CIS 4560-01

Group 3

Title: " Tutorial: A Comprehensive Tempo-Spatial Analysis of Chicago Crime Data: 01/01/2001 to 22/07/2020"

Members:

Jerry Petty

Erik Pacheco

Mayowa Toyinbo

Francisco Figueroa

Gerardo Camarena

Eric Ventura

Instructor:

Jongwook Woo

Date:

05/19/2024

Institution:

California State University, Los Angeles

Objectives:

- Analyze temporal trends in Chicago crime incidents from January 1, 2001, to July 22, 2020.
- Explore spatial patterns of crime incidents across different neighborhoods and areas within Chicago.
- Investigate the distribution of crime incidents based on types of crime, time of occurrence, and location.
- Provide insights into factors influencing crime rates and patterns over time and space.

Platform Specifications:

- Hadoop Distributed File System (HDFS): Utilized for storing and managing the dataset of Chicago Crime from 2001 to 2020.
- Apache Hive: Used for querying and analyzing structured data within Hadoop.
- Beeline: Employed as the interactive shell for Hive to execute queries and commands.
- Additional Tools/Libraries:
    - Tableau: Utilized for creating interactive visualizations and dashboards to analyze and present the data effectively.
    - Microsoft Excel Map 3D: Employed for spatial visualization and analysis of crime data, enhancing our understanding of spatial patterns.

Step 1: Uploading the Dataset to Hadoop

Purpose: To download the Chicago Crime dataset from Kaggle and upload it to the Hadoop file system (HDFS).

1. **Create Kaggle Account**:
    - Go to Kaggle.
    - Sign up for a new account if you don't have one already.
2. **Download Dataset from Kaggle**:
    - Visit the dataset page using the link provided: Chicago Crime Dataset.
    - Click on the "Download" button to download the dataset zip file.
3. **Unzip the Archive**:
    - Once the download is complete, locate the downloaded zip file on your local machine.
    - Unzip the **archive.zip** file to extract its contents.
4. **Move the CSV File**:
    - Navigate to the extracted folder containing the dataset.
    - Find the file named **Crimes_-_2001_to_Present.csv**.
    - Move this CSV file to your Hadoop user directory. Replace **/user/home** with your actual Hadoop user directory path.

Step 2: Securely Copying Files to Hadoop Cluster

Purpose: To transfer the Chicago Crime dataset from your local machine to the Hadoop server using SCP (Secure Copy Protocol).

1. **Open a Terminal:**
    - **Open a terminal on your local machine.**
2. **Enter SCP Command:**
    - **Enter the following SCP command in the terminal:**

      scp /Users/mayowatoyinbo/Crimes_-_2001_to_Present.csv username@IP_address_or_hostname:/home/username/

    - Replace **/Users/mayowatoyinbo/Crimes_-_2001_to_Present.csv** with the full path to the CSV file on your local machine.

- Replace **<username>** with your username on the Hadoop server.
- Replace **<IP_address_or_hostname>** with the IP address or hostname of your Hadoop server.
- Ensure that you have the necessary permissions to access the destination directory on the Hadoop server.

3. **Enter Password:**
    - You will be prompted to enter your password for the Hadoop server. Enter it and press Enter.
4. **File Transfer:**
    - Once the command is executed successfully, the file will be securely copied from your local machine to the specified directory on the Hadoop server.

Now, the Chicago Crime dataset has been securely transferred to the Hadoop server and is ready for further processing and analysis.

Step 3: Connect to Hadoop Cluster and Upload Dataset to HDFS

Purpose: Connect to the Hadoop cluster via SSH and upload the Chicago Crime dataset to HDFS.

1. **Open Another Terminal**:
    - Open a new terminal on your local machine.
2. **Connect to the Hadoop Cluster via SSH**:
    - Enter the following command to connect to the Hadoop server via SSH:

      ssh <username>@<IP_address_or_hostname>

    - Replace **<username>** with your username on the Hadoop server.
    - Replace **<IP_address_or_hostname>** with the IP address or hostname of your Hadoop server.

    For example:

      ssh ffigue10@129.146.148.35

    - Enter your password when prompted.
3. **Create a Directory in HDFS:**
    - Once connected to the Hadoop cluster, create a directory in HDFS using the following command:

hdfs dfs -mkdir Crimes

4. **List HDFS Contents**:
   - Check the contents of HDFS to verify that the directory was created successfully:

     hdfs dfs -ls

5. **Upload the Dataset to HDFS:**
   - Use the following command to upload the CSV file to the newly created directory in HDFS:

     hdfs dfs -put Crimes_-_2001_to_Present.csv Crimes/

6. **Confirm File Upload:**
   - Check the contents of the **Crimes/** directory in HDFS to confirm that the file was uploaded successfully:

     hdfs dfs -ls Crimes/

Now, the Chicago Crime dataset has been uploaded to HDFS and is ready for further processing and analysis.

Step 4: Open Beeline

Purpose: Launch Beeline to connect to the Hive server for SQL querying.

1. **Open Beeline**:
   - In your terminal, simply type the command:

     Beeline

   - Press Enter.
2. **Connection to Hive Server**:
   - Beeline will establish a connection to the default Hive server.
3. **Connected to Beeline**:
   - Once connected, you will see the Beeline prompt, indicating that you are connected to the Hive server and ready to execute Hive queries.

Now, you are ready to execute SQL queries using Beeline to analyze the Chicago Crime dataset in Hive.

Step 5: Run show tables;

Purpose: To display the list of tables available in the current Hive database.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

      show tables;

    - Press Enter to execute the command.
2. **View Table List**:
    - Beeline will display a list of tables available in the current Hive database, if any.
3. **Interpretation**:
    - This command helps you to see what tables are available in the Hive database. If you've already created a table for the Chicago Crime dataset, it should appear in this list.

Now, you can proceed with further querying or analysis based on the available tables in the Hive database.

Step 6: Run the DROP TABLE

Purpose: To drop the table named **crime** if it exists in the current Hive database.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

      DROP TABLE IF EXISTS crime;

    - Press Enter to execute the command.
2. **Confirmation**:
    - If the **crime** table exists, it will be dropped. If it doesn't exist, nothing will happen.
    - Beeline will display a confirmation message indicating whether the table was dropped or not.
3. **Interpretation**:
    - This command ensures that any existing **crime** table is removed before creating a new one. This is useful to avoid conflicts or errors when re-creating tables.

Now, you are ready to proceed with further operations on the **crime** table in Hive.

Step 7: CREATE EXTERNAL TABLE: Crime

Purpose: To create an external table named **crime** in Hive to store the Chicago Crime dataset.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

```
CREATE EXTERNAL TABLE IF NOT EXISTS crime(
    ID INT,
    Case_Number STRING,
    Date_Time STRING,
    Block STRING,
    IUCR STRING,
    Primary_Type STRING,
    Description STRING,
    Location_Description STRING,
    Arrest BOOLEAN,
    Domestic BOOLEAN,
    Beat INT,
    District INT,
    Ward INT,
    Community_Area INT,
    FBI_Code STRING,
    X_Coordinate INT,
    Y_Coordinate INT,
    Year STRING,
    Updated_On TIMESTAMP,
    Latitude DOUBLE,
    Longitude DOUBLE,
    Location STRING
)
ROW FORMAT DELIMITED
```

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/username/Crimes'

TBLPROPERTIES ('skip.header.line.count'='1');

- Press Enter to execute the command.

2. **Interpretation**:
    - This command creates an external table named **crime** in Hive.
    - The table schema is defined with various columns corresponding to the attributes in the Chicago Crime dataset.
    - Data in the table is delimited by commas (**,**).
    - The data files are stored as text files in the specified location in HDFS (**/user/username/Crimes**).
    - The property **skip.header.line.count** is set to **1** to skip the header line in the CSV file.
    - If the table already exists, it will not be re-created due to the **IF NOT EXISTS** clause.

Now, the **crime** table is ready for querying and analysis in Hive.

Step 8: Run the SELECT Query

Purpose: To retrieve a sample of data from the **crime** table to ensure it was successfully created and loaded.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

      SELECT * FROM crime LIMIT 5;

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command selects all columns (*****) from the **crime** table.
    - The **LIMIT 5** clause ensures that only the first 5 rows are returned, providing a sample of the dataset.
    - You will see the first 5 rows of the Chicago Crime dataset displayed in the result.

3. **Review the Data**:
    - Check the output to verify that the data appears as expected, ensuring that the table was created and loaded correctly.

Now, you have successfully executed a query on the **crime** table in Hive, and you can proceed with further analysis or queries as needed.

Step 9: Preparing for Table Creation - top_crime_types

Purpose: Before creating the table **top_crime_types**, we need to ensure that any existing table with the same name is dropped to avoid conflicts.

1. **Run the Command**:
    - In the Beeline prompt, type the following command to drop the existing **top_crime_types** table:

      ```
      DROP TABLE IF EXISTS top_crime_types;
      ```

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command drops the table **top_crime_types** if it already exists, ensuring a clean slate for the new table creation.
    - If the table doesn't exist, no action is taken.
3. **Note**:
    - Dropping a table deletes all data and metadata associated with that table. Be cautious before executing this command, as it cannot be undone.

With the existing **top_crime_types** table dropped (if it existed), we can now proceed with creating the table and processing the data to identify the top crime types.

Step 10: Create Table top_crime_types

Purpose: To create a table **top_crime_types** that captures the top 15 crime types based on their frequency in the dataset.

Context: By identifying the most prevalent crime types within a given dataset, authorities can prioritize their efforts in addressing specific criminal activities. Creating a table to capture the top 15 crime types based on their frequency provides a concise overview of the predominant criminal behaviors within a particular area or jurisdiction.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

```
CREATE TABLE IF NOT EXISTS top_crime_types
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'Crimes/top_crime_types'
AS
SELECT Primary_Type, COUNT(*) AS crime_count
FROM crime
GROUP BY Primary_Type
ORDER BY crime_count DESC
LIMIT 15;
```

- Press Enter to execute the command.

2. **Interpretation**:
    - This command creates a new table **top_crime_types**.
    - The table will be stored as text files with fields delimited by commas.
    - The table will be created at the specified location **'Crimes/top_crime_types'**.
    - The data for this table is derived from a select query on the **crime** table:
        - It counts occurrences of each crime type (**Primary_Type**).
        - Groups them together.
        - Orders them in descending order based on frequency.
        - Limits the results to the top 15 crime types.
3. **Note**:
    - This command combines table creation with data population in a single step using a **SELECT** statement.
    - The location **'Crimes/top_crime_types'** should be adjusted based on your HDFS directory structure.

      With this command, the **top_crime_types** table will be created and populated with the top 15 crime types based on their frequency.

Step 11: Retrieve Data from top_crime_types Table

Purpose: To retrieve the top 15 crime types along with their frequencies from the **top_crime_types** table.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

      SELECT * FROM top_crime_types LIMIT 15;

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command selects all columns (*****) from the **top_crime_types** table.
    - The **LIMIT 15** clause ensures that only the first 15 rows are returned, representing the top 15 crime types based on frequency.
    - You will see the top 15 crime types along with their corresponding frequencies displayed in the result.
3. **Note**:
    - Ensure that the table **top_crime_types** has been created and populated correctly before executing this command.
    - Review the output to confirm that it matches your expectations and provides meaningful insights into the prevalent crime types.

With this command, you'll be able to see the top 15 crime types and their frequencies, which can be useful for prioritizing law enforcement efforts and understanding the predominant criminal behaviors within the dataset.

Step 12: Check the Contents of HDFS Directory for top_crime_types Table

Purpose: To verify the presence of the **top_crime_types** table in the Hadoop Distributed File System (HDFS).

1. **Run the Command**:
    - In the terminal, type the following command:

      hdfs dfs -ls Crimes/top_crime_types

    - Press Enter to execute the command.
2. **Interpretation**:

- This command lists the contents of the directory where the **top_crime_types** table is stored in HDFS.
- If the table has been successfully created and populated, you should see some files listed.

3. **Note**:
    - If the table was created and populated correctly, you should see files representing the data partitions of the **top_crime_types** table.
    - Each file typically represents a partition of the table's data, depending on the size and configuration.

Executing this command helps to ensure that the **top_crime_types** table has been properly stored in HDFS and is accessible for further analysis or use.

Step 13: Retrieve Data from HDFS

Purpose: To download the data from HDFS to your local file system.

1. **Run the Command**:
    - In the terminal, type the following command:

      ```
      hdfs dfs -get /user/username/Crimes/top_crime_types/000000_0
      top_crime_types.csv
      ```

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command retrieves the data file (usually named **000000_0**) containing the **top_crime_types** table data from HDFS.
    - It saves the data to a local file named **top_crime_types.csv**.
    - Adjust the path and file name as needed based on your HDFS directory structure and preferred local file name.
3. **Note**:
    - After executing this command, you'll have the **top_crime_types.csv** file in your current working directory.
    - You can then use this CSV file for further analysis or visualization using tools like Excel, Python, or R.

Executing this command enables you to work with the data from the **top_crime_types** table locally, allowing for easier analysis and manipulation.

Step 14: Transfer Data from Hadoop to Local Machine

Purpose: To copy the **top_crime_types.csv** file from the Hadoop server to your local machine.

1. **Open Another Terminal:**
    - Open a new terminal on your local machine.
2. **Run the Command**:
    - In the new terminal, type the following command:

      scp username@IP_address_or_hostname:/home/username/top_crime_types.csv .
    - Press Enter to execute the command.
2. **Interpretation**:
    - This command uses SCP (Secure Copy Protocol) to copy the **top_crime_types.csv** file from your Hadoop server to your local machine.
    - Adjust the username, IP address, and file paths as needed based on your Hadoop server configuration and the file location.
3. **Note**:
    - The period (**.**) at the end of the command specifies the current directory as the destination for the copied file.
    - After executing this command, the **top_crime_types.csv** file will be available in the current directory of your local machine.

Executing this command allows you to transfer the **top_crime_types.csv** file from the Hadoop server to your local machine for further analysis or visualization.

Step 15: Preparing for Table Creation - Arrest Analysis Table

Purpose: Before creating the **arrest_analysis** table, we need to drop any existing table with the same name to ensure a clean slate.

Context: This table analyzes the percentage of reported crimes resulting in arrests, providing insights into law enforcement effectiveness.

1. **Run the Command**:
    - In the Beeline prompt, type the following command to drop the existing **arrest_analysis** table:

      DROP TABLE IF EXISTS arrest_analysis;

- Press Enter to execute the command.

2. **Interpretation**:
    - This command drops the **arrest_analysis** table if it already exists.
    - Dropping the table ensures that we start with a fresh table when creating the **arrest_analysis** table.
    - If the table doesn't exist, no action is taken.
3. **Note**:
    - Dropping a table deletes all data and metadata associated with it. Be sure to use this command judiciously.

With this step, we ensure that any existing **arrest_analysis** table is removed, preparing for the creation of a new table for arrest analysis.

Step 16: Create Table arrest_analysis

Purpose: To create a table **arrest_analysis** that analyzes the percentage of reported crimes resulting in arrests.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

```
CREATE TABLE arrest_analysis
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'Crimes/arrest_analysis'
AS
SELECT
   crime.primary_type,
   SUM(CASE WHEN crime.arrest = true THEN 1 ELSE 0 END) AS arrests,
   COUNT(*) AS total_crimes,
   (SUM(CASE WHEN crime.arrest = true THEN 1 ELSE 0 END) / COUNT(*))
* 100 AS arrest_percentage
FROM
```

```
    crime
GROUP BY
    crime.primary_type;
```

- Press Enter to execute the command.

2. **Interpretation**:
    - This command creates a new table **arrest_analysis**.
    - The table will be stored as text files with fields delimited by commas.
    - The table will be created at the specified location **'Crimes/arrest_analysis'**.
    - The data for this table is derived from a select query on the **crime** table:
        - It calculates the total number of arrests and total crimes for each crime type (**primary_type**).
        - It calculates the percentage of arrests for each crime type.
        - The **SUM(CASE WHEN ...)** construct is used to count the number of arrests (**true** values) and total crimes.
        - The **GROUP BY** clause groups the data by **primary_type**.
3. **Note**:
    - Ensure that the **crime** table exists and is properly structured before running this command.
    - Adjust the file location based on your HDFS directory structure.
    - This command combines table creation with data population in a single step using a **SELECT** statement.

With this command, the **arrest_analysis** table will be created, providing insights into the percentage of reported crimes resulting in arrests for different crime types.

Step 17: Retrieve Data from arrest_analysis Table

Purpose: To retrieve data from the **arrest_analysis** table to examine the percentage of reported crimes resulting in arrests for different crime types.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

```
SELECT * FROM arrest_analysis LIMIT 15;
```

- Press Enter to execute the command.

2. **Interpretation**:
    - This command selects all columns (*) from the **arrest_analysis** table.
    - The **LIMIT 15** clause ensures that only the first 15 rows are returned.
    - You will see the top 15 rows of the **arrest_analysis** table displayed in the result, showing the crime type, total number of arrests, total number of crimes, and the percentage of arrests.
3. **Note**:
    - Ensure that the **arrest_analysis** table has been created and populated correctly before executing this command.
    - Review the output to confirm that it provides meaningful insights into the percentage of reported crimes resulting in arrests for different crime types.

Executing this command allows you to examine the data from the **arrest_analysis** table, helping you understand the effectiveness of law enforcement efforts for various types of crimes.

Step 18: Check Contents of HDFS Directory for arrest_analysis Table

Purpose: To verify the presence of the **arrest_analysis** table in the Hadoop Distributed File System (HDFS).

1. **Run the Command**:
    - In the terminal, type the following command:

      hdfs dfs -ls Crimes/arrest_analysis

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command lists the contents of the directory where the **arrest_analysis** table is stored in HDFS.
    - If the table has been successfully created and populated, you should see some files listed.
3. **Note**:
    - If the table was created and populated correctly, you should see files representing the data partitions of the **arrest_analysis** table.

- Each file typically represents a partition of the table's data, depending on the size and configuration.

Executing this command helps to ensure that the **arrest_analysis** table has been properly stored in HDFS and is accessible for further analysis or use.

Step 19: Remove Data Files from HDFS

Purpose: To remove data files from HDFS that were generated as part of the **arrest_analysis** table.

1. **Run the Command**:
    - In the terminal, type the following command:

      hdfs dfs -rm 00000*_0

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command removes the data files (**00000*_0**) generated as part of the **arrest_analysis** table from HDFS.
    - The ***** wildcard ensures that all files starting with **00000** and ending with **_0** are removed.
3. **Note**:
    - Be cautious when using the **rm** command with HDFS, as it permanently deletes data files.
    - Ensure that you're only removing the files you intend to delete.

Executing this command helps to clean up unnecessary data files from HDFS, ensuring efficient use of storage space.

Step 20: Retrieve Data from HDFS and Combine

Purpose: To download data files from HDFS generated as part of the **arrest_analysis** table and combine them into a single CSV file.

1. **Retrieve Data Files**:
    - In the terminal, type the following command to retrieve data files from HDFS:

      hdfs dfs -get /user/username/Crimes/arrest_analysis/00000*_0

    - Press Enter to execute the command. This will download the necessary data files.
2. **Combine Data Files**:

- After downloading the files, use the following command to combine them into a single CSV file:

  cat 000000_0 000001_0 000002_0 000003_0 000004_0 000005_0 000006_0 000007_0 000008_0 000009_0 > arrest_analysis.csv

- Press Enter to execute the command.

3. **Interpretation**:
   - The **hdfs dfs -get** command retrieves the necessary data files from HDFS.
   - The **cat** command is used to concatenate the downloaded files into a single CSV file named **arrest_analysis.csv**.
   - The **>** operator is used to redirect the concatenated output to the **arrest_analysis.csv** file.
4. **Note**:
   - Adjust the file names and paths as necessary based on your HDFS directory structure and the number of data files.
   - Ensure that you are in the correct directory when executing these commands.

Executing these commands will download the data files from HDFS and combine them into a single CSV file named **arrest_analysis.csv**, which you can use for further analysis or visualization.

Step 21: Transfer Data from Hadoop to Local Machine

Purpose: To copy the **arrest_analysis.csv** file from the Hadoop server to your local machine.

1. **Open Another Terminal:**
   - Open a new terminal on your local machine.
2. **Run the Command**:
   - In the new terminal, type the following command:

     scp username@IP_address_or_hostname:/home/username/arrest_analysis.csv .

   - Press Enter to execute the command.
3. **Interpretation**:
   - This command uses SCP (Secure Copy Protocol) to copy the **arrest_analysis.csv** file from your Hadoop server to your local machine.

- Adjust the username, IP address, and file paths as needed based on your Hadoop server configuration and the file location.

4. **Note**:
    - The period (**.**) at the end of the command specifies the current directory as the destination for the copied file.
    - After executing this command, the **arrest_analysis.csv** file will be available in the current directory of your local machine.

Executing this command allows you to transfer the **arrest_analysis.csv** file from the Hadoop server to your local machine for further analysis or visualization.

Step 22: Preparing for Table Creation - Temporal Analysis Table

Purpose: Before creating the **temporal_analysis** table, ensure any existing table with the same name is dropped to avoid conflicts.

Context: This table provides insights into the temporal distribution of crimes by year and month, allowing for the analysis of trends over time.

1. **Run the Command**:
    - In the Beeline prompt, type the following command to drop the existing **temporal_analysis** table:

      DROP TABLE IF EXISTS temporal_analysis;

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command drops the **temporal_analysis** table if it already exists.
    - Dropping the table ensures that we start with a fresh table when creating the **temporal_analysis** table.
    - If the table doesn't exist, no action is taken.
3. **Note**:
    - Dropping a table deletes all data and metadata associated with it. Be sure to use this command judiciously.

With this step, we ensure that any existing **temporal_analysis** table is removed, preparing for the creation of a new table for temporal analysis.

Step 23: Create Table temporal_analysis

Purpose: To create a table **temporal_analysis** that provides insights into the temporal distribution of crimes by year and month.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

```
CREATE TABLE temporal_analysis
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'Crimes/temporal_analysis'
AS
SELECT
   YEAR(FROM_UNIXTIME(UNIX_TIMESTAMP(crime.date_time, 'MM/dd/yyyy hh:mm:ss a'))) AS crime_year,
   MONTH(FROM_UNIXTIME(UNIX_TIMESTAMP(crime.date_time, 'MM/dd/yyyy hh:mm:ss a'))) AS crime_month,
   COUNT(*) AS crime_count
FROM
   crime
GROUP BY
   YEAR(FROM_UNIXTIME(UNIX_TIMESTAMP(crime.date_time, 'MM/dd/yyyy hh:mm:ss a'))),
   MONTH(FROM_UNIXTIME(UNIX_TIMESTAMP(crime.date_time, 'MM/dd/yyyy hh:mm:ss a')))
ORDER BY
   crime_year, crime_month;
```

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command creates a new table **temporal_analysis**.
    - The table will be stored as text files with fields delimited by commas.

- The table will be created at the specified location **'Crimes/temporal_analysis'**.
- The data for this table is derived from a select query on the **crime** table:
    - It extracts the year and month from the **date_time** column of the **crime** table using **FROM_UNIXTIME** and **UNIX_TIMESTAMP** functions.
    - It counts the number of crimes for each year-month combination.
    - The **GROUP BY** clause groups the data by year and month.
    - The **ORDER BY** clause orders the data by year and month.

3. **Note**:
    - Ensure that the **crime** table exists and is properly structured before running this command.
    - Adjust the file location based on your HDFS directory structure.

With this command, the **temporal_analysis** table will be created, providing insights into the temporal distribution of crimes by year and month.

Step 24: Retrieve Data from temporal_analysis Table

Purpose: To retrieve data from the **temporal_analysis** table to examine the temporal distribution of crimes by year and month.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

      SELECT * FROM temporal_analysis LIMIT 15;

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command selects all columns (*****) from the **temporal_analysis** table.
    - The **LIMIT 15** clause ensures that only the first 15 rows are returned.
    - You will see the top 15 rows of the **temporal_analysis** table displayed in the result, showing the crime year, crime month, and the corresponding crime count.
3. **Note**:
    - Ensure that the **temporal_analysis** table has been created and populated correctly before executing this command.

- Review the output to confirm that it provides meaningful insights into the temporal distribution of crimes.

Executing this command allows you to examine the data from the **temporal_analysis** table, helping you understand the distribution of crimes over different years and months.

Step 25: Check Contents of HDFS Directory and Retrieve Data

Purpose: To check the contents of the **temporal_analysis** directory in HDFS and retrieve the data.

1. **Check Contents of HDFS Directory**:
    - In the terminal, type the following command to check the contents of the directory where **temporal_analysis** data is stored in HDFS:

      hdfs dfs -ls Crimes/temporal_analysis

    - Press Enter to execute the command.
2. **Retrieve Data**:
    - After checking the contents, type the following command to retrieve the data file (**000000_0**) from HDFS:

      hdfs dfs -get /user/username/Crimes/temporal_analysis/000000_0 temporal_analysis.csv

    - Press Enter to execute the command.
3. **Interpretation**:
    - The first command lists the contents of the directory **Crimes/temporal_analysis** in HDFS, which should include data files.
    - The second command retrieves the data file (**000000_0**) from HDFS and saves it as **temporal_analysis.csv** in the current directory.
4. **Note**:
    - Ensure that the directory paths and file names are correct based on your HDFS structure.
    - The retrieved data file will be used for further analysis or visualization.

By executing these commands, you can verify the presence of data files in HDFS and retrieve them for local use.

Step 26: Transfer Data from Hadoop to Local Machine

Purpose: To copy the **temporal_analysis.csv** file from the Hadoop server to your local machine.

1. **Open Another Terminal:**
    - Open a new terminal on your local machine.
2. **Run the Command**:
    - In the new terminal, type the following command:

      scp username@IP_address_or_hostname:/home/username/temporal_analysis.csv .
    - Press Enter to execute the command.
3. **Interpretation**:
    - This command uses SCP (Secure Copy Protocol) to copy the **temporal_analysis.csv** file from your Hadoop server to your local machine.
    - Adjust the username, IP address, and file paths as needed based on your Hadoop server configuration and the file location.
4. **Note**:
    - The period (**.**) at the end of the command specifies the current directory as the destination for the copied file.
    - After executing this command, the **temporal_analysis.csv** file will be available in the current directory of your local machine.

Executing this command allows you to transfer the **temporal_analysis.csv** file from the Hadoop server to your local machine for further analysis or visualization.

Step 27: Preparing for Table Creation - Geospatial Analysis Table

Purpose: Before creating the **geospatial_analysis** table, ensure any existing table with the same name is dropped to avoid conflicts.

Context: This table extracts geospatial data, including latitude and longitude, from the dataset, excluding entries with missing coordinates.

1. **Run the Command**:
    - In the Beeline prompt, type the following command to drop the existing **geospatial_analysis** table:

      DROP TABLE IF EXISTS geospatial_analysis;
    - Press Enter to execute the command.
2. **Interpretation**:

- This command drops the **geospatial_analysis** table if it already exists.
- Dropping the table ensures that we start with a fresh table when creating the **geospatial_analysis** table.
- If the table doesn't exist, no action is taken.

3. **Note**:
    - Dropping a table deletes all data and metadata associated with it. Be sure to use this command judiciously.

With this step, we ensure that any existing **geospatial_analysis** table is removed, preparing for the creation of a new table for geospatial analysis.

Step 28: Create Table geospatial_analysis

Purpose: To create a table **geospatial_analysis** that extracts geospatial data, including latitude and longitude, from the dataset, excluding entries with missing coordinates.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

```
CREATE TABLE geospatial_analysis
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'Crimes/geospatial_analysis'
AS
SELECT
   crime.primary_type,
   crime.latitude,
   crime.longitude
FROM
   crime
WHERE
   crime.latitude IS NOT NULL
```

AND crime.longitude IS NOT NULL;

- Press Enter to execute the command.

2. **Interpretation**:
    - This command creates a new table **geospatial_analysis**.
    - The table will be stored as text files with fields delimited by commas.
    - The table will be created at the specified location **'Crimes/geospatial_analysis'**.
    - The data for this table is derived from a select query on the **crime** table:
        - It selects the crime type (**primary_type**), latitude, and longitude columns from the **crime** table.
        - It includes only records where both latitude and longitude are not null.
3. **Note**:
    - Ensure that the **crime** table exists and is properly structured before running this command.
    - Adjust the file location based on your HDFS directory structure.

With this command, the **geospatial_analysis** table will be created, focusing on geospatial data, ready for further analysis or visualization.

Step 29: Retrieve Data from geospatial_analysis Table

Purpose: To retrieve data from the **geospatial_analysis** table to examine geospatial information.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

      SELECT * FROM geospatial_analysis LIMIT 15;

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command selects all columns (*****) from the **geospatial_analysis** table.
    - The **LIMIT 15** clause ensures that only the first 15 rows are returned.
    - You will see the top 15 rows of the **geospatial_analysis** table displayed in the result, showing the crime type, latitude, and longitude.
3. **Note**:

- Ensure that the **geospatial_analysis** table has been created and populated correctly before executing this command.
- Review the output to confirm that it provides meaningful geospatial information.

Executing this command allows you to examine the data from the **geospatial_analysis** table, providing insights into the geographic distribution of crimes.

Step 30: Check Contents of HDFS Directory and Remove Data Files

Purpose: To check the contents of the **geospatial_analysis** directory in HDFS and remove data files.

1. **Check Contents of HDFS Directory**:
   - In the terminal, type the following command to check the contents of the directory where **geospatial_analysis** data is stored in HDFS:

     ```
     hdfs dfs -ls Crimes/geospatial_analysis
     ```

   - Press Enter to execute the command.
2. **Remove Data Files**:
   - After checking the contents, type the following command to remove the data files (**00000*_0**) from HDFS:

     ```
     hdfs dfs -rm 00000*_0
     ```

   - Press Enter to execute the command.
3. **Interpretation**:
   - The first command lists the contents of the directory **Crimes/geospatial_analysis** in HDFS, which should include data files.
   - The second command removes the data files (**00000*_0**) from HDFS.
4. **Note**:
   - Ensure that the directory paths and file names are correct based on your HDFS structure.
   - Be cautious when using the **rm** command with HDFS, as it permanently deletes data files.
   - Verify the files you are deleting to avoid accidental data loss.

Executing these commands helps in managing the data files in HDFS, ensuring efficient storage and maintenance.

Step 31: Retrieve Data from HDFS

Purpose: To retrieve the **geospatial_analysis.csv** file from HDFS.

1. **Run the Command**:
    - In the terminal, type the following command to retrieve the data file (**000000_0**) from HDFS:

      hdfs dfs -get /user/username/Crimes/geospatial_analysis/000000_0 geospatial_analysis.csv

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command retrieves the data file **000000_0** from the **geospatial_analysis** directory in HDFS.
    - The file will be saved as **geospatial_analysis.csv** in the current directory.
3. **Note**:
    - Ensure that the directory paths and file names are correct based on your HDFS structure.
    - After executing this command, the **geospatial_analysis.csv** file will be available in the current directory of your local machine.

Executing this command allows you to retrieve the **geospatial_analysis.csv** file from HDFS for further analysis or visualization.

Step 32: Transfer Data from Hadoop to Local Machine

Purpose: To copy the **geospatial_analysis.csv** file from the Hadoop server to your local machine.

1. **Open Another Terminal:**
    - Open a new terminal on your local machine.
2. **Run the Command**:
    - In the new terminal, type the following command:

      scp username@IP_address_or_hostname:/home/username/geospatial_analysis.csv .

    - Press Enter to execute the command.
3. **Interpretation**:

- This command uses SCP (Secure Copy Protocol) to copy the **geospatial_analysis.csv** file from your Hadoop server to your local machine.
- Adjust the username, IP address, and file paths as needed based on your Hadoop server configuration and the file location.

4. **Note**:
    - The period (**.**) at the end of the command specifies the current directory as the destination for the copied file.
    - After executing this command, the **geospatial_analysis.csv** file will be available in the current directory of your local machine.

Executing this command allows you to transfer the **geospatial_analysis.csv** file from the Hadoop server to your local machine for further analysis or visualization.

Step 33: Preparing for Table Creation - crime_by_location_description Table

Purpose: Before creating the **crime_by_location_description** table, ensure any existing table with the same name is dropped to avoid conflicts.

Context: This table aims to analyze crime occurrences based on the top 15 location descriptions, providing insights into the types of locations where crimes are most prevalent.

1. **Run the Command**:
    - In the Beeline prompt, type the following command to drop the existing **crime_by_location_description** table:

```
DROP TABLE IF EXISTS crime_by_location_description;
```

1.
    - Press Enter to execute the command.
2. **Interpretation**:
    - This command drops the **crime_by_location_description** table if it already exists.
    - Dropping the table ensures that we start with a fresh table when creating the **crime_by_location_description** table.
    - If the table doesn't exist, no action is taken.
3. **Note**:
    - Dropping a table deletes all data and metadata associated with it. Be sure to use this command judiciously.

With this step, we ensure that any existing **crime_by_location_description** table is removed, preparing for the creation of a new table for location-based crime analysis.

Step 34: Create Table crime_by_location_description

Purpose: To create a table **crime_by_location_description** that analyzes crime occurrences based on the top 15 location descriptions.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

      ```
      CREATE TABLE crime_by_location_description
      ROW FORMAT DELIMITED
      FIELDS TERMINATED BY ','
      STORED AS TEXTFILE
      LOCATION 'Crimes/crime_by_location_description'
      AS
      SELECT
        location_description,
        COUNT(*) AS crime_count
      FROM
        crime
      GROUP BY
        location_description
      ORDER BY
        crime_count DESC
      LIMIT 15;
      ```

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command creates a new table **crime_by_location_description**.
    - The table will be stored as text files with fields delimited by commas.

- The table will be created at the specified location **'Crimes/crime_by_location_description'**.
- The data for this table is derived from a select query on the **crime** table:
    - It selects the location description (**location_description**) and counts the occurrences of each description.
    - It groups the data by location description.
    - It orders the data by the crime count in descending order.
    - It limits the results to the top 15 location descriptions.

3. **Note**:
    - Ensure that the **crime** table exists and is properly structured before running this command.
    - Adjust the file location based on your HDFS directory structure.

With this command, the **crime_by_location_description** table will be created, focusing on analyzing crime occurrences based on location descriptions.

Step 35: Retrieve Data from crime_by_location_description Table

Purpose: To retrieve data from the **crime_by_location_description** table to examine crime occurrences based on location descriptions.

1. **Run the Command**:
    - In the Beeline prompt, type the following command:

      SELECT * FROM crime_by_location_description LIMIT 15;

    - Press Enter to execute the command.
2. **Interpretation**:
    - This command selects all columns (*****) from the **crime_by_location_description** table.
    - The **LIMIT 15** clause ensures that only the first 15 rows are returned.
    - You will see the top 15 rows of the **crime_by_location_description** table displayed in the result, showing the location description and the corresponding crime count.
3. **Note**:

- Ensure that the **crime_by_location_description** table has been created and populated correctly before executing this command.
- Review the output to confirm that it provides meaningful insights into crime occurrences based on location descriptions.

Executing this command allows you to examine the data from the **crime_by_location_description** table, providing insights into crime occurrences based on different location descriptions.

Step 36: Check Contents of HDFS Directory and Remove Data Files

Purpose: To check the contents of the **crime_by_location_description** directory in HDFS and remove data files.

1. **Check Contents of HDFS Directory**:
   - In the terminal, type the following command to check the contents of the directory where **crime_by_location_description** data is stored in HDFS:

     hdfs dfs -ls Crimes/crime_by_location_description

   - Press Enter to execute the command.
2. **Remove Data Files**:
   - After checking the contents, type the following command to remove the data files (**00000*_0**) from HDFS:

     hdfs dfs -rm 00000*_0

   - Press Enter to execute the command.
3. **Interpretation**:
   - The first command lists the contents of the directory **Crimes/crime_by_location_description** in HDFS, which should include data files.
   - The second command removes the data files (**00000*_0**) from HDFS.
4. **Note**:
   - Ensure that the directory paths and file names are correct based on your HDFS structure.
   - Be cautious when using the **rm** command with HDFS, as it permanently deletes data files.
   - Verify the files you are deleting to avoid accidental data loss.

Executing these commands helps in managing the data files in HDFS, ensuring efficient storage and maintenance.

Step 37: Retrieve Data from HDFS

Purpose: To retrieve the data file (**000000_0**) from HDFS.

1. **Run the Command**:
    - In the terminal, type the following command to retrieve the data file **000000_0** from HDFS:

      hdfs dfs -get /user/username/Crimes/crime_by_location_description/000000_0
    - Press Enter to execute the command.
2. **Interpretation**:
    - This command retrieves the data file **000000_0** from the **crime_by_location_description** directory in HDFS.
    - The file will be saved in the current directory with the same name.
3. **Note**:
    - Ensure that the directory paths and file names are correct based on your HDFS structure.
    - After executing this command, the **000000_0** file will be available in the current directory of your local machine.

Executing this command allows you to retrieve the data file from HDFS for further analysis or visualization.

Step 38: Transfer Data from Hadoop to Local Machine

Purpose: To copy the **000000_0** file from the Hadoop server to your local machine and rename it as **crime_by_location_description.txt**.

1. **Open Another Terminal:**
    - Open a new terminal on your local machine.
2. **Run the Command**:
    - In the new terminal, type the following command:

      scp username@IP_address_or_hostname:/home/username/000000_0 crime_by_location_description.txt
    - Press Enter to execute the command.

3. **Interpretation**:
    - This command uses SCP (Secure Copy Protocol) to copy the **000000_0** file from your Hadoop server to your local machine.
    - It renames the copied file as **crime_by_location_description.txt**.
4. **Note**:
    - Adjust the username, IP address, and file paths as needed based on your Hadoop server configuration and the file location.
    - After executing this command, the **crime_by_location_description.txt** file will be available in the current directory of your local machine.

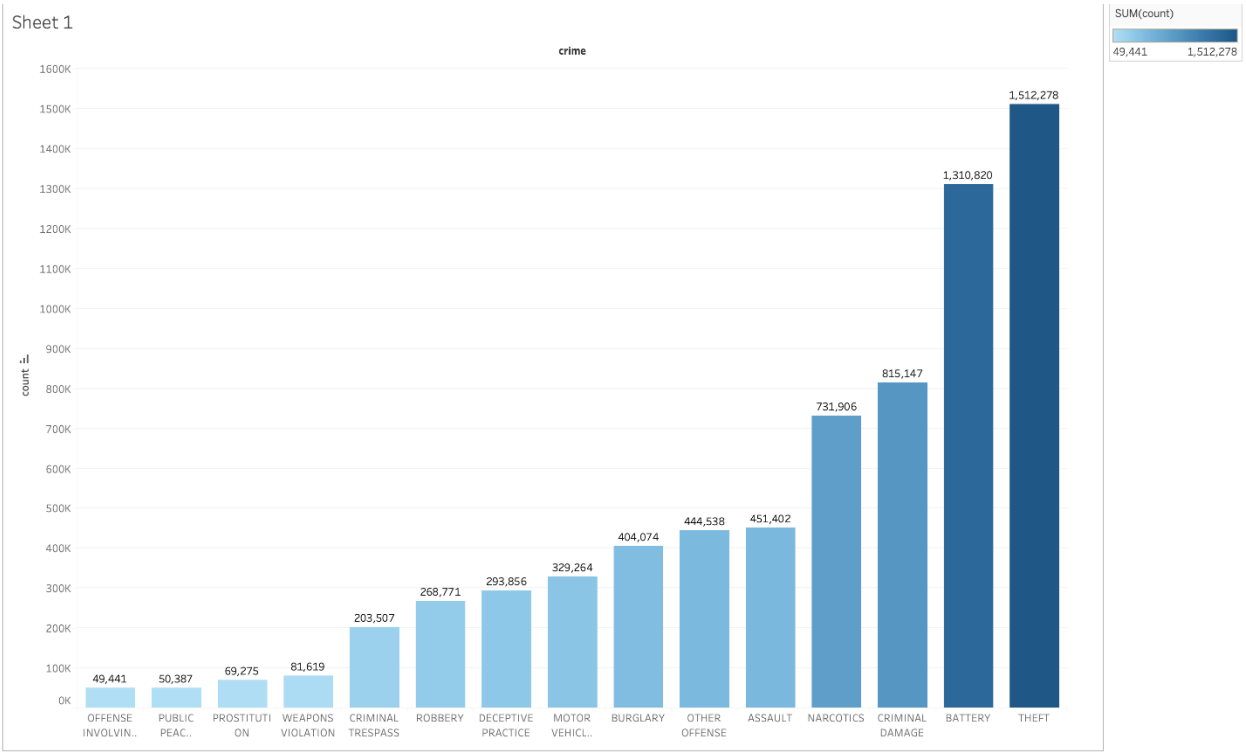
Executing this command allows you to transfer the data file from the Hadoop server to your local machine with the desired name for further analysis or processing.

Step 39: Top Crime Types Tableau Visualization

Purpose: To create a visualization of the top crime types using Tableau.

1. **Open Tableau**:
    - Launch Tableau on your computer.
2. **Link CSV File**:
    - Click on 'Text File' to link with all your CSV and text files saved in one folder.
    - Navigate to the folder where you saved the **top_crime_types.csv** file.
    - Click on **top_crime_types.csv** to open it.
3. **Rename Fields**:
    - In the Data Source tab, rename **F1** to **crime**, and **F2** to **count**.
4. **Create Visualization**:
    - Drag the **crime** field to the Columns shelf and the **count** field to the Rows shelf.
    - Click on 'Analysis' in the top toolbar and select 'Sort' > 'Ascending' to sort the crime types by count in ascending order.
5. **Add Color Gradient**:
    - Drag and drop the **crime** field to the Color option in the Marks card to show a gradient based on crime type.
6. **Show Mark Labels**:

- In the Marks card, click on 'Label', then click on 'Show Mark Labels' to display the count of each crime type.
- You should see something like this:



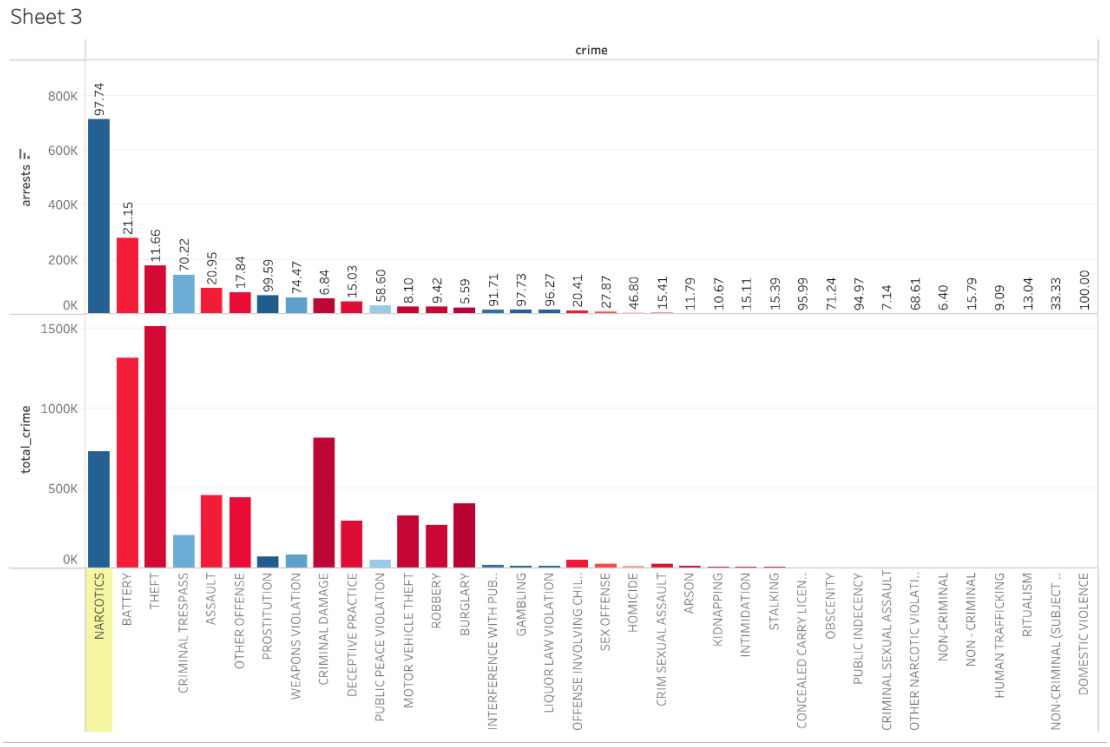
7. **Save and Export**:
   - Click on 'File' > 'Save As' to save the visualization.
   - Choose the desired format (e.g., .twb for Tableau Workbook).
   - Provide a meaningful name for the file (e.g., top_crime_types_visualization).
8. **Publish (Optional)**:
   - If you want to share the visualization, click on 'Server' > 'Publish Workbook' to publish it to Tableau Server or Tableau Public.
9. **Review and Analyze**:
   - Review the visualization to gain insights into the top crime types.
   - Analyze the distribution and frequency of different crime types.

By following these steps, you can create a Tableau visualization to effectively visualize and analyze the top crime types based on their frequency. This visualization provides valuable insights for understanding the most prevalent crime types in the dataset.

Step 40: Arrest Analysis Tableau Visualization

Purpose: To create a visualization analyzing arrests in relation to total crimes using Tableau.

1. **Rename Fields**:
    - In the Data Source tab, rename **F1** to **crime**, **F2** to **arrests**, **F3** to **total_crime**, and **F4** to **arrest_percentage**.
2. **Create Visualization**:
    - Drag the **crime** field to the Columns shelf, the **total_crime** field to the Rows shelf, and the **arrests** field to the Rows shelf.
    - Click on 'Analysis' in the top toolbar and select 'Sort' > 'Descending' to sort the crimes by total crime count in descending order.
3. **Add Color Gradient**:
    - Drag the **arrest_percentage** field to the Color option in the Marks card to color the bars based on the arrest percentage.
    - Drag the **arrest_percentage** field to the 'ALL Color' shelf.
    - Change the color palette to 'Red-Blue Diverging' to visualize the arrest percentages effectively.
    - You should see something like this:



4. **Save and Export**:
    - Click on 'File' > 'Save As' to save the visualization.
    - Choose the desired format (e.g., .twb for Tableau Workbook).
    - Provide a meaningful name for the file (e.g., arrest_analysis_visualization).
5. **Publish (Optional)**:
    - If you want to share the visualization, click on 'Server' > 'Publish Workbook' to publish it to Tableau Server or Tableau Public.
6. **Review and Analyze**:
    - Review the visualization to understand the relationship between arrests and total crimes for each crime type.
    - Analyze the arrest percentages and their distribution across different crime types.

By following these steps, you can create a Tableau visualization to analyze arrests in relation to total crimes, providing insights into law enforcement effectiveness for different crime types.

Step 41: Temporal Analysis Tableau Visualization

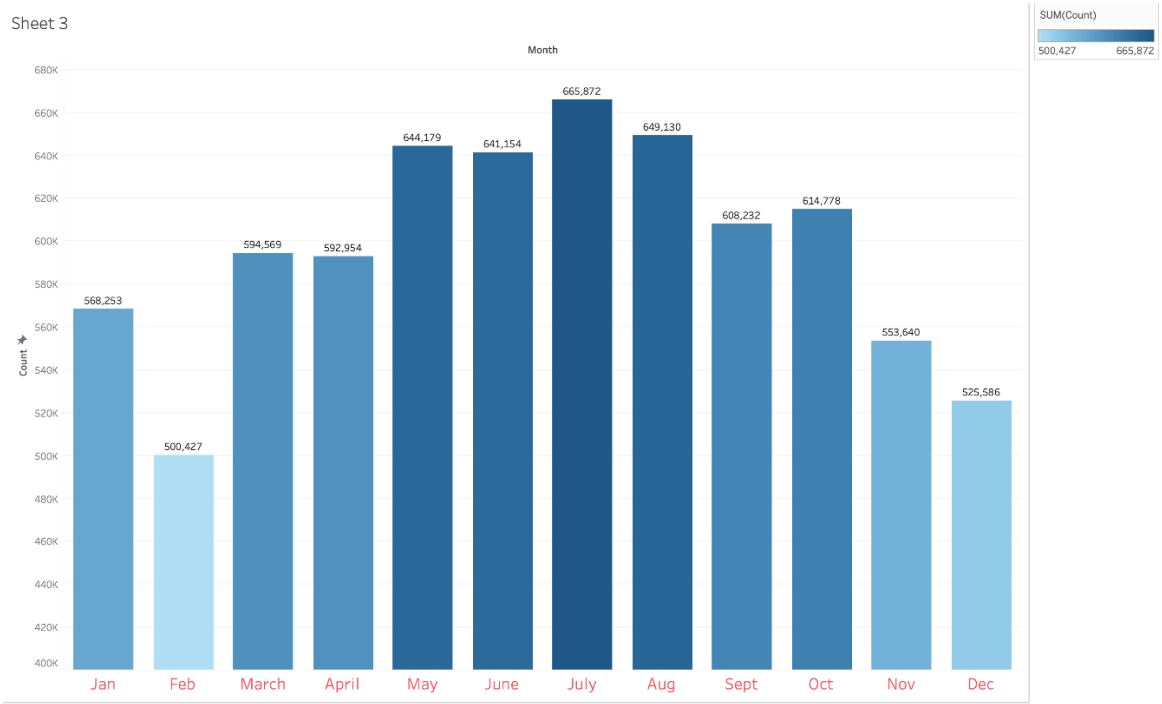
Purpose: To create a visualization analyzing the temporal distribution of crimes by year and month using Tableau.

1. **Data Preparation in Excel**:
    - Open the dataset in Excel.
    - Rename **\N** to **Year**, **\N 1** to **Month**, and **6855508** as **Count**.
    - Save the modified dataset.
2. **Create Calculated Field**:
    - In Tableau, create a calculated field with the following formula:

      DATE(STR([Year]) + "-" + STR([Month]) + "-01")
    - This will create a new calculated field representing the date.
3. **Create First Visualization**:
    - Drag the new calculated field to the Columns shelf and **Count** to the Rows shelf.
    - Click on the new calculated field in the Columns shelf, then click on 'Continuous' to ensure proper time series display.

- Click on the new calculated field again, go to 'More' > 'Month' to display the data at the month level.
- Right-click on the graph and select 'Trend Lines' > 'Show Trend Lines'.
- Right-click on the highest point in the graph, select 'Annotate' > 'Point', then click 'OK'.

4. **Create Second Visualization**:
    - Add a new sheet.
    - Drag **Month** to the Columns shelf and **Count** to the Rows shelf.
    - Right-click on the Count axis and select 'Edit Axis'.
    - Unselect 'Include Zero' and select 'Custom' for the range. Set the fixed start value to 400,000 and set the end to 'Automatic'.
    - Set the second tab to 'Uniform'.
    - Drag **Count** to the Marks card label to display the count.
    - Right-click on **Month** and select 'Discrete'.
    - Right-click on **Month** and select 'Edit Alias' to rename the months.
    - Drag **Count** to Color in the Marks card.
    - You should see something like this:



5. **Save and Export**:
    - Click on 'File' > 'Save As' to save the visualizations.
    - Choose the desired format (e.g., .twb for Tableau Workbook).
    - Provide meaningful names for the files.
6. **Publish (Optional)**:
    - If you want to share the visualizations, click on 'Server' > 'Publish Workbook' to publish them to Tableau Server or Tableau Public.
7. **Review and Analyze**:
    - Review the visualizations to understand the temporal distribution of crimes by year and month.
    - Analyze the trends and patterns over time to gain insights into crime fluctuations.

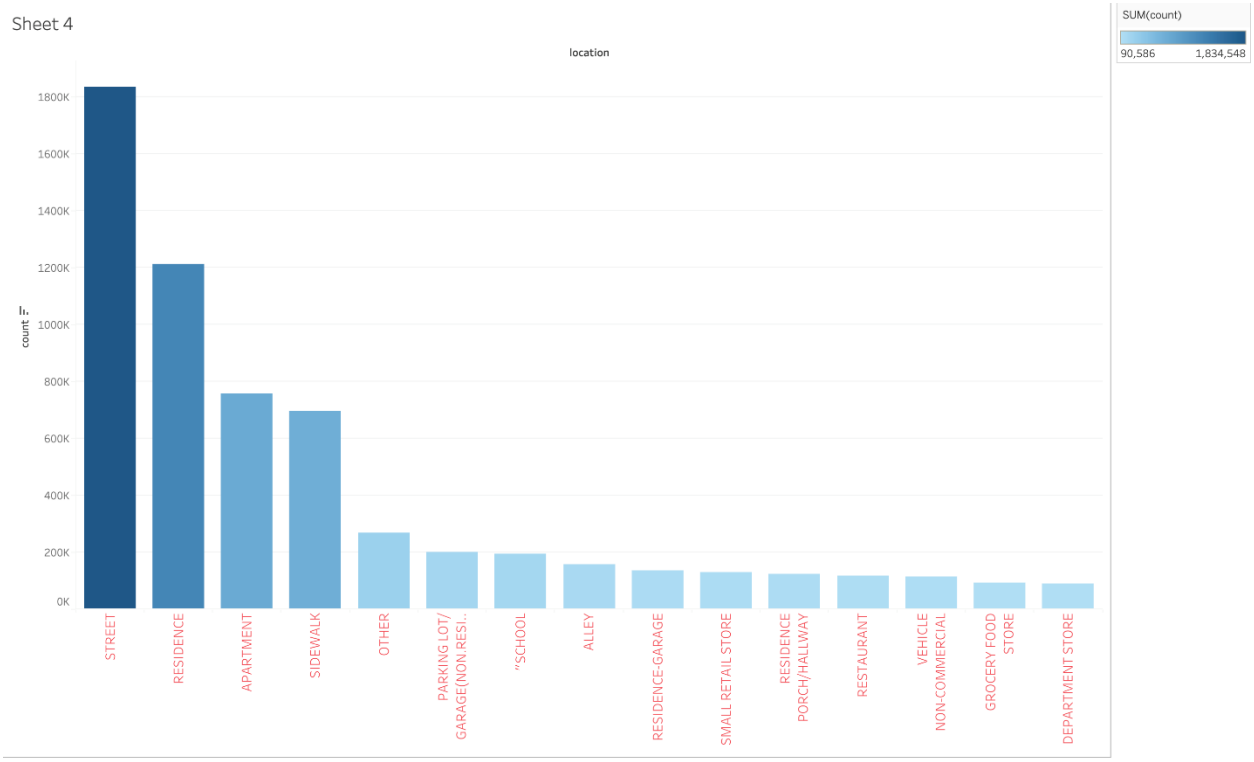
By following these steps, you can create Tableau visualizations to analyze the temporal distribution of crimes, providing valuable insights into crime trends over different time periods.

Step 42: Crime by Location Description Tableau Visualization

Purpose: To create a visualization analyzing crime occurrences based on location descriptions using Tableau.

1. **Rename Fields**:
    - In the Data Source tab, rename **F1** to **Location** and **F2** to **Count**.
2. **Create Visualization**:
    - Drag the **Location** field to the Columns shelf and the **Count** field to the Rows shelf.
    - Click on 'Analysis' in the top toolbar and select 'Sort' > 'Descending' to sort locations by crime count in descending order.
3. **Format Location**:
    - Right-click on the **Location** field in the Columns shelf and select 'Format'.
    - Adjust the formatting options as needed to ensure the location names are displayed appropriately.
4. **Add Color**:
    - Drag the **Count** field to the Color option in the Marks card to color the bars based on crime count.

- You should see something like this:



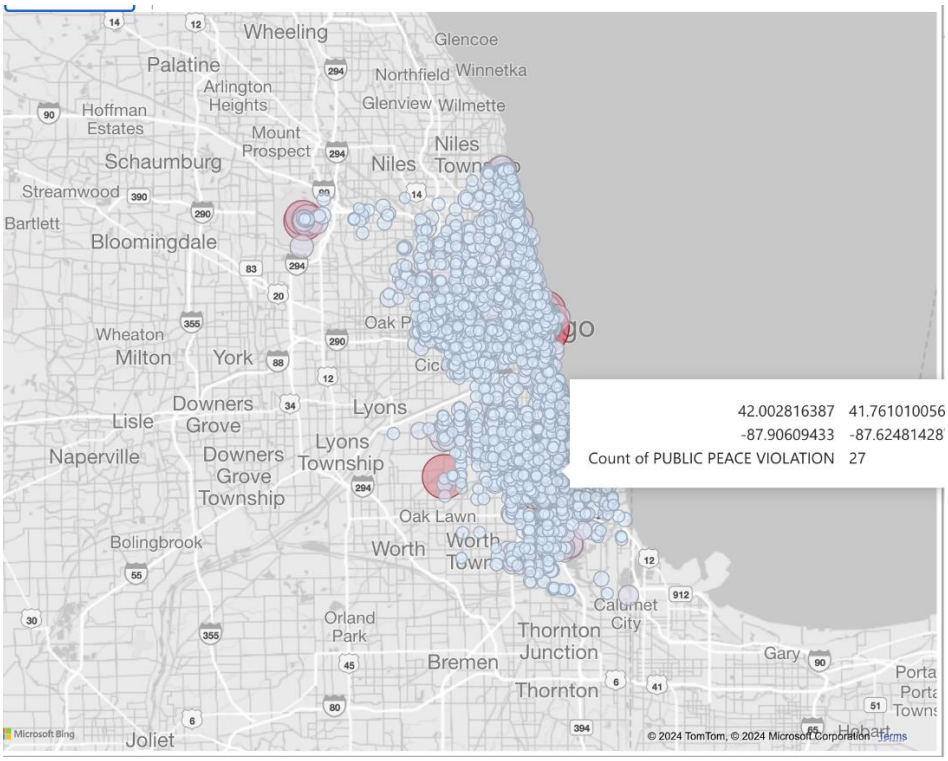
5. **Save and Export**:
    - Click on 'File' > 'Save As' to save the visualization.
    - Choose the desired format (e.g., .twb for Tableau Workbook).
    - Provide a meaningful name for the file.
6. **Publish (Optional)**:
    - If you want to share the visualization, click on 'Server' > 'Publish Workbook' to publish it to Tableau Server or Tableau Public.
7. **Review and Analyze**:
    - Review the visualization to understand crime occurrences based on location descriptions.
    - Analyze the distribution of crimes across different locations.

By following these steps, you can create a Tableau visualization to analyze crime occurrences based on location descriptions, providing insights into where crimes are most prevalent.

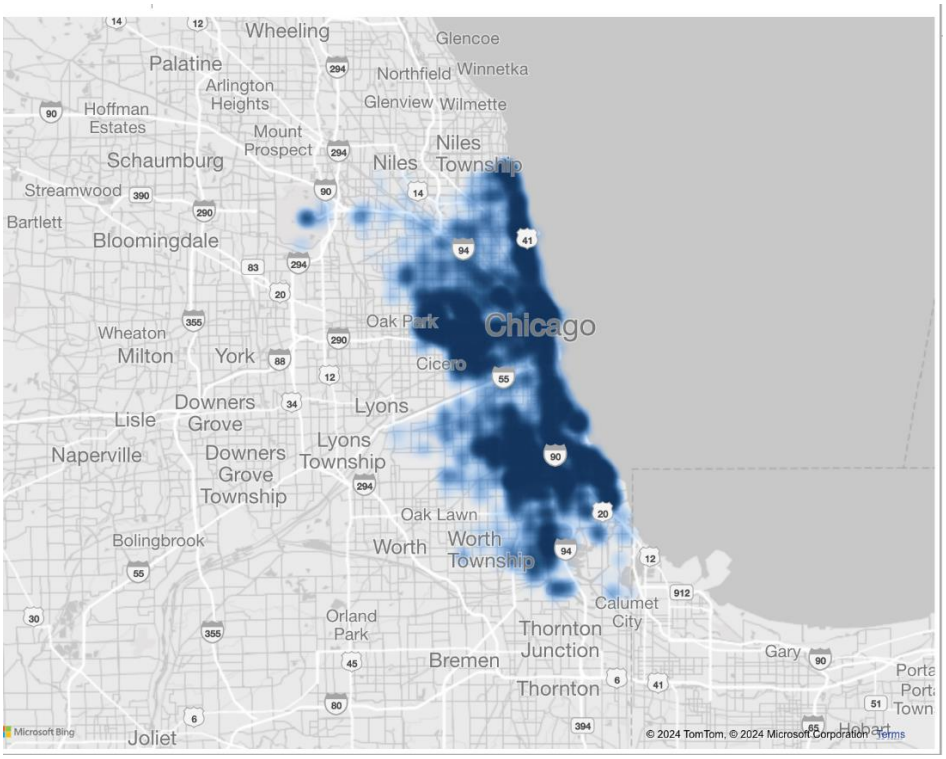
Step 43: Geospatial Analysis in Power BI

Purpose: To create a geospatial analysis of crime data using Power BI.

1. **Open Power BI and Log In**:
    - Open Power BI Desktop and log in to your account.
2. **Navigate to Workspace**:
    - Click on "Your Workspace" in the left pane to access your workspace.
3. **Create a New Report**:
    - Click on "+ New" and select "Semantic Model".
    - Click on "CSV" to import your CSV file containing the crime data.
4. **Add Data to Maps**:
    - Once the data is loaded, click on "Maps" from the Visualizations pane.
    - Drag the latitude field (usually labeled as **42.....**) to the Latitude field in the Fields pane.
    - Drag the longitude field (usually labeled as **-87.....**) to the Longitude field in the Fields pane.
    - Drag the field representing the level of public peace (e.g., **Public Peace**) to the Size field in the Visualizations pane to represent bubble size.
5. **Format Visualization**:
    - Go to the Format options for the visualizations.
    - Change the map style to "Grayscale" to improve visualization clarity.
6. **Conditional Formatting**:
    - Go to the Colors section, then click on "(fx) Conditional Formatting".
    - Choose the field representing the count of crimes (e.g., **Count of Crime**).
    - Set blue as the color for the lowest value and red for the highest value to visualize the concentration of crimes effectively.
    - You should see something like this:



7. **Exploring Heat Maps**:
    - Click on "Heat Maps" to see how crime is spread across Chicago.
    - You should see something like this:



This geospatial analysis in Power BI allows you to visualize crime data on a map, enabling you to understand the distribution and concentration of crimes across different areas in Chicago.

End of the Tutorial:

In this tutorial, we have covered the process of uploading a CSV file to Hadoop and performing various analyses on the Chicago Crime dataset from January 1, 2001, to July 22, 2020. We started by uploading the dataset, then proceeded to create several tables for different analyses, including temporal, geospatial, and location-based analyses.

We learned how to execute commands in Hadoop using Beeline and interact with HDFS to manage data files. By creating and querying tables, we gained insights into crime trends, top crime types, arrest analysis, crime occurrences based on location descriptions, and geospatial distribution.

Moving beyond Hadoop, we also explored other data analysis tools such as Tableau and Power BI to visualize and further analyze the data.

- **Temporal Analysis:** We analyzed crime trends over time, identifying patterns and fluctuations in crime occurrences by year and month.
- **Top Crime Types Tableau Visualization:** Using Tableau, we visualized the top crime types, providing insights into the most prevalent criminal behaviors within the dataset.
- **Arrest Analysis Tableau Visualization:** We analyzed arrests in relation to total crimes, offering insights into law enforcement effectiveness for different crime types.
- **Geospatial Analysis in Power BI:** We visualized crime occurrences on a map, allowing us to understand the distribution and concentration of crimes across different areas in Chicago.

These analyses can be valuable for law enforcement agencies, policymakers, and researchers to understand crime patterns and make informed decisions. By following this tutorial, you have gained foundational knowledge in Hadoop-based data analysis, as well as skills in using other powerful data analysis tools such as Tableau and Power BI. You can now apply these techniques to analyze other datasets of interest.