Fundamentals of Machine Learning

Concepts, Techniques and Tools to Build Intelligent Systems

Module 2

Data Science

Ali Samanipour

May. 2023

Ali Samanipour
linkedin.com/in/Samanipour

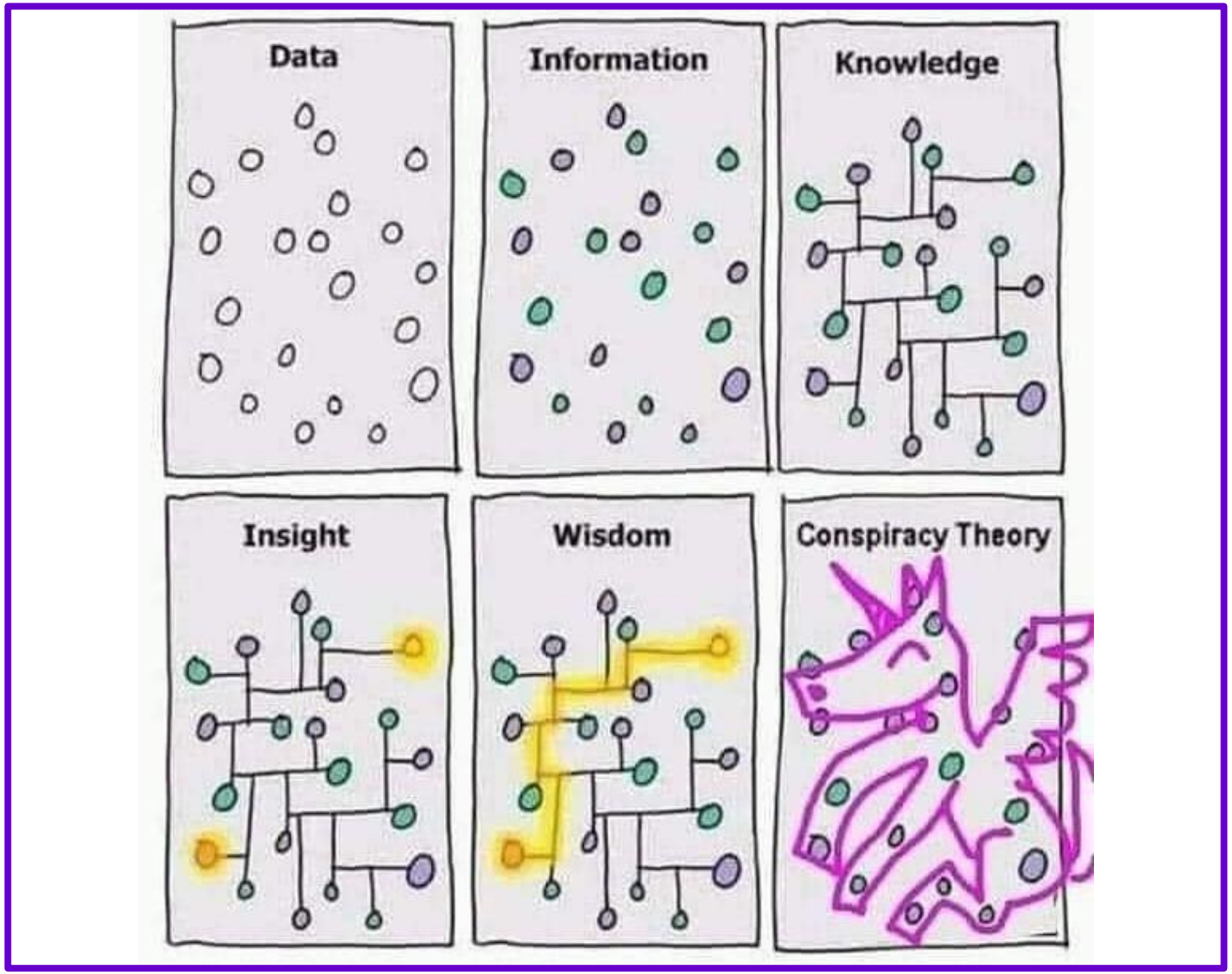
1. What is Data?
2. O.S.E.M.N Framework
3. Exploratory Data Analysis
4. Investigating The Data
5. Hypothesis Generation and Validation
6. More Data Visualization Example

What is Data?

“facts and statistics collected together for reference or analysis.”

Data, Information, Knowledge, Insight and Wisdom

Data and **information** or **knowledge** are often used interchangeably; however, **data becomes information when it is viewed in context or post analysis**



Types of Data

Structured data - Structured data is generally stored in **tabular form**, and it can be **stored in a relational database**. It can be names, phone numbers, location, or other metrics like distance, loan amount, etc. and generally, we can query the relational table with SQL.

Semi-structured data - Semi-structured data is similar to structured data, but it **does not follow the conventional relational table structure**. Files like XML, JSON, etc. are examples of semi-structured data.

Unstructured data - As the name suggests, unstructured data **follows no formal structure** or relational table. E.g., texts, tweets from Twitter, Media (Audio-Video, etc.)

Lets Start Our Project (Boston crime dataset)

data dictionary is "**a set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.**"

Field Name, Data Type, Required	Description
[incident_num] [varchar] (20) NOT NULL,	Internal BPD report number
[offense_code] [varchar] (25) NULL,	Numerical code of offense description
[Offense_Code_Group_Description] [varchar] (80) NULL,	Internal categorization of [offense_description]
[Offense_Description] [varchar] (80) NULL,	The primary descriptor of the incident
[district] [varchar] (10) NULL,	What district the crime was reported in
[reporting_area] [varchar](10) NULL,	RA number associated with the location where the crime was reported from.
[shooting][char] (1) NULL,	Indicated, a shooting took place.
[occurred_on] [datetime2](7) NULL,	Earliest date and time the incident could have taken place
[UCR_Part] [varchar](25) NULL,	Universal Crime Reporting Part number (1,2, 3)
[street] [varchar](50) NULL,	Street name the incident took place

Knowing The Data

The best way to know more about the data is to get your hands dirty.

1. What is Data?
2. O.S.E.M.N Framework
3. Exploratory Data Analysis
4. Investigating The Data
5. Hypothesis Generation and Validation
6. More Data Visualization Example

O.S.E.M.N. framework

All Machine Learning Projects and Data Science Projects have a basic framework named O.S.E.M.N. (**Obtaining**, **Scrubbing**, **Exploring**, **Modeling**, **INterpreting**), and we can see with framework Data Fetching, Data Cleaning, and Data Exploring takes up 60% of the pipeline

What is Data Obtaining?

All the steps required to gather the data are considered Data Obtaining/Fetching.

```
1  import pandas as pd
2
3  crime_data = pd.read_csv("data/boston_crime_dataset.csv")
```

What is Data Scrubbing?

Data Scrubbing is a process of cleaning the data which will be fit for use in the next stage that is Data Exploration and Analysis

During this phase, we will mostly focus on handling incorrect data, missing values, and errors related to the data structures.

Data Scrubbing, also known as Data Cleaning, takes up the maximum time during the process of Data Analysis

Finding Data Types

Depending on the data type, different data cleaning techniques can be applied. And it's not just cleaning; **we need to scrub the data logically to reduce ambiguity**.

```
crime_data.dtypes
```

```
INCIDENT_NUMBER          object
OFFENSE_CODE              int64
OFFENSE_CODE_GROUP       object
OFFENSE_DESCRIPTION      object
DISTRICT                 object
REPORTING_AREA           object
SHOOTING                 object
OCCURRED_ON_DATE         object
YEAR                      int64
MONTH                     int64
DAY_OF_WEEK              object
HOUR                      int64
UCR_PART                 object
STREET                   object
Lat                     float64
Long                    float64
Location                 object
dtype: object
```

Sample data

we can see all the data and how it looks and why it is of the given data type

```
crime_data.loc[1]
```

```
INCIDENT_NUMBER                          I192068458
OFFENSE_CODE                                   3112
OFFENSE_CODE_GROUP          Landlord/Tenant Disputes
OFFENSE_DESCRIPTION       LANDLORD - TENANT SERVICE
DISTRICT                                        C11
REPORTING_AREA                                  336
SHOOTING                                        NaN
OCCURRED_ON_DATE                2019-08-28 20:53:00
YEAR                                           2019
MONTH                                             8
DAY_OF_WEEK                                Wednesday
HOUR                                              20
UCR_PART                                  Part Three
STREET                                     NORTON ST
Lat                                          42.3063
Long                                        -71.0686
Location                (42.30626521, -71.06864556)
Name: 1, dtype: object
```

How to Handle Missing Data?

Handling missing values from the data will **improve the quality** of the data we are using, and in turn, it will yield **accurate analysis**

1. Are there any missing values?

2. Are the missing values significant enough to handle it?

```
crime_data.isnull().sum()
```

```
INCIDENT_NUMBER          0
OFFENSE_CODE             0
OFFENSE_CODE_GROUP       0
OFFENSE_DESCRIPTION      0
DISTRICT              2146
REPORTING_AREA           0
SHOOTING                 0
OCCURRED_ON_DATE         0
YEAR                     0
MONTH                    0
DAY_OF_WEEK              0
HOUR                     0
UCR_PART               109
STREET               12233
Lat                  27378
Long                 27378
Location                 0
dtype: int64
```

Find the count of missing values

As per the missing value report, we can see that around 1723 records have True/Yes rest all the records are missing

```
crime_data.SHOOTING.value_counts(dropna=False)
```

```
NaN     415383
Y         1723
Name: SHOOTING, dtype: int64
```

Find the count of missing values

have first replaced all the missing values with “N” as we concluded before that in this case of “NaN,” it is the same as “N.

```
1  crime_data.SHOOTING.fillna('N', inplace=True)
2  crime_data.SHOOTING.replace({'Y':True, 'N':False}, inplace=True)
```

```
crime_data.SHOOTING.value_counts(dropna=False)
```

```
False    415383
True       1723
Name: SHOOTING, dtype: int64
```

Find the count of missing values

we can see that it is not a binary value. So, figuring out the right missing value is next to impossible. Individually, finding the solution for columns can be fruitful, but here **we will make a general approach to solve the missing value**

```
crime_data.STREET.value_counts(dropna=False)
```

```
WASHINGTON ST          18869
NaN                    12233
BLUE HILL AVE          10347
BOYLSTON ST             9329
DORCHESTER AVE          6584
TREMONT ST              6461
HARRISON AVE            6237
MASSACHUSETTS AVE       6204
CENTRE ST               5773
COMMONWEALTH AVE        5394
HYDE PARK AVE           4635
COLUMBIA RD             4227
HUNTINGTON AVE          3911
RIVER ST                3798
```

What to Do with Duplicate Values?

know that there are many duplicates of the same “INCIDENT_NUMBER” Now we should carefully inspect all the duplicate values and start thinking of a solution.

```
crime_data.INCIDENT_NUMBER.value_counts()
```

```
I152071596        20
I172053750        18
I192025403        15
I162067346        14
I182051210        14
I130041200-00     13
I162030584        13
I182093742        12
I162045234        12
I192008813        12
I152097957        12
I182044546        12
I070720870-00     11
I192062990        11
I130194606-00     11
```

Viewing some of the duplicate records

Records of the incident number “I192009132” which has ten duplicates.

```
crime_data[crime_data.INCIDENT_NUMBER == "I192009132"].head(3).T
```

	55846	55847	55848
INCIDENT_NUMBER	I192009132	I192009132	I192009132
OFFENSE_CODE	1841	111	2010
OFFENSE_CODE_GROUP	Drug Violation	Homicide	HOME INVASION
OFFENSE_DESCRIPTION	DRUGS - POSS CLASS A - INTENT TO MFR DIST DISP	MURDER, NON-NEGLIGIENT MANSLAUGHTER	HOME INVASION
DISTRICT	D4	D4	D4
REPORTING_AREA	273	273	273
SHOOTING	Y	Y	Y
OCCURRED_ON_DATE	2019-02-04 12:35:00	2019-02-04 12:35:00	2019-02-04 12:35:00
YEAR	2019	2019	2019
MONTH	2	2	2
DAY_OF_WEEK	Monday	Monday	Monday
HOUR	12	12	12
UCR_PART	Part Two	Part One	NaN
STREET	NORTHAMPTON ST	NORTHAMPTON ST	NORTHAMPTON ST
Lat	42.3373	42.3373	42.3373
Long	-71.0792	-71.0792	-71.0792
Location	(42.33729692, -71.07919582)	(42.33729692, -71.07919582)	(42.33729692, -71.07919582)

Viewing some of the duplicate records

Here, there will be two categories of features: one which will be **similar** throughout the duplicates and two which will change and will be **inconsistent** throughout the duplicates

```
crime_data[crime_data.INCIDENT_NUMBER == "I192009132"].head(3).T
```

	55846	55847	55848
INCIDENT_NUMBER	I192009132	I192009132	I192009132
OFFENSE_CODE	1841	111	2010
OFFENSE_CODE_GROUP	Drug Violation	Homicide	HOME INVASION
OFFENSE_DESCRIPTION	DRUGS - POSS CLASS A - INTENT TO MFR DIST DISP	MURDER, NON-NEGLIGIENT MANSLAUGHTER	HOME INVASION
DISTRICT	D4	D4	D4
REPORTING_AREA	273	273	273
SHOOTING	Y	Y	Y
OCCURRED_ON_DATE	2019-02-04 12:35:00	2019-02-04 12:35:00	2019-02-04 12:35:00
YEAR	2019	2019	2019
MONTH	2	2	2
DAY_OF_WEEK	Monday	Monday	Monday
HOUR	12	12	12
UCR_PART	Part Two	Part One	NaN
STREET	NORTHAMPTON ST	NORTHAMPTON ST	NORTHAMPTON ST
Lat	42.3373	42.3373	42.3373
Long	-71.0792	-71.0792	-71.0792
Location	(42.33729692, -71.07919582)	(42.33729692, -71.07919582)	(42.33729692, -71.07919582)

Viewing some of the duplicate records

"OFFENSE_DESCRIPTION" is different for all the records but "Location," "OCCURRED_ON_DATE," are the same.

```
crime_data[crime_data.INCIDENT_NUMBER == "I192009132"].head(3).T
```

	55846	55847	55848
INCIDENT_NUMBER	I192009132	I192009132	I192009132
OFFENSE_CODE	1841	111	2010
OFFENSE_CODE_GROUP	Drug Violation	Homicide	HOME INVASION
OFFENSE_DESCRIPTION	DRUGS - POSS CLASS A - INTENT TO MFR DIST DISP	MURDER, NON-NEGLIGIENT MANSLAUGHTER	HOME INVASION
DISTRICT	D4	D4	D4
REPORTING_AREA	273	273	273
SHOOTING	Y	Y	Y
OCCURRED_ON_DATE	2019-02-04 12:35:00	2019-02-04 12:35:00	2019-02-04 12:35:00
YEAR	2019	2019	2019
MONTH	2	2	2
DAY_OF_WEEK	Monday	Monday	Monday
HOUR	12	12	12
UCR_PART	Part Two	Part One	NaN
STREET	NORTHAMPTON ST	NORTHAMPTON ST	NORTHAMPTON ST
Lat	42.3373	42.3373	42.3373
Long	-71.0792	-71.0792	-71.0792
Location	(42.33729692, -71.07919582)	(42.33729692, -71.07919582)	(42.33729692, -71.07919582)

Viewing some of the duplicate records

We can conclude that there is a **discrepancy** in the data, and we need to pick some more random samples and test the hypothesis.

```
crime_data[crime_data.INCIDENT_NUMBER == "I192009132"].head(3).T
```

	55846	55847	55848
INCIDENT_NUMBER	I192009132	I192009132	I192009132
OFFENSE_CODE	1841	111	2010
OFFENSE_CODE_GROUP	Drug Violation	Homicide	HOME INVASION
OFFENSE_DESCRIPTION	DRUGS - POSS CLASS A - INTENT TO MFR DIST DISP	MURDER, NON-NEGLIGIENT MANSLAUGHTER	HOME INVASION
DISTRICT	D4	D4	D4
REPORTING_AREA	273	273	273
SHOOTING	Y	Y	Y
OCCURRED_ON_DATE	2019-02-04 12:35:00	2019-02-04 12:35:00	2019-02-04 12:35:00
YEAR	2019	2019	2019
MONTH	2	2	2
DAY_OF_WEEK	Monday	Monday	Monday
HOUR	12	12	12
UCR_PART	Part Two	Part One	NaN
STREET	NORTHAMPTON ST	NORTHAMPTON ST	NORTHAMPTON ST
Lat	42.3373	42.3373	42.3373
Long	-71.0792	-71.0792	-71.0792
Location	(42.33729692, -71.07919582)	(42.33729692, -71.07919582)	(42.33729692, -71.07919582)

What to Do with Duplicate Values?

We only have approx. 12% of duplicate data, and considering the data size, we can see it is not **statistically significant**. So, we can either keep or remove the duplicates because the change won't impact the analysis significantly

```
print("Unique: " + str(crime_data.INCIDENT_NUMBER.unique().__len__()))
print("Total Count: " + str(crime_data.INCIDENT_NUMBER.count()))
```

```
Unique: 367158
Total Count: 417106
```

```
# Percentage of Duplicates
((crime_data.INCIDENT_NUMBER.count() - crime_data.INCIDENT_NUMBER.unique().__len__())\
/ crime_data.INCIDENT_NUMBER.count()) * 100
```

```
11.974893672112124
```

Dropping duplicates

So, the strategy to drop the duplicates is to treat the first duplicate record as unique and drop the rest of the records.

```
crime_data.drop_duplicates(subset="INCIDENT_NUMBER",
                           inplace=True,
                           keep='first')
```

```
crime_data.shape
```

```
(367158, 17)
```

1. What is Data?
2. O.S.E.M.N Framework
3. Exploratory Data Analysis
4. Investigating The Data
5. Hypothesis Generation and Validation
6. More Data Visualization Example

Exploratory Data Analysis (EDA)

Data Exploring is a combination of Art and Science. We need to ask the **right kind of question** and use the **right tool** to analyze the results.

Data Analysis is a process of **inspecting**, **cleansing**, **transforming**, and **modeling** data **to find new insights, draw conclusions, and supporting decision-making**

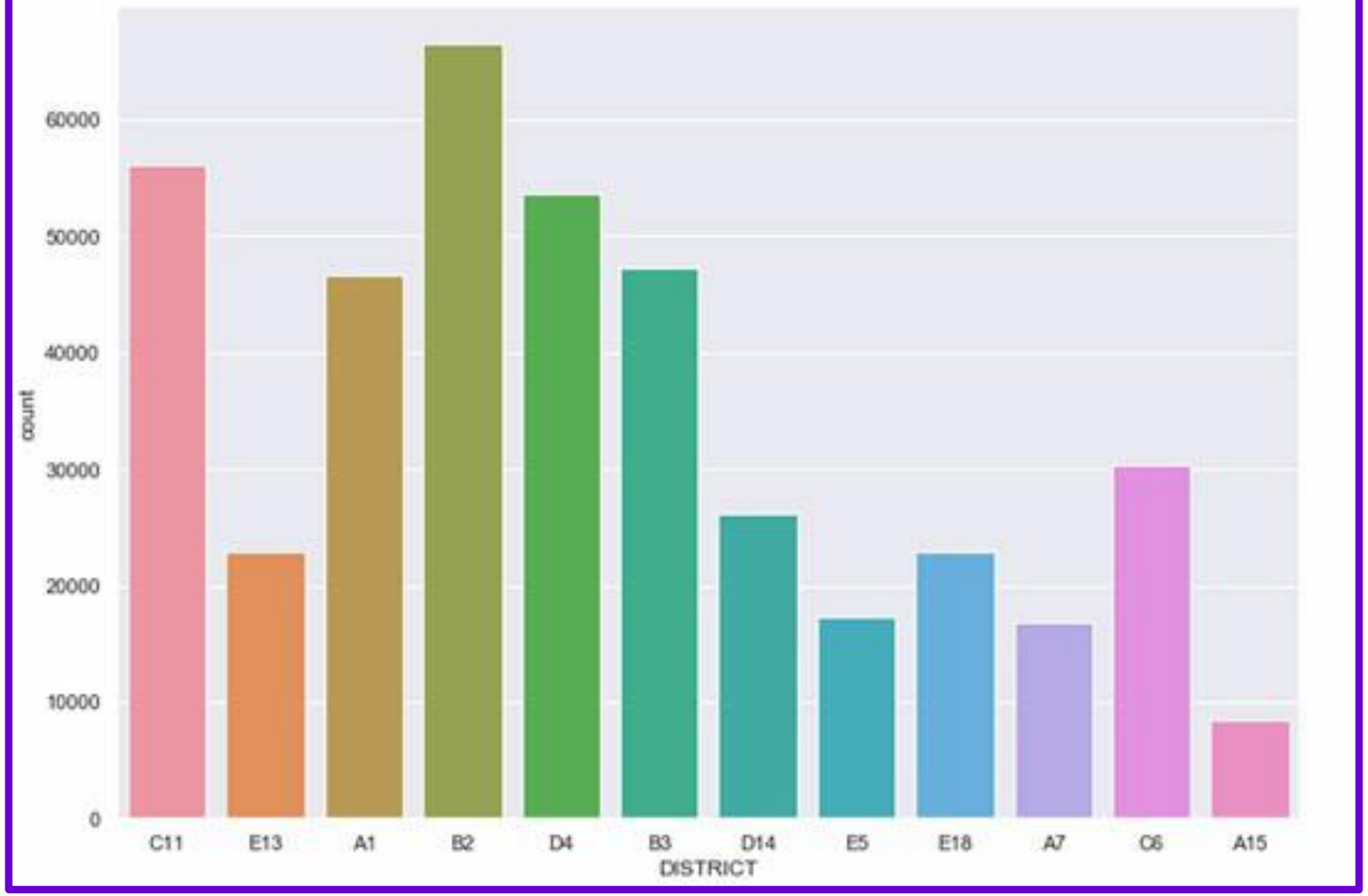
Exploratory Data Analysis (EDA) is an approach/philosophy to analyze a given data and **derive information about the characteristics** of the data **using Graphical Visualizations**

EDA: Univariate Feature Analysis (Discrete Distribution)

How the frequency of every unique observation looks over the sample space

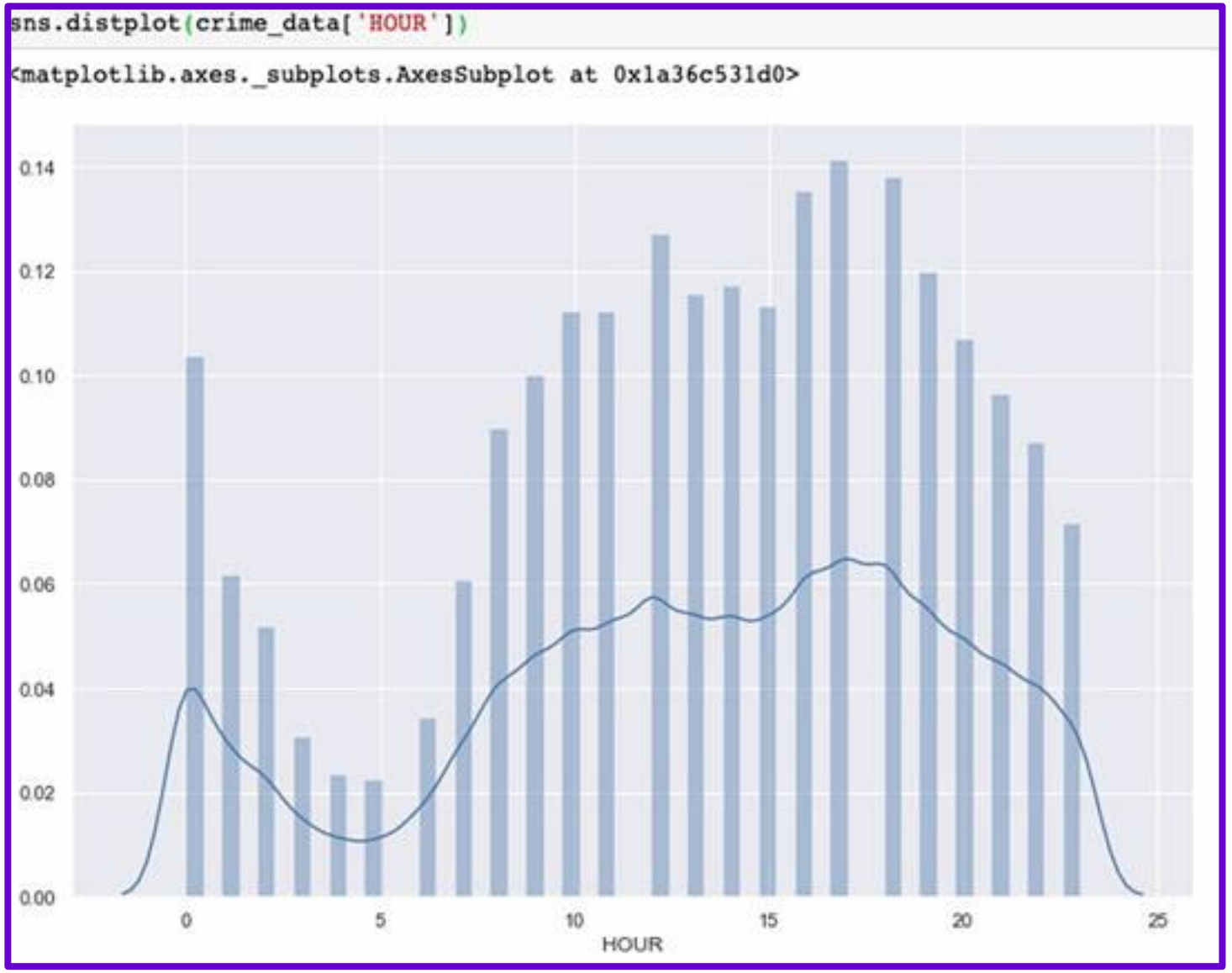
```
sns.countplot(data=crime_data,
              x='DISTRICT'
             )
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a21ec3978>
```



EDA: Univariate Feature Analysis (Continues Distribution)

We can infer that crime rates are highest from 16:00 hour to 20:00 hour, and again there is a rise in crime rates at midnight 00:00 hours. As this is considering the entire dataset, we can create a hypothesis that each year will have a similar distribution.

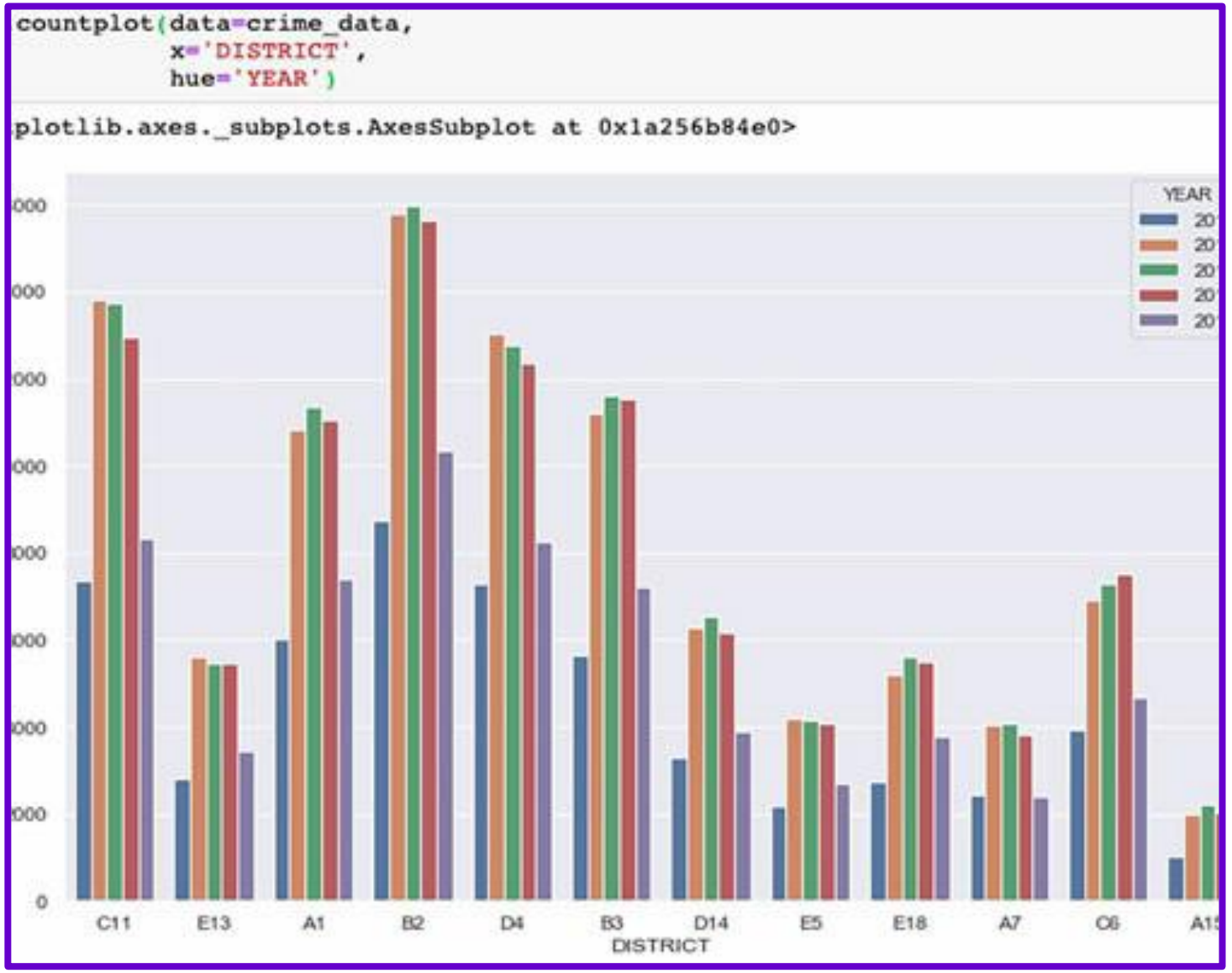


EDA: Multivariate Feature Analysis

To analyze more than one feature at the same time, we have multiple techniques like Regression Analysis, Cluster Analysis, Correlation Analysis, etc.

```
countplot(data=crime_data,
          x='DISTRICT',
          hue='YEAR')
```

```
plotlib.axes._subplots.AxesSubplot at 0x1a256b84e0>
```



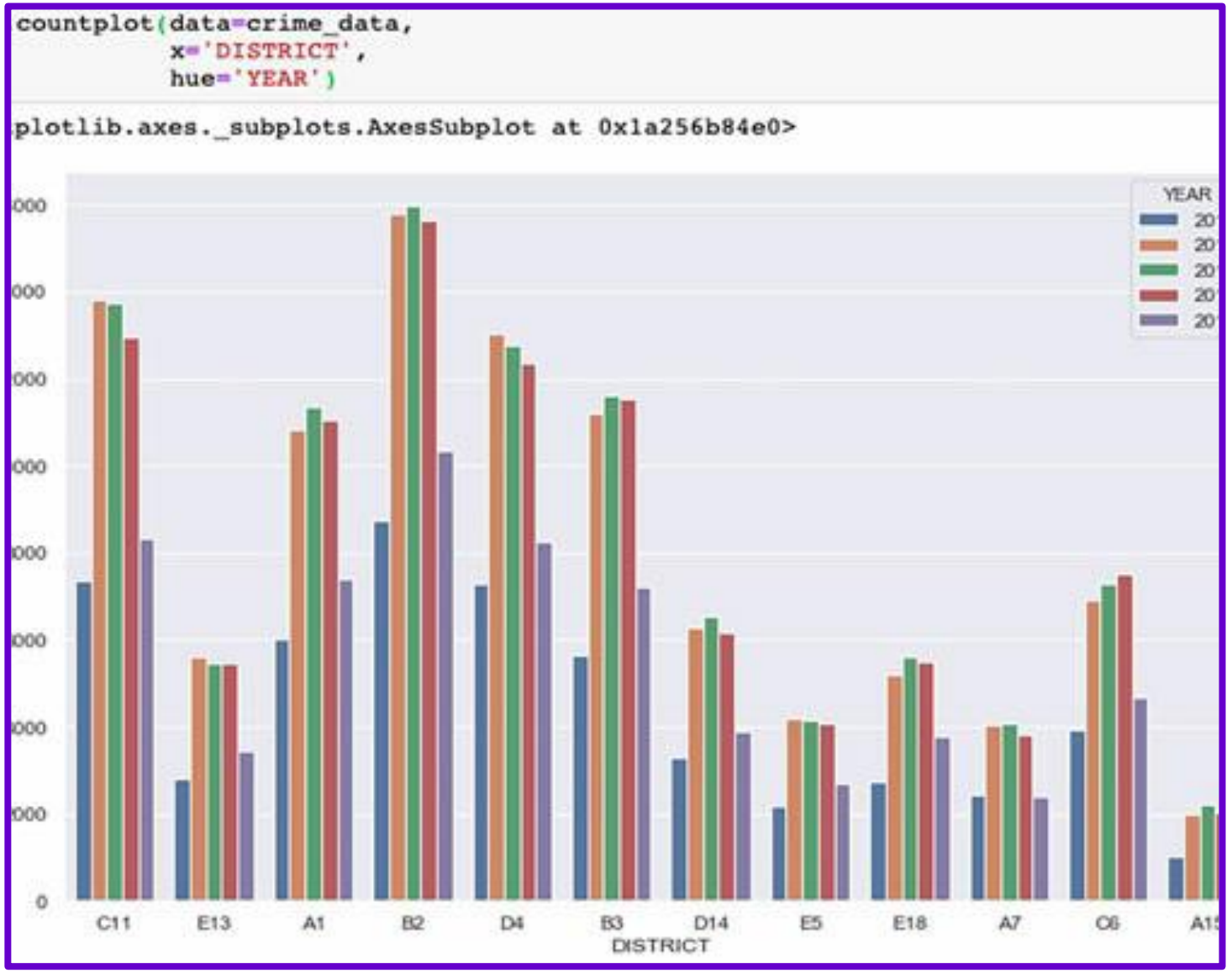
1. What is Data?
2. O.S.E.M.N Framework
3. Exploratory Data Analysis
4. Investigating The Data
5. Hypothesis Generation and Validation
6. More Data Visualization Example

Investigating The Data

From the chart, we can see there is a spike of Crime from the Year 2015 to the Year 2016 for all the Districts. And there is a sudden decrease in Crime rates in 2019.

```
countplot(data=crime_data,
          x='DISTRICT',
          hue='YEAR')
```

```
plotlib.axes._subplots.AxesSubplot at 0x1a256b84e0>
```



Minimum and maximum values

If we see the minimum date “2015-06-15 00:00:00”, we can see that approximately the first six months we do not have any data, and similarly for the maximum date “2019-08-28 21:00:00” we only have data until August.

```
min(crime_data.OCCURRED_ON_DATE)
```

```
'2015-06-15 00:00:00'
```

```
max(crime_data.OCCURRED_ON_DATE)
```

```
'2019-08-28 21:00:00'
```

Investigating The Data

Seeing the partial data, we can drop all the records for the years 2015 and 2019. After dropping, we will have a complete set of data that can be analyzed annually.

```
crime_data = crime_data[crime_data["YEAR"].isin([2016,2017,2018])]
```

```
crime_data.YEAR.value_counts(dropna=False)
```

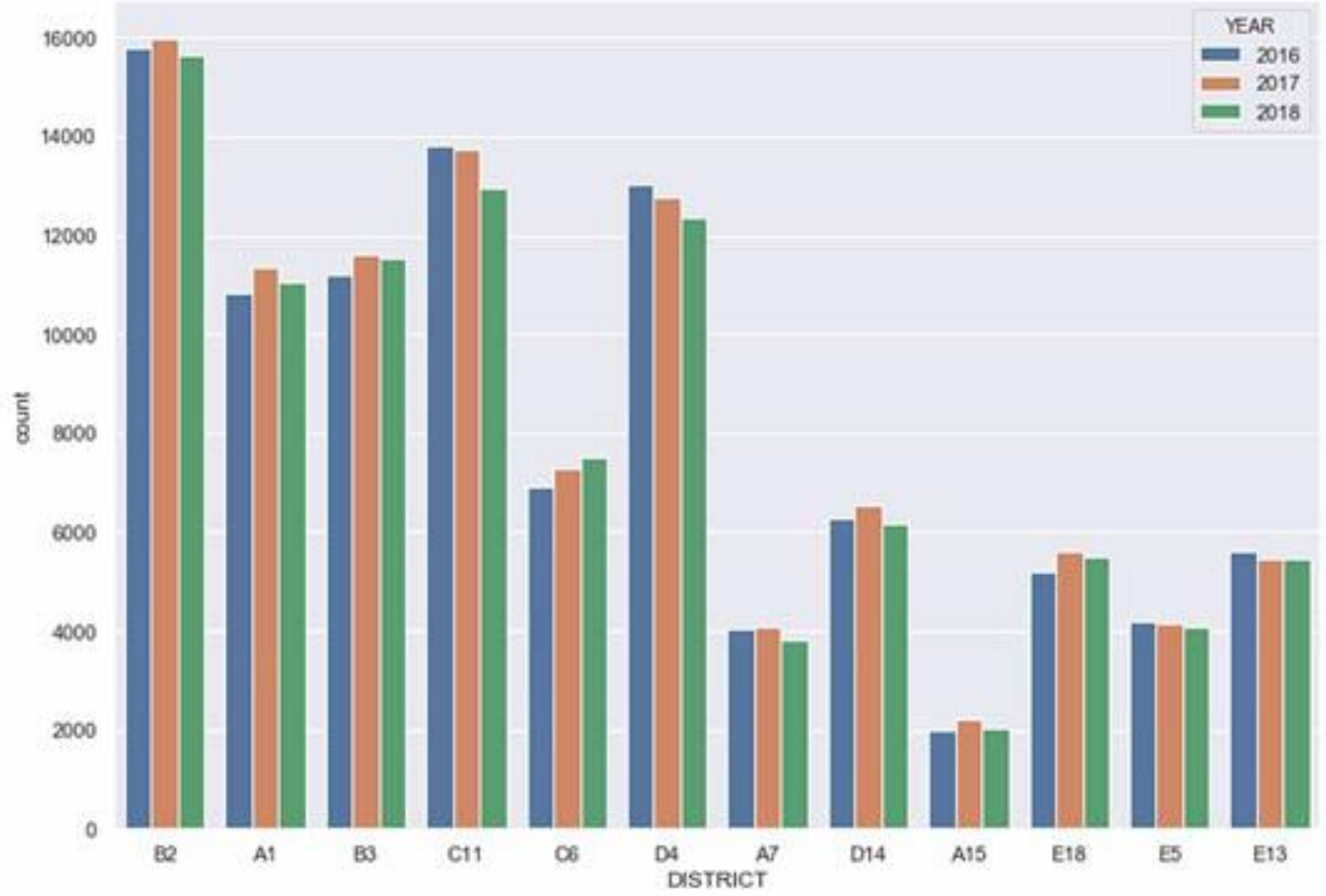
```
2017    101317
2016     99415
2018     98808
Name: YEAR, dtype: int64
```

Investigating The Data

With the complete details, we now expect there won't be any change in frequency, as we had seen for years 2015 and 2019.

```
sns.countplot(data=crime_data,
              x='DISTRICT',
              hue='YEAR')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2bc39d68>
```

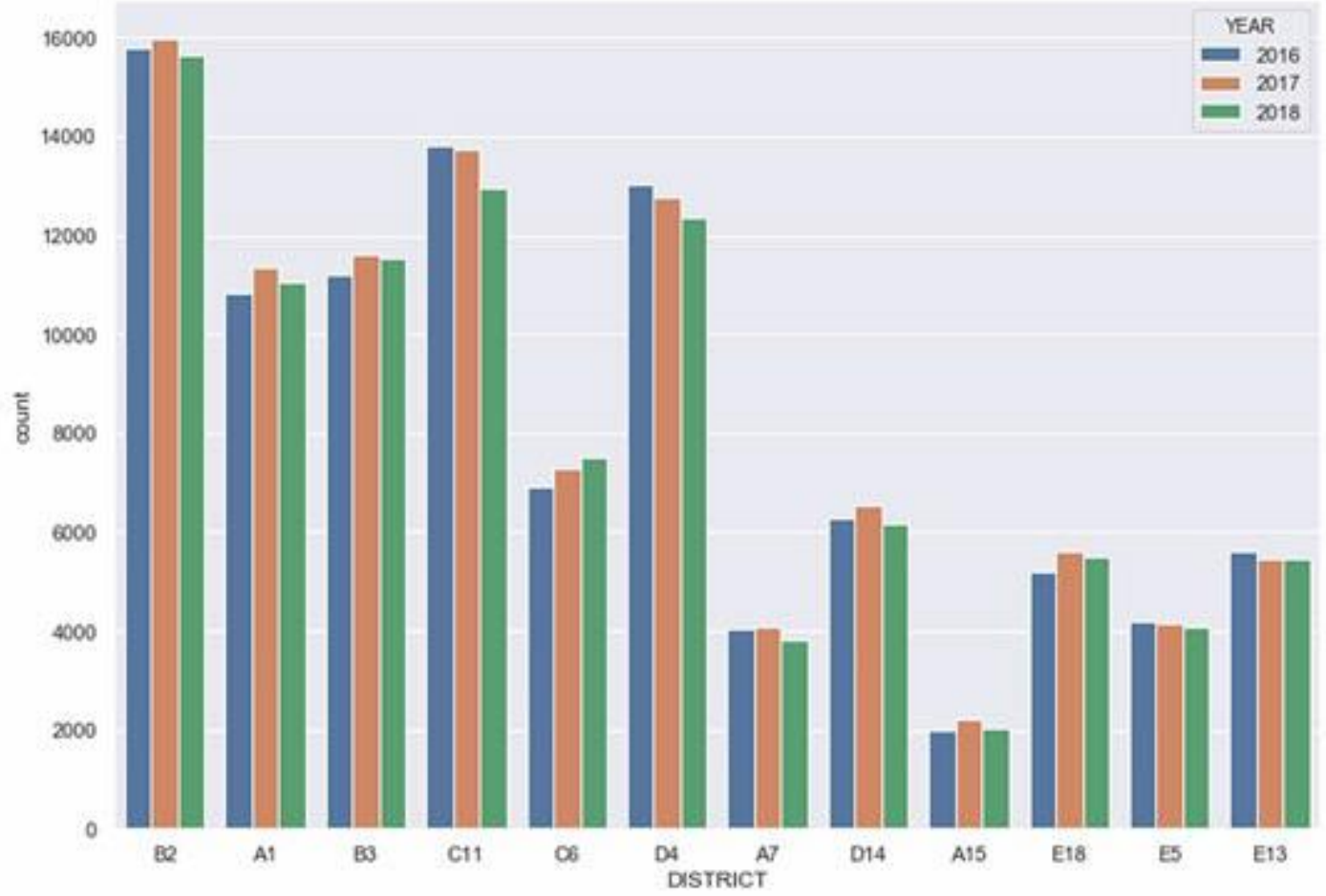


Investigating The Data

With the complete details, we now expect there won't be any change in frequency, as we had seen for years 2015 and 2019.

```
sns.countplot(data=crime_data,
              x='DISTRICT',
              hue='YEAR')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2bc39d68>
```



Investigating The Data

Frequency Table before dropping data on 2015 and 2019

```
crime_data.DISTRICT.value_counts()

B2      66506
C11     56172
D4      53707
B3      47210
A1      46659
C6      30321
D14     26125
E18     22852
E13     22814
E5      17338
A7      16781
A15      8475
Name: DISTRICT, dtype: int64
```

Frequency Table after dropping data on 2015 and 2019

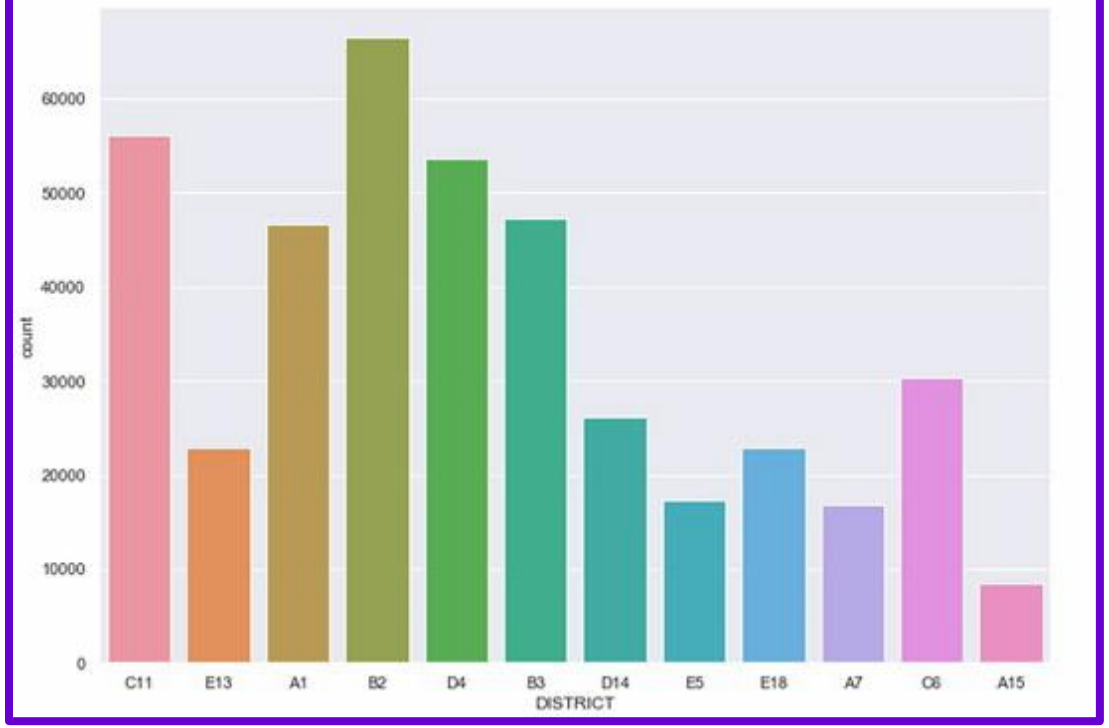
```
crime_data.DISTRICT.value_counts()

B2      47424
C11     40492
D4      38153
B3      34353
A1      33239
C6      21713
D14     18967
E13     16550
E18     16329
E5      12451
A7      11949
A15      6218
Name: DISTRICT, dtype: int64
```

Investigating The Data

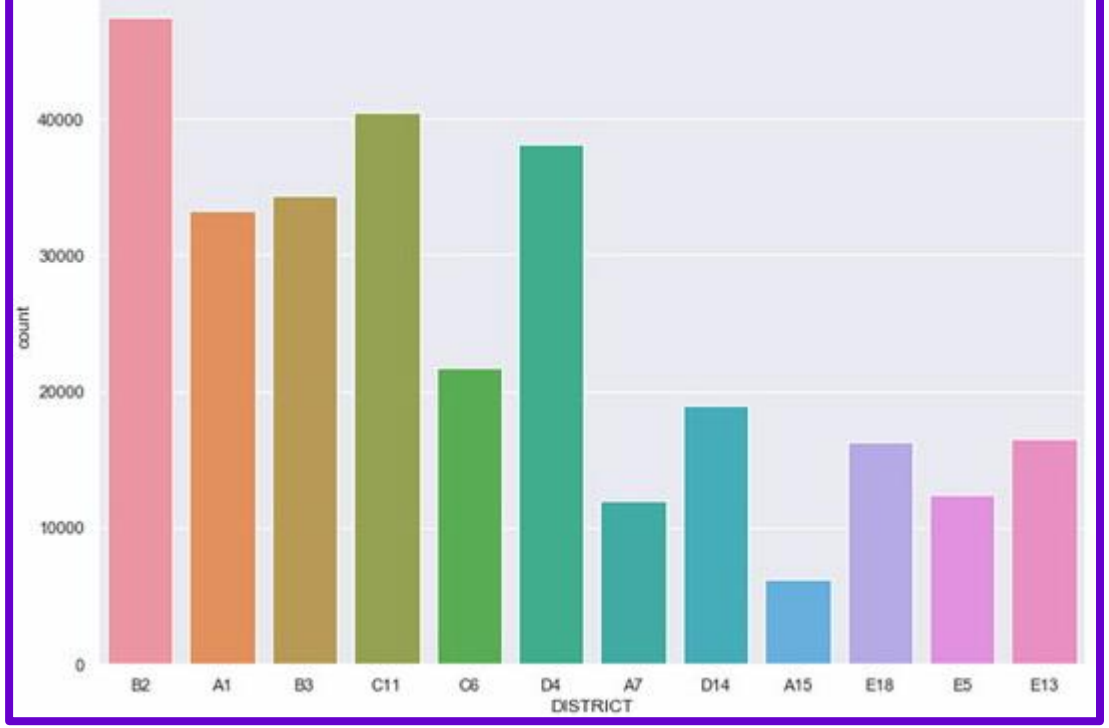
Bar Chart before dropping data on 2015 and 2019

```
sns.countplot(data=crime_data,
              x='DISTRICT'
             )
<matplotlib.axes._subplots.AxesSubplot at 0x1a21ec3978>
```



Bar Chart after dropping data on 2015 and 2019

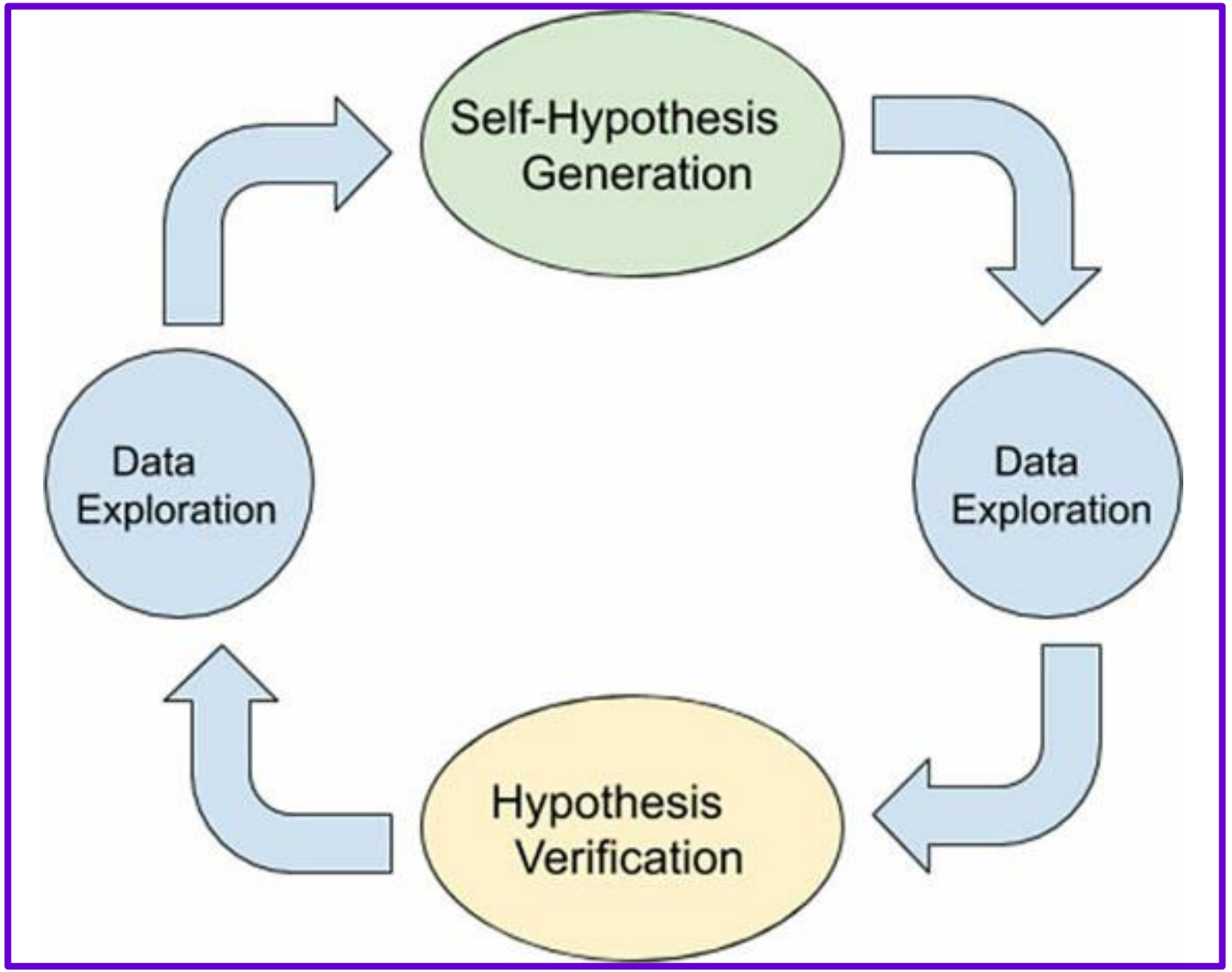
```
sns.countplot(data=crime_data,
              x='DISTRICT')
<matplotlib.axes._subplots.AxesSubplot at 0x1a29673cc0>
```



1. What is Data?
2. O.S.E.M.N Framework
3. Exploratory Data Analysis
4. Investigating The Data
5. Hypothesis Generation and Validation
6. More Data Visualization Example

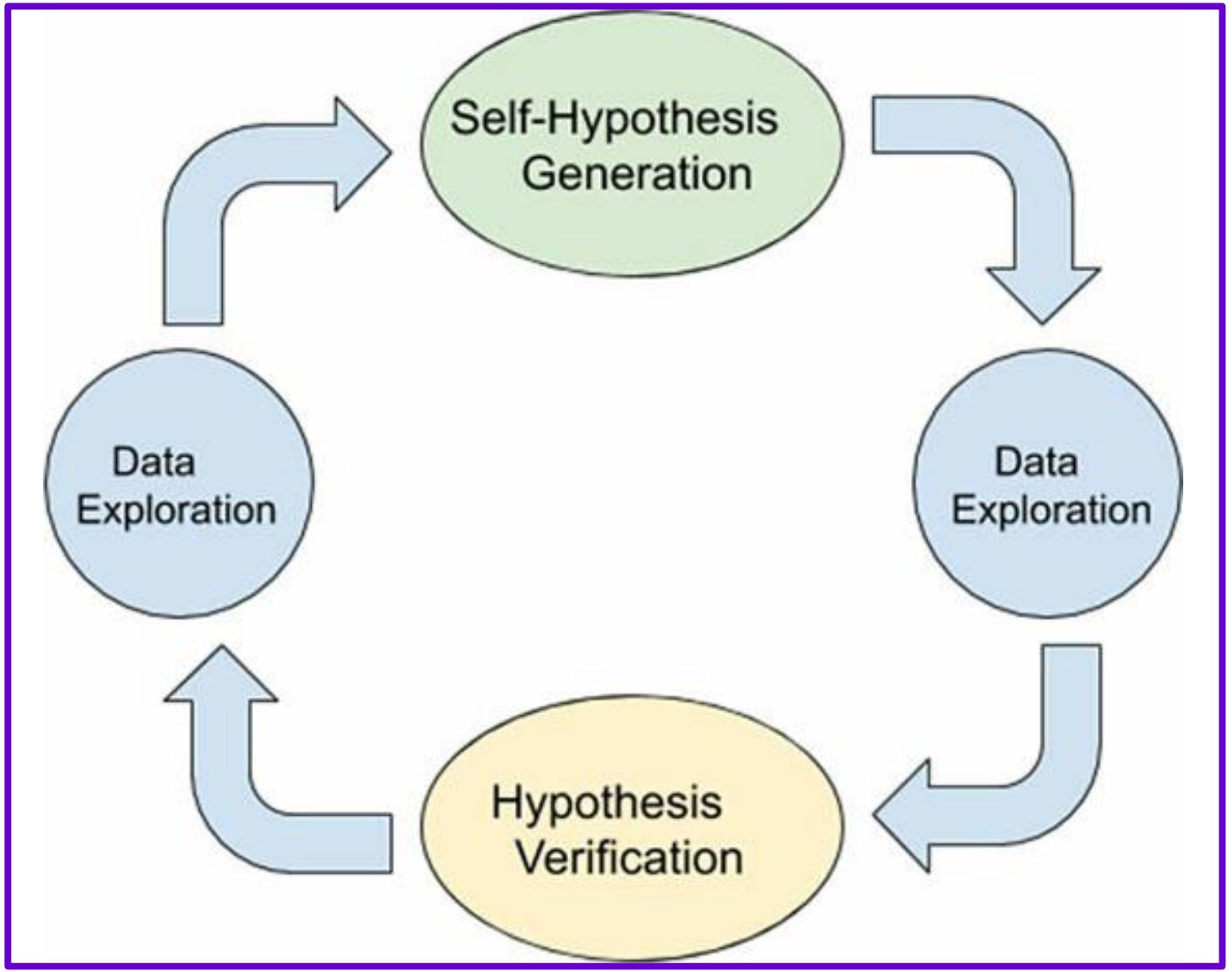
Iterative process of data exploring

We should always keep in mind that Data Exploring is an iterative process



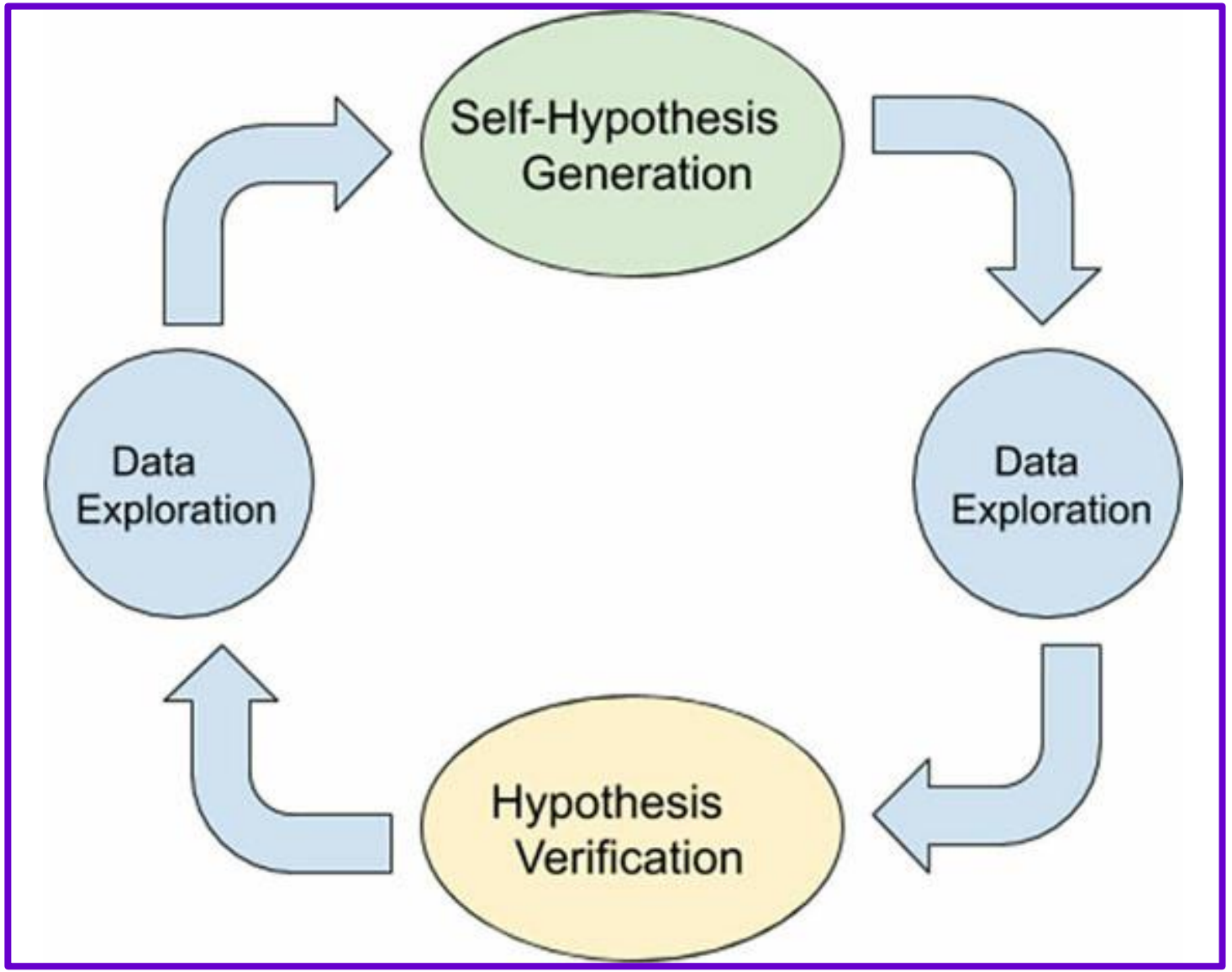
Iterative process of data exploring (1. Hypothesis)

A Hypothesis is a theory that is proposed with some **limited knowledge** of the data. For example, after seeing the data, we can guess that a particular feature has certain characteristics, or it will behave in a particular way.



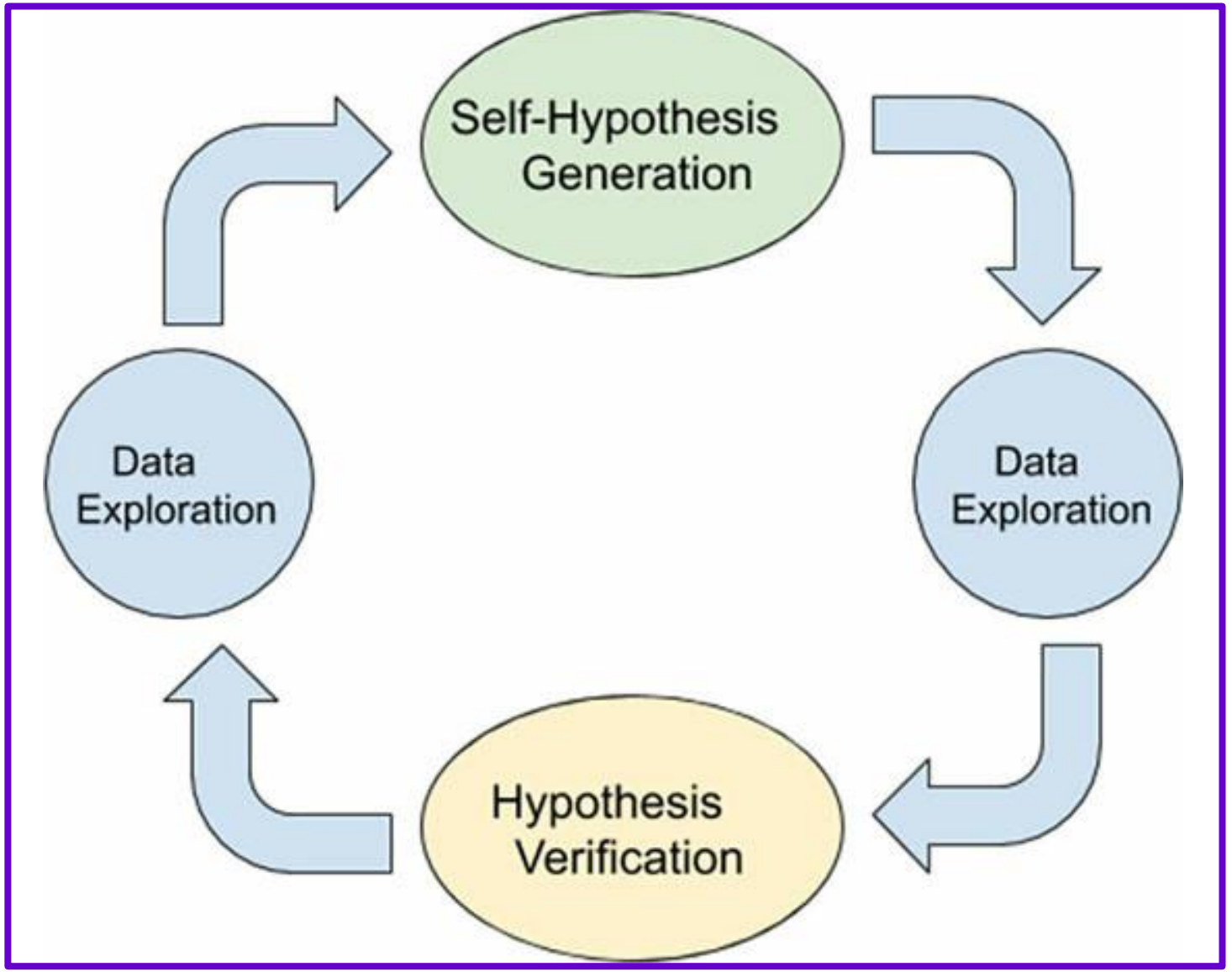
Iterative process of data exploring
(Hypothesis Generation)

A Hypothesis is a theory that is proposed with some **limited knowledge** of the data. For example, after seeing the data, we can guess that a particular feature has certain characteristics, or it will behave in a particular way.



Iterative process of data exploring (Hypothesis Validation)

When we prove or validate or have concrete evidence to support our hypothesis, the process is known as **hypothesis validation**.



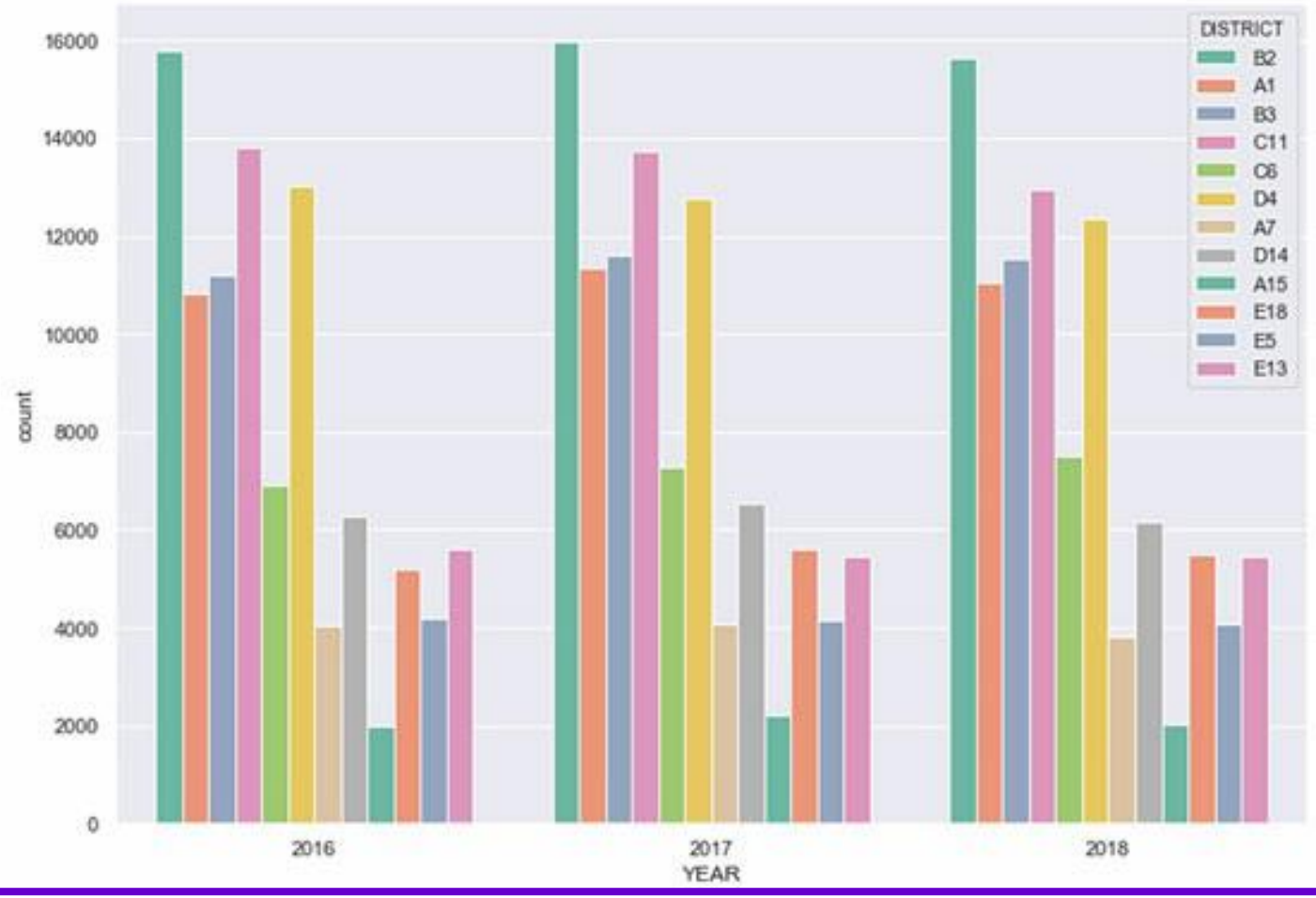
1. What is Data?
2. O.S.E.M.N Framework
3. Exploratory Data Analysis
4. Investigating The Data
5. Hypothesis Generation and Validation
6. More Data Visualization Example

Examples of visualization to know more about the data

Count distribution of crime rates for different districts and year

```
sns.countplot(data=crime_data,
              x='YEAR',
              hue='DISTRICT', palette="Set2")
```

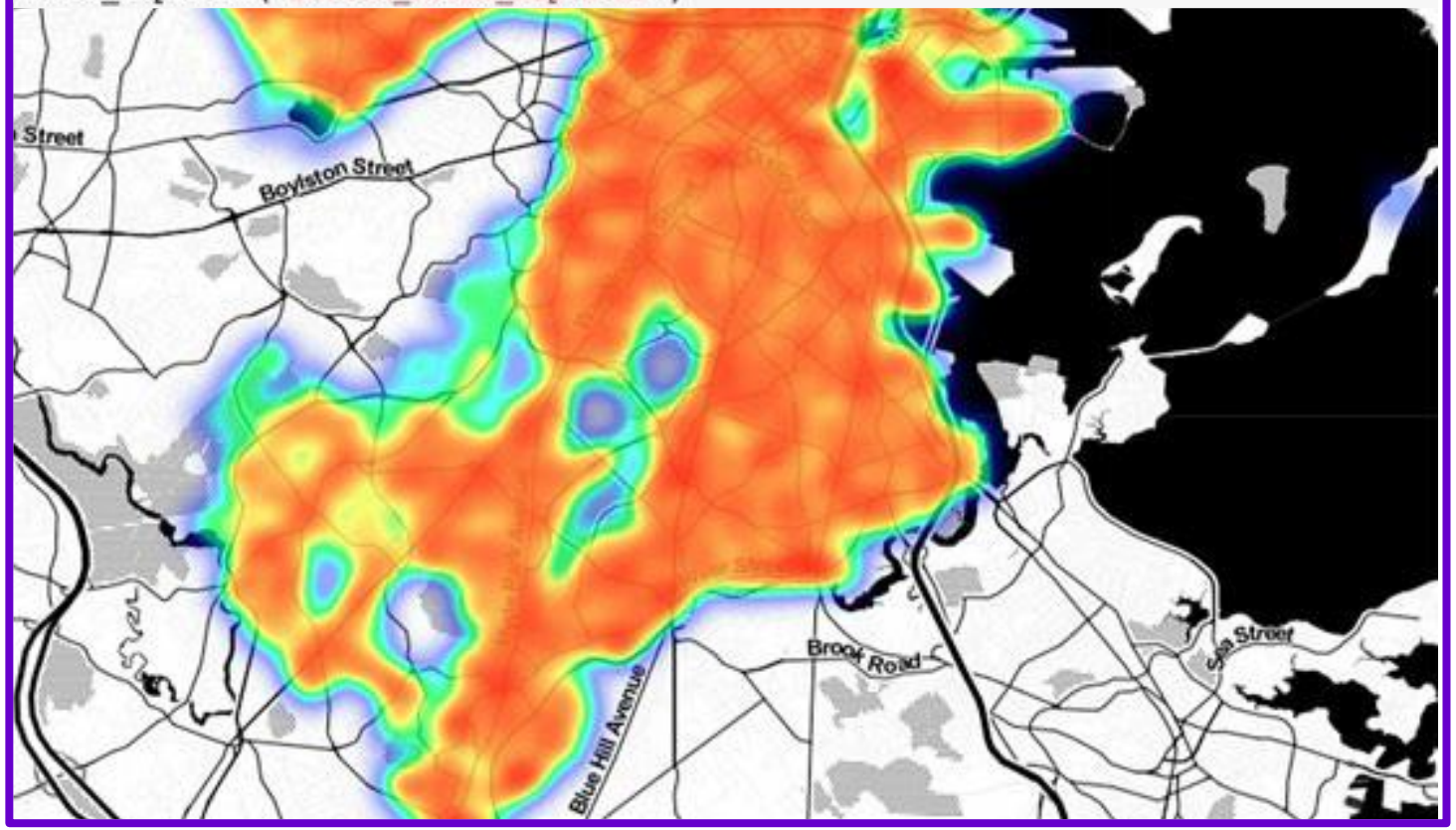
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a2488cc50>
```



Examples of visualization to know more about the data

As “Lat” and “Long” stand for Latitude and Longitude respectively, those values from the dataset can be used to plot a map with the count of crimes for each location.

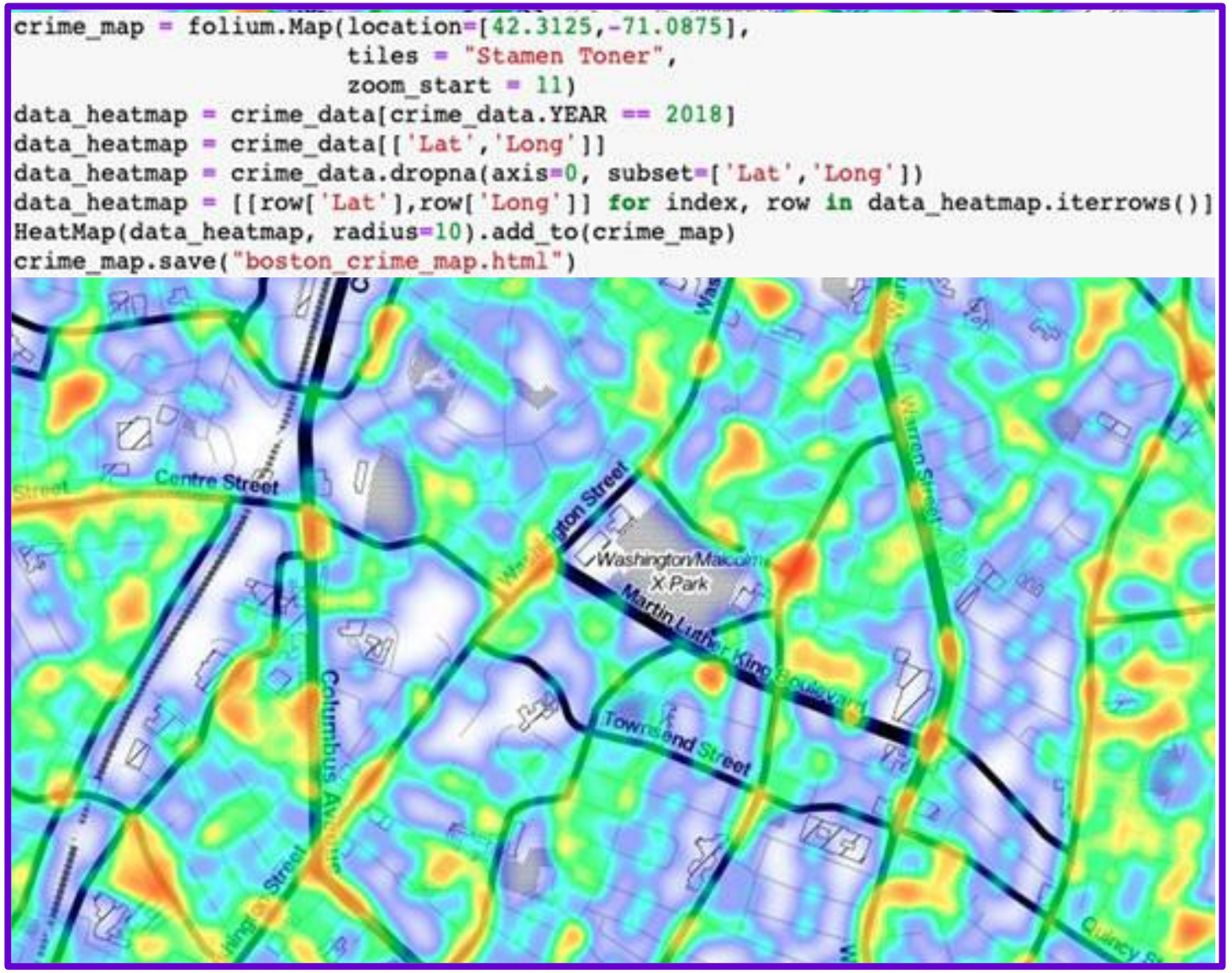
```
crime_map = folium.Map(location=[42.3125,-71.0875],
                       tiles = "Stamen Toner",
                       zoom_start = 11)
data_heatmap = crime_data[crime_data.YEAR == 2018]
data_heatmap = crime_data[['Lat','Long']]
data_heatmap = crime_data.dropna(axis=0, subset=['Lat','Long'])
data_heatmap = [[row['Lat'],row['Long']] for index, row in data_heatmap.iterrows()]
HeatMap(data_heatmap, radius=10).add_to(crime_map)
crime_map.save("boston_crime_map.html")
```



Examples of visualization to know more about the data

As “Lat” and “Long” stand for Latitude and Longitude respectively, those values from the dataset can be used to plot a map with the count of crimes for each location.

```
crime_map = folium.Map(location=[42.3125,-71.0875],
                       tiles = "Stamen Toner",
                       zoom_start = 11)
data_heatmap = crime_data[crime_data.YEAR == 2018]
data_heatmap = crime_data[['Lat','Long']]
data_heatmap = crime_data.dropna(axis=0, subset=['Lat','Long'])
data_heatmap = [[row['Lat'],row['Long']] for index, row in data_heatmap.iterrows()]
HeatMap(data_heatmap, radius=10).add_to(crime_map)
crime_map.save("boston_crime_map.html")
```



Course References

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson, 2021.

[2] T. Ghosh and S. K. B. Math, *Practical Mathematics for AI and Deep Learning: A Concise yet In-Depth Guide on Fundamentals of Computer Vision, NLP, Complex Deep Neural Networks and Machine Learning (English Edition)*. BPB Publications, 2022.

[3] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge University Press, 2020.

[4] T. V. Geetha and S. Sendhilkumar, *Machine Learning: Concepts, Techniques and Applications*. CRC Press LLC, 2023.

[5] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2023.

[6] O. Theobald, *Machine Learning for Absolute Beginners: A Plain English Introduction (Third Edition)*. Scatterplot Press, 2021.

Accessing Course Resource

linkedin.com/in/Samanipour

t.me/SamaniGroup

github.com/Samanipour