AMRITA SCHOOL OF ENGINEERING
BANGALORE

DEPARTMENT OF CSE

OPEN LAB -15CSE387

USING DATA SCIENCE TO EXPLORE, ANALYZE AND VISUALIZE THE CHICAGO CRIME DATASET

Ramshankar - BL.EN.U4CSE16106
Srikanth - BL.EN.U4CSE16126

Manishankar - BL.EN.U4CSE16111
Jaswanth - BL.EN.U4CSE16060

What the Project is About

Introduction

- Crime : "A Major Disruption"
- Project analyses data to find answers, communicates results
- Data often gives the best answers

Dataset

- Chicago Crime Dataset for the year 2018
- .csv format
- 65 MB in size
- Original Shape : (265698, 22)

Python Libraries Used

Libraries Used

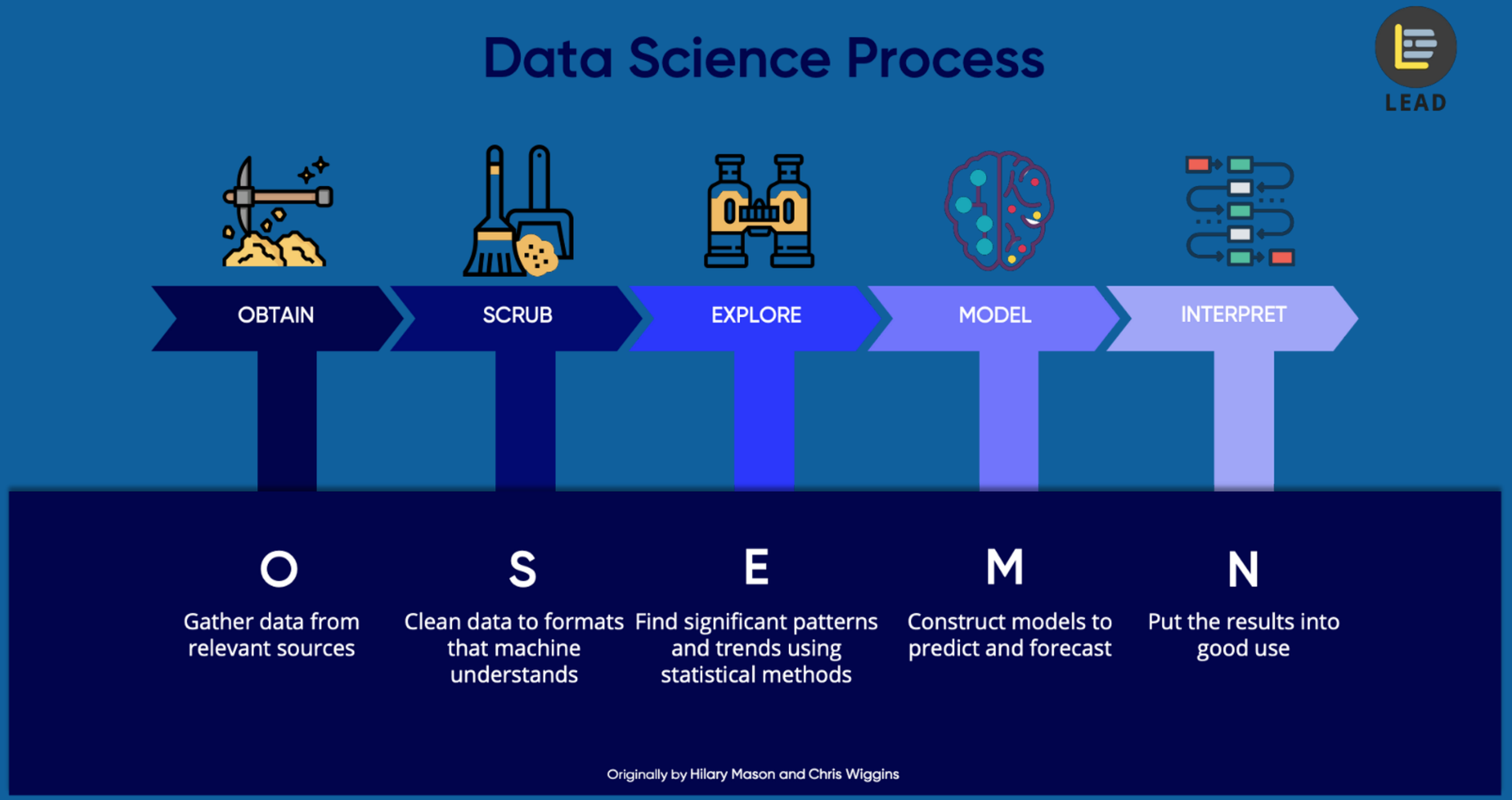
- Data Wrangling Library
pandas

- Data Visualization Libraries
matplotlib
seaborn
bokeh (Interactive Visualizations)

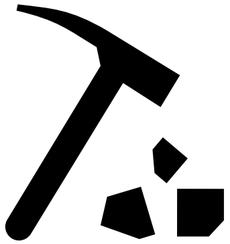
- ML/DL Libraries
sklearn
keras

- Miscellaneous
numpy (For math operations)
datetime (Python datetime module)

THE OSEMN FRAMEWORK



The Project Workflow (With Regards to the OSEMN Framework)

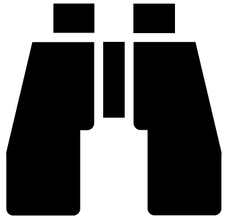


Obtain

[2 Hours]

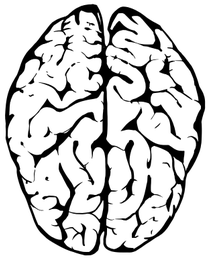
Scrub

[5 Hours]



Explore

[15 Hours]



Model

[8 Hours]

Interpret

[2 Hours]

DATA ACQUISITION

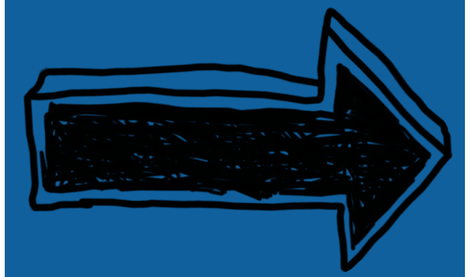
```
1  # First 5 rows of our dataset
2  df.head()
```

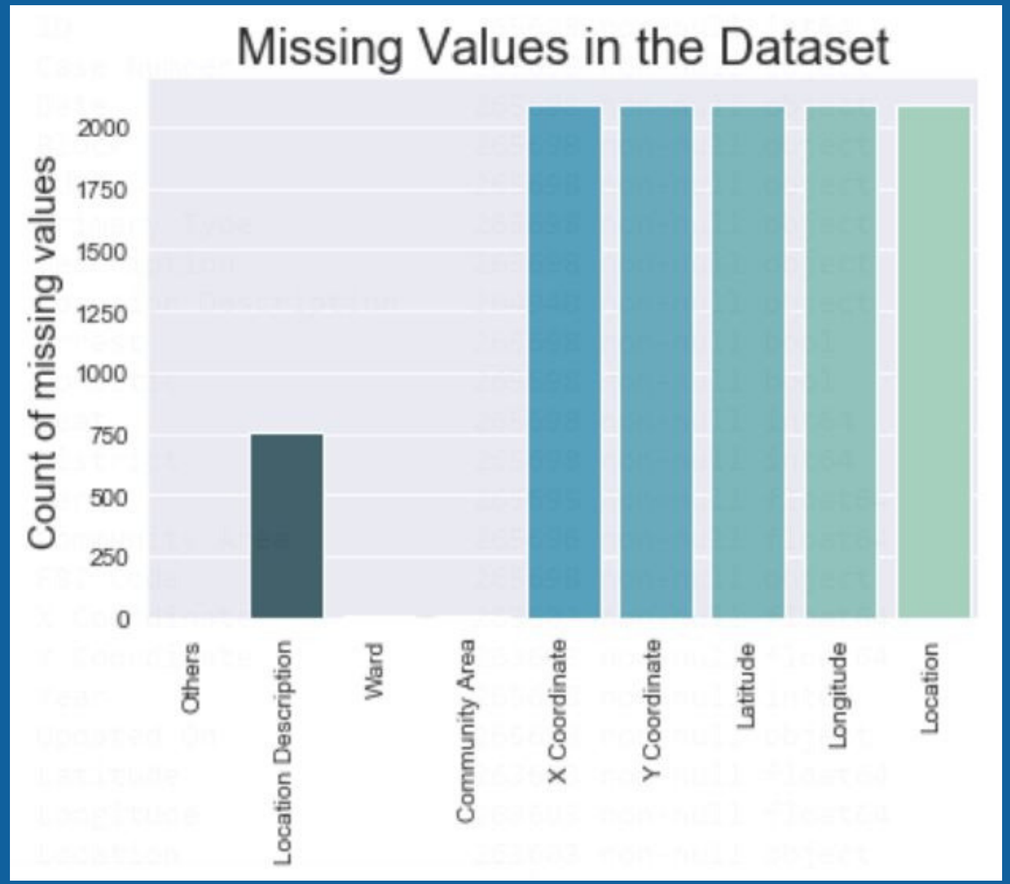
	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	Ward	Community Area	FBI Code	X Coordinate
0	11268908	JB202456	03/28/2018 07:50:00 AM	004XX W DIVISION ST	0860	THEFT	RETAIL THEFT	GROCERY FOOD STORE	True	False	...	27.0	8.0	06	1173175.0
1	11210587	JB124894	01/22/2018 12:10:00 AM	0000X E CHESTNUT ST	0281	CRIM SEXUAL ASSAULT	NON-AGGRAVATED	HOTEL/MOTEL	False	False	...	2.0	8.0	02	1176408.0
2	11207682	JB120881	01/18/2018 04:50:00 PM	006XX N LECLAIRE AVE	041A	BATTERY	AGGRAVATED: HANDGUN	STREET	False	False	...	37.0	25.0	04B	1142252.0
3	11599687	JC157279	12/24/2018 09:00:00 AM	011XX W 15TH ST	1130	DECEPTIVE PRACTICE	FRAUD OR CONFIDENCE GAME	RESIDENCE	False	False	...	11.0	28.0	11	NaN
4	11599643	JC157354	12/18/2018 10:00:00 AM	007XX E 89TH PL	2825	OTHER OFFENSE	HARASSMENT BY TELEPHONE	OTHER	False	True	...	8.0	44.0	26	NaN

First 5 rows of the dataset

DATA ACQUISITION

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 265698 entries, 0 to 265697
Data columns (total 22 columns):
ID                      265698 non-null int64
Case Number             265698 non-null object
Date                    265698 non-null object
Block                   265698 non-null object
IUCR                    265698 non-null object
Primary Type            265698 non-null object
Description             265698 non-null object
Location Description    264940 non-null object
Arrest                  265698 non-null bool
Domestic                265698 non-null bool
Beat                    265698 non-null int64
District                265698 non-null int64
Ward                    265695 non-null float64
Community Area          265696 non-null float64
FBI Code                265698 non-null object
X Coordinate            263603 non-null float64
Y Coordinate            263603 non-null float64
Year                    265698 non-null int64
Updated On              265698 non-null object
Latitude                263603 non-null float64
Longitude               263603 non-null float64
Location                263603 non-null object
dtypes: bool(2), float64(6), int64(4), object(10)
```





All 22 columns

Missing Values

DEALING WITH MISSING VALUES

```
# The simplest cleaning technique here would be to drop all the rows with atleast one missing value
df = df.dropna()
df.info()
```

We dropped the rows with atleast one missing value

```
# How much of the data has been retained after this removal ?
print(round(262960 / 265698 * 100,2), "percentage of the data has been retained.")
```

```
98.97 percentage of the data has been retained.
```

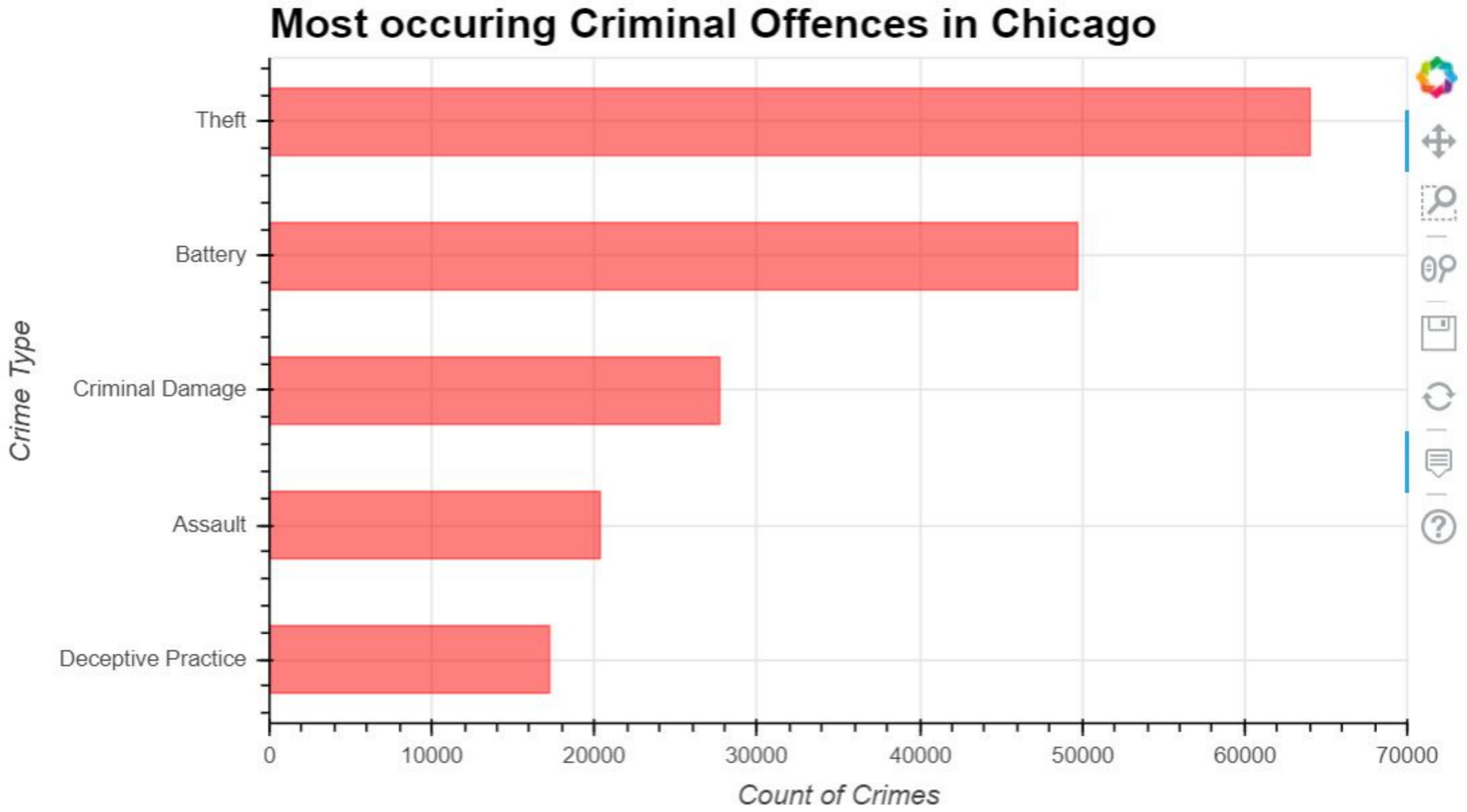
Dropping the rows will usually result in **clean datasets and produce well-behaved** data. But often, it removes a lot of information that reduces result accuracy. However, in our case, since **98.97% of the data** is retained and since there is practically no other way to work around the type of missing values we have, we shall go ahead with this slightly diminished dataset

It turned out to be the most efficient way in our case

Q and A with Chicago Crime Data 2018

ASKING QUESTIONS TO THE DATA AND EXTRACTING ANSWERS TO AID IN BETTER DECISION MAKING

Which are the top 5 crimes that occurred in Chicago City in 2018

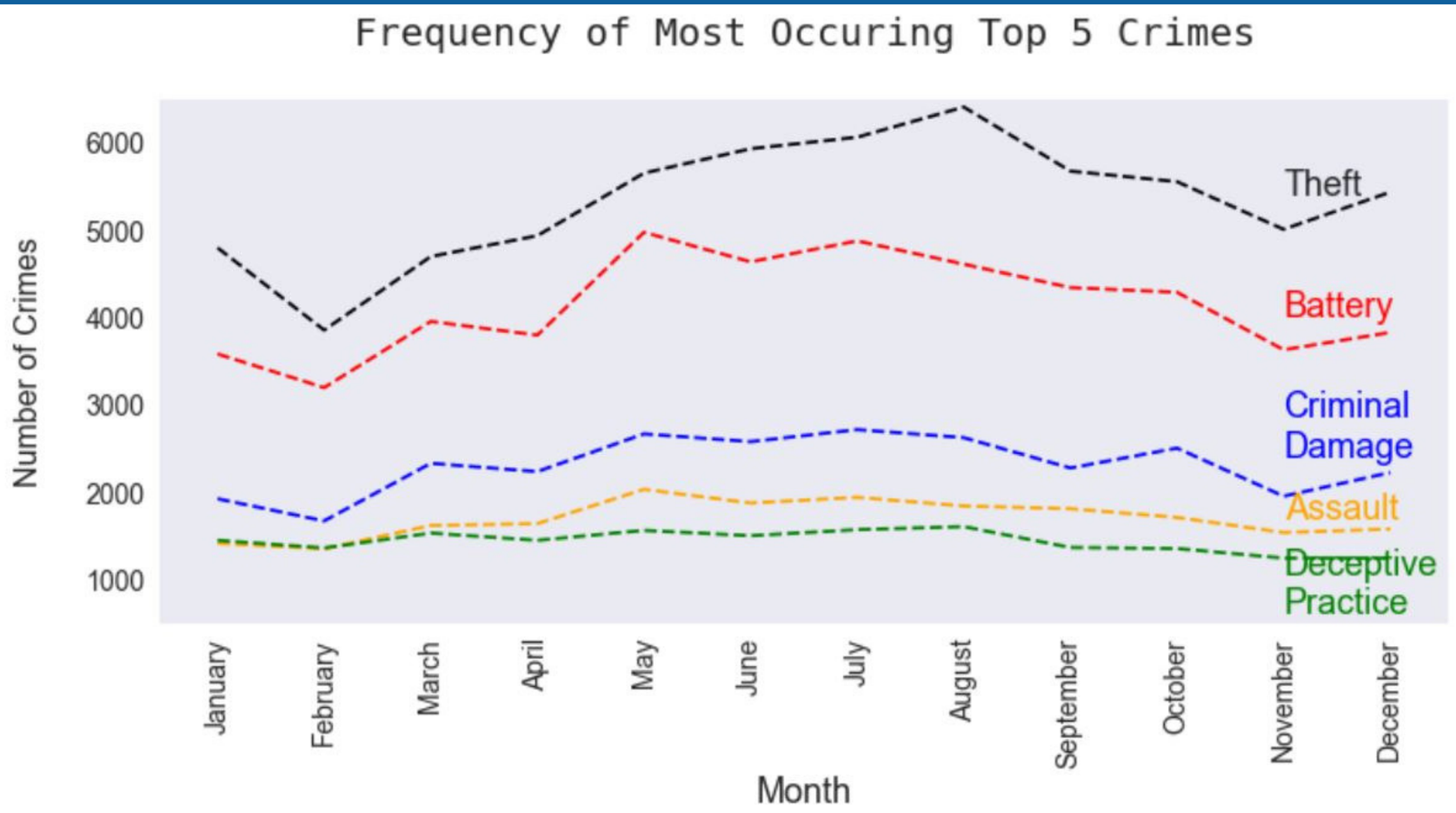


- Theft was the most occurring crime with a count of 64,302

- High counts of crimes like Battery and Assault, indicate the presence of a physically violent community

- Interactive plot with Bokeh

The Frequency of the top 5 crimes over the year



- Theft is on a phenomenal rise during the Summer

- All offenses are at their lowest at February (except Deceptive Prac.)

- Assault and Dec. Prac. have a smoother curve than the other 3

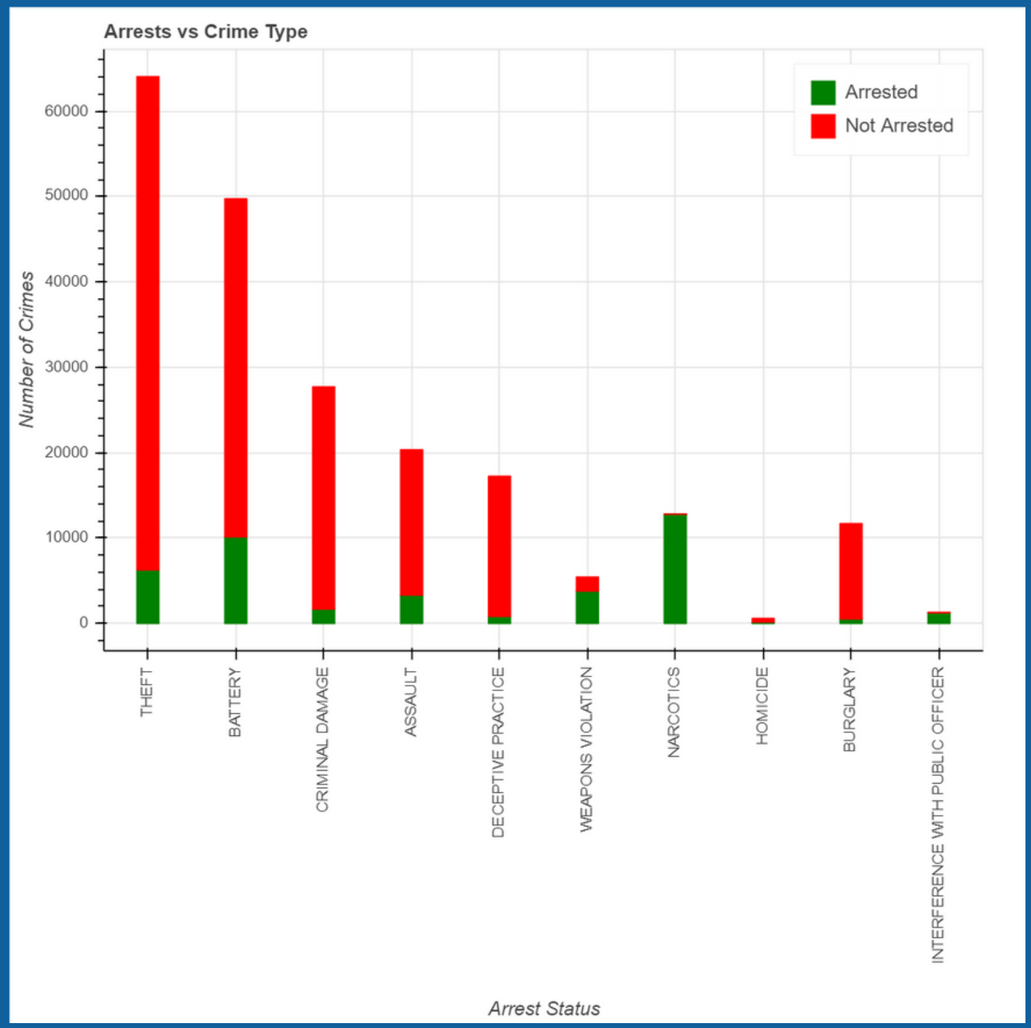
Arrests in the City of Chicago

```
# df['Arrest'].head()
l = df["Arrest"].value_counts()
false = l[0]
true = l[1]

arrest = pd.DataFrame({'Status':['Not Arrested','Arrested'],'Value':list(l)})
print("Percentage of arrests of all reported crimes :",false/(false+true)*100,'!')
```

```
Percentage of arrests of all reported crimes : 80.16352296927289 !
```

That's a stoking **80%** chance for evading an arrest!

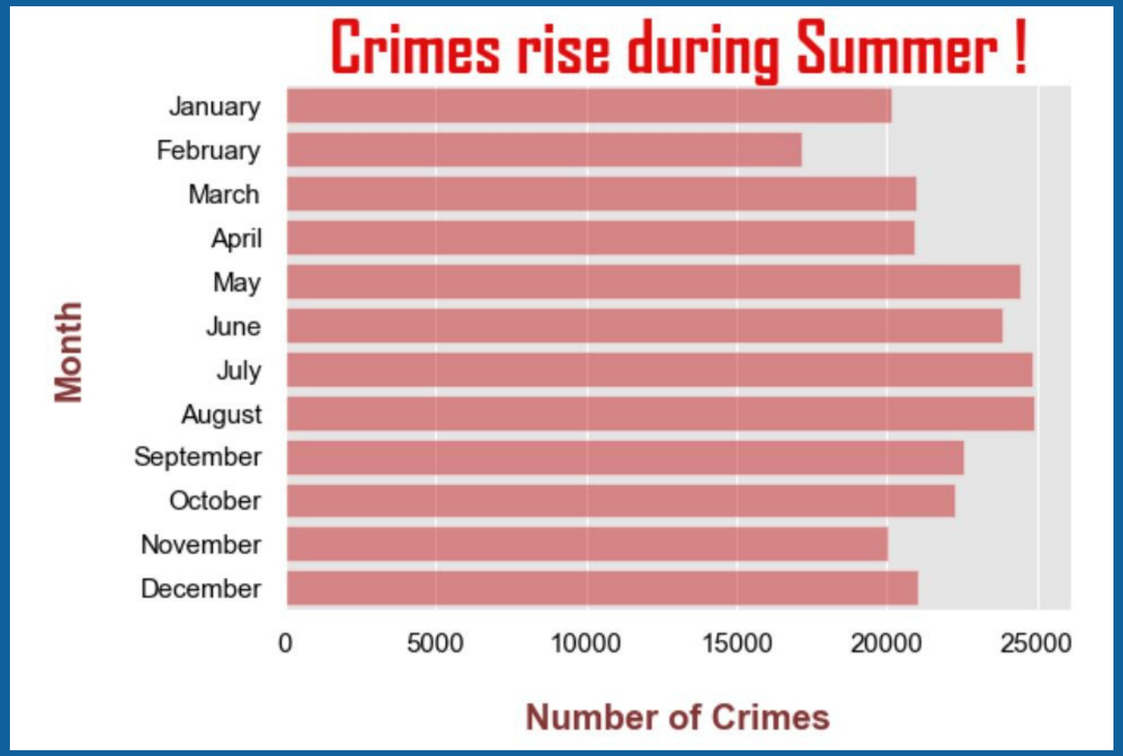


- 80% of the crimes saw no arrests !

- Most crimes have seen a lot of "No Arrests" than "Arrests"

- However, it is good to see that "Narcotics" has a 99% arrest rate ! Even "Weapons Violation" has a good arrest rate.

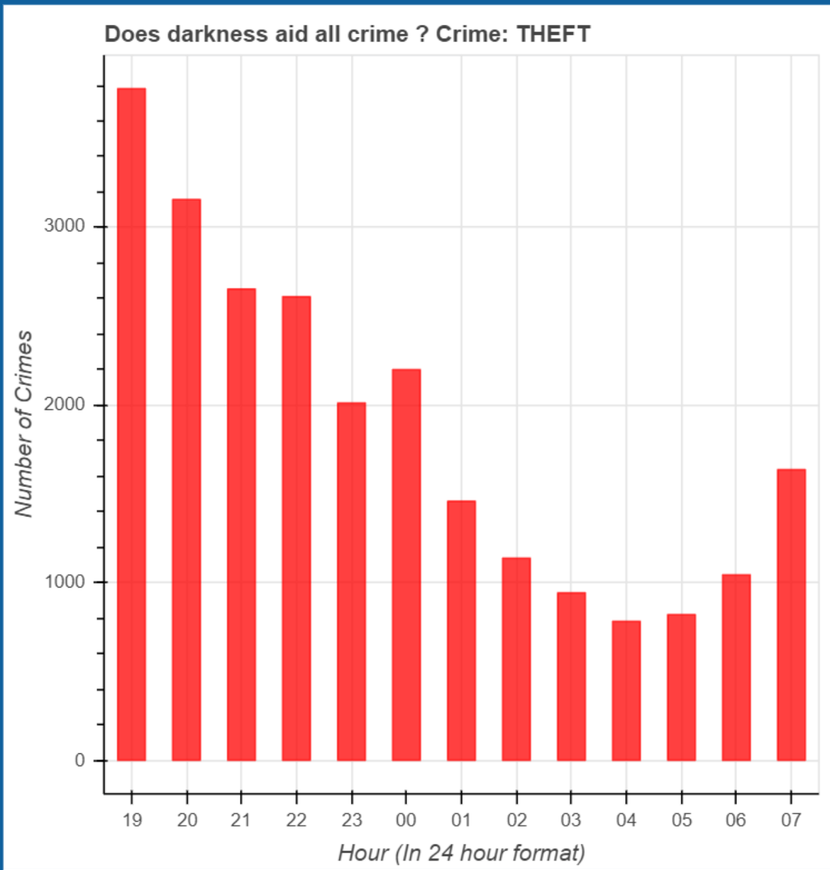
Crime vs Time

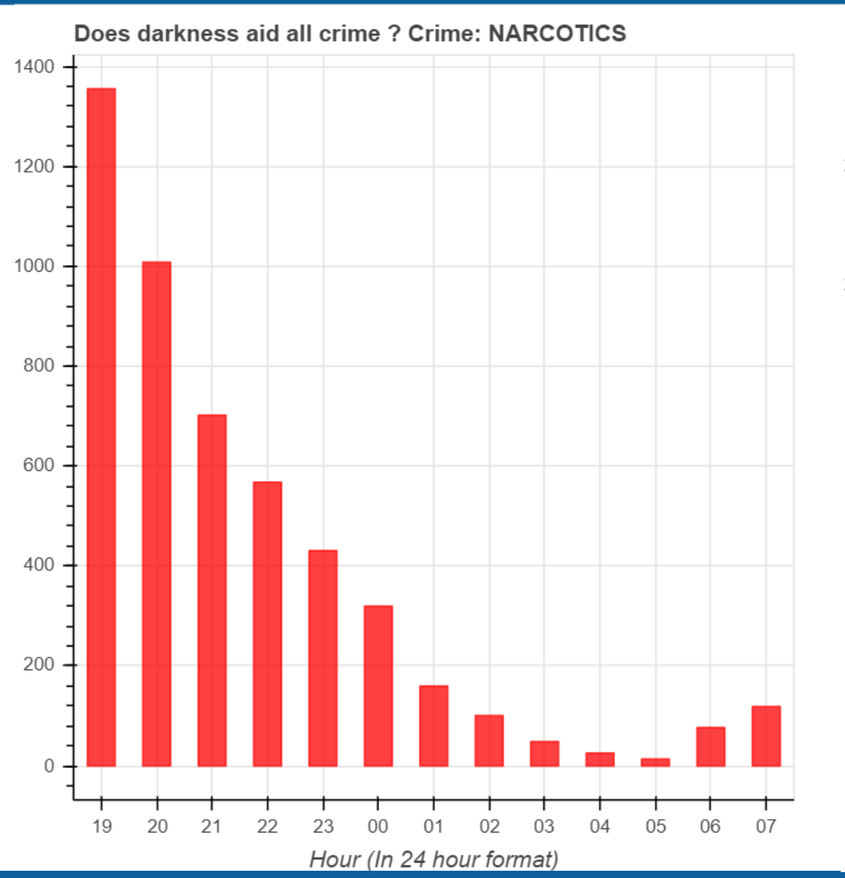


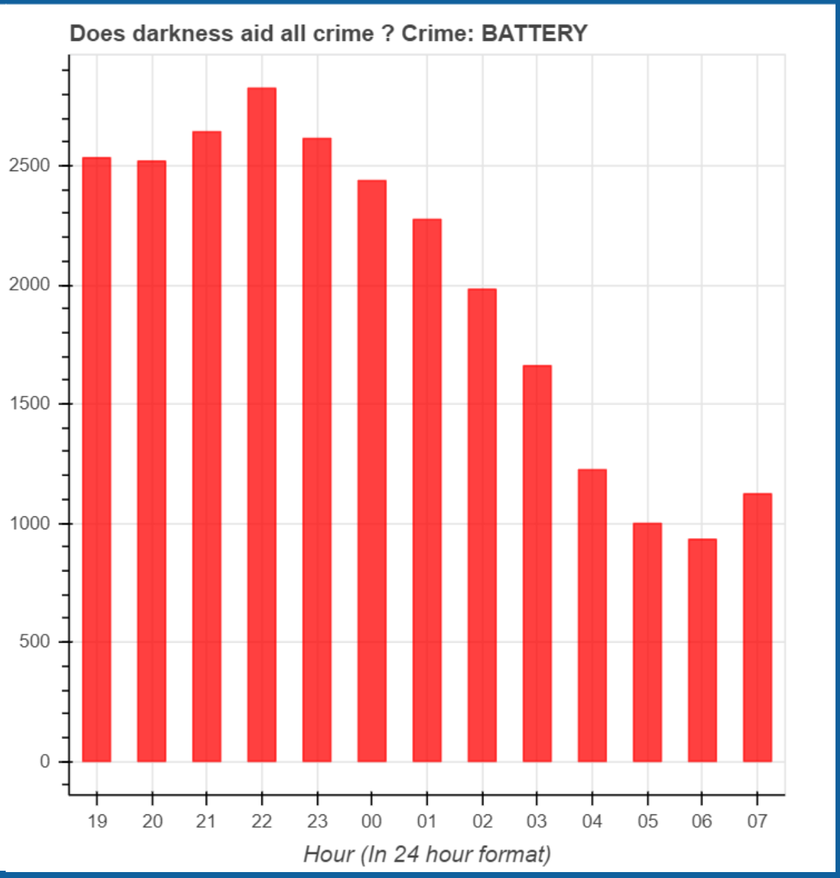
- The heat might lead to aggressive behaviour
- Vacations, so more people

- Criminals need sleep too !

Does darkness aid all crime ?





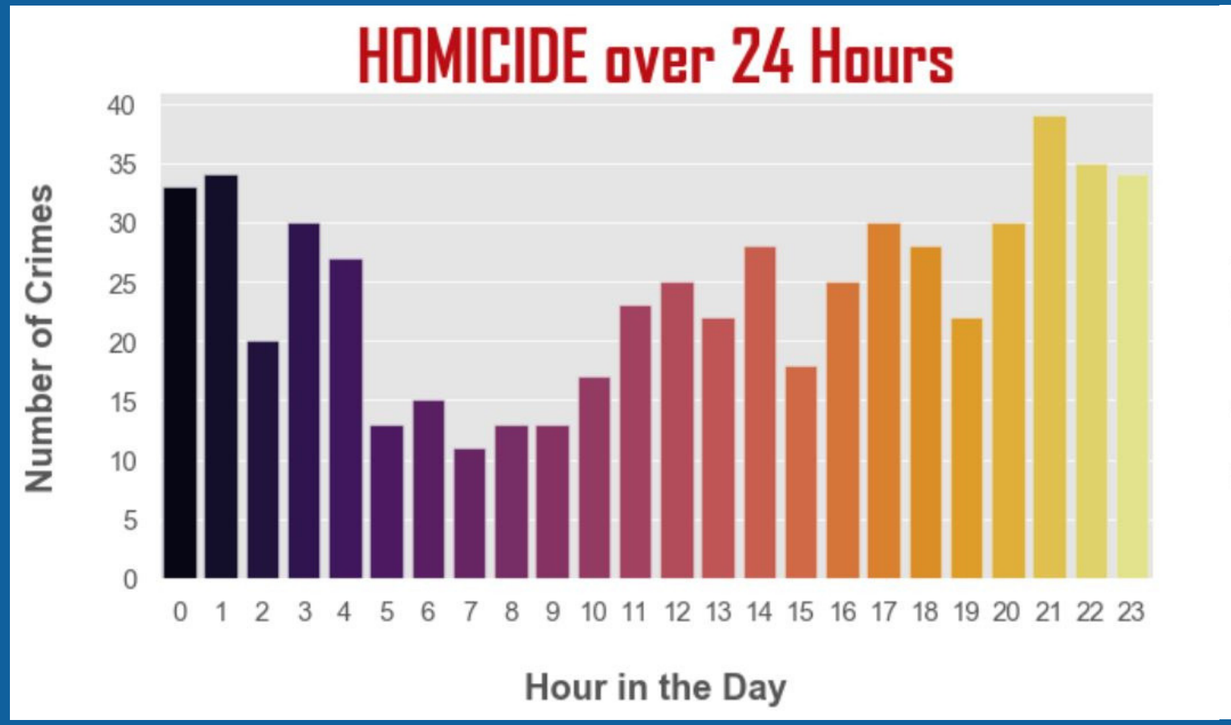


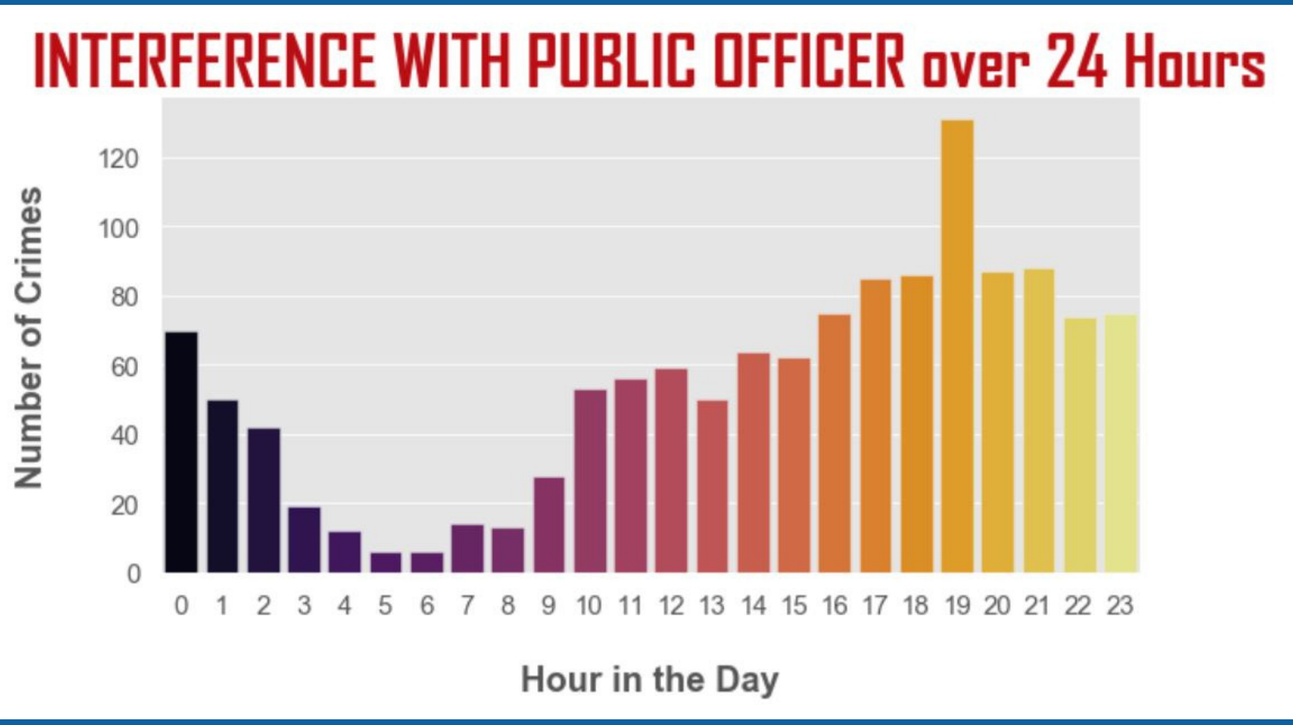
Theft starts dropping after sunset

Narcotics drops too after sunset

Battery rises after sunset, hits a peak and starts dropping

Crime Pattern over the day (Few Examples)

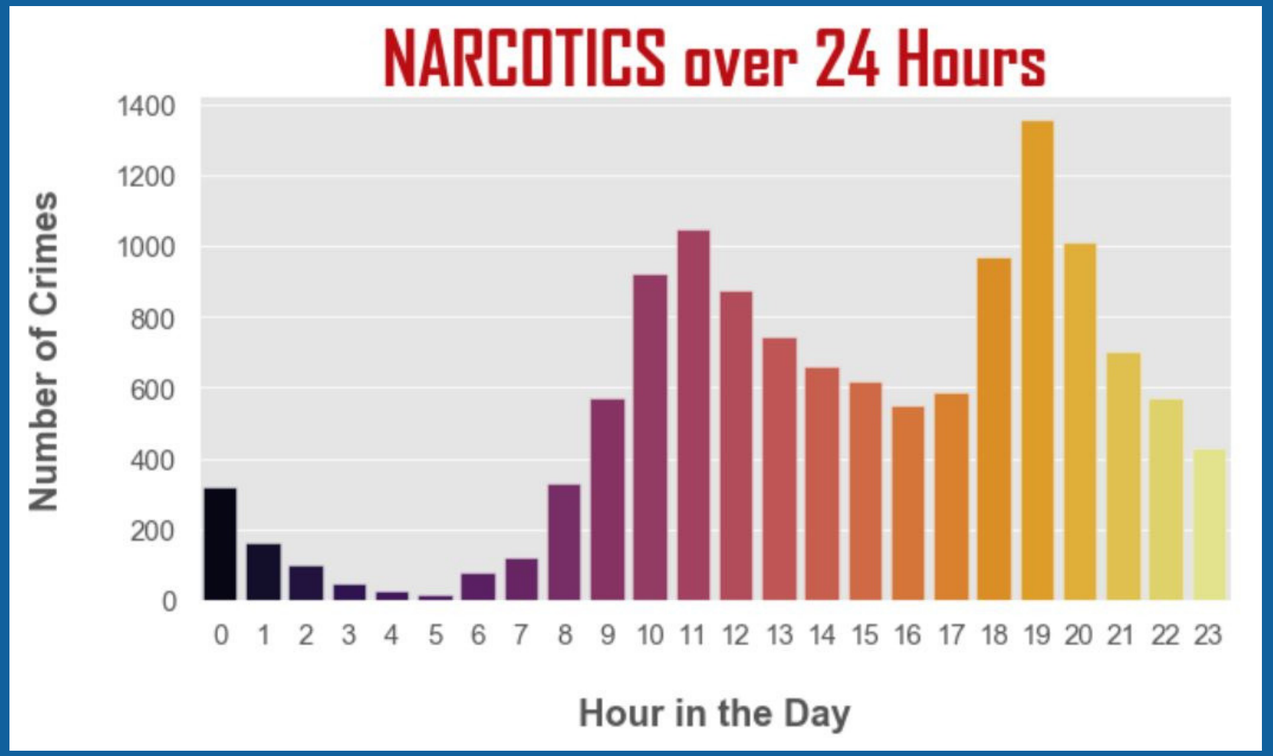




After sunset, Homicide rates increase upto 1:00 a.m. This can be related to "Organized Crime"

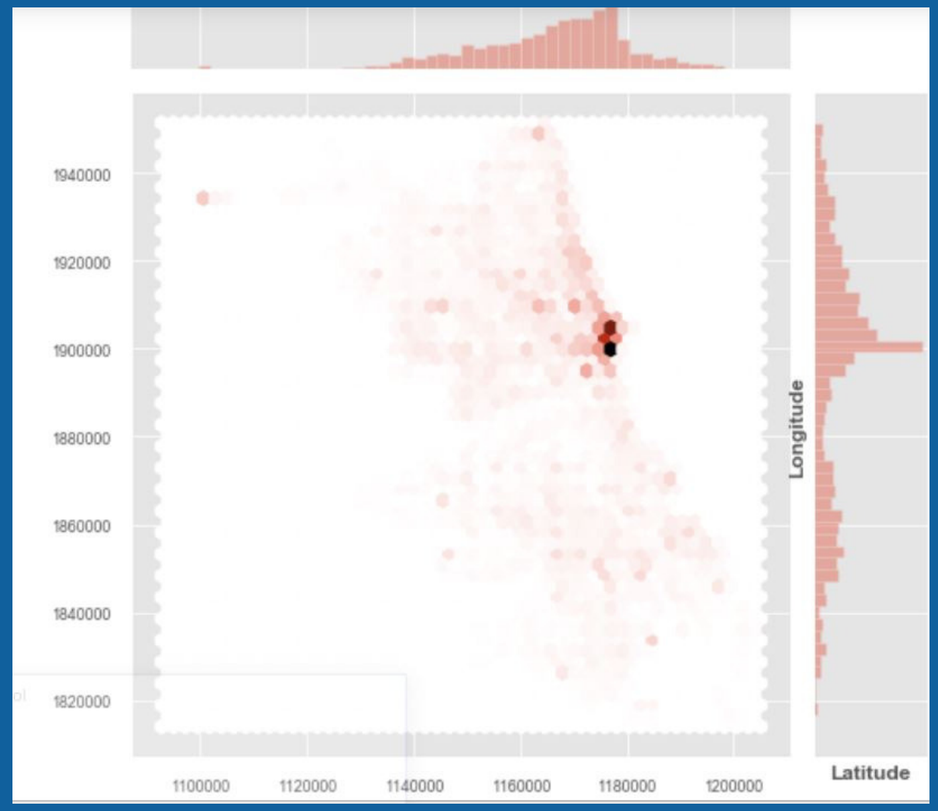
Maybe, there is maximum traffic on the road at 7:00 p.m !

Crime Pattern over the day (Few Examples)

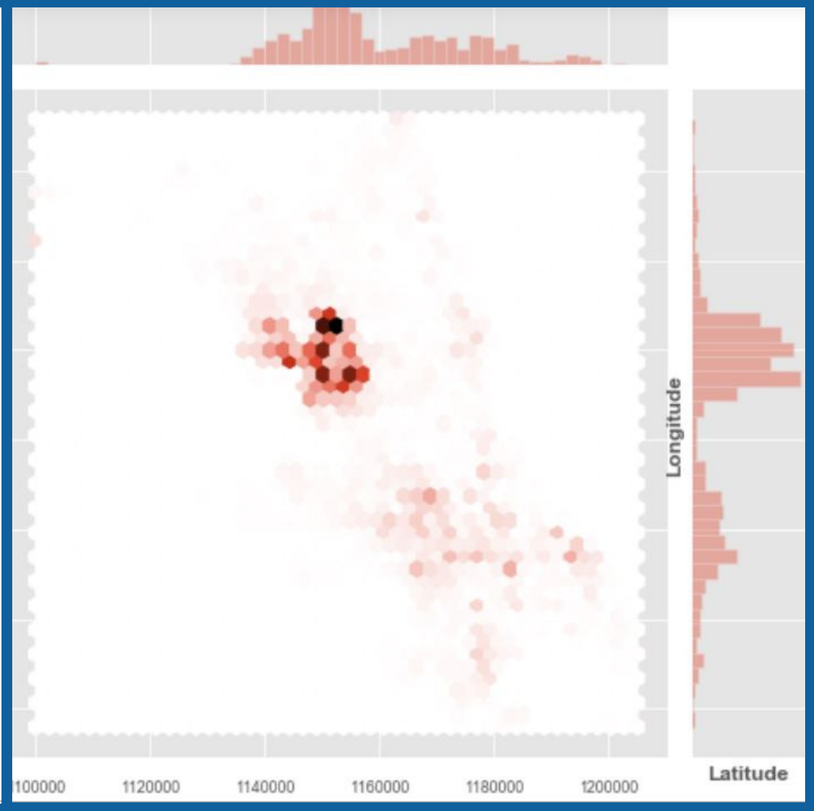


Clear Indicative Patterns of Organized Crime !

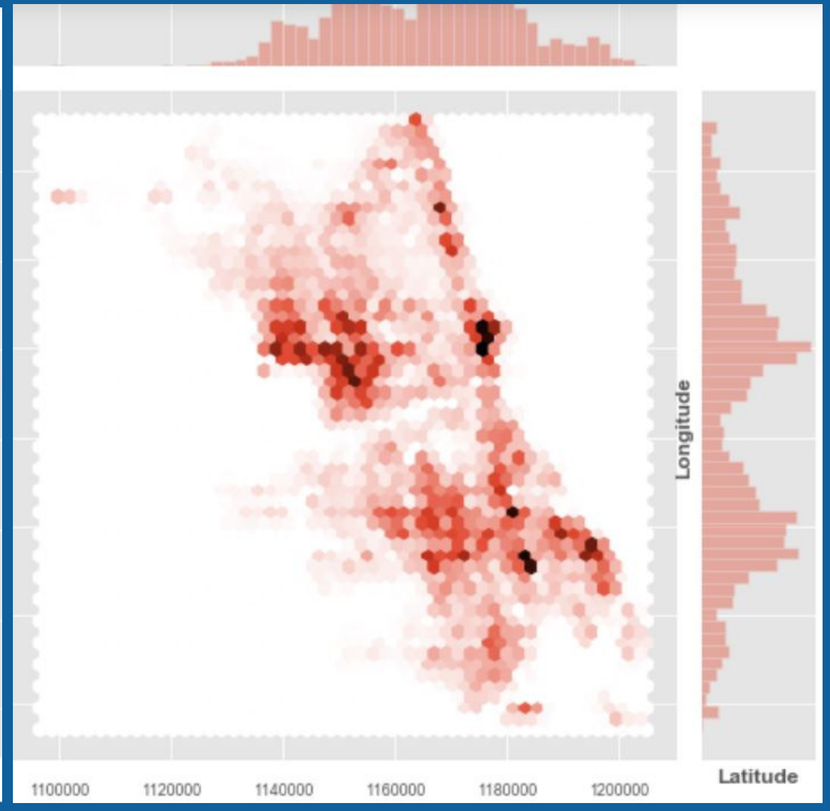
Crime Concentration in Chicago in 2018 (Few Examples)



Theft is spread across Chicago with a large concentration in Mid East Chicago

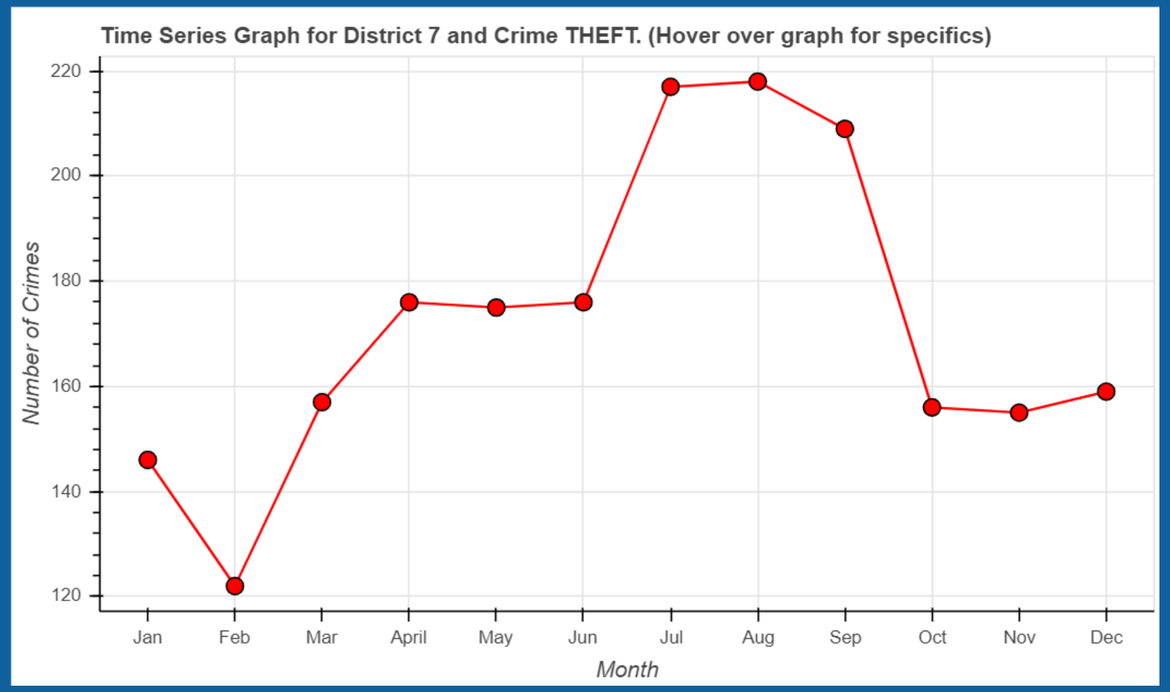


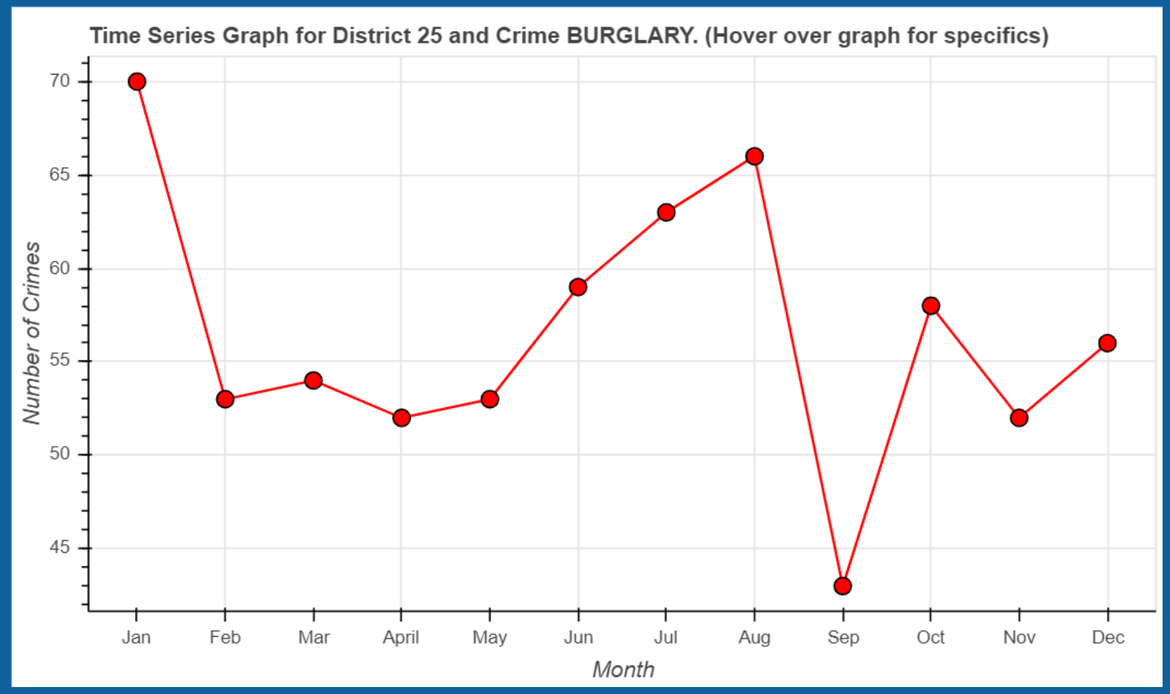
Narcotics is "Highly Prevalent" in the Western part. Can be an indicator of a narcotics gang here.



Battery Crimes have no exclusive localization.

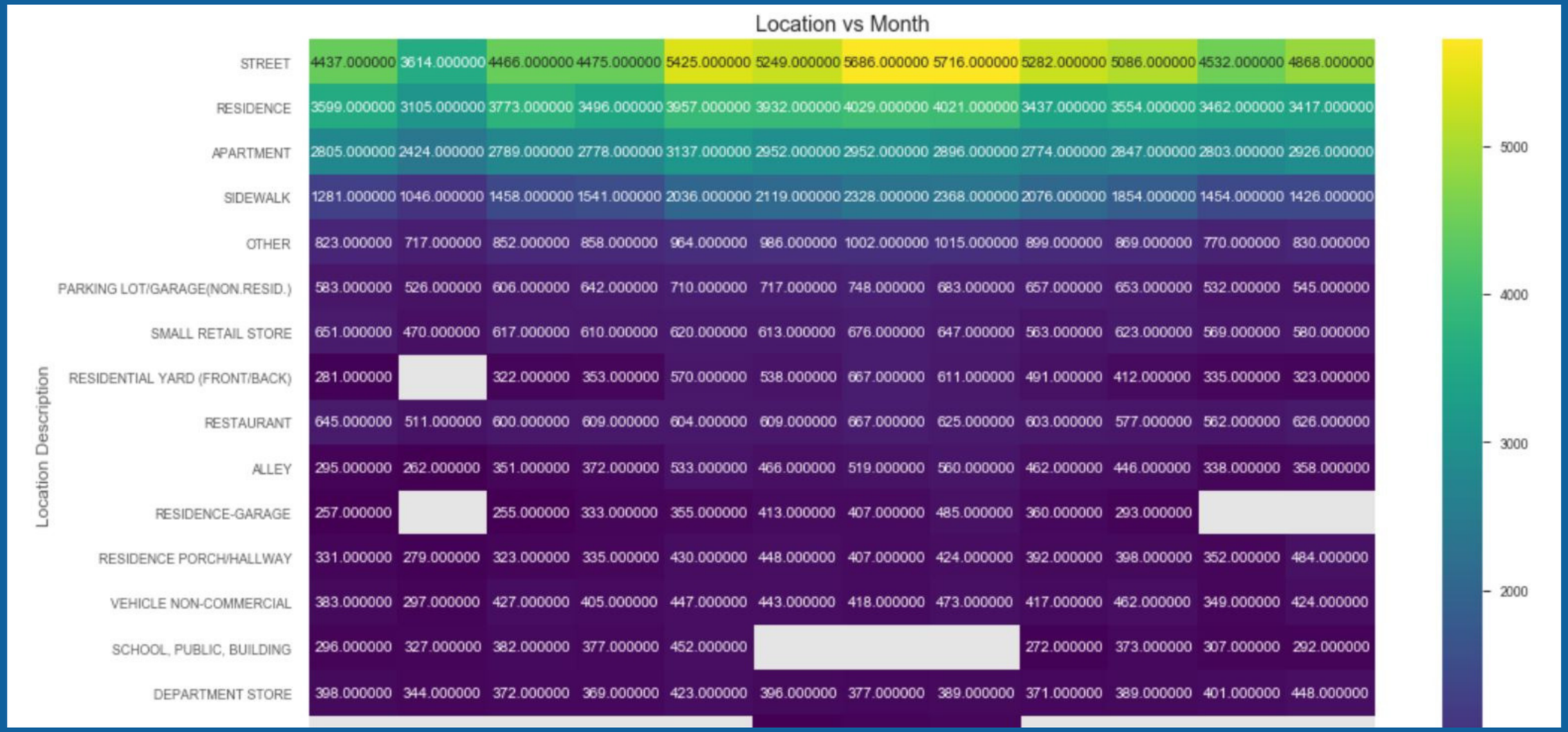
Choose district; Choose crime; Visualize
(2 examples...there are 26 x 32 combinations possible)



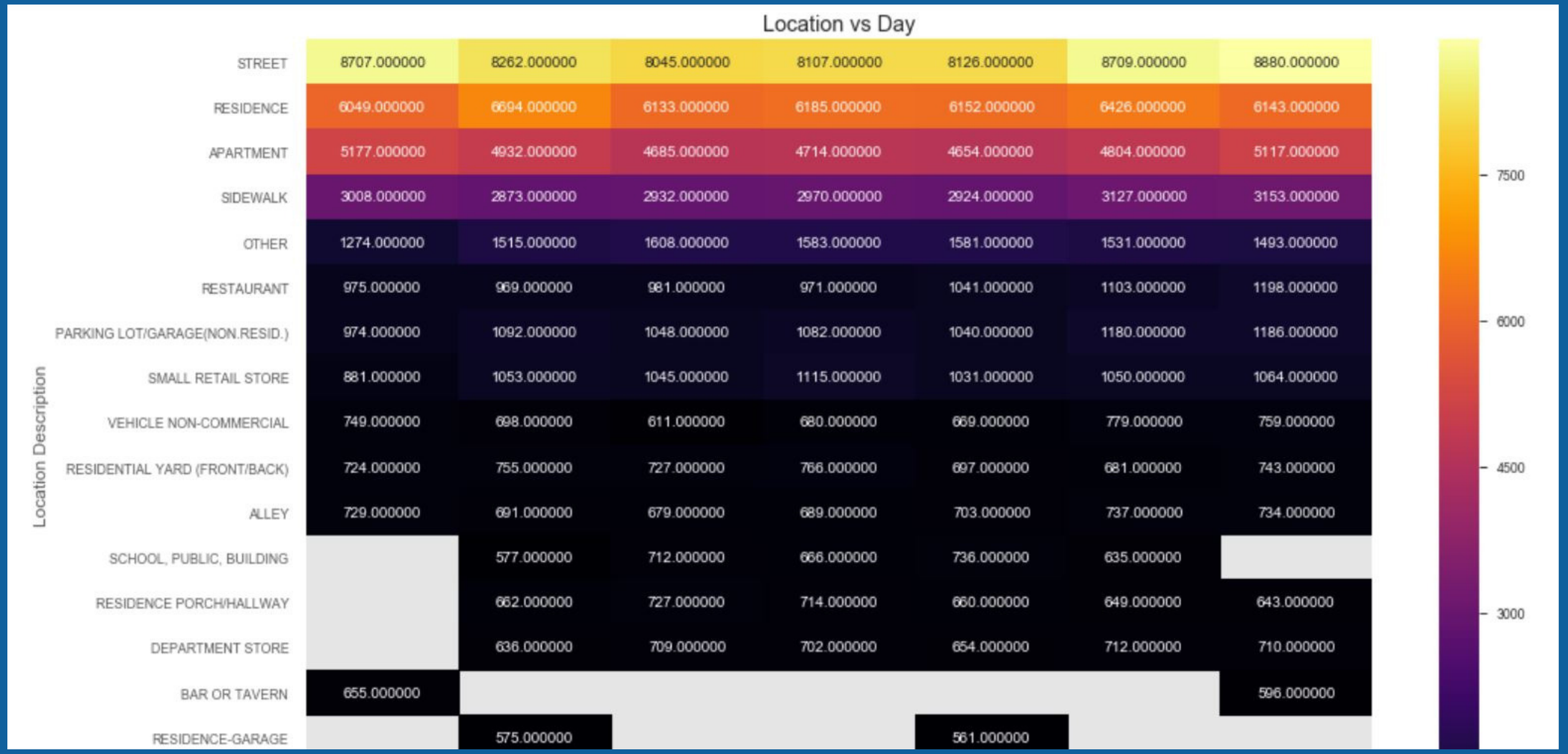


Such Interactive plots will give Decision Makers the chocie to make the right decision from the right data

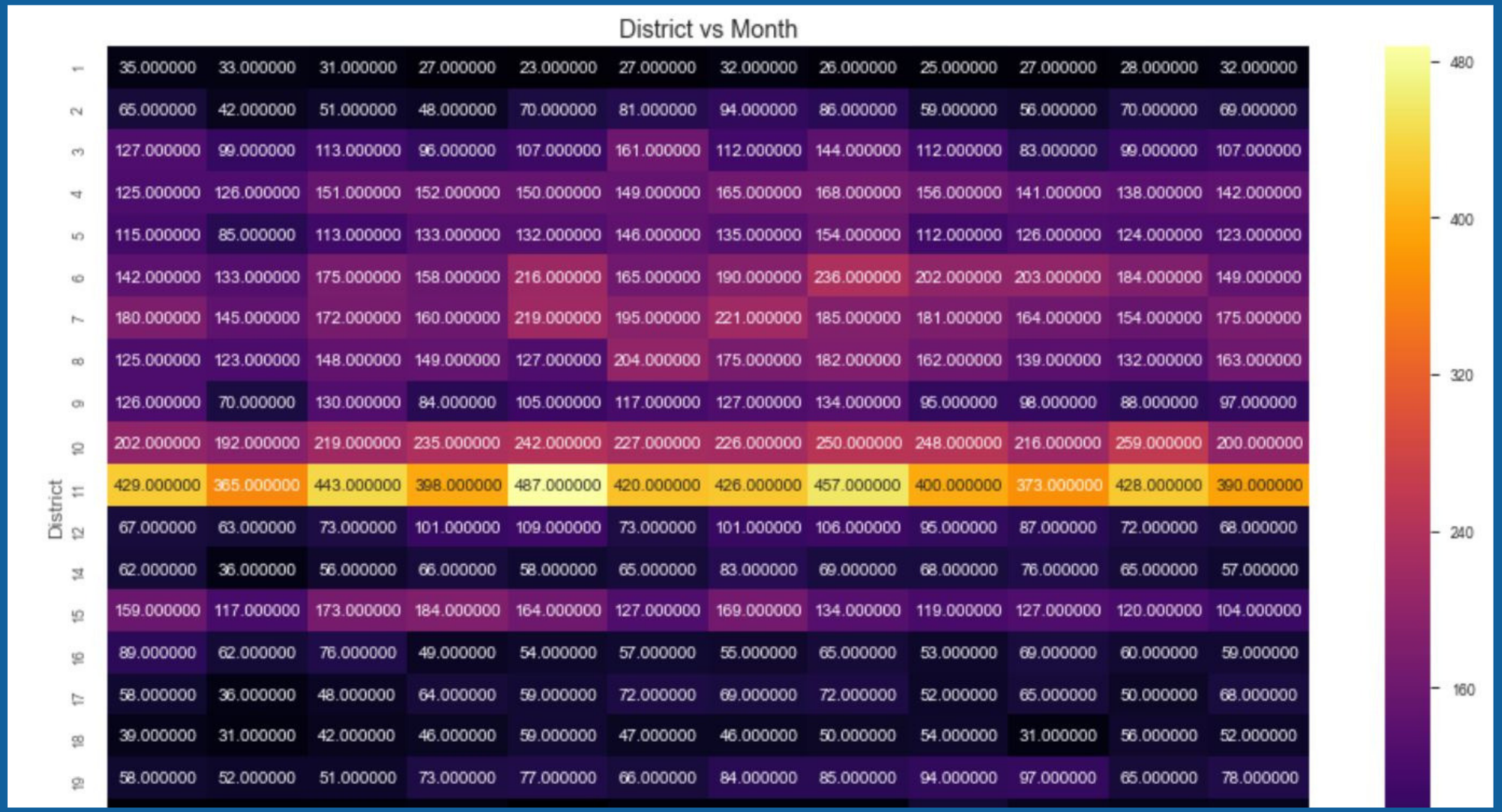
Heatmap 1 : Location vs Month



Heatmap 2 : Location vs Day



Heatmap 3 : District vs Month



Algorithm 1 : Classifying into Crime Type

```
1  X = df[['Arrest', 'Domestic', 'Beat', 'Community Area',
2                    'Latitude', 'Longitude', 'Year', 'Hour_Day']]
3  y = df['Primary Type']
```

```
1  clf = OneVsRestClassifier(AdaBoostClassifier())
2  clf.fit(X_train, y_train)
```

```
OneVsRestClassifier(estimator=AdaBoostClassifier(algorithm='SAMME.R', base_estimator=None,
          learning_rate=1.0, n_estimators=50, random_state=None),
          n_jobs=None)
```

```
1  print(("Accuracy: %.3f%% (%.3f%%)") % (results.mean()*100.0, results.std()*100.0))
```

```
Accuracy: 36.537% (0.090%)
```

Algorithm 2 : Identifying Crime Hotspots

```
# Creating our explicit dataset
cri4 = new_df.groupby(['Month_num','Day','District','Hour'], as_index=False).agg({"Primary Type":"count"})
cri4 = cri4.sort_values(by=['District'], ascending=False)
cri4.head()
```

	Month_num	Day	Hour	District	Primary Type	Alarm
27483	8	4	9	31	1	0
20779	6	5	18	31	1	0
18214	6	0	21	31	1	0
22331	7	1	14	25	13	1
22329	7	1	12	25	10	1

These are the categories we have specified :

- 0 : Low Alarm
- 1 : Medium Alarm
- 2 : High Alarm

So, let's take 7 as the threshold values for categorising the crimes as low and high crime rates.

- 0-7 : Low Crime Rate
- 8-15 : Medium Crime Rate
- 16-33 : High Crime Rate

Algorithm 2 : Identifying Crime Hotspots

```
Accuracy: 70.33850757434075


----------Confusion Matrix-----------------------------------
Predicted Alarm     0     1    2
Actual Alarm
0                5926  1421   59
1                1316  1516  182
2                  50   144   80

----------Classification Report-------------------------------------
              precision    recall  f1-score   support

           0       0.81      0.80      0.81      7406
           1       0.49      0.50      0.50      3014
           2       0.25      0.29      0.27       274

   micro avg       0.70      0.70      0.70     10694
   macro avg       0.52      0.53      0.52     10694
weighted avg       0.71      0.70      0.71     10694


UAR -> 0.5317062995784726
```

Decision Tree

```
Accuracy: 73.91995511501777


----------Confusion Matrix-----------------------------------
Predicted Alarm     0     1   2
Actual Alarm
0                6398   949  10
1                1524  1453  71
2                  49   186  54

----------Classification Report-------------------------------------
              precision    recall  f1-score   support

           0       0.80      0.87      0.83      7357
           1       0.56      0.48      0.52      3048
           2       0.40      0.19      0.25       289

   micro avg       0.74      0.74      0.74     10694
   macro avg       0.59      0.51      0.54     10694
weighted avg       0.72      0.74      0.73     10694


UAR -> 0.5110684007157605
```

Random Forest

THANK YOU