Chicago Crime Prevention

Code

AUTHOR
Alex & Nelly

PUBLISHED
December 7, 2024

Partner 1 (name and cnet ID): Suyuan Fang - suyuanfang Partner 2 (name and cnet ID): Jiaxuan nie - Jnie21

Executive Summary

This report analyzes Chicago's crime trends, with an emphasis on the correlation between crime categories, geographic locations, and seasonal fluctuations. The analysis identifies critical factors that influence crime patterns by utilizing data from the City of Chicago's public records. Actionable strategies for reducing crime rates have been informed by a combination of spatial and temporal analysis techniques. The following are suggested: the implementation of targeted seasonal crime prevention programs, the expansion of welfare initiatives, the enhancement of urban planning, and the expansion of employment opportunities.

Intro

Chicago Crime rate have been a persistent issue ove the past decade, and just recently, there have been dramatic movement in the police department that puts people in concern of the stabiltiy in safety. #Mayor Brandon Johnson, who was sworn in earlier this year, was one of the many to advocate for defunding the police in 2020 after Floyd was murdered by a Minneapolis police officer, saying "I don't look at it as a slogan" and adding, "It's an actual political goal," during a podcastThe city reduced the police department by 614 positions and cut funding by 2.7% in 2021 in the immediate aftermath of Floyd's murder. Since then, the department has grown in size and spending.

While the city maintained staffing levels for 2024 at the same level as the previous year, the city reported a 30% increase in crime in 2022. Murders were down 13% from 804 in 2021 to 699 in 2022, but the city stated property crime drove the spike in crime and was up 44% and violent crime was up 1% in 2022 as compared to 2021.

tools and information they would present could help broach discussion of the causes and solutions to the city's deadly crime problem, and perhaps even help to answer that oft-asked question that she—and likely many other Chicagoans—field when they travel: "'Is it really as bad as all the news says it is?'" # Research question What are the key factors influencing crime patterns in terms of type, location, and seasonality over time, and how can this information inform targeted crime prevention strategies?

Data

▶ Code

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4145424 entries, 0 to 4145423
Data columns (total 23 columns):
 #   Column                Dtype
---  ------                -----
 0   Unnamed: 0            int64
 1   ID                    int64
 2   Case Number           object
```

```
 3   Date                  object
 4   Block                 object
 5   IUCR                  object
 6   Primary Type          object
 7   Description           object
 8   Location Description  object
 9   Arrest                bool
 10  Domestic              bool
 11  Beat                  int64
 12  District              float64
 13  Ward                  float64
 14  Community Area        float64
 15  FBI Code              object
 16  X Coordinate          float64
 17  Y Coordinate          float64
 18  Year                  int64
 19  Updated On            object
 20  Latitude              float64
 21  Longitude             float64
 22  Location              object
dtypes: bool(2), float64(7), int64(4), object(10)
memory usage: 672.1+ MB
```

The analysis employs data from the City of Chicago's open data portal. The research was centered on the identification of spatial concentrations of crimes and their correlations with neighborhood demographics following the extraction, cleansing, and processing of data. Geographic visualization techniques were employed to map crime density and locations, while temporal analyses were conducted to investigate year-over-year changes and seasonality. Ultimately, an interactive Shiny App was launched to facilitate the dynamic exploration of crime trends, thereby assisting policymakers and stakeholders in visualizing the findings and customizing their interventions.

Data Cleaning

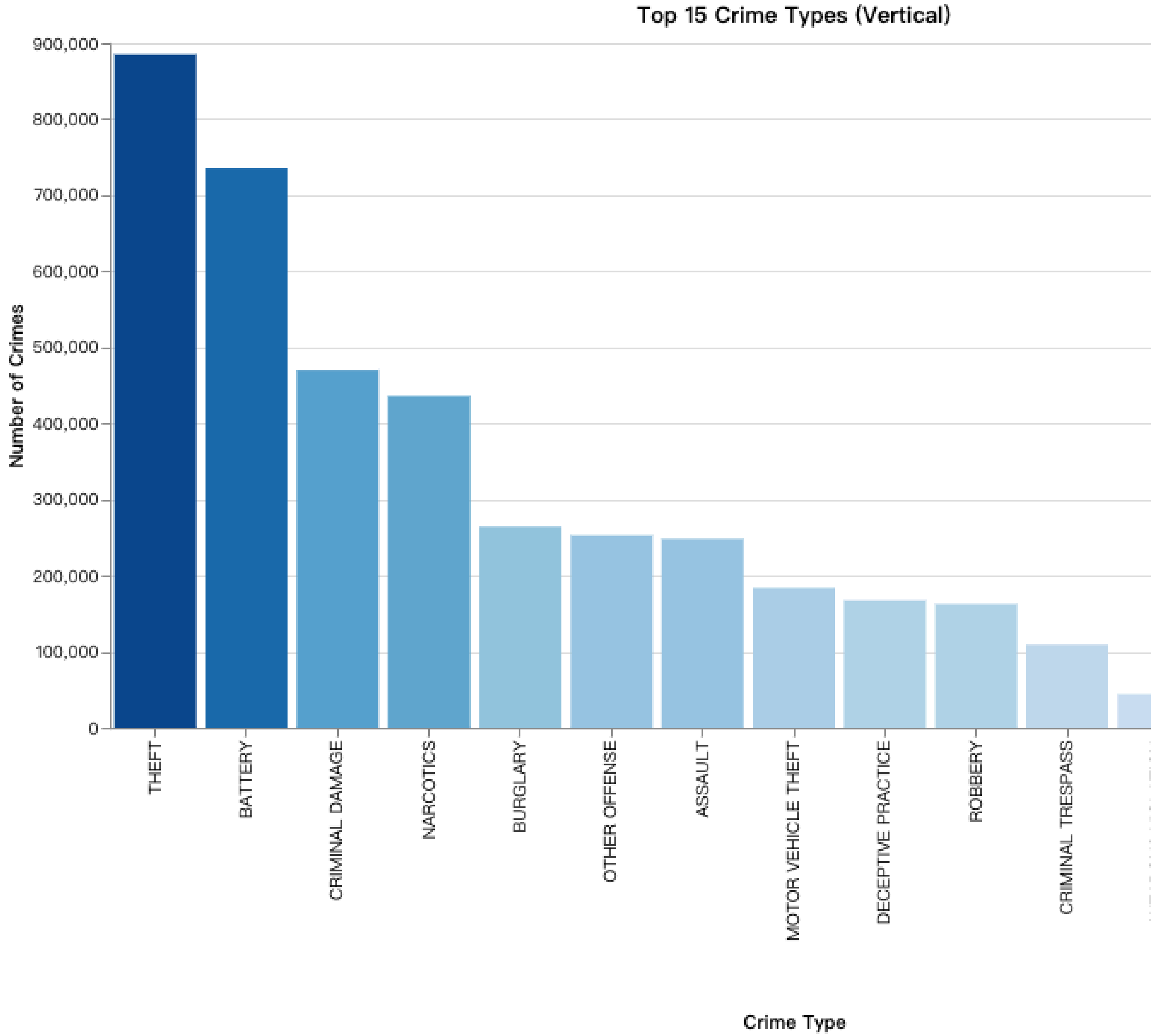
▶ Code

```
                      Missing Values  Percentage
Location                       65728    1.585556
Longitude                      65728    1.585556
Latitude                       65728    1.585556
Y Coordinate                   65728    1.585556
X Coordinate                   65728    1.585556
Location Description            1949    0.047016
Community Area                  1495    0.036064
District                          84    0.002026
Ward                              77    0.001857
Case Number                        7    0.000169
Updated On                         0    0.000000
Year                               0    0.000000
FBI Code                           0    0.000000
Unnamed: 0                         0    0.000000
ID                                 0    0.000000
Domestic                           0    0.000000
Arrest                             0    0.000000
```

```
Description                      0    0.000000
Primary Type                     0    0.000000
IUCR                             0    0.000000
Block                            0    0.000000
Date                             0    0.000000
Beat                             0    0.000000
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 4134067 entries, 2008-10-07 12:39:00 to 2016-05-03 23:38:00
Data columns (total 10 columns):
 #   Column                Dtype  
---  ------                -----  
 0   Block                 object 
 1   Primary Type          object 
 2   Description           object 
 3   Location Description  object 
 4   Arrest                bool   
 5   Domestic              bool   
 6   Year                  int64  
 7   Latitude              float64
 8   Longitude             float64
 9   Location              object 
dtypes: bool(2), float64(2), int64(1), object(5)
memory usage: 291.7+ MB
None
```

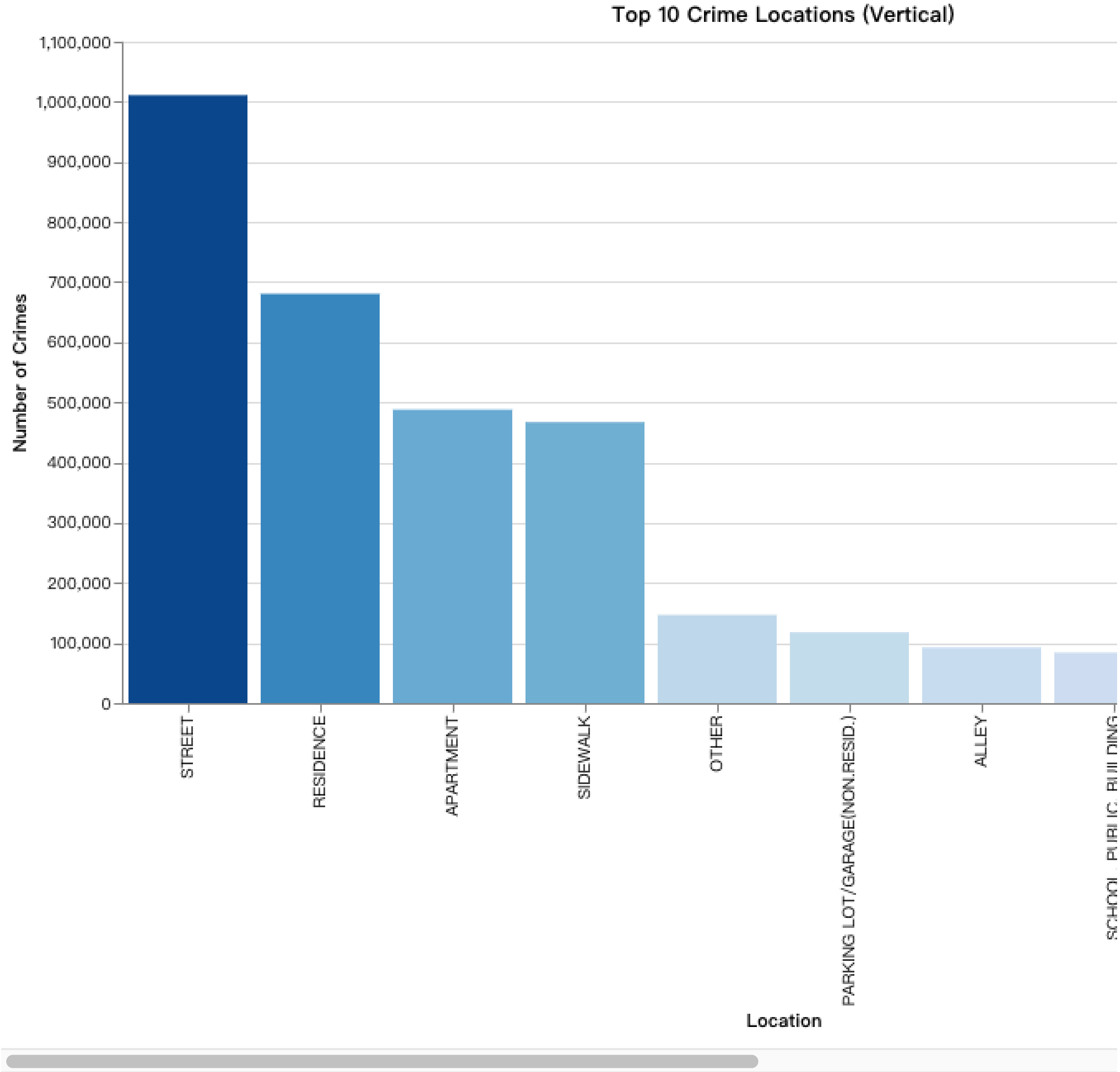
Data Visualization

▶ Code



We utilizes Altair, a visualization library, to create an interactive bar chart that displays the top 15 crime types in Chicago. First, the Altair renderer is enabled using alt.renderers.enable(“default”) to ensure proper visualization within the Jupyter Notebook, and alt.data_transformers.disable_max_rows() removes any row limitations to handle larger datasets. The data preparation step involves calculating the frequency of each crime type using value_counts() on the Primary Type column of the dataset analysis_data, followed by resetting the index and renaming the columns to Crime Type and Count. This creates a summarized DataFrame with crime types and their respective counts. The Altair chart is constructed using alt.Chart() to initialize the data, .mark_bar() to specify a bar chart, and .encode() to configure the axes, bar colors, and tooltip interactivity. The x-axis displays crime types sorted in descending order, while the y-axis shows their corresponding counts. Additional properties, such as chart title, width, and height, are customized using .properties(). Finally, the chart is displayed with chart.show(), rendering a visually engaging and interactive bar chart titled “Top 15 Crime Types (Vertical),” where theft emerges as the most frequent crime type.

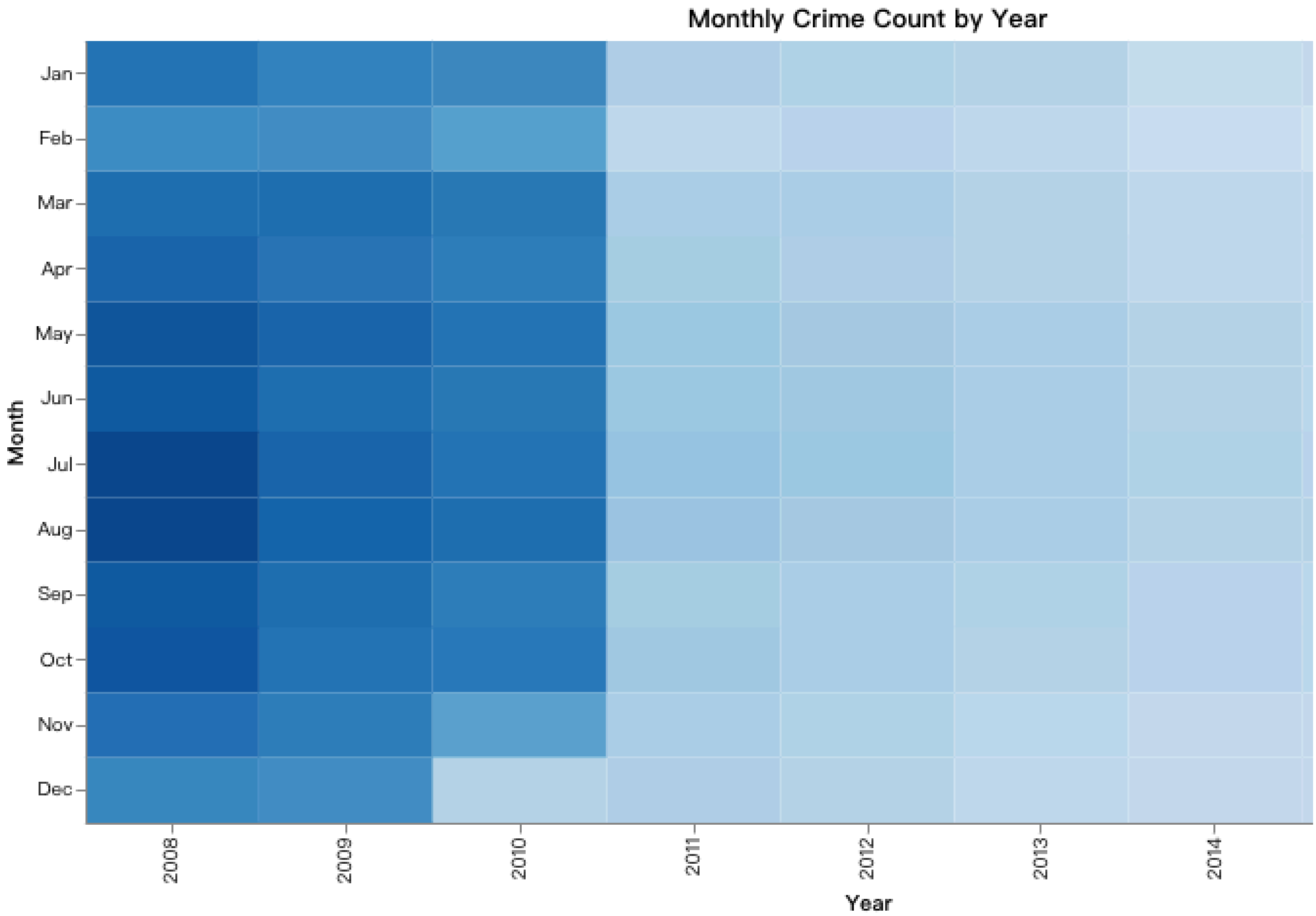
▶ Code



Then generate a bar chart showcasing the top 10 locations where crimes occur, revealing that "STREET" is the most common location for crimes. The data preparation begins by calculating the frequency of each location using the value_counts() method on the Location Description column of the analysis_data dataset, which is then converted into a DataFrame using reset_index(). The columns are renamed to Location Description and Count for clarity. The Altair chart is constructed by passing the top 10 locations (head(10)) to alt.Chart() and specifying a bar chart using .mark_bar(). The encode() function is used to configure the x-axis for location descriptions (nominal data sorted by count in descending order), the y-axis for crime counts, and the bar colors based on count values using a blue color scheme. Tooltips are added for interactivity, allowing users to hover over bars to view detailed information about each location and its corresponding count. The chart's appearance is further customized with a title, width, and height using .properties(). Finally, the chart is displayed with location_chart.show(), rendering a clear visualization where "STREET" is identified as the location with the highest number of crimes.

Heatmap for Monthly Crime Counts by Year

▶ Code



For the heatmap, I first organized the data by resampling it to get monthly crime counts. I extracted the year and month from the data and converted numeric months into descriptive names like "Jan" for January. Using this processed data, I generated a heatmap where the x-axis represents the year, the y-axis represents the months (ordered from January to December), and the color intensity indicates the number of crimes in each month. Darker shades represent higher crime counts, and tooltips allow users to hover over each cell to view details. This visualization highlights trends, such as higher crime rates from 2008 to 2010 and the lowest average monthly crime rate in 2015.

Geospatial Map

▼ Code

```
import altair as alt
import pandas as pd
import geopandas as gpd
alt.data_transformers.disable_max_rows()

chicago_boundary = gpd.read_file("/Users/suyuanfang/Desktop/Pyhton/final project/trans

chicago_boundary = alt.Data(values=chicago_boundary.__geo_interface__["features"])

map_marks = analysis_data[['Latitude', 'Longitude', 'Year']].dropna()
map_marks = map_marks[
    (map_marks['Latitude'] >= -90) & (map_marks['Latitude'] <= 90) &
    (map_marks['Longitude'] >= -180) & (map_marks['Longitude'] <= 180)
]
```

```
map_marks_sampled = map_marks.groupby('Year').apply(lambda x: x.sample(min(len(x), 200

selector = alt.selection_single(
    fields=['Year'],
    bind=alt.binding_select(options=sorted(map_marks_sampled['Year'].unique())),
    name="Select Year"
)

background_map = alt.Chart(chicago_boundary).mark_geoshape(
    fill="lightgray",
    stroke="black"
).properties(
    width=1000,
    height=700
).project(
    type='mercator'
)

crime_map = alt.Chart(map_marks_sampled).mark_circle(size=20, opacity=0.5).encode(
    longitude=alt.Longitude('Longitude:Q', title='Longitude'),
    latitude=alt.Latitude('Latitude:Q', title='Latitude'),
    color=alt.value("blue"),
    tooltip=['Latitude:Q', 'Longitude:Q', 'Year:O']
).add_selection(
    selector
).transform_filter(
    selector
)

final_map = background_map + crime_map

final_map.properties(
    title="Crime Locations in Chicago by Year"
).show()
selected_year = 2008
year_data = map_marks_sampled[map_marks_sampled['Year'] == selected_year]
```

For the geospatial map, I used a GeoJSON file to overlay crime locations on a map of Chicago, showing the city's boundaries for context. I cleaned the data by removing invalid latitude and longitude values to ensure accuracy. To handle the large dataset, I applied stratified random sampling, limiting the number of data points to 2,000 per year. This keeps the map clear and easy to interpret while still representing the data well. An interactive year selector allows users to filter crime locations for specific years, making it simple to focus on a particular time frame. The map uses a light gray base layer with black borders to display Chicago's boundaries, and individual crime locations are plotted as semi-transparent blue circles. Tooltips provide details like latitude, longitude, and year when hovered over, adding interactivity. This map lets users explore how crime patterns change over time and across different areas of the city. It highlights trends, such as shifts in hotspots or consistent problem areas, offering useful insights for better understanding and decision-making.

▼ Code

```
import altair as alt
import pandas as pd
import geopandas as gpd

chicago_boundary = gpd.read_file("/Users/suyuanfang/Desktop/Pyhton/final project/trans

chicago_boundary_alt = alt.Data(values=chicago_boundary.__geo_interface__["features"])

theft_data = analysis_data[['Latitude', 'Longitude', 'Year', 'Primary Type']].dropna()
theft_data = theft_data[
    (theft_data['Latitude'] >= -90) & (theft_data['Latitude'] <= 90) &
    (theft_data['Longitude'] >= -180) & (theft_data['Longitude'] <= 180) &
    (theft_data['Primary Type'] == 'THEFT')
]

theft_sampled = theft_data.groupby('Year').apply(lambda x: x.sample(min(len(x), 2000),

selector = alt.selection_single(
    fields=['Year'],
    bind=alt.binding_select(options=sorted(theft_sampled['Year'].unique())),
    name="Select Year"
)

background_map = alt.Chart(chicago_boundary_alt).mark_geoshape(
    fill="lightgray",
    stroke="black"
).properties(
    width=1000,
    height=700
).project(
    type='mercator'
)

theft_map = alt.Chart(theft_sampled).mark_circle(size=20, opacity=0.5).encode(
    longitude=alt.Longitude('Longitude:Q', title='Longitude'),
    latitude=alt.Latitude('Latitude:Q', title='Latitude'),
    color=alt.value("orange"),
    tooltip=['Latitude:Q', 'Longitude:Q', 'Year:O']
).add_selection(
    selector
).transform_filter(
    selector
)

final_map = background_map + theft_map

final_map.properties(
    title="Theft Crime Locations in Chicago by Year"
).show()
```

We created a script that reads a GeoJSON file containing Chicago's boundary data and converts it into a format compatible with Altair. By filtering the dataset for valid latitude and longitude values and focusing specifically on the "THEFT" crime type, we ensured that the analysis exclusively concentrates on larceny offenses, keeping the data relevant. To improve visualization clarity and performance, we applied stratified random sampling, limiting the dataset to a maximum of 2,000 points per year.

We used alt.selection_single() to implement a dynamic year selector, allowing users to interactively choose specific years from a dropdown menu. To ensure geographic accuracy, we employed the Mercator projection to render the background map of Chicago, with black borders and a light gray fill. Theft crime locations are overlaid as semi-transparent orange circles, and tooltips display details such as latitude, longitude, and year when hovered over. Our interactive visualization, "Theft Crime Locations in Chicago by Year (With Map Background)," combines the boundary map and crime data to help users explore spatial and temporal patterns of theft. This tool offers valuable insights into theft hotspots and trends over time, supporting data-driven decision-making and resource allocation.

Shiny App

import pandas as pd from shiny import App, render, ui, reactive import altair as alt import nest_asyncio

nest_asyncio.apply()

analysis_data = pd.read_csv('/Users/suyuanfang/Desktop/Pyhton/final project/analysis_data.csv')

crime_types = analysis_data['Primary Type'].unique()

app_ui = ui.page_fluid(ui.h1("Chicago Crime Data Visualization"), ui.input_selectize("crime_types", "Select Crime Types:", choices=sorted(crime_types.tolist()), selected=crime_types.tolist()[:5], multiple=True,), ui.output_ui("crime_plot"))

def server(input, output, session): @reactive.Calc def filtered_data(): selected_types = input.crime_types() if not selected_types: return pd.DataFrame(columns=['Primary Type']) return analysis_data[analysis_data['Primary Type'].isin(selected_types)]

```
@output
@render.ui
def crime_plot():
    data = filtered_data()
    if data.empty:
        empty_chart = alt.Chart(pd.DataFrame({'x': [], 'y': []})).mark_bar()
        return ui.HTML(empty_chart.to_html())

    crime_counts = data['Primary Type'].value_counts().reset_index()
    crime_counts.columns = ['Crime_Type', 'Count']

    chart = (
        alt.Chart(crime_counts)
        .mark_bar()
        .encode(
            x=alt.X('Crime_Type:N', sort='-y', title='Crime Type'),
            y=alt.Y('Count:Q', title='Number of Crimes'),
            color=alt.Color('Count:Q', scale=alt.Scale(scheme='blues')),
            tooltip=['Crime_Type', 'Count'],
        )
        .properties(
            title='Number of Crimes for Selected Crime Types',
```

```
            width='container',
            height=400,
        )
        .configure_axis(labelAngle=-45)
    )
    return ui.HTML(chart.to_html())
```

app = App(app_ui, server)

if **name** == “**main**”: app.run()

This Shiny app allows users to explore Chicago crime data interactively. It loads crime data from a CSV file and provides a dropdown menu where users can select one or more crime types to analyze. Based on the selected types, the app filters the data and dynamically generates a bar chart using Altair. The chart displays the number of crimes for each selected type, with color intensity representing the counts and tooltips showing details when hovering over the bars. The app ensures a smooth user experience by dynamically updating the chart in response to user input and handling cases where no data is selected.

recommendations and conclusion

In order to address these obstacles, it is imperative to implement a multifaceted strategy that combines immediate interventions with long-term structural reforms. It is advised that seasonal crime prevention programs be implemented to address specific crime categories during their peak periods. For example, the implementation of targeted measures during the winter can address property crimes, while the deployment of additional law enforcement resources and community outreach programs during the summer can help mitigate the increase in larceny.

In order to address the root causes of illicit activity, it is essential to expand welfare programs. Investing in education, mental health services, and affordable housing can significantly reduce crime rates by addressing the underlying social inequalities. These initiatives should prioritize communities that have been identified as high-crime areas through the analysis.

Urban planning has a substantial impact on crime prevention. By enhancing the visibility and illumination of public spaces, particularly in high-crime areas, it is possible to deter criminal activities. The establishment of secure neighborhoods is facilitated by the promotion of mixed-use developments that integrate residential, commercial, and recreational spaces, thereby fostering informal surveillance and community engagement.

Finally, the reduction of crime requires the expansion of employment opportunities. Residents can acquire the requisite skills for sustainable employment through vocational training programs, while job opportunities can be generated in vulnerable communities through partnerships with local businesses. In addition to improving economic prospects, these initiatives enhance a sense of community stability and resilience.

The findings of this report emphasize the importance of data-driven policymaking in Chicago’s crime prevention initiatives. By integrating seasonal prevention strategies, welfare support, urban redevelopment, and employment growth, policymakers can develop comprehensive solutions that promote safety and equity. The holistic approach of addressing crime through socioeconomic empowerment and sustainable development is consistent with the progressive vision, which will enable a more prosperous and secure Chicago.