

GROUP 16

CRIME DATA ANALYSIS(2016-2019)

1 DATA CLEANING

1.1 CRIME DATA

The screenshot shows the RStudio interface. In the top-left pane, there is a script editor with the following R code:

```
removal.R* Untitled1* data_all*
1 data_all <- list.files(path = "C:/Users/Akhil/Downloads/crimeData",
2                         pattern = "*.csv", full.names = TRUE) %>%
3   lapply(read_csv) %>%
4   bind_rows
5
6 view(data_all)
7
8 setwd("C:/Users/Akhil/Downloads/merged_crimeData")
9
10 write.csv(data_all, "merged_crimeData.csv", row.names = FALSE)
```

In the bottom-left pane, the R console output is displayed:

```
R 4.2.3 . ~/DQ/DQtuts/
i Specify the column types or set show_col_types = FALSE to quiet this message.
Rows: 86075 Columns: 12
--- Column specification ---
delim: ","
chr (9): Crime ID, Month, Reported by, Falls within, Location, LSOA ...
dbl (2): Longitude, Latitude
lg1 (1): Context

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
> view(data_all)
```

The right side of the interface shows the Environment pane with a data frame named `data_all` containing 4198571 observations and 12 variables, and the Packages pane showing installed packages like `readr`, `readr`, `clipr`, `NCmisc`, `prettyunits`, `vroom`, and `foreign`.

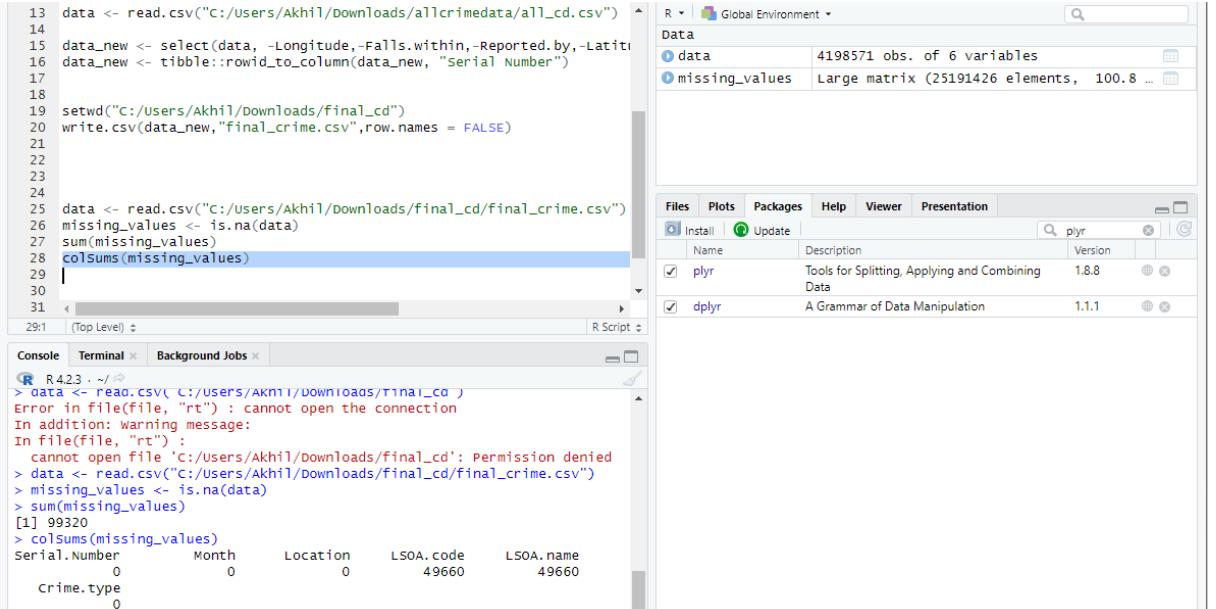
The screenshot shows the RStudio Console tab. The output displays a list of crime categories with their corresponding counts:

| Category | Count |
|---------------------------|--------|
| Anti-social behaviour | 923719 |
| Bicycle theft | 77825 |
| Burglary | 303722 |
| Criminal damage and arson | 234029 |
| Drugs | 144472 |
| Other crime | 41563 |
| Other theft | 454127 |
| Possession of weapons | 24258 |
| Public order | 192743 |
| Robbery | 125748 |
| Shoplifting | 181590 |
| Theft from the person | 179744 |
| Vehicle crime | 424777 |

CRIME TYPES IN RAW DATA

```
Console Terminal × Background Jobs ×
R 4.2.1 · ~/DQ/DQtuts/ ↗
> d %>%
+ group_by(Crime.type) %>%
+ summarize(count = n()) %>%
+ arrange(Crime.type)
# A tibble: 6 × 2
  Crime.type          count
  <chr>              <int>
1 Anti-social behaviour    947977
2 Drugs                  144472
3 Other crime            700369
4 Public order            192743
5 Theft                  1322756
6 Violence and sexual offences 879754
>
```

crime categorisation



The screenshot shows the RStudio interface with the following components:

- Script Editor:** Displays R code for reading and manipulating crime data.
- Console:** Shows the execution of the R code. It includes an error message about permission denied when trying to open a file.
- Global Environment:** Shows the current environment with two objects:
 - data:** A tibble with 4198571 observations and 6 variables.
 - missing_values:** A large matrix with 25191426 elements.
- Packages:** Shows the installed packages: **plyr** (version 1.8.8) and **dplyr** (version 1.1.1).

File Edit Code View Plots Session Build Debug Profile Tools Help

removal.R* Untitled1* data_all*

```

1
2
3
4
5
6 data <- read.csv("C:/Users/Akhil/Downloads/merged_crimedata/merged_crime.csv")
7 data_new <- select(data, -Longitude, -Falls.within, -Reported.by, -Latitude)
8 data_new <- tibble::rowid_to_column(data_new, "Serial Number")
9
10
11
12
13
14
15
16
17
18
19
20

```

Console Terminal Background Jobs

```

R 4.2.3 - C:/Users/Akhil/Downloads/merged_crimedata/
> data <- read.csv("C:/Users/Akhil/Downloads/merged_crimedata/merged_crime.csv")
> data_new <- select(data, -Longitude,-Falls.within,-Reported.by,-Latitude,-Crime.ID,-Last.outcome.category,-Context)
>

```

Environment History Connections Tutorial

Global Environment

Data

- data 4198571 obs. of 12 variables
- data_all 4198571 obs. of 12 variables
- data_new 4198571 obs. of 5 variables

Files Plots Packages Help Viewer Presentation

Install Update

| Name | Description | Version |
|---------|--|---------|
| bit | Classes and Methods for Fast Memory-Efficient Boolean Selections | 4.0.5 |
| bit64 | A S3 Class for Vectors of 64bit Integers | 4.0.5 |
| cli | Helpers for Developing Command Line Interfaces | 3.6.1 |
| clipr | Read and Write from the System Clipboard | 0.8.0 |
| cpp11 | A C++11 Interface for R's C Interface | 0.4.3 |
| crayon | Colored Terminal Output | 1.5.2 |
| dplyr | A Grammar of Data Manipulation | 1.1.1 |
| english | Translate Integers into English | 1.2-6 |
| fanari | ANSI Control Sequence Adapter String Functions | 1.0.1 |

KStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

new.R* data_new* data* crimeall.R*

```

57 View(data)
58
59 data_new <- select(data, -Longitude,-Location,-Falls.within,-Reported.
60
61 View(data_new)
62 names(data_new)[4] <- "Location"
63 View(data_new)
64 missing_values <- is.na(data_new)
65
66 sum(missing_values)
67 data_new <- data.frame(lapply(data_new, function(x) ifelse(is.na(x),
68 View(data_new)
69 setwd("C:/Users/Akhil/downloads/sruthi"))
70 write.csv(data_new, "crime_1.csv", row.names = FALSE)
71
72
73
74
75

```

Console Terminal Background Jobs

```

R 4.2.3 - ~/
cat.path, get.ext, rmv.ext

> data_new <- select(data, -Longitude,-Location,-Falls.within,-Reported.by,
-Latitude,-Crime.ID,-Context)
> names(data_new)[4] <- "Location"
> View(data_new)
> missing_values <- is.na(data_new)
>
> sum(missing_values)
[1] 1023788
> data_new <- data.frame(lapply(data_new, function(x) ifelse(is.na(x), "NO
DATA", x)))
> View(data_new)

```

Environment History Connections Tutorial

To Console To Source

data\$Month[data\$Month == "2018-03"
data\$Month[data\$Month == "2018-04"
data\$Month[data\$Month == "2018-05"
data\$Month[data\$Month == "2018-06"
data\$Month[data\$Month == "2018-07"
data\$Month[data\$Month == "2018-08"
data\$Month[data\$Month == "2018-09"
data\$Month[data\$Month == "2018-10"
data\$Month[data\$Month == "2018-11"
data\$Month[data\$Month == "2018-12"
data\$Month[data\$Month == "2019-01"
data\$Month[data\$Month == "2019-02"
data\$Month[data\$Month == "2019-03"
data\$Month[data\$Month == "2019-04"
data\$Month[data\$Month == "2019-05"
data\$Month[data\$Month == "2019-06"
data\$Month[data\$Month == "2019-07"
data\$Month[data\$Month == "2019-08"
data\$Month[data\$Month == "2019-09"
data\$Month[data\$Month == "2019-10"
data\$Month[data\$Month == "2019-11"
data\$Month[data\$Month == "2019-12"]

Files Plots Packages Help Viewer P

Install Update

| Name | Description |
|-------------|--|
| reader | Suite of Functions to FIles |
| readr | Read Rectangular Text |
| clipr | Read and Write from th |
| prettyunits | Pretty, Human Readabl Quantities |
| vroom | Read and Write Rectan |
| foreign | Read Data Stored by 'N Stata', 'Systat', 'Weka', |

```

41 data$Month[data$Month == "2018-11"] <- "2018-Nov"
42 data$Month[data$Month == "2018-12"] <- "2018-Dec"
43
44 data$Month[data$Month == "2019-01"] <- "2019-Jan"
45 data$Month[data$Month == "2019-02"] <- "2019-Feb"
46 data$Month[data$Month == "2019-03"] <- "2019-Mar"
47 data$Month[data$Month == "2019-04"] <- "2019-Apr"
48 data$Month[data$Month == "2019-05"] <- "2019-May"
49 data$Month[data$Month == "2019-06"] <- "2019-Jun"
50 data$Month[data$Month == "2019-07"] <- "2019-Jul"
51 data$Month[data$Month == "2019-08"] <- "2019-Aug"
52 data$Month[data$Month == "2019-09"] <- "2019-Sep"
53 data$Month[data$Month == "2019-10"] <- "2019-Oct"
54 data$Month[data$Month == "2019-11"] <- "2019-Nov"
55 data$Month[data$Month == "2019-12"] <- "2019-Dec"
56
57 View(data)
58

```

| Outcome | count |
|---|---------|
| | <int> |
| 1 Awaiting court outcome | 44326 |
| 2 Court case unable to proceed | 8696 |
| 3 Court result unavailable | 57275 |
| 4 Defendant found not guilty | 22044 |
| 5 Defendant sent to Crown Court | 272 |
| 6 Formal action is not in the public interest | 29 |
| 7 Investigation complete; no suspect identified | 1066426 |
| 8 Local resolution | 55391 |
| 9 NO DATA | 923719 |
| 10 offender deprived of property | 1283 |
| 11 offender fined | 15568 |
| 12 offender given a caution | 48694 |
| 13 offender given a drugs possession warning | 32234 |
| 14 offender given absolute discharge | 206 |
| 15 offender given community sentence | 19333 |
| 16 offender given conditional discharge | 7810 |
| 17 offender given penalty notice | 18721 |
| 18 offender given suspended prison sentence | 10520 |
| 19 offender ordered to pay compensation | 1482 |
| 20 offender otherwise dealt with | 2129 |
| 21 offender sent to prison | 23459 |
| 22 Status update unavailable | 1265058 |
| 23 Suspect charged as part of another case | 142 |
| 24 Unable to prosecute suspect | 3 |
| 25 Under investigation | 563251 |

STAR SCHEMA

Business Question 1

List & clarify relevant business question(s) that need to be answered

What are the trends and patterns in crime, population and income across different regions of London from 2016 to 2019

Identify and name possible Dimension tables

DimTime

DimLocation

Identify lowest level of granularity (for each Dimension)

Time = Year

Location - Region

Identify the Metrics of the problem.

Quantity of crime recorded yearly.

Increase or decrease of crime numbers over the year.

Define suitable attributes of the Dimension tables

DimTime = Year, month and quarter

DimLocation = MSOA code, Location name

| |
|---|
| |
| Define the Fact table (keys and metrics) |
| Crime Population Key: MSOA code Metrics: Quantity, type and time |
| Draw the Star Schema |

Business Question 2

| |
|---|
| List & clarify relevant business question(s) that need to be answered |
| WHAT ARE THE TOP 3 TYPES OF CRIMES RECORDED IN THE 10 MOST POPULOUS LOCATIONS IN LONDON FROM 2016 TO 2019 |
| Identify and name possible Dimension tables |
| DIMLOCATION DIMTIME DIMPOPULATION |
| Identify lowest level of granularity (for each Dimension) |
| DIMLOCATION = Region DIMTIME = YEAR DIMPOPULATION = YEAR |
| Identify the Metrics of the problem |
| TYPES OF CRIME QUANTITY TIME |
| Define suitable attributes of the Dimension tables |

DIMLOCATION = MSOA CODE, LOCATION NAME

DIMTIME = YEAR

Define the Fact table (keys and metrics)

CRIME

Draw the Star Schema

BUSINESS QUESTION 3

List & clarify relevant business question(s) that need to be answered

WHAT IS THE HIGHEST CRIME RATE AND MOST OCCURRED TYPE OF CRIME RECORDED IN DIFFERENT LOCATIONS IN LONDON QUARTERLY FROM 2016 TO 2019

Identify and name possible Dimension tables

DIMLOCATION

DIMTIME

Identify lowest level of granularity (for each Dimension)

DIMLOC = REGION

DIMTIME = QUARTERLY

Identify the Metrics of the problem

QUANTITY/COUNTS

CRIME TYPE

TIME

Define suitable attributes of the Dimension tables

DIMLOC = MSOA CODE, LOCATION NAME

DIMTIME = YEAR, MONTH, QUARTER

Define the Fact table (keys and metrics)

CRIME

KEY: crime id, msoa codes, crime types

Draw the Star Schema

BUSINESS QUESTION 4

List & clarify relevant business question(s) that need to be answered

HOW MANY REPORTED CRIME CASES WERE COMPLETED BY INVESTIGATION IN ALL REGIONS OF LONDON IN 2016 TO 2019

Identify and name possible Dimension tables

DIM LOCATION

DIMTIME

DIMOUTCOME

Identify lowest level of granularity (for each Dimension)

DIMLOCATION = REGION

DIMTIME = YEAR

Identify the Metrics of the problem

QUANTITY

Define suitable attributes of the Dimension tables

DIMLOCATION = MSOA CODE, LOCATION NAME

DIMTIME = YEAR

Define the Fact table (keys and metrics)

CRIME
KEY: CRIME ID, MSOA CODE, CRIME OUTCOME

Draw the Star Schema

BUSINESS QUESTION 5

List & clarify relevant business question(s) that need to be answered

WHICH SUB-REGIONS WITHIN THE TOP 3 REGIONS WITH THE HIGHEST CRIME RATE IN LONDIN ARE MOST IN NEED OF INCREASED SECURITY MEASURES IN 2019

Identify and name possible Dimension tables

DIMLOCATION
DIMTIME

Identify lowest level of granularity (for each Dimension)

DIMLOCATION = SUBREGION
DIMTIME = YEAR

Identify the Metrics of the problem

LOCATION, QUANTITY

Define suitable attributes of the Dimension tables

LOCATION = LSOA CODE, MSOA CODE, LOCATION NAME
TIME = YEARS

Define the Fact table (keys and metrics)

CRIME

KEY: CRIME ID, LSOA CODE,

| |
|-------------------|
| METRICS: QUANTITY |
| |

TRANSFORMATION IN HIVE (DATA WAREHOUSING)

Table creation for uploading csv to hive

```
install.packages  
install.packages(ODBC)  
install.packages(RCPP)  
library(DBI)  
library(ODBC)  
library(rcpp)  
  
con <- DBI::dbconnect (odbc::()),  
  Driver = "Cloudera ODBC Driver for Apache Hive",  
  Host = "sandbox-hdp.hortonworks.com",  
  UID = "hive"  
  PWD =""  
  port = 10000)  
dbwriteTable(con,"Crime",crime)  
dbwriteTable(con,"Population",population)  
dbwriteTable(con,"Income",income)
```

The screenshot shows the Data Studio Analytics Compose interface. On the left sidebar, under 'Compose', there is a 'Queries' section with a 'Compose' button, a 'Database' section with a 'default' database selected, and a 'Reports' section.

The main area is titled 'Compose' and shows a query for a 'Crime' table. The code is as follows:

```
1 CREATE TABLE Crime (
2   Crime_ID BIGINT,
3   Crime_date STRING,
4   LSOA_Code STRING,
5   MSOA_Code STRING,
6   Location STRING,
7   Crime_Type STRING,
8   Outcome STRING
9 )
10 ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
11 STORED AS TEXTFILE;
12
13 LOAD DATA INPATH'/user/maria_dev/ADNP/crime_latest1.csv' OVERWRITE INTO TABLE Crime;
14
15 SELECT * from Crime LIMIT
16
17 DROP TABLE Crime
```

Below the code, there are buttons for 'EXECUTE', 'SAVE AS', 'VISUAL EXPLAIN', and checkboxes for 'Show Results' and 'Download Results'. The 'RESULTS' tab is selected, showing the following data:

| CRIME.CRIME_ID | CRIME.CRIME_DATE | CRIME.LSOA_CODE | CRIME.MSOA_CODE | CRIME.LOCATION | CRIME.CRIME_TYPE | CRIME.OUTCOME |
|----------------|------------------|-----------------|-----------------|---------------------------|------------------|---------------------|
| 1 | 2016-Jan | E01000027 | E09000002 | Barking and Dagenham 001A | Theft | Investigation Co ed |
| 2 | 2016-Jan | E01000027 | E09000002 | Barking and Dagenham 001A | Drugs | Investigation Co ed |
| 3 | 2016-Jan | E01000027 | E09000002 | Barking and Dagenham 001A | Drugs | Investigation Co ed |

There is also an 'EXPORT DATA' button at the bottom right.

The screenshot shows the Data Studio Analytics Compose interface. The left sidebar is identical to the first screenshot, with 'Compose' selected under 'Queries'.

The main area is titled 'Compose' and shows a query for an 'Income' table. The code is as follows:

```
1 CREATE TABLE Income (
2   Income_ID BIGINT,
3   MSOA_Code STRING,
4   Location STRING,
5   Pay_per_week_2016 BIGINT,
6   Pay_per_week_2017 BIGINT,
7   Pay_per_week_2018 BIGINT,
8   Pay_per_week_2019 BIGINT
9 )
10 ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
11 STORED AS TEXTFILE;
12
13 LOAD DATA INPATH'/user/maria_dev/ADNP/final_income.csv' OVERWRITE INTO TABLE Income;
14
15 SELECT * from Income
16 SELECT count(*) from Income
```

Below the code, there are buttons for 'EXECUTE', 'SAVE AS', 'VISUAL EXPLAIN', and checkboxes for 'Show Results' and 'Download Results'. The 'RESULTS' tab is selected, showing the following data:

| _C0 |
|-----|
| 32 |

There is also an 'EXPORT DATA' button at the bottom right.

The screenshot shows a MySQL Workbench interface with multiple tabs at the top: 'Saved', 'Worksheetcrime1', 'WorksheetCRIMENEW2', 'Worksheetpopulation', 'Worksheet5', and a new tab icon. The main area contains the following SQL code:

```
1 CREATE TABLE Population (
2 Population_ID BIGINT,
3 MSOA_Code STRING,
4 Location STRING,
5 Population_2016 STRING,
6 Population_2017 STRING,
7 Population_2018 STRING,
8 Population_2019 STRING
9 )
10 ROW FORMAT DELIMITED FIELDS TERMINATED BY ","
11 STORED AS TEXTFILE;
12
13
14 LOAD DATA INPATH '/user/maria_dev/ADMP/final_population.csv' OVERWRITE INTO TABLE Population;
15
16 SELECT * from population
```

Below the code are buttons for 'EXECUTE', 'SAVE AS', 'VISUAL EXPLAIN', 'Show Results' (checked), and 'Download Results'. The 'RESULTS' tab is selected in the results pane, which displays the following data:

| POPULATION.POPULATION_ID | POPULATION.MSOA_CODE | POPULATION.LOCATION | POPULATION.POPULATION_2016 | POPULATION.POPULATION_2017 | POPULATION.POPULATION_2018 | POPUL |
|--------------------------|----------------------|---------------------|----------------------------|----------------------------|----------------------------|-------|
| 1 | 1 | E09000002 | Barking and Dagenham | 462.2 | 461 | 479.1 |
| 2 | 2 | E09000003 | Barnet | 485.6 | 522.6 | 536.6 |
| 3 | 3 | E09000004 | Bexley | 487 | 513 | 512.7 |
| 4 | 4 | E09000005 | Brent | 467 | 471 | 479.9 |

There are also 'LOG' and 'EXPORT DATA' buttons in the results pane.

Dimension and fact table creation in hive

Saved Worksheet1 * dimincome * Worksheet3 * +

```
1 DROP TABLE dimIncome
2
3 CREATE TABLE dimIncome (
4 Income_ID BIGINT,
5 Income BIGINT,
6 PRIMARY KEY (Income_ID) DISABLE NOVALIDATE );
7
8
9 INSERT INTO dimIncome (Income_ID,Income)
10 SELECT Row_number() over(),Pay_per_week
11 FROM Income group by pay_per_week;
12
13 SELECT COUNT(*) FROM DIMINCOME
```

EXECUTE SAVE AS ▾ VISUAL EXPLAIN Show Results Download Results

RESULTS LOG EXPORT DATA ⌂ ⌂ ⌂

_C0
98

Saved Worksheet1 * +

```
1 Insert into dimTime select Crime_date, split(Crime_date,'-')[0],split(Crime_date,'-')[1],
2 CONCAT(
3   'Q',
4   CAST(
5     (
6       MONTH(
7         FROM_UNIXTIME(UNIX_TIMESTAMP(Crime_date, 'yyyy-MMM'))
8       ) - 1
9     ) DIV 3 + 1 AS STRING
10   )
11 )
12 from Crime group by Crime_Date
13
14
15 select count (*) from dimtime
```

EXECUTE SAVE AS ▾ VISUAL EXPLAIN Show Results Download Results

RESULTS LOG

_C0
48

```
12 from Crime group by Crime_Date
```

EXECUTE

SAVE AS ▾

VISUAL EXPLAIN

Show Results Download

 RESULTS

 LOG

| CRIME_DATE | _C1 | _C2 | _C3 |
|------------|------|-----|-----|
| 2016-Apr | 2016 | Apr | Q2 |
| 2016-Dec | 2016 | Dec | Q4 |
| 2016-Jul | 2016 | Jul | Q3 |
| 2016-Sep | 2016 | Sep | Q3 |
| 2018-Apr | 2018 | Apr | Q2 |
| 2018-Feb | 2018 | Feb | Q1 |

```
1 SELECT dc.crime_id,d1.location_id,dt.time_id
2 FROM (( crime c
3 inner join dimCrime dc
4 on (dc.Crime_type = c.Crime_type
5 and dc.Outcome = c.Outcome))
6 inner join dimlocation d1
7 on (d1.sub_location= c.location))
8 inner join dimtime dt
9 On c.crime_date=dt.time_id)
```

EXECUTE **SAVE AS ▾** **VISUAL EXPLAIN** Show Results Do

RESULTS **LOG** **VISUAL EXPLAIN**

| DC.CRIME_ID | DLLLOCATION_ID | DT.TIME_ID |
|-------------|----------------|------------|
| 12 | 1 | 2016-Apr |
| 17 | 1 | 2019-Sep |

```
Saved dimincome dimcrime dimpopulation dimlocation dimtime fact

1 CREATE TABLE dimIncome (
2 Income_ID BIGINT,
3 Income BIGINT,
4 PRIMARY KEY (Income_ID) DISABLE NOVALIDATE );
5
6
7 INSERT INTO dimIncome SELECT DISTINCT Income from Income
8
9
```

EXECUTE **SAVE AS ▾** **VISUAL EXPLAIN** Show Results Download Results

RESULTS **LOG** **VISUAL EXPLAIN**

Query completed.

```
1 SELECT dc.crime_id,d1.location_id,dt.time_id
2 FROM (( crime c
3 inner join dimCrime dc
4 on (dc.Crime_type = c.Crime_type
5 and dc.Outcome = c.Outcome))
6 inner join dimlocation d1
7 on (d1.sub_location= c.location))
8 inner join dimtime dt
9 On c.crime_date=dt.time_id)
```

EXECUTE **SAVE AS ▾** **VISUAL EXPLAIN** Show Results Do

RESULTS **LOG** **VISUAL EXPLAIN**

| DC.CRIME_ID | DLLLOCATION_ID | DT.TIME_ID |
|-------------|----------------|------------|
| 12 | 1 | 2016-Apr |
| 17 | 1 | 2019-Sep |

Saved dimincome dimcrime dimpopulation dimlocation dimtime factcrime1

```
1 CREATE TABLE dimTime (
2 Time_ID BIGINT,
3 Year BIGINT,
4 Month STRING,
5 PRIMARY KEY (Time_ID) DISABLE NOVALIDATE );
6
```

EXECUTE **SAVE AS ▾** **VISUAL EXPLAIN** Show Results Download Results

RESULTS **LOG** **VISUAL EXPLAIN**

Query completed.

Saved dimincome dimcrime dimpopulation dimlocation dimtime factcrime1 factcrime2 factcrime3

```
1 CREATE TABLE factCrime3 (
2 Crime_ID BIGINT,
3 Location_ID BIGINT,
4 Time_ID BIGINT,
5 Crime_count BIGINT,
6
7 PRIMARY KEY(Crime_ID,Location_ID,Time_ID) DISABLE NOVALIDATE);
8
```

EXECUTE **SAVE AS ▾** **VISUAL EXPLAIN** Show Results Download Results

RESULTS **LOG** **VISUAL EXPLAIN**

EXPORT DATA **◀** **▶**

Query completed.

Saved Worksheet1 * 222 Worksheet3 * Worksheet4 * today3 +

```
1
2 INSERT INTO factCrime3 SELECT dc.crime_id,dl.location_id,dt.time_id,count(*) as crime_count
3 FROM ((crime c
4 inner join dimCrime dc
5 on (dc.Crime_type = c.Crime_type
6 and dc.Outcome = c.Outcome))
7 inner join dimlocation dl
8 on (dl.sub_location= c.location))
9 inner join dimtime dt
10 On c.crime_date=dt.time_id)
11 GROUP BY dc.crime_id, dl.location_id,dt.time_id
12
13 |select * from factcrime3
```

EXECUTE SAVE AS ▾ VISUAL EXPLAIN Show Results Download Results

RESULTS LOG % VISUAL EXPLAIN

| FACTCRIME3.CRIME_ID | FACTCRIME3.LOCATION_ID | FACTCRIME3.TIME_ID | FACTCRIME3.CRIME_COUNT |
|---------------------|------------------------|--------------------|------------------------|
| 2 | 1 | 2016-Apr | 24 |
| 8 | 1 | 2016-Apr | 12 |
| 11 | 1 | 2016-Apr | 24 |

Saved Worksheet1 * +

```
1 INSERT INTO dimpopulation (Population_ID,Population)
2 SELECT row_number() over(),population_count
3 FROM population group by population_count;
4
5
6 SELECT count(*) from dimpopulation|
```

EXECUTE **SAVE AS ▾** **VISUAL EXPLAIN** Show Results Download Resu

RESULTS **LOG**

_C0

123

```
Saved ▾ Worksheet1 * x tactcrime1 * x Worksheet3 * x +  
1 CREATE TABLE factCrime1 (  
2 Crime_ID BIGINT,  
3 Population_ID BIGINT,  
4 Income_ID BIGINT,  
5 Location_ID BIGINT,  
6 Time_ID BIGINT,  
7 Population_Count BIGINT,  
8 Income_Sum BIGINT,  
9 Crime_Count BIGINT,  
10  
11 PRIMARY KEY(Crime_ID,Population_ID,Income_ID,Location_ID,Time_ID) DISABLE NOVALIDATE);  
12  
13  
14  
15
```

EXECUTE **SAVE AS** **▼** **VISUAL EXPLAIN** Show Results Download Results

```
Saved ▾ dimincome * x dimcrime * x dimpopulation * x dimlocation * x dimtime * x factcrime1 * x factcrime2 * x  
1 CREATE TABLE factCrime2 (  
2 Crime_ID BIGINT,  
3 Population_ID BIGINT,  
4 Location_ID BIGINT,  
5 Time_ID BIGINT,  
6 Crime_Count BIGINT,  
7 Population_Count BIGINT,  
8  
9 PRIMARY KEY(Crime_ID,Population_ID,Location_ID,Time_ID) DISABLE NOVALIDATE);  
10
```

EXECUTE **SAVE AS** **▼** **VISUAL EXPLAIN** Show Results Download Results

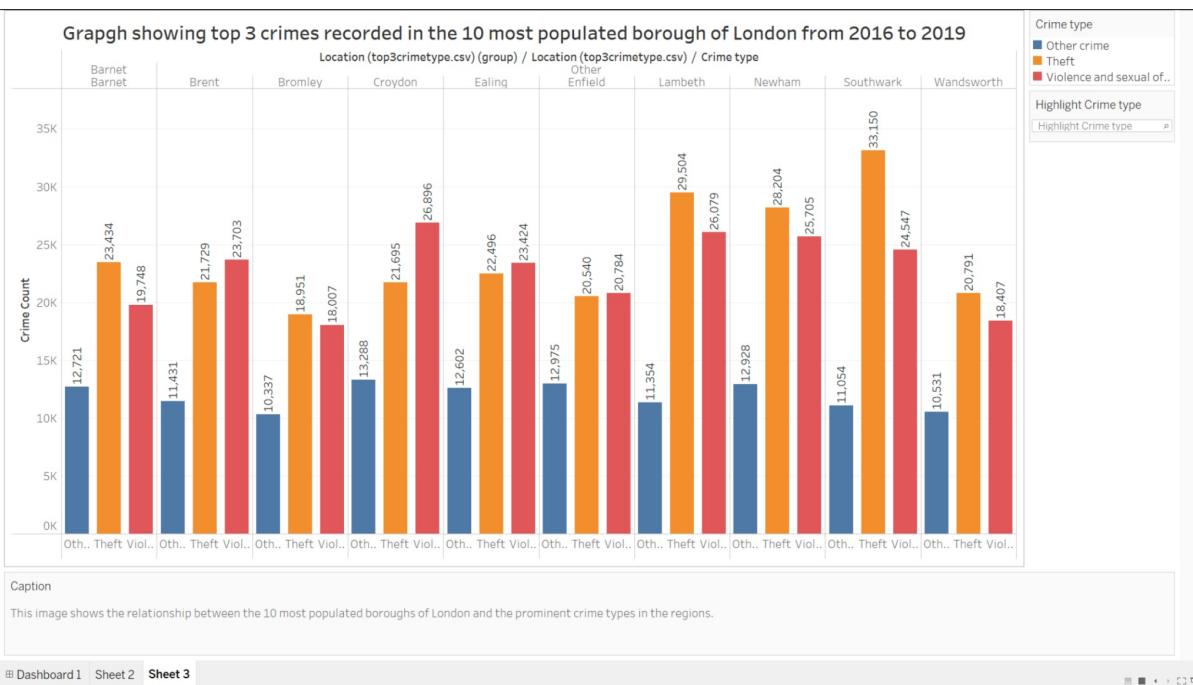
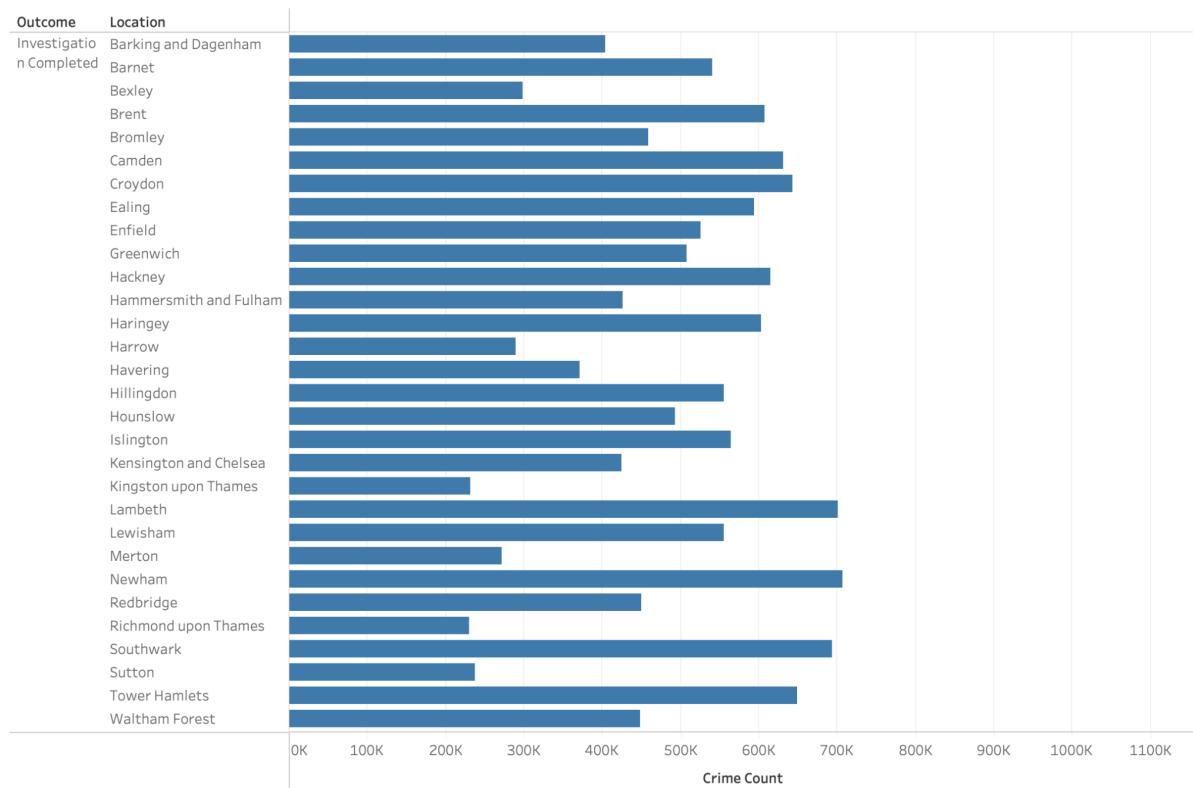
RESULTS **LOG** **▼ VISUAL EXPLAIN** **EXPORT D**

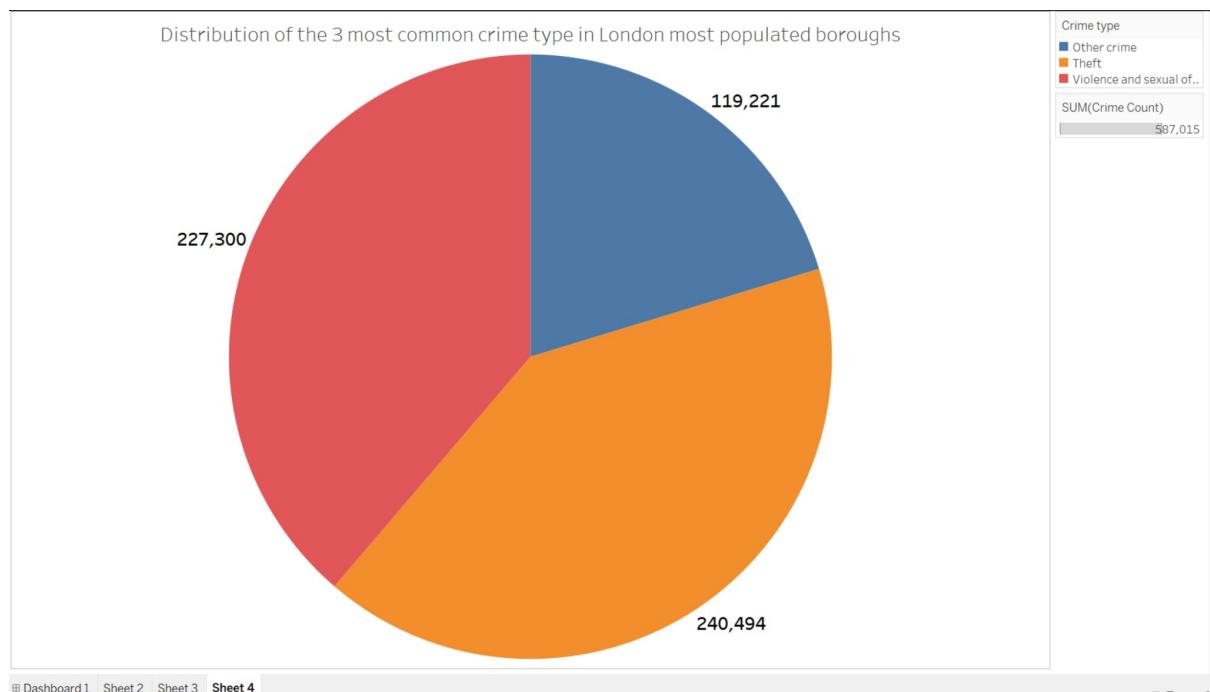
Query completed.

25

DATA VISUALISATION PART

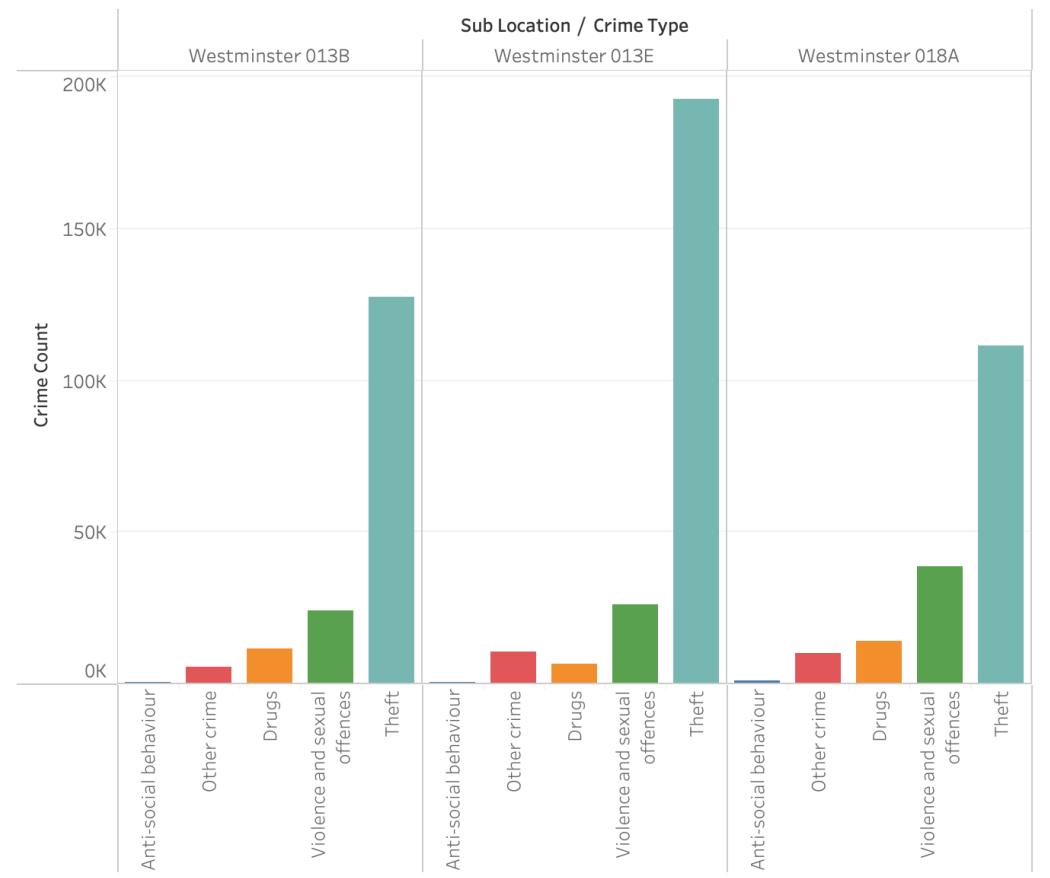
Completed investigations count by location





Dashboard 1 Sheet 2 Sheet 3 **Sheet 4**

Top 3 subregions with highest crimes by crime type



Sub Location
Westminster 013B
Westminster 013E
Westminster 018A

