Chicago Crime Social Network Analysis

By: Madisen LeShoure, Daan Mansour, Ji Eun Kim

Brief Overview of the Analysis

Introduction

- The Dataset
- Identify Unique Variables
  - Nodes, Edges, Node Attributes
- Subsetting and extracting nodes
- Creating edges
- Adding nodes
- Assigning node attributes
  - Color, shape
- Visualizing the network structure
- Findings of analysis

What social network do we want to analyze?

- Our analysis aims to investigate the network of crime occurrences in Chicago using social network analysis.
- By examining the connections between geographic location and crime descriptions, we aim to reveal patterns and insights into the city's crime dynamics.
- The purpose of this analysis is to explore the interconnected nature of crime & geographic location in Chicago through the lens of social network analysis.
- Determine if there are prevalent communities of crime
- Determine Centrality, Degree Centrality

The Dataset

CHICAGO DATA PORTAL

- Our dataset comes directly from the Chicago Police Department via their Data Portal.
- This dataset contains every reported incident of crime (except murders) that have occurred in Chicago over the past year.
  - Minus the most recent seven days of data.
- The data is extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system and is updated weekly.
- 258K incidents

```python
import pandas as pd
df = pd.read_csv('Crimes_-_One_year_prior_to_present_20240418.csv')
df
```

✓ 1.2s

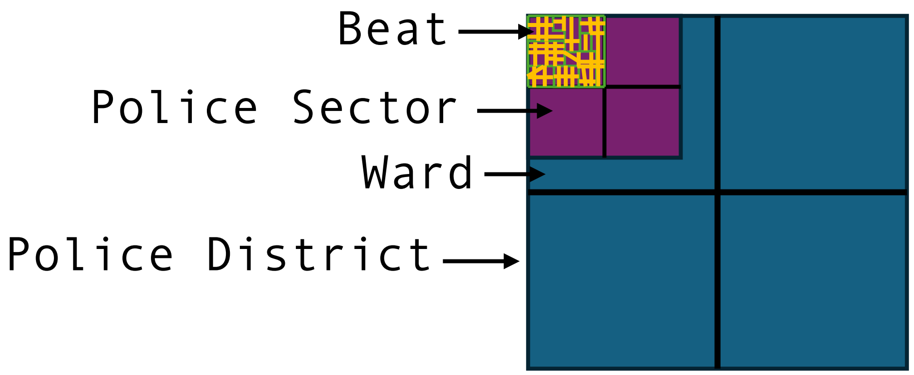
	CASE#	DATE OF OCCURRENCE	BLOCK	IUCR	PRIMARY DESCRIPTION	SECONDARY DESCRIPTION	LOCATION DESCRIPTION	ARREST	DOMESTIC	BEAT	WARD	FBI CD
0	JG497095	11/08/2023 08:50:00 PM	025XX N KEDZIE BLVD	0810	THEFT	OVER $500	STREET	N	N	1414	35.0	06
1	JG496991	11/08/2023 03:14:00 PM	0000X W CHICAGO AVE	0560	ASSAULT	SIMPLE	STREET	N	N	1832	42.0	08A
2	JG497145	11/08/2023 10:55:00 PM	019XX W 47TH ST	051A	ASSAULT	AGGRAVATED - HANDGUN	SIDEWALK	N	N	931	15.0	04A
3	JH179051	03/07/2024 02:15:00 PM	070XX S STATE ST	0820	THEFT	$500 AND UNDER	GROCERY FOOD STORE	Y	N	322	6.0	06
4	JH178785	03/07/2024 04:53:00 AM	077XX S CARPENTER ST	0810	THEFT	OVER $500	STREET	N	N	612	17.0	06
...	...	...	...	...	...	...	...	...	...	...	...	...
258121	JG373700	07/01/2023 06:10:00 PM	038XX N Clark ST	1154	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT $300 AND UNDER	NaN	N	N	1923	44.0	11
258122	JG300737	06/14/2023 12:07:00 PM	087XX S MUSKEGON AVE	141C	WEAPONS VIOLATION	UNLAWFUL USE - OTHER DANGEROUS WEAPON	ALLEY	N	N	423	7.0	15

The Dataset: Unique Variables

- There are 17 attributes in the dataset
  - I.e., Crime location description, FBI CD, case #, date of occurrence, IUCR, ward, etc.
- We focused on 3 key attributes
  - **Beat Numbers**
    - A beat is the smallest police geographic area - each beat has a dedicated police beat car. The beat indicates where the crime has occurred on the smallest geographic scale.
  - **Crime Primary Description**
    - Primary Description of the Illinois Uniform Crime Reporting (IUCR) of each crime incident.
  - **Ward**
    - The ward is the City Council district where the incident occurred.

```
df.columns
```

```
Index(['CASE#', 'DATE  OF OCCURRENCE', 'BLOCK', ' IUCR',
       ' PRIMARY DESCRIPTION', ' SECONDARY DESCRIPTION',
       ' LOCATION DESCRIPTION', 'ARREST', 'DOMESTIC', 'BEAT', 'WARD', 'FBI CD',
       'X COORDINATE', 'Y COORDINATE', 'LATITUDE', 'LONGITUDE', 'LOCATION'],
      dtype='object')
```



3 to 5 beats make up a police sector, and three sectors make up a police district. The Chicago Police Department has 22 police districts.

Unique Variables

- Nodes
  - The unique variable that serves as nodes is ‘BEAT’. Because Beats make up wards. Beats connect different areas of the city based on crime occurrences, which can in turn reveal patterns in crime and identify communities of crime.
    - Represented by Shape [-]
  - The second unique variable that serves as nodes is 'PRIMARY DESCRIPTION'. Crime descriptions establish the connections between beats in the network.
    - Represented by Shape [0]
- Edges
  - Edges represent the frequency of crime within a beat. Denoted by a number or thickness of the line connecting nodes.
- Node Attributes
  - For node classification we use wards to classify the beats (beats are associated with wards)
    - Denoted by the color of the nodes
  - Centrality
  - Degree centrality

```
df[df['BEAT'] == 322]
```

✓ 0.0s

	CASE#	DATE OF OCCURRENCE	BLOCK	IUCR	PRIMARY DESCRIPTION	SECONDARY DESCRIPTION	LOCATION DESCRIPTION	ARREST	DOMESTIC	BEAT	WARD
3	JH179051	03/07/2024 02:15:00 PM	070XX S STATE ST	0820	THEFT	$500 AND UNDER	GROCERY FOOD STORE	Y	N	322	6.0
140	JH118133	01/16/2024 01:00:00 PM	001XX E 70TH ST	1310	CRIMINAL DAMAGE	TO PROPERTY	APARTMENT	N	N	322	6.0
327	JG496329	11/08/2023 09:30:00 AM	005XX E 71ST ST	1570	SEX OFFENSE	PUBLIC INDECENCY	CTA BUS STOP	N	N	322	6.0
392	JG513792	11/21/2023 04:00:00 PM	070XX S MICHIGAN AVE	0486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	N	Y	322	6.0
714	JG513564	11/21/2023 04:00:00 PM	069XX S STATE ST	0870	THEFT	POCKET-PICKING	CTA TRAIN	N	N	322	6.0
...	...	...	...	...	...	...	...	...	...	...	...
257136	JG291027	06/06/2023 07:36:00 PM	069XX S STATE ST	0470	PUBLIC PEACE VIOLATION	RECKLESS CONDUCT	STREET	Y	N	322	6.0
257538	JG263612	05/17/2023 04:17:00 AM	006XX E 71ST ST	1310	CRIMINAL DAMAGE	TO PROPERTY	APARTMENT	N	N	322	6.0
257740	JG298834	06/13/2023 01:00:00 AM	069XX S INDIANA AVE	0820	THEFT	$500 AND UNDER	STREET	N	Y	322	6.0
258030	JG278857	05/12/2023 09:00:00 AM	004XX E 69TH ST	0810	THEFT	OVER $500	APARTMENT	N	N	322	6.0
258031	JG377607	08/10/2023 06:00:00 PM	066XX S STATE ST	0460	BATTERY	SIMPLE	GAS STATION	N	N	322	6.0

Subsetting and Extracting Nodes into Pandas DataFrame

- Our data consists of 258126 **rows** and **17 columns**. We need to subset our rows and columns for the following reasons:
  1) Size of data
  2) **Selecting necessary variables which are:**

```python
data = df[['BEAT',' PRIMARY DESCRIPTION','WARD']]
data
```
[130] Python

```python
beat_counts = data["beats"].value_counts()
sorted_counts = beat_counts.sort_values(ascending=False)
t100_beats = sorted_counts.head(100)
t100_beats
```
[135] Python

```
beats
1834    3162
123     2086
1831    1917
421     1903
624     1852
        ...
1215    1048
114     1046
1223    1041
924     1039
914     1033
Name: count, Length: 100, dtype: int64
```

Nodes: 'BEAT' -> 'beat', 'PRIMARY DESCRIPTION' -> 'primary description'
Nodes attribute: color based on 'WARD' -> 'ward', degree centrality

Subsetting and Extracting Nodes into Pandas DataFrame

```python
fdata = data[data['beats'].isin(t100_beats.index)]
fdata
```

[43] ✓ 0.0s

	beats	primary descrip	wards
4	612	theft	17.0
8	822	other offense	14.0
9	612	motor vehicle theft	17.0
16	413	assault	7.0
18	1112	theft	27.0
...	...	...	...
258117	211	motor vehicle theft	4.0
258119	1031	battery	22.0
258122	423	weapons violation	7.0
258123	312	battery	20.0
258124	1824	battery	2.0

129410 rows × 3 columns

We filtered '**data**' based on whether the values in the '**beats**' column are in the '**t100_beats**' and the results are put into '**fdata**'

Creating Edges

```python
fdata['new_col'] = list(zip(fdata['primary descrip'], fdata['beats']))
fdata['new_col']
```

```python
edges=[]
for idx, val in fdata.iterrows():
    if len(val['new_col']) == 0: #when there are no mentions, we skip the iteration
        continue
    for beat in val['new_col']:
        edges.append((val['beats'], val['primary descrip']))
            #we append the tuple of the beat and the crime description to the edges list
```

✓ 1.3s

- In making our edges, we iterated through our column which contained a list with each entry being the primary description of the crime and the beat it took place on

Adding Nodes

```
for node in G.nodes():
    if node in fdata['beats'].values: #check if the node is in the username column
        G.nodes[node]['wards']=fdata[fdata['beats']==node]['wards'].unique()[0]
    else: #if the node is not in the username column, we assign the title attribute as Unknown
        G.nodes[node]['wards']='Unknown'
```

✓ 1.3s

- When adding our nodes, we made sure that if there was an issue with the dataset that it wouldn't cause issues on our end. We made sure that if the beat somehow didn't have a ward that it'd be marked as such with its ward being named 'Unknown' which could clue us in to missing data since a beat must have a ward it belongs to.

Assigning color as Node Attribute

```
import networkx as nx
G = nx.Graph()
G.add_edges_from(edges)
```
[17] ✓ 0.2s

```
len(list(G.nodes))
```
[18] ✓ 0.0s

… 130

- Unique values of '**wards**' are retrieved and assigned to a randomly generated seaborn hls color palette which is stored in a new dictionary '**state_colors_dict**'

- We created a color palette by first creating an empty dictionary '**state**' and mapped '**wards**' to each corresponding column in '**beats**' which is stored in the '**state**' dictionary.

```
state={}
for idx, row in fdata.iterrows():
    state[row['beats']]=row['wards']

import random
import seaborn as sns
# Get the unique values from the state dictionary
unique_states = list(set(state.values()))

# Generate a color palette using seaborn
color_palette = sns.color_palette("hls", len(unique_states))

# Create a dictionary to map each unique state to a color
state_colors_dict = {key: color_palette[i] for i, key in enumerate(unique_states)}
```
✓ 1.7s

Assigning color as Node Attribute

- We created a new dictionary called '**color_mapped**' where each beat (key) is associated with a color
- If a ward has a color assigned in '**state_colors_dict**', the corresponding beat in '**color_mapped**' gets that color, if not, it remains 'None'
- Nodes from '**beats**' are then assigned a color '**(0,0,0)**' corresponding from the '**color_mapped**' dictionary

```
from collections import defaultdict
default_dict=defaultdict(lambda: None, state)
for key, value in state.items():
    if value in state_colors_dict.keys():
        default_dict[key]=state_colors_dict.get(value)
color_mapped=dict(default_dict)
```

[56] ✓ 0.0s

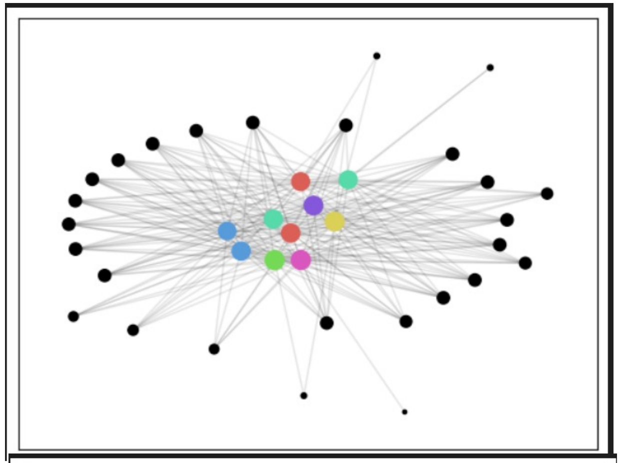
```
for node in G.nodes():
    if node in fdata['beats'].values:
        G.nodes[node]['color']=color_mapped[node]
    else:
        G.nodes[node]['color']=(0,0,0)
```

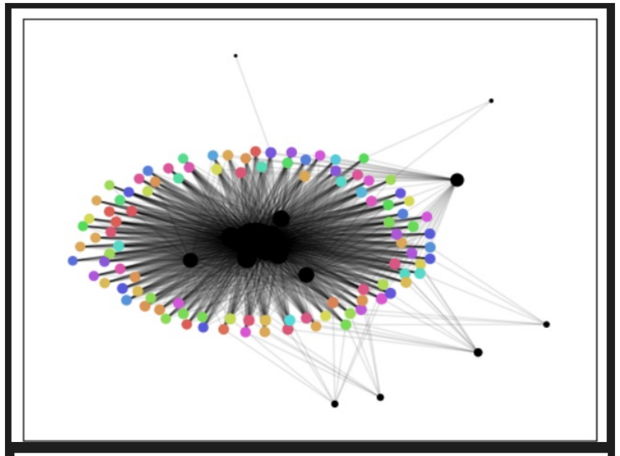
[57] ✓ 0.1s

Visualizing Network Structure through Degree Centrality

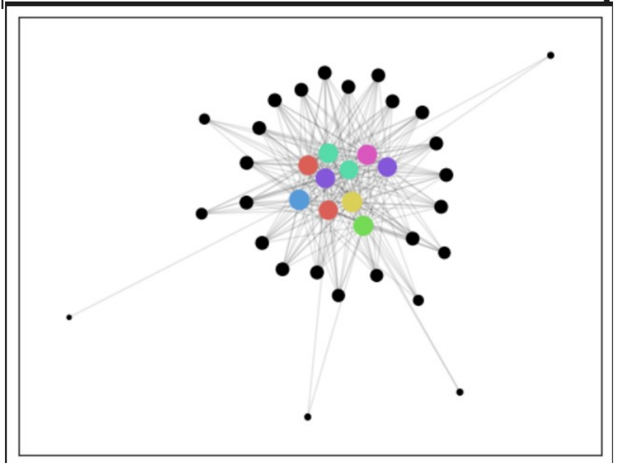
Top 10 most common beats

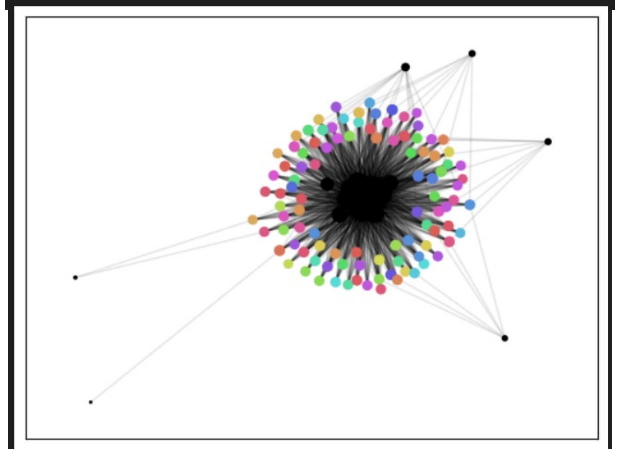
Top 100 most common beats





Kamada Kawai

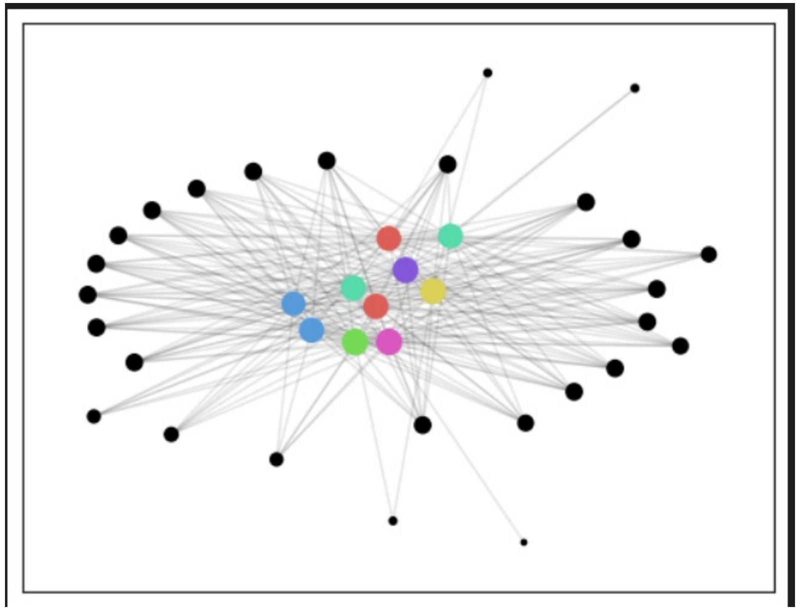




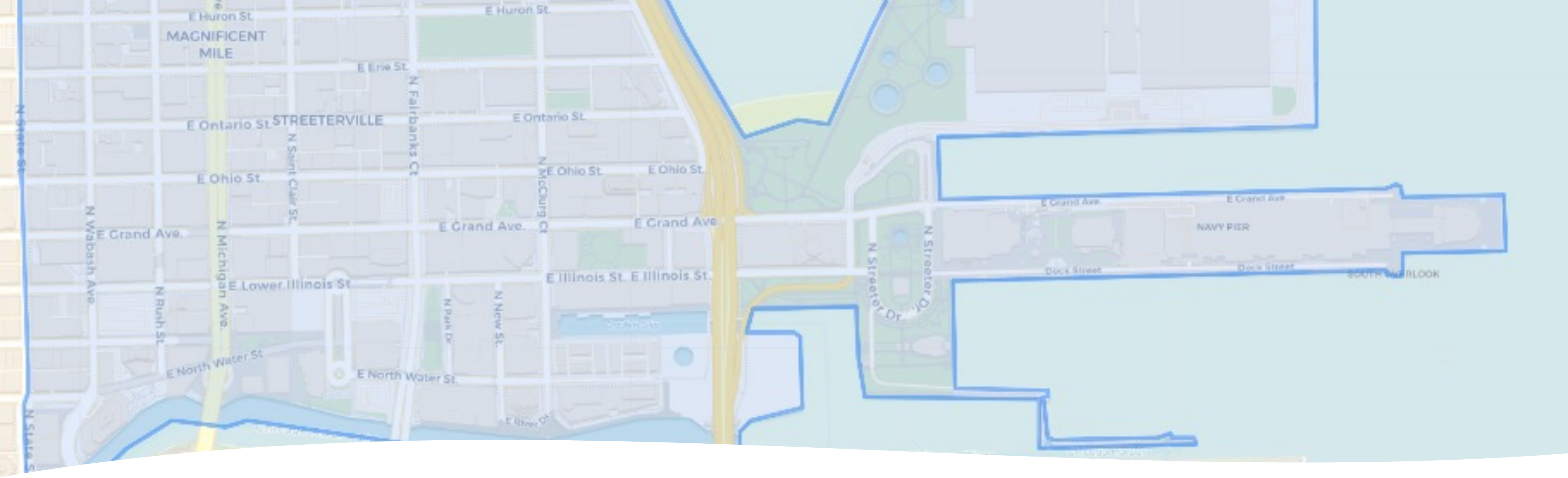
Spring Layout

Analysis on Visualization

```
#primary descrip
#theft                                  57385
#battery                                44648
#criminal damage                        29791
#motor vehicle theft                    26912
#assault                                22813
#other offense                          15876
#deceptive practice                     15466
#robbery                                10993
#weapons violation                       8411
#burglary                                7315
#narcotics                               5323
#criminal trespass                       4623
#offense involving children              1652
#criminal sexual assault                 1571
#sex offense                             1298
#public peace violation                   884
#homicide                                 612
#interference with public officer         586
#arson                                    507
#stalking                                 490
#intimidation                             191
#concealed carry license violation        191
#prostitution                             187
#liquor law violation                     185
#kidnapping                               134
#obscenity                                 47
#gambling                                  16
#human trafficking                          8
#public indecency                           6
#Name: count, dtype: int64
```



With the top ten most common beats and as the graph shows, highest frequency crimes we can see that despite the varying frequencies of certain crimes, crime as a total has remained at a similar volume across each of the ten beats.



Where is it happening.

- By sorting our list of wards by most common, beat 1834 comes up a total of 3612 times for this past year. This beat covers Navy Pier and the neighborhood of Streeterville.

Analysis

Degree centrality: Node connectivity, local influence

Closeness centrality: Proximity to other nodes, efficient communication

Betweenness centrality: Bridging roles, broker

```
sorted(nx.closeness_centrality(G).items(), key=lambda x:x[1], reverse=True)[:5]
```

✓ 0.0s

```
[('theft', 0.8164556962025317),
 ('other offense', 0.8164556962025317),
 ('motor vehicle theft', 0.8164556962025317),
 ('assault', 0.8164556962025317),
 ('battery', 0.8164556962025317)]
```

+ Code + Markdown

```
sorted(nx.degree_centrality(G).items(), key=lambda x:x[1], reverse=True)[:5]
```

✓ 0.0s

```
[('theft', 0.7751937984496124),
 ('other offense', 0.7751937984496124),
 ('motor vehicle theft', 0.7751937984496124),
 ('assault', 0.7751937984496124),
 ('battery', 0.7751937984496124)]
```

```
sorted(nx.betweenness_centrality(G).items(), key=lambda x:x[1], reverse=True)[:5]
```

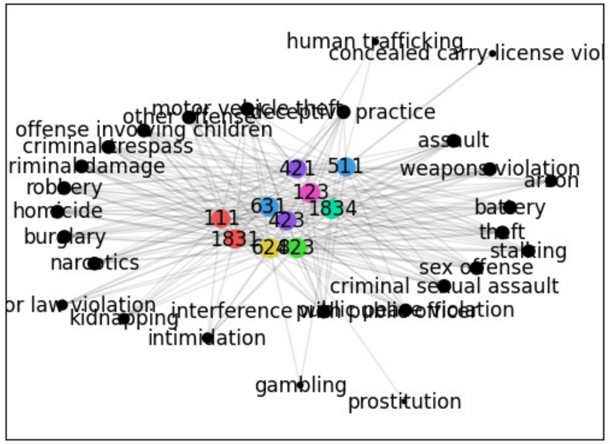
✓ 0.1s

```
[('theft', 0.036644015983084643),
 ('other offense', 0.036644015983084643),
 ('motor vehicle theft', 0.036644015983084643),
 ('assault', 0.036644015983084643),
 ('battery', 0.036644015983084643)]
```

Conclusion

- In conclusion, we found that many different crimes proportionally create an almost equal level of crime across multiple beats, despite the different frequency at which they happened.

- Going further, it'd be interesting to look at things such as social determinants of health to understand the root of the issue and the effects it can cause on the communities that these beats cover.
    - Top 10 most common beats



    - Top 100 most common beats