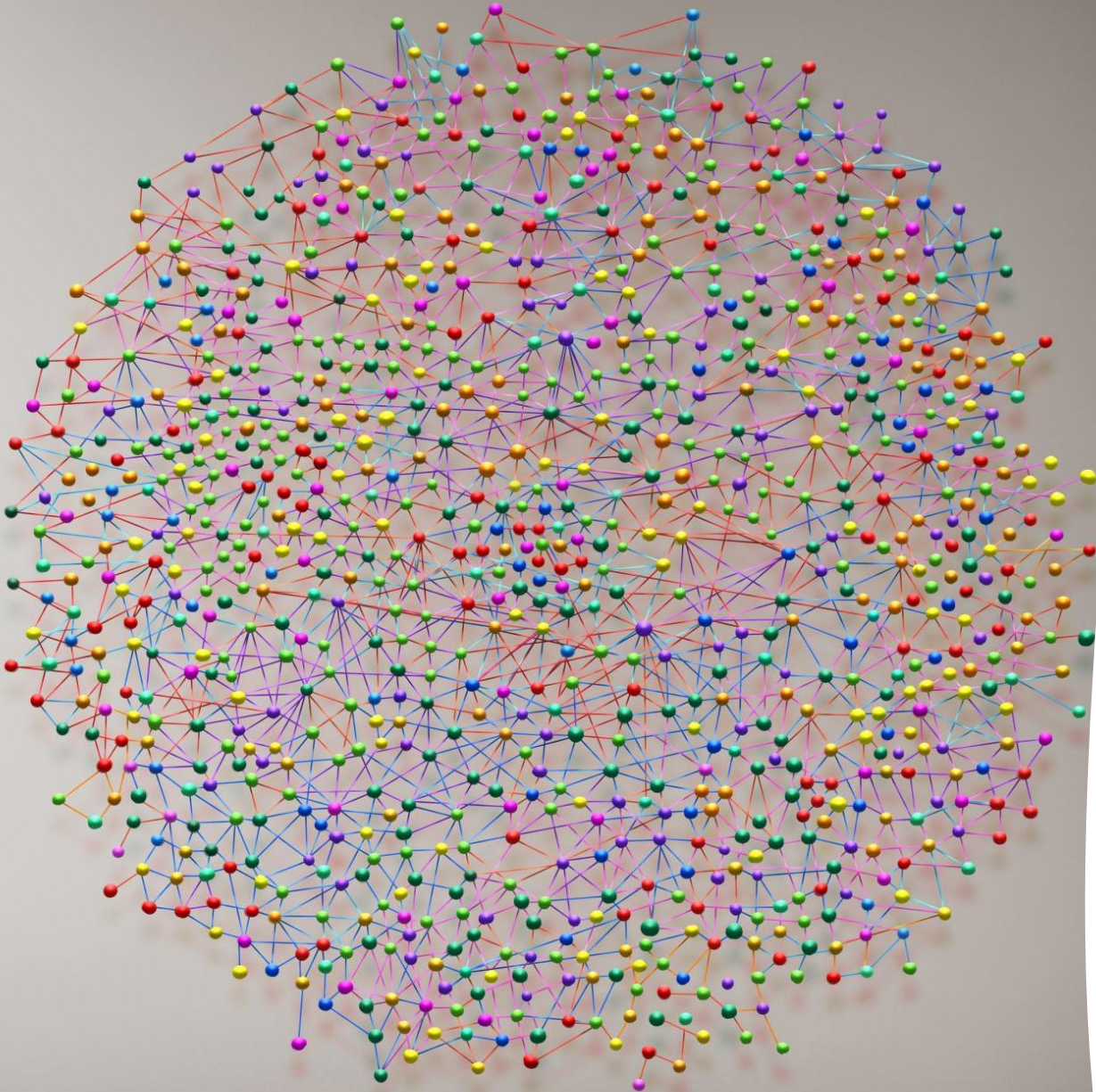




Exploratory Data Analysis –EDA Presentation

MSc Data and Artificial Intelligence
Module: L7 - Data Mining & Visualisation

Module code: ACCA7015

Lecturer: Dr Nough Elmitwally

Majib Miah 2234516

Submission deadline: 2:00 pm, Friday,
06/03/2026

Tools used in this EDA



Weka: Used to inspect the dataset structure, identify missing values, and apply cleaning filters (e.g., ReplaceMissingValues).



Tableau: Used for visual exploration, grouping, and multivariate analysis (District comparison, monthly trend, hotspot map).



PyCharm (Python): Used to run code that checks data quality and generates reproducible charts saved as PNG files.



Python makes the process repeatable and provides evidence of method through code and generated charts.



Why these tools were chosen



Benefit Using three tools strengthens reliability because results are cross-checked and visuals are clearer for the audience.



Weka is quick for data inspection and simple cleaning.



Tableau is best for clear visuals, especially geographic maps and grouped comparisons.

Aim

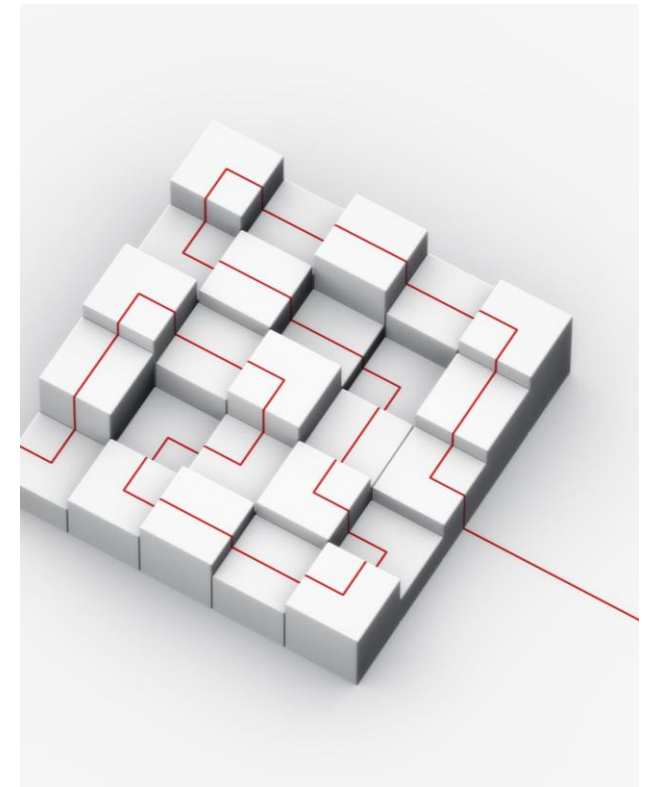
To explore, clean, and analyse the Chicago crime dataset for 2022 using Exploratory Data Analysis (EDA) in order to identify patterns, trends, and relationships that can support decision-making.

What this means in practice:

I am not building a prediction model here.

The focus is on understanding the data and producing clear insights supported by evidence.

Benefit A clear aim keeps the work focused on meaningful insights rather than producing charts with no purpose.



An abstract background graphic on the left side of the slide. It features several vertical orange bars of varying heights. Overlaid on these bars is a white line chart with circular markers at each data point. Some data points are labeled with numbers: '183.102' and '154.178'. The overall aesthetic is modern and data-driven.

Objectives

Objectives of the analysis

Clean and prepare the dataset for analysis (handle missing values, check formats).

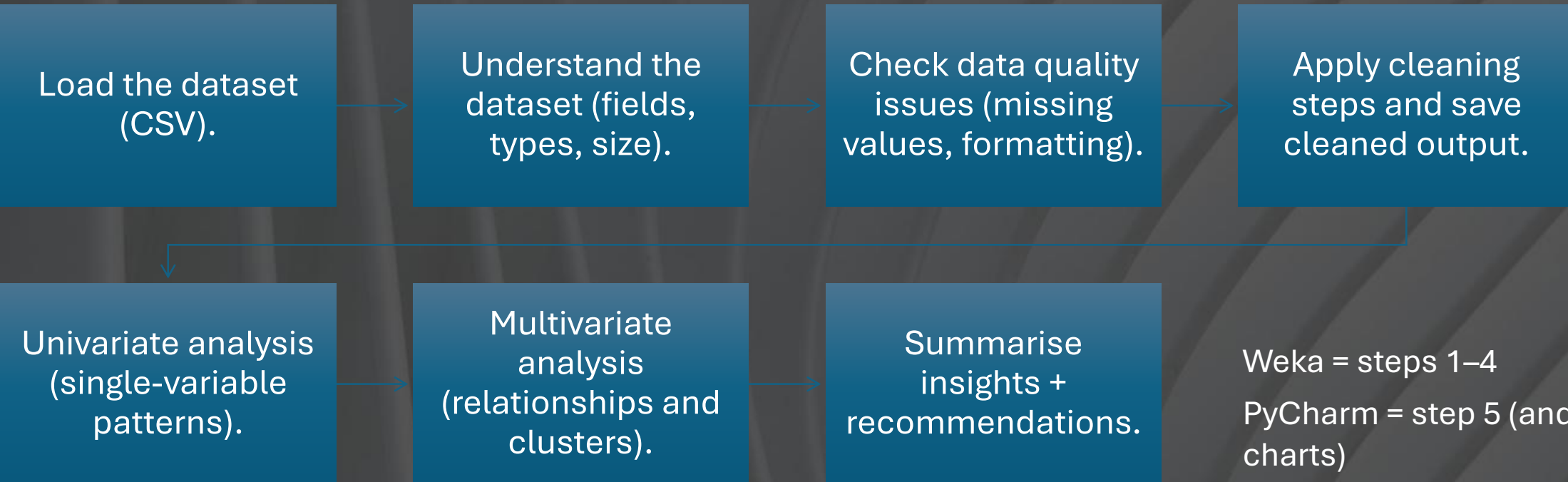
Perform **univariate EDA** to understand individual variables (crime types, arrests, time patterns).

Perform **multivariate EDA** to explore relationships (district vs crime, month vs crime, location hotspots, arrest rate patterns).

Summarise findings into **actionable recommendations**.

Benefit These objectives ensure I cover the full EDA pipeline and meet the marking criteria.

EDA Workflow



Weka = steps 1–4

PyCharm = step 5 (and supporting charts)

Tableau = step 6 (district, month, hotspot)

Benefit A structured workflow reduces mistakes and makes the analysis easier to justify during questions.

Dataset Overview (Weka screenshot)

Weka Explorer

PreprocessClassifyClusterAssociateSelect attributesVisualize

Open file...Open URL...Open DB...Generate...UndoEdit...Save...

FilterChooseMultiFilter -F "weka.filters.AllFilter" -S 1

Current relationRelation: cleaned_crime-weka.filters.unsupervised.attribute.ReplaceMissingValuesInstances: 234891

AttributesAttributes: 19Sum of weights: 234891

AllNoneInvertPattern

No.

☐ ID

☐ Case Number

☐ Date

☐ Block

☐ IUOCR

☒ Primary Type

☐ Description

☐ Arrest

☐ Domestic

☐ District

☐ Ward

☐ Community Area

☐ Year

☐ Latitude

☐ Longitude

☐ Month

☐ Hour

☐ Day_of_Week

☐ Time_Period

Name

Remove

Selected attributeName: Primary TypeMissing: 0 (0%)Distinct: 31Type: NominalUnique: 0 (0%)

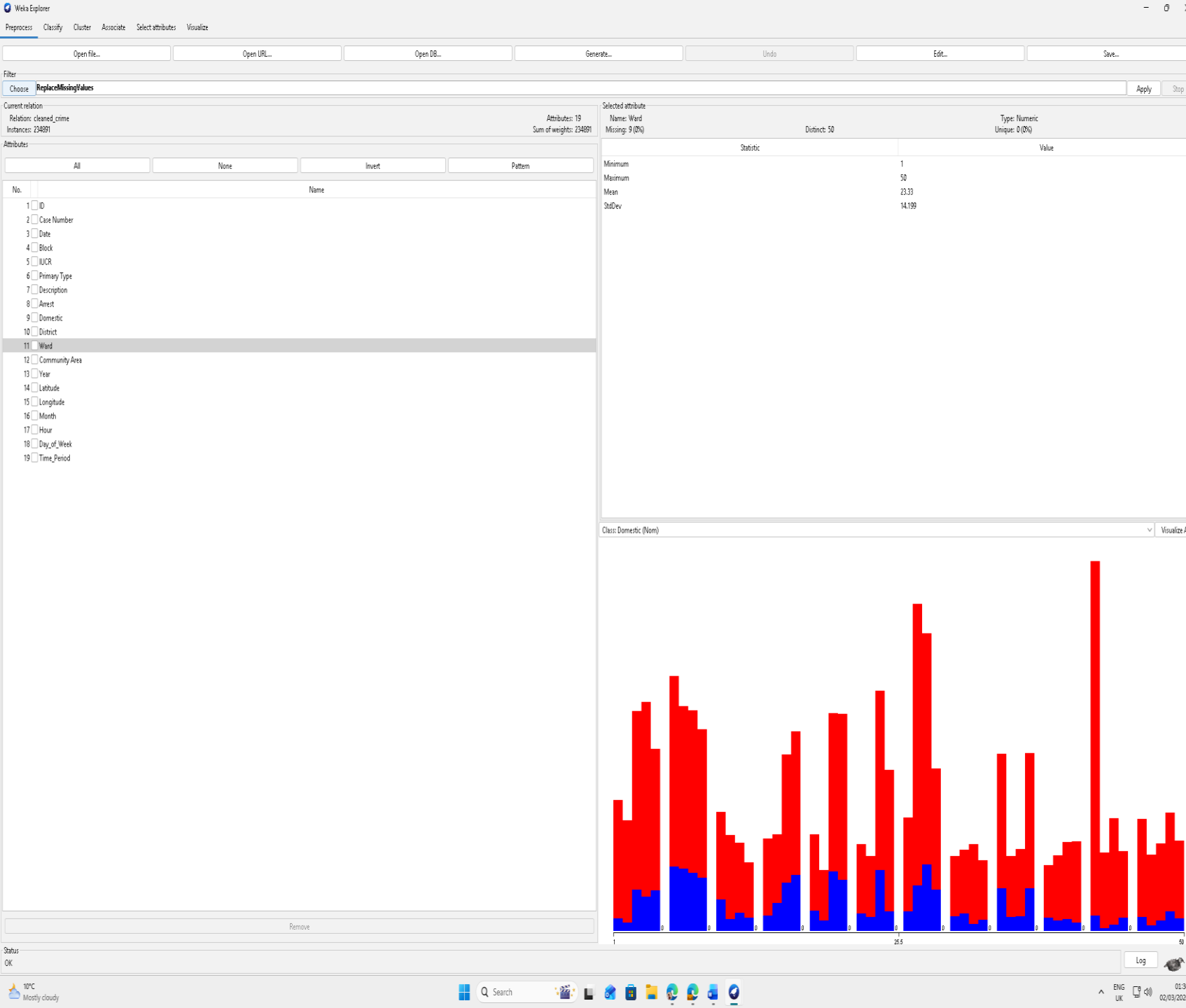
No.	Label	Count	Weight
1	OTHER OFFENSE	14371	14371
2	SEX OFFENSE	1138	1138
3	WEAPONS VIOLATION	8710	8710
4	BATTERY	40806	40806
5	THEFT	53625	53625
6	CRIMINAL SEXUAL ASSAULT	1486	1486
7	ROBBERY	8953	8953
8	BURGLARY	7564	7564
9	ARSON	421	421
10	OFFENSE INVOLVING CHILDREN	1764	1764
11	MOTOR VEHICLE THEFT	21386	21386
12	DECEPTIVE PRACTICE	15371	15371
13	ASSAULT	20738	20738
14	CRIMINAL DAMAGE	27028	27028
15	NARCOTICS	4037	4037
16	CRIMINAL TRESPASS	4202	4202
17	HOMICIDE	732	732
18	STALKING	429	429
19	INTERFERENCE WITH PUBLIC OFFICER	392	392
20	OBSCENITY	41	41
21	PUBLIC PEACE VIOLATION	706	706
22	INTIMIDATION	178	178
23	KIDNAPPING	116	116
24	LIQUOR LAW VIOLATION	203	203
25	OTHER NARCOTIC VIOLATION	7	7
26	PROSTITUTION	283	283

Class: Arrest (Nom)Visualize All

10°C Mostly cloudy

Search

01:4902/03/2026



Data Quality Check: Missing Values

Looked for missing values across attributes in Weka.

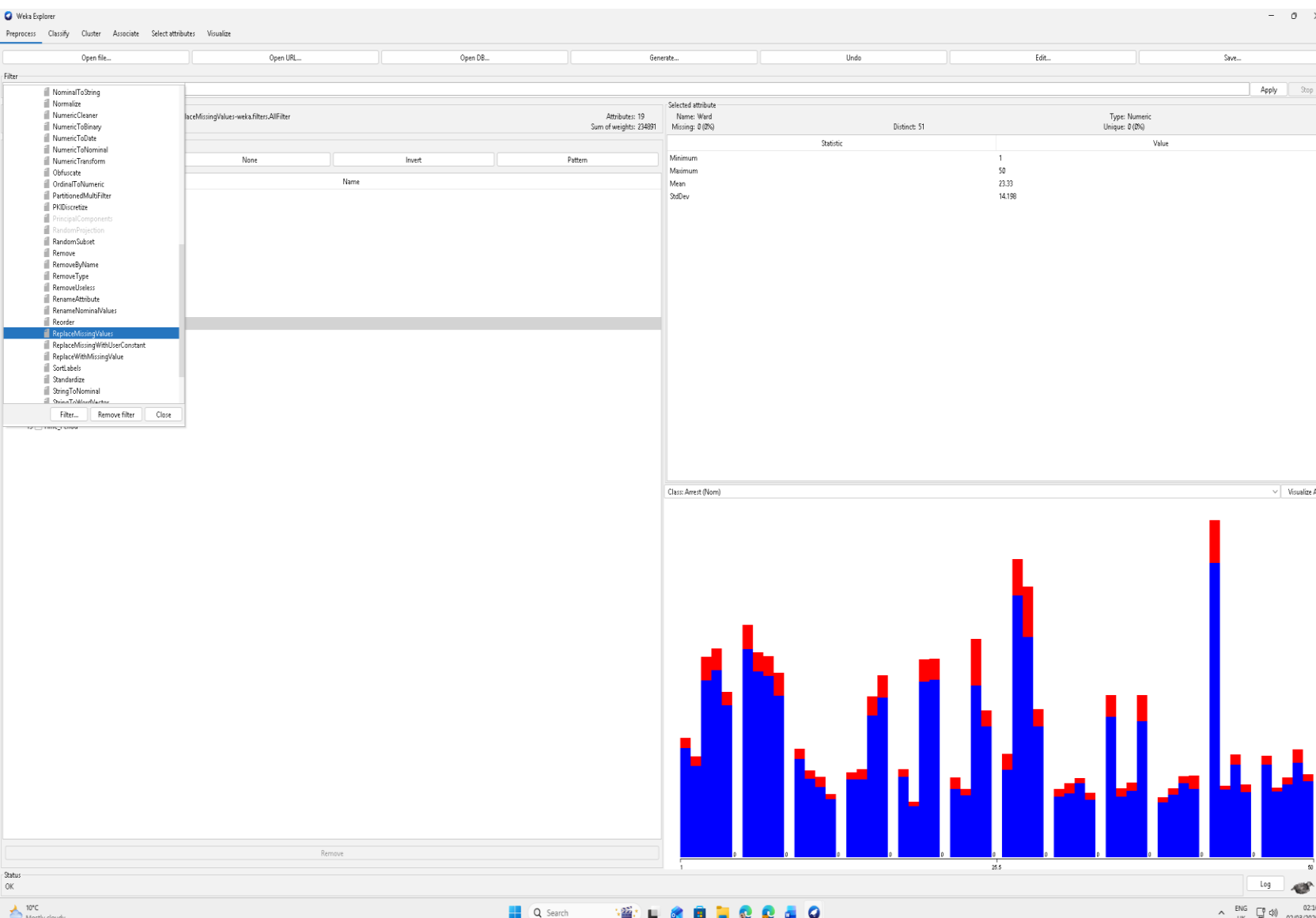
Found missing values in **Ward**

Why missing values matter

Missing ward information can distort area-based summaries.

It can reduce the accuracy of grouping and hotspot interpretation if location metadata is incomplete.

Data Cleaning Applied ReplaceMissingValues t



Applied Weka filter
Replace Missing Values. Exported a
cleaned CSV to use consistently in
Tableau and Python.

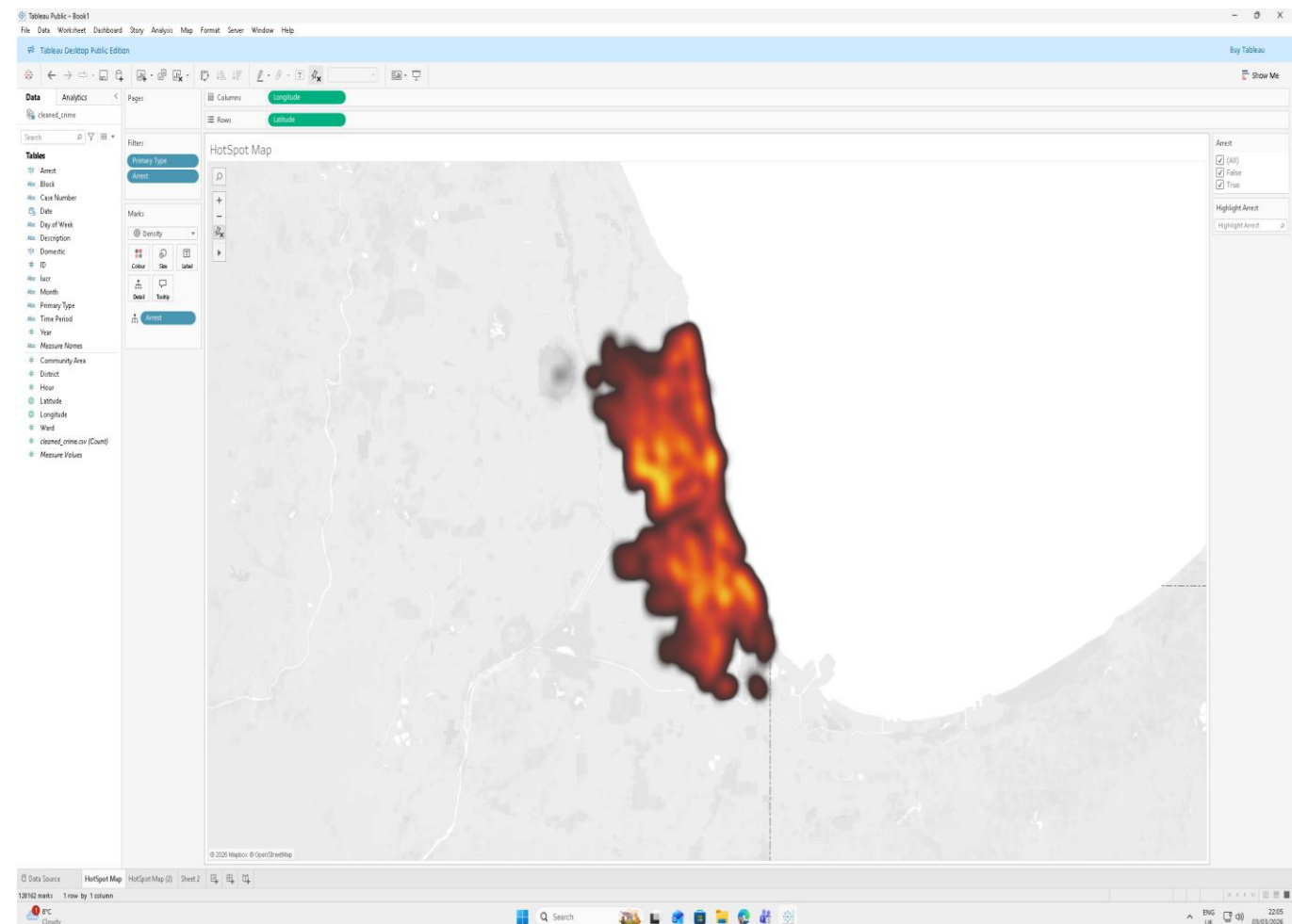
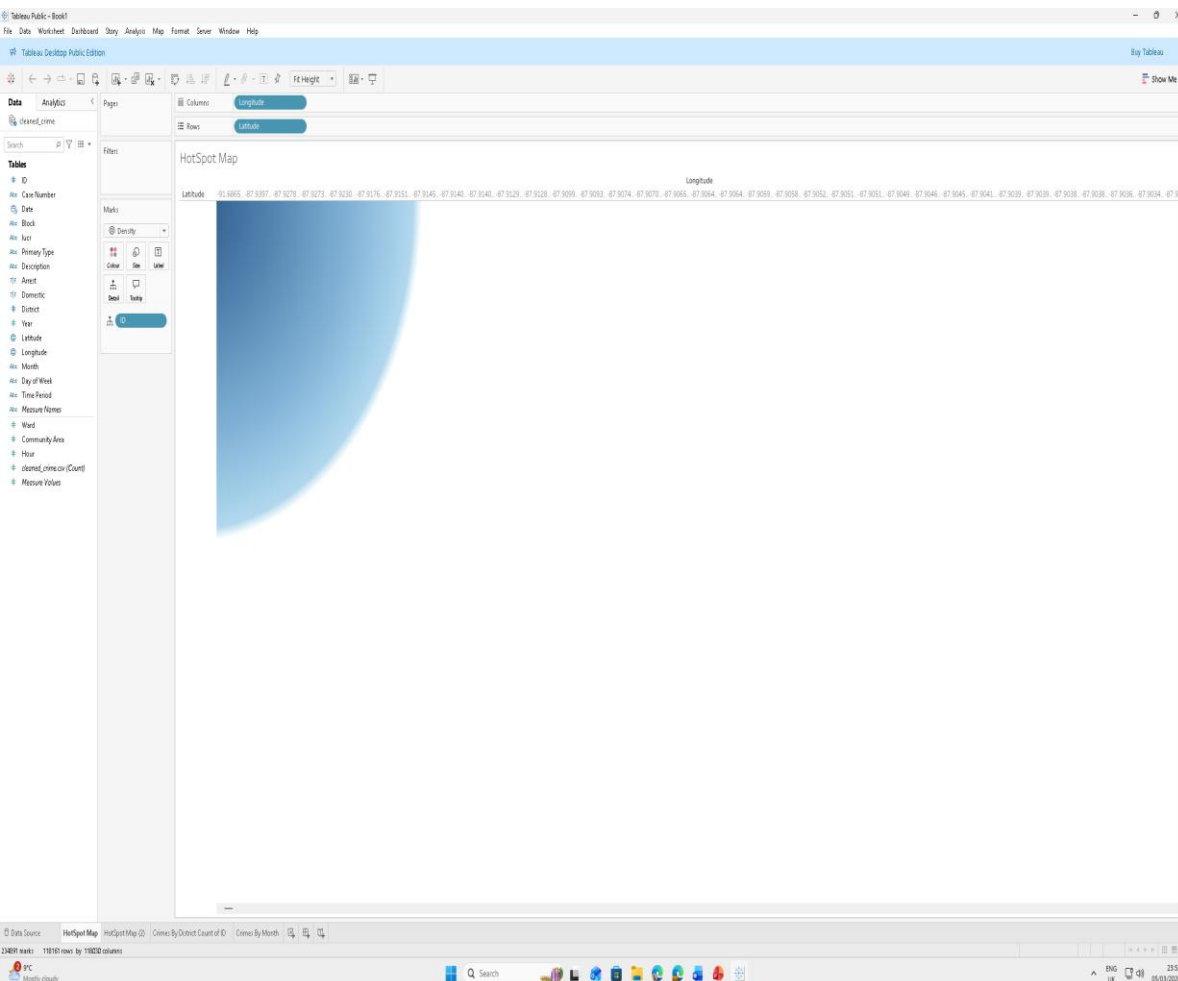
Filling missing values prevents blanks
affecting counts, groupings, and
visualisations. It improves reliability of
EDA results because every record can be
included in analysis.

Benefit Cleaner data leads to cleaner
visuals and stronger confidence in
results.

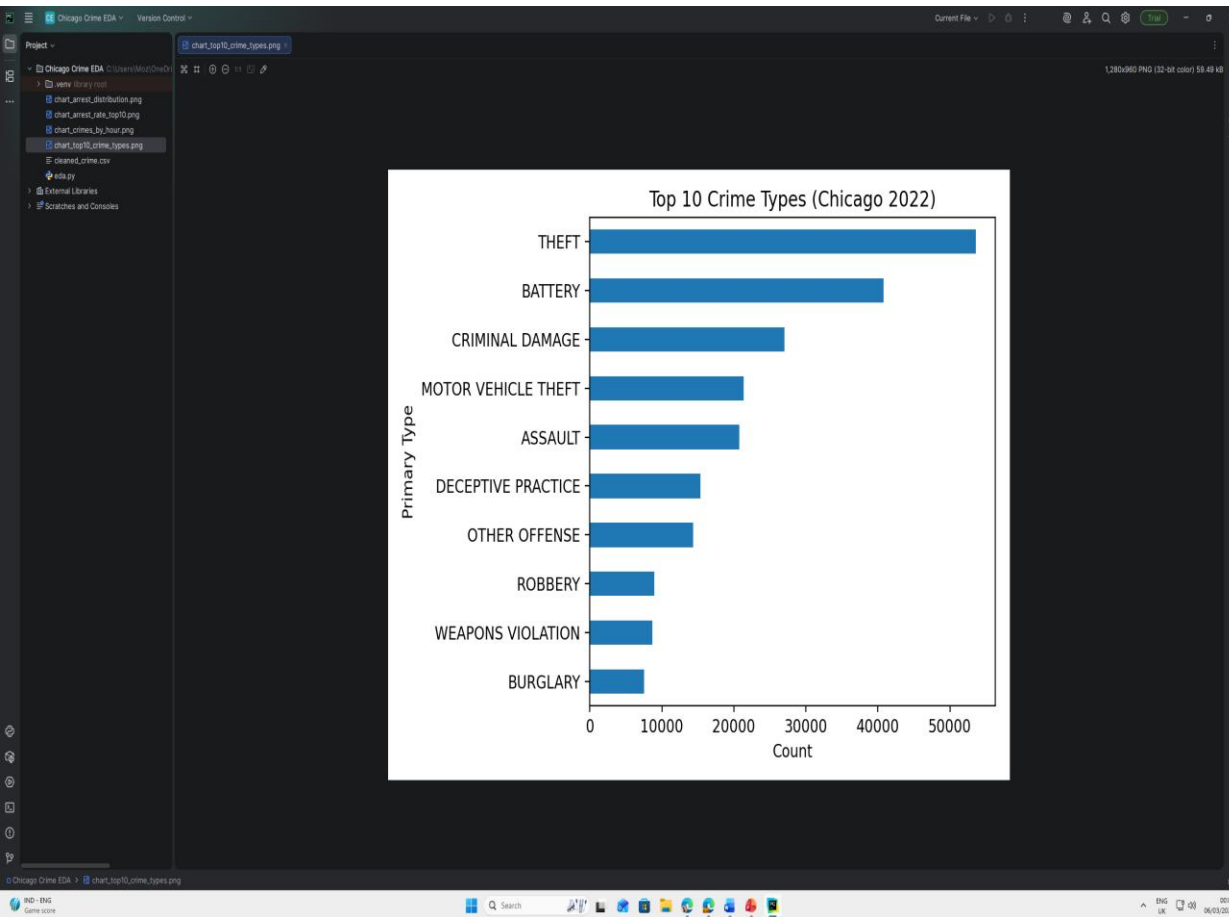
Practical Handling Tableau Aggregation

Problem encountered in Tableau: Latitude and Longitude initially appeared as AVG(Latitude) and AVG(Longitude) which collapsed the map to only a few points.

Fix applied: Converted Latitude and Longitude to Dimensions (blue). Added ID to the Detail shelf to plot record-level marks. Ensured District is a Dimension (blue) so Tableau can correctly count incidents by district.



Univariate EDA: Top 10 Crime Types

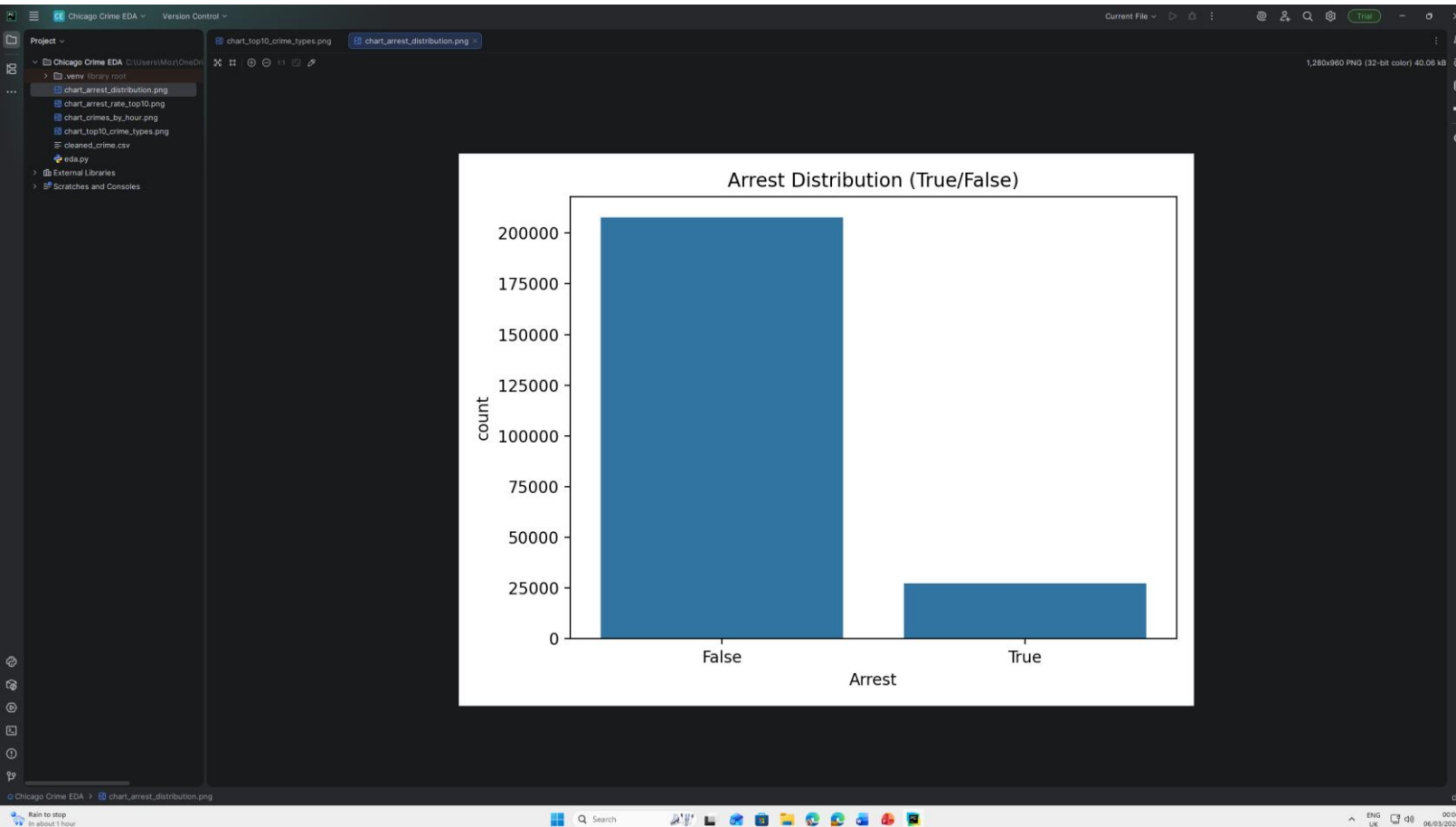


What the chart shows: Top 10 crime categories by total count in 2022.

Crime is concentrated: a small set of categories represent a large proportion of incidents. **This is useful because we can prioritise prevention and analysis around the categories driving most incidents.**

Benefit: Helps identify which crime categories should be targeted first for maximum impact.

Univariate EDA: Arrest Distribution



What the chart shows: Comparison of Arrest = True vs Arrest = False.

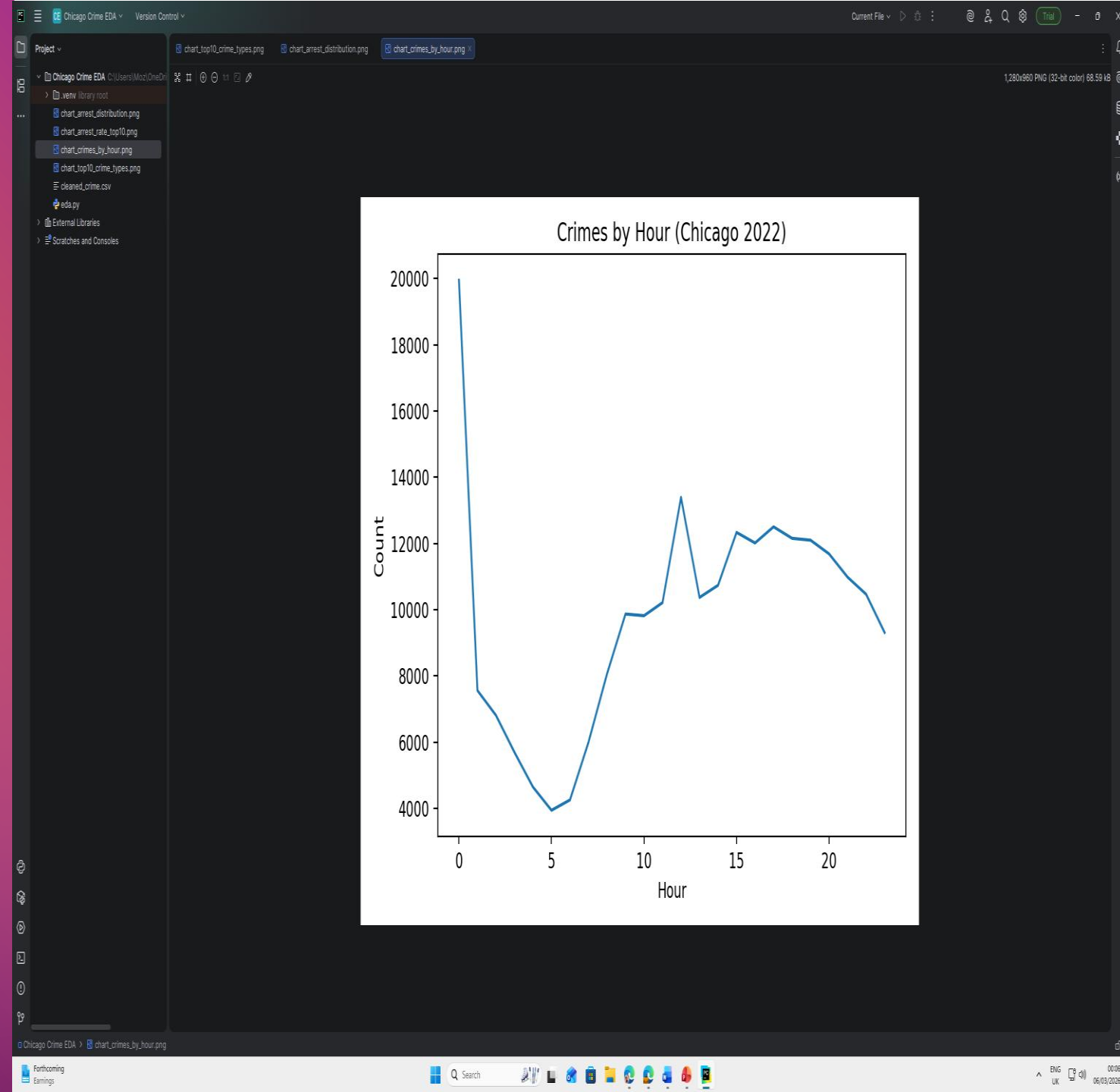
Most incidents result in no arrest based on the distribution. Arrest outcomes help evaluate how often crimes lead to enforcement action and can vary by crime type and context.

Univariate EDA Crimes by Hour

What the chart shows: Crime frequency across hours of the day.

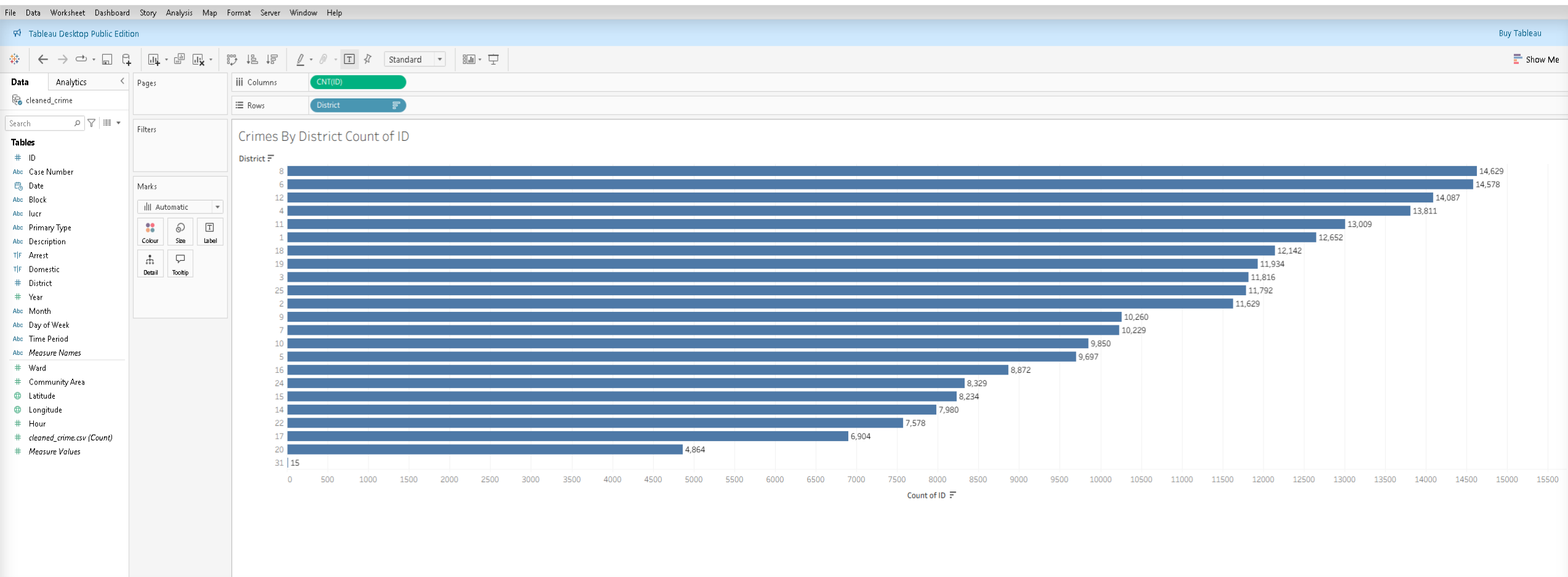
Crime is not evenly distributed through the day. Peaks suggest time windows where risk is higher.

Benefit: Supports smarter scheduling and time-based prevention strategies.

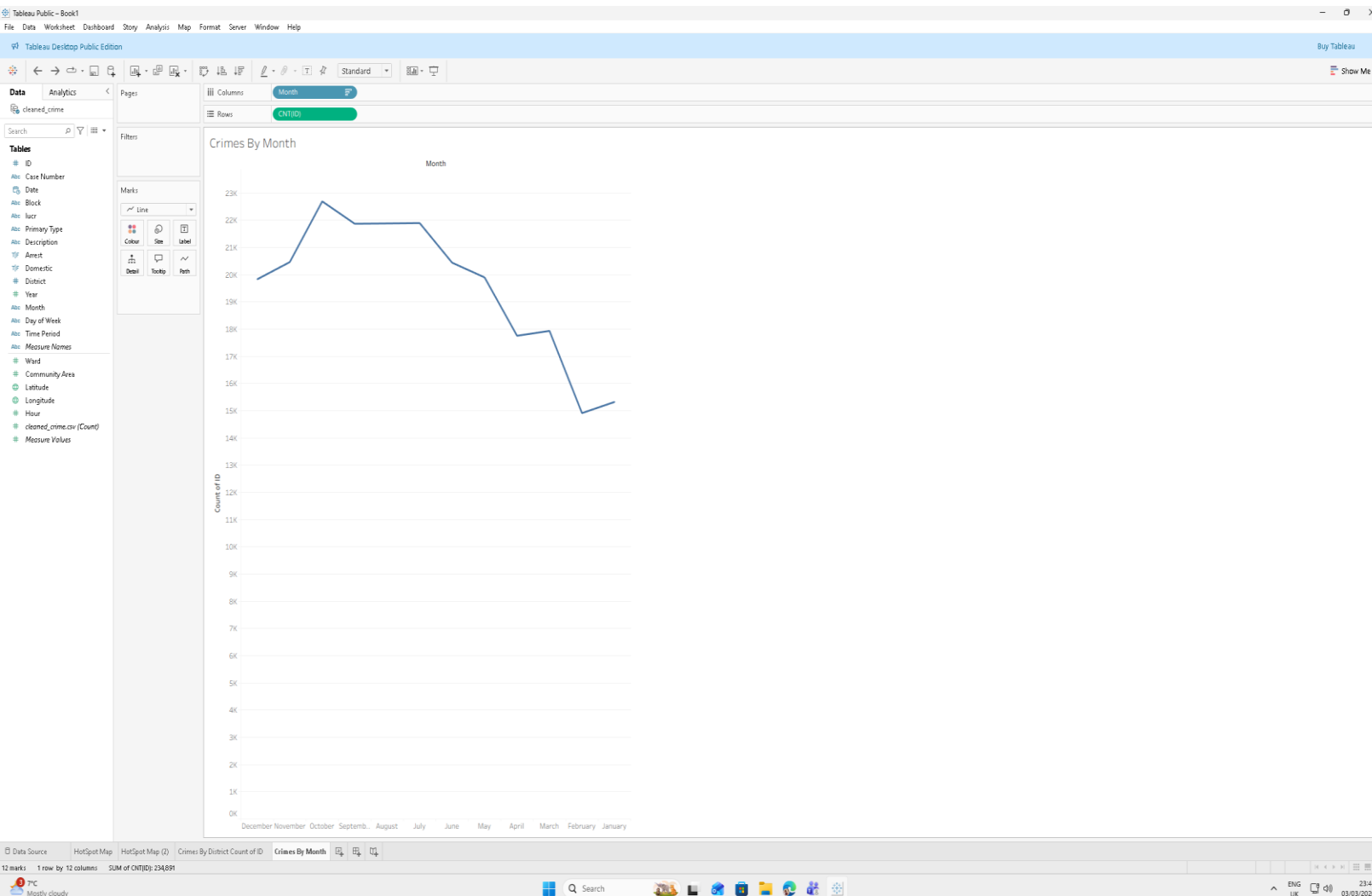


Multivariate EDA: Crimes by District

- **Method:** Grouped incidents by District and calculated totals using COUNT(ID).
- **Finding:** Highest volumes: District 8, then District 6.
- **Why this matters:** District-level comparisons show where the workload and risk are concentrated. High-volume districts can be prioritized for patrol allocation or targeted interventions.
- **Benefit:** Helps decision-makers allocate resources more effectively.



Multivariate EDA: Crimes by Month (Tableau line chart)



Aggregated incidents by Month using COUNT(ID) (after fixing month sorting).

Finding: November appears as the highest month in the dataset.

Helps anticipate demand and plan prevention rather than reacting late.

Multivariate EDA: Hotspot Map (Tableau density map)

Method Plotted Latitude/Longitude and used Density mapping.

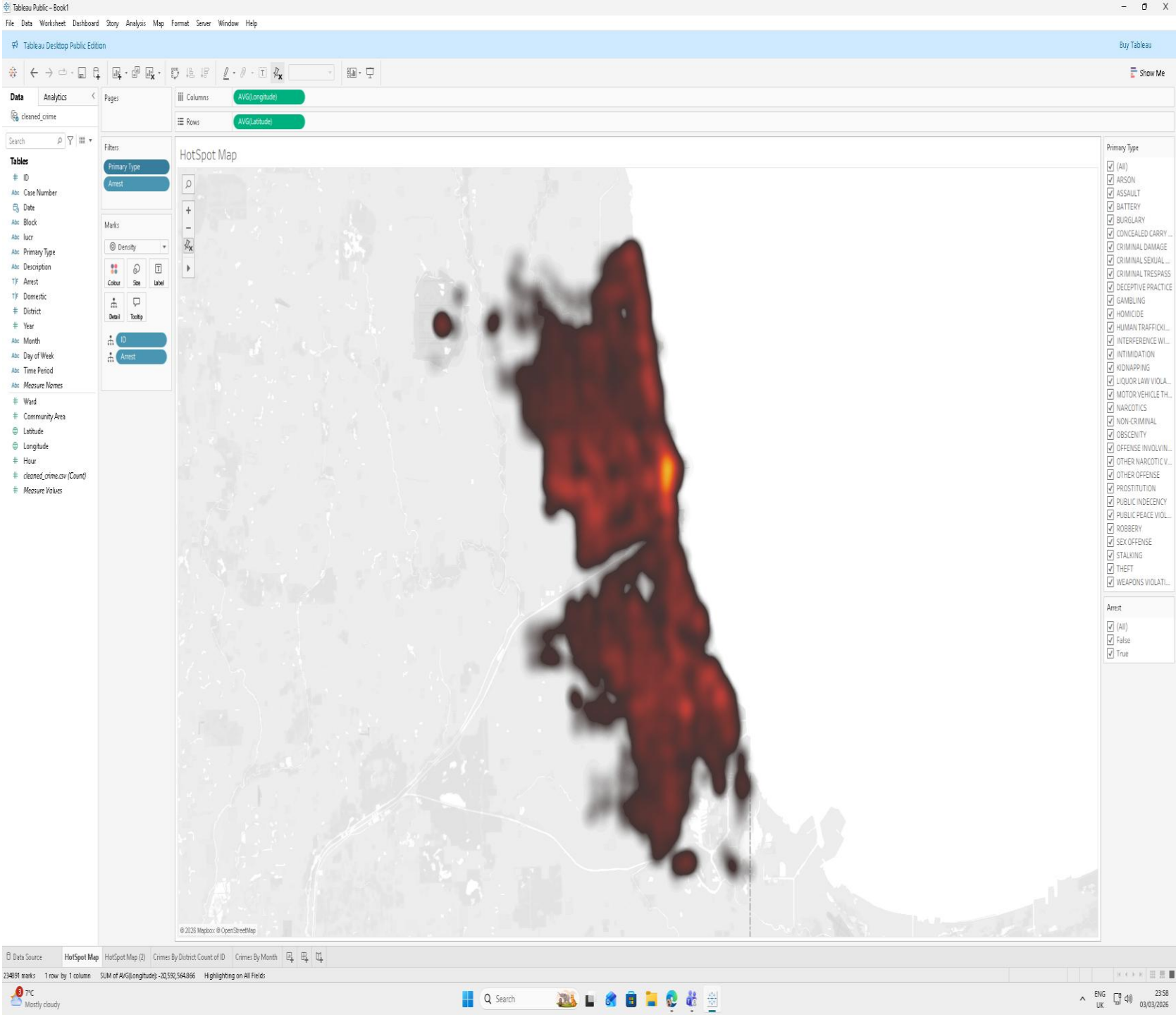
Finding: Crimes cluster into hotspots rather than being evenly spread.

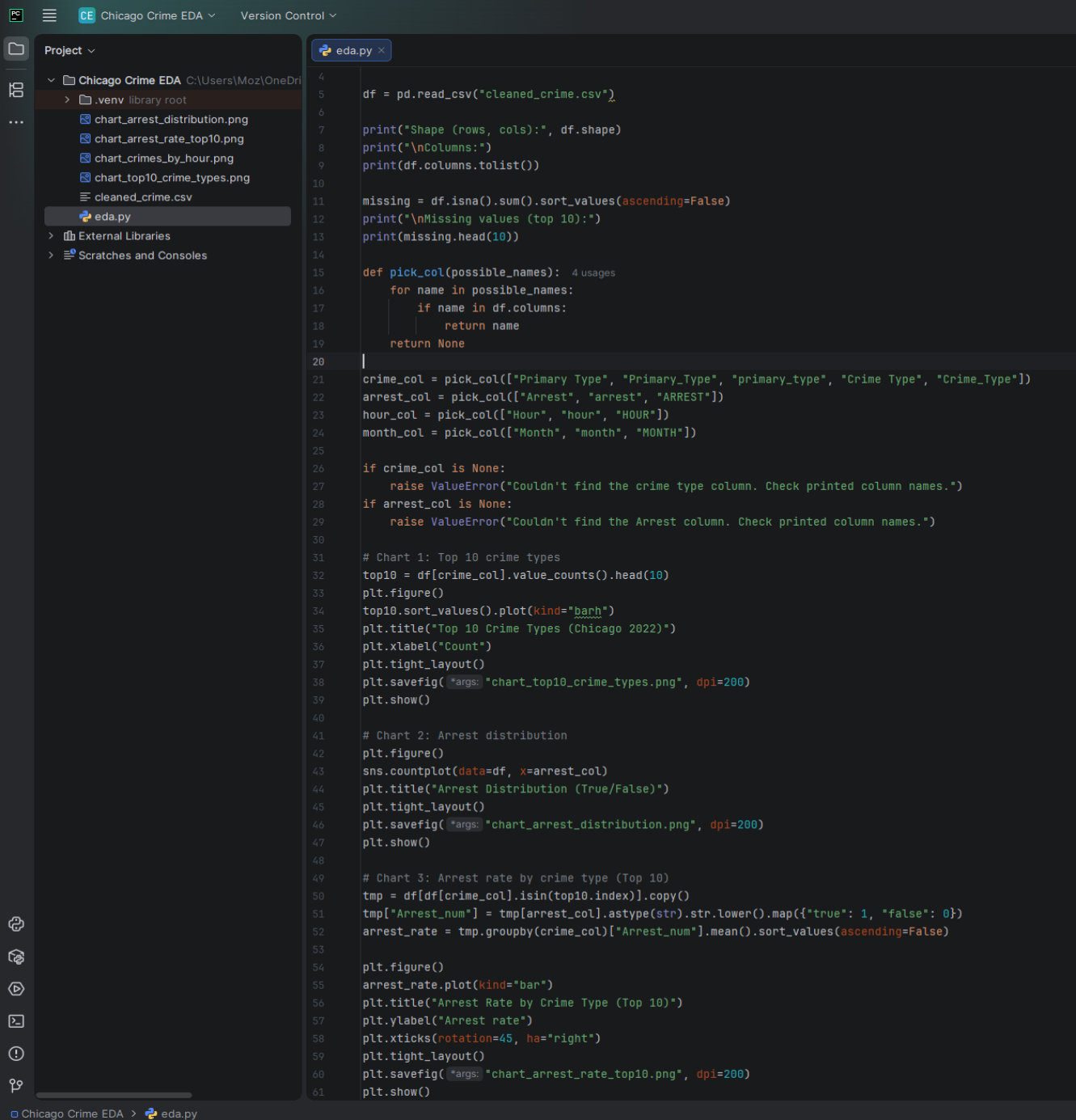
Why this matters

Place-based interventions work best when hotspots exist (patrol routes, lighting, CCTV, community programs).

This supports focusing resources on smaller areas for larger overall benefit.

Benefit: Identifies exactly where targeted prevention can be most effective.





The screenshot shows the PyCharm IDE with a project named "Chicago Crime EDA". The left sidebar displays the project structure, including a "cleaned_crime.csv" file and several generated charts (e.g., "chart_arrest_distribution.png", "chart_arrest_rate_top10.png", "chart_crimes_by_hour.png", "chart_top10_crime_types.png"). The main editor window shows the "eda.py" script, which is a Python program using Pandas and Matplotlib to analyze crime data. The script loads the "cleaned_crime.csv" file, checks for missing values, and generates three charts: a bar chart of top 10 crime types, a count plot of arrest distribution, and a bar chart of arrest rate by crime type. The script uses a helper function "pick_col" to select columns based on case-insensitive matching.

```
4 df = pd.read_csv("cleaned_crime.csv")
5
6 print("Shape (rows, cols):", df.shape)
7 print("\nColumns:")
8 print(df.columns.tolist())
9
10
11 missing = df.isna().sum().sort_values(ascending=False)
12 print("\nMissing values (top 10):")
13 print(missing.head(10))
14
15 def pick_col(possible_names): 4 usages
16     for name in possible_names:
17         if name in df.columns:
18             return name
19     return None
20
21 crime_col = pick_col(["Primary Type", "Primary_Type", "primary_type", "Crime Type", "Crime_Type"])
22 arrest_col = pick_col(["Arrest", "arrest", "ARREST"])
23 hour_col = pick_col(["Hour", "hour", "HOURL"])
24 month_col = pick_col(["Month", "month", "MONTH"])
25
26 if crime_col is None:
27     raise ValueError("Couldn't find the crime type column. Check printed column names.")
28 if arrest_col is None:
29     raise ValueError("Couldn't find the Arrest column. Check printed column names.")
30
31 # Chart 1: Top 10 crime types
32 top10 = df[crime_col].value_counts().head(10)
33 plt.figure()
34 top10.sort_values().plot(kind="barh")
35 plt.title("Top 10 Crime Types (Chicago 2022)")
36 plt.xlabel("Count")
37 plt.tight_layout()
38 plt.savefig("args: chart_top10_crime_types.png", dpi=200)
39 plt.show()
40
41 # Chart 2: Arrest distribution
42 plt.figure()
43 sns.countplot(data=df, x=arrest_col)
44 plt.title("Arrest Distribution (True/False)")
45 plt.tight_layout()
46 plt.savefig("args: chart_arrest_distribution.png", dpi=200)
47 plt.show()
48
49 # Chart 3: Arrest rate by crime type (Top 10)
50 tmp = df[df[crime_col].isin(top10.index)].copy()
51 tmp["Arrest_num"] = tmp[arrest_col].astype(str).str.lower().map({"true": 1, "false": 0})
52 arrest_rate = tmp.groupby(crime_col)["Arrest_num"].mean().sort_values(ascending=False)
53
54 plt.figure()
55 arrest_rate.plot(kind="bar")
56 plt.title("Arrest Rate by Crime Type (Top 10)")
57 plt.ylabel("Arrest rate")
58 plt.xticks(rotation=45, ha="right")
59 plt.tight_layout()
60 plt.savefig("args: chart_arrest_rate_top10.png", dpi=200)
61 plt.show()
```

Code Used (PyCharm / Python)

Loaded the dataset using Pandas (read_csv).

Checked missing values with isna().sum().

Generated charts

Saved charts as PNG files for the presentation.

Why this matters: Makes the analysis repeatable and consistent.

Easy to update if the dataset changes.

Automates calculations (like arrest rate) instead of doing them manually.

How arrest rate was calculated: Converted Arrest to 1 for True and 0 for False, then took the average for each crime type.

Benefit: The code proves the method and reduces human error in counting and grouping.

Executive Summary + Recommendations

Executive summary (key findings):

- Crime is concentrated in a small set of top categories (Top 10 dominate).
- Arrest outcomes are mostly False overall.
- District crime volumes are uneven, with District 8 and District 6 highest.
- Monthly trend shows variation, with November the highest month.
- The hotspot map shows clear geographic clustering.

Recommendations (actionable):

1. Prioritize patrol/resources in District 8 and 6.
2. Align staffing/patrol timing with the peak hours shown in the hourly chart.
3. Target prevention on the highest-frequency crime types (Top categories).
4. Use hotspot areas for focused interventions (patrol routes, CCTV, lighting, community work).
5. Review low-arrest categories using the arrest-rate chart to improve enforcement strategy.

Benefit: These actions connect directly to the patterns found, so they are evidence-based rather than guesswork.