Economic Disparity and Its Influence on Non-Violent and Violent Crime Patterns Across Chicago's Community Areas

1.Project 1

1.1 Introduction

1.1.1 Introduction and Data Source

Chicago's diverse urban landscape is concurrently marked by the complex challenges of multifaceted criminal activities. To comprehensively understand the dynamics of crime patterns within the city, it's imperative to consider a broad spectrum of factors. This study delves into the economic variables that influence the variation in non-violent and violent crime incidents across different community areas in Chicago from 2014 to 2018. Utilizing community area, primary crime type, location description, arrest rate, and domestic variables as independent factors, and crime counts as the dependent variable, the research aims to uncover the underlying trends in crime incidents across geographic locales. The analysis highlights the distribution of crime type patterns, incorporating income level and unemployment rate to scrutinize the socio-economic factors affecting crime rates.

In clarifying the distribution of different crime types, the term 'crime rate' in this paper refers specifically to the number of particular crime incidents divided by the total crime occurrences within the area (e.g., 'theft rate' signifies the number of thefts divided by the total crime count, and similarly for 'battery rate'). Previous studies, such as those by Arnio and Baumer, have primarily focused on demographic factors influencing crimes, noting significant impacts from socio-economic disadvantages and residential stability on violent crime rates, with notable variances across Chicago's community clusters (Arnio & Baumer, 2012). Their work established a correlation between crimes and demographic aspects, such as the percentage of African American residents, hinting at the potential influence of demography on our findings. Consequently, while our research does not directly focus on demographic factors, we include population as a variable to control for in our regression analysis. Through this approach, we aim to provide insights for law enforcement strategies, community safety initiatives, and urban planning efforts to enhance protective measures against crime.

This research merged three four dataset. We utilize a public dataset from the City of Chicago, derived from the Chicago Police Department's CLEAR system, covering detailed crime information from 2001 to 2024, with our focus narrowed to the years 2014 to 2018. The dataset includes 22 columns of comprehensive crime data. To ensure data integrity, we performed preliminary data cleaning and statistical analysis to formulate accurate and

reliable conclusions. Since the original dataset primarily showcased crime type distribution, we integrated additional datasets from the Chicago Health Atlas to include socio-economic factors such as income level, unemployment rate, and educational attainment. Moreover, due to the absence of readily available datasets on housing prices in Chicago, we resorted to web scraping data from Hausmarket for average and median listing house prices by community areas.

Our findings reveal that community areas with higher theft rates tend to exhibit lower arrest rates compared to areas with higher battery rates. This discrepancy may suggest that areas with reported higher crime rates could be attributed to higher reporting rates, whereas lower crime rates in other areas might result from underreporting. This notion is supported by initiatives like the Community-Academic Collaboration to Prevent Violence in Chicago (CACPVC), which emphasizes the importance of community engagement in the research process to effectively address non-violent crimes such as theft (Ellyin et al., 2021). Areas with active community participation and reporting mechanisms have seen improvements in crime reduction and arrest rates, highlighting the critical role of community collaboration in crime prevention efforts. Therefore, our study also explores the relationship between arrest rates and socio-economic factors, including income level, to offer comprehensive strategies for reducing crime incidence.

Focusing on economic disparity, the research by Dong, Egger, and Guo examines income level and housing prices as indicators of absolute poverty and economic inequality, respectively, to determine their impact on crime types (Dong, Egger & Guo, 2020). Contrary to findings in other contexts that suggest poverty as the primary driver of violent crime, our study aims to explore the specific dynamics within Chicago, considering both economic equality (through housing prices) and income levels. This nuanced approach allows us to investigate the broader implications of economic conditions on crime rates, offering insights for governmental strategies to address these challenges. Additionally, Fabian and Abdul-Razak's research concludes that income plays a crucial role in fueling violence; they posit that government interventions to increase income could lead to a reduction in violent crimes (Fabian & Abdul-Razak, 2018). Drawing on these findings, our paper focuses on communities with low income levels and high rates of violent crime, offering insights for governmental strategies to address these issues. Anser and colleagues further recommend that the government should foster economic growth as a means to decrease crime rates nationally (Anser et al., 2020). By highlighting the importance of examining both wealth levels and income distribution, our research contributes to a deeper understanding of the complex interplay between economic factors and crime rates, ultimately providing evidence-based conclusions for more effective crime prevention strategies.

By understanding the relationship between various geographical areas and crime types in-depth, we may uncover the underlying patterns of criminal activity in Chicago, as well as identify the loopholes in law enforcement. Haller stated that the citizen reformers' moral values are not widely recognized by society, and their legal values are not supported by politicians and officials (Haller, 1970). It indicates the inconsistency between the citizens and governments, hence influencing the trust in government and

even in law. Building on the comprehensive analysis of economic factors influencing crime types in Chicago, the subsequent sections of this paper will delve deeper into the data-driven insights obtained from our study. This exploration will not only enhance our understanding of the complex dynamics shaping crime patterns in Chicago but also inform targeted interventions and policy-making. Through a detailed examination of the data, we will endeavor to unravel the nuanced interactions between economic variables and crime, paving the way for informed strategies to foster safer urban environments.

Source:https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2/about_data

Reference:

Anser, M. K., Yousaf, Z., Nassani, A. A., Alotaibi, S. M., Kabbani, A., & Zaman, K. (2020). Dynamic linkages between poverty, inequality, crime, and social expenditures in a panel of 16 countries: Two-step GMM Estimates. Journal of Economic Structures, 9(1). https://doi.org/10.1186/s40008-020-00220-6

Arnio, A. N., & Baumer, E. P. (2012). Demography, foreclosure, and crime: Demographic Research, 26, 449–488. https://doi.org/10.4054/demres.2012.26.18

Dong, B., Egger, P. H., & Guo, Y. (2020). Is poverty the mother of crime? evidence from homicide rates in China. PLOS ONE, 15(5). https://doi.org/10.1371/journal.pone.0233034

Ellyin, A., Day, K., Samuel, J., Bartell, T., McGill, D., Sheehan, K., & Levin, R. (2021). Community-Engaged Research to Develop a Chicago Violence Research Agenda and Recommendations to Support Future Community Engagement. https://doi.org/10.21203/rs.3.rs-469978/v1

Fabian Adekoya, A., & Abdul-Razak, N. A. (2018). Unemployment and violence: ARDL endogeneity approach. NOVIEMBRE 2018, 37(2). https://doi.org/10.29105/ensayos37.2-2

Haller, M. H. (1970). Urban crime and criminal justice: The chicago case. The Journal of American History, 57(3), 619. https://doi.org/10.2307/1917978

1.1.2 Outcome and explanatory variable

1.1.2.1 Y Variable (Outcome):

We choose crime_by_community as our outcome variable,which is a new variable that we retrive from the original dataset, counting crime incidents for each community area. For the independent variables, we also count crime incidents while groupby community area to keep consistency. We create four crime_count variables: gb_crime_type, gb_crime_loc, gb_crime_arrest and gb_crime_domestic.
It plays a pivotal role in our investigation as it encapsulates the crime rates in various district. By focusing on the number of crime incidents, we aim to discern patterns and relationships between this key variable and the selected explanatory variables. Understanding the count of crime provides a foundational framework for our research, enabling a more nuanced exploration into the geographical dynamics of criminal

activities in Chicago. As we delve into the relationships with the independent variables, the insights derived from crime_by_community will contribute significantly to unraveling the complex factors influencing the safety in different areas, ultimately informing strategies for effective law enforcement and crime prevention.

1.1.2.2 X Variables (Explanatory Variables):

1. Community Area:

The community was distinct into 77 community areas in Chicago, expressed by numeric code, the name of each area corresponds to the image shown below. Understanding the frequency of crime incidents across various community areas is the focus of this research, which reveals the crime pattern from aspects of social, economic and demographic factors.

2. Primary Type:

The primary type categorizes different crime incidence types. To understand law enforcement effectiveness, we choose Primary Type as another variable. Although column iucr is a formal code including description and type, we choose primary type because it is more informative and doesn't require the knowledge of iucr code. Analyzing crime types in relation to crime count can reveal insights into areas where law enforcement actions might be more successful or need improvement. Especially for non-violent and violent crimes, the variable identifies the most prominent crime type in each community area, which helps address distinct crime problems and offers insights into where law enforcement might be more or less effective. For example, by understanding the predominant type of crime in each community area, we can identify patterns across different areas that share the same primary crime type. This approach allows for collective improvement of similar issues, as strategies successful in one community can offer insights to others in similar circumstances. Identifying the most prevalent types of crimes can also provide the government with clear indications of which issues require more resources, thereby informing more effective resource allocation strategies.

3. Location Description:

Location description refers to the type of environment. To investigate further geographic factors, location description adds more details on community areas, which can provide insights into the environmental context of the crime. For example, what type of crimes might be more common in certain types of locations (e.g. theft on the street), allowing us to identify spatial patterns. It can reveal important spatial patterns, offering insights into how the physical environment influences crime types in various community areas, which identify the environmental factors influencing non-violence and violence. Furthermore, location-specific crime mitigation could be examined by investigating the location types that appear to have high crime frequency.

4. Arrest:

Arrest, as a dummy variable, refers to whether a crime ends in arrest, which provides critical insights into law enforcement effectiveness, crime deterrence, and socio-demographic impacts on policing strategies. It helps identify patterns in crime resolution and informs targeted, equitable policy development across Chicago's community areas.

5. Domestic:

Domestic is also a dummy variable, representing whether the crime is domestic-related. It acknowledges the unique nature of domestic crimes, which often have different reporting, response, and resolution dynamics compared to other crime types. Analyzing domestic incidents separately can highlight the need for specialized law enforcement training, resources, and policies to address and prevent these crimes effectively within communities.

For temporal analysis, it is essential to determine whether time significantly impacts crime incidents to identify trends and patterns. Understanding the primary type of crime, arrest status, and domestic incidents is crucial for assessing law enforcement effectiveness. This analysis can inform governmental strategies to enhance crime reduction efforts in specific areas. In spatial analysis, examining community areas and location descriptions aids in pinpointing geographical hotspots and blocks, offering a detailed perspective at the area level.

1	Rogers Park	27	East Garfield Park	53	West Pullman
2	West Ridge	28	Near West Side	54	Riverdale
3	Uptown	29	North Lawndale	55	Hegewisch
4	Lincoln Square	30	South Lawndale	56	Garfield Ridge
5	North Center	31	Lower West Side	57	Archer Heights
6	Lake View	32	Loop	58	Brighton Park
7	Lincoln Park	33	Near South Side	59	McKinley Park
8	Near North Side	34	Armour Square	60	Bridgeport
9	Edison Park	35	Douglas	61	New City
10	Norwood Park	36	Oakland	62	West Elsdon
11	Jefferson Park	37	Fuller Park	63	Gage Park
12	Forest Glen	38	Grand Boulevard	64	Clearing
13	North Park	39	Kenwood	65	West Lawn
14	Albany Park	40	Washington Park	66	Chicago Lawn
15	Portage Park	41	Hyde Park	67	West Englewood
16	Irving Park	42	Woodlawn	68	Englewood
17	Dunning	43	South Shore	69	Greater Grand Crossing
18	Montclare	44	Chatham	70	Ashburn
19	Belmont Cragin	45	Avalon Park	71	Auburn Gresham
20	Hermosa	46	South Chicago	72	Beverly
21	Avondale	47	Burnside	73	Washington Heights
22	Logan Square	48	Calumet Heights	74	Mount Greenwood
23	Humboldt Park	49	Roseland	75	Morgan Park
24	West Town	50	Pullman	76	O'Hare
25	Austin	51	South Deering	77	Edgewater
26	West Garfield Park	52	East Side		

1.1.2.3 Reason of choosing these variables:

The 5 independent variables are chosen to investigate the intricate correlation between geographical areas and crime types in Chicago. 'community_area' captured geographical specificity, analyzing the areas. 'location_description' adds more detailed of environmental context, analyzing specific types of location. 'primary_type', 'arrest' as well as 'domestic' indicates law enforcement effectiveness, analyze whether crime prevention measures is successful. The combined analysis of these variables provides a holistic understanding of the multifaceted factors contributing to crime hotspots in Chicago,

hence we can generate a main view of the correlation between geographic areas and crime incidents. Therefore, it reveals the complexities of crime patterns, offering actionable insights for targeted interventions and policy decisions for the local government.In this case, the safty of the community could be improved more effectively.

1.2. Data Cleaning and Summaries

1.2.1 Import data

In [1]:
```
! pip install stargazer
```

```
Requirement already satisfied: stargazer in d:\anacoda\lib\site-packages (0.0.6)
```

In [1]:
```
# Import libararies needed
import numpy as np
import pandas as pd
import seaborn as sns
import warnings
import geopandas as gpd

import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
from matplotlib.pyplot import figure
from matplotlib.patches import Patch

import requests
from bs4 import BeautifulSoup

import statsmodels.api as sm
from statsmodels.iolib.summary2 import summary_col
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor, plot_tree
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn import tree
from sklearn import metrics

from sklearn.ensemble import BaggingClassifier, RandomForestClassifier, BaggingR
from sklearn.metrics import mean_squared_error,confusion_matrix, classification_
```

In [3]:
```
# Read the original data and show the first 5 rows
chicago_crime_all = pd.read_csv('Crimes_-_2001_to_Present_20240124.csv')

chicago_crime_all.head()
```

Out[3]:

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Loca Descrip
0	5741943	HN549294	08/25/2007 09:22:18 AM	074XX N ROGERS AVE	0560	ASSAULT	SIMPLE	OT
1	25953	JE240540	05/24/2021 03:06:00 PM	020XX N LARAMIE AVE	0110	HOMICIDE	FIRST DEGREE MURDER	STF
2	26038	JE279849	06/26/2021 09:24:00 AM	062XX N MC CORMICK RD	0110	HOMICIDE	FIRST DEGREE MURDER	PARK
3	13279676	JG507211	11/09/2023 07:30:00 AM	019XX W BYRON ST	0620	BURGLARY	UNLAWFUL ENTRY	APARTM
4	13274752	JG501049	11/12/2023 07:59:00 AM	086XX S COTTAGE GROVE AVE	0454	BATTERY	AGGRAVATED P.O. - HANDS, FISTS, FEET, NO / MIN...	SM RE ST

5 rows × 22 columns

In [4]:
```
# Check how long is the data
len(chicago_crime_all)
```

Out[4]: 7983176

1.2.2 Modify data format and column names, and retrive the data we need (2014 to 2019)

In [5]:
```
# Modify the format of Date
chicago_crime_all['Date'] = pd.to_datetime(chicago_crime_all['Date'], format='%m

# Filter data for 2018 to 2023
start_date = '2014-01-01'
end_date = '2018-12-31'
chicago_crime = chicago_crime_all.loc[(chicago_crime_all['Date'] >= start_date)
chicago_crime
```

Out[5]:

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	De
29	12536164	JE439378	2015-09-24 00:00:00	031XX W 53RD PL	1753	OFFENSE INVOLVING CHILDREN	SEXUAL ASSAULT OF CHILD BY FAMILY MEMBER	AP
30	12536166	JE439332	2014-09-07 00:00:00	031XX W 53RD PL	1753	OFFENSE INVOLVING CHILDREN	SEXUAL ASSAULT OF CHILD BY FAMILY MEMBER	AP
46	13158716	JG362691	2018-11-09 00:00:00	017XX N NASHVILLE AVE	0265	CRIMINAL SEXUAL ASSAULT	AGGRAVATED - OTHER	R
56	13188119	JG397237	2015-05-15 00:00:00	041XX W 24TH PL	1754	OFFENSE INVOLVING CHILDREN	AGGRAVATED SEXUAL ASSAULT OF CHILD BY FAMILY M...	AP
119	13193781	JG397432	2015-06-18 00:00:00	031XX S KOSTNER AVE	1752	OFFENSE INVOLVING CHILDREN	AGGRAVATED CRIMINAL SEXUAL ABUSE BY FAMILY MEMBER	R
...	...	...	...	...	...	...	...	
7978576	13135737	JG335293	2018-03-15 00:00:00	079XX S HONORE ST	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300	AP
7978849	13153094	JG355091	2017-01-01 09:20:00	088XX S EUCLID AVE	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300	
7979628	13077134	JG264727	2018-09-30 00:00:00	028XX N MC VICKER AVE	0266	CRIMINAL SEXUAL ASSAULT	PREDATORY	R
7980730	13169093	JG374535	2017-09-08 10:00:00	006XX E 89TH ST	0810	THEFT	OVER $500	AP
7981524	13069800	JG256610	2016-05-01 10:00:00	062XX S LANGLEY AVE	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300	AP

1347933 rows × 22 columns

In [6]:
```
# Check the length again
len(chicago_crime)
```

Out[6]:
```
1347933
```

In [7]:
```
# Check the name of columns
chicago_crime.columns
```

Out[7]:
```
Index(['ID', 'Case Number', 'Date', 'Block', 'IUCR', 'Primary Type',
       'Description', 'Location Description', 'Arrest', 'Domestic', 'Beat',
       'District', 'Ward', 'Community Area', 'FBI Code', 'X Coordinate',
       'Y Coordinate', 'Year', 'Updated On', 'Latitude', 'Longitude',
       'Location'],
      dtype='object')
```

In [8]:
```
# Convert all the names to lowercase, and replace space by _:
chicago_crime.columns = chicago_crime.columns.str.lower().str.replace(' ', '_')
chicago_crime.columns
```

Out[8]:
```
Index(['id', 'case_number', 'date', 'block', 'iucr', 'primary_type',
       'description', 'location_description', 'arrest', 'domestic', 'beat',
       'district', 'ward', 'community_area', 'fbi_code', 'x_coordinate',
       'y_coordinate', 'year', 'updated_on', 'latitude', 'longitude',
       'location'],
      dtype='object')
```

Now, we have all the column names in the same format, and date is converted to the format eligible for process.

1.2.3 Get summaries of the data

To avoid community_area be treated as a numerical vairable, we first convert it to category

In [9]:
```
chicago_crime['community_area'] = chicago_crime['community_area'].astype('catego
```

Make "True"=1 and "False"=0 for category variables arrest and domestic before summarize the dataset.

In [10]:
```
chicago_crime['arrest_boolean'] = chicago_crime['arrest'].apply(lambda x: 1 if x
chicago_crime['domestic_boolean'] = chicago_crime['domestic'].apply(lambda x: 1

chicago_crime['arrest_boolean'] = chicago_crime['arrest_boolean'].astype('catego
chicago_crime['domestic_boolean'] = chicago_crime['domestic_boolean'].astype('ca
```

In [11]:
```
# Retrive the selected variables and get the summary of the data
selected_columns = ['community_area','primary_type', 'location_description', 'ar

# Generating summary statistics
summary_statistics = chicago_crime[selected_columns].describe(include='all')

# Displaying the summary statistics
summary_statistics
```

Out[11]:

	community_area	primary_type	location_description	arrest_boolean	domestic_b
count	1347933.0	1347933	1343224	1347933	1
unique	77.0	34	183	2	
top	25.0	THEFT	STREET	0	
freq	83688.0	310055	303690	1038774	1

1.2.4 Drop null values

The length of location_description is less than the length of data, which means there are null values. The drop of null values in the variables is undertaken in this study. It is noteworthy that null values exclusively represent a negligible proportion within the variable, as opposed to the entirety of the dataset. Rather than substituting null values with alternative data, direct removal is preferred. This approach is chosen due to the marginal impact null values in the variable are anticipated to have on the overall dataset, and it is not expected to significantly compromise the integrity of our final conclusions.

In [12]:
```
chicago_crime.dropna(subset=['location_description'], inplace=True)
```

1.3 Summary Statistics Tables

Check the summary of selected data again:

In [13]:
```
summary_statistics = chicago_crime[selected_columns].describe(include='all')

summary_statistics
```

Out[13]:

	community_area	primary_type	location_description	arrest_boolean	domestic_b
count	1343224.0	1343224	1343224	1343224	1
unique	77.0	34	183	2	
top	25.0	THEFT	STREET	0	
freq	83456.0	309941	303690	1034077	1

Count indicates the number of non-null values for each variable.
Unique represents the number of unique values for each variable.
The top indicates the most frequently occurring value for each variable.
Freq represents the frequency (count) of the most frequently occurring value.

As our variables are not numerical, measures like 'mean' and 'sd' are meaningless.
Therefore, these measurements are not shown in the summary table.

According to the table, we have a basic view of our dataset:

Community Area: Since we dropped null values, the length now is the same as the dataset. There are 77 unique values, corresponding to the 77 community areas in Chicago. Area 25 is the most frequently reported location for crimes, which requires further exploration of the specific socio-economic, environmental, or law enforcement-related factors. Other areas with higher crime rates also require further investigation of factors contributing to the crime incidents. For researchers and policymakers, this highlights a need to delve into why this area stands out and what interventions could mitigate these challenges.

Primary Type: Because of the same length as the dataset, the 'primary_type' column presents no null values. There are 34 unique values, corresponding to 34 crime types, with 'THEFT' emerging as the most prevalent. A closer examination into the factors contributing to the frequent occurrence of theft is required, informing law enforcement strategies tailored to different crime types. Understanding the intrinsic causes of the high occurrence of crime types could inform preventive measures, hence increasing community safety.

Location Description: The elimination of null values in the 'Location Description' column results in a dataset with no missing values. There are 191 unique values, corresponding to location types. "STREET" has the highest frequency, which suggests public spaces are significant hotspots for criminal activity. The potential reason could be the ease of access for criminals and escape routes, which promotes the law enforcement and potential environmental design of Chicago.

Arrest: The 2 unique values suggest that it is a binary variable with True and False only, which were substituted by 1 and 0 for the summary table. The dominance of "0"(that is, False) indicates that a large number of reported crimes do not result in an arrest, which reflects challenges in law enforcement's ability to catch perpetrators. This situation pushes for strategies to enhance investigative resources or reallocate police sources, etc.

Domestic: Similarly, it is a binary variable with True and False substituted by 1 and 0. A higher frequency of "0"(that is, False) indicates that the majority of crimes are not domestic-related. These non-domestic crimes underscore the importance of focusing safety measures within homes, while it may promote a potential enhancement of law enforcement as well as international support.

In summary, a comprehensive overview of the dataset reveals key patterns in terms of community areas, primary crime types, location descriptions, arrests and domestic. Subsequent investigations should delve into the impact of each variable on the crime incidents, specifically, the factors contributing to the areas with high crime frequency (e.g. area 25 and streets) and potential interventions to mitigate such occurrences.

1.3.1 Create new dataframes counting crime

1.3.1.1 Count crime for commnity area

In [14]:
```
crime_by_community = chicago_crime['community_area'].value_counts().reset_index(
crime_by_community.columns = ['community_area', 'crime_count_by_community']
```

```
# Display the new DataFrame
crime_by_community
```

Out[14]:

	community_area	crime_count_by_community
0	25.0	83456
1	8.0	53792
2	32.0	45266
3	29.0	44441
4	43.0	43893
...	...	...
72	55.0	3024
73	18.0	2995
74	12.0	2518
75	47.0	1952
76	9.0	1331

77 rows × 2 columns

1.3.1.2 Count crime for primary type grouby community area

In [15]:
```
gb_crime_type = chicago_crime.groupby(['community_area', 'primary_type'], observ
gb_crime_type
```

Out[15]:

	community_area	primary_type	crime_count
0	1.0	ARSON	11
1	1.0	ASSAULT	1289
2	1.0	BATTERY	3665
3	1.0	BURGLARY	888
4	1.0	CRIM SEXUAL ASSAULT	136
...	...	...	...
2224	77.0	ROBBERY	376
2225	77.0	SEX OFFENSE	73
2226	77.0	STALKING	4
2227	77.0	THEFT	3835
2228	77.0	WEAPONS VIOLATION	62

2229 rows × 3 columns

1.3.1.3 Count crime for location description grouby community area

In [16]:
```
gb_crime_loc = chicago_crime.groupby(['community_area', 'location_description'],
gb_crime_loc
```

Out[16]:

	community_area	location_description	crime_count
0	1.0	AIRPORT TERMINAL UPPER LEVEL - NON-SECURE AREA	1
1	1.0	ALLEY	545
2	1.0	ANIMAL HOSPITAL	3
3	1.0	APARTMENT	4449
4	1.0	APPLIANCE STORE	8
...	...	...	...
6275	77.0	VEHICLE - OTHER RIDE SERVICE	1
6276	77.0	VEHICLE - OTHER RIDE SHARE SERVICE (E.G., UBER...	2
6277	77.0	VEHICLE NON-COMMERCIAL	123
6278	77.0	VEHICLE-COMMERCIAL	10
6279	77.0	WAREHOUSE	6

6280 rows × 3 columns

1.3.1.4 Count crime for arrest grouby community area

In [17]:
```
gb_crime_arrest = chicago_crime.groupby(['community_area', 'arrest'], observed=T
gb_crime_arrest
```

Out[17]:

	community_area	arrest	crime_count
0	1.0	False	15124
1	1.0	True	3905
2	2.0	False	14350
3	2.0	True	2594
4	3.0	False	13995
...	...	...	...
149	75.0	True	2175
150	76.0	False	6573
151	76.0	True	2341
152	77.0	False	10095
153	77.0	True	2221

154 rows × 3 columns

1.3.1.2 Count crime for domestic grouby community area

In [18]:
```
gb_crime_domestic = chicago_crime.groupby(['community_area', 'domestic'], observ
gb_crime_domestic
```

Out[18]:

	community_area	domestic	crime_count
0	1.0	False	15947
1	1.0	True	3082
2	2.0	False	14254
3	2.0	True	2690
4	3.0	False	16023
...	...	...	...
149	75.0	True	2277
150	76.0	False	8398
151	76.0	True	516
152	77.0	False	10761
153	77.0	True	1555

154 rows × 3 columns

In [19]:
```
chicago_crime['date'] = pd.to_datetime(chicago_crime['date']).dt.date

# Step 2 & 3: Group by 'date' and 'community_area', then count crimes
gb_crime_date = chicago_crime.groupby(['date', 'community_area'], observed=True)

gb_crime_date
```

Out[19]:

	date	community_area	crime_count
0	2014-01-01	1.0	17
1	2014-01-01	2.0	11
2	2014-01-01	3.0	13
3	2014-01-01	4.0	9
4	2014-01-01	5.0	6
...	...	...	...
134680	2018-12-31	32.0	1
134681	2018-12-31	35.0	1
134682	2018-12-31	44.0	1
134683	2018-12-31	46.0	1
134684	2018-12-31	69.0	1

134685 rows × 3 columns

1.4. Plots, Histograms, Figures

1.4.1 Plot the histogram of community area and crime count

In [20]:
```
crime_by_community_sorted = crime_by_community.sort_values('crime_count_by_commu

# Plotting
plt.figure(figsize=(14, 8))
colors = 'skyblue'
plt.bar(crime_by_community_sorted['community_area'], crime_by_community_sorted['

plt.ylabel('Number of Crimes')
plt.xlabel('Community Area')
plt.title('Figure 1.1.Number of Crimes by Community Area')

plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

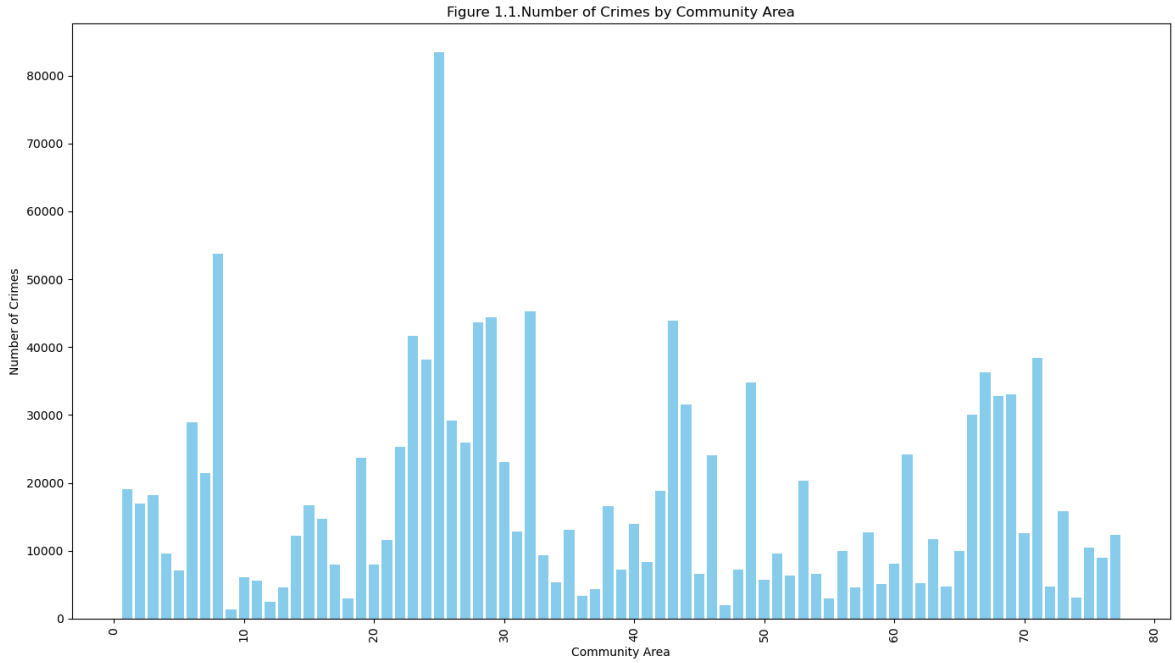


Figure 1., the bar chart, provides a visual exposition of the frequency of crime incidents across different community areas, which highlights the top 2 community areas (25 and 8) with the most frequent crime occurrence by the green bars. Area 25 has nearly reached 80000 crimes, while area 8 exceeds 60000, highlighting the prevalence of crime in these regions. Area 25, Austin, is a large city with significant large population size, which leads to a complex social environment and demographic complexity. Area 8, Near North Side, generates high income per capita, which may result into a high crime frequency of crimes like theft. As a result, we will be focusing on these two areas in our research to better understand the geographical variation of crime incidents. The graphical representation illustrates a pronounced heterogeneity in crime prevalence among the community areas, with certain districts exhibiting higher crime rates, which promotes deeper socio-economic investigation. The variable of neighbourhood area is essential in illustrating the spatial distribution of criminal activities in Chicago, which provides both a statistical overview of crime concentrations across communities, as well as the subsequent qualitative assessments of public safety strategies. We will further explore the economic reason for these abnormal areas and hence find some crime patterns for mitigate the high crime frequency. However, there could be population effect across different geographics, hence we will add population as a vairable to get rid of its effect and improve the accurancy of our research.

1.4.2 Plot the histogram of primary type grouby community area and crime count

In [21]:
```
total_crime_counts = gb_crime_type.groupby('primary_type')['crime_count'].sum().

# Sorting values by total crime count to identify top 3 crimes
total_crime_counts_sorted = total_crime_counts.sort_values(by='crime_count', asc

colors =  'skyblue'

# Plotting
plt.figure(figsize=(12, 8))
```

```
plt.bar(total_crime_counts_sorted['primary_type'], total_crime_counts_sorted['cr
plt.xlabel('Crime Type')
plt.ylabel('Total Crime Count')
plt.xticks(rotation=90, ha="right")
plt.title('Figure 1.2. Total Crime Count by Type')
plt.tight_layout()
plt.show()
```

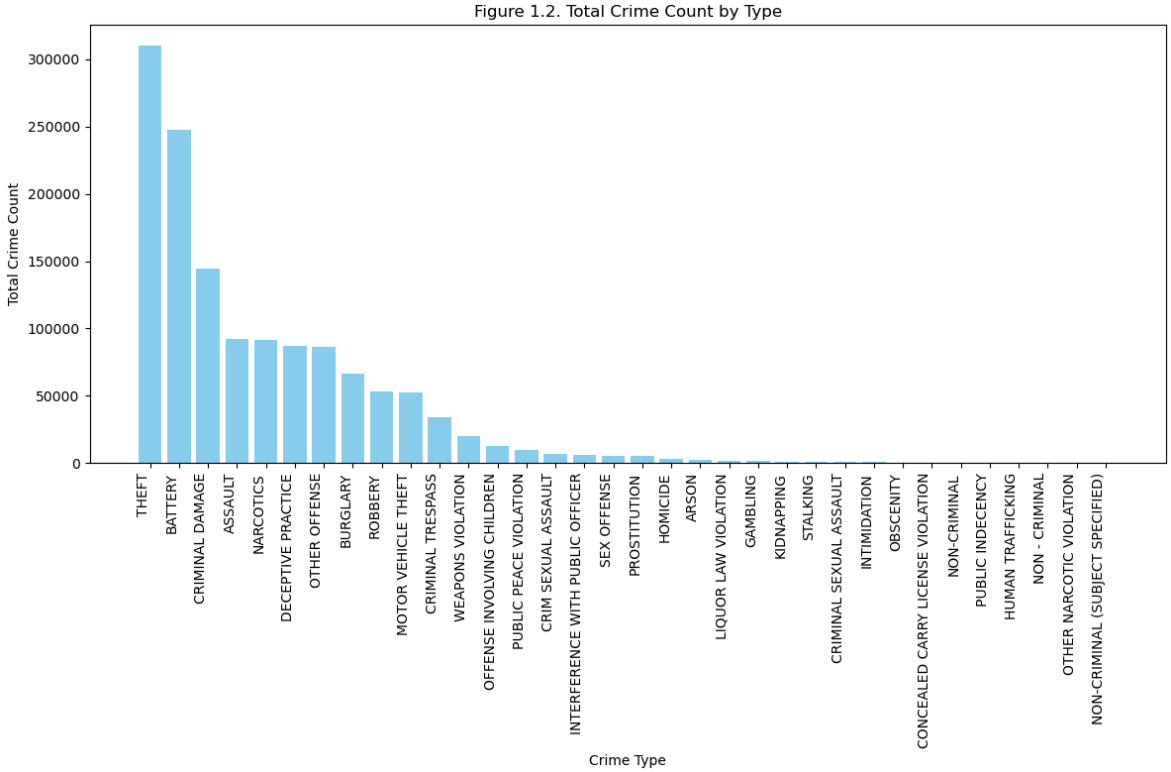


Figure 2., the bar chart, visualizes the differential prevalence of crime types. Theft is the primary crime type across all the community areas, which suggests a trend towards property-related offences within the area or law enforcement. Battery and criminal damage ranked after the theft, indicating the disparity in crime occurrences. The less frequent but grave crimes such as homicide represented by shorter bars show its rarity. The chart shows key points for understanding city crime, highlighting where actions of crime-solving work effectively. It uses numbers to help decide where to focus police work and community actions to make certain areas safer, based on the crime trends it identifies. Theft and battery are two disctinct crime types, which we will use theft as the represent of non-violent crime and battery as the violent crime incidents to further explore our research. There are other crimes in violence or non-violence, while we focus on these two based on Chicago's specific situation.

1.4.2.1 Plot the histogram of primary type grouby community area and crime count for area 25 and 8

In [22]:
```
ca_25_agg = gb_crime_type[gb_crime_type['community_area'] == 25].groupby('primar
ca_8_agg = gb_crime_type[gb_crime_type['community_area'] == 8].groupby('primary_

# Merge to ensure both community areas have the same primary_types for compariso
merged = ca_25_agg.merge(ca_8_agg, on='primary_type', how='outer', suffixes=('_2

ind = np.arange(len(merged))
```

```
width = 0.35

fig, ax = plt.subplots(figsize=(14, 8))
bars1 = ax.bar(ind - width/2, merged['crime_count_25'], width, label='Community
bars2 = ax.bar(ind + width/2, merged['crime_count_8'], width, label='Community A

ax.set_xlabel('Primary Type')
ax.set_ylabel('Crime Count')
ax.set_title('Figure 1.3.Crime Counts by Primary Type for Community Areas 25 and
ax.set_xticks(ind)
ax.set_xticklabels(merged['primary_type'], rotation=90)
ax.legend()

plt.show()
```

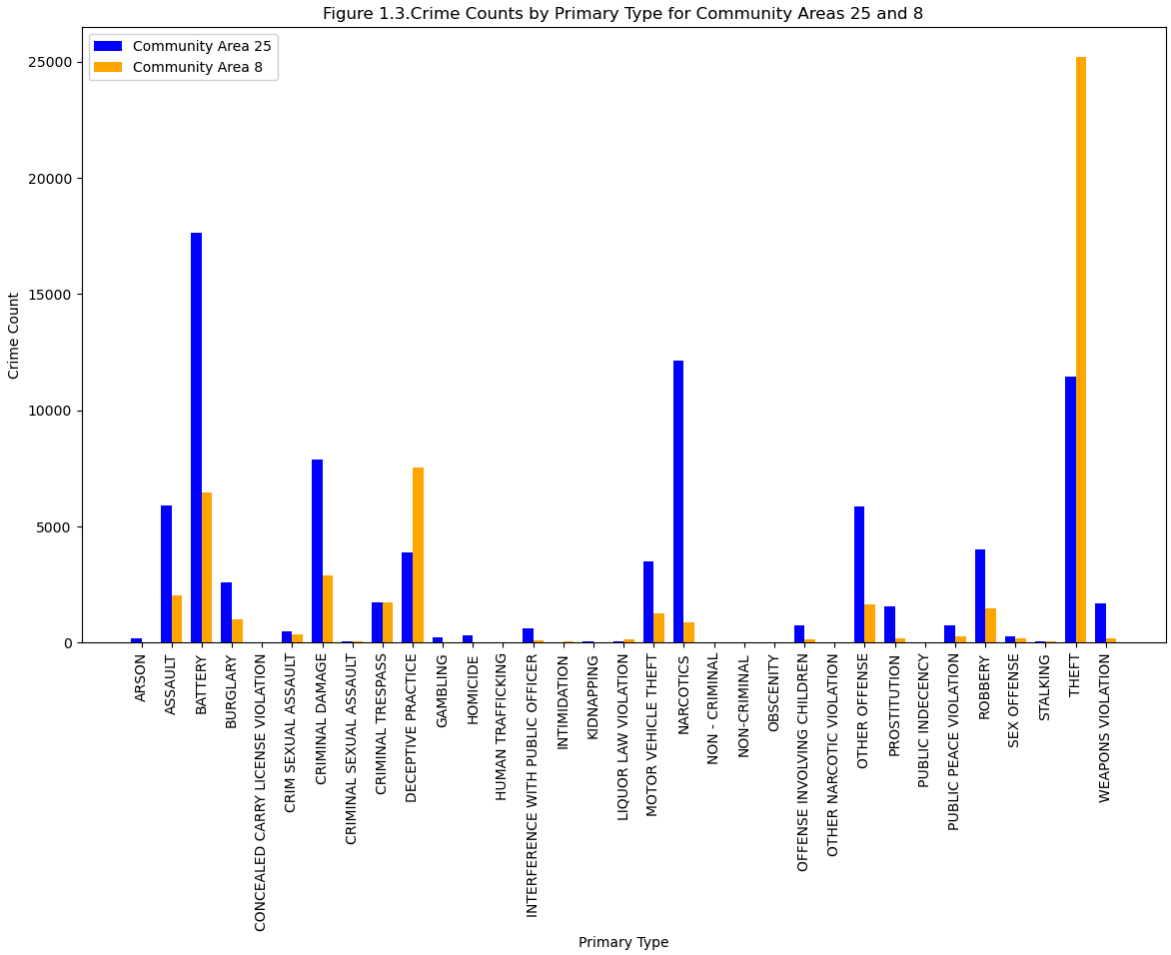


Figure 3. compares crime types in the top two areas with the highest crime incidents, exploring the crime type variation between them. Area 25 has more types of crime and a higher crime frequency, with battery occupying the dominant position. Most crimes in Area 8 are theft, with a small percentage of other crime types. This shows the differentiation of crime patterns across various community areas, reinforcing the need for area-specific crime prevention and resource allocation strategies. The data presented could potentially guide targeted policing efforts, community support initiatives, and further sociological research into the factors underpinning such disparate criminal landscapes. Again, there could be population effect that community 25 and 8 has more cimes because there are larger population, hence we cannot make conclusion here that these two areas have the highest crime rate. However, our research is focusing on the

difference between the theft and battery incidents, which make it less important to focus on the per capita basis.

1.4.3 Plot the histogram of location descrpition grouby community area and crime count

In [23]:
```
crime_counts_by_location = gb_crime_loc.groupby('location_description')['crime_c

# Sort by crime_count in descending order to identify the most common locations
crime_counts_by_location_sorted = crime_counts_by_location.sort_values('crime_co

# Focus on the top N locations for a clearer plot
top_n = 20
crime_counts_by_location_top_n = crime_counts_by_location_sorted.head(top_n)

colors = 'skyblue'

# Plotting
plt.figure(figsize=(12, 8))
bars = plt.bar(crime_counts_by_location_top_n['location_description'], crime_cou
plt.ylabel('Crime Count')
plt.xlabel('Location Description')
plt.title(f'Figure 1.4.Top {top_n} Crime Locations Across All Community Areas')
plt.xticks(rotation=90)  # Rotate the x-axis labels for better readability

plt.tight_layout()
plt.show()
```

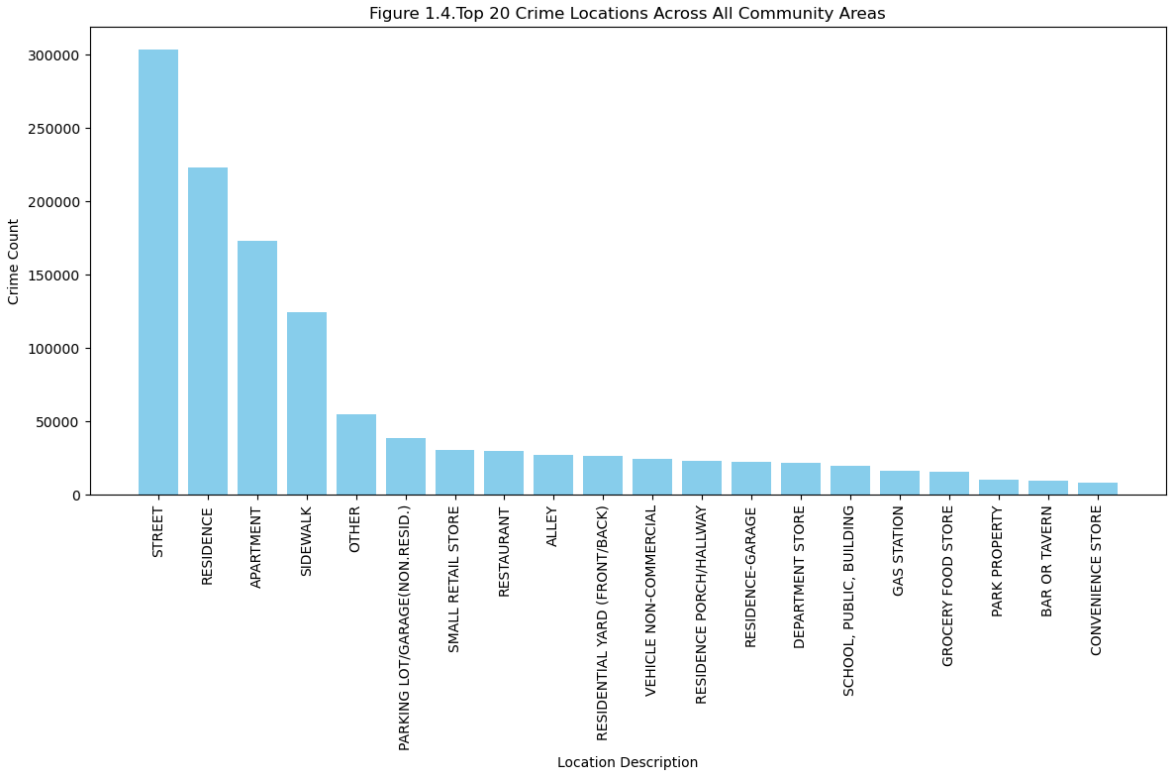


Figure 4. shows the locations that have the most crime, which adds more geographical details on the community area. The "STREET" is the location with the highest crime rate, compared to other locations like "APARTMENT" and "RESIDENCE". This suggests that crimes happen more often in open, public areas, probably because there are more people and it's easier for criminals to escape. Locations like "DEPARTMENT STORE" and

"GAS STATION" see fewer crimes, which might be because they're more secure or crimes there are not reported as much. This information highlights the need to focus on preventing crime in public spaces and suggests that community efforts to make these places safer are important. It also helps the police decide where to concentrate their resources and how to plan their strategies to deal with crime more effectively in places where it happens most.

1.4.3.1 Plot the histogram of location description grouby community area and crime count for area 25 and 8

In [24]:
```
warnings.filterwarnings('ignore', category=FutureWarning)

filtered_df = gb_crime_loc[gb_crime_loc['community_area'].isin([25, 8])]

# Aggregate to find the top 20 locations by total crime count across both areas
top_locations = filtered_df.groupby('location_description')['crime_count'].sum()

# Filter again for only the top 20 locations
top_filtered_df = filtered_df[filtered_df['location_description'].isin(top_locat

# Pivot the data for plotting
pivot_table = top_filtered_df.pivot_table(index='location_description', columns=

# Sort the pivot table to match the order of top locations
pivot_table = pivot_table.set_index('location_description').reindex(top_location

ind = np.arange(len(pivot_table))
width = 0.35

fig, ax = plt.subplots(figsize=(14, 10))

bars1 = ax.bar(ind - width/2, pivot_table[25], width, label='Community Area 25',
bars2 = ax.bar(ind + width/2, pivot_table[8], width, label='Community Area 8', c

ax.set_xlabel('Location Description')
ax.set_ylabel('Crime Count')
ax.set_title('Figure 1.5.Top 20 Crime Locations by Location Description for Comm
ax.set_xticks(ind)
ax.set_xticklabels(pivot_table['location_description'], rotation=90)
ax.legend()

fig.tight_layout()

plt.show()
```

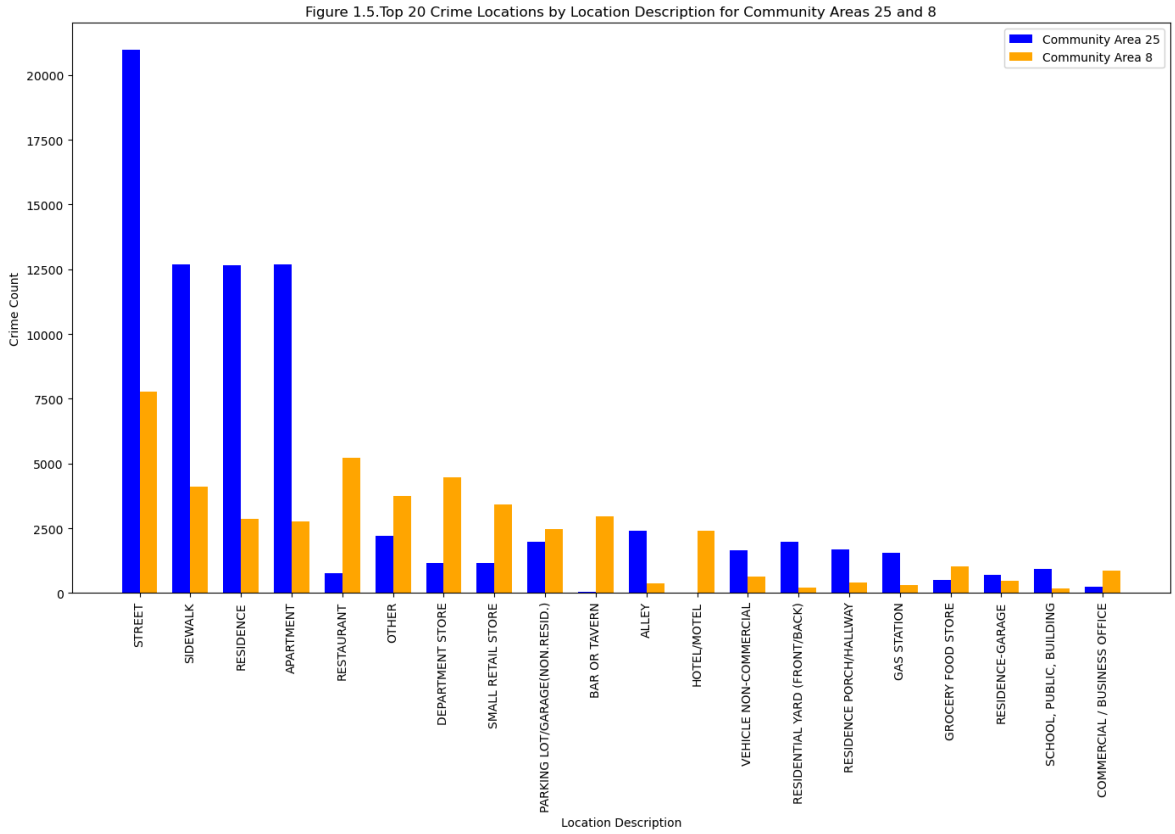
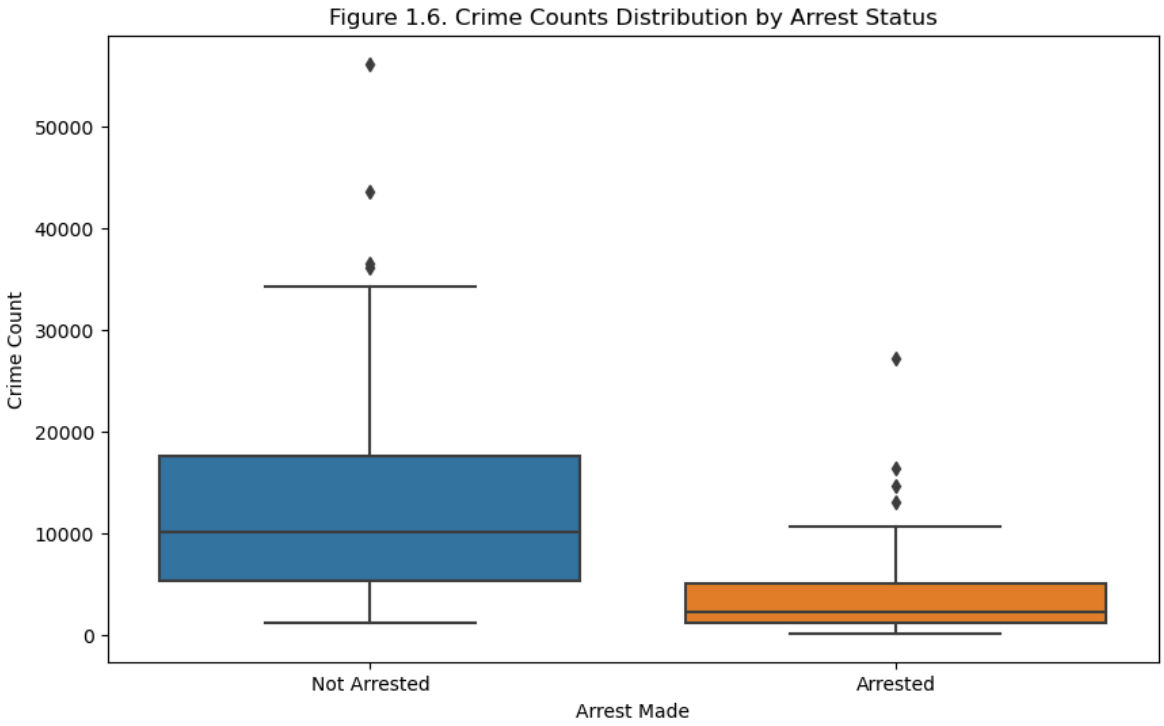


Figure 5. compares the locations where crimes happen in two areas with the most frequent crime incidents. They have common trends of high crime rates in locations, like streets and apartments, in both areas. However, Area 25 has more crimes, especially on the streets, which suggests that public spaces there might be particularly prone to crime. Area 8 has fewer crimes but follows a similar pattern in terms of where crimes happen. The chart uses different colours to make it easy to see the differences between the two areas, pointing out that we need to use different strategies to prevent crime based on where it happens. This information can help police and community groups decide where to focus their efforts, showing how important the location is when trying to reduce crime and the positive impact that targeted actions can have.

1.4.4 Plot the histogram of arrest grouby community area and crime count

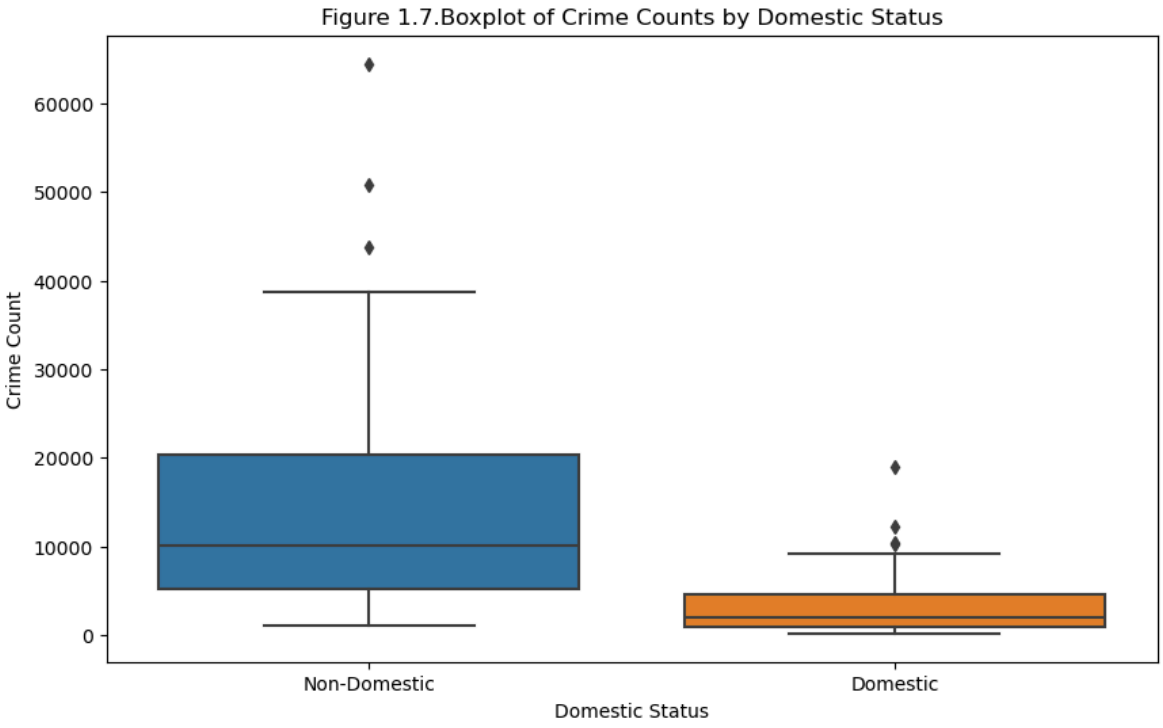
In [25]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='arrest', y='crime_count', data=gb_crime_arrest)

plt.title('Figure 1.6. Crime Counts Distribution by Arrest Status')
plt.xlabel('Arrest Made')
plt.ylabel('Crime Count')
plt.xticks([0, 1], ['Not Arrested', 'Arrested'])

plt.show()
```



The boxplot Figure 6. compares crimes that ended in arrests to those that did not, promoting insight into the effectiveness of law enforcement. Arrest rate is a potential effect of high crime frequency, hence we are interested in it and want to find the patterns and the correlations with crime rate. The "Not Arrested" group has a higher median number of crimes and shows more variation in the crime frequency, which means a large percentage of crimes don't lead to arrests. The reason could be the challenges in catching criminals or because many crimes are not serious enough to result in arrests. On the other side, the "Arrested" group has a lower median and less variation, showing that crimes leading to arrests happen less often and more consistently. There are outliers in both groups, with not arrested having larger outliers, which might showcase with a lot of crime or times when law enforcement was particularly successful. This information is important for understanding what affects arrest rates and how police resources can be used better to improve public safety and the effectiveness of the criminal justice system.

1.4.5 Plot the histogram of domestic grouby community area and crime count

In [26]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(x='domestic', y='crime_count', data=gb_crime_domestic)

plt.title('Figure 1.7.Boxplot of Crime Counts by Domestic Status')
plt.xlabel('Domestic Status')
plt.ylabel('Crime Count')
plt.xticks([0, 1], ['Non-Domestic', 'Domestic'])

plt.show()
```



The boxplot Figure 7. illustrates the crime counts split by non-domestic and domestic, which highlights a clear difference in the frequency of crimes occurring. Crimes that are not related to domestic issues have a wider range of occurrences and a higher median, indicating the higher frequency of crimes outside the home, shown by a larger interquartile range and median value. In contrast, domestic crimes, though occurring less often, have a more concentrated number of incidents around a lower median. It indicates that while these incidents are less frequent, they are closely packed together in terms of occurrence, which doesn't imply they are less serious. There are outliers in both categories, which represent exceptionally high crime counts. These insights are vital for understanding the different dynamics of crime within and outside domestic environments, and they can help in creating focused strategies and policies to more effectively combat and reduce these crimes.

2. Project 2

2.1. The Main Message

This research explores the correlation between economic disparity and criminal nature (non-violence and violence), specifically theft and battery, across Chicago's community areas. Findings indicate that higher-income areas report more theft, while areas with higher unemployment rates experience more battery crimes, with the latter facing higher arrest rates.

2.1.1 Add Per Capita Income Data

Retrive the PCI data source from Chicago Health Atlas, which will be merged and proceed in the later part. It is only used for the main message plot here, and the

explanation is shown in 2.3.2.

Source: https://chicagohealthatlas.org/indicators/PCI?topic=per-capita-income

In [27]:
```
chicago_health = pd.read_csv('Chicago Health Atlas Data Download - Community are
chicago_health.columns = chicago_health.columns.str.lower()
chicago_health.head()
```

Out[27]:

	layer	name	geoid	ump	pci	edb
0	Community area	Rogers Park	1	6.702234	29865.85132	86.626179
1	Community area	Norwood Park	10	3.480262	45905.69890	92.690008
2	Community area	Jefferson Park	11	4.798231	38012.99518	89.911429
3	Community area	Forest Glen	12	3.836323	55414.64903	94.304692
4	Community area	North Park	13	5.660225	37313.52802	89.166067

At current stage, we use theft counts divided by the total crime counts since we are focusing on the difference between the crime types and so does battery rate, rather than using population. The research is aim to explore the different patterns in different crime types, hence the comparison of crime incidents is more suitable than per capita based crime rate. Even though the community areas with larger population may report higher crime per capita, different crime types are facing the same effect. To be simpler, the areas may have more people will have both higher theft and battery, which offset the population effect in the comparison. However, to get rid of the population effect for higher crime rates in community areas with larger population, we add it in the regression part to control it.

In [28]:
```
# Count theft rate
theft_counts = chicago_crime[chicago_crime['primary_type'] == 'THEFT'].groupby('
total_counts = chicago_crime.groupby('community_area').size()
theft_rate = (theft_counts / total_counts).fillna(0).reset_index(name='theft_rat

theft_rate
```

Out[28]:

	community_area	theft_rate
0	1.0	0.256503
1	2.0	0.233180
2	3.0	0.265578
3	4.0	0.283788
4	5.0	0.369161
...	...	...
72	73.0	0.186677
73	74.0	0.265881
74	75.0	0.232895
75	76.0	0.293134
76	77.0	0.311384

77 rows × 2 columns

In [29]:
```
merged_theft = pd.merge(theft_rate, chicago_health, left_on='community_area', ri
merged_theft
```

Out[29]:

	community_area	theft_rate	layer	name	geoid	ump	pci
0	1.0	0.256503	Community area	Rogers Park	1	6.702234	29865.85132
1	2.0	0.233180	Community area	West Ridge	2	7.747411	29013.96665
2	3.0	0.265578	Community area	Uptown	3	4.543366	49199.37059
3	4.0	0.283788	Community area	Lincoln Square	4	4.226640	50564.15055
4	5.0	0.369161	Community area	North Center	5	3.458962	76270.63838
...	...	...	...	...	...	...	...
72	73.0	0.186677	Community area	Washington Heights	73	17.879829	25715.74717
73	74.0	0.265881	Community area	Mount Greenwood	74	3.826828	42097.66764
74	75.0	0.232895	Community area	Morgan Park	75	12.109424	34466.26312
75	76.0	0.293134	Community area	O'Hare	76	7.787595	30096.72359
76	77.0	0.311384	Community area	Edgewater	77	5.431179	48396.59969

77 rows × 8 columns

In [30]:
```
# Count battery rate
battery_counts = chicago_crime[chicago_crime['primary_type'] == 'BATTERY'].group
battery_rate = (battery_counts / total_counts).reset_index(name='battery_rate')
battery_rate.columns = ['community_area', 'battery_rate']

battery_rate
```

Out[30]:

	community_area	battery_rate
0	1.0	0.192601
1	2.0	0.171565
2	3.0	0.183798
3	4.0	0.159193
4	5.0	0.091864
...	...	...
72	73.0	0.187244
73	74.0	0.147020
74	75.0	0.182104
75	76.0	0.101526
76	77.0	0.168480

77 rows × 2 columns

In [31]:
```
merged_battery = pd.merge(battery_rate, chicago_health, left_on='community_area'
merged_battery
```

Out[31]:

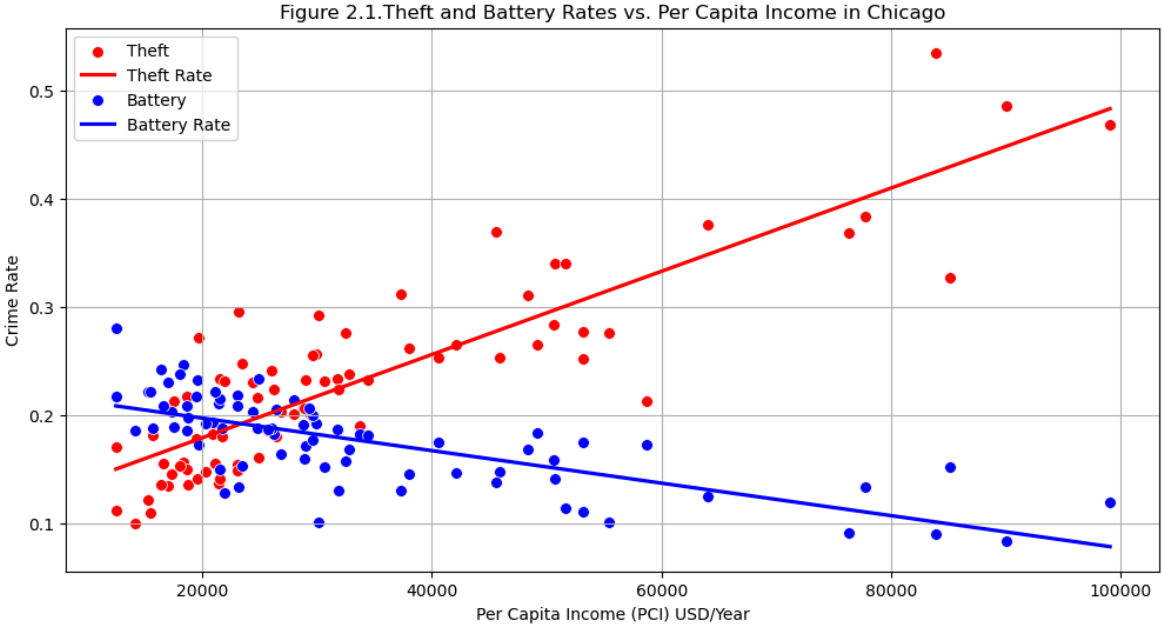
	community_area	battery_rate	layer	name	geoid	ump	p
0	1.0	0.192601	Community area	Rogers Park	1	6.702234	29865.851
1	2.0	0.171565	Community area	West Ridge	2	7.747411	29013.966
2	3.0	0.183798	Community area	Uptown	3	4.543366	49199.370
3	4.0	0.159193	Community area	Lincoln Square	4	4.226640	50564.150
4	5.0	0.091864	Community area	North Center	5	3.458962	76270.638
...	...	...	...	...	...	...	
72	73.0	0.187244	Community area	Washington Heights	73	17.879829	25715.747
73	74.0	0.147020	Community area	Mount Greenwood	74	3.826828	42097.667
74	75.0	0.182104	Community area	Morgan Park	75	12.109424	34466.263
75	76.0	0.101526	Community area	O'Hare	76	7.787595	30096.723
76	77.0	0.168480	Community area	Edgewater	77	5.431179	48396.599

77 rows × 8 columns

2.1.2 Plot Main Message Graph

In [32]:
```
plt.figure(figsize=(12, 6))
sns.scatterplot(data=merged_theft, x='pci', y='theft_rate', s=50, color='red', l
sns.regplot(data=merged_theft, x='pci', y='theft_rate', scatter=False, color='re
sns.scatterplot(data=merged_battery, x='pci', y='battery_rate', s=50, color='blu
sns.regplot(data=merged_battery, x='pci', y='battery_rate', scatter=False, color

plt.title('Figure 2.1.Theft and Battery Rates vs. Per Capita Income in Chicago')
plt.xlabel('Per Capita Income (PCI) USD/Year')
plt.ylabel('Crime Rate')
plt.legend()
plt.grid(True)
plt.show()
```

Figure 2.1.Theft and Battery Rates vs. Per Capita Income in Chicago



The figure2.1 titled "Theft and Battery Rates vs. Per Capita Income in Chicago" illustrates the relationship between per capita income and theft and battery rates in Chicago. The horizontal axis represents the annual per capita income (PCI) in US dollars; the vertical axis represents the theft or the battery rate.

The theft rate and the battery rate are calculated as the counts divided by the total crime number.

Each red dot on the scatter plot represents the theft rate of each community area in Chicago. The graph also includes a red trend line that shows the relationship between the theft rate and per capita income through the set of data points. The slope of the trend line shows a positive relationship, which means community areas with higher per capita intend to report higher theft rates.

Each blue dot on the scatter plot represents the battery rate of each community area in Chicago. The graph also includes a blue trend line that shows the relationship between battery rate and per capita income. On the opposite of the theft rate, the slope of the trend line is negative, which indicates that community areas with lower income levels appear higher battery rates.

It can be observed from the chart that most of the theft and battery in areas with low per capita income (around 20,000 USD) are concentrated in the lower range, while the rates in community areas with high per capita income (more than 60,000 USD) are relatively higher. However, the distribution of data points in the scatterplot shows some dispersion, suggesting that there may be other factors besides per capita income that influence theft and battery rates.

The distinct patterns represented by theft and battery rates across community areas result from the essential of the two crimes. Theft incidents are considered non-violent while battery crimes usually get involved in violent activities. Hence the wealthier community areas would be attractive for theft criminals because of the tangible assets,

whereas the areas considered poverty may drive the social emotion leading to violent phenomenon.

The rationale of the relationship might be as follows: Wealthier neighborhoods often possess more valuable items, making them prime targets for theft, whereas residents of less affluent areas tend to have fewer possessions, thus diminishing the potential rewards for theft. Individuals from lower socioeconomic backgrounds are more prone to experiencing social and economic pressures. Such stress can lead individuals to express their dissatisfaction through violent behavior or seek to alter their circumstances via violent methods. Additionally, in some economically disadvantaged communities, the scarcity of educational resources can hinder residents' ability to pursue nonviolent solutions to their problems.

The information provided by this scatter is essential for studying the relationship between economic conditions and different types of crimes. It can help policymakers and researchers explore the safety status of communities at different economic levels and consider how to factor in community economic status in resource allocation and prevention strategies. However, it is also important to note that correlation does not mean causation. This relationship may also be affected by other variables not shown in the figure, which will be analyzed in details later.

At current stage, income level is the only economic indicator, while it only reprents the dynamic economic background of the redsidents. Income level shows the poverty and richness across the community areas, whereas the promoter push people to commit crimes could be various. People with low income by living in an area with low living cost may not choose to steall. However, people with middle income level living in the areas where cannot afford the rental fees are likely to steal.

2.2 Maps and Interpretations

In [33]:
```
community_areas_gdf = gpd.read_file("https://data.cityofchicago.org/api/geospati
community_areas_gdf.head()
```

Out[33]:

	area	area_num_1	area_numbe	comarea	comarea_id	community	perimeter	sha
0	0.0	35	35	0.0	0.0	DOUGLAS	0.0	4.600
1	0.0	36	36	0.0	0.0	OAKLAND	0.0	1.691
2	0.0	37	37	0.0	0.0	FULLER PARK	0.0	1.991
3	0.0	38	38	0.0	0.0	GRAND BOULEVARD	0.0	4.849
4	0.0	39	39	0.0	0.0	KENWOOD	0.0	2.907

In [34]:
```
community_areas_gdf.rename(columns={'area_numbe': 'community_area'}, inplace=Tru
```

In [35]:
```
community_areas_gdf
```

Out[35]:

	area	area_num_1	community_area	comarea	comarea_id	community	perimeter
0	0.0	35	35	0.0	0.0	DOUGLAS	0.0
1	0.0	36	36	0.0	0.0	OAKLAND	0.0
2	0.0	37	37	0.0	0.0	FULLER PARK	0.0
3	0.0	38	38	0.0	0.0	GRAND BOULEVARD	0.0
4	0.0	39	39	0.0	0.0	KENWOOD	0.0
...	...	...	...	...	...	...	...
72	0.0	74	74	0.0	0.0	MOUNT GREENWOOD	0.0
73	0.0	75	75	0.0	0.0	MORGAN PARK	0.0
74	0.0	76	76	0.0	0.0	OHARE	0.0
75	0.0	77	77	0.0	0.0	EDGEWATER	0.0
76	0.0	9	9	0.0	0.0	EDISON PARK	0.0

77 rows × 10 columns

In [36]:
```
crime_counts = chicago_crime.groupby(['community_area', 'primary_type']).size().

crime_counts_sorted = crime_counts.sort_values(['community_area', 'count'], asce
gb_crime_top_type = crime_counts_sorted.groupby('community_area').head(1).reset_

gb_crime_top_type.rename(columns={'primary_type': 'top_primary_type'}, inplace=T

gb_crime_top_type.head()
```

Out[36]:

	community_area	top_primary_type	count
0	1.0	THEFT	4881
1	2.0	THEFT	3951
2	3.0	THEFT	4829
3	4.0	THEFT	2715
4	5.0	THEFT	2600

In [37]:
```
print(community_areas_gdf['community_area'].unique())
print(gb_crime_top_type['community_area'].unique())
```

```
['35' '36' '37' '38' '39' '4' '40' '41' '42' '1' '11' '12' '13' '14' '15'
 '16' '17' '18' '19' '2' '20' '21' '22' '23' '24' '25' '26' '27' '28' '29'
 '3' '30' '31' '33' '34' '10' '8' '32' '43' '44' '45' '46' '47' '59' '6'
 '48' '49' '5' '50' '51' '52' '53' '54' '55' '56' '57' '58' '60' '61' '62'
 '63' '64' '65' '66' '67' '68' '69' '7' '70' '71' '72' '73' '74' '75' '76'
 '77' '9']
[1.0, 2.0, 3.0, 4.0, 5.0, ..., 73.0, 74.0, 75.0, 76.0, 77.0]
Length: 77
Categories (77, float64): [1.0, 2.0, 3.0, 4.0, ..., 74.0, 75.0, 76.0, 77.0]
```

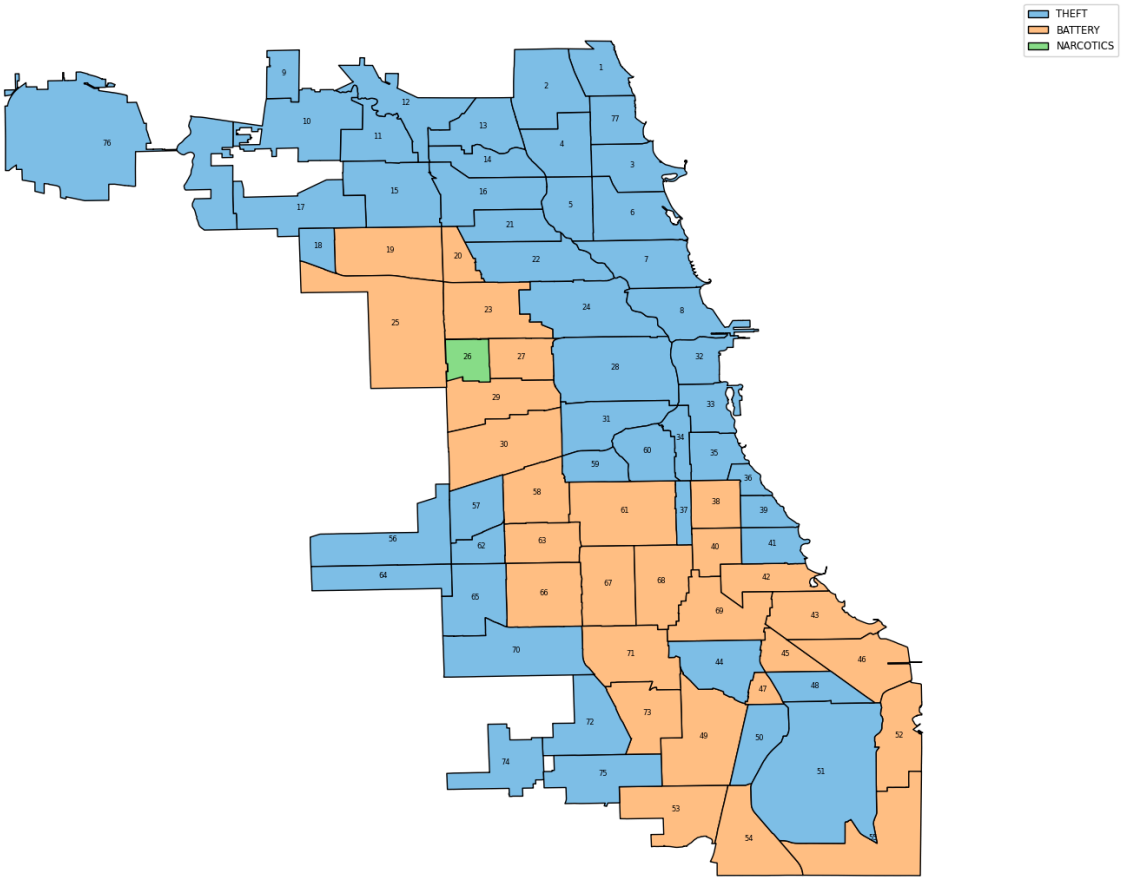
In [38]:
```
# Convert crime_by_community community_area values from float strings to integer
gb_crime_top_type['community_area'] = gb_crime_top_type['community_area'].apply(

merged_gdf = community_areas_gdf.merge(gb_crime_top_type, on='community_area')

print(merged_gdf.shape)
```

```
(77, 12)
```

In [39]:
```
# Function to blend color with white
def lighten_color(color, amount=0.5):
    """
    Lightens the given color by mixing it with white.
    Amount > 1 will lighten the color, while amount < 1 will darken it.
    """
    import colorsys
    try:
        c = mcolors.cnames[color]
    except:
        c = color
    c = colorsys.rgb_to_hls(*mcolors.to_rgb(c))
    return colorsys.hls_to_rgb(c[0], 1 - amount * (1 - c[1]), c[2])
```

```
colors = list(mcolors.TABLEAU_COLORS.keys())
unique_crime_types = gb_crime_top_type['top_primary_type'].unique()
crime_type_to_color = {crime: lighten_color(colors[i % len(colors)], amount=0.5)

fig, ax = plt.subplots(1, figsize=(15, 15))

for ctype, data in merged_gdf.groupby('top_primary_type'):
    color = crime_type_to_color[ctype]
    data.plot(color=color, ax=ax, edgecolor='black')

for idx, row in merged_gdf.iterrows():
    centroid = row.geometry.centroid
    ax.text(centroid.x, centroid.y, str(row['community_area']), ha='center', fon

ax.set_aspect('equal')
ax.axis('off')
plt.title('Figure 2.2.Top Primary Crime Type by Community Area in Chicago')

legend_elements = [Patch(facecolor=crime_type_to_color[ctype], edgecolor='black'
ax.legend(handles=legend_elements, bbox_to_anchor=(1.05, 1), loc='upper left', f

plt.show()
```

Figure 2.2.Top Primary Crime Type by Community Area in Chicago



The map "Top Primary Crime Type by Community Area in Chicago" categorizes each community area by the most prevalent type of crime, using the same colour for the same crime types. There appeared two major categories: theft in blue, battery in orange and a single narcotics in green.

The map illustrates a spatial distribution of crime types across Chicago. The theft appears to be the predominant crime type in the majority of community areas, which are mostly concentrated in the Eastern North and a few in the Western South. Battery is also common, clustering towards the Eastern center and Eastern South in Chicago. Compared to the arrest rate map, it appears a similar pattern, which means that battery got a higher arrest rate than theft. It could be blamed on the essentials of the two different types of crime. Battery, being a violent crime, often receives more immediate attention from law enforcement compared to theft, which can be non-violent. Meanwhile, battery cases often involve direct physical harm to individuals, who can then report the crime, sometimes while the offender is still on the scene.

The prevalence of certain crime types may be influenced by socioeconomic factors such as income levels, education, employment rates, and demographic profiles. For example, theft may be more prevalent in areas with high foot traffic and commercial activity, while battery could be more common in areas with nightlife or domestic disturbances.

For areas with high rates of battery, initiatives might include domestic violence resources or conflict resolution programs. While areas with high theft rates could benefit from improved lighting and surveillance. Identifying the top crime type in each area can lead to focused questions about the conditions that contribute to such patterns and how they might relate to broader socioeconomic or demographic trends observed in Chicago.

In [40]:
```
print(community_areas_gdf['community_area'].unique())
print(crime_by_community['community_area'].unique())
```

```
['35' '36' '37' '38' '39' '4' '40' '41' '42' '1' '11' '12' '13' '14' '15'
 '16' '17' '18' '19' '2' '20' '21' '22' '23' '24' '25' '26' '27' '28' '29'
 '3' '30' '31' '33' '34' '10' '8' '32' '43' '44' '45' '46' '47' '59' '6'
 '48' '49' '5' '50' '51' '52' '53' '54' '55' '56' '57' '58' '60' '61' '62'
 '63' '64' '65' '66' '67' '68' '69' '7' '70' '71' '72' '73' '74' '75' '76'
 '77' '9']
[25.0, 8.0, 32.0, 29.0, 43.0, ..., 55.0, 18.0, 12.0, 47.0, 9.0]
Length: 77
Categories (77, float64): [1.0, 2.0, 3.0, 4.0, ..., 74.0, 75.0, 76.0, 77.0]
```

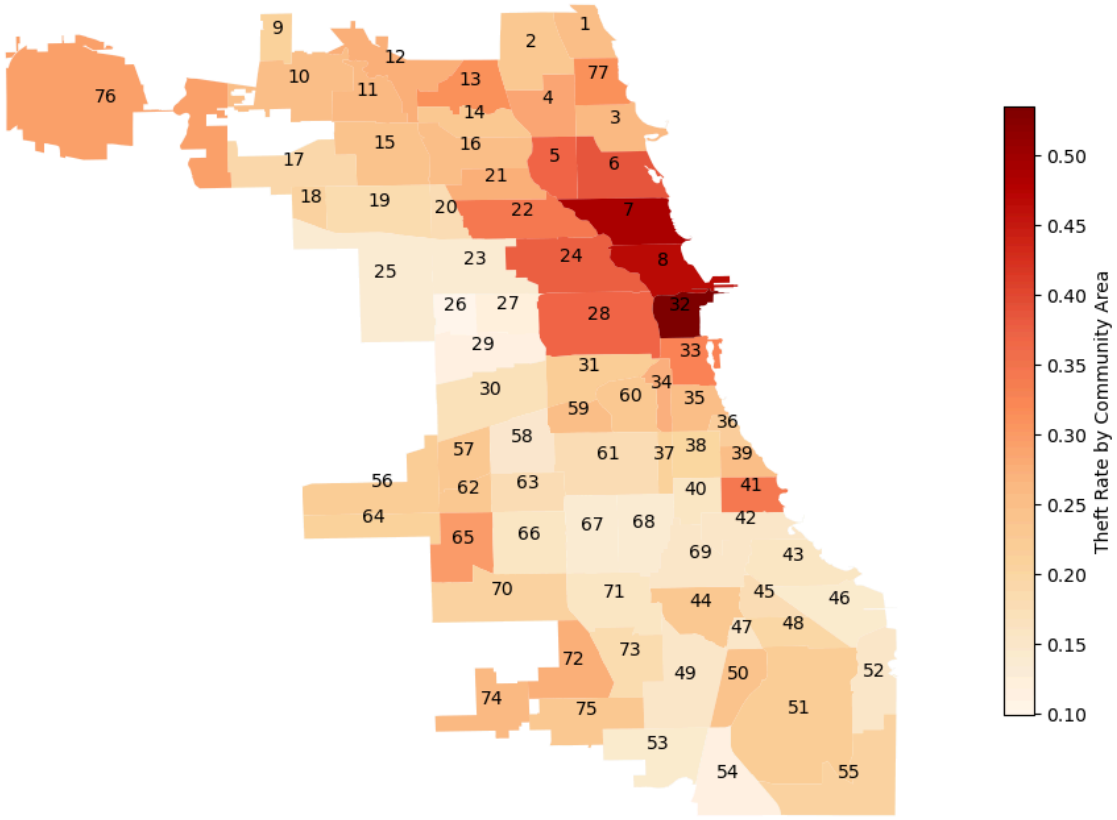
In [41]:
```
crime_by_community['community_area'] = crime_by_community['community_area'].appl

merged_gdf = community_areas_gdf.merge(crime_by_community, on='community_area')

print(merged_gdf.shape)
```

```
(77, 11)
```

In [42]:
```
community_areas_gdf['community_area'] = community_areas_gdf['community_area'].ap
merged_theft['community_area'] = merged_theft['community_area'].apply(lambda x:

# Merging the dataframes
merged_data = community_areas_gdf.merge(merged_theft, on='community_area')
# Ensure the merged_data is not empty
if not merged_data.empty:
    # Plotting
    fig, ax = plt.subplots(1, 1, figsize=(12, 12))
    merged_data.plot(column='theft_rate', ax=ax, legend=True, cmap='OrRd',
                     legend_kwds={'label': "Theft Rate by Community Area",
                                  'orientation': "vertical", 'shrink': 0.5})
```

```
    for idx, row in merged_data.iterrows():
        # Check for valid geometry before attempting to access centroid
        if not row.geometry.is_empty and row.geometry.is_valid:
            centroid = row.geometry.centroid
            # Annotating community area name
            plt.annotate(text=row['community_area'], xy=(centroid.x, centroid.y)
                         horizontalalignment='center', fontsize=10, color='black

    ax.set_title('Figure 2.3.Theft Rate Across Community Areas in Chicago', font
    ax.axis('off')
    ax.set_aspect('equal', 'box')  # Explicitly setting the aspect ratio

    plt.show()
else:
    print("Merged data is empty. Check the data merging process.")
```

Figure 2.3.Theft Rate Across Community Areas in Chicago



The heatmap "Theft Rate Across Community Areas in Chicago" based on area division, represents the theft rate in different community areas. The colour on the map changes from light orange to dark red, representing changes in theft rates from low to high. Inside each community area, there is a value marked, which represents the theft rate in that area.

It can be seen from the map that the area numbered 32(Loop) has the highest theft rate, which is 0.54; while the area numbered 26(West Garfield Park) has the lowest theft rate, which is 0.10. The map shows that central areas tend to have higher theft rates, while peripheral areas tend to have lower rates. Also, the high theft rate areas cluster towards the Eastern North in Chicago. This pattern may reflect factors such as high population

density and high-income levels in urban central areas, which may increase the incidence of theft. Additionally, areas with high theft rates are often clustered together, creating an apparent "hot spot," suggesting that the theft problem may be systemic in certain communities.

Since theft rate and battery rate are show opposite patterns refer to the primary type distribution map, the partterns of battery rate can be viewed as the community areas with darker shade experience less battery and lighter color experience more battery than theft. Despite the positive correlation of population and crime per capita, as we can see from the graph that the differences between battery and theft are clear. In other words, community areas with darker shades have more theft than battery incidents, while lighter shades have more battery than theft incidents. The difference is our main focus, which helps exploring the reason of areas with a significant difference among various crime types.

Overall, this heat map is an important tool for city planners, law enforcement and community organizations to understand and respond to urban crime patterns. Through such visualization, relevant departments can allocate resources more effectively, develop targeted preventive measures, and monitor trends in theft rates.

In [43]:
```
chicago_crime['community_area'] = chicago_crime['community_area'].astype(str)

# Group by community_area and calculate the total number of crimes and arrests
gb_crime = chicago_crime.groupby('community_area')['arrest'].agg(['sum', 'count'

# Calculate the arrest rate
gb_crime['arrest_rate'] = gb_crime['sum'] / gb_crime['count']

# Rename columns for clarity
gb_crime.rename(columns={'sum': 'total_arrests', 'count': 'total_crimes', 'commu

# Create the final DataFrame with arrest rates
gb_crime_arrest_rate = gb_crime[['community_area', 'arrest_rate']]

gb_crime_arrest_rate
```

Out[43]:

	community_area	arrest_rate
0	1.0	0.205213
1	10.0	0.136453
2	11.0	0.161686
3	12.0	0.102462
4	13.0	0.151033
...	...	...
72	75.0	0.207261
73	76.0	0.262621
74	77.0	0.180335
75	8.0	0.189564
76	9.0	0.116454

77 rows × 2 columns

In [44]:
```
print(community_areas_gdf['community_area'].unique())
print(gb_crime_arrest_rate['community_area'].unique())
```

```
['35' '36' '37' '38' '39' '4' '40' '41' '42' '1' '11' '12' '13' '14' '15'
 '16' '17' '18' '19' '2' '20' '21' '22' '23' '24' '25' '26' '27' '28' '29'
 '3' '30' '31' '33' '34' '10' '8' '32' '43' '44' '45' '46' '47' '59' '6'
 '48' '49' '5' '50' '51' '52' '53' '54' '55' '56' '57' '58' '60' '61' '62'
 '63' '64' '65' '66' '67' '68' '69' '7' '70' '71' '72' '73' '74' '75' '76'
 '77' '9']
['1.0' '10.0' '11.0' '12.0' '13.0' '14.0' '15.0' '16.0' '17.0' '18.0'
 '19.0' '2.0' '20.0' '21.0' '22.0' '23.0' '24.0' '25.0' '26.0' '27.0'
 '28.0' '29.0' '3.0' '30.0' '31.0' '32.0' '33.0' '34.0' '35.0' '36.0'
 '37.0' '38.0' '39.0' '4.0' '40.0' '41.0' '42.0' '43.0' '44.0' '45.0'
 '46.0' '47.0' '48.0' '49.0' '5.0' '50.0' '51.0' '52.0' '53.0' '54.0'
 '55.0' '56.0' '57.0' '58.0' '59.0' '6.0' '60.0' '61.0' '62.0' '63.0'
 '64.0' '65.0' '66.0' '67.0' '68.0' '69.0' '7.0' '70.0' '71.0' '72.0'
 '73.0' '74.0' '75.0' '76.0' '77.0' '8.0' '9.0']
```

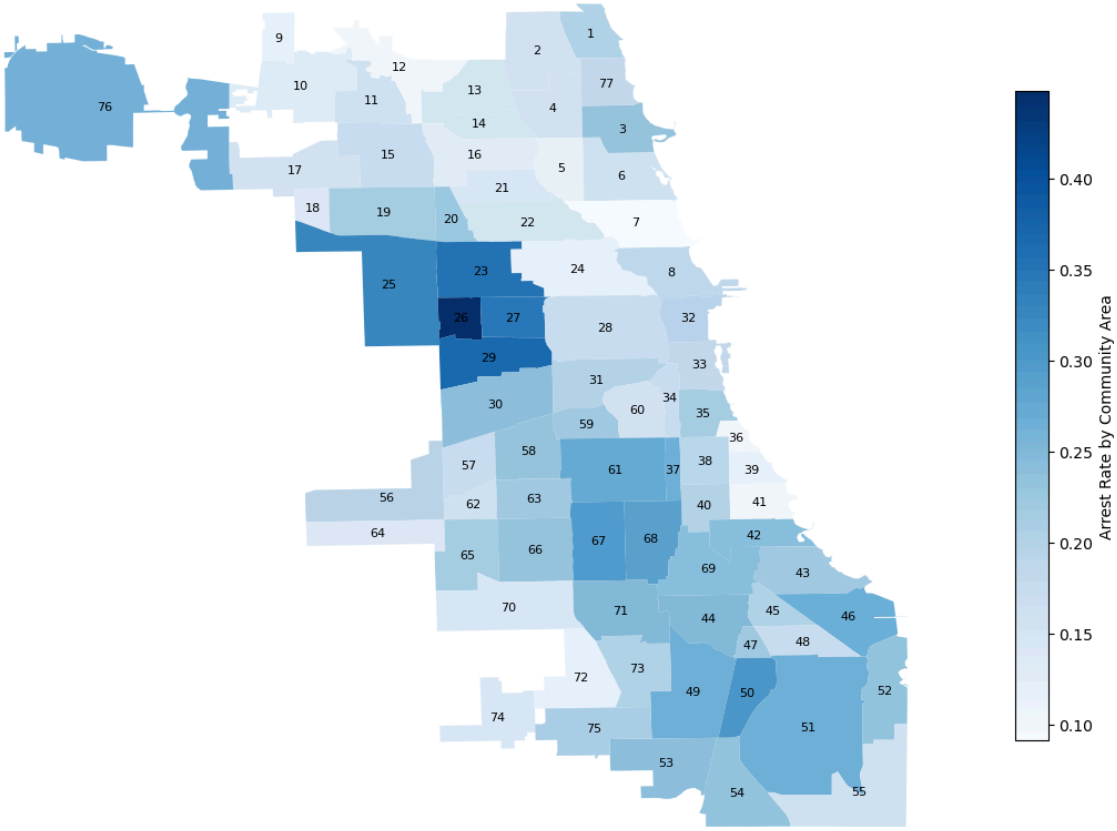
In [45]:
```
# Convert crime_by_community community_area values from float strings to integer
gb_crime_arrest_rate = gb_crime_arrest_rate.copy()
gb_crime_arrest_rate['community_area'] = gb_crime_arrest_rate['community_area'].

merged_gdf = community_areas_gdf.merge(gb_crime_arrest_rate, on='community_area'

print(merged_gdf.shape)
```

```
(77, 11)
```

In [46]:
```
fig, ax = plt.subplots(1, 1, figsize=(12, 12))
merged_gdf.plot(column='arrest_rate', ax=ax, cmap='Blues', legend=True,
                legend_kwds={'label': "Arrest Rate by Community Area", 'orientat

ax.set_aspect('equal')

# Annotate each community area with its code and arrest rate in percentage
for idx, row in merged_gdf.iterrows():
```

```
    centroid = row.geometry.centroid
    arrest_rate_percentage = f"{row['arrest_rate'] * 100:.1f}%"
    label = f"{row['community_area']}"
    ax.text(centroid.x, centroid.y, label, fontsize=8, ha='center', va='center')

plt.title('Figure 2.4.Arrest Rate by Community Area in Chicago')
plt.tight_layout()
ax.axis('off')
plt.show()
```

Figure 2.4.Arrest Rate by Community Area in Chicago



The map "Arrest Rate by Community Area" is also a heatmap, which links the darker shades to higher arrest rates and lighter colours to lower arrest rates. In other words, the community areas with darker shades might be considered relatively more effective police sources. The map is crucial for understanding the efficiency of law enforcement efforts across different community areas and can also provide insights into crime prevention.

The map reveals that the community area with the highest arrest rate is Area 26 West Garfield Park with 44.8%, while the lowest arrest rate is Area 7 Lincoln Park with 9.1%. That is to say, there are almost half of the criminals ended up arrested in Area 26 while 90% of criminals were not arrested in Area 7, which is extremely low that the government has to take measures to effectively mitigate the criminal activities.

From the perspective of spatial trends, some adjacent areas show contrasting arrest rates. There is also a clustering of higher arrest rates towards the center to the West and Southern East in Chicago, while the areas close to borders are shown in lower arrest rates. Compared to the theft rate map, it appears the pattern that community areas with

higher theft rates report lower arrest rates, which reveals social-economic factors like police sources and law enforcement.

Because of the distinct nature of theft and battery, it could be relatively more difficult to track non-violent incidents such as theft compared to violent crimes which usually appear as proofs helping to arrest. Hence, the government should focus on the arrest rate issue as it connects to the crime rate directly. Crimes may happen more frequently if the criminals realize that they won't get an arrest, which aggravates criminal activities if the issue cannot be solved in the long term.

2.3 Merging with A New Dataset

2.3.1 Data Source

The new dataset "Chicago Health Atlas Data Download - Community areas.csv" is retrived from Chicago Health Atlas, including unemployement rate(UMP), per capita income(PCI), and high school graduation rate(EDB) from 2014 to 2018, corresponding to the five years pre-pandemic in this paper. Per capita income provides the economic aspect, unemployment rate represents the social perspective, and high school graduation rate gives the demograohical insights. With the social-economic indicators, we could analyze the underlying factors influencing the crime rate in each area combing the localized situation with our research.

Source: https://chicagohealthatlas.org/neighborhood

2.3.2 Data Merging

2.3.2.1 Data Merging and Cleaning

In [47]:
```
chicago_crime['community_area'] = chicago_crime['community_area'].astype(float).
chicago_health['community_area'] = chicago_health['geoid'].apply(lambda x: int(f

chicago_merged = chicago_crime.merge(chicago_health, on='community_area', how='l

chicago_merged
```

Out[47]:

	id	case_number	date	block	iucr	primary_type	description
0	12536164	JE439378	2015-09-24	031XX W 53RD PL	1753	OFFENSE INVOLVING CHILDREN	SEXUAL ASSAULT OF CHILD BY FAMILY MEMBER
1	12536166	JE439332	2014-09-07	031XX W 53RD PL	1753	OFFENSE INVOLVING CHILDREN	SEXUAL ASSAULT OF CHILD BY FAMILY MEMBER
2	13158716	JG362691	2018-11-09	017XX N NASHVILLE AVE	0265	CRIMINAL SEXUAL ASSAULT	AGGRAVATED - OTHER
3	13188119	JG397237	2015-05-15	041XX W 24TH PL	1754	OFFENSE INVOLVING CHILDREN	AGGRAVATED SEXUAL ASSAULT OF CHILD BY FAMILY M...
4	13193781	JG397432	2015-06-18	031XX S KOSTNER AVE	1752	OFFENSE INVOLVING CHILDREN	AGGRAVATED CRIMINAL SEXUAL ABUSE BY FAMILY MEMBER
...	...	...	...	...	...	...	...
1343219	13125149	JG322052	2017-01-01	021XX W HOWARD ST	0820	THEFT	$500 AND UNDER
1343220	13135737	JG335293	2018-03-15	079XX S HONORE ST	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300
1343221	13077134	JG264727	2018-09-30	028XX N MC VICKER AVE	0266	CRIMINAL SEXUAL ASSAULT	PREDATORY
1343222	13169093	JG374535	2017-09-08	006XX E 89TH ST	0810	THEFT	OVER $500
1343223	13069800	JG256610	2016-05-01	062XX S LANGLEY AVE	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300

1343224 rows × 30 columns

In [48]:
```
if 'Layer' in chicago_merged:
    chicago_merged = chicago_merged.drop(columns=['Layer'])
```

```
chicago_merged.columns = chicago_merged.columns.str.lower()

chicago_merged
```

Out[48]:

	id	case_number	date	block	iucr	primary_type	description
0	12536164	JE439378	2015-09-24	031XX W 53RD PL	1753	OFFENSE INVOLVING CHILDREN	SEXUAL ASSAULT OF CHILD BY FAMILY MEMBER
1	12536166	JE439332	2014-09-07	031XX W 53RD PL	1753	OFFENSE INVOLVING CHILDREN	SEXUAL ASSAULT OF CHILD BY FAMILY MEMBER
2	13158716	JG362691	2018-11-09	017XX N NASHVILLE AVE	0265	CRIMINAL SEXUAL ASSAULT	AGGRAVATED - OTHER
3	13188119	JG397237	2015-05-15	041XX W 24TH PL	1754	OFFENSE INVOLVING CHILDREN	AGGRAVATED SEXUAL ASSAULT OF CHILD BY FAMILY M...
4	13193781	JG397432	2015-06-18	031XX S KOSTNER AVE	1752	OFFENSE INVOLVING CHILDREN	AGGRAVATED CRIMINAL SEXUAL ABUSE BY FAMILY MEMBER
...	...	...	...	...	...	...	...
1343219	13125149	JG322052	2017-01-01	021XX W HOWARD ST	0820	THEFT	$500 AND UNDER
1343220	13135737	JG335293	2018-03-15	079XX S HONORE ST	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300
1343221	13077134	JG264727	2018-09-30	028XX N MC VICKER AVE	0266	CRIMINAL SEXUAL ASSAULT	PREDATORY
1343222	13169093	JG374535	2017-09-08	006XX E 89TH ST	0810	THEFT	OVER $500
1343223	13069800	JG256610	2016-05-01	062XX S LANGLEY AVE	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300

1343224 rows × 30 columns

2.3.3 Mapping with the Merged Data

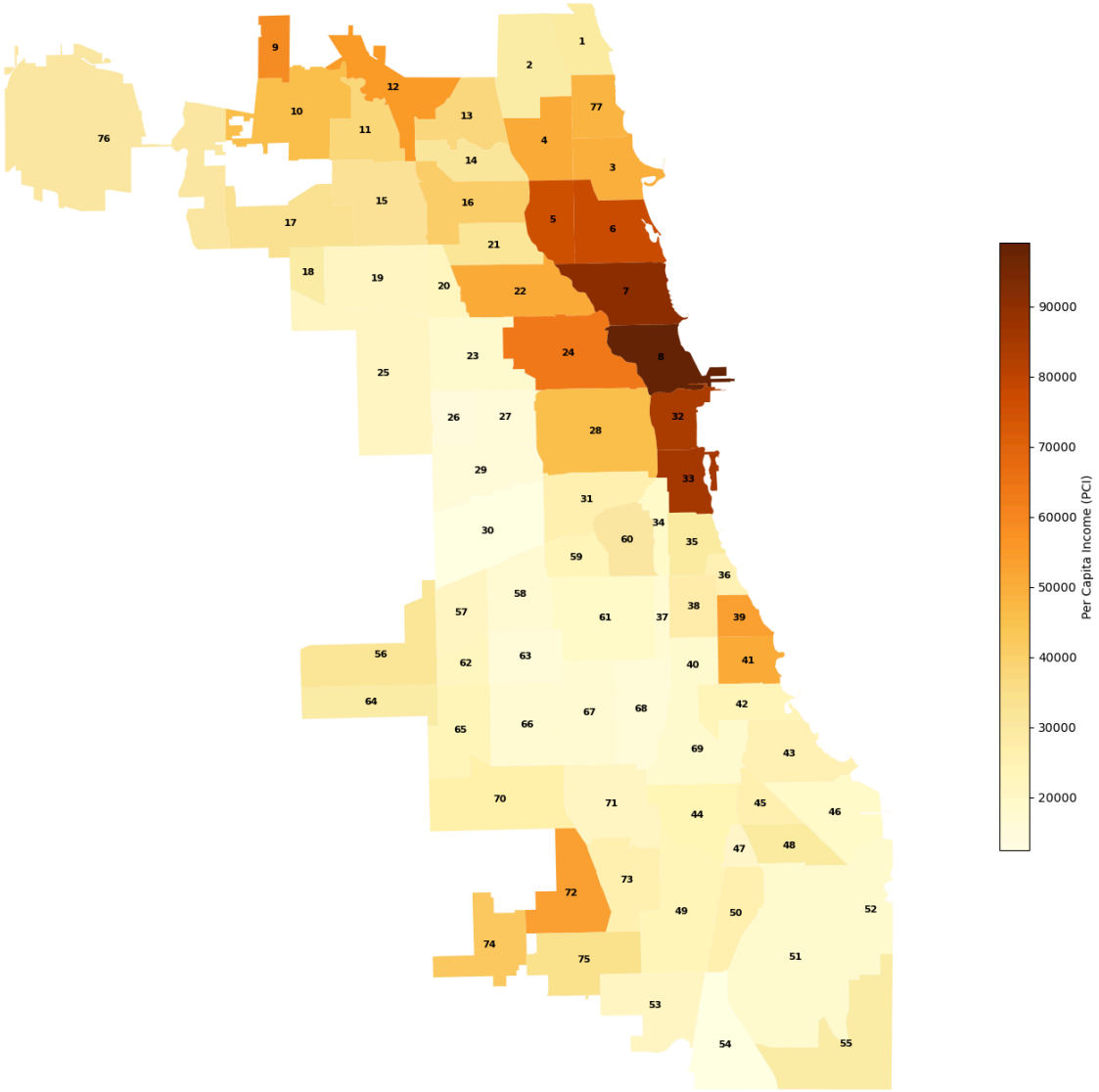
2.3.3.1 Map Per Capita Income

In [49]:
```
print(community_areas_gdf['community_area'].dtype)
print(chicago_health['community_area'].dtype)

community_areas_gdf['community_area'] = community_areas_gdf['community_area'].as
chicago_health['community_area'] = chicago_health['community_area'].astype(str)

merged_gdf = community_areas_gdf.merge(chicago_health[['community_area', 'pci']]

print(merged_gdf.columns)
```

```
object
int64
Index(['area', 'area_num_1', 'community_area', 'comarea', 'comarea_id',
       'community', 'perimeter', 'shape_area', 'shape_len', 'geometry', 'pci'],
      dtype='object')
```

In [50]:
```
fig, ax = plt.subplots(1, 1, figsize=(18, 18))
merged_gdf.plot(column='pci', ax=ax, legend=True,
                legend_kwds={'label': "Per Capita Income (PCI)",
                             'orientation': "vertical",
                              'shrink': 0.5},
                cmap='YlOrBr')  # Use 'YlOrBr' colormap for yellow to brown grad

for idx, row in merged_gdf.iterrows():
    centroid = row.geometry.centroid
    community_area_text = f"{row['community_area']}"
    pci_text = f"{row['pci']:.2f}"

    ax.text(centroid.x, centroid.y, community_area_text, ha='center', va='center
            fontsize=8, fontweight='bold', color='black')

plt.title('Figure 2.5.PCI in Community Areas in Chicago ')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
ax.axis('off')
plt.show()
```

Figure 2.5.PCI in Community Areas in Chicago



The map "PCI in Community Areas in Chicago" illustrates the per capita income in each community area in Chicago. PCI stands for the average income earned per person in a certain area(USD per year), which reflects the income level. In the heatmap, the darker the shade represents the higher the PCI while the lighter color refers to the lower PCI.

From the spatial perspective, it appears the higher CPI clustering towards the Eastern North in Chicago, with community area 8 Near North Side as the highest one. The community areas close to the borders also generate a higher CPI level compared to the areas in the center. Compared to the theft rate map and the crime type map, it appears the patterns that community areas generate higher income report higher theft rates, while community areas with lower income report battery as the primary crime type. Notably, communities 5(North Center),6(Lake View),7(Lincoln Park),8(Near North Side) and 32(Loop) generate high income and also encounter high theft rates. It reveals the problem of law enforcement of the protection of residents' property and the necessity of more strict monitoring of public areas.

For the community areas with high income, the potential valuable properties attract more steals. The poverty and richness are revealed by the income level disparity, which indicate that community areas in hardships tend to experience more violence because of the social pressure or other potential factors.

Additionally, for lower-income community areas, battery crimes emerged as the predominant crime type, which may reflect the economic pressures and social tensions in these neighbourhoods. This analysis not only reveals the link between crime types and economic status but also highlights the importance of policing strategies that target the specific needs of different communities.

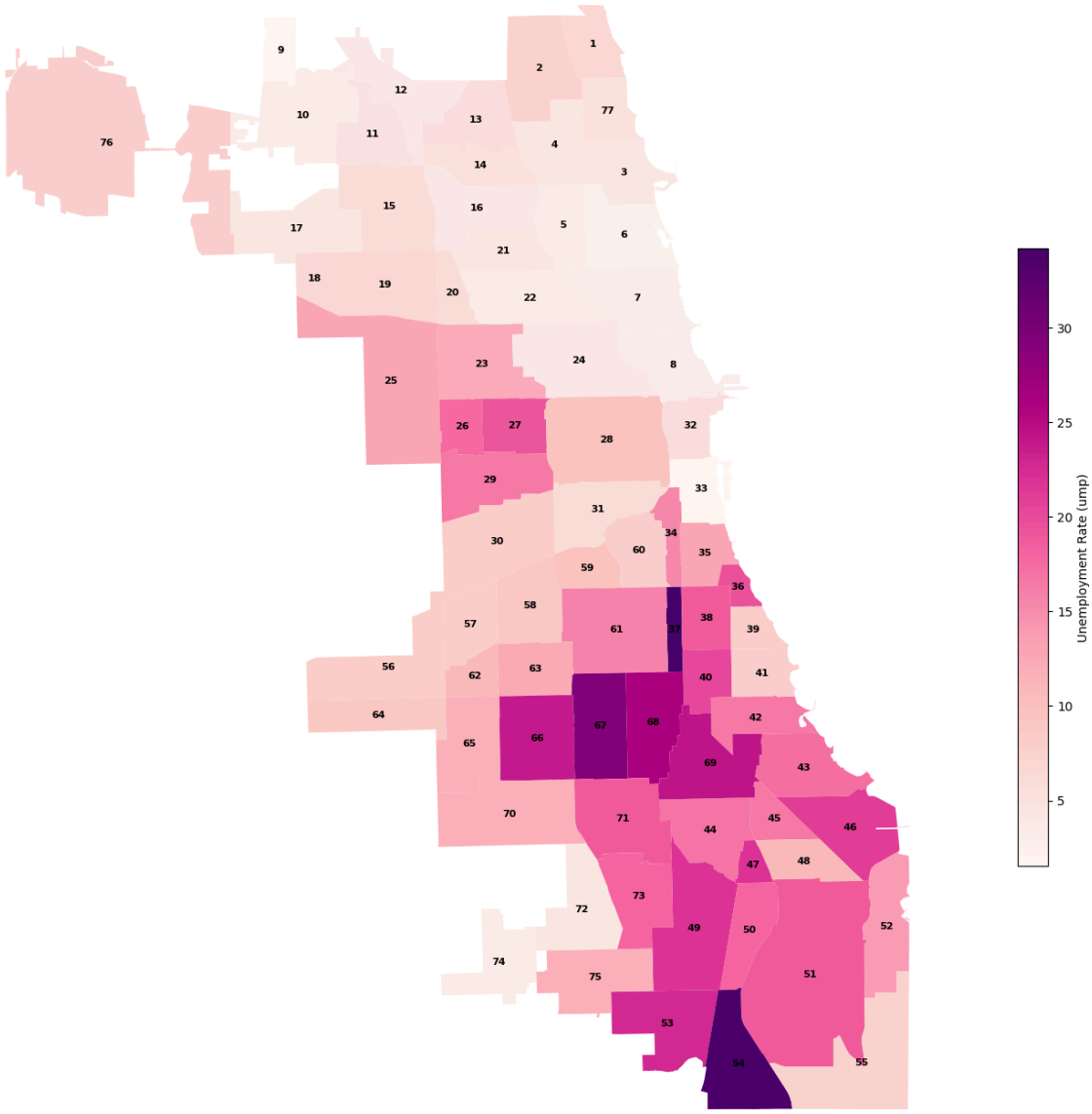
2.3.3.2 Map Unemployment Rate

In [51]:
```
community_areas_gdf['community_area'] = community_areas_gdf['community_area'].as
chicago_health['community_area'] = chicago_health['community_area'].astype(str)

merged_gdf_ump = community_areas_gdf.merge(chicago_health[['community_area', 'um

print(merged_gdf_ump.columns)
```

```
Index(['area', 'area_num_1', 'community_area', 'comarea', 'comarea_id',
       'community', 'perimeter', 'shape_area', 'shape_len', 'geometry', 'ump'],
      dtype='object')
```

In [52]:
```
fig, ax = plt.subplots(1, 1, figsize=(18, 18))
merged_gdf_ump.plot(column='ump', ax=ax, legend=True,
                    legend_kwds={'label': "Unemployment Rate (ump)",
                                 'orientation': "vertical",
                                 'shrink': 0.5},  # Adjust legend size
                    cmap='RdPu')  # Use 'Greys' colormap for white to grey gradi

for idx, row in merged_gdf_ump.iterrows():
    centroid = row.geometry.centroid
    community_area_text = f"{row['community_area']}"
    ump_text = f"{row['ump']:.2f}"

    ax.text(centroid.x, centroid.y, community_area_text, ha='center', va='center
            fontsize=8, fontweight='bold', color='black')

plt.title('Figure 2.6.UMP in Community Areas in Chicago')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
ax.axis('off')
plt.show()
```

Figure 2.6.UMP in Community Areas in Chicago



The map titled "UMP in Community Areas in Chicago" illustrates the unemployment rate of community areas in Chicago with darker shades representing higher unemployment rates. It appears an explicit geographic pattern that the unemployment rate is higher in the South of Chicago and the North part is not facing the problem.

The highest unemployment rate appears in community areas 37(Fuller Park) and 54(Riverdale) at 34.2% and 33.73%, respectively. Compared to the theft rate map and the crime type map, it appears the pattern of community areas with higher unemployment rates tend to have a higher frequency of battery crimes, while communities with lower unemployment rates have a relatively higher proportion of theft crimes.

The phenomenon reflects the localized factors influencing crime type across community areas. Communities with higher unemployment rates may face increased economic stress and social dissatisfaction, leading to an increase in violent crimes such as battery. In contrast, although communities with lower unemployment rates have better economic conditions, they may have more targets for theft, thereby attracting theft crimes. This

analysis has important implications for understanding crime prevention and community management, especially mitigating the problem of high theft rate.

By identifying the dominant crime types in a given community, policymakers and law enforcement agencies can more effectively allocate resources and target prevention and intervention strategies to improve the overall safety and well-being of the community. For example, communities with higher unemployment could increase employment support while social services may be needed to reduce assault crimes; for communities with lower unemployment, increased protection of physical property may be needed to prevent theft crimes.

3. Project 3

3.1 Potential Data to Scrape

Economic disparity encompasses a wide array of perspectives. In our previous paper, we employed per capita income as the primary metric to illustrate economic disparities across community areas based on income levels. While income is typically viewed as a periodic cash flow—received monthly or annually—it is just one aspect of economic factors. Wealth, on the other hand, represents a more static dimension of economic status, encapsulating what an individual currently possesses. A comprehensive analysis of economic conditions requires the integration of both dynamic (income) and static (wealth) elements. To this end, we have selected house prices, including both average price per square foot and median list price, as additional metrics to represent the static aspect of economic status.

The data was extracted from https://www.openhausrealty.com/blog/unlocking-the-diversity-of-chicagos-77-community-areas-a-comprehensive-guide/, a website that aggregates links to detailed pages for each of Chicago's 77 community areas, necessitating individual access to each area's specific link.

Our previous research concluded that community areas with higher income levels tend to report higher incidences of non-violent crime, such as theft, whereas areas with lower income levels are more prone to violent crimes, such as battery. By incorporating house price data to represent wealth, we aim to elucidate the relationship between wealth levels and crime types, thereby providing a more nuanced understanding of economic disparity and its effects on societal outcomes. The variable per capita income in our previous research reflects the poverty level, while the house price can be the indicator of living-cost, rental fee, and economic inequality. In this case, mergeing with the dataset of house price help us have a more explicit, broader view on the economic indicators from more perspectives. Combining income level and house price, we can further explore whether the main factor influencing crime type is poverty level or economic inequality.

3.2 Potential Challenges

The primary challenge in my project involves the intricate navigation required on the website, which compiles information for all 77 community areas through individual links, rather than directly presenting house prices on a single page. This complexity necessitates navigating through multiple pages or interacting with the site, such as clicking on each community area's picture, to access house price information for each community. Moreover, the presentation of house prices, while visually resembling a table, is not structured as one, complicating the identification of the correct HTML class to scrape.

Despite these hurdles, I devised a strategy to overcome the navigation challenge. My approach involves scraping the main URL to gather the links for the 77 community areas and compiling them into a list. Subsequently, I crafted a function designed to extract the house prices (average price per square foot and median listing price) for a given link. Utilizing a for loop, this function is applied to each link, enabling the collection of house price data for all 77 community areas. Although this method is time-consuming, it successfully circumvents the navigation issue. To address the issue of identifying the correct HTML class for extracting house price data, I discovered that the desired information resides exclusively within the si-property-stats__item-value class. By targeting this specific div class, I was able to efficiently gather the necessary data.

Looking ahead, I might employ a similar strategy for scraping websites that require interaction. However, this approach may not be viable or could become excessively time-consuming if a site necessitates extensive navigation. This experience underscores the complexity of web scraping in the face of varying website structures and the need for adaptive strategies to efficiently gather data. If the website's structure evolves to include other types of information within the same class, the method of setting class may lose its efficacy. In such scenarios, it will be imperative to revaluate and identify a more specific class element that reliably corresponds to the information we seek to scrape.

3.3 Scraping Data from a Website

Firstly, define a function to retrieve the link for each community area in the main url and store them into a list as our first loop. With the list, we can create a new for loop to repeat retrieving house price for each link.

In [53]:
```
# Define a function to get the link for each community area in the main url
def get_community_links(main_url):
    response = requests.get(main_url)
    soup = BeautifulSoup(response.content, 'html.parser')
    links = soup.find_all('a', href=True)
    community_links = [link['href'] for link in links if 'community-area' in lin
    return community_links
```

In [54]:
```
# Get all the links for the community areas and store into a list
main_url = 'https://www.openhausrealty.com/blog/unlocking-the-diversity-of-chica
community_links = get_community_links(main_url)
```

The links are duplicated in different formats, so we filter all the links in the completed format that could be used directly.

In [55]:
```
# Filter all the links in the completed format that we need
filtered_links = {link for link in community_links if link.startswith('https://w
```

Now, we define a function to get average house price for each community area's individual link and return the value.

In [56]:
```
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
# Define a function that get the average price per sq.ft for a given link
def get_avg_price_per_sqft(url):
    response = requests.get(url, headers=headers)
    soup = BeautifulSoup(response.content, 'html.parser')
    card = soup.find_all('div', class_ = 'si-property-stats__item-value')
    price_div = card[2]  # This is your specific div
    price_text = price_div.get_text()
    price_value = float(price_text.replace('$', '').strip())
    return price_value
```

Similarly, we define a function to get median house listing price for each community area's individual link and return the value.

In [57]:
```
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
# Define a function that get the median price for a given link
def get_med_price(url):
    response = requests.get(url, headers=headers)
    soup = BeautifulSoup(response.content, 'html.parser')
    card = soup.find_all('div', class_ = 'si-property-stats__item-value')
    price_med = card[3]
    price_med_text = price_med.get_text()
    price_med_value = price_med_text.replace('$', '').strip()
    return price_med_value
```

We firstly use for loop to get the individual link in the link list that we created above. Then in each for loop we call the two functions we defined to get the average house price and the median house listing price for each individual link and store them into a new list. Finally, we convert the list to dataframe.

In [58]:
```
community_data = []

# Loop through the filtered links and get the price for each community
for link in filtered_links:
    # Call the get_avg_price_per_sqft to get the average price for a given commu
    ave_price = get_avg_price_per_sqft(link)
    # Call the get_med_price to get the median price for a given community area
    med_price = get_med_price(link)
    # Get the community area name from the link
    community_name = link.split('/')[-2]
    # Append community name, average price and median price for each community a
    community_data.append({'community_area_house': community_name, 'ave_price($/

# Create a pandas DataFrame and transform the list to DataFrame
chicago_houseprice = pd.DataFrame(community_data)
```

```
chicago_houseprice
```

Out[58]:

	community_area_house	ave_price($/Sq.Ft)	med_price($)
0	near-north-side	492.30	627,450
1	garfield-ridge	249.83	380,000
2	north-center	400.57	760,000
3	south-chicago	115.54	187,000
4	south-shore	110.40	181,000
...	...	...	...
71	east-side	166.07	234,900
72	forest-glen	271.36	589,450
73	austin	178.93	320,000
74	belmont-cragin	235.36	399,500
75	hyde-park	225.62	285,000

76 rows × 3 columns

The community names are messy so we uniform the names for further merge in the later part.

In [59]:
```
chicago_houseprice['community_area_house'] = chicago_houseprice['community_area_
chicago_houseprice
```

Out[59]:

	community_area_house	ave_price($/Sq.Ft)	med_price($)
0	near north side	492.30	627,450
1	garfield ridge	249.83	380,000
2	north center	400.57	760,000
3	south chicago	115.54	187,000
4	south shore	110.40	181,000
...	...	...	...
71	east side	166.07	234,900
72	forest glen	271.36	589,450
73	austin	178.93	320,000
74	belmont cragin	235.36	399,500
75	hyde park	225.62	285,000

76 rows × 3 columns

3.4 Visualizing the Scraped Dataset

In [60]:
```
comm_area = pd.read_csv('comm_areas.csv')
comm_area
```

Out[60]:

	Area_Number	Community_Name
0	1	Rogers Park
1	2	West Ridge
2	3	Uptown
3	4	Lincoln Square
4	5	North Center
...	...	...
72	73	Washington Heights
73	74	Mount Greenwood
74	75	Morgan Park
75	76	O'Hare
76	77	Edgewater

77 rows × 2 columns

In [61]:
```
comm_area['Community_Name'] = comm_area['Community_Name'].str.lower()
comm_area
```

Out[61]:

	Area_Number	Community_Name
0	1	rogers park
1	2	west ridge
2	3	uptown
3	4	lincoln square
4	5	north center
...	...	...
72	73	washington heights
73	74	mount greenwood
74	75	morgan park
75	76	o'hare
76	77	edgewater

77 rows × 2 columns

In [62]:
```
chicago_houseprice['community_area_house'] = chicago_houseprice['community_area_
chicago_houseprice['community_area_house'] = chicago_houseprice['community_area_
chicago_houseprice['community_area_house'] = chicago_houseprice['community_area_
chicago_houseprice['community_area_house'] = chicago_houseprice['community_area_
chicago_houseprice['community_area_house'] = chicago_houseprice['community_area_
chicago_houseprice['community_area_house'] = chicago_houseprice['community_area_
chicago_houseprice['community_area_house'] = chicago_houseprice['community_area_
```

In [63]:
```
chicago_house = pd.merge(chicago_houseprice, comm_area, how='inner', left_on='co
```

There is a missing value for loop community area 32, hence I retrived the data from the website and set it manually. https://www.realtor.com/realestateandhomes-search/The-Loop_Chicago_IL/overview

In [64]:
```
chicago_house.loc[len(chicago_house)] = ['loop', 400.00, 459900, 32, 'loop']

chicago_house
```

Out[64]:

	community_area_house	ave_price($/Sq.Ft)	med_price($)	Area_Number	Community_
0	near north side	492.30	627,450	8	near nor
1	garfield ridge	249.83	380,000	56	garfiel
2	north center	400.57	760,000	5	north
3	south chicago	115.54	187,000	46	south c
4	south shore	110.40	181,000	43	south
...	...	...	...	...	
72	forest glen	271.36	589,450	12	fore
73	austin	178.93	320,000	25	
74	belmont cragin	235.36	399,500	19	belmont
75	hyde park	225.62	285,000	41	hyc
76	loop	400.00	459900	32	

77 rows × 5 columns

In [65]:
```
crime_theft_battery = pd.merge(theft_rate, battery_rate, on='community_area', ho

crime_theft_battery
```

Out[65]:

	community_area	theft_rate	battery_rate
0	1.0	0.256503	0.192601
1	2.0	0.233180	0.171565
2	3.0	0.265578	0.183798
3	4.0	0.283788	0.159193
4	5.0	0.369161	0.091864
...	...	...	...
72	73.0	0.186677	0.187244
73	74.0	0.265881	0.147020
74	75.0	0.232895	0.182104
75	76.0	0.293134	0.101526
76	77.0	0.311384	0.168480

77 rows × 3 columns

In [66]:
```
crime_theft_battery['community_area'] = crime_theft_battery['community_area'].as
chicago_house['Area_Number'] = chicago_house['Area_Number'].astype(str)
crime_house = pd.merge(crime_theft_battery, chicago_house, left_on='community_ar
```

In [67]:
```
crime_house = crime_house.drop(['Area_Number', 'Community_Name'], axis=1)

crime_house
```

Out[67]:

	community_area	theft_rate	battery_rate	community_area_house	ave_price($/Sq.Ft)
0	1	0.256503	0.192601	rogers park	220.52
1	2	0.233180	0.171565	west ridge	199.92
2	3	0.265578	0.183798	uptown	285.76
3	4	0.283788	0.159193	lincoln square	314.80
4	5	0.369161	0.091864	north center	400.57
...	...	...	...	...	...
72	73	0.186677	0.187244	washington heights	150.58
73	74	0.265881	0.147020	mount greenwood	245.94
74	75	0.232895	0.182104	morgan park	170.82
75	76	0.293134	0.101526	o'hare	238.40
76	77	0.311384	0.168480	edgewater	277.86

77 rows × 6 columns

In [68]:
```
crime_house['ave_price($/Sq.Ft)'] = pd.to_numeric(crime_house['ave_price($/Sq.Ft
crime_house['theft_rate'] = pd.to_numeric(crime_house['theft_rate'], errors='coe
crime_house['battery_rate'] = pd.to_numeric(crime_house['battery_rate'], errors=

crime_house.dropna(subset=['ave_price($/Sq.Ft)', 'theft_rate', 'battery_rate'],

plt.figure(figsize=(12, 6))

plt.scatter(crime_house['ave_price($/Sq.Ft)'], crime_house['theft_rate'], color=
plt.scatter(crime_house['ave_price($/Sq.Ft)'], crime_house['battery_rate'], colo

z_theft = np.polyfit(crime_house['ave_price($/Sq.Ft)'], crime_house['theft_rate'
p_theft = np.poly1d(z_theft)

z_battery = np.polyfit(crime_house['ave_price($/Sq.Ft)'], crime_house['battery_r
p_battery = np.poly1d(z_battery)

plt.plot(crime_house['ave_price($/Sq.Ft)'], p_theft(crime_house['ave_price($/Sq.
plt.plot(crime_house['ave_price($/Sq.Ft)'], p_battery(crime_house['ave_price($/S

# Adding labels and legend
plt.xlabel('Average Price per Sq.Ft ($)')
plt.ylabel('Rate')
plt.title('Figure 3.1.Theft and Battery Rates vs. Average Price per Sq.Ft')
plt.legend()

# Show plot
plt.show()
```

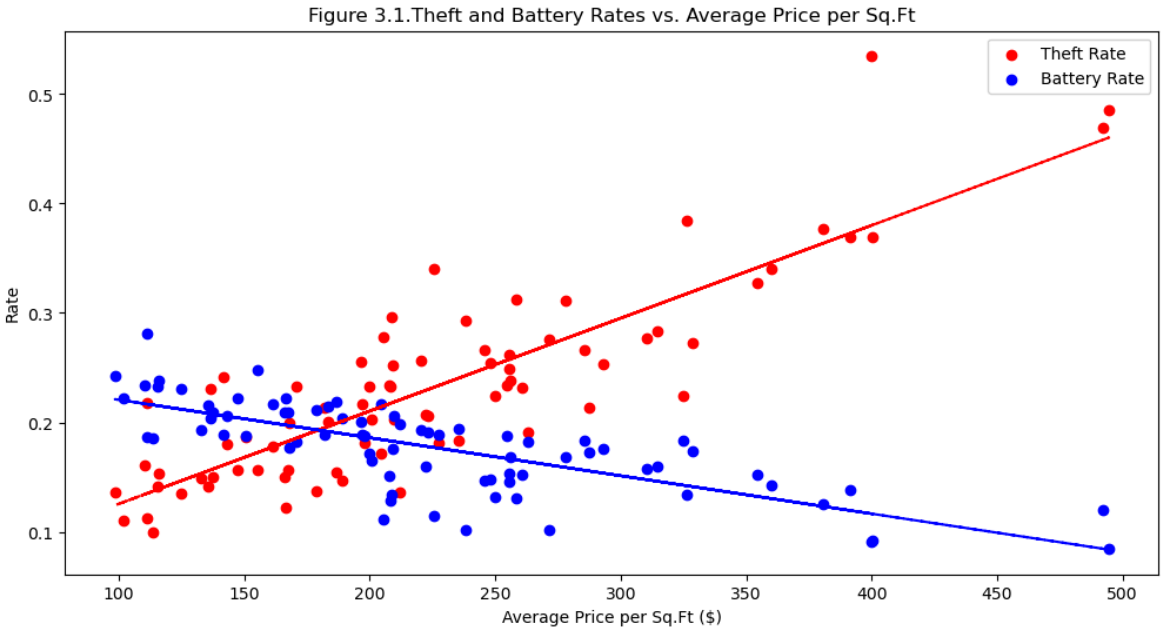


Figure 3.1, a scatter plot, depicts the relationship between theft and battery rates and the average house price per square foot in Chicago, and the pattern is similar to Figure 2.1. Red dots represent theft rates, while blue dots signify battery rates. The trend lines across these dots distinctly illustrate a positive correlation between the theft rate and the average house price, whereas there is a negative correlation between the battery rate and the average house price. In essence, community areas with higher average house prices tend to have a higher frequency of non-violent crimes (primarily theft), whereas areas with lower house prices exhibit a higher frequency of violent crimes (primarily

battery). This observation reinforces the conclusion drawn in our previous paper: there is a positive correlation between income level and non-violent crime rate. Income level reflects the dynamic financial conditions of community areas, while house price offers a static perspective.

The phenomenon observed is logical, as communities with higher income levels tend to have greater purchasing power, which in turn sustains high price levels. Moreover, elevated house prices indicate higher demand, reflecting the increased purchasing power of individuals capable of affording property investments. Consumer confidence is also a key indicator, as people are more inclined to purchase homes when they feel financially secure.

By integrating the insights from Figure 2.1 with those from Figure 3.1, we can formulate an evidence-supported conclusion: community areas with better economic conditions are more likely to experience higher rates of non-violent crimes, whereas areas with poorer economic conditions may encounter social and economic stresses that lead to a higher prevalence of violent activitie

Income level, representing the poverty and richness across community areas only reflect the disposable income of residents. The average house price could reflect both economic inequality and the living cost. Rental fee is one of the main parts of living cost. If peopel can afford rental fee, they may not need to steal, while if not, they might commit theft incidents due to the high living stress. Combine the income level and the house price, we could figure out the economic disparity from different perspectives, hence to provide more accurate and data-driven insights for further exploration.s.

In [69]:
```
community_areas_gdf['community_area'] = community_areas_gdf['community_area'].st
crime_house['community_area_house'] = crime_house['community_area'].str.strip()

merged_gdf_ave_price = community_areas_gdf.merge(crime_house, on='community_area
```

In [70]:
```
fig, ax = plt.subplots(1, 1, figsize=(18, 18))
merged_gdf_ave_price.plot(column='ave_price($/Sq.Ft)', cmap='OrRd', linewidth=0.
                          legend_kwds={'label': "Average Price($/sq.ft)",
                                       'orientation': "vertical",
                                       'shrink': 0.5})

# Iterate over each row to annotate the community area and average price
for idx, row in merged_gdf_ave_price.iterrows():
    # Determine the centroid for the annotation
    centroid = row['geometry'].centroid
    community_area_text = row['community_area']
    ave_price_text = f"{int(row['ave_price($/Sq.Ft)'])}"

    # Annotate community area
    ax.text(centroid.x, centroid.y, community_area_text, ha='center', va='center
            fontsize=10, fontweight='bold', color='black')

ax.set_title('Figure 3.2.Average House Price per Sq.Ft by Community Area in Chic
ax.axis('off')
plt.show()
```

Figure 3.2.Average House Price per Sq.Ft by Community Area in Chicago

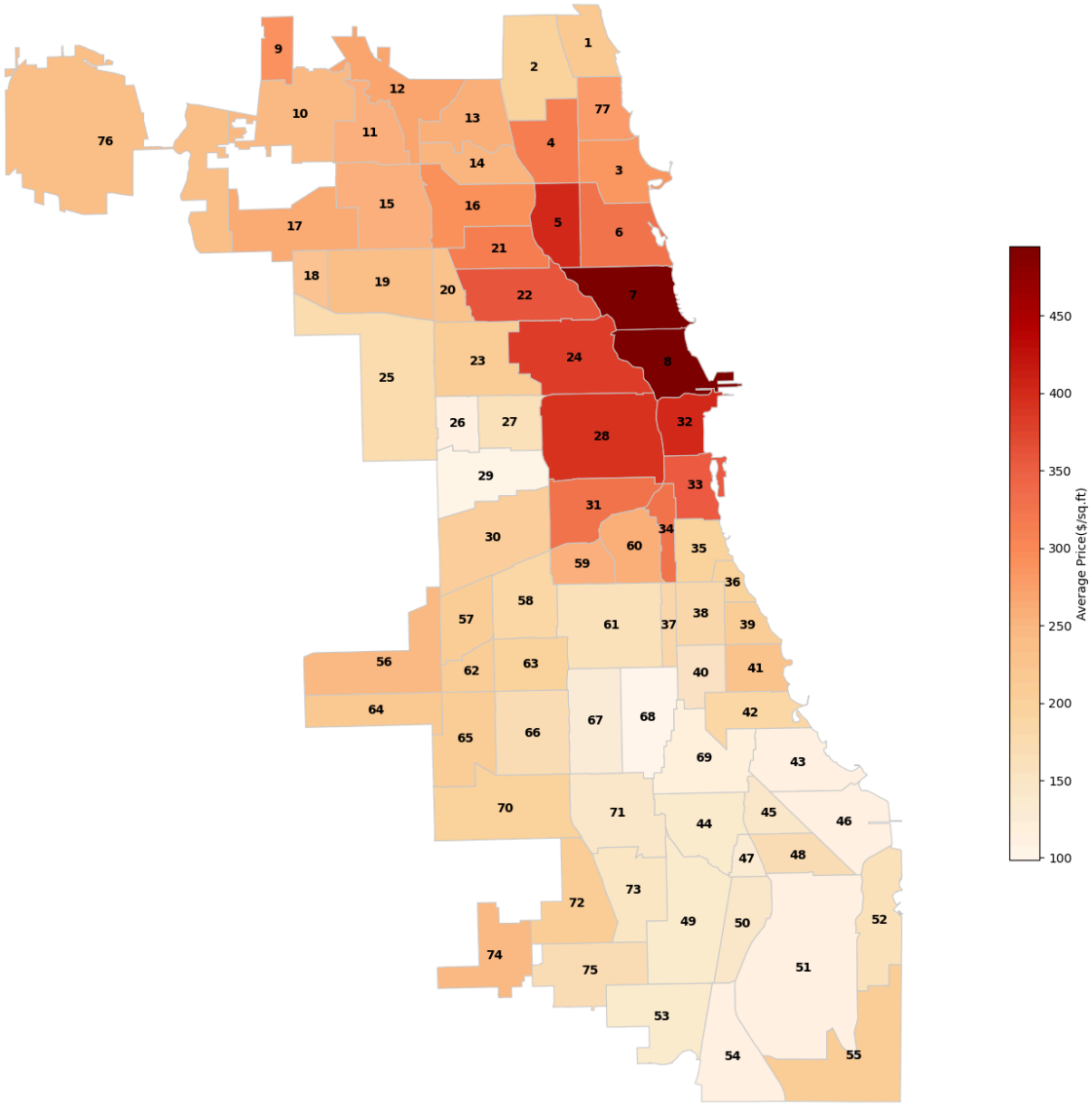


Figure 3.2, a heatmap, displays the average house prices per square foot in Chicago, using color intensity to represent price levels. The darkest shades, indicating the highest prices, are predominantly found in the northeastern part of Chicago, with the northern regions generally exhibiting darker tones compared to the southern areas. Notably, community area 8 cholds the highest average house price at USD 496 per square foot, closely followed by community area 7 at USD 491 per square foot.

When compared to Figure 2.5, which is a heatmap of per capita income in Chicago, there's a noticeable parallel in trends. Specifically, community areas Near North Side 8 and Lincoln Park 7 not only exhibit the highest house prices but also the highest income levels, underscoring a correlation between income and housing costs. Additionally, regions with elevated per capita incomes consistently show higher house prices, reinforcing our hypothesis.

Moreover, an analysis juxtaposed with Figure 2.3, which maps the theft rates across Chicago's communities, reveals that areas 8 and 7 also experience high theft rates. Furthermore, area 32, which has the highest theft rate, also ranks high in terms of per

capita income and house price. This observation lends further support to the positive relationship between wealth and non-violent crime rates, suggesting that affluent areas, while attracting non-violent crimes like theft, are not necessarily immune to crime but exhibit different crime patterns compared to less affluent areas.

In [71]:
```
merged_gdf_ave_price['med_price($)'] = pd.to_numeric(merged_gdf_ave_price['med_p
```

In [72]:
```
fig, ax = plt.subplots(1, 1, figsize=(18, 18))
merged_gdf_ave_price.plot(column='med_price($)', cmap='OrRd', linewidth=0.8, ax=
                          legend_kwds={'label': "Median Price($)",
                                       'orientation': "vertical",
                                       'shrink': 0.5})

for idx, row in merged_gdf_ave_price.iterrows():
    # Determine the centroid for the annotation
    centroid = row['geometry'].centroid
    community_area_text = row['community_area']

    # Ensure the value is treated as a string and remove commas before convertin
    med_price_text = f"{int(str(row['med_price($)']).replace(',', ''))}"

    # Annotate community area
    ax.text(centroid.x, centroid.y, community_area_text, ha='center', va='center
            fontsize=10, fontweight='bold', color='black')

ax.set_title('Figure 3.3.Median House Listing Price by Community Area in Chicago
ax.axis('off')
plt.show()
```

Figure 3.3.Median House Listing Price by Community Area in Chicago

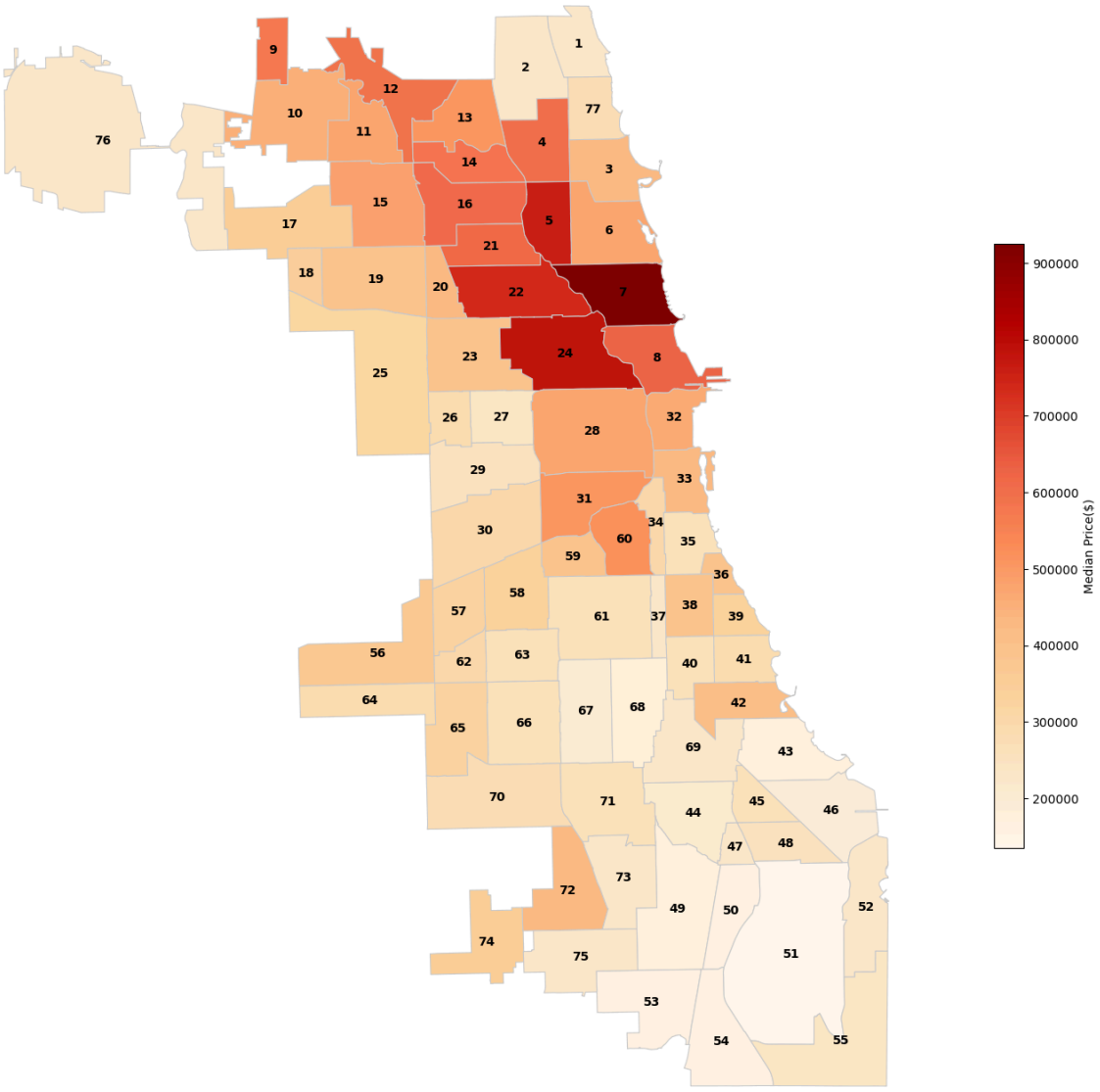


Figure 3.3, another heatmap, depicts median house listing prices across Chicago, with darker shades indicating higher prices. This map exhibits a spatial pattern akin to Figure 3.2, which maps average house prices per square foot, showing a clustering of higher median house prices in the northeastern part of Chicago, with northern areas generally pricier than southern ones. Specifically, Community 7 Lincoln Park stands out with the highest median listing price at USD 9,000,000, aligning with its high average house price. While Community 8 Near North Side also displays a high median house price, it is slightly lower than that of Community 7, possibly due to the inclusion of various housing types like condos, apartments, and houses in the median price calculation. Different housing types, along with other factors such as size, location, and amenities, can significantly influence prices.

In this research, we do not delve deeply into individual house price analytics but instead use these prices as a broad indicator of economic conditions to assess the financial landscape across communities. The overarching trend observed is that areas with higher income levels tend to have higher average and median house prices, which correlates with an increased incidence of non-violent crimes, particularly theft. This observation

lends further support to our conclusion regarding the relationship between wealth and non-violent crime rates in community areas.

3.5 Adding a New Dataset

The dataset contains the total population across each community area in Chicago, which is retrived from robparal, downloaded as a csv file and uploaded to the file. By opting to incorporate population data into the dataset, the research can now be approached from a population-based perspective. The potential reason for community areas have more crime incidents could just be there are more people, to avoid this effect, we add the population as a new variable to calculate the crime rates based on capita. When we comparing acorss geographic, we standardize the variables by population to ensure that every comparison is at the same level.

Datasource: https://robparal.com/chicago-data/

In [73]:
```
chicago_population = pd.read_csv("Chicago Community Areas Population.csv")
chicago_population['community_area'] = chicago_population['community_area'].str.
chicago_population
```

Out[73]:

	community_area	population
0	Rogers Park	54,872
1	West Ridge	77,212
2	Uptown	58,424
3	Lincoln Square	41,713
4	North Center	35,705
...	...	...
72	Washington Heights	27,354
73	Mount Greenwood	19,342
74	Morgan Park	21,674
75	O'Hare	12,549
76	Edgewater	57,022

77 rows × 2 columns

In [74]:
```
chicago_population['community_area'] = chicago_population['community_area'].str.
chicago_crime_pop = pd.merge(chicago_population, comm_area, left_on='community_a
chicago_crime_pop = chicago_crime_pop.drop(columns=['Community_Name'])
chicago_crime_pop = chicago_crime_pop.rename(columns={'community_area': 'communi
chicago_crime_pop = chicago_crime_pop.rename(columns={'Area_Number': 'community_
chicago_crime_pop = chicago_crime_pop.rename(columns={'population': 'chicago_pop
```

In [75]:
```
chicago_crime_pop
```

Out[75]:

	community_name	chicago_pop	community_area
0	rogers park	54,872	1
1	west ridge	77,212	2
2	uptown	58,424	3
3	lincoln square	41,713	4
4	north center	35,705	5
...	...	...	...
72	washington heights	27,354	73
73	mount greenwood	19,342	74
74	morgan park	21,674	75
75	o'hare	12,549	76
76	edgewater	57,022	77

77 rows × 3 columns

In [76]:
```
chicago_crime_merged = pd.merge(chicago_merged, chicago_crime_pop, on='community
```

In [77]:
```
chicago_crime_merged
```

Out[77]:

	id	case_number	date	block	iucr	primary_type	descriptic
0	12536164	JE439378	2015-09-24	031XX W 53RD PL	1753	OFFENSE INVOLVING CHILDREN	SEXU ASSAULT C CHILD I FAMI MEMBI
1	12536166	JE439332	2014-09-07	031XX W 53RD PL	1753	OFFENSE INVOLVING CHILDREN	SEXU ASSAULT C CHILD I FAMI MEMBI
2	13211439	JG424021	2016-01-01	030XX W 55TH ST	1754	OFFENSE INVOLVING CHILDREN	AGGRAVATI SEXU ASSAULT C CHILD I FAMILY M
3	12982936	JG142891	2018-11-30	055XX S TROY ST	1754	OFFENSE INVOLVING CHILDREN	AGGRAVATI SEXU ASSAULT C CHILD I FAMILY M
4	10399166	HZ135577	2015-04-15	050XX S FAIRFIELD AVE	1752	OFFENSE INVOLVING CHILDREN	AGGRAVATI CRIMINA SEXU ABUSE I FAMI MEMBI
...	...	...	...	...	...	...	
1343219	12193411	JD399262	2017-01-11	129XX S CARONDOLET AVE	1754	OFFENSE INVOLVING CHILDREN	AGGRAVATI SEXU ASSAULT C CHILD I FAMILY M
1343220	12235817	JD448323	2018-10-24	027XX E 130TH ST	1153	DECEPTIVE PRACTICE	FINANCIA IDENTI THEFT OVI $ 3(
1343221	12286879	JE133683	2018-01-01	127XX S MUSKEGON AVE	0266	CRIMINAL SEXUAL ASSAULT	PREDATOI
1343222	13167324	JG372822	2015-05-14	028XX E 130TH ST	1310	CRIMINAL DAMAGE	T PROPER
1343223	13086365	JG276527	2016-09-01	131XX S BALTIMORE AVE	1754	OFFENSE INVOLVING CHILDREN	AGGRAVATI SEXU ASSAULT C CHILD I FAMILY M

1343224 rows × 32 columns

In [78]:
```
community_areas_gdf['community_area'] = community_areas_gdf['community_area'].as
chicago_crime_pop['community_area'] = chicago_crime_pop['community_area'].astype

merged_gdf_pop = community_areas_gdf.merge(chicago_crime_pop, on='community_area

merged_gdf_pop['chicago_pop'] = merged_gdf_pop['chicago_pop'].str.replace(',', '
```

In [79]:
```
fig, ax = plt.subplots(1, 1, figsize=(18, 18))
merged_gdf_pop.plot(column='chicago_pop', cmap='Blues', linewidth=0.8, ax=ax, ed
                    legend_kwds={'label': "Population by Community Area",
                                 'orientation': "vertical",
                                'shrink': 0.5})

for idx, row in merged_gdf_pop.iterrows():
    centroid = row['geometry'].centroid
    community_area_text = row['community_area']
    ax.text(centroid.x, centroid.y, community_area_text, ha='center', va='center
            fontsize=8, color='black')

ax.set_title('Figure 3.4.Population by Community Area in Chicago')
ax.axis('off')
plt.show()
```

Figure 3.4.Population by Community Area in Chicago

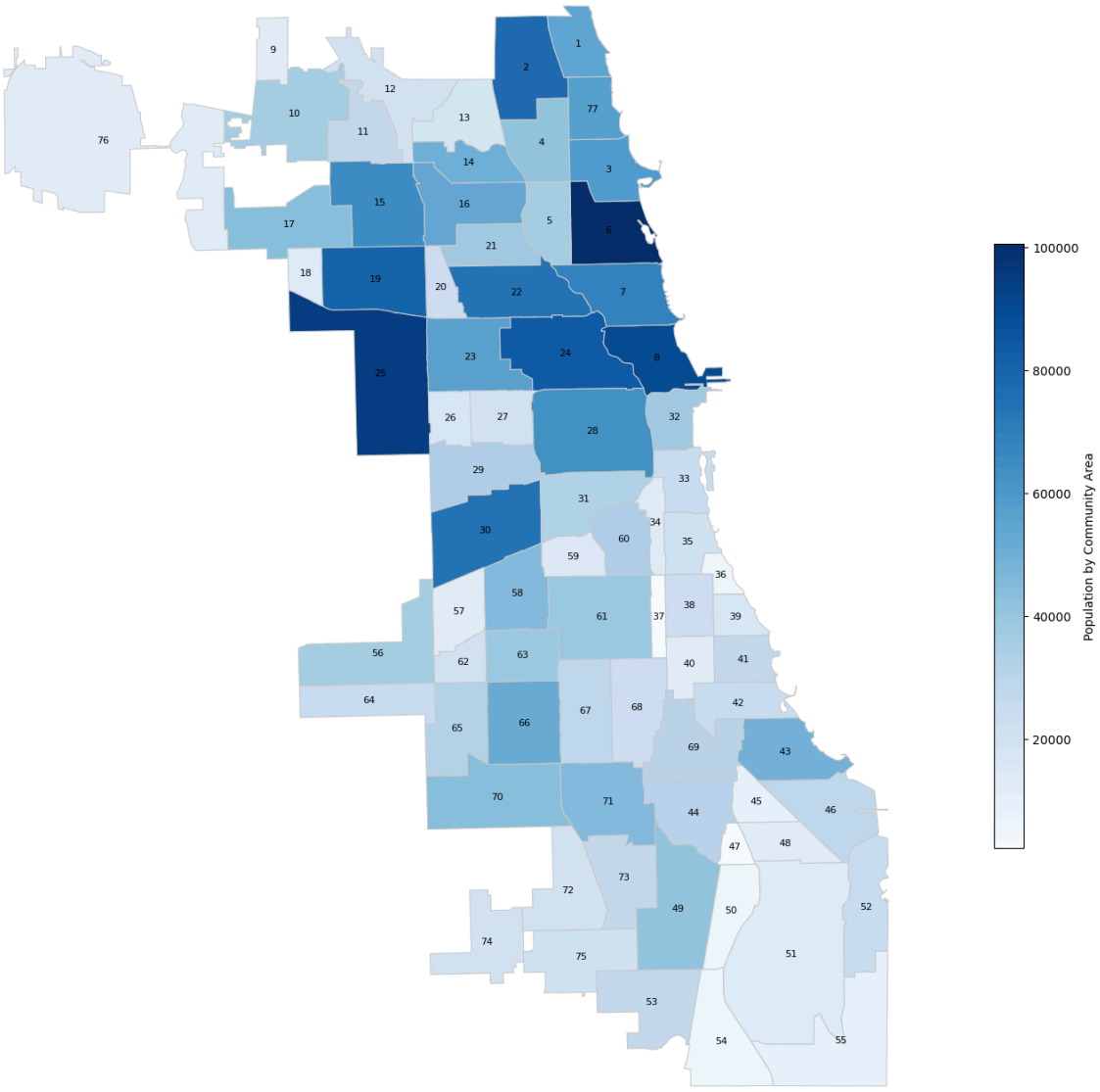


Figure 3.4, a heatmap, depicts the distribution of the population across each community area in Chicago, with darker shades indicating higher populations. From a spatial perspective, the majority of Chicago's population is concentrated towards the northern part, particularly in areas 25, 6, and 8, which are the most populous. This phenonemenon is consistent to our assumption that areas with larger population tend to have higher cirme frequency, hence we will include population as a variable in the regression part to eliminate its effet. Nevertheless, for this research, we are focusing on the difference and the comparison between the theft and battery. If both of the two types of crime are facing the same effect, the difference can offset the bias.

In comparison to the per capita income map, it is apparent that areas with higher income levels tend to have larger populations. This suggests that people are inclined to move to areas with better economic conditions, and conversely, areas with higher populations may generate greater income due to the availability of more labor. When comparing this to the housing price map, a positive correlation between house prices and population density emerges, implying that areas with larger populations generate higher demand for housing, thus driving up prices. However, the relationship between population

density and theft rates is not evident in this map, but will be further analyzed in the subsequent map.

In [80]:
```
chicago_crime_pop['community_area'] = chicago_crime_pop['community_area'].astype
chicago_crime_merged['community_area'] = chicago_crime_merged['community_area'].

theft_counts = chicago_crime_merged[chicago_crime_merged['primary_type'] == 'THE

battery_counts = chicago_crime_merged[chicago_crime_merged['primary_type'] == 'B

merged_df = pd.merge(chicago_crime_pop, theft_counts, on='community_area', how='
merged_df = pd.merge(merged_df, battery_counts, on='community_area', how='left')

if merged_df['chicago_pop'].dtype == object:
    merged_df['chicago_pop'] = pd.to_numeric(merged_df['chicago_pop'].str.replac

merged_df['theft_count'] = merged_df['theft_count'].fillna(0)
merged_df['battery_count'] = merged_df['battery_count'].fillna(0)

merged_df['theft_per_capita'] = (merged_df['theft_count'] / merged_df['chicago_p
merged_df['battery_per_capita'] = (merged_df['battery_count'] / merged_df['chica

chicago_crime_per_cap = merged_df[['community_area', 'theft_per_capita', 'batter

chicago_crime_per_cap
```

Out[80]:

	community_area	theft_per_capita	battery_per_capita
0	1	88.952471	66.791806
1	2	51.170802	37.649588
2	3	82.654389	57.202520
3	4	65.087623	36.511399
4	5	72.818933	18.120711
...	...	...	...
72	73	108.283980	108.613000
73	74	41.981181	23.213732
74	75	112.761834	88.170158
75	76	208.223763	72.117300
76	77	67.254744	36.389464

77 rows × 3 columns

In [81]:
```
community_areas_gdf['community_area'] = community_areas_gdf['community_area'].as
chicago_crime_per_cap.loc[:, 'community_area'] = chicago_crime_per_cap['communit

merged_gdf_crime = community_areas_gdf.merge(chicago_crime_per_cap, on='communit

fig, ax = plt.subplots(1, 1, figsize=(18, 18))
merged_gdf_crime.plot(column='theft_per_capita', cmap='Reds', linewidth=0.8, ax=
                      legend_kwds={'label': "Theft per Capita by Community Area"
```

```
                                    'orientation': "vertical",
                                    'shrink': 0.5})

for idx, row in merged_gdf_crime.iterrows():
    centroid = row['geometry'].centroid
    community_area_text = row['community_area']
    ax.text(centroid.x, centroid.y, community_area_text, ha='center', va='center
            fontsize=8, color='black')

ax.set_title('Figure 3.5.Theft per Capita by Community Area in Chicago')
ax.axis('off')
plt.show()
```

Figure 3.5.Theft per Capita by Community Area in Chicago

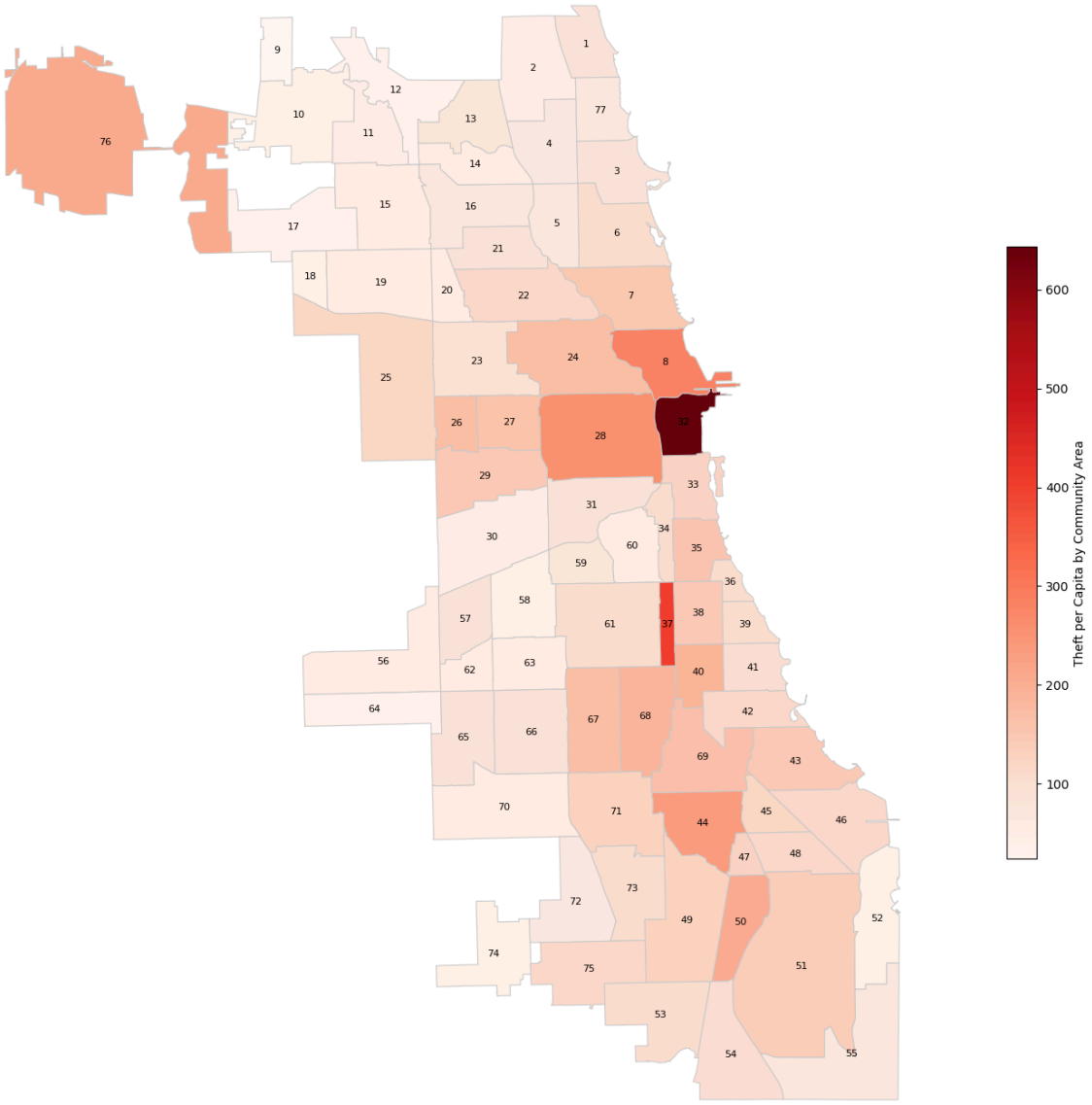


Figure 3.5, also a heatmap, delineates the theft per capita in each community area of Chicago, with darker shades indicating higher theft rates per capita. Similar to previous findings, the spatial analysis reveals that theft per capita is higher in Northern Chicago compared to the South, with community areas 32, 37, 28, and 8 experiencing the highest rates of theft per capita. In relation to the population distribution map, there appears to be a weak positive correlation between theft per capita and population size. This suggests that larger populations, which bring about greater diversity, may contribute to

an increase in crime rates. When this is compared to Figure 2.2, the map of primary crime types, it is clear that areas with higher theft per capita often report theft as the most frequent crime. Additionally, a positive correlation is observed between house prices and theft per capita. This may indicate that areas with higher house prices, which typically reflect better economic conditions, attract more thefts as criminals target more valuable assets.

Final Project

4.1 Regression

4.1.1 Economic Relationship

The current paper examines the link between economic conditions, non-violent crimes and violent offences with a particular focus on battery and theft. We want to find out whether people from different economic backgrounds commit nonviolent crimes such as theft at similar rates. Furthermore, we wish to establish the main factors that affect the types of crimes committed especially taking into account the impact of absolute economic levels together with wealth-poverty disparities.

Previous studies show that there is a generally positive relationship between an individual's income and assets and their likelihood of committing non-violent offenses. Nonetheless, it is unclear what form this correlation takes—linear or nonlinear? To answer this question we will consider two variables—income and assets held. Our argument here is based on the idea of diminishing marginal utility according to which although more money leads to greater satisfaction, each additional dollar gives slightly less pleasure than before. As a result, while rich people may continue performing property crimes, but their number might be reduced in comparison with those in lower income brackets. The evidence from our previous data as well as graphical analysis supports our hypothesis in this regard. For instance, Figure 2.1 demonstrates a broader distribution of crimes among individuals with higher annual incomes, suggesting that wealthier individuals may feel less compelled to engage in these crimes, including both non-violent and violent acts.

Also, we analyze the influence of poverty or residing in community areas with vast wealth inequalities on crime. Research demonstrates that it is not just poverty but also a sense of relative deprivation that leads to theft. However, this dynamic is complicated. For example, the rich may feel poor compared to the super-rich, however they are less likely to steal for survival. Our charts especially figure 2.1 supports this argument as well. It indicates that lower than sixty thousand dollars annual income have higher non-violent crimes within them as shown by our graphical representations. On the other hand, richer areas have fewer such crimes implying that sufficiency in economic means probably reduces stealing tendencies.

4.1.2 Choose Xs

Prepare the final merged dataset.

In [82]:
```
crime_house['community_area'] = crime_house['community_area'].astype(int)
chicago_crime_final = pd.merge(chicago_crime_merged, crime_house, on='community_
```

1. pci: Chicago's per capita income is measured in dollars and it's called pci. By examining the area per capita income, one can learn more about the economic status of that neighborhood. In most cases, higher per capita income means better economic conditions including higher property values and more thefts. The intent is to include pci so as to establish a relationship between crime rates committed non-violently with the earnings of different neighborhoods. Our objective is primarily to find out if areas with high incomes have a high incidence of theft.

2. ump: The 'ump' stands for unemployment rate for community area within Chicago. Unemployment rate was chosen because it has direct correlation with income levels which indicate economic hardship. A rise in unemployment may lead to an increase in criminal activities due to financial pressure or lower price associated with unlawful deals. Conversely, communities with highest levels of unemployment are usually characterized by lowest average earnings and increased deprivation in general terms. UMP will enable us understand how crime relates to variations in wealth, poverty and income among other factors such as unemployment rate. For instance, areas that experience higher joblessness might undergo increased poverty levels which may be associated with increased theft numbers.

3. edb: The 'edb' variable refers to high school completion, which is a key measure of educational attainment. Education level is also associated with economic status and crime rates. Places with high levels of education tend to have more income and more wealth. Furthermore, better educated neighborhoods may have lower crime rates in general. With the edb we can examine educational factors that affect crime rates.

4. chicago_pop: 'chicago_pop' represents the demographic of the Chicago area. The theft rate and battery rate in this study reflect the proportion of these offenses in the overall crime rate, unadjusted for the general population. This approach has been chosen to focus on factors affecting crime patterns rather than considering percentage of population violent. However, demographics are taken into account in this part to understand whether higher crime rates in some areas are simply due to higher population densities.

5. arrest_boolean: 'arrest_boolean' is a dummy variable where 0 means non-arrested and 1 means arrested. The variable is important for understanding law enforcement and police resource allocation. It can also highlight differences in arrest rates for nonviolent and violent crimes. Community areas with better economic condition may have better surveillance and consequently more arrests. We examine whether high theft rates in some areas are due more to reported incidents than to the actual prevalence of battery thefts.

6. domestic_boolean: 'Domestic Boolean' is another dummy variable where 0 represents non-domestic residents, and 1 indicates domestic residents. Like chicago population, they include examining demographic patterns in neighborhoods and whether higher crime rates are due to higher non-household or household populations. The variables help control other factors to understand the economy of the factors that best define crime patterns.

7. ave_price: 'ave_price' means the average housing price per square foot in the Chicago area, expressed in USD. It is an important indicator of the economic status of the community. Unlike the pci, which measures income volatility, house prices reflect purchasing power and wealth in the area. In areas where homes are expensive, rising property values can lead to an increase in thefts. We include both PCI and house prices to examine the effects of income and wealth on crime patterns.

8. med_price: 'med_price' represents the median listing price of a home in USD in local areas in Chicago. Like 'ave_price', 'med_price' also highlights the level of wealth and property values in the area. The inclusion of mean and median values is the goal of longitudinal analysis.

9. log_pci: 'log_pci' applies a logarithmic transformation to the pci variable. This adjustment addresses the skewness of the distribution of income indicators, increases the robustness of the regression model, and adjusts for nonlinear relationships. This adjustment is consistent with our assumption of a nonlinear relationship between income inequality and the crime characteristic line coincides.

10. log_ave_price: Similarly, 'Log Ave Price' applies a logarithmic transformation to the house price distribution, facilitating nonlinear analysis in our model.

11. arrest_pci_interaction: 'arrest_pci_interaction' stands for the multiplier of 'arrest_boolean' and 'pci' variable. This interaction variable can shed light on how economic conditions and crime rates influence law enforcement efficiency. We aim to determine the effect of economic status on arrest rates and, indirectly, crime rates.

These selected variables include social, demographic, and law enforcement, all of which aim to examine how economic disparity affects crime types. Our goal is to create a more accurate model that remains simple, which we will achieve by controlling nonstationary effects, manipulating variables, and associated interaction variables

4.1.3 Run Regressions

4.1.3.1 Separate Regressions

In [83]:
```
X = chicago_crime_final[['pci', 'ump']]
Y = chicago_crime_final['theft_rate']

X = sm.add_constant(X)

reg_pci_ump = sm.OLS(Y, X).fit()
```

```
predictions = reg_pci_ump.predict(X)

print(reg_pci_ump.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.877
Model:                            OLS   Adj. R-squared:                  0.877
Method:                 Least Squares   F-statistic:                 4.785e+06
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:33   Log-Likelihood:             2.4258e+06
No. Observations:             1343224   AIC:                        -4.852e+06
Df Residuals:                 1343221   BIC:                        -4.851e+06
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.1076      0.000    762.794      0.000       0.107       0.108
pci         4.162e-06      2e-09   2077.890      0.000    4.16e-06    4.17e-06
ump           -0.0015   6.23e-06   -245.027      0.000      -0.002      -0.002
==============================================================================
Omnibus:                    81036.699   Durbin-Watson:                   0.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           107793.839
Skew:                           0.562   Prob(JB):                         0.00
Kurtosis:                       3.814   Cond. No.                     1.71e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.71e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [84]:
```
X = chicago_crime_final[['pci', 'ump','edb']]
Y = chicago_crime_final['theft_rate']

X = sm.add_constant(X)

reg_pci_ump_edb = sm.OLS(Y, X).fit()
predictions = reg_pci_ump_edb.predict(X)

print(reg_pci_ump_edb.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.877
Model:                            OLS   Adj. R-squared:                  0.877
Method:                 Least Squares   F-statistic:                 3.197e+06
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:34   Log-Likelihood:             2.4271e+06
No. Observations:             1343224   AIC:                        -4.854e+06
Df Residuals:                 1343220   BIC:                        -4.854e+06
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.1270      0.000    312.755      0.000       0.126       0.128
pci         4.279e-06   3.05e-09   1405.302      0.000    4.27e-06    4.29e-06
ump           -0.0014   6.68e-06   -210.077      0.000      -0.001      -0.001
edb           -0.0003   5.88e-06    -51.132      0.000      -0.000      -0.000
==============================================================================
Omnibus:                    76155.302   Durbin-Watson:                   0.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           100109.350
Skew:                           0.542   Prob(JB):                         0.00
Kurtosis:                       3.783   Cond. No.                     4.93e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.93e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [85]:
```
chicago_crime_final['chicago_pop'] = pd.to_numeric(chicago_crime_final['chicago_

chicago_crime_final.replace([np.inf, -np.inf], np.nan, inplace=True)

chicago_pop_mean = chicago_crime_final['chicago_pop'].mean()

chicago_crime_final['chicago_pop'].fillna(chicago_pop_mean, inplace=True)

X = chicago_crime_final[['chicago_pop', 'arrest_boolean']]
Y = chicago_crime_final['theft_rate']


X = sm.add_constant(X)

reg_pop_arrest = sm.OLS(Y, X).fit()
predictions = reg_pop_arrest.predict(X)

print(reg_pop_arrest.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.120
Model:                            OLS   Adj. R-squared:                  0.120
Method:                 Least Squares   F-statistic:                 9.186e+04
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:37   Log-Likelihood:             1.1049e+06
No. Observations:             1343224   AIC:                        -2.210e+06
Df Residuals:                 1343221   BIC:                        -2.210e+06
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const              0.1686      0.000    837.998      0.000       0.168       0.16
9
chicago_pop     1.453e-06    3.6e-09    403.489      0.000    1.45e-06    1.46e-0
6
arrest_boolean    -0.0291      0.000   -133.714      0.000      -0.030      -0.02
9
==============================================================================
Omnibus:                   180180.722   Durbin-Watson:                   0.026
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           268543.832
Skew:                           0.990   Prob(JB):                         0.00
Kurtosis:                       3.935   Cond. No.                     1.39e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.39e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [86]:
```
X = chicago_crime_final[['chicago_pop', 'arrest_boolean', 'domestic_boolean']]
Y = chicago_crime_final['theft_rate']

X = sm.add_constant(X)

reg_pop_arrest_domestic = sm.OLS(Y, X).fit()
predictions = reg_pop_arrest_domestic.predict(X)

print(reg_pop_arrest_domestic.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.142
Model:                            OLS   Adj. R-squared:                  0.142
Method:                 Least Squares   F-statistic:                 7.418e+04
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:38   Log-Likelihood:            1.1217e+06
No. Observations:             1343224   AIC:                        -2.243e+06
Df Residuals:                 1343220   BIC:                        -2.243e+06
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
===
                       coef    std err          t      P>|t|      [0.025      0.9
75]
--------------------------------------------------------------------------------
---
const                0.1793      0.000    866.494      0.000       0.179       0.
180
chicago_pop       1.406e-06   3.56e-09    394.395      0.000     1.4e-06    1.41e
-06
arrest_boolean      -0.0309      0.000   -143.304      0.000      -0.031      -0.
030
domestic_boolean    -0.0432      0.000   -184.809      0.000      -0.044      -0.
043
==============================================================================
Omnibus:                   168472.867   Durbin-Watson:                   0.074
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           244820.309
Skew:                           0.951   Prob(JB):                         0.00
Kurtosis:                       3.872   Cond. No.                     1.54e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.54e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [87]:
```
X = chicago_crime_final[['pci', 'ave_price($/Sq.Ft)']]
Y = chicago_crime_final['theft_rate']

X = sm.add_constant(X)

reg_ave_price = sm.OLS(Y, X).fit()
predictions = reg_ave_price.predict(X)

print(reg_ave_price.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.901
Model:                            OLS   Adj. R-squared:                  0.901
Method:                 Least Squares   F-statistic:                 6.110e+06
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:38   Log-Likelihood:            2.5717e+06
No. Observations:             1343224   AIC:                        -5.143e+06
Df Residuals:                 1343221   BIC:                        -5.143e+06
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
=====
                           coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-----
const                   0.0431   7.57e-05    569.788      0.000       0.043
0.043
pci                  2.863e-06    2.9e-09    988.917      0.000    2.86e-06    2.8
7e-06
ave_price($/Sq.Ft)      0.0004   6.26e-07    633.032      0.000       0.000
0.000
==============================================================================
Omnibus:                   113633.016   Durbin-Watson:                   0.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           155697.509
Skew:                           0.710   Prob(JB):                         0.00
Kurtosis:                       3.876   Cond. No.                     1.02e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.02e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [128…

```
chicago_crime_final['med_price($)'] = pd.to_numeric(chicago_crime_final['med_pri

med_price_mean = chicago_crime_final['med_price($)'].dropna().mean()
chicago_crime_final['med_price($)'].fillna(med_price_mean, inplace=True)

X = chicago_crime_final[['pci', 'ave_price($/Sq.Ft)', 'med_price($)']]
Y = chicago_crime_final['theft_rate']

X = sm.add_constant(X)

reg_ave_med_price = sm.OLS(Y, X).fit()
predictions = reg_ave_med_price.predict(X)

print(reg_ave_med_price.summary())

linreg_mse = mean_squared_error(Y, predictions)
print("Mean Squared Error:", linreg_mse)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.912
Model:                            OLS   Adj. R-squared:                  0.912
Method:                 Least Squares   F-statistic:                 4.627e+06
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        16:49:28   Log-Likelihood:             2.6493e+06
No. Observations:             1343224   AIC:                        -5.299e+06
Df Residuals:                 1343220   BIC:                        -5.298e+06
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
=====
                           coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-----
const                    0.0480   7.24e-05    662.282      0.000       0.048
0.048
pci                   2.659e-06   2.78e-09    956.952      0.000    2.65e-06    2.6
6e-06
ave_price($/Sq.Ft)       0.0006   8.18e-07    764.695      0.000       0.001
0.001
med_price($)         -1.371e-07   3.38e-10   -405.404      0.000   -1.38e-07   -1.3
6e-07
==============================================================================
Omnibus:                    31415.634   Durbin-Watson:                   0.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            51267.637
Skew:                           0.224   Prob(JB):                         0.00
Kurtosis:                       3.845   Cond. No.                     1.01e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.01e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
Mean Squared Error: 0.001133420345781528
```

In [89]:
```
X = chicago_crime_final[['pci','ump']].copy()

X['log_pci'] = np.log(X['pci'])
X.drop(['pci'], axis=1, inplace=True)

Y = chicago_crime_final['theft_rate']

X = sm.add_constant(X)

reg_logpci_ump = sm.OLS(Y, X).fit()
predictions = reg_logpci_ump.predict(X)

print(reg_logpci_ump.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.870
Model:                            OLS   Adj. R-squared:                  0.870
Method:                 Least Squares   F-statistic:                 4.483e+06
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:39   Log-Likelihood:             2.3875e+06
No. Observations:             1343224   AIC:                        -4.775e+06
Df Residuals:                 1343221   BIC:                        -4.775e+06
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.6839      0.001  -1641.195      0.000      -1.686      -1.682
ump        -4.367e-05   6.97e-06     -6.261      0.000   -5.73e-05      -3e-05
log_pci        0.1866   9.33e-05   2001.117      0.000       0.186       0.187
==============================================================================
Omnibus:                    94132.517   Durbin-Watson:                   0.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           127111.344
Skew:                           0.621   Prob(JB):                         0.00
Kurtosis:                       3.855   Cond. No.                         511.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

In [90]:
```
X = chicago_crime_final[['pci', 'ave_price($/Sq.Ft)', 'med_price($)']].copy()

X['log_ave_price'] = np.log(X['ave_price($/Sq.Ft)'])
X = X[['pci', 'log_ave_price', 'med_price($)']]

Y = chicago_crime_final['theft_rate']
X = sm.add_constant(X)

reg_logave_price = sm.OLS(Y, X).fit()
predictions = reg_logave_price.predict(X)

print(reg_logave_price.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.907
Model:                            OLS   Adj. R-squared:                  0.907
Method:                 Least Squares   F-statistic:                 4.377e+06
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:40   Log-Likelihood:             2.6154e+06
No. Observations:             1343224   AIC:                        -5.231e+06
Df Residuals:                 1343220   BIC:                        -5.231e+06
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const            -0.4070      0.001   -595.532      0.000      -0.408      -0.406
pci            3.384e-06   2.25e-09   1505.330      0.000    3.38e-06    3.39e-06
log_ave_price     0.1053      0.000    700.060      0.000       0.105       0.106
med_price($)  -1.045e-07   3.26e-10   -320.316      0.000   -1.05e-07   -1.04e-07
==============================================================================
Omnibus:                    22396.870   Durbin-Watson:                   0.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            39980.573
Skew:                           0.120   Prob(JB):                         0.00
Kurtosis:                       3.810   Cond. No.                     9.48e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 9.48e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [91]:
```
X = chicago_crime_final[['pci', 'ave_price($/Sq.Ft)', 'med_price($)']].copy()

X['log_pci'] = np.log(X['pci'])

X = X[['log_pci', 'ave_price($/Sq.Ft)', 'med_price($)']]
Y = chicago_crime_final['theft_rate']


X = sm.add_constant(X)

reg_logpci_ave_med = sm.OLS(Y, X).fit()
predictions = reg_logpci_ave_med.predict(X)

print(reg_logpci_ave_med.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.919
Model:                            OLS   Adj. R-squared:                  0.919
Method:                 Least Squares   F-statistic:                 5.069e+06
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:40   Log-Likelihood:             2.7054e+06
No. Observations:             1343224   AIC:                        -5.411e+06
Df Residuals:                 1343220   BIC:                        -5.411e+06
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
=====
                           coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-----
const                  -1.0090      0.001  -1025.600      0.000      -1.011      -
1.007
log_pci                 0.1120      0.000   1054.824      0.000       0.112
0.112
ave_price($/Sq.Ft)      0.0007   7.38e-07    893.584      0.000       0.001
0.001
med_price($)       -1.626e-07   3.21e-10   -507.035      0.000   -1.63e-07   -1.6
2e-07
==============================================================================
Omnibus:                    74022.579   Durbin-Watson:                   0.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           116641.430
Skew:                           0.468   Prob(JB):                         0.00
Kurtosis:                       4.098   Cond. No.                     1.43e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.43e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [92]:

```
X = chicago_crime_final[['chicago_pop', 'arrest_boolean', 'domestic_boolean', 'p
Y = chicago_crime_final['theft_rate'].copy()

if X['arrest_boolean'].dtype.name == 'category':
    X['arrest_boolean'] = X['arrest_boolean'].cat.codes.astype(float)

if X['pci'].dtype.name == 'category':
    X['pci'] = X['pci'].cat.codes.astype(float)

X['arrest_pci_interaction'] = X['arrest_boolean'] * X['pci']

X = sm.add_constant(X)

reg_interaction = sm.OLS(Y, X).fit()
predictions = reg_interaction.predict(X)

print(reg_interaction.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             theft_rate   R-squared:                       0.876
Model:                            OLS   Adj. R-squared:                  0.876
Method:                 Least Squares   F-statistic:                 1.895e+06
Date:                Fri, 05 Apr 2024   Prob (F-statistic):               0.00
Time:                        12:58:42   Log-Likelihood:             2.4200e+06
No. Observations:             1343224   AIC:                        -4.840e+06
Df Residuals:                 1343218   BIC:                        -4.840e+06
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
=========
                              coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
---------
const                       0.0901    8.9e-05   1012.027      0.000       0.090
0.090
chicago_pop            -2.845e-07   1.48e-09   -191.819      0.000   -2.87e-07
-2.82e-07
arrest_boolean             -0.0112      0.000    -78.295      0.000      -0.011
-0.011
domestic_boolean           -0.0061      9e-05    -67.622      0.000      -0.006
-0.006
pci                     4.565e-06   1.79e-09   2551.899      0.000    4.56e-06
4.57e-06
arrest_pci_interaction  1.736e-07   3.75e-09     46.341      0.000    1.66e-07
1.81e-07
==============================================================================
Omnibus:                    51267.339   Durbin-Watson:                   0.016
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            72154.813
Skew:                           0.388   Prob(JB):                         0.00
Kurtosis:                       3.829   Cond. No.                     2.86e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.86e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

4.1.3.2 Regression Tables

Regression models for regression table 1.

$$\text{TheftRate} = \beta_0 + \beta_1 \times \text{PerCapitaIncome} + \beta_2 \times \text{UnemploymentRate} + e$$

$$\text{TheftRate} = \beta_0 + \beta_1 \times \text{PerCapitaIncome} + \beta_2 \times \text{UnemploymentRate} + \beta_3 \times \text{HighS}$$

$$\text{TheftRate} = \beta_0 + \beta_1 \times \log(\text{PerCapitaIncome}) + \beta_2 \times \text{UnemploymentRate} + e$$

In [93]:
```
stargazer = Stargazer([reg_pci_ump, reg_pci_ump_edb, reg_logpci_ump])

stargazer.custom_columns(['reg_pci_ump', 'reg_pci_ump_edb', 'reg_logpci_ump'], [
stargazer.significant_digits(4)
stargazer.rename_covariates({'const': 'Intercept', 'pci': 'Per Capita Income', '
```

```
stargazer.covariate_order(['const', 'pci', 'ump', 'edb', 'log_pci'])

display(HTML(stargazer.render_html()))
```

	Dependent variable: theft_rate		
	reg_pci_ump	reg_pci_ump_edb	reg_logpci_ump
	(1)	(2)	(3)
Intercept	0.1076***	0.1270***	-1.6839***
	(0.0001)	(0.0004)	(0.0010)
Per Capita Income	0.0000***	0.0000***	
	(0.0000)	(0.0000)	
Unemployment Rate	-0.0015***	-0.0014***	-0.0000***
	(0.0000)	(0.0000)	(0.0000)
High School Graduation Rate		-0.0003***	
		(0.0000)	
log_pci			0.1866***
			(0.0001)
Observations	1343224	1343224	1343224
R^2	0.8769	0.8772	0.8697
Adjusted R^2	0.8769	0.8772	0.8697
Residual Std. Error	0.0398 (df=1343221)	0.0397 (df=1343220)	0.0409 (df=1343221)
F Statistic	4785245.1745*** (df=2; 1343221)	3197241.9338*** (df=3; 1343220)	4483374.7446*** (df=2; 1343221)
Note:			*p<0.1; **p<0.05; ***p<0.01

Regression 1 (reg_pci_ump) examines the relationship between theft rate and per capita income (PCI) and the unemployment rate. This analysis reveals a small increase in theft rates (denoted by a coefficient close to zero but positive) as per capita income increases, indicating a small effect of increasing income on theft. The statistical significance of these results is confirmed by three asterisks at the the 1% level. On the contrary, high unemployment which can indicate economic hardship indicates theft numbers have dropped slightly, which is somewhat unexpected. While the reasons could be various and we won't explore deeply in this research.

In Regression 2 (reg_pci_ump_edb), we include an additional variable: high school graduation rate. This extended model suggests that theft rates decline as more

individuals graduate from high school. Despite the small positive effect of per capita income on theft, the unemployment rate maintains the inverse relationship. All coefficients marked with three asterisks show statistical significance, although their minute size may again indicate economic insignificance.

Regression 3 (reg_logpci_ump) is varied by the PCI logarithm used. This approach shifts our focus to percentage change rather than absolute revenue growth. Although not a major concern for negative obstruction, the positive correlation between PCI records and theft rates is noteworthy. This suggests that theft rates have risen disproportionately in response to higher income percentages, highlighting the relationship between income and theft in the relevant variable and supporting our hypothesis that on nonlinear relationships.

R-square values, which indicate how well our models describe the data, are surprisingly high in all three regressions. With reg_pci_ump_edb showing very high values, these models, each with an R-square value greater than 0.8, perform well in predicting the dataset.

Furthermore, the F statistic is high and significant enough to reinforce the reliability of the models and the validity of the observed relationship between variability and elevation rates.

Overall, these three selected regressions have been compiled into a single table to identify different components of income and explore possible factors and nonlinear relationships Model reg_pci_ump shows a positive but not insignificant relationship with income a between and theft rates are obtained and an inverse relationship with unemployment rate is revealed. Model reg_pci_ump_edb introduces education as a variable, slightly increasing the R-squared value and showing a weak inverse relationship between education level and corruption rate. Model reg_logpci_ump, while similar to reg_pci_ump in variability, uses logarithmic transformation use in PCI displaying a percentage change.

In conclusion, the study shows a positive but relatively low correlation between income and theft rates, with other factors such as unemployment and education also playing a role.

Regression models for regression table 2.

$$\text{TheftRate} = \beta_0 + \beta_1 \times \text{CommunityAreaPopulation} + \beta_2 \times \text{ArrestBoolean} + e$$

$$\text{TheftRate} = \beta_0 + \beta_1 \times \text{CommunityAreaPopulation} + \beta_2 \times \text{ArrestBoolean} + \beta_3 \times \text{D}$$

In [94]:
```
stargazer = Stargazer([reg_pop_arrest, reg_pop_arrest_domestic, reg_interaction]

stargazer.custom_columns(['reg_pop_arrest', 'reg_pop_arrest_domestic', 'reg_inte
stargazer.significant_digits(4)
stargazer.rename_covariates({'const': 'Intercept', 'chicago_pop': 'Community Are
stargazer.covariate_order(['const', 'chicago_pop', 'arrest_boolean', 'domestic_b
```

```
display(HTML(stargazer.render_html()))
```

	Dependent variable: theft_rate		
	reg_pop_arrest	reg_pop_arrest_domestic	reg_interaction
	(1)	(2)	(3)
Intercept	0.1686***	0.1793***	0.0901***
	(0.0002)	(0.0002)	(0.0001)
Community Area Population	0.0000***	0.0000***	-0.0000***
	(0.0000)	(0.0000)	(0.0000)
Arrest Boolean	-0.0291***	-0.0309***	-0.0112***
	(0.0002)	(0.0002)	(0.0001)
Domestic Boolean		-0.0432***	-0.0061***
		(0.0002)	(0.0001)
Arrest*PCI			0.0000***
			(0.0000)
Observations	1343224	1343224	1343224
R^2	0.1203	0.1421	0.8759
Adjusted R^2	0.1203	0.1421	0.8759
Residual Std. Error	0.1063 (df=1343221)	0.1050 (df=1343220)	0.0399 (df=1343218)
F Statistic	91859.9549*** (df=2; 1343221)	74181.8081*** (df=3; 1343220)	1895499.9971*** (df=5; 1343218)
Note:			*p<0.1; **p<0.05; ***p<0.01

In Regression 1 (reg_pop_arrest), we have two predictors: the population of the community area and an arrest boolean. The coefficient of population is extremely small, but it's positive, meaning there is a slight increase in theft rates as the population increases. However, given the size of the coefficient, this increase is likely very minimal. The negative coefficient of arrest is a bit larger, suggesting that in areas with more arrests, the theft rate goes down. This could be due to the deterrent effect of policing or successful crime prevention efforts.

In Regression 2 (reg_pop_arrest_domestic), we add a third variable domestic. This has a negative coefficient as well, indicating that theft rates are lower in areas with more domestic incidents. It suggests that demographic is a potential factor of crime rate while we don't dig into it in this paper.

In Regression 3 (reg_interaction), we add a interactive variable of arrest times pci. Theft rate decreases with arrests and when thefts are domestic, but population size and the interaction between arrests and another variable (PCI) have no practical impact despite being statistically significant.

Looking at the Adjusted R-squared values, they are relatively low for the first two models while high for regression 3. This tells us that, while the regression 1 and 2 are statistically significant (as we can see from the F Statistic), they don't explain a very large portion of the variation in theft rates. As for regression 3, a new interactive term increases the adjusted R-sqaured sigificantly, indicating that we may need to add more variables to improve the models or we need to consider other regression types such as including more non-linear regression terms.

The F Statistics are high and have very significant p-values, so we can be confident that the relationships we've found are statistically significant and not just due to random chance.

Generally, we could conlude that in places with more people, theft rates are higher. However, more arrests seem to bring the theft rates down a bit, which makes sense if thieves are being caught. These findings may indicate that while economic conditions like income and unemployment have certain effects on non-violent crime rates, other factors such as population density and the efficacy of law enforcement also play crucial roles.

Regression models for regression table 3.

$$\text{TheftRate} = \beta_0 + \beta_1 \times \text{AverageHousePrice} + \beta_2 \times \text{PCI} + e$$

$$\text{TheftRate} = \beta_0 + \beta_1 \times \text{AverageHousePrice} + \beta_2 \times \text{MedianHouseListingPrice} + \beta_3 \times$$

$$\text{TheftRate} = \beta_0 + \beta_1 \times \log(\text{AverageHousePrice}) + \beta_2 \times \text{PCI} + e$$

$$\text{TheftRate} = \beta_0 + \beta_1 \times \text{MedianHouseListingPrice} + \beta_2 \times \log(\text{PCI}) + e$$

In [95]:
```
stargazer = Stargazer([reg_ave_price, reg_ave_med_price, reg_logave_price, reg_l

stargazer.custom_columns(['reg_ave_price','reg_ave_med_price', 'reg_logave_price
stargazer.significant_digits(4)
stargazer.rename_covariates({'const': 'Intercept', 'pci': 'pci', 'ave_price($/Sq
stargazer.covariate_order(['const', 'ave_price($/Sq.Ft)', 'med_price($)', 'log_a

display(HTML(stargazer.render_html()))
```

				Dependent variable: theft_rate
	reg_ave_price	reg_ave_med_price	reg_logave_price	reg_logpci_ave_med
	(1)	(2)	(3)	(4)
Intercept	0.0431***	0.0480***	-0.4070***	-1.0090***
	(0.0001)	(0.0001)	(0.0007)	(0.0010)
Average House Price ($/Sq.Ft)	0.0004***	0.0006***		0.0007***
	(0.0000)	(0.0000)		(0.0000)
Median House Listing Price ($)		-0.0000***	-0.0000***	-0.0000***
		(0.0000)	(0.0000)	(0.0000)
Logged Average House Price			0.1053***	
			(0.0002)	
pci	0.0000***	0.0000***	0.0000***	
	(0.0000)	(0.0000)	(0.0000)	
log_pci				0.1120***
				(0.0001)
Observations	1343224	1343224	1343224	1343224
R^2	0.9010	0.9118	0.9072	0.9188
Adjusted R^2	0.9010	0.9118	0.9072	0.9188
Residual Std. Error	0.0357 (df=1343221)	0.0337 (df=1343220)	0.0345 (df=1343220)	0.0323 (df=1343220)
F Statistic	6110091.6711*** (df=2; 1343221)	4626582.4853*** (df=3; 1343220)	4377403.8495*** (df=3; 1343220)	5068870.9573*** (df=3; 1343220)
Note:				*p<0.1; **p<0.05; ***p<0.01

All models show a statistically significant relationship with three asterisks (***), indicating a p-value of less than 0.01, which means these findings are highly unlikely to be due to chance. We include pci for the first three regressions to control fixed effect.

Regression 1 (reg_ave_price): This model uses the average house price per square foot as a predictor for theft rates. There's a positive relationship, indicating that as the average price per square foot increases, the theft rate increases slightly.

Regression 2 (reg_ave_med_price): This regression adds the median house listing price as another predictor. The coefficients for both the average price per square foot and median listing price are positive, suggesting that as house prices increase, theft rate also increases. However, the coefficient for median house listing price is almost zero, indicating economically insignifiance.

Regression 3 (reg_logave_price): Here, a logged average house price is used. The positive coefficient for the logged average house price shows that when the average house price increases by a certain percentage, the theft rate also increases significantly.

Regression 4 (reg_logpci_ave_med): The fourth model includes both the logged per capita income (log_pci) and the median house listing price. The negative coefficient for median listing price has effectively no change, but the logged per capita income has a strong positive relationship with theft rates, much like in the third model.

The R-squared and Adjusted R-squared values are very high in all models, especially considering social science data, indicating that a substantial portion of the variance in theft rates is explained by these models.

The F Statistics are extremely high and statistically significant across all models, supporting the reliability of these models.

In conclusion, the analysis is showing that there's a consistent relationship between housing prices and theft rates. Higher housing prices are related with higher theft rates, and this effect becomes much more noticeable when looking at percentage changes in prices (as with the logged values). The findings could suggest that areas with higher housing values may attract more thefts, which is consistent with our assumption of community areas with higher wealth level tends to have higher theft rate.

4.1.3.3 Preferred Specification

The regression model 'reg_ave_med_price' is identified as the preferred specification, despite not having the highest R-squared value among the models. Its R-squared value, exceeding 0.91, is exceptionally high, indicative of a strong explanatory power. In comparison to the 'reg_logpci_ave_med' model, 'reg_ave_med_price' offers a slight improvement in R-squared value. To maintain simplicity in our model, opting for 'reg_ave_med_price' without applying logarithmic transformation is a strategic choice.

This model includes three key variables that are central to our analysis: per capita income, average house price, and median house listing price. Collectively, these variables represent both income and wealth levels, effectively capturing the economic disparities across different community areas. Notably, the coefficient of average house price stands out as the largest across all ten regressions. This observation leads us to hypothesize that average house price has the most significant impact on theft rate, suggesting that wealth or property value is a predominant factor influencing non-violent crime.

This assumption, along with the model's implications, will be further explored in subsequent sections focusing on feature importance. Additionally, the

'reg_ave_med_price' model will serve as a reference point for comparison with our machine learning analyses, providing a comprehensive understanding of the factors influencing theft rates in different economic contexts.

4.1.3.4 Regressions Evaluation

Based on the P-values, adjusted R-squared values, and F-statistics, our regression models demonstrate relatively high predictive accuracy and reliability. While it's possible to enhance these models further by adding more terms, applying additional transformations, and creating interaction variables, the current models strike a balance between simplicity and informative value, which aligns with the objectives of this research paper. Maintaining model simplicity is essential to ensure clarity and accessibility, and the current level of accuracy is deemed sufficiently robust for the purposes of our analysis.

Moreover, I can use measures like Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to assess the performance of my regressions. The AIC is a measure of the fitness of a statistical model for a given set of data, which requires a trade-off between of model fitness and model robustness. A lower AIC value indicates a better model. When comparing multiple models, the one with the lowest AIC is generally preferred. The BIC is similar to the AIC but includes higher penalties for models with more parameters. Like the AIC, a low BIC value indicates a good model. BIC is particularly useful for comparing models at different prediction rates and is most likely to penalize complex models and avoid overfitting.

4.1.3.5 Regression Results

The three regression tables, encompassing ten regression models, collectively provide statistical evidence supporting our hypothesis that community areas with superior economic conditions (in terms of both income and wealth levels) are associated with higher rates of non-violent crimes, particularly theft. Regression Table 1 highlights a positive correlation between per capita income and the theft rate, accounting for potential confounding factors like unemployment rate and education level. Additionally, a logarithmic transformation of per capita income is employed to assess the non-linear dynamics of this relationship.

Regression Table 2 delineates a positive correlation between population size and theft rate, alongside a negative correlation between the rates of arrest, domestic crimes, and theft. Notably, the accuracy of the first two models is relatively low. However, the introduction of an interaction term between income level and arrest rate in Model 3 markedly improves accuracy. This finding aligns with our assumption that areas with higher income levels may have better surveillance and, consequently, higher reported crime rates. Although the value of this relationship is minimal, it is considered a relevant factor in our analysis, albeit not a primary focus.

In Regression Table 3, we identify the most economically significant variable and the best-fitting model: average house price. These models establish a positive correlation

between house price and theft rate, taking into account both average and median house listing prices and applying a logarithmic transformation to per capita income.

In summary, the collective results of these regression analyses substantiate our research premise that community areas with elevated income and wealth levels are more likely to experience higher rates of non-violent crimes, specifically theft.

4.2 Machine Learning

4.2.1 Objective Function

$$\min_{j,s} \left[\sum_{i:x_{ij} \leq s, x_i \in R_1} (\text{TheftRate}_i - \hat{\text{TheftRate}}_{R1})^2 + \sum_{i:x_{ij} > s, x_i \in R_2} (\text{TheftRate}_i - \hat{\text{TheftRate}}_{R}$$

In our endeavor to construct a regression tree using machine learning techniques, we employ the objective function above with the goal of minimizing residual errors. This regression tree is designed to unearth the relationship between economic disparities and the prevalence of various types of crime within Chicago's community areas. In essence, this allows us to not only explore the connection but also to forecast the crime rate for different crime categories based on the economic conditions prevailing in these areas. The significance of this analysis is to furnish policymakers with crucial insights that can aid in mitigating high crime rates in specific areas. This could lead to more informed decisions regarding policy resource allocation, urban planning, and similar strategies.

The primary objective of regression trees is to organize the data into subsets that exhibit as much homogeneity as possible. Our function specifically aims to identify the optimal values for 'j' and 's' that minimize the sum of the squared differences between actual and predicted theft rates across the defined subsets. Here, 's' acts as a threshold to divide the data into these subsets. For each data point, we calculate the squared difference between the actual theft rate (TheftRate_i) and the predicted theft rate for a given region (TheftRate and TheftRateR2R1). By applying our objective function, we facilitate a comparison between our regression models and those produced by the regression tree, enhancing our understanding and predictive capabilities in this area.

4.2.2 Regularization Parameters

Machine learning models often grapple with the challenge of balancing fitness and avoiding overfitting, a dilemma that can be mitigated by tweaking the regularization parameters. One such parameter is tree depth, which indicates the number of splits between the root and leaf of a tree. A deeper tree has the capacity to capture more features, but this increased complexity also heightens the risk of overfitting. To counter this, reducing the tree depth can simplify the model, though it's crucial to find an optimal depth that maintains accuracy.

Beyond directly adjusting tree depth, the model's complexity can also be managed by altering the minimum number of samples required for a split or to define a leaf. For example, increasing the minimum samples for a split means the tree will not create a new split unless there are enough samples available, and similarly for the minimum samples required for a leaf. These regularization parameters are key to reducing model overfitting. By selecting the appropriate parameters, we can strike a delicate balance between model variance and bias, thereby enhancing the model's predictive performance and reliability.

4.2.3 Regression Tree

In [97]:
```
df = chicago_crime_final

X = df[['arrest_boolean', 'domestic_boolean', 'ump', 'pci', 'edb', 'chicago_pop'
y = df['theft_rate']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_

scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

regressor = DecisionTreeRegressor(random_state=42)
regressor.fit(X_train, y_train)

y_pred = regressor.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
```

```
Mean Squared Error: 5.774616709174158e-27
```

The mean sqaured error is extremely low, hence the model has a good fitness while might existing an overfitting issue.

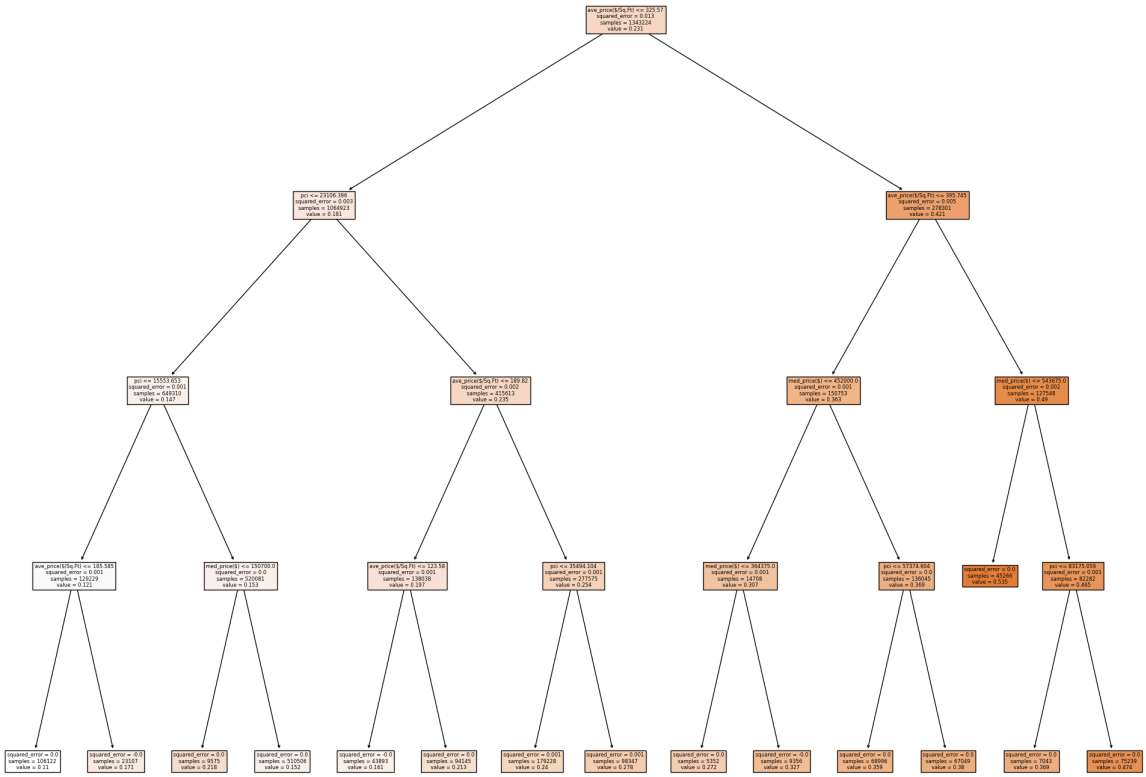
In [118…
```
sqft_tree = tree.DecisionTreeRegressor(max_depth=4).fit(X,y)
```

In [129…
```
y_pred_tree = sqft_tree.predict(X)
tree_mse = metrics.mean_squared_error(y, y_pred_tree)

print('Mean Squared Error:', metrics.mean_squared_error(y, y_pred_tree))
```

```
Mean Squared Error: 0.00035426373898338653
```

The tree depth was set to 4, which generated a much higher MSE than above while is still significantly low.

In [115…
```
sqrf_fig = plt.figure(figsize=(25,20))
sqrf_fig = tree.plot_tree(sqft_tree, feature_names=X.columns, filled=True)
```



The decision tree is partitioned based on various features and corresponding thresholds. Within this decision tree, it is observed that the squared errors are significantly lower compared to those in our regression model, suggesting a overfitting. We have fine-tuned the tree's depth, selecting 4 as the optimal level to balance between model accuracy and complexity. A depth of less than 4 results in increased residual errors, whereas a depth greater than 4 tends to include an excessive number of features, which adds little value to our model.

The root node is initiated with the feature "ave_price($/sq.ft.) < 325.57," deemed the most crucial for our model. If the data satisfy this threshold, the process advances to the left branch; if not, it moves to the right. The left branch is further divided based on per capita income, subdividing the dataset anew. The recurrence of average price as a feature in various branches underscores its significance in predicting theft rates. In contrast, the right branch identifies the feature median price as a new threshold. Per capita income, average price, and median price emerge as the three most frequently occurring features in the decision tree, highlighting their substantial influence on theft rate predictions. This aligns with our initial assumptions. Other variables such as population and unemployment rates are less prevalent, suggesting they have a minimal impact on our theft rate forecasts.

In summary, the decision tree offers a more precise modeling approach compared to our regression models, effectively identifying the most pertinent features within our dataset.

4.2.4 Random Forest

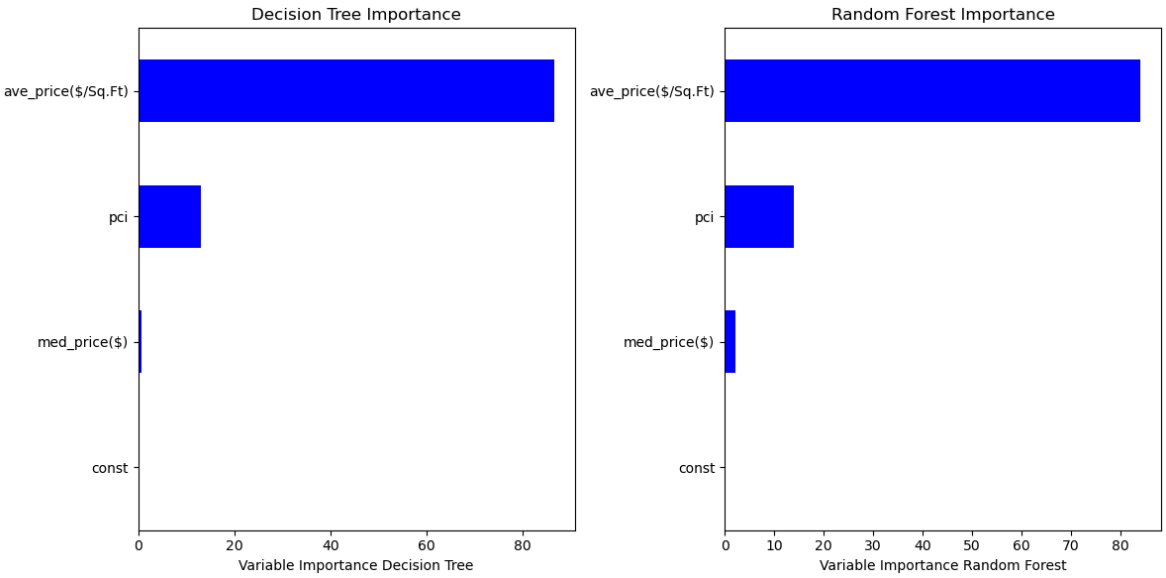
In [130…
```
regr2 = RandomForestRegressor(max_features=5, random_state=1)
regr2.fit(X, y)
pred = regr2.predict(X)

rf_mse = mean_squared_error(y, pred)
rf_mse
```

Out[130… 3.938496478824122e-27

The MSE of random forest is extremely lower than decission tree, while we need to consider the overfitting problem. When comparing the performance of a Random Forest to a single decision tree, it's important to consider a few factors. A decision tree is quite straightforward and easy to interpret because we can see exactly how the inputs are split at each node. However, decision trees can easily become very complex and tend to overfit the training data if not properly constrained, which can lead to poor performance on unseen data. In this scenario, I would choose random forest as a better preditor as its extremely low MSE, indicating almost perfectly predicting the data. There could be potential overfitting problem in the random forest, while we have set the max features to 5, limiting the complexity of the model.

4.2.5 Importance Matrix

In [127…
```
plt.figure(figsize=(12, 6))

# Plot for the decision tree
plt.subplot(1, 2, 1)
Importance = pd.DataFrame({'Importance': sqft_tree.feature_importances_ * 100},
Importance.sort_values('Importance', axis=0, ascending=True).plot(kind='barh', c
plt.xlabel('Variable Importance Decision Tree')
plt.title('Decision Tree Importance')
plt.gca().legend_ = None

# Second Importance plot for Random Forest
plt.subplot(1, 2, 2)
Importance = pd.DataFrame({'Importance': regr2.feature_importances_ * 100}, inde
Importance.sort_values('Importance', axis=0, ascending=True).plot(kind='barh', c
plt.xlabel('Variable Importance Random Forest')
plt.title('Random Forest Importance')
plt.gca().legend_ = None

plt.tight_layout()
plt.show()
```



The importance matrices are presented above, featuring the decision tree on the left and the random forest on the right. In both models, the predominant features are average price, per capita income, and median price, with average price taking precedence, accounting for more than 80% of the importance. This significant majority suggests a possible direct link between the cost of living, as reflected by property prices, and theft incidence. The fact that average price has a more substantial impact on theft rates than per capita income suggests that property values have a considerably greater effect on theft occurrences than income levels. This could indicate that individuals might resort to theft as a means to cope with unaffordable housing costs. Although per capita income is an important factor, its significance is markedly less than that of average price, constituting approximately 18% in both models. This distinction underscores that while personal wealth plays a critical role, it does not influence theft rates as significantly as property prices. This could imply that even in contexts of higher income levels, the cost of property might pose a more acute economic challenge that individuals face. Meanwhile, median price exerts a minimal effect in both models, with a slightly higher impact observed in the random forest. These insights could serve as a basis for policy intervention. Policymakers might consider implementing measures such as providing economic assistance to help individuals afford housing or enacting regulations to cap rental fees.

4.2.6 Compare Results

In [139…

```
linreg = LinearRegression()
reg_tree = DecisionTreeRegressor(random_state=42)
rf = RandomForestRegressor(random_state=42)

# Fit the models
linreg.fit(X_train, y_train)
reg_tree.fit(X_train, y_train)
rf.fit(X_train, y_train)
```

Out[139…

```
RandomForestRegressor(random_state=42)
```

In [143…

```
linreg_mse = mean_squared_error(y_test, linreg.predict(X_test))
tree_mse = mean_squared_error(y_test, reg_tree.predict(X_test))
rf_mse = mean_squared_error(y_test, rf.predict(X_test))

# Create the performance table
performance_table = {
    'Model': ['Linear Regression', 'Decission Tree', 'Random Forest'],
    'MSE': [linreg_mse, tree_mse, rf_mse],
    'Score (%)': [round(linreg.score(X_test, y_test)*100, 4),
                  round(reg_tree.score(X_test, y_test)*100, 4),
                  round(rf.score(X_test, y_test)*100, 4)]
}

performance_table = pd.DataFrame(performance_table)
performance_table
```

Out[143…

	Model	MSE	Score (%)
0	Linear Regression	9.565656e-04	92.5547
1	Decission Tree	5.774617e-27	100.0000
2	Random Forest	1.027092e-27	100.0000

In the table above, we dive into three analysis methods: Linear Regression, Decision Tree, and Random Forest, evaluating them based on their Mean Squared Error (MSE) and performance score. The Linear Regression method has relatively high MSE and comparatively lower performance score, though it still achieves an impressive 92% score. Although Linear Regression offers valuable insights into how economic factors generally influence theft rates, it may not fully capture all the factors. On the other hand, both the Decision Tree and Random Forest methods show perfect scores of 100%, indicating exceptional fit with the data. Decision Tree shows a slightly higher MSE than the Random Forest. This near-perfect performance raises concerns about overfitting, which means that these models might be overfitted to the sample to predict future outcomes accurately outside of it.

Focusing on Linear Regression, this model employs the OLS method. Its higher MSE, compared to the more complex machine learning models, could be attributed to its simplicity and the assumption that economic conditions affect theft rates uniformly across different scenarios. In contrast, the machine learning models, including Decision Tree and Random Forest, do not rely on such fixed assumptions. Instead, they construct a complex set of conditions to make predictions, which allows them to adapt to various data subsets.

From an economic perspective, Linear Regression gives us a broad understanding of how changes in economic conditions might influence theft rates, suggesting a linear relationship where theft rates adjust consistently with economic changes, all else being constant. Conversely, Decision Tree and Random Forest models provide more nuanced

insights, identifying specific conditions under which economic improvements could lead to reduced theft rates, or conversely, where they might not have a straightforward effect. These models provide insights to policymakers and the government to further take measures to mitigate the crime rate problems across community areas in Chicago.

4.3 Conclusion

The analysis of crime across Chicago's community areas has revealed how crime type is influenced by a mix of socioeconomic, demographic, and environmental factors, highlighting the complex nature of urban crime in Chicago. By examining crime statistics through different types of charts and maps, we've seen variations in crime rates across areas and the types of crimes that are most common, such as theft and battery. Variations between areas with different social-economic conditions suggest that the city and the environment design play a big role in crime occurrence.

The data also shows the effectiveness of law enforcement in making arrests and the differences between domestic and non-domestic crimes as well as violent and non-violent crimes, pointing to the need for varied strategies in combating crime. The results indicate that factors like population density, urban design and other socioeconomic factors are all important in understanding the high frequency of crimes in certain areas. The community areas with higher income levels tend to be attractive for theft crimes while community areas with high unemployment aggregate violent crimes such as battery. Additionally, the arrest rate of battery is overall higher than theft, which indicates the essential of violent and non-violent crimes, whereas the governments and the policymakers should not ignore the high rate of non-violent crimes like theft, since the low arrest rate could be the obstruct to mitigate the crime rate. This means that the potential mitigation of crime in Chicago should encompass various aspects: improving economic conditions, engaging communities, redesigning urban spaces for better safety, and using data to guide policing. By focusing on the root causes of crime and adapting our strategies to each neighbourhood's unique situation, we can work towards making all parts of Chicago safer.

Wealthier community areas usually have more expensive assets, which attract criminals who want to steal, since thieves may pay more attention to the areas where they are more likely to get higher-value stolen goods. Additionally, poorer areas may experience higher levels of socioeconomic stress, such as unemployment and lower educational opportunities, which are all considered factors that promote violent crime. Furthermore, wealthy community areas may have more safeguards that might reduce violent crimes, while non-violent crimes like theft are harder to find evidence for, and the time elapsed also makes non-violent crimes harder to result in arrests. The insights from this research enhance our understanding of crime in Chicago; they also provide practical guidance for those looking to make real changes, from policymakers to urban planners and police.

Drawing on insights from regression models and machine learning analyses, we've come to understand the significant impact of various factors on crime types and rates. Among these, average house price emerges as the paramount determinant, exerting a far greater

influence than income levels. This finding underscores the preeminence of static economic factors over dynamic ones in affecting crime rates. Notably, high rental costs signal not only a community's robust purchasing power but also a dilemma for residents unable to afford housing, potentially driving them toward theft. Beyond economic indicators, the models identify additional variables such as unemployment rates, education levels, arrest rates, and population size. Yet, these factors appear to exert minimal impact on crime rates.

From this analysis, it is evident that economic disparities significantly shape crime dynamics across Chicago's communities, with areas boasting higher average house prices also experiencing elevated rates of non-violent crimes, such as theft. This insight offers policymakers and governmental bodies a foundation for crafting interventions aimed at mitigating these disparities. Strategies might include offering rental subsidies, imposing rental fee caps, and reallocating police resources to address the underlying issues more effectively

By using a comprehensive approach that considers all the different factors that affect crime, we can make strides in reducing crime and improving life in Chicago. While the research still has some limitations. Firstly, there are more crime types in non-violence and violence while we only use that as the representation of non-violence and battery as the representation of violence. Although it is the rationale for the specific situation in Chicago with the data supported, when it comes to other cities, the conclusion might not be precise. Secondly, the regression models may not capture all the potential variables, since our research is focusing on the economic factors, we only include some social and demographic variables such as unemployment rate, education level, population, etc. There are some unseen features, encompassing black-and-white percentages, price level, etc. These variables also may impact the crime rate significantly but we didn't include them, so we have to include such a dataset in the research to further improve the accuracy. Thirdly, the relationship of regression models may not fit the patterns. Although we applied both linear and non-linear regression to our dataset, there could be a more complex correlation between them. In further research, we may try more regressions based on the machine learning result or create more features such as the square of income.

Even with these limitations, the study still provides comprehensive insights to policymakers and the government to further improve the crime problems across community areas in Chicago. The community areas with high-income levels and house prices tend to experience more theft incidents than battery; areas with lower income per capita and the average house price report more battery incidents than theft. Moreover, the essential factor that causes crime is house price, which suggests that economic inequality and high living costs promote non-violence crimes. To address these problems, the government could provide subsidies for the rental fees and set limitations to it. For the areas with high income, they need to improve the super violence and increase more police sources to improve the low arrest rate.

4.4 Next Steps

This research focuses on the specific social phenomena in Chicago based on the community area level, hence we want to apply the research to a broader case such as the whole of America. To achieve this goal, we need to include more potential factors in our regression models as well as think about regression correlations with higher fitness. We will merge more datasets into our research to explore all the features impacting our conclusion, such as the hardship index. The high fitness of the machine learning models may cause the overfitting problem, while its extremely low MSE provides insights to scholars to delve into the models and improve them to increase the accuracy. In this case, this research is open for advancements and improvements.