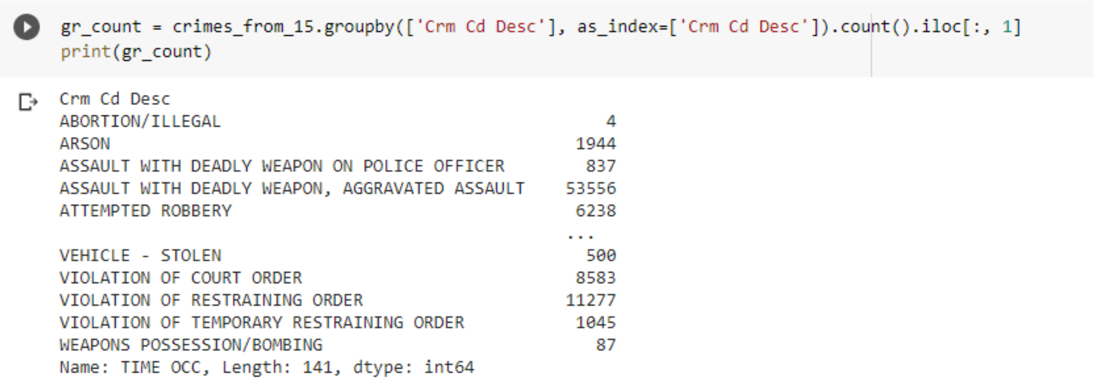


Number of Crimes in Area ID 12 i.e 77th Street


Number of Crimes in Area ID 3 i.e Southwest
14000
12000
10000
8000
6000
4000
2000
0
BATTERY - SIMPLE ASSAULT
THEFT PLAIN - PETTY ($950 & UNDER)
BURGLARY
BURGLARY FROM VEHICLE
INTIMATE PARTNER - SIMPLE ASSAULT
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
ROBBERY
THEFT OF IDENTITY
VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
VANDALISM - MISDEAMEANOR ($399 OR UNDER)

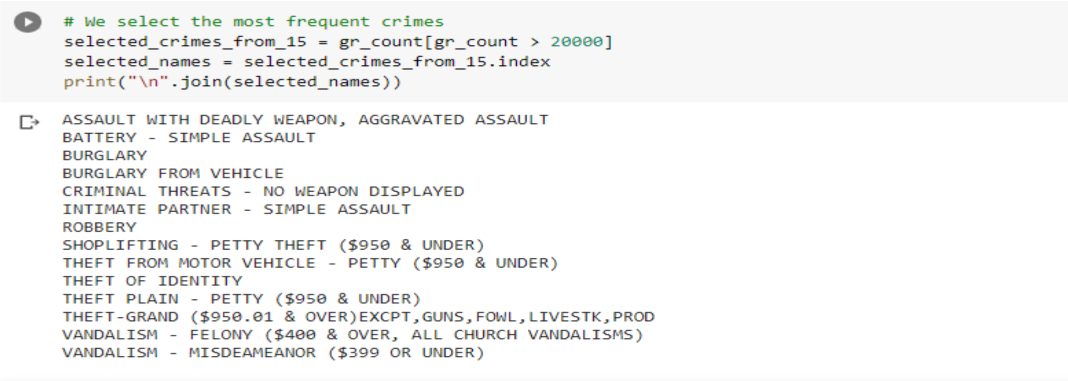


We analyze crimes from 2015



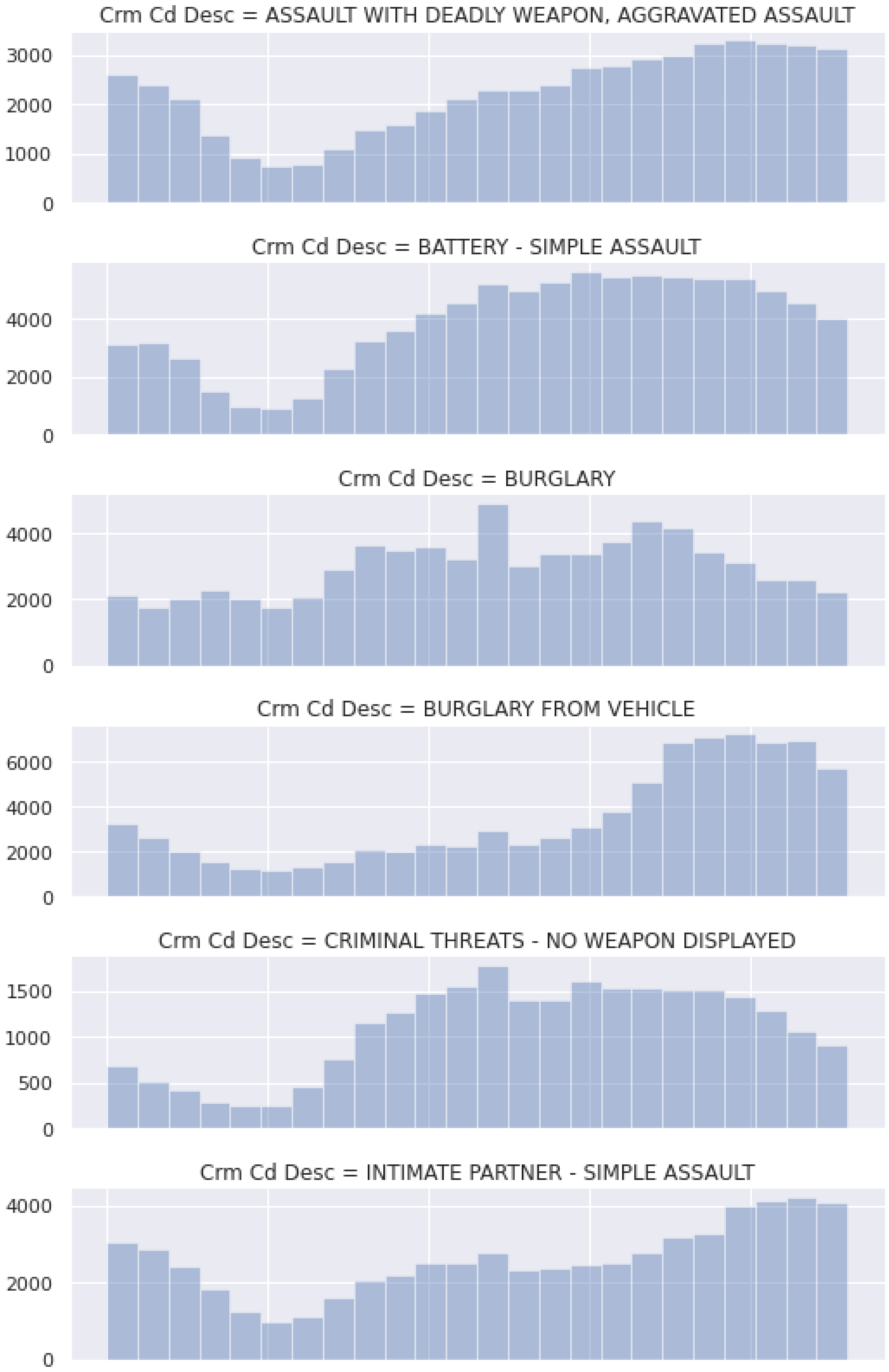
```
gr_count = crimes_from_15.groupby(['Crm Cd Desc'], as_index=['Crm Cd Desc']).count().iloc[:, 1]
print(gr_count)
```

```
Crm Cd Desc
ABORTION/ILLEGAL                                      4
ARSON                                              1944
ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER        837
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT    53556
ATTEMPTED ROBBERY                                  6238
                                                 ...
VEHICLE - STOLEN                                    500
VIOLATION OF COURT ORDER                           8583
VIOLATION OF RESTRAINING ORDER                    11277
VIOLATION OF TEMPORARY RESTRAINING ORDER           1045
WEAPONS POSSESSION/BOMBING                           87
Name: TIME OCC, Length: 141, dtype: int64
```



```
# We select the most frequent crimes
selected_crimes_from_15 = gr_count[gr_count > 20000]
selected_names = selected_crimes_from_15.index
print("\n".join(selected_names))
```

```
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
BATTERY - SIMPLE ASSAULT
BURGLARY
BURGLARY FROM VEHICLE
CRIMINAL THREATS - NO WEAPON DISPLAYED
INTIMATE PARTNER - SIMPLE ASSAULT
ROBBERY
SHOPLIFTING - PETTY THEFT ($950 & UNDER)
THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)
THEFT OF IDENTITY
THEFT PLAIN - PETTY ($950 & UNDER)
THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LIVESTK,PROD
VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
VANDALISM - MISDEAMEANOR ($399 OR UNDER)
```



Crm Cd Desc = ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT

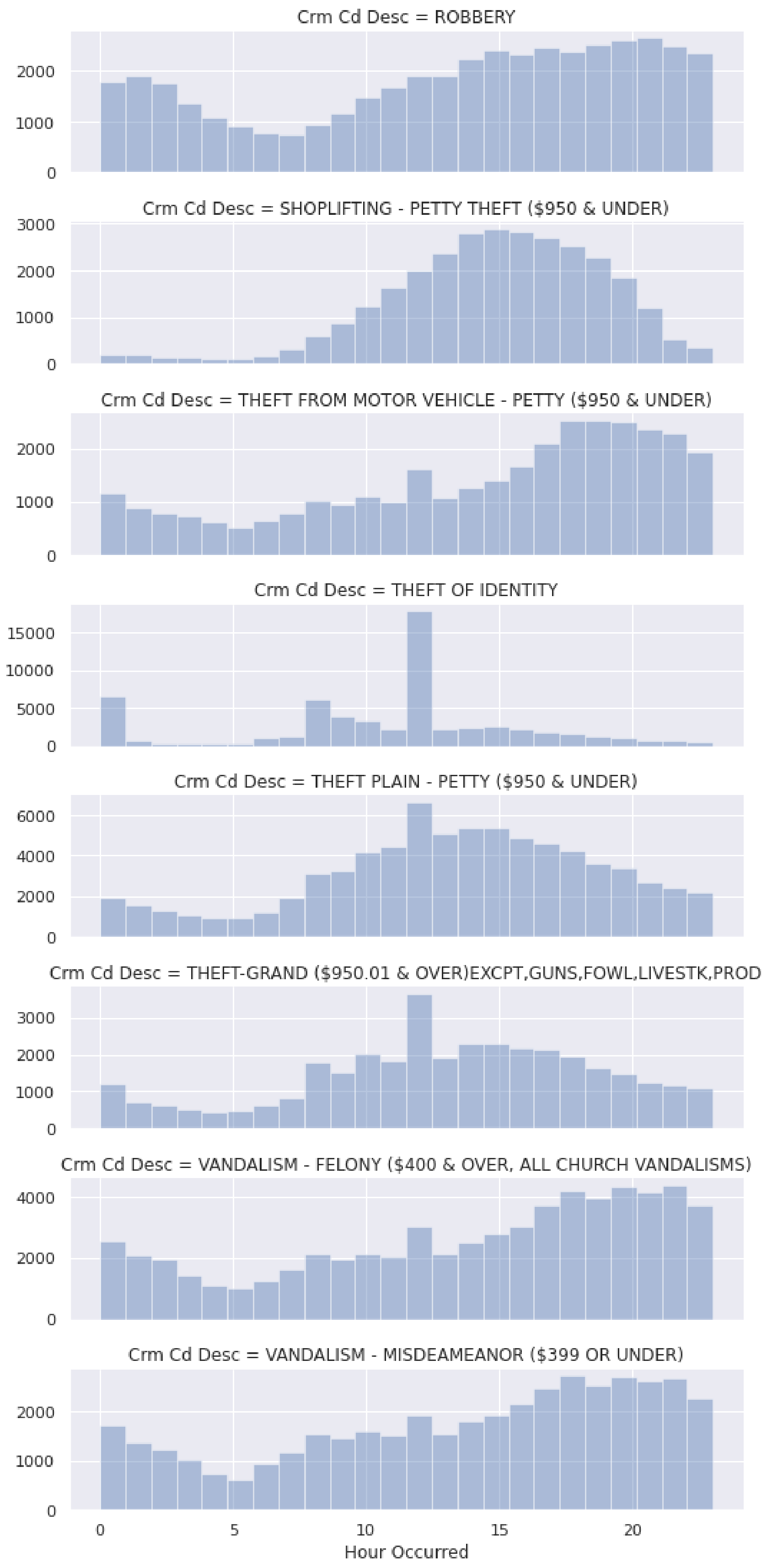
Crm Cd Desc = BATTERY - SIMPLE ASSAULT

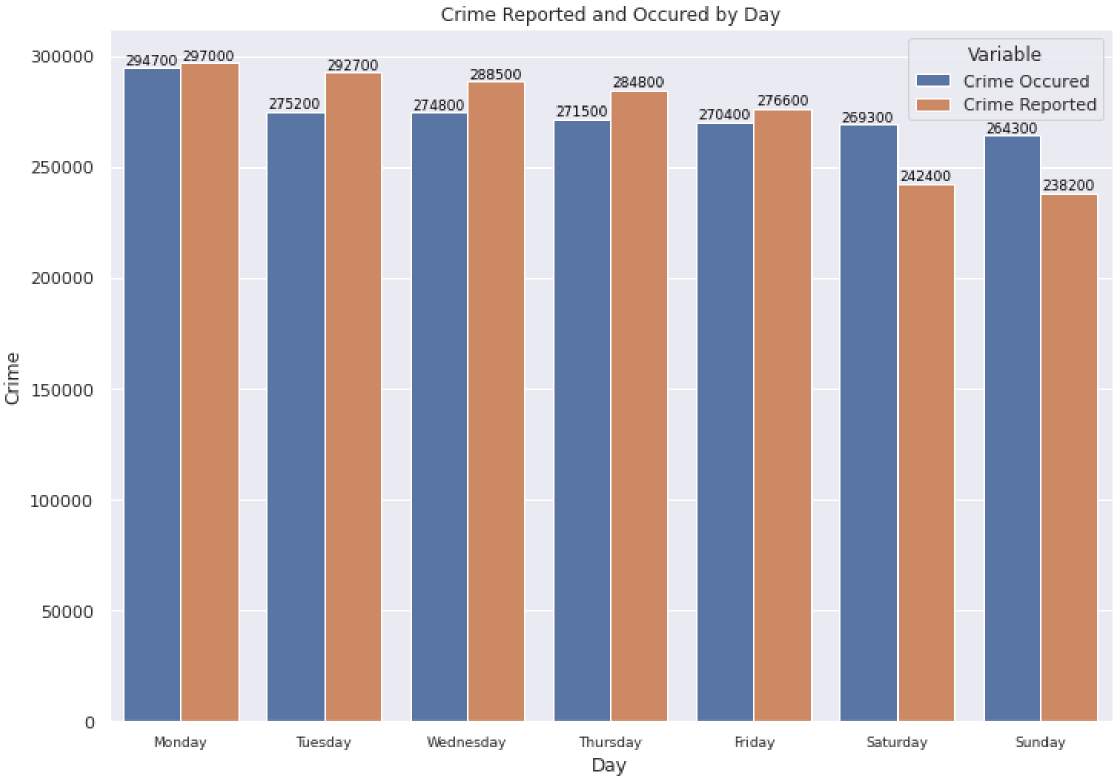
Crm Cd Desc = BURGLARY

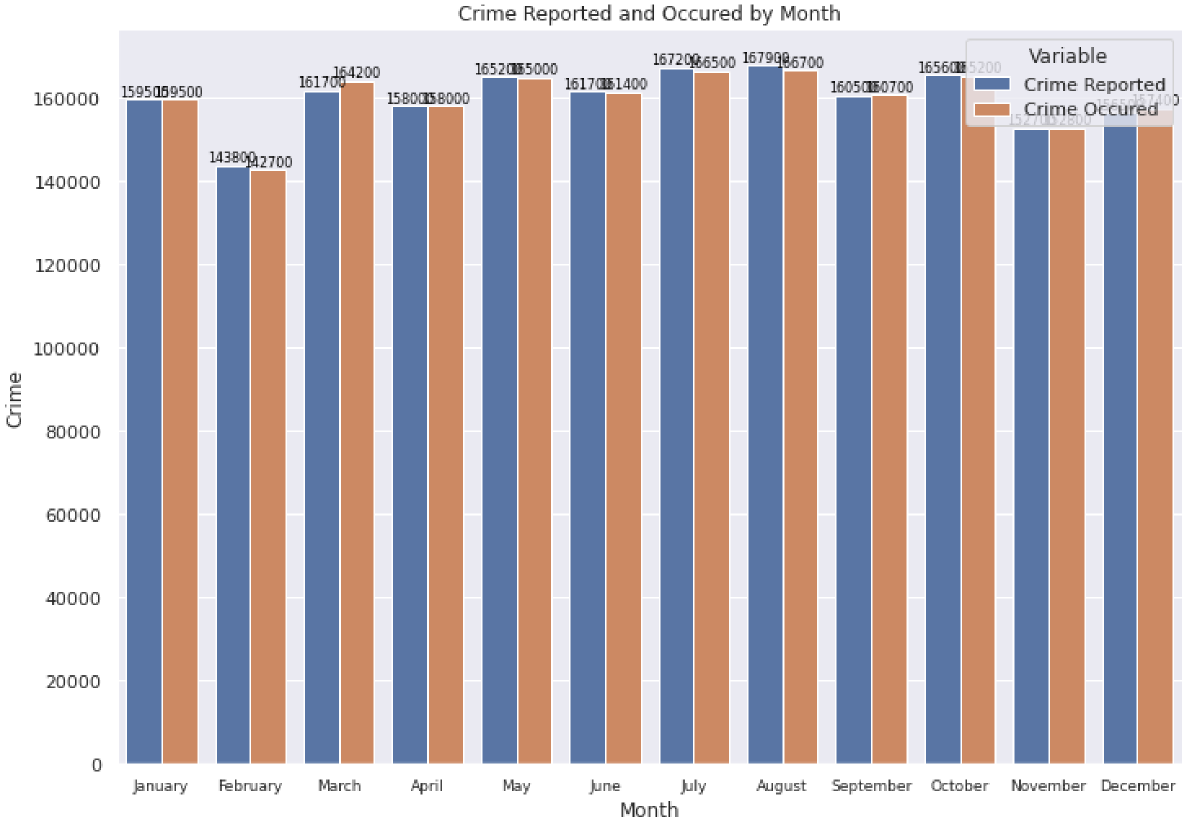
Crm Cd Desc = BURGLARY FROM VEHICLE

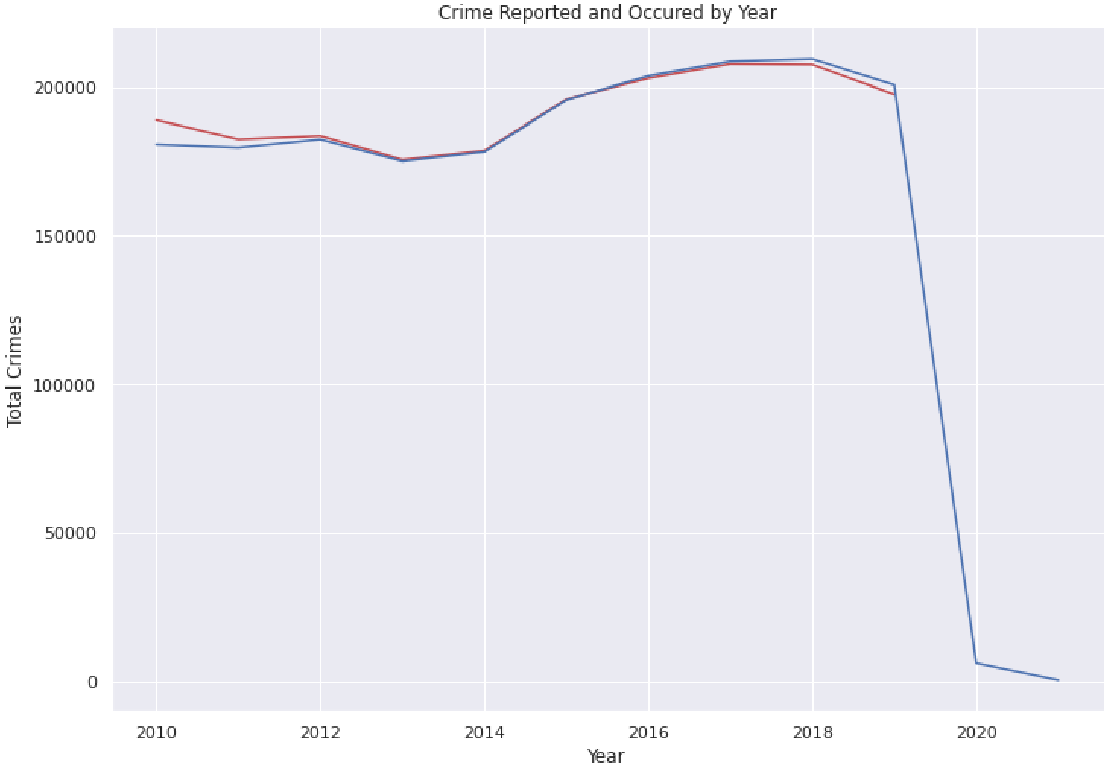
Crm Cd Desc = CRIMINAL THREATS - NO WEAPON DISPLAYED

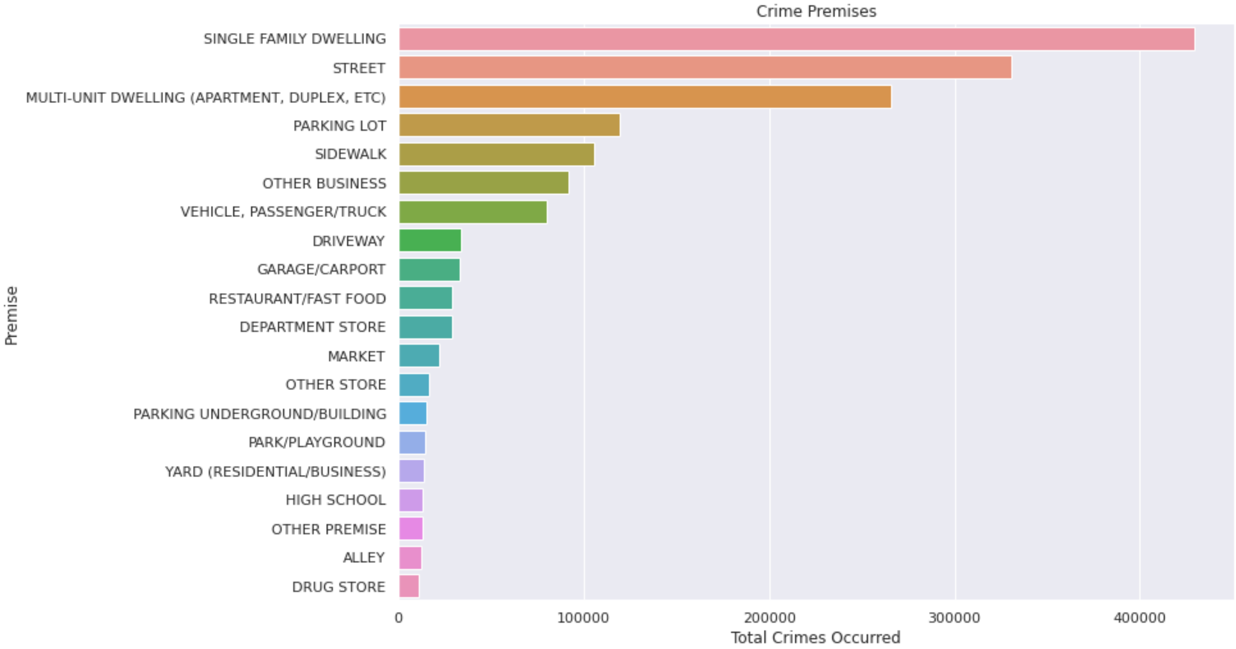
Crm Cd Desc = INTIMATE PARTNER - SIMPLE ASSAULT

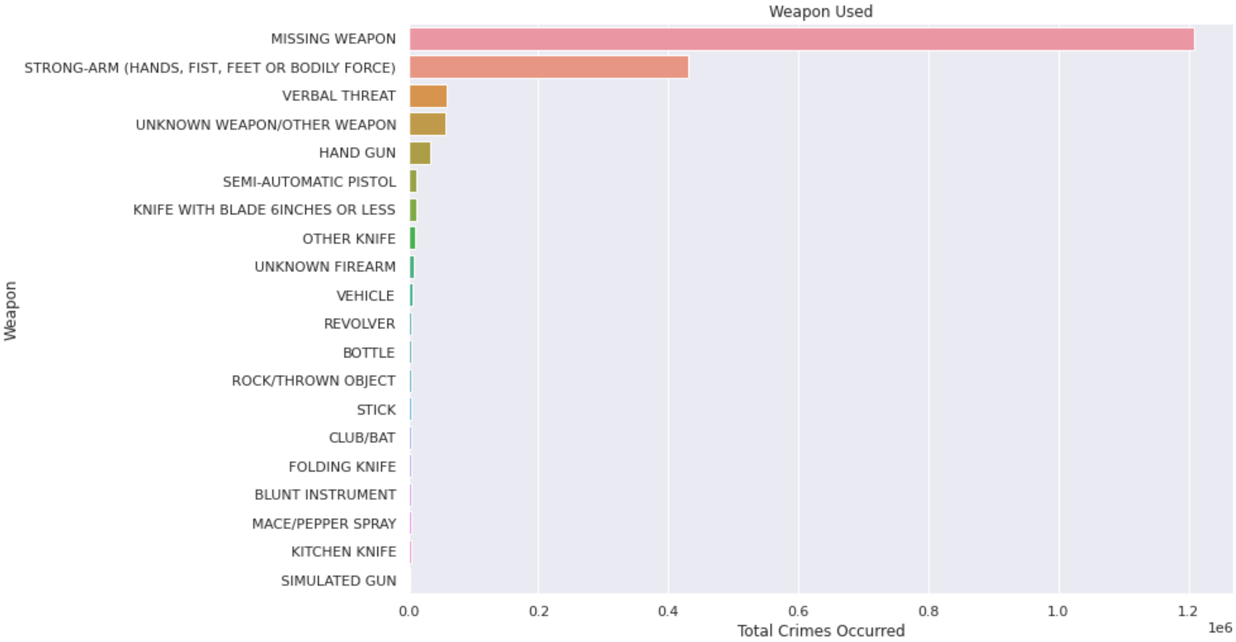

Crm Cd Desc = ROBBERY
0
1000
2000
Crm Cd Desc = SHOPLIFTING - PETTY THEFT ($950 & UNDER)
0
1000
2000
3000
Crm Cd Desc = THEFT FROM MOTOR VEHICLE - PETTY ($950 & UNDER)
0
1000
2000
Crm Cd Desc = THEFT OF IDENTITY
0
5000
10000
15000
Crm Cd Desc = THEFT PLAIN - PETTY ($950 & UNDER)
0
2000
4000
6000
Crm Cd Desc = THEFT-GRAND ($950.01 & OVER)EXCPT,GUNS,FOWL,LIVESTK,PROD
0
1000
2000
3000
Crm Cd Desc = VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
0
2000
4000
Crm Cd Desc = VANDALISM - MISDEAMEANOR ($399 OR UNDER)
0
1000
2000
0
5
10
15
20
Hour Occurred

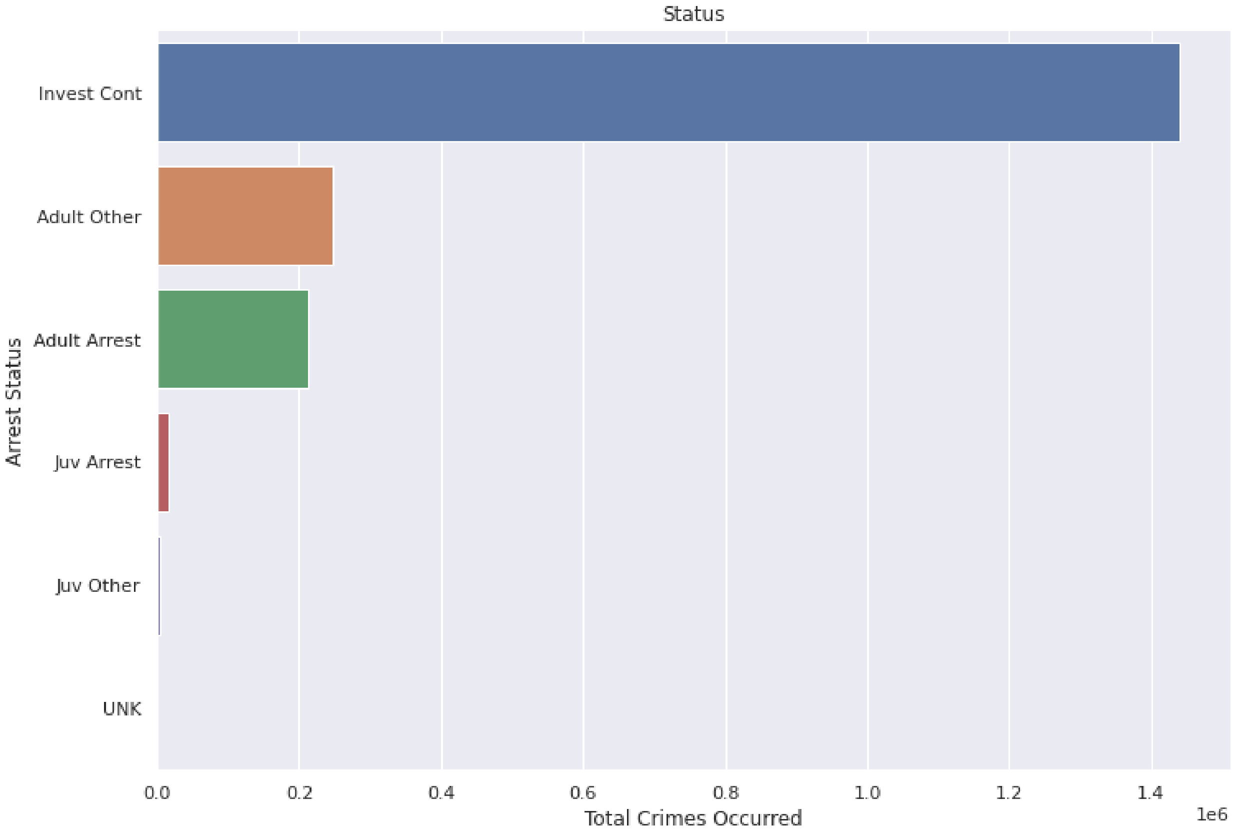

Crime Reported and Occured by Day
Variable
Crime Occured
Crime Reported
Crime
Day
0
50000
100000
150000
200000
250000
300000
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
294700
297000
275200
292700
274800
288500
271500
284800
270400
276600
269300
242400
264300
238200

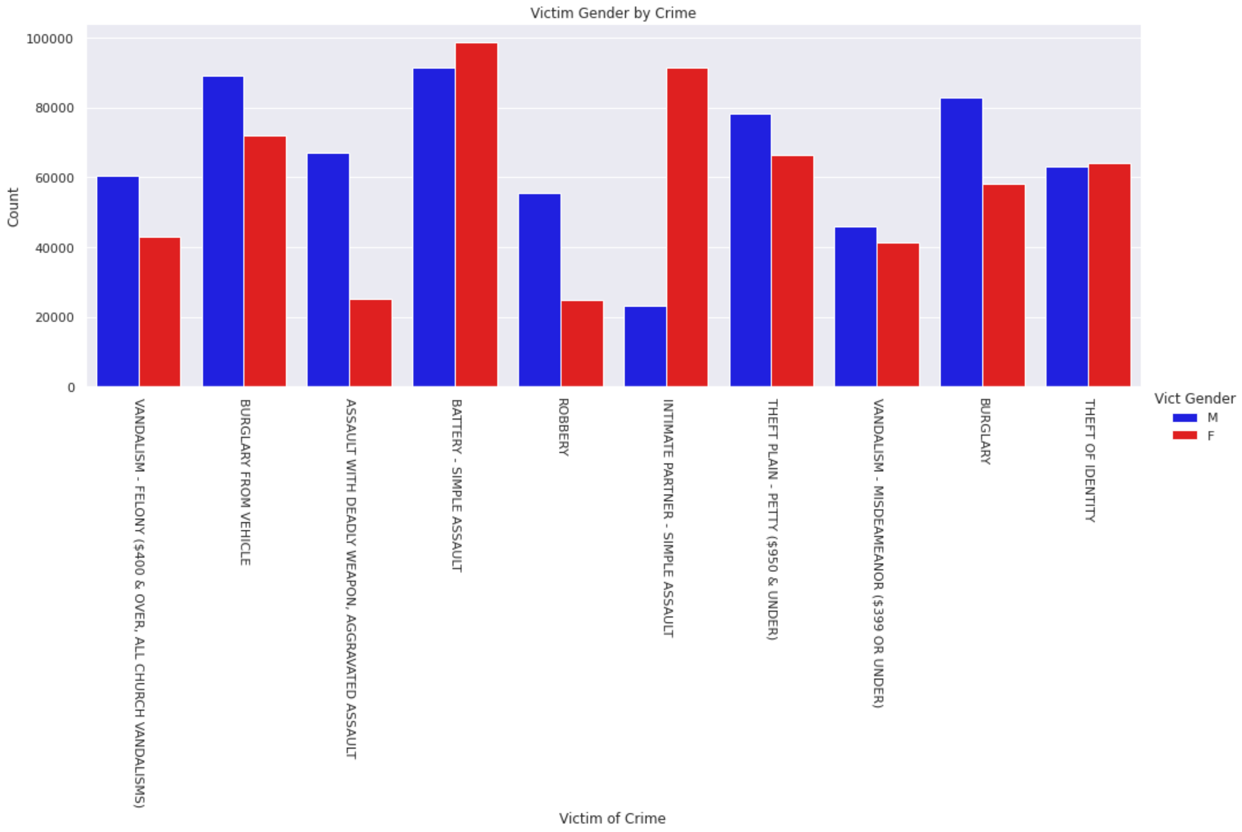

Crime Reported and Occured by Month
Variable
Crime Reported
Crime Occured
Crime
Month
0
20000
40000
60000
80000
100000
120000
140000
160000
January
February
March
April
May
June
July
August
September
October
November
December
159500
159500
143800
142700
161700
164200
158000
158000
165200
165000
161700
161400
167200
166500
167900
166700
160500
160700
165600
165200
152700
152800
156500
157400


Crime Reported and Occured by Year
Total Crimes
200000
150000
100000
50000
0
2010
2012
2014
2016
2018
2020
Year

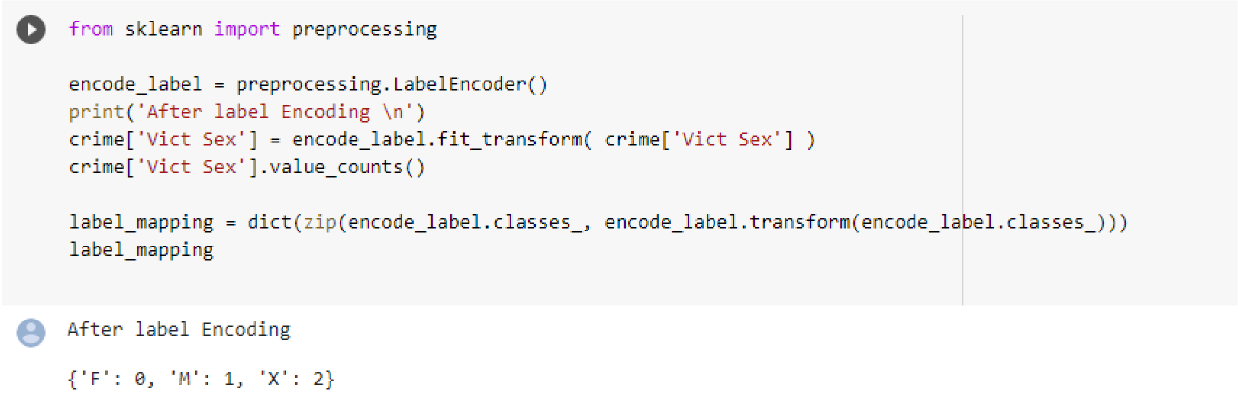

Crime Premises
SINGLE FAMILY DWELLING
STREET
MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)
PARKING LOT
SIDEWALK
OTHER BUSINESS
VEHICLE, PASSENGER/TRUCK
DRIVEWAY
GARAGE/CARPORT
RESTAURANT/FAST FOOD
DEPARTMENT STORE
MARKET
OTHER STORE
PARKING UNDERGROUND/BUILDING
PARK/PLAYGROUND
YARD (RESIDENTIAL/BUSINESS)
HIGH SCHOOL
OTHER PREMISE
ALLEY
DRUG STORE
Premise
0
100000
200000
300000
400000
Total Crimes Occurred


Weapon Used
MISSING WEAPON
STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)
VERBAL THREAT
UNKNOWN WEAPON/OTHER WEAPON
HAND GUN
SEMI-AUTOMATIC PISTOL
KNIFE WITH BLADE 6INCHES OR LESS
OTHER KNIFE
UNKNOWN FIREARM
VEHICLE
REVOLVER
BOTTLE
ROCK/THROWN OBJECT
STICK
CLUB/BAT
FOLDING KNIFE
BLUNT INSTRUMENT
MACE/PEPPER SPRAY
KITCHEN KNIFE
SIMULATED GUN
Weapon
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e6
Total Crimes Occurred


Status
Invest Cont
Adult Other
Adult Arrest
Juv Arrest
Juv Other
UNK
Arrest Status
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e6
Total Crimes Occurred

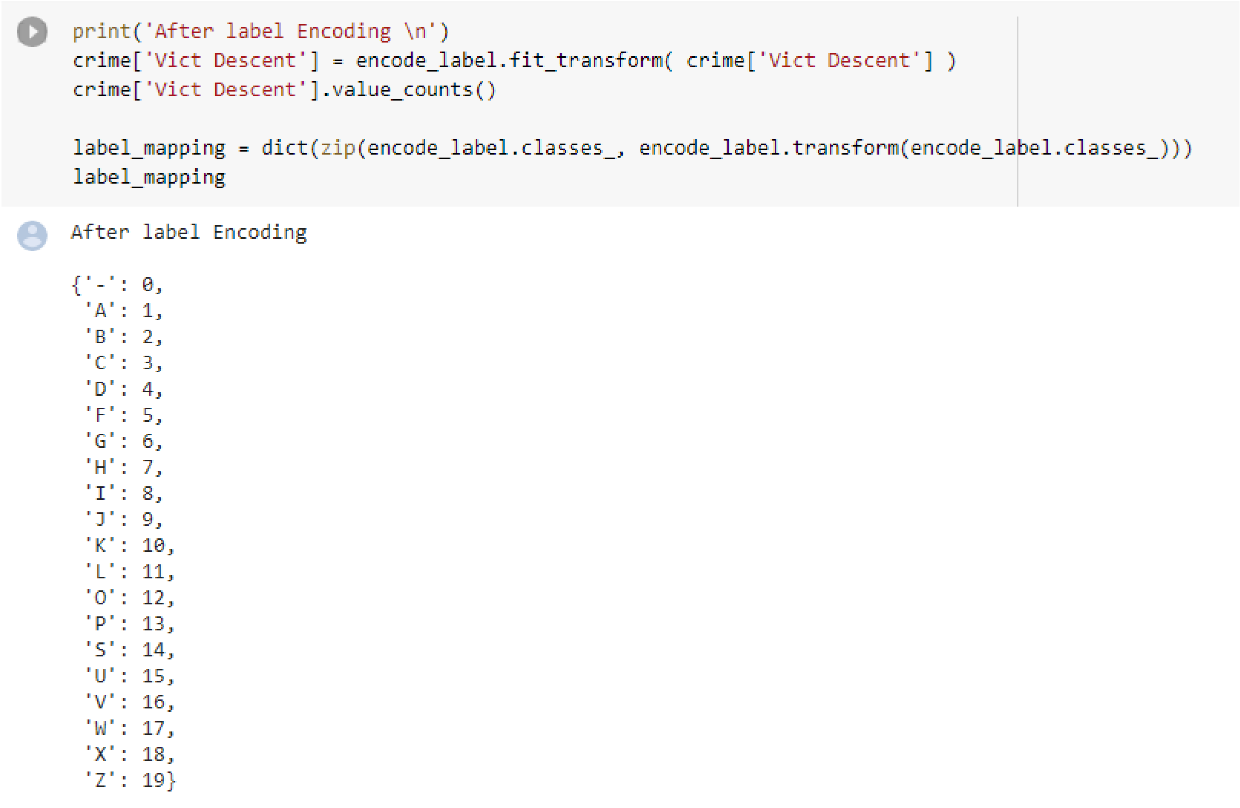

Victim Gender by Crime
Count
100000
80000
60000
40000
20000
0
VANDALISM - FELONY ($400 & OVER, ALL CHURCH VANDALISMS)
BURGLARY FROM VEHICLE
ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT
BATTERY - SIMPLE ASSAULT
ROBBERY
INTIMATE PARTNER - SIMPLE ASSAULT
THEFT PLAIN - PETTY ($950 & UNDER)
VANDALISM - MISDEAMEANOR ($399 OR UNDER)
BURGLARY
THEFT OF IDENTITY
Victim of Crime
Vict Gender
M
F

5. Data modeling



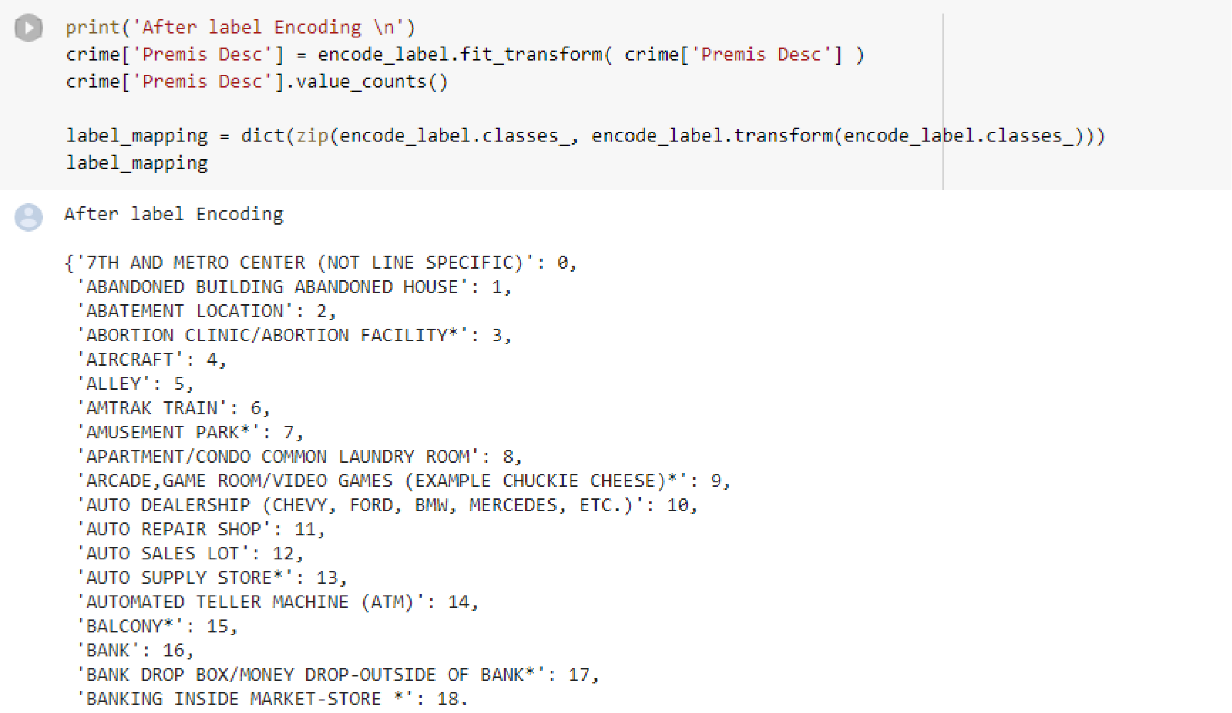
```
from sklearn import preprocessing

encode_label = preprocessing.LabelEncoder()
print('After label Encoding \n')
crime['Vict Sex'] = encode_label.fit_transform( crime['Vict Sex'] )
crime['Vict Sex'].value_counts()

label_mapping = dict(zip(encode_label.classes_, encode_label.transform(encode_label.classes_)))
label_mapping
```

```
After label Encoding

{'F': 0, 'M': 1, 'X': 2}
```



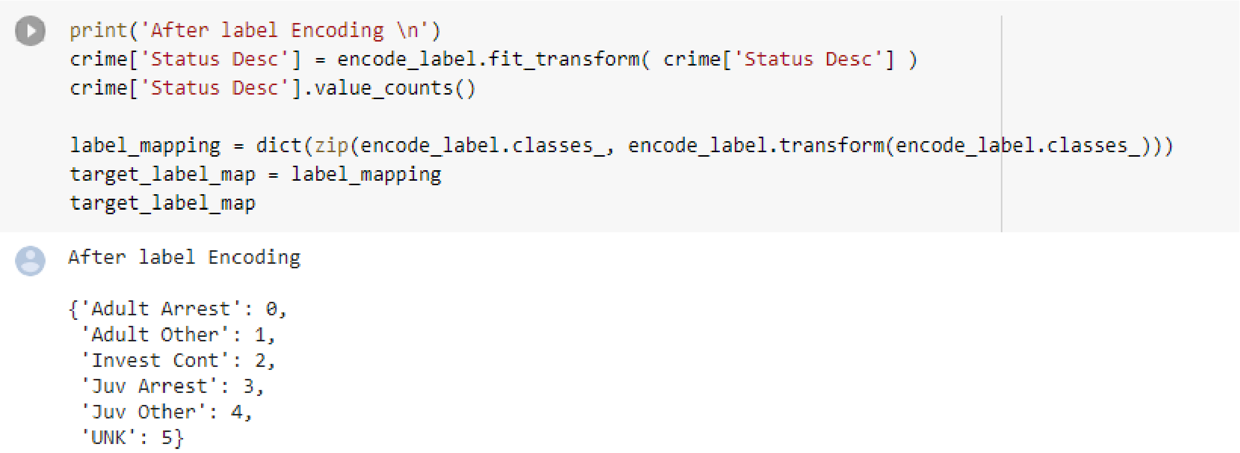
```
print('After label Encoding \n')
crime['Vict Descent'] = encode_label.fit_transform( crime['Vict Descent'] )
crime['Vict Descent'].value_counts()

label_mapping = dict(zip(encode_label.classes_, encode_label.transform(encode_label.classes_)))
label_mapping
```

```
After label Encoding

{'-': 0,
 'A': 1,
 'B': 2,
 'C': 3,
 'D': 4,
 'F': 5,
 'G': 6,
 'H': 7,
 'I': 8,
 'J': 9,
 'K': 10,
 'L': 11,
 'O': 12,
 'P': 13,
 'S': 14,
 'U': 15,
 'V': 16,
 'W': 17,
 'X': 18,
 'Z': 19}
```



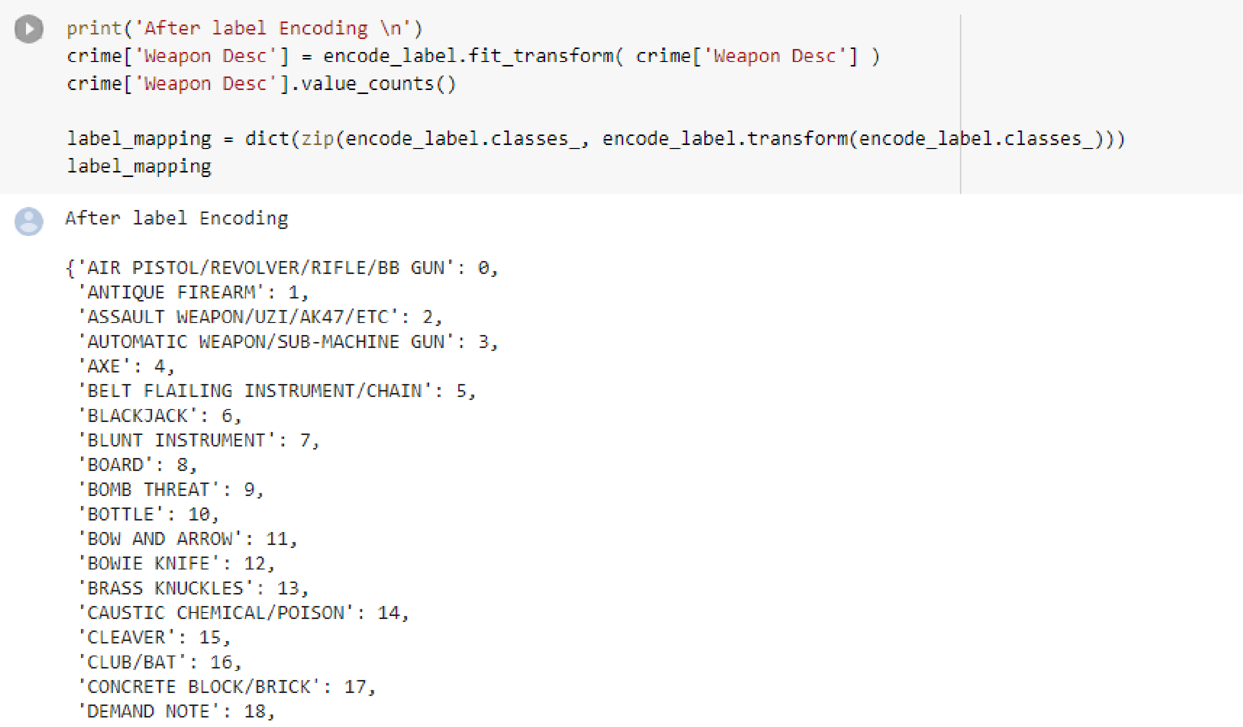
```
print('After label Encoding \n')
crime['Premis Desc'] = encode_label.fit_transform( crime['Premis Desc'] )
crime['Premis Desc'].value_counts()

label_mapping = dict(zip(encode_label.classes_, encode_label.transform(encode_label.classes_)))
label_mapping
```

```
After label Encoding

{'7TH AND METRO CENTER (NOT LINE SPECIFIC)': 0,
 'ABANDONED BUILDING ABANDONED HOUSE': 1,
 'ABATEMENT LOCATION': 2,
 'ABORTION CLINIC/ABORTION FACILITY*': 3,
 'AIRCRAFT': 4,
 'ALLEY': 5,
 'AMTRAK TRAIN': 6,
 'AMUSEMENT PARK*': 7,
 'APARTMENT/CONDO COMMON LAUNDRY ROOM': 8,
 'ARCADE,GAME ROOM/VIDEO GAMES (EXAMPLE CHUCKIE CHEESE)*': 9,
 'AUTO DEALERSHIP (CHEVY, FORD, BMW, MERCEDES, ETC.)': 10,
 'AUTO REPAIR SHOP': 11,
 'AUTO SALES LOT': 12,
 'AUTO SUPPLY STORE*': 13,
 'AUTOMATED TELLER MACHINE (ATM)': 14,
 'BALCONY*': 15,
 'BANK': 16,
 'BANK DROP BOX/MONEY DROP-OUTSIDE OF BANK*': 17,
 'BANKING INSIDE MARKET-STORE *': 18.
```



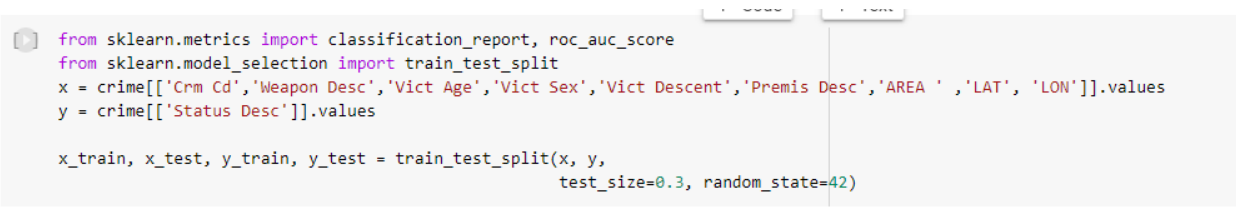
```
print('After label Encoding \n')
crime['Status Desc'] = encode_label.fit_transform( crime['Status Desc'] )
crime['Status Desc'].value_counts()

label_mapping = dict(zip(encode_label.classes_, encode_label.transform(encode_label.classes_)))
target_label_map = label_mapping
target_label_map
```

```
After label Encoding

{'Adult Arrest': 0,
 'Adult Other': 1,
 'Invest Cont': 2,
 'Juv Arrest': 3,
 'Juv Other': 4,
 'UNK': 5}
```

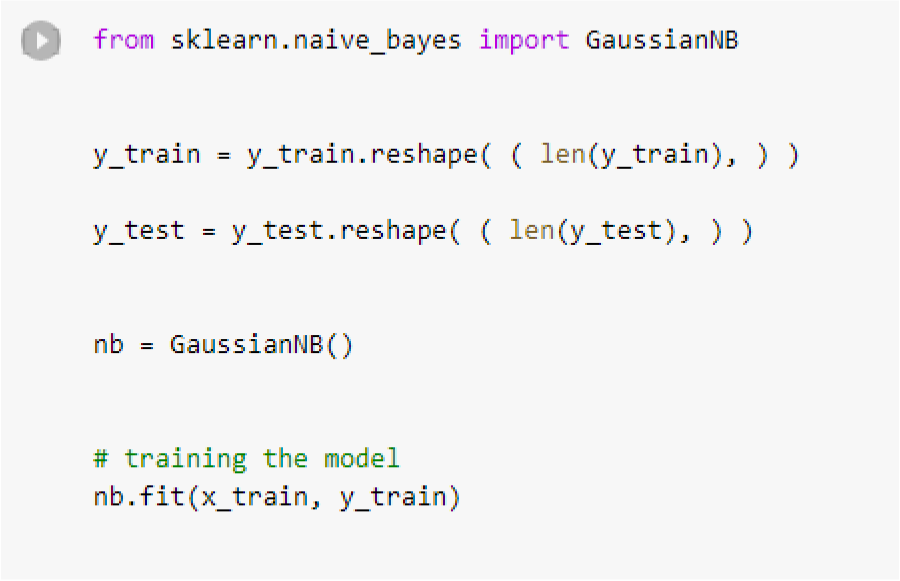


```
print('After label Encoding \n')
crime['Weapon Desc'] = encode_label.fit_transform( crime['Weapon Desc'] )
crime['Weapon Desc'].value_counts()

label_mapping = dict(zip(encode_label.classes_, encode_label.transform(encode_label.classes_)))
label_mapping
```

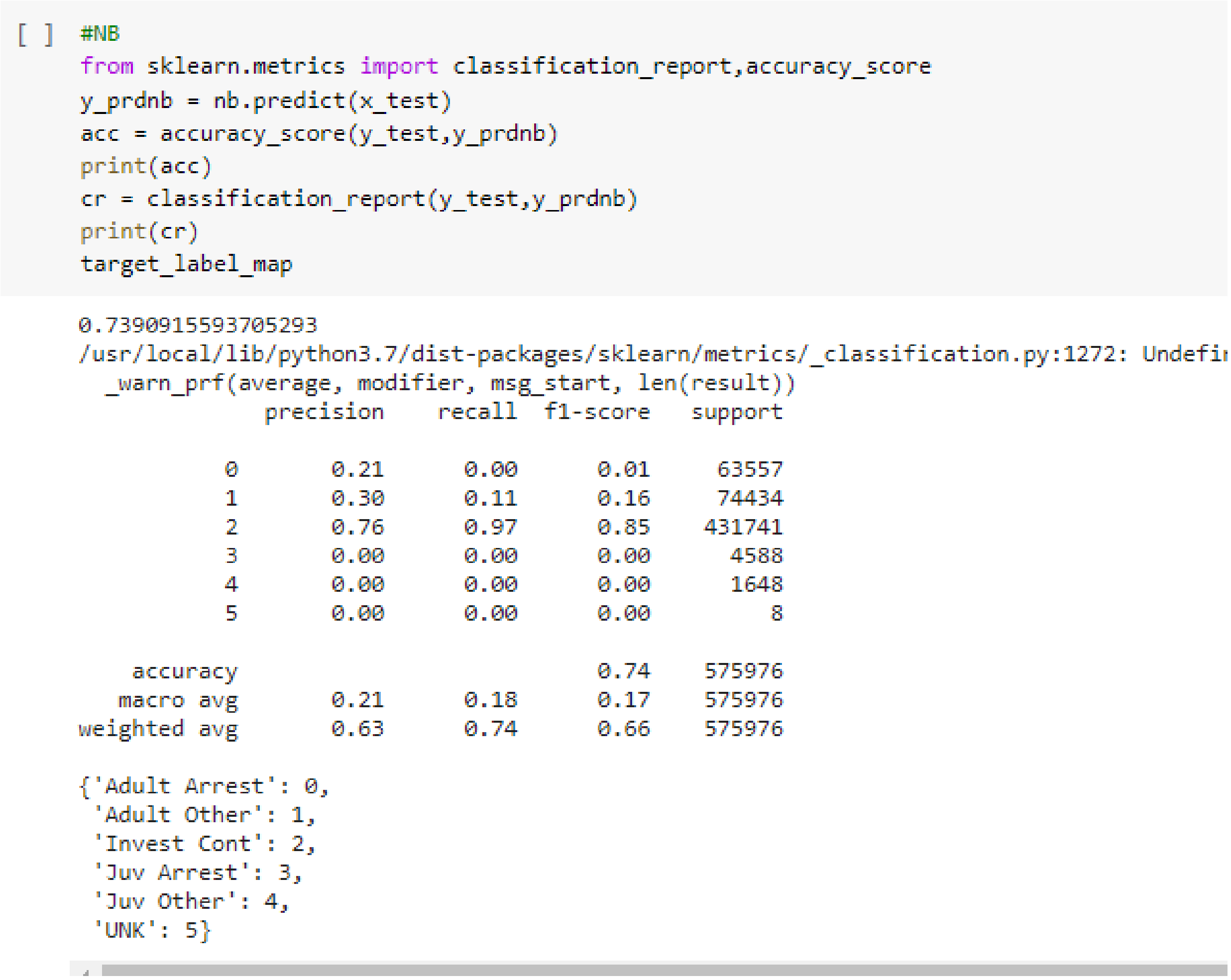
```
After label Encoding

{'AIR PISTOL/REVOLVER/RIFLE/BB GUN': 0,
 'ANTIQUE FIREARM': 1,
 'ASSAULT WEAPON/UZI/AK47/ETC': 2,
 'AUTOMATIC WEAPON/SUB-MACHINE GUN': 3,
 'AXE': 4,
 'BELT FLAILING INSTRUMENT/CHAIN': 5,
 'BLACKJACK': 6,
 'BLUNT INSTRUMENT': 7,
 'BOARD': 8,
 'BOMB THREAT': 9,
 'BOTTLE': 10,
 'BOW AND ARROW': 11,
 'BOWIE KNIFE': 12,
 'BRASS KNUCKLES': 13,
 'CAUSTIC CHEMICAL/POISON': 14,
 'CLEAVER': 15,
 'CLUB/BAT': 16,
 'CONCRETE BLOCK/BRICK': 17,
 'DEMAND NOTE': 18,
```



```
from sklearn.metrics import classification_report, roc_auc_score
from sklearn.model_selection import train_test_split
x = crime[['Crm Cd','Weapon Desc','Vict Age','Vict Sex','Vict Descent','Premis Desc','AREA ' ,'LAT', 'LON']].values
y = crime[['Status Desc']].values

x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.3, random_state=42)
```

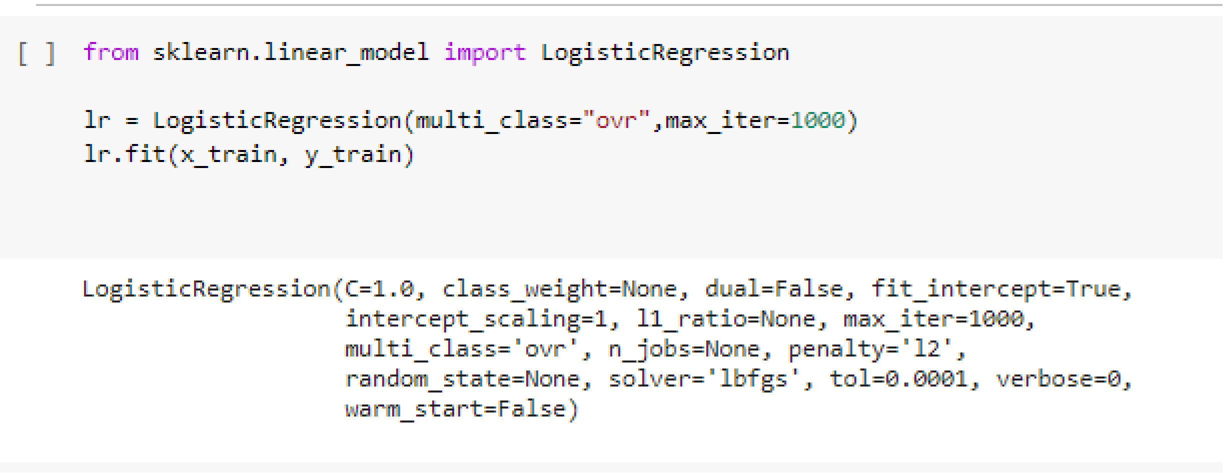


```
from sklearn.naive_bayes import GaussianNB


y_train = y_train.reshape( ( len(y_train), ) )

y_test = y_test.reshape( ( len(y_test), ) )


nb = GaussianNB()


# training the model
nb.fit(x_train, y_train)
```



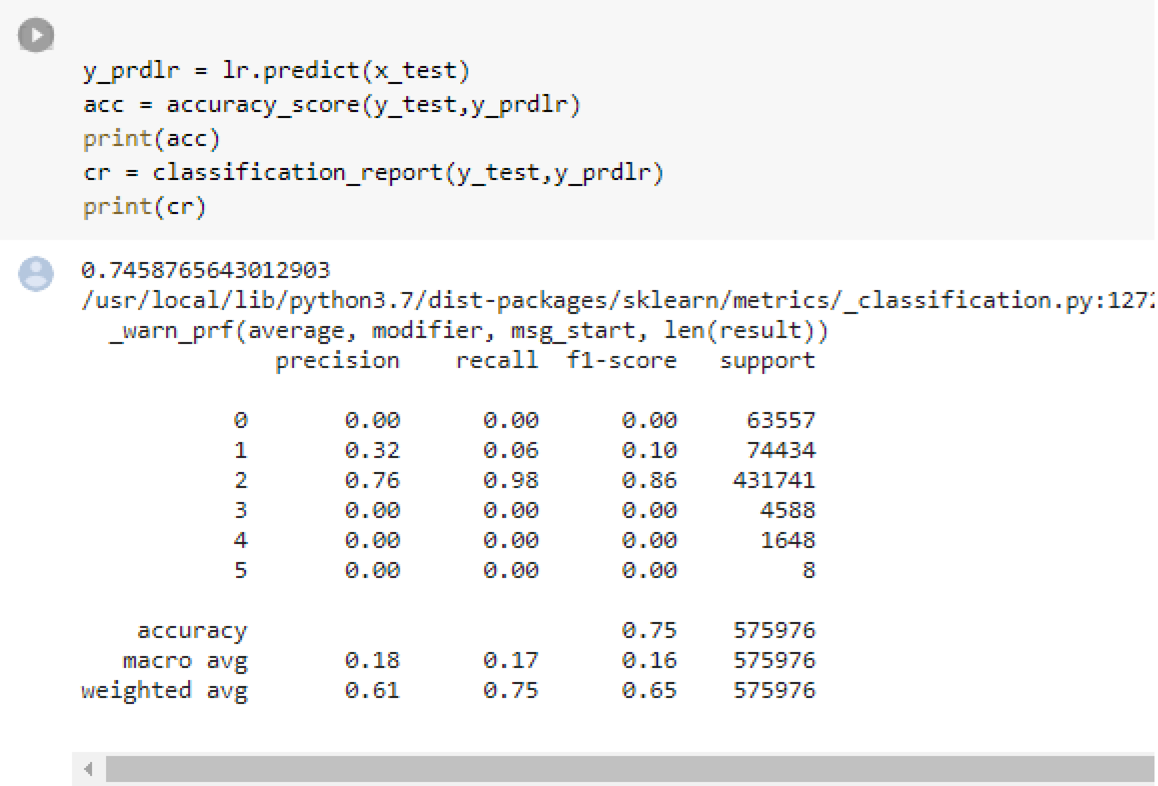
```
[ ] #NB
    from sklearn.metrics import classification_report,accuracy_score
    y_prdnb = nb.predict(x_test)
    acc = accuracy_score(y_test,y_prdnb)
    print(acc)
    cr = classification_report(y_test,y_prdnb)
    print(cr)
    target_label_map
```

```
0.7390915593705293
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272: Undefi
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.21      0.00      0.01     63557
           1       0.30      0.11      0.16     74434
           2       0.76      0.97      0.85    431741
           3       0.00      0.00      0.00      4588
           4       0.00      0.00      0.00      1648
           5       0.00      0.00      0.00         8

    accuracy                           0.74    575976
   macro avg       0.21      0.18      0.17    575976
weighted avg       0.63      0.74      0.66    575976

{'Adult Arrest': 0,
 'Adult Other': 1,
 'Invest Cont': 2,
 'Juv Arrest': 3,
 'Juv Other': 4,
 'UNK': 5}
```



```
[ ] from sklearn.linear_model import LogisticRegression

    lr = LogisticRegression(multi_class="ovr",max_iter=1000)
    lr.fit(x_train, y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=1000,
                   multi_class='ovr', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```



```
y_prdlr = lr.predict(x_test)
acc = accuracy_score(y_test,y_prdlr)
print(acc)
cr = classification_report(y_test,y_prdlr)
print(cr)
```

```
0.7458765643012903
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:127
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.00      0.00      0.00     63557
           1       0.32      0.06      0.10     74434
           2       0.76      0.98      0.86    431741
           3       0.00      0.00      0.00      4588
           4       0.00      0.00      0.00      1648
           5       0.00      0.00      0.00         8

    accuracy                           0.75    575976
   macro avg       0.18      0.17      0.16    575976
weighted avg       0.61      0.75      0.65    575976
```

--- x ---THE END --- x ---