System Analysis

IMPORT LIBRARIES:-

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        from wordcloud import WordCloud
        import seaborn as sns
        from sklearn.cluster import KMeans
        from bubbly.bubbly import bubbleplot
        from plotly.graph_objs import Scatter, Figure, Layout
```

Cleaning of Data:-

Data cleaning is the process of detecting and correcting (or removing) corrupt or inaccurate records from a Data set. Hence we find out the missing and 'NaN' values in the data set. There were many 'NaN' values specially in the last rows of the dataset.

```
dataset1=pd.read_csv('Chicago_Crimes_2001_to_2004.csv',error_bad_lines=False)
dataset2=pd.read_csv('Chicago_Crimes_2005_to_2007.csv',error_bad_lines=False)
dataset3=pd.read_csv('Chicago_Crimes_2008_to_2011.csv',error_bad_lines=False)
dataset4=pd.read_csv('Chicago_Crimes_2012_to_2017.csv',error_bad_lines=False)
```

dataset1	DataFrame	(1923515, 23)	Column names: Unnamed: 0, ID, Case Number, Date, Block, IUCR, Primary ...
dataset2	DataFrame	(1872343, 23)	Column names: Unnamed: 0, ID, Case Number, Date, Block, IUCR, Primary ...
dataset3	DataFrame	(2688710, 23)	Column names: Unnamed: 0, ID, Case Number, Date, Block, IUCR, Primary ...
dataset4	DataFrame	(1456714, 23)	Column names: Unnamed: 0, ID, Case Number, Date, Block, IUCR, Primary ...

```
#data preparation
#1 data cleaning: removal of null values

dataset1 = dataset1.dropna(how='any',axis=0)
dataset2 = dataset2.dropna(how='any',axis=0)
dataset3 = dataset3.dropna(how='any',axis=0)
dataset4 = dataset4.dropna(how='any',axis=0)
```

dataset1	DataFrame	(1205641, 23)	Column names: Unnamed: 0, ID, Case Number, Date, Block, IUCR, Primary ...
dataset2	DataFrame	(1862832, 23)	Column names: Unnamed: 0, ID, Case Number, Date, Block, IUCR, Primary ...
dataset3	DataFrame	(2658375, 23)	Column names: Unnamed: 0, ID, Case Number, Date, Block, IUCR, Primary ...
dataset4	DataFrame	(1418365, 23)	Column names: Unnamed: 0, ID, Case Number, Date, Block, IUCR, Primary ...

It can be observed that a huge amount of information is lost while removing rows with null value so in place of removal.

Determining the count of null and missing values in columns in all four datasets:

```
#data preprocessing
#count number of nans in column
print( dataset1.isnull().sum())
print( dataset2.isnull().sum())
print( dataset3.isnull().sum())
print( dataset4.isnull().sum())
```

1.

```
Unnamed: 0                  0
ID                          0
Case Number                 0
Date                        0
Block                       0
IUCR                        0
Primary Type                0
Description                 0
Location Description       16
Arrest                      0
Domestic                    0
Beat                        0
District                    2
Ward                   700132
Community Area         700247
FBI Code                    0
X Coordinate            30691
Y Coordinate            30691
Year                        0
Updated On                  0
Latitude                30691
Longitude               30692
Location                30692
dtype: int64
```

2.

```
Unnamed: 0                  0
ID                          0
Case Number                 0
Date                        0
Block                       0
IUCR                        0
Primary Type                0
Description                 0
Location Description       16
Arrest                      0
Domestic                    0
Beat                        0
District                    2
Ward                   700132
Community Area         700247
FBI Code                    0
X Coordinate            30691
Y Coordinate            30691
Year                        0
Updated On                  0
Latitude                30691
Longitude               30692
Location                30692
dtype: int64
```

3.

```
Unnamed: 0                  0
ID                          0
Case Number                 6
Date                        0
Block                       0
IUCR                        0
Primary Type                0
Description                 0
Location Description      291
Arrest                      0
Domestic                    0
Beat                        0
District                   83
Ward                       63
Community Area           1455
FBI Code                    0
X Coordinate            28645
Y Coordinate            28645
Year                        0
Updated On                  0
Latitude                28645
Longitude               28645
Location                28645
dtype: int64
```

4.

```
Unnamed: 0                0
ID                        0
Case Number               1
Date                      0
Block                     0
IUCR                      0
Primary Type              0
Description               0
Location Description   1658
Arrest                    0
Domestic                  0
Beat                      0
District                  1
Ward                     14
Community Area           40
FBI Code                  0
X Coordinate          37083
Y Coordinate          37083
Year                      0
Updated On                0
Latitude              37083
Longitude             37083
Location              37083
dtype: int64
```

Formatting of Data:-

Data formatting depends upon the purpose of your data, elements in the data and much more. Data formatting enhances the visual appearance of our worksheet. Hence we find out that Time and Date column are in Object Data type which can create a problem afterwards. So, I converted TIME Column (HH:MM:SS) format into a new column i.e. 'HOUR' column and DATE Column (YYYY-MM-DD) format into a new column i.e. 'MONTH' column.

```
#formatting data
dataset1.Date = pd.to_datetime(dataset1.Date,format='%m/%d/%Y %I:%M:%S %p')
dataset2.Date = pd.to_datetime(dataset2.Date,format='%m/%d/%Y %I:%M:%S %p')
dataset3.Date = pd.to_datetime(dataset3.Date,format='%m/%d/%Y %I:%M:%S %p')
dataset4.Date = pd.to_datetime(dataset4.Date,format='%m/%d/%Y %I:%M:%S %p')
```

```
'''setting index to date'''
dataset1.index=pd.DatetimeIndex(dataset1.Date)
dataset2.index=pd.DatetimeIndex(dataset2.Date)
dataset3.index=pd.DatetimeIndex(dataset3.Date)
dataset4.index=pd.DatetimeIndex(dataset4.Date)
```

```
'''adding time hour hour grouping'''
dataset1['time_hour']=dataset1['Date'].apply(lambda x: x.hour)
dataset1['month']=dataset1['Date'].apply(lambda x: x.month)
dataset1['year']=dataset1['Date'].apply(lambda x: x.year)

dataset2['time_hour']=dataset2['Date'].apply(lambda x: x.hour)
dataset2['month']=dataset2['Date'].apply(lambda x: x.month)
dataset2['year']=dataset2['Date'].apply(lambda x: x.year)

dataset3['time_hour']=dataset3['Date'].apply(lambda x: x.hour)
dataset3['month']=dataset3['Date'].apply(lambda x: x.month)
dataset3['year']=dataset3['Date'].apply(lambda x: x.year)

dataset4['time_hour']=dataset4['Date'].apply(lambda x: x.hour)
dataset4['month']=dataset4['Date'].apply(lambda x: x.month)
dataset4['year']=dataset4['Date'].apply(lambda x: x.year)
```

```
df=dataset4[dataset4['Year']==2017]
```

454320	6233785	10818314	JA118825	2017-01-15 08:00:00	019XX N BISSELL ST
454321	6233786	10818383	JA118875	2017-01-15 22:00:00	049XX S CAMPBELL AVE
454322	6233787	10818469	JA118907	2017-01-15 10:25:00	068XX S ASHLAND AVE
454323	6233788	10818470	JA117892	2017-01-15 18:00:00	029XX W 40TH ST
454324	6233789	10818529	JA119012	2017-01-15 15:00:00	030XX N BROADWAY
454325	6233790	10818581	JA118997	2017-01-15 03:00:00	081XX S INGLESIDE AVE
454326	6233791	10818607	JA119037	2017-01-15 12:00:00	052XX N LONG AVE
454327	6233792	10818629	JA118872	2017-01-15 12:00:00	012XX N KEELER AVE
454328	6233793	10818650	JA118998	2017-01-15 11:00:00	064XX S MORGAN ST
454329	6233794	10818652	JA118802	2017-01-15 21:00:00	044XX N KASSON AVE
454330	6233795	10818675	JA119042	2017-01-15 20:00:00	035XX N RETA AVE
454331	6233796	10818680	JA118028	2017-01-15 18:00:00	009XX W LAKE ST
454332	6233797	10818710	JA118787	2017-01-15 02:59:00	037XX W NORTH AVE
454333	6233798	10818813	JA119274	2017-01-15 09:00:00	078XX W ADDISON ST
454334	6233799	10818822	JA119102	2017-01-15 21:53:00	022XX W TOUHY AVE
454335	6233800	10818841	JA119190	2017-01-15 01:00:00	061XX S WHIPPLE ST
454336	6233801	10818848	JA117378	2017-01-15 20:50:00	012XX W 47TH ST
454337	6233802	10818858	JA118902	2017-01-15 01:00:00	032XX W 55TH ST
454338	6233803	10818859	JA119280	2017-01-15 10:30:00	070XX S WASHTENAW AVE
454339	6233804	10818897	JA119261	2017-01-15 05:20:00	032XX N WESTERN AVE
454340	6233805	10818925	JA119245	2017-01-15 21:00:00	044XX S WHIPPLE ST

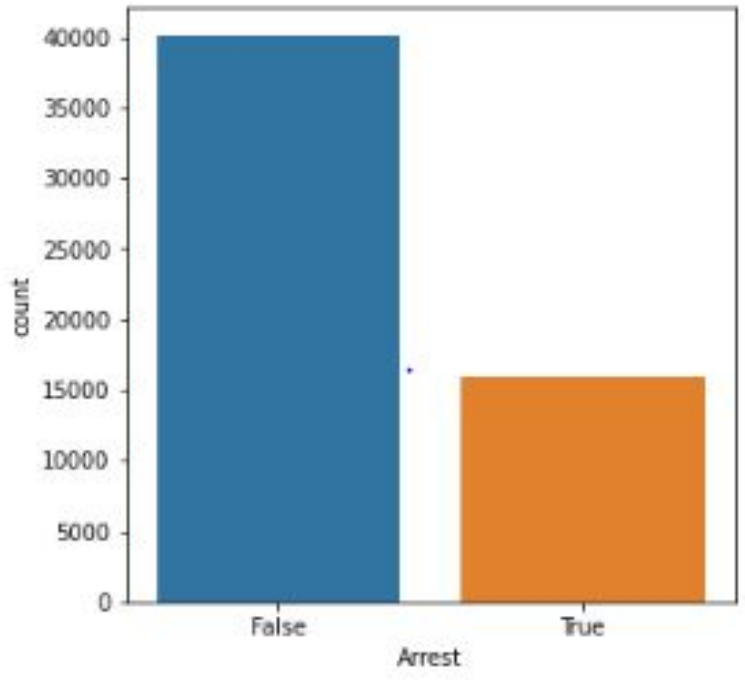
Data exploration:

To have a good understanding of data, several plots like arrest vs non arrest, word cloud for frequent crimes, frequency plot of crime by types were plotted so that we can have some insights from the data.

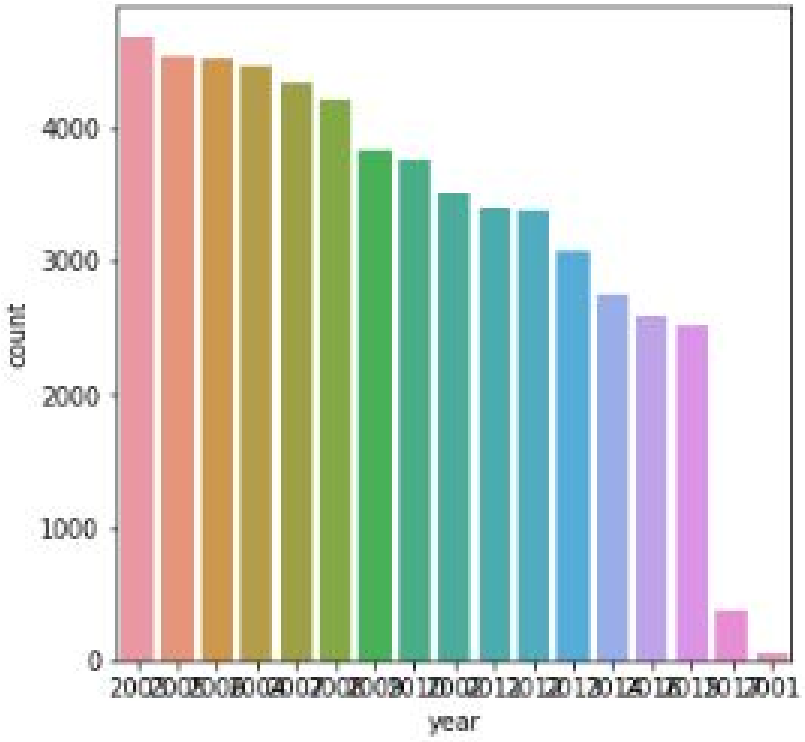
Frequent Crime:

Frequent crime location:

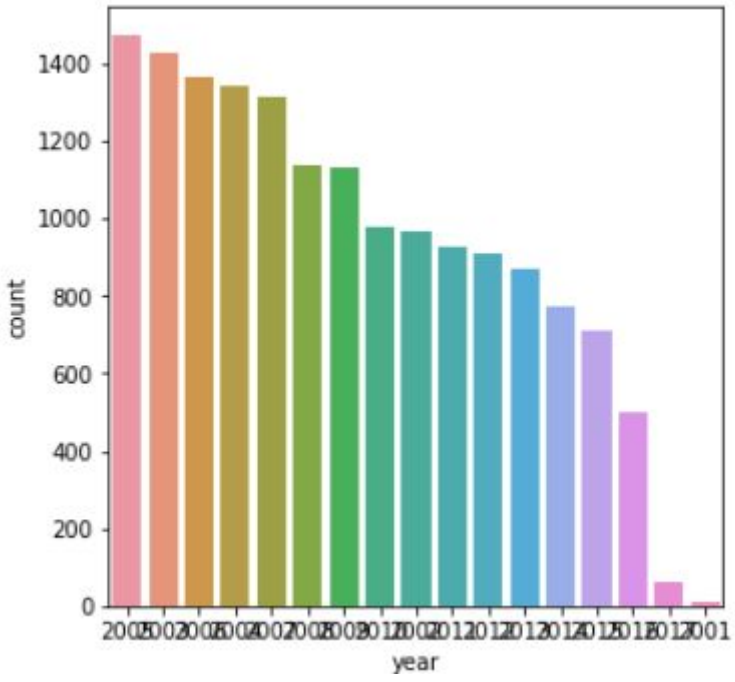
Total arrest:



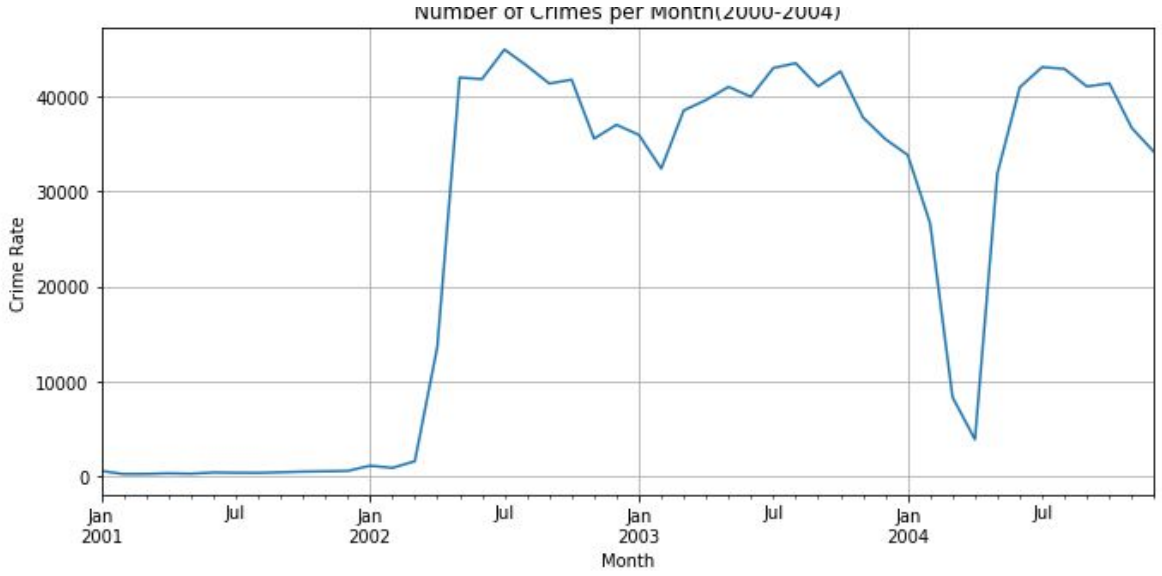
Arrest over period:



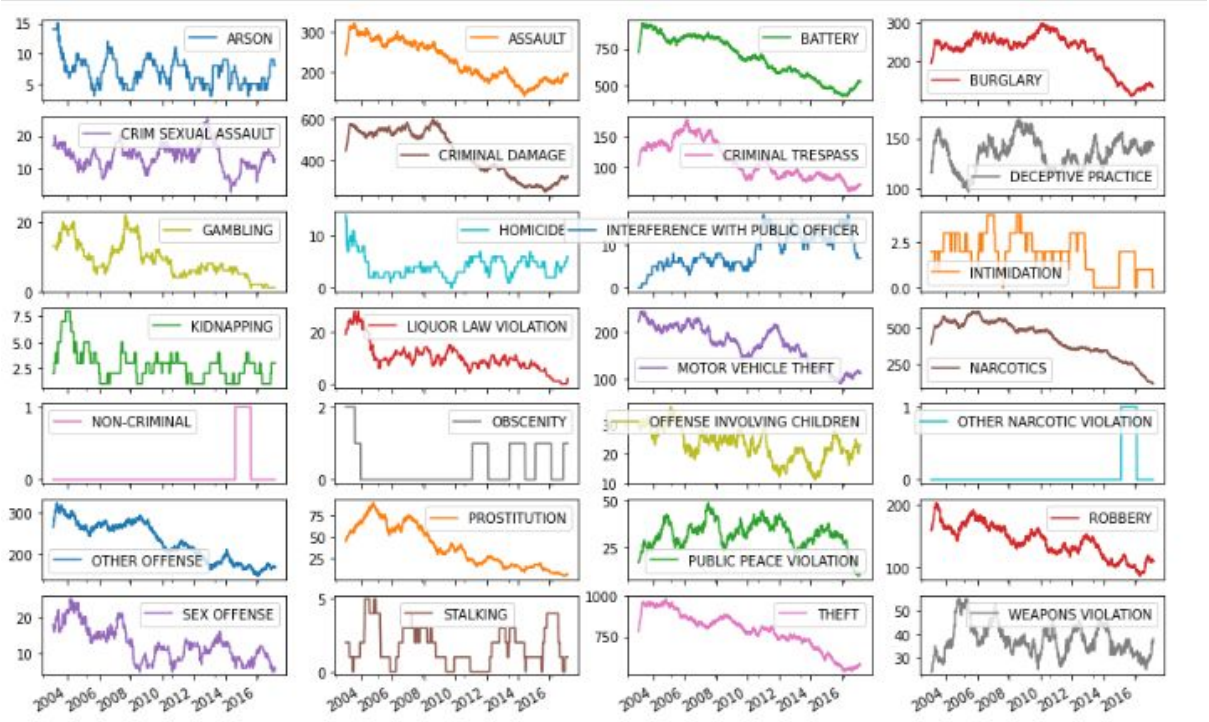
Arrest over period:



Monthly analysis:

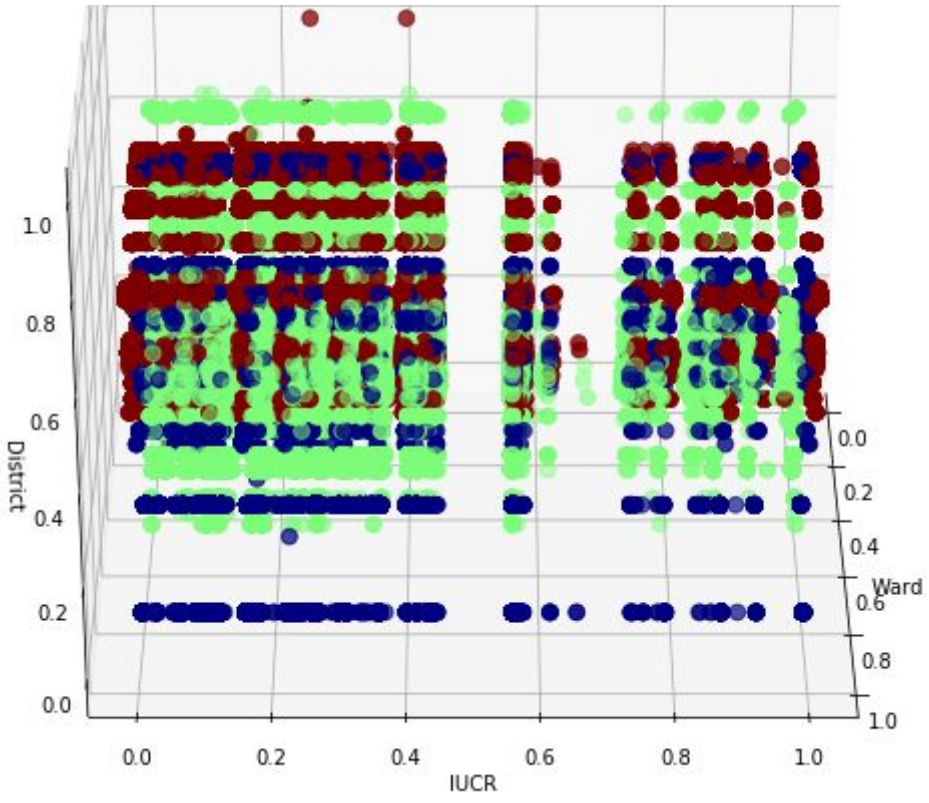


Individual crime type trend:



Data Modeling:

The nature of crimes varies in the dataset therefore we have used a clustering model. K-means is the most suitable algorithm for this analysis since it is easy to build clusters using k-means. We have made clusters with the help of data containing location, type and also with time and IUCR.



Analysing Model:

The clusters obtained after applying k-means help us to draw a number of outcomes. Also the frequency plots shows the percentage of arrest, primary crime type etc