Analysing Criminal Case Outcomes: A Data Science Perspective on Crown Prosecution Service Dataset Using R

Project Owner: Onur Tuncay

Table of Contents

List of Figures

List of Tables

1 Introduction

The increasing demand for data-driven decision-making within the criminal justice system highlights the significance of open government data. Due to the increasing data and digitalization in this field, the importance of data science techniques and data visualization techniques has increased (Castro-Toledo et al., 2023). For instance, Lavorgna and Ugwudike (2021) investigated the narratives and frames surrounding the adoption of data-driven technologies in criminal justice systems. In addition, their work revealed that optimistic, neutral, and oppositional frames often legitimize these tools as necessary for crime control, while ignoring their potential harms and inequalities in digital participation.

This study examines the Crown Prosecution Service (CPS) Case Outcomes by Principal Offence Category dataset. It provides insights into how criminal cases are processed and concluded in England and Wales. The dataset includes monthly breakdowns of case outcomes by main offence category. Moreover, this study uses data science methods and data visualisation techniques to identify patterns, trends and inequalities within the criminal justice system over a minimum of 24 months between 2014 and 2015. As a result, this study provides a valuable opportunity for data-driven interpretation and decision-making for the criminal justice system.

1.1 Motivation of the Study and Ethical Considerations

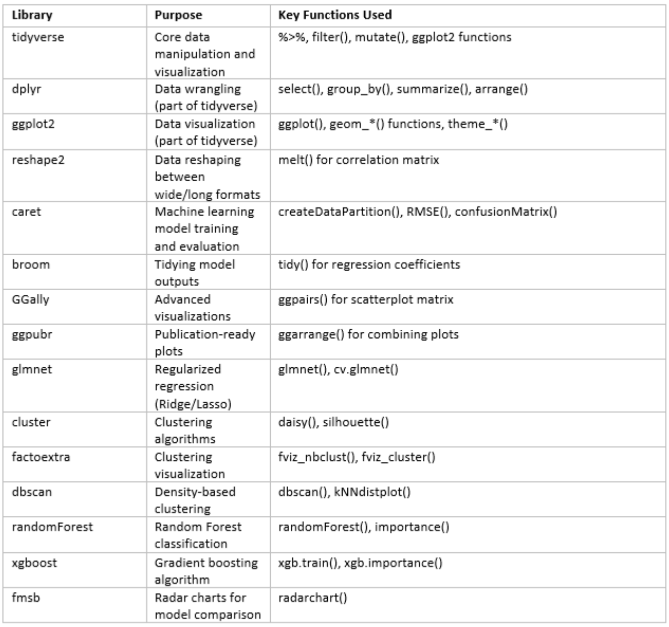
The main motivation for this analysis is to apply the advanced data analytics techniques learned throughout the CT7202 module to a real-world government dataset using the R language. With the increasing public interest in justice reform and transparency, it is increasingly important to understand how different crime categories are interpreted and how outcomes vary across time and geography. The scope of this study includes data integration and cleaning, descriptive analysis, predictive modeling, and critical evaluation of the tools and techniques used. In ethical perspective, the dataset used in this study is licensed under the Open Government License and is available for use in this study. Similarly, the techniques and tools used in this study are ethically acceptable as they are open source.

1.2 Experimental Setup

The developments were performed on an Acer Aspire A315-44P laptop featuring an AMD Ryzen 5 5500U processor (12 threads, ~2.1GHz), 16GB of RAM, and a Windows 11 Pro (64-bit) operating system. The hardware supported DirectX 12 and utilized a 38GB page file for memory management. This configuration provided a reliable platform for executing machine learning models and optimization algorithms, balancing computational efficiency with resource constraints for medium-scale experimental workloads.

1.3 R Libraries Used in this Study

In this study several R libraries were used to enhance analysis and modelling phases. Table 1.1 provides a systematic categorization of R libraries used in the criminal case outcome analysis, along with their primary functions and key methodological roles. The selected packages collectively address the full analytical pipeline.



Library	Purpose	Key Functions Used
tidyverse	Core data manipulation and visualization	%>%, filter(), mutate(), ggplot2 functions
dplyr	Data wrangling (part of tidyverse)	select(), group_by(), summarize(), arrange()
ggplot2	Data visualization (part of tidyverse)	ggplot(), geom_*() functions, theme_*()
reshape2	Data reshaping between wide/long formats	melt() for correlation matrix
caret	Machine learning model training and evaluation	createDataPartition(), RMSE(), confusionMatrix()
broom	Tidying model outputs	tidy() for regression coefficients
GGally	Advanced visualizations	ggpairs() for scatterplot matrix
ggpubr	Publication-ready plots	ggarrange() for combining plots
glmnet	Regularized regression (Ridge/Lasso)	glmnet(), cv.glmnet()
cluster	Clustering algorithms	daisy(), silhouette()
factoextra	Clustering visualization	fviz_nbclust(), fviz_cluster()
dbscan	Density-based clustering	dbscan(), kNNdistplot()
randomForest	Random Forest classification	randomForest(), importance()
xgboost	Gradient boosting algorithm	xgb.train(), xgb.importance()
fmsb	Radar charts for model comparison	radarchart()

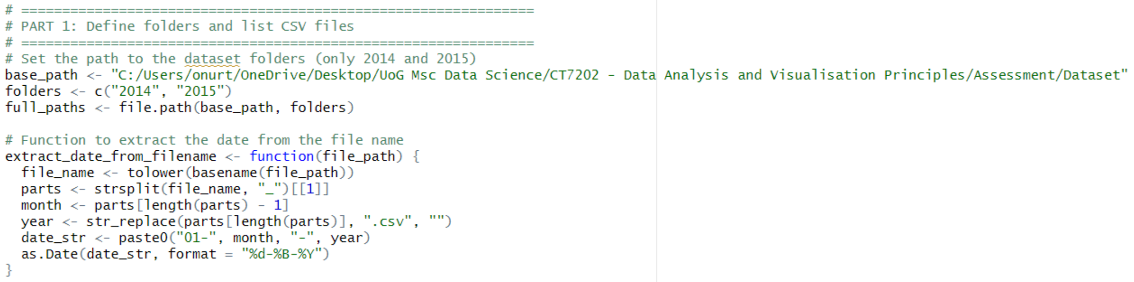
Table 1.1. Overview of R Libraries and Their Applications in the Analysis

2 Data Integration and Preparation

This section covers the processes undertaken to prepare the CPS dataset for analysis. Significant preprocessing steps included verifying data consistency, handling missing values, and standardizing column names and formats to maintain structural integrity. Furthermore, new variables created using feature engineering to enhance the analysis process.

2.1 Loading and Merging Monthly Data Files

In this part, loading and merging data processes have performed. Figure 2.1 indicates the initial setup process for this step. First, the *base_path* variable identifies the directory containing the dataset files, focusing specifically in 2014 and 2015. The folders vector specifies the subdirectories corresponding to these two years, and *full_paths* creates the full directory paths for each year by combining the base path with the folder names.

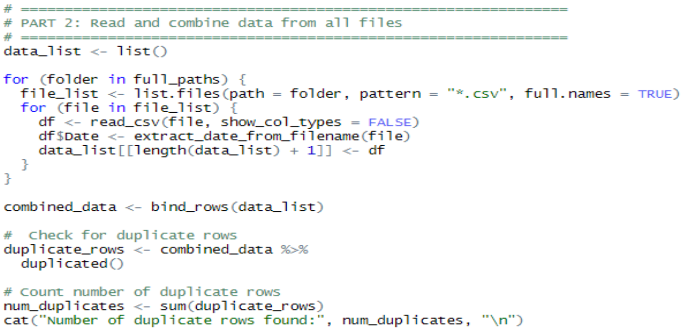


```r
# ==============================================================
# PART 1: Define folders and list CSV files
# ==============================================================
# Set the path to the dataset folders (only 2014 and 2015)
base_path <- "C:/Users/onurt/OneDrive/Desktop/UoG Msc Data Science/CT7202 - Data Analysis and Visualisation Principles/Assessment/Dataset"
folders <- c("2014", "2015")
full_paths <- file.path(base_path, folders)

# Function to extract the date from the file name
extract_date_from_filename <- function(file_path) {
  file_name <- tolower(basename(file_path))
  parts <- strsplit(file_name, "_")[[1]]
  month <- parts[length(parts) - 1]
  year <- str_replace(parts[length(parts)], ".csv", "")
  date_str <- paste0("01-", month, "-", year)
  as.Date(date_str, format = "%d-%B-%Y")
}
```

Figure 2.1. Defining Data Folder Paths and Date Extraction Function

To facilitate temporal analysis, a function called *extract_date_from_filename* developed. This function extracts a date object from each csv file name by parsing the month and year information. This automatic date extraction is critical to correctly aligning data records with their corresponding reporting periods during subsequent analysis. In this way, the first variable that will improve the analysis and visualizations is added while reading the dataset.

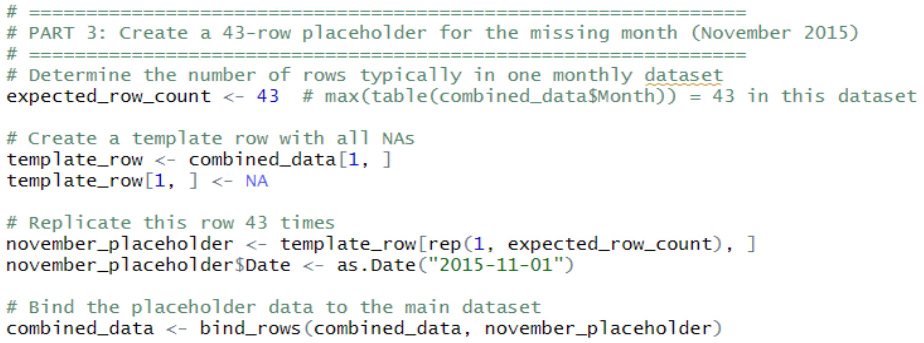
Figure 2.2 given below demonstrates the process of reading multiple monthly csv files into a unified dataset. A list object was used to sequentially store the data frames read from each file. For each dataset, a Date variable was added based on the filename using the previously defined date extraction function. After all files processed, the *bind_rows* function applied to merge the datasets into a single comprehensive data frame called *combined_data*. Thus, the process of reading data from multiple files and adding the Date column is completed. After reading dataset, duplicate records were checked, and no duplicate records were found.



```r
# ==============================================================
# PART 2: Read and combine data from all files
# ==============================================================
data_list <- list()

for (folder in full_paths) {
  file_list <- list.files(path = folder, pattern = "*.csv", full.names = TRUE)
  for (file in file_list) {
    df <- read_csv(file, show_col_types = FALSE)
    df$Date <- extract_date_from_filename(file)
    data_list[[length(data_list) + 1]] <- df
  }
}

combined_data <- bind_rows(data_list)

#  Check for duplicate rows
duplicate_rows <- combined_data %>%
  duplicated()

# Count number of duplicate rows
num_duplicates <- sum(duplicate_rows)
cat("Number of duplicate rows found:", num_duplicates, "\n")
```

Figure 2.2. Read and Combine Data from All Files

Figure 2.3 given below indicates the initial step to address the issue of a missing month (November 2015) in the dataset. To provide a continuous timeline for analysis, a placeholder was created for this missing period. On the other hand, this template was replicated 43 times to match the typical number of monthly rows observed in the data. Each replicated row was assigned the date "01-11-2015" in the Month column. Finally, the placeholder rows were added to the main dataset to ensure that there were no gaps in the monthly series and to maintain the structural consistency of the dataset for further analysis. Additionally, rows for November 2015 have filled with NAs and these NAs will be handled in next steps.



```
# ================================================================
# PART 3: Create a 43-row placeholder for the missing month (November 2015)
# ================================================================
# Determine the number of rows typically in one monthly dataset
expected_row_count <- 43  # max(table(combined_data$Month)) = 43 in this dataset

# Create a template row with all NAs
template_row <- combined_data[1, ]
template_row[1, ] <- NA

# Replicate this row 43 times
november_placeholder <- template_row[rep(1, expected_row_count), ]
november_placeholder$Date <- as.Date("2015-11-01")

# Bind the placeholder data to the main dataset
combined_data <- bind_rows(combined_data, november_placeholder)
```

Figure 2.3. Create Rows for Missing Period

After adding new rows for November 2015, the data was checked monthly. Figure 2.4 shows that each month in the dataset contains 43 rows. This shows the suitability of the operation to the structure of the database.

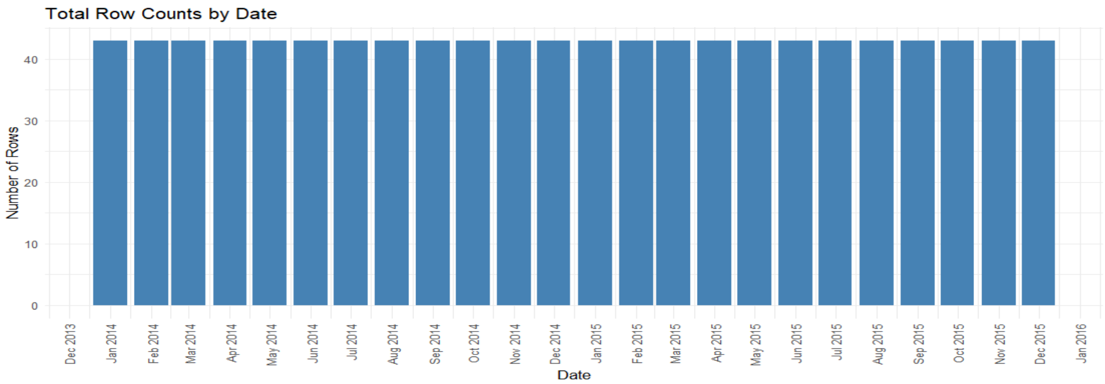
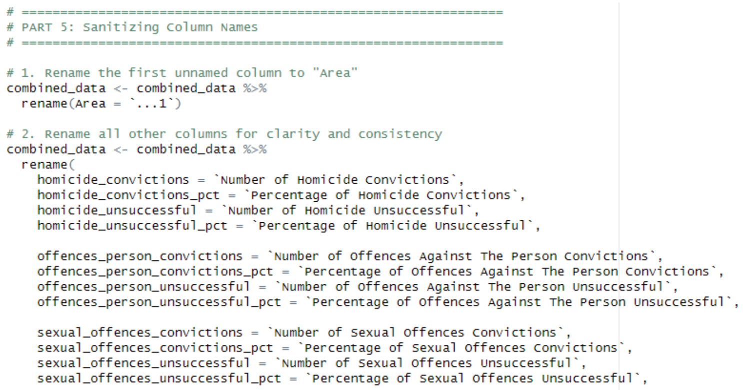


Figure 2.4. Total Row Counts by Date

2.2 Sanitizing Column Names

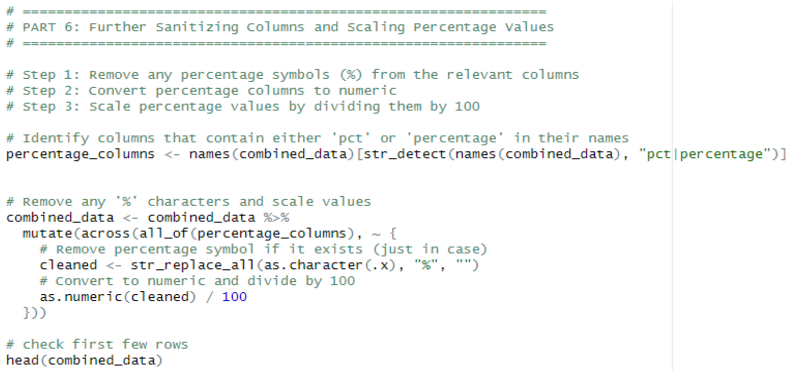
As seen an example in Figure 2.5, the column names were standardized to improve consistency and readability simplify further analysis. First, the unnamed column, *"...1"*, was determined to represent geographic regions and was therefore renamed *Area*, an important area for regional comparisons in later stages. In addition, other column names contained spaces, mixed case, and special characters, which could have made coding and plotting difficult. To address this, all column names were converted to lowercase, spaces were replaced with underscores, and unnecessary characters were removed. Overall, this cleaning step prepared the dataset for more efficient data processing tasks.



```
# ================================================================
# PART 5: Sanitizing Column Names
# ================================================================

# 1. Rename the first unnamed column to "Area"
combined_data <- combined_data %>%
  rename(Area = `...1`)

# 2. Rename all other columns for clarity and consistency
combined_data <- combined_data %>%
  rename(
    homicide_convictions = `Number of Homicide Convictions`,
    homicide_convictions_pct = `Percentage of Homicide Convictions`,
    homicide_unsuccessful = `Number of Homicide Unsuccessful`,
    homicide_unsuccessful_pct = `Percentage of Homicide Unsuccessful`,

    offences_person_convictions = `Number of Offences Against The Person Convictions`,
    offences_person_convictions_pct = `Percentage of Offences Against The Person Convictions`,
    offences_person_unsuccessful = `Number of Offences Against The Person Unsuccessful`,
    offences_person_unsuccessful_pct = `Percentage of Offences Against The Person Unsuccessful`,

    sexual_offences_convictions = `Number of Sexual Offences Convictions`,
    sexual_offences_convictions_pct = `Percentage of Sexual Offences Convictions`,
    sexual_offences_unsuccessful = `Number of Sexual Offences Unsuccessful`,
    sexual_offences_unsuccessful_pct = `Percentage of Sexual Offences Unsuccessful`,
```

Figure 2.5. Coding Process of Sanitizing Column Names

In the next step of data preprocessing, all columns containing *pct* or *percentage* in their dataset names were identified, as seen in Figure 2.6. Once the relevant columns were identified, these fields were prepared for normalization because some percentage values contained embedded *"%"* characters that needed to be removed. By standardizing these columns, the dataset was made more suitable for quantitative analysis and modelling.



```
# ================================================================
# PART 6: Further Sanitizing Columns and Scaling Percentage Values
# ================================================================

# Step 1: Remove any percentage symbols (%) from the relevant columns
# Step 2: Convert percentage columns to numeric
# Step 3: Scale percentage values by dividing them by 100

# Identify columns that contain either 'pct' or 'percentage' in their names
percentage_columns <- names(combined_data)[str_detect(names(combined_data), "pct|percentage")]


# Remove any '%' characters and scale values
combined_data <- combined_data %>%
  mutate(across(all_of(percentage_columns), ~ {
    # Remove percentage symbol if it exists (just in case)
    cleaned <- str_replace_all(as.character(.x), "%", "")
    # Convert to numeric and divide by 100
    as.numeric(cleaned) / 100
  }))

# check first few rows
head(combined_data)
```

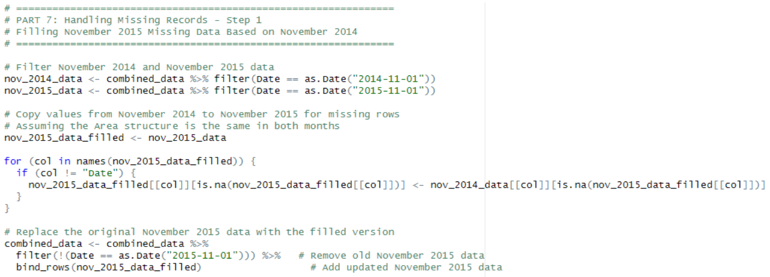
Figure 2.6. Locating and Preparing Percentage Fields for Cleaning

2.3 Handling Missing Data

It is very important to handle missing values because many algorithms are sensitive to missing values. In this study, to complete the 24-month continuous data, the empty parts in November 2015 were filled and 3 columns containing missing values were considered.

2.3.1 Handling Missing Data for November 2015

In this step, missing records in November 2015 were handled by referring to the available data in November 2014. It was observed that the structural patterns between the fields remained relatively consistent between these two periods. Therefore, missing values in November 2015 were imputed using the corresponding entries in November 2014. In this way, this approach aimed to preserve the integrity of the dataset without introducing an external bias. In addition, an alternative method using the averages of the previous 3 months for each field was tested. However, this did not lead to a significant improvement in data quality or consistency. As a result, imputation was performed using the previous year's data and this was chosen as the most reasonable solution to preserve data integrity. Figure 2.7 presents the imputation process in this process.



```r
# ================================================================
# PART 7: Handling Missing Records - Step 1
# Filling November 2015 Missing Data Based on November 2014
# ================================================================

# Filter November 2014 and November 2015 data
nov_2014_data <- combined_data %>% filter(Date == as.Date("2014-11-01"))
nov_2015_data <- combined_data %>% filter(Date == as.Date("2015-11-01"))

# Copy values from November 2014 to November 2015 for missing rows
# Assuming the Area structure is the same in both months
nov_2015_data_filled <- nov_2015_data

for (col in names(nov_2015_data_filled)) {
  if (col != "Date") {
    nov_2015_data_filled[[col]][is.na(nov_2015_data_filled[[col]])] <- nov_2014_data[[col]][is.na(nov_2015_data_filled[[col]])]
  }
}

# Replace the original November 2015 data with the filled version
combined_data <- combined_data %>%
  filter(!(Date == as.Date("2015-11-01"))) %>%   # Remove old November 2015 data
  bind_rows(nov_2015_data_filled)                  # Add updated November 2015 data
```

Figure 2.7. Imputation of November 2015 Data Based on Historical Records

2.3.2 Handling Missing Columns

Figure 2.8 given below shows the number of missing values in three variables with missing data. Accordingly, there are 402, 402 and 13 missing values in the column's *homicide_convictions_pct*, *homicide_unsuccessful_pct* and *percentage_l_motoring_unsuccessful*.

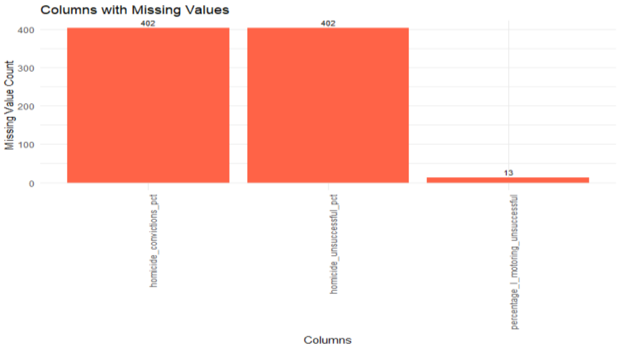
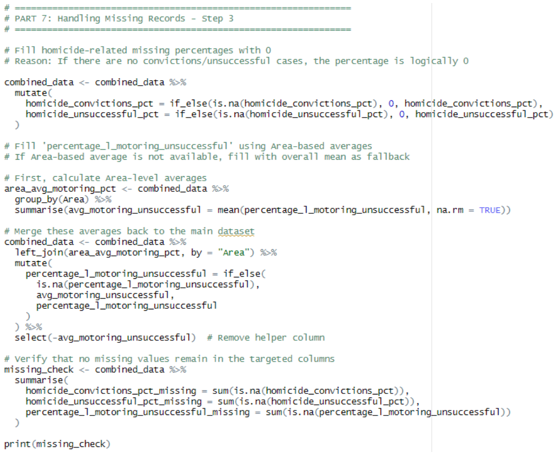


Figure 2.8. Bar Chart of Missing Values

After visualizing the missing values, the handling process was performed by applying the steps in Figure 2.9. In detail, the missing values in the *homicide_convictions_pct* and *homicide_unsuccessful_pct* fields were logically replaced with 0, since the presence of 0 in the dataset of recorded convictions or unsuccessful cases implied a zero percentage. On the other hand, a more detailed approach was applied for the *percentage_l_motoring_unsuccessful* variable. In this step, missing values were filled using the Area-based average percentage. Furthermore, if an area-specific average was not available, the grand average across all Areas was used as a filling value.

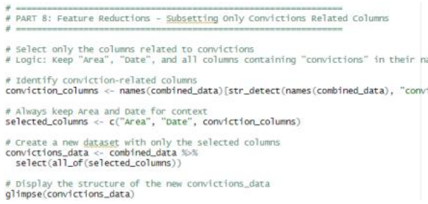


```r
# ================================================================
# PART 7: Handling Missing Records - Step 3
# ================================================================

# Fill homicide-related missing percentages with 0
# Reason: If there are no convictions/unsuccessful cases, the percentage is logically 0

combined_data <- combined_data %>%
  mutate(
    homicide_convictions_pct = if_else(is.na(homicide_convictions_pct), 0, homicide_convictions_pct),
    homicide_unsuccessful_pct = if_else(is.na(homicide_unsuccessful_pct), 0, homicide_unsuccessful_pct)
  )

# Fill 'percentage_l_motoring_unsuccessful' using Area-based averages
# If Area-based average is not available, fill with overall mean as fallback

# First, calculate Area-level averages
area_avg_motoring_pct <- combined_data %>%
  group_by(Area) %>%
  summarise(avg_motoring_unsuccessful = mean(percentage_l_motoring_unsuccessful, na.rm = TRUE))

# Merge these averages back to the main dataset
combined_data <- combined_data %>%
  left_join(area_avg_motoring_pct, by = "Area") %>%
  mutate(
    percentage_l_motoring_unsuccessful = if_else(
      is.na(percentage_l_motoring_unsuccessful),
      avg_motoring_unsuccessful,
      percentage_l_motoring_unsuccessful
    )
  ) %>%
  select(-avg_motoring_unsuccessful)  # Remove helper column

# Verify that no missing values remain in the targeted columns
missing_check <- combined_data %>%
  summarise(
    homicide_convictions_pct_missing = sum(is.na(homicide_convictions_pct)),
    homicide_unsuccessful_pct_missing = sum(is.na(homicide_unsuccessful_pct)),
    percentage_l_motoring_unsuccessful_missing = sum(is.na(percentage_l_motoring_unsuccessful))
  )

print(missing_check)
```

Figure 2.9. Final Imputation of Missing Percentage Values Using Logical and Area-Based Methods

2.4 Subset Creation Only Convictions Related Columns

In this step, subset creation was performed by selecting only the columns related to convictions to obtain new data for analysis. Specifically, all variables containing the word *convictions* and the basic *Area* and *Date* columns were used. Furthermore, this allowed us to create a more focused dataset by isolating the key information directly related to the number and percentage of convictions. The aim of this reduction was to simplify the dataset for future analyses, making it easier to work with relevant features while removing unnecessary variables. Figure 2.10 indicates this process.

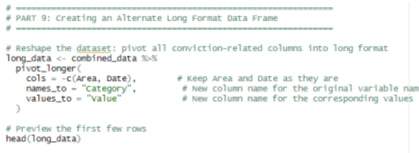


```
# ====================================================================
# PART 8: Feature Reductions - Subsetting Only Convictions Related Columns
# ====================================================================

# Select only the columns related to convictions
# Logic: Keep "Area", "Date", and all columns containing "convictions" in their names

# Identify conviction-related columns
conviction_columns <- names(combined_data)[str_detect(names(combined_data), "convictions")]

# Always keep Area and Date for context
selected_columns <- c("Area", "Date", conviction_columns)

# Create a new dataset with only the selected columns
convictions_data <- combined_data %>%
  select(all_of(selected_columns))

# Display the structure of the new convictions_data
glimpse(convictions_data)
```

Figure 2.10. Subset Creation Focused on Conviction Features

2.5 Creating an Alternate Long Format Data Frame

In this step, the dataset was converted from wide format to long format, as shown in Figure 2.11. Initially, the dataset had many separate columns for different conviction types and their percentages, making analysis and visualization more complex. In this step, all columns except *Area* and *Date* were combined into two columns. This restructuring significantly simplified the dataset structure, allowing for easier filtering and grouping.



```
# ====================================================================
# PART 9: Creating an Alternate Long Format Data Frame
# ====================================================================

# Reshape the dataset: pivot all conviction-related columns into long format
long_data <- combined_data %>%
  pivot_longer(
    cols = -c(Area, Date),        # Keep Area and Date as they are
    names_to = "Category",         # New column name for the original variable names
    values_to = "Value"            # New column name for the corresponding values
  )

# Preview the first few rows
head(long_data)
```

Figure 2.11. Transforming the Dataset to Long Format

2.6 Creating an Alternate Aggregated Data

In this step, an alternative dataset was created by aggregating all convictions and unsuccessful cases into two total columns, *total_convictions* and *total_unsuccessful*. Columns that contained only total numbers were selected, ending with *_convictions* and *_unsuccessful*, and fields based on percentages were excluded. To simplify the data, aggregation was applied row by row for each *Area* and *Date* combination. A bar chart was created to visualize the total number of convictions per field, highlighting differences between regions in terms of case outcomes as given in Figure 2.12.

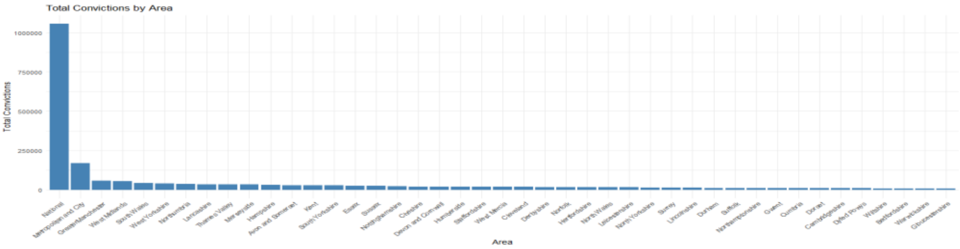
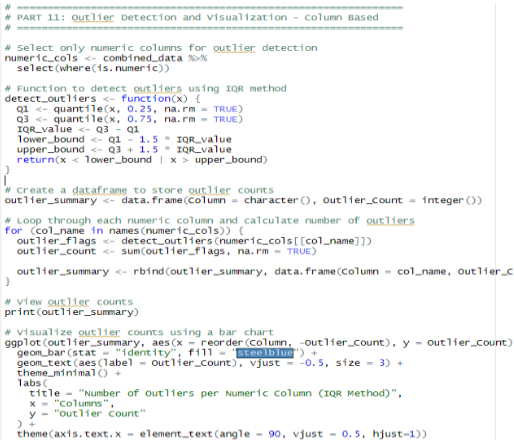


Figure 2.12. Total Convictions Aggregated by Area

2.7 Outlier Detection & Handling

In this step, outlier detection was performed for all numeric variables in the dataset using the IQR method. Figure 2.13 illustrates the R code developed for systematically detecting and counting the number of outliers per numeric column. This step aimed to ensure that any outlier values, which might influence later analyses, were identified in a structured way.



```
# ===============================================================
# PART 11: Outlier Detection and Visualization - Column Based
# ===============================================================

# Select only numeric columns for outlier detection
numeric_cols <- combined_data %>%
  select(where(is.numeric))

# Function to detect outliers using IQR method
detect_outliers <- function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR_value <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR_value
  upper_bound <- Q3 + 1.5 * IQR_value
  return(x < lower_bound | x > upper_bound)
}

# Create a dataframe to store outlier counts
outlier_summary <- data.frame(Column = character(), Outlier_Count = integer())

# Loop through each numeric column and calculate number of outliers
for (col_name in names(numeric_cols)) {
  outlier_flags <- detect_outliers(numeric_cols[[col_name]])
  outlier_count <- sum(outlier_flags, na.rm = TRUE)

  outlier_summary <- rbind(outlier_summary, data.frame(Column = col_name, Outlier_Count = outlier_count))
}

# View outlier counts
print(outlier_summary)

# Visualize outlier counts using a bar chart
ggplot(outlier_summary, aes(x = reorder(Column, -Outlier_Count), y = Outlier_Count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  geom_text(aes(label = Outlier_Count), vjust = -0.5, size = 3) +
  theme_minimal() +
  labs(
    title = "Number of Outliers per Numeric Column (IQR Method)",
    x = "Columns",
    y = "Outlier Count"
  ) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

Figure 2.13. Process of Outlier Detection

Finally, the results of the outlier detection process are summarized in Figure 2.14. The bar chart shows the number of outliers detected in each numeric column. Although several variables indicated notable numbers of outliers, no data handling step was performed. This decision was made because the dataset originates from an official source and represents real-world observations. Besides, the dataset size was quite small. Modifying these records could risk distorting the integrity and authenticity of the data.

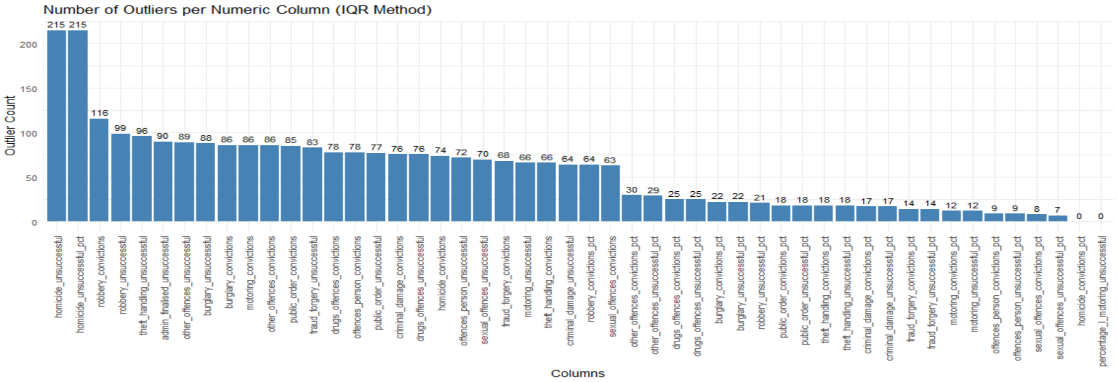
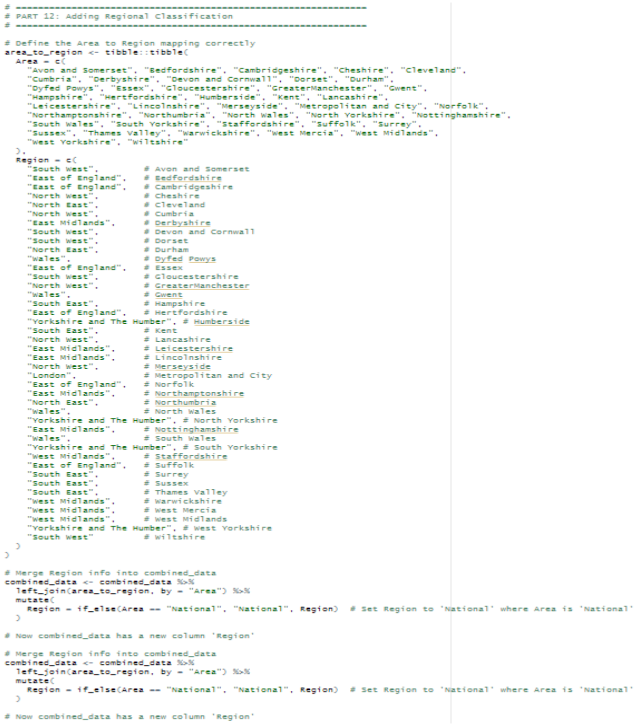


Figure 2.14. Number of Outliers Using IQR Method

2.8 Feature Engineering

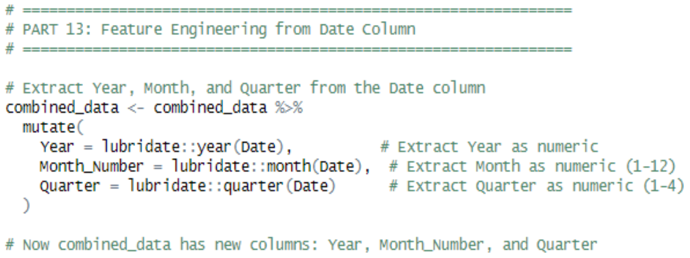
To enhance the analytical capabilities of the dataset, a new feature called *Region* was created based on the *Area* variable. Each area was mapped to its corresponding geographical region in England and Wales. This mapping was done automatically using the area information, ensuring consistency across the dataset. Figure 2.15 indicates the regional classification process and how it is incorporated into the dataset.



```r
# ===============================================================
# PART 12: Adding Regional Classification
# ===============================================================

# Define the Area to Region mapping correctly
area_to_region <- tibble::tibble(
  Area = c(
    "Avon and Somerset", "Bedfordshire", "Cambridgeshire", "Cheshire", "Cleveland",
    "Cumbria", "Derbyshire", "Devon and Cornwall", "Dorset", "Durham",
    "Dyfed Powys", "Essex", "Gloucestershire", "GreaterManchester", "Gwent",
    "Hampshire", "Hertfordshire", "Humberside", "Kent", "Lancashire",
    "Leicestershire", "Lincolnshire", "Merseyside", "Metropolitan and City", "Norfolk",
    "Northamptonshire", "Northumbria", "North Wales", "North Yorkshire", "Nottinghamshire",
    "South Wales", "South Yorkshire", "Staffordshire", "Suffolk", "Surrey",
    "Sussex", "Thames Valley", "Warwickshire", "West Mercia", "West Midlands",
    "West Yorkshire", "Wiltshire"
  ),
  Region = c(
    "South West",         # Avon and Somerset
    "East of England",    # Bedfordshire
    "East of England",    # Cambridgeshire
    "North West",         # Cheshire
    "North East",         # Cleveland
    "North West",         # Cumbria
    "East Midlands",      # Derbyshire
    "South West",         # Devon and Cornwall
    "South West",         # Dorset
    "North East",         # Durham
    "Wales",              # Dyfed Powys
    "East of England",    # Essex
    "South West",         # Gloucestershire
    "North West",         # GreaterManchester
    "Wales",              # Gwent
    "South East",         # Hampshire
    "East of England",    # Hertfordshire
    "Yorkshire and The Humber", # Humberside
    "South East",         # Kent
    "North West",         # Lancashire
    "East Midlands",      # Leicestershire
    "East Midlands",      # Lincolnshire
    "North West",         # Merseyside
    "London",             # Metropolitan and City
    "East of England",    # Norfolk
    "East Midlands",      # Northamptonshire
    "North East",         # Northumbria
    "Wales",              # North Wales
    "Yorkshire and The Humber", # North Yorkshire
    "East Midlands",      # Nottinghamshire
    "Wales",              # South Wales
    "Yorkshire and The Humber", # South Yorkshire
    "West Midlands",      # Staffordshire
    "East of England",    # Suffolk
    "South East",         # Surrey
    "South East",         # Sussex
    "South East",         # Thames Valley
    "West Midlands",      # Warwickshire
    "West Midlands",      # West Mercia
    "West Midlands",      # West Midlands
    "Yorkshire and The Humber", # West Yorkshire
    "South West"          # Wiltshire
  )
)

# Merge Region info into combined_data
combined_data <- combined_data %>%
  left_join(area_to_region, by = "Area") %>%
  mutate(
    Region = if_else(Area == "National", "National", Region)  # Set Region to 'National' where Area is 'National'
  )

# Now combined_data has a new column 'Region'

# Merge Region info into combined_data
combined_data <- combined_data %>%
  left_join(area_to_region, by = "Area") %>%
  mutate(
    Region = if_else(Area == "National", "National", Region)  # Set Region to 'National' where Area is 'National'
  )

# Now combined_data has a new column 'Region'
```

Figure 2.15. Integration of Regional Information into Dataset

Furthermore, feature engineering was performed on the *Date* variable to extract additional temporal attributes, as shown in Figure 2.16. Three numeric variables were created, *Year*, *Month_Number*, and *Quarter*. The creation of these variables improves the structure of the dataset for temporal analysis. Additionally, this allows to make more detailed examination of seasonal patterns.

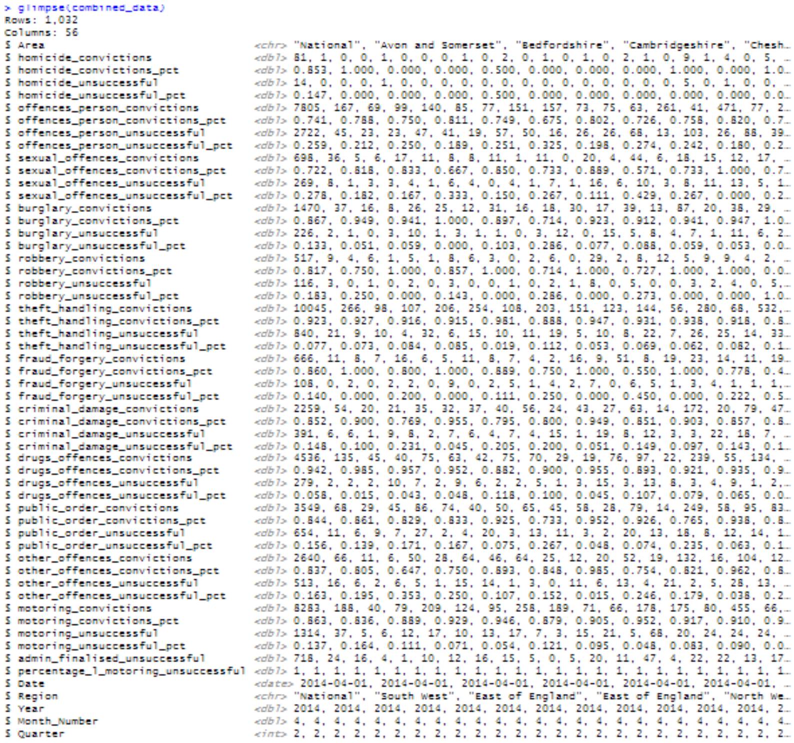


```r
# ===============================================================
# PART 13: Feature Engineering from Date Column
# ===============================================================

# Extract Year, Month, and Quarter from the Date column
combined_data <- combined_data %>%
  mutate(
    Year = lubridate::year(Date),           # Extract Year as numeric
    Month_Number = lubridate::month(Date),  # Extract Month as numeric (1-12)
    Quarter = lubridate::quarter(Date)      # Extract Quarter as numeric (1-4)
  )

# Now combined_data has new columns: Year, Month_Number, and Quarter
```

Figure 2.16. Feature Engineering on Date Column to Extract Temporal Variables

2.9 Controlling Data Types

After completing the feature engineering steps, the data types of all variables in the dataset were carefully examined. It was verified that character-based fields such as *Area* and *Region* were correctly assigned as character types. Moreover, numeric fields representing convictions counts and percentages were properly stored in double types. Additionally, added temporal features were also verified to be of double type. Figure 2.17 indicates the structure of the columns in dataset. Overall, the structure of the dataset was now found to be fully suitable for subsequent descriptive and predictive modelling tasks.



```
> glimpse(combined_data)
Rows: 1,032
Columns: 56
$ Area                               <chr> "National", "Avon and Somerset", "Bedfordshire", "Cambridgeshire", "Chesh…
$ homicide_convictions               <dbl> 81, 1, 0, 0, 1, 0, 0, 0, 1, 0, 2, 0, 1, 0, 1, 0, 2, 1, 0, 9, 1, 4, 0, 5, …
$ homicide_convictions_pct           <dbl> 0.853, 1.000, 0.000, 0.000, 0.500, 0.000, 0.000, 0.000, 1.000, 0.000, 1.0…
$ homicide_unsuccessful              <dbl> 14, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5, 0, 1, 0, 0, …
$ homicide_unsuccessful_pct          <dbl> 0.147, 0.000, 0.000, 0.000, 0.500, 0.000, 0.000, 0.000, 0.000, 0.000, 0.0…
$ offences_person_convictions        <dbl> 7805, 167, 69, 99, 140, 85, 77, 151, 157, 73, 75, 63, 261, 41, 471, 77, 2…
$ offences_person_convictions_pct    <dbl> 0.741, 0.788, 0.750, 0.811, 0.749, 0.675, 0.802, 0.726, 0.758, 0.820, 0.7…
$ offences_person_unsuccessful       <dbl> 2722, 45, 23, 23, 47, 41, 19, 57, 50, 16, 26, 26, 68, 13, 103, 26, 88, 39…
$ offences_person_unsuccessful_pct   <dbl> 0.259, 0.212, 0.250, 0.189, 0.251, 0.325, 0.198, 0.274, 0.242, 0.180, 0.2…
$ sexual_offences_convictions        <dbl> 698, 36, 5, 6, 17, 11, 8, 8, 11, 1, 11, 0, 20, 4, 44, 6, 18, 15, 12, 17, …
$ sexual_offences_convictions_pct    <dbl> 0.722, 0.818, 0.833, 0.667, 0.850, 0.733, 0.889, 0.571, 0.733, 1.000, 0.7…
$ sexual_offences_unsuccessful       <dbl> 269, 8, 1, 3, 3, 4, 1, 6, 4, 0, 4, 1, 7, 1, 16, 6, 10, 3, 8, 11, 13, 5, 1…
$ sexual_offences_unsuccessful_pct   <dbl> 0.278, 0.182, 0.167, 0.333, 0.150, 0.267, 0.111, 0.429, 0.267, 0.000, 0.2…
$ burglary_convictions               <dbl> 1470, 37, 16, 8, 26, 25, 12, 31, 16, 18, 30, 17, 39, 13, 87, 20, 38, 29, …
$ burglary_convictions_pct           <dbl> 0.867, 0.949, 0.941, 1.000, 0.897, 0.714, 0.923, 0.912, 0.941, 0.947, 1.0…
$ burglary_unsuccessful              <dbl> 226, 2, 1, 0, 3, 10, 1, 3, 1, 1, 0, 3, 12, 0, 15, 5, 8, 4, 7, 1, 11, 6, 2…
$ burglary_unsuccessful_pct          <dbl> 0.133, 0.051, 0.059, 0.000, 0.103, 0.286, 0.077, 0.088, 0.059, 0.053, 0.0…
$ robbery_convictions                <dbl> 517, 9, 4, 6, 1, 5, 1, 8, 6, 3, 0, 2, 6, 0, 29, 2, 8, 12, 5, 9, 9, 4, 2, …
$ robbery_convictions_pct            <dbl> 0.817, 0.750, 1.000, 0.857, 1.000, 0.714, 1.000, 0.727, 1.000, 1.000, 0.0…
$ robbery_unsuccessful               <dbl> 116, 3, 0, 1, 0, 2, 0, 3, 0, 0, 1, 0, 2, 1, 8, 0, 5, 0, 0, 3, 2, 4, 0, 5,…
$ robbery_unsuccessful_pct           <dbl> 0.183, 0.250, 0.000, 0.143, 0.000, 0.286, 0.000, 0.273, 0.000, 0.000, 1.0…
$ theft_handling_convictions         <dbl> 10045, 266, 98, 107, 206, 254, 108, 203, 151, 123, 144, 56, 280, 68, 532,…
$ theft_handling_convictions_pct     <dbl> 0.923, 0.927, 0.916, 0.915, 0.981, 0.888, 0.947, 0.931, 0.938, 0.918, 0.8…
$ theft_handling_unsuccessful        <dbl> 840, 21, 9, 10, 4, 32, 6, 15, 10, 11, 19, 5, 10, 8, 22, 7, 26, 25, 14, 33…
$ theft_handling_unsuccessful_pct    <dbl> 0.077, 0.073, 0.084, 0.085, 0.019, 0.112, 0.053, 0.069, 0.062, 0.082, 0.1…
$ fraud_forgery_convictions          <dbl> 666, 11, 8, 7, 16, 6, 5, 11, 8, 7, 4, 2, 16, 9, 51, 8, 19, 23, 14, 11, 19…
$ fraud_forgery_convictions_pct      <dbl> 0.860, 1.000, 0.800, 1.000, 0.889, 0.750, 1.000, 0.550, 1.000, 0.778, 0.4…
$ fraud_forgery_unsuccessful         <dbl> 108, 0, 2, 0, 2, 2, 0, 9, 0, 2, 5, 1, 4, 2, 7, 0, 6, 5, 1, 3, 4, 1, 1, 1,…
$ fraud_forgery_unsuccessful_pct     <dbl> 0.140, 0.000, 0.200, 0.000, 0.111, 0.250, 0.000, 0.450, 0.000, 0.222, 0.5…
$ criminal_damage_convictions        <dbl> 2259, 54, 20, 21, 35, 32, 37, 40, 56, 24, 43, 27, 63, 14, 172, 20, 79, 47…
$ criminal_damage_convictions_pct    <dbl> 0.852, 0.900, 0.769, 0.955, 0.795, 0.800, 0.949, 0.851, 0.903, 0.857, 0.8…
$ criminal_damage_unsuccessful       <dbl> 391, 6, 6, 1, 9, 8, 2, 7, 6, 4, 7, 4, 15, 1, 19, 8, 12, 3, 3, 22, 18, 7, …
$ criminal_damage_unsuccessful_pct   <dbl> 0.148, 0.100, 0.231, 0.045, 0.205, 0.200, 0.051, 0.149, 0.097, 0.143, 0.1…
$ drugs_offences_convictions         <dbl> 4536, 135, 45, 40, 75, 63, 42, 75, 70, 29, 19, 76, 97, 22, 239, 55, 134, …
$ drugs_offences_convictions_pct     <dbl> 0.942, 0.985, 0.957, 0.952, 0.882, 0.900, 0.955, 0.893, 0.921, 0.935, 0.9…
$ drugs_offences_unsuccessful        <dbl> 279, 2, 2, 2, 10, 7, 2, 9, 6, 2, 2, 5, 1, 3, 15, 3, 13, 8, 3, 4, 9, 1, 2,…
$ drugs_offences_unsuccessful_pct    <dbl> 0.058, 0.015, 0.043, 0.048, 0.118, 0.100, 0.045, 0.107, 0.079, 0.065, 0.0…
$ public_order_convictions           <dbl> 3549, 68, 29, 45, 86, 74, 40, 50, 65, 45, 58, 28, 79, 14, 249, 58, 95, 83…
$ public_order_convictions_pct       <dbl> 0.844, 0.861, 0.829, 0.833, 0.925, 0.733, 0.952, 0.926, 0.765, 0.938, 0.8…
$ public_order_unsuccessful          <dbl> 654, 11, 6, 9, 7, 27, 2, 4, 20, 3, 13, 11, 3, 2, 20, 13, 18, 8, 12, 14, 1…
$ public_order_unsuccessful_pct      <dbl> 0.156, 0.139, 0.171, 0.167, 0.075, 0.267, 0.048, 0.074, 0.235, 0.063, 0.1…
$ other_offences_convictions         <dbl> 2640, 66, 11, 6, 50, 28, 64, 46, 64, 25, 12, 20, 52, 19, 132, 16, 104, 12…
$ other_offences_convictions_pct     <dbl> 0.837, 0.805, 0.647, 0.750, 0.893, 0.848, 0.985, 0.754, 0.821, 0.962, 0.8…
$ other_offences_unsuccessful        <dbl> 513, 16, 6, 2, 6, 5, 1, 15, 14, 1, 3, 0, 11, 6, 13, 4, 21, 2, 5, 28, 13, …
$ other_offences_unsuccessful_pct    <dbl> 0.163, 0.195, 0.353, 0.250, 0.107, 0.152, 0.015, 0.246, 0.179, 0.038, 0.2…
$ motoring_convictions               <dbl> 8283, 188, 40, 79, 209, 124, 95, 258, 189, 71, 66, 178, 175, 80, 455, 66,…
$ motoring_convictions_pct           <dbl> 0.863, 0.836, 0.889, 0.929, 0.946, 0.879, 0.905, 0.952, 0.917, 0.910, 0.9…
$ motoring_unsuccessful              <dbl> 1314, 37, 5, 6, 12, 17, 10, 13, 17, 7, 3, 15, 21, 5, 68, 20, 24, 24, 24, …
$ motoring_unsuccessful_pct          <dbl> 0.137, 0.164, 0.111, 0.071, 0.054, 0.121, 0.095, 0.048, 0.083, 0.090, 0.0…
$ admin_finalised_unsuccessful       <dbl> 718, 24, 16, 4, 1, 10, 12, 16, 15, 5, 0, 5, 20, 11, 47, 4, 22, 22, 13, 17…
$ percentage_1_motoring_unsuccessful <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
$ Date                               <date> 2014-04-01, 2014-04-01, 2014-04-01, 2014-04-01, 2014-04-01, 2014-04-01, …
$ Region                             <chr> "National", "South West", "East of England", "East of England", "North We…
$ Year                               <dbl> 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2…
$ Month_Number                       <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4…
$ Quarter                            <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2…
```

Figure 2.17. Summary of Dataset Variables and Types

3 Descriptive Analysis

In this section, a descriptive analysis is conducted to provide an initial understanding of the basic characteristics of the dataset and the patterns that emerge. This analysis aims to answer the question of what happened in the past data.

3.1 Summary Statistics

The analysis first examined the summary statistics. In detail, it is found that crimes against persons, theft and handling offences and motoring offences show the highest average number of convictions, with averages exceeding 400 cases. For example, crimes against persons have an average conviction value of 437.7, theft offences have 445.6 and driving offences have 401.8. As seen in Figure 3.1, this suggest that these crime categories contribute significantly to the overall case volume.

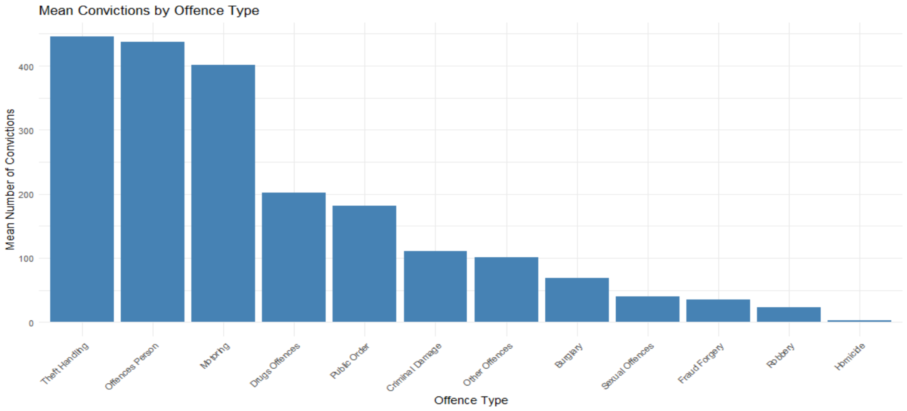


Figure 3.1. Mean Convictions by Offence Type

Furthermore, conviction success rates are generally high for most crime types, typically ranging between 75% and 95%. On the other hand, drug offences achieved a significantly high conviction rate of approximately 86.5%, while a relatively low average number of unsuccessful cases was maintained. Figure 3.2 indicates the conviction and unsuccessful rates by offence type. However, sexual offences stand out as an area of concern. While the conviction rate for sexual offences remains at approximately 74.8%, the average number of unsuccessful cases is 15,7 cases.

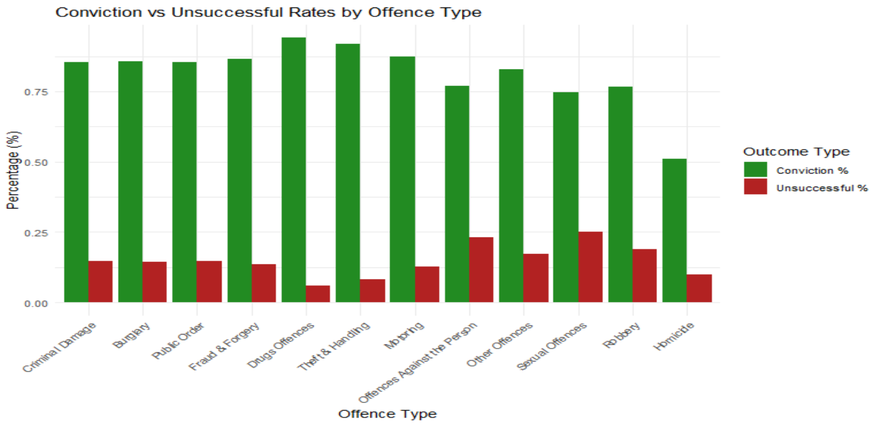
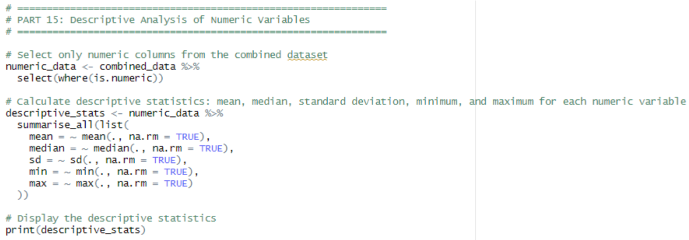


Figure 3.2. Conviction vs Unsuccessful Rates by Offence Type

The percentage of unsuccessful cases (25.1%) is higher than for other crime categories. This may reflect the inherent complexities associated with the prosecution of sexual offences. Overall, although most crime types show strong conviction rates, issues such as sexual offences and administrative closures should be examined in detail in subsequent analyses. Figure 3.3 indicates the calculation process of statistical values of the numerical variables.



```r
# ================================================================
# PART 15: Descriptive Analysis of Numeric Variables
# ================================================================

# Select only numeric columns from the combined dataset
numeric_data <- combined_data %>%
  select(where(is.numeric))

# Calculate descriptive statistics: mean, median, standard deviation, minimum, and maximum for each numeric variable
descriptive_stats <- numeric_data %>%
  summarise_all(list(
    mean = ~ mean(., na.rm = TRUE),
    median = ~ median(., na.rm = TRUE),
    sd = ~ sd(., na.rm = TRUE),
    min = ~ min(., na.rm = TRUE),
    max = ~ max(., na.rm = TRUE)
  ))

# Display the descriptive statistics
print(descriptive_stats)
```

Figure 3.3 Calculation of Statistical Values

3.2 Data Distributions

In this section, data distribution was analysed, and different visualization techniques were used. First, a pie chart was created showing the proportional distribution of the various types of convictions in the dataset, as shown in Figure 3.4. The visualization reveals a significant concentration of convictions across several dominant crime categories. Theft handling convictions account for the largest share, accounting for approximately 21.7% of all convictions recorded. This is followed by offences person convictions and motoring convictions, representing 21.3% and 19.6% respectively. These three categories alone account for over 60% of the total convictions, suggesting a systemic focus or prevalence in these areas of criminal activity. In contrast, categories such as homicide convictions (0.1%), robbery (1.1%), fraud and forgery (1.7%) and sexual offences (1.9%) make up a relatively small proportion of the dataset. While these crimes are generally more violent in nature, their lower frequency may reflect their lower prevalence and higher sophistication in prosecution or reporting practices. To comment, it highlights the importance of resource allocation and policy attention to high-volume crimes such as theft and public order offences, whilst also acknowledging the differential treatment required for less frequent but high-serious offences such as homicide and sexual offences.

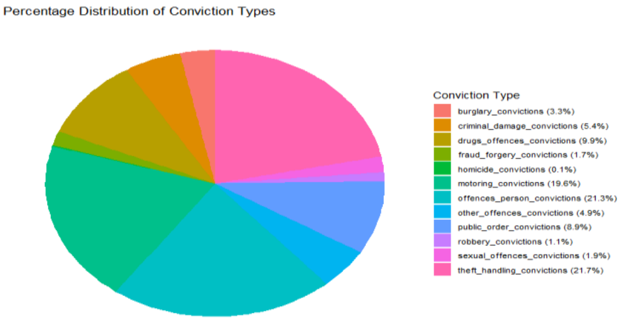


Figure 3.4. Percentage Distribution of Conviction Types

Figure 3.5 presents two histograms showing the frequency distributions of total convictions and unsuccessful convictions across regions and time. The distribution of total convictions is skewed to the right. Most values are concentrated below 2000 cases, suggesting that high conviction counts are relatively rare. Besides, unsuccessful cases are largely concentrated below 500, suggesting that most jurisdictions report low numbers of unsuccessful prosecutions. There are outliers in both distributions, suggesting that they experience significantly higher numbers. This disparity highlights the unequal distribution of criminal case outcomes. This will be addressed in subsequent visualizations when evaluating the data regionally.

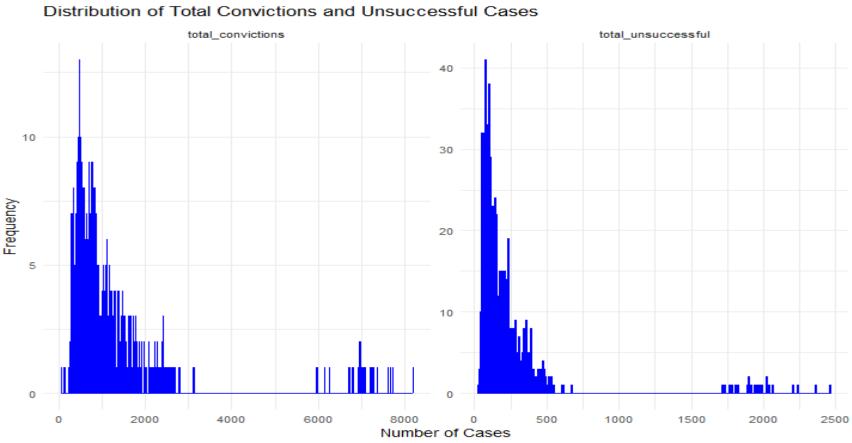


Figure 3.5. Distribution of Total Convictions and Unsuccessful Cases

Figure 3.6 indicates the variation in convictions for offences against the person across different areas using a boxplot. The Metropolitan and City area shows the highest conviction counts with a wider interquartile range. This indicates greater variability within that region. Most other regions exhibit lower median values and tighter distributions, suggesting consistency in conviction numbers. The presence of multiple outliers across several regions reflects exceptional cases with unusually high or low convictions. This variation can be addressed by factors such as population density and crime rates. Overall, the figure emphasizes the regional disparity in handling offences against individuals.

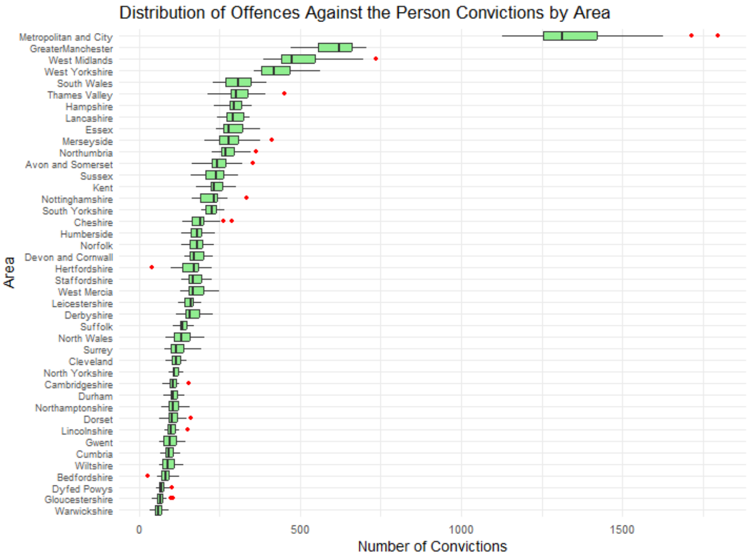


Figure 3.6. Distribution of Offences Against the Person Convictions by Area

Figure 3.7 indicates the distribution of sexual offence convictions across various CPS areas in England and Wales. The boxplot reveals significant regional disparities, with the Metropolitan and City regions showing the widest range and highest average number of convictions. Areas such as Greater

Manchester and West Yorkshire also show relatively high conviction numbers. This can be explained by larger populations a higher reporting rate. In other context, areas such as Gloucestershire, Dyfed Powys and North Wales consistently report lower conviction numbers. The presence of outliers, particularly in metropolitan areas, may reveal variations in convictions due to large-scale investigations.

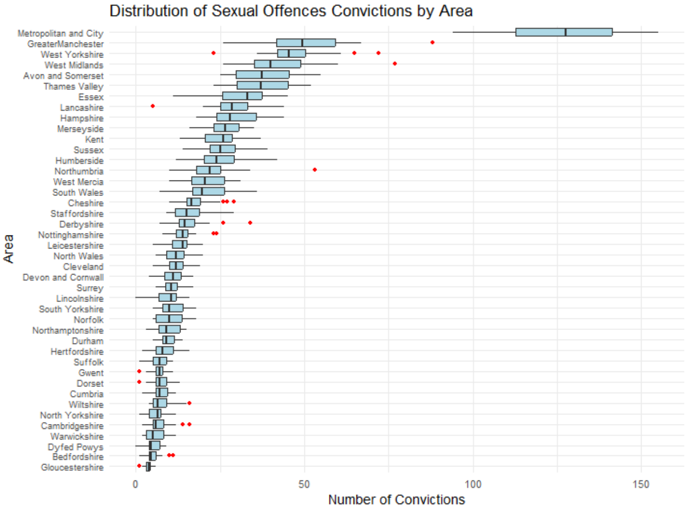


Figure 3.7. Distribution of Sexual Offences Convictions by Area

Figure 3.8 illustrates the log-scaled density distribution of different conviction types. In this part log-scaled applied to numbers considering data volumes and targeting interpretable visualization. This figure allows comparison across offences with vastly different frequencies. The motoring convictions exhibit a sharp peak at lower conviction counts, indicating high frequency but with low variation. Drugs and theft handling convictions show a broader distribution. In contrast, homicide and sexual offences convictions have flatter and more dispersed densities. This indicates their rarity. Overlapping curves also reveal how some categories, such as burglary and offences against the person, share similar density patterns. This plot is significant in identifying which offence types dominate the justice system and how consistently they are processed across jurisdictions.

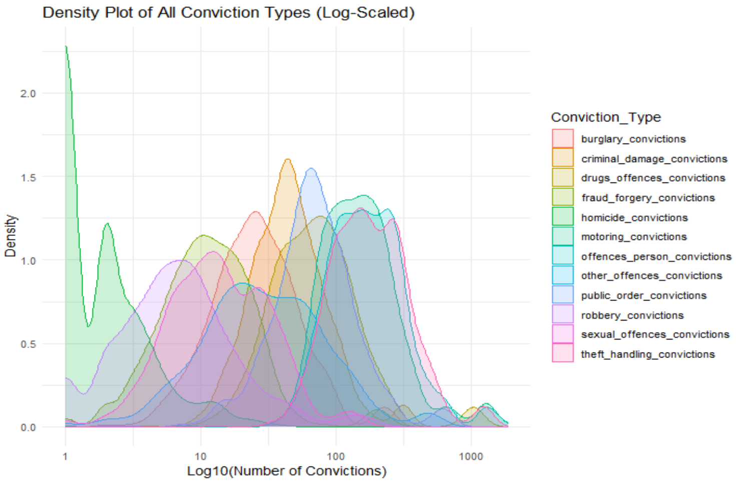


Figure 3.8. Density Plot of All Conviction Types

Conversely, Figure 3.9 presents a log-scaled density plot for various types of unsuccessful cases. The highest density is observed for *motoring_unsuccessful* cases, which peak sharply at lower values. In contrast, homicide, sexual offences, and robbery unsuccessful cases display flatter and more dispersed patterns, this reflects their low frequency but relatively variable distribution. The *admin_finalised_unsuccessful* and *public_order_unsuccessful* types also show moderate densities. It reflects the importance of administrative and procedural factors on case outcomes for these categories.

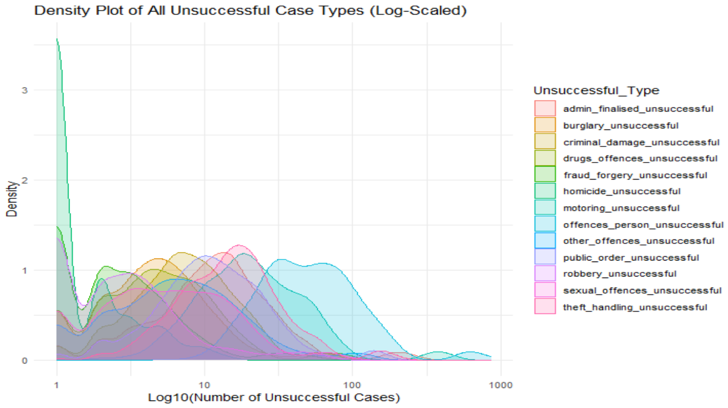


Figure 3.9. Density Plot of All Unsuccessful Case Types

3.3 Visualizations & Observations

This section covers a visual exploration of the dataset to uncover meaningful patterns and relationships. Through a series of time series analyses, regional comparisons, and correlation-based visualizations, this part aims to better understand how convictions outcomes vary across time, regions, and offence types. These visual insights provide critical support for identifying key trends and guiding further interpretation.

3.3.1 Time Series Trends and Visualizations

This part focuses on identifying temporal trends in criminal case outcomes since the dataset consist of time-oriented structure. These time series visualizations help to uncover periodic patterns and long-term shifts within the prosecution process. For example, Figure 3.10 shows national trends in criminal case outcomes between 2014 and 2016. Over this 24-month period, the total number of convictions consistently exceeded 40,000 cases per month. However, a decline was seen towards the end of 2015. In comparison, unsuccessful cases remained significantly lower. They typically amounted to around 8,000 to 10,000 per month. This suggests a stable national prosecution system, with many cases resulting in successful convictions, despite minor changes over time.

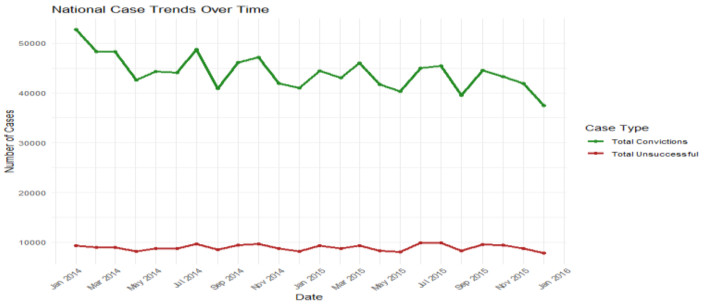


Figure 3.10 Overall (National) Case Trends Over Time

Figure 3.11 given below presents the monthly case outcome trends for each region across England and Wales, excluding national numbers. Across all regions, total convictions showed in green consistently outnumber unsuccessful cases, reflecting a stable pattern of judicial outcomes. However, notable regional differences are visible. For instance, London and the North West report the highest volume of convictions, often exceeding 6,000 and 5,000 cases respectively. In contrast, regions like Wales and the North East show substantially lower volumes, often below 2,500 cases per month. On the other hand, South West and East of England exhibit a slight downward trend in convictions over time, possibly indicating regional shifts in crime volume or prosecution policy.

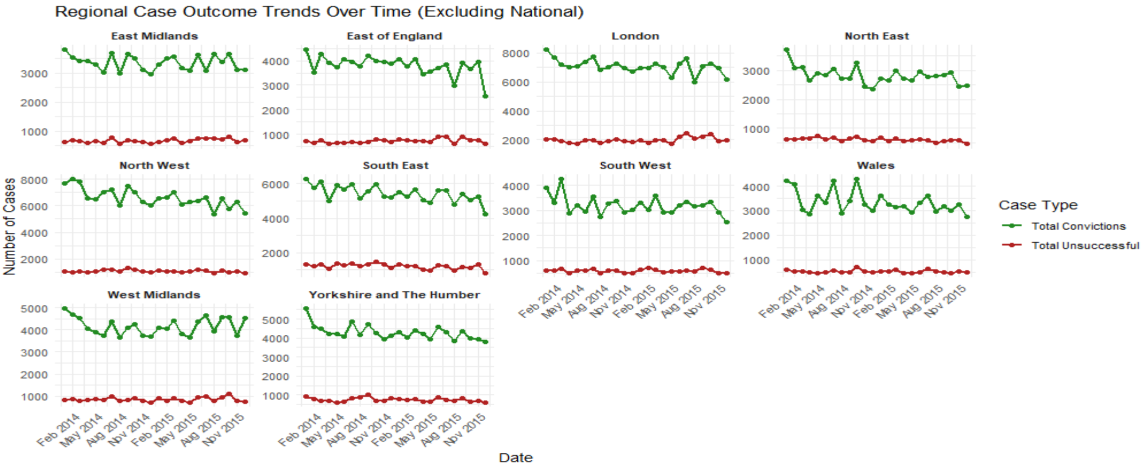


Figure 3.11. Regional Case Outcome Trends Over Time

It can be argued that trends in large areas such as London should be examined, building on Figure 3.5. In this context, Figure 3.12 shows the trends in 24-month conviction percentages for different types of crime in London. Detailly, offences such as motoring and fraud and forgery show high conviction rates more than 90%, indicating strong evidentiary support or simplified case structures. In contrast, conviction rates for drug offences and public order offences are significantly lower and fluctuate more, typically between 60% and 80%. Detailly, this indicates difficulties in gathering evidence or increased complexity in the cases concerned. Homicide, although lower in volume, generally maintains a stable and high conviction rate.

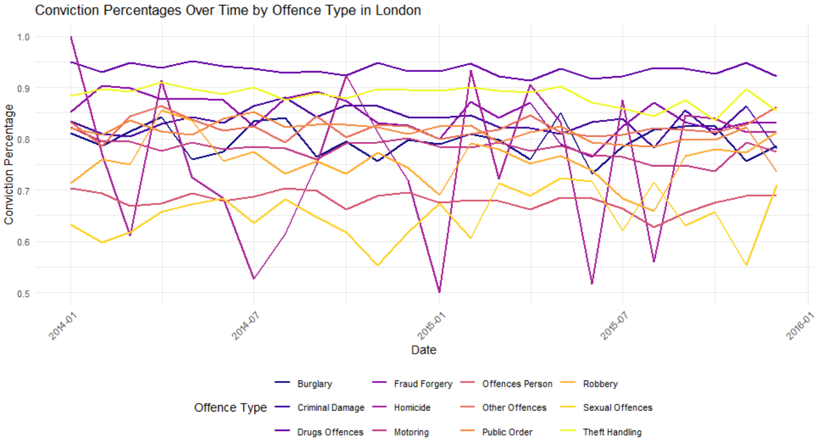


Figure 3.12. Conviction Percentages Over Time by Offence Type in London

Figure 3.13 presents a time series analysis of unsuccessful conviction outcomes across various crime categories in the London area. Detailly, fraud and forgery offences show the highest volatility and peak values in unsuccessful outcomes. These types exceed 40% for periods of approximately 3 months. Sexual offences and robbery offences, typically show high levels of unsuccessful outcomes, ranging from 25% to 35%. Additionally, unsuccessful outcomes peak towards the end of the year, reflecting the seasonal trend complexity associated with these cases. In contrast, motoring offences show the lowest unsuccessful outcome rate over the observed period, generally remaining below 10%. This data can be attributable to the stricter evidentiary standards of motoring offences. While most crime types show relative stability over time, some crimes, such as public order and theft, show moderate fluctuations that may warrant more contextual investigation. Overall, this visualization highlights not only cross-crime differences in conviction rates, but also temporal dynamics that may reflect legal factors.

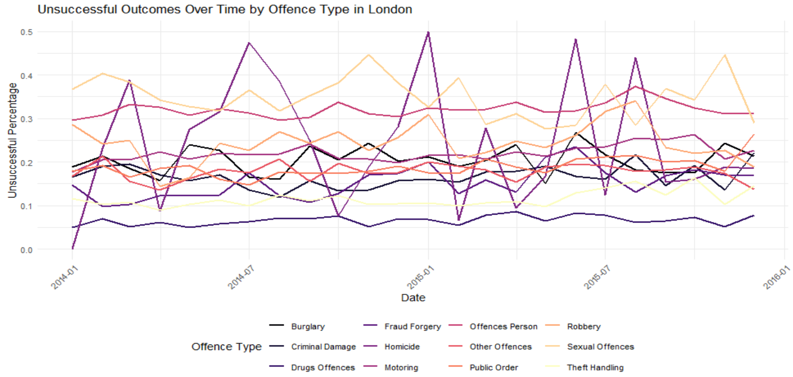


Figure 3.13. Unsuccessful Outcome Percentages Over Time by Offence Type in London

3.3.2 Correlations and Heatmaps

Figure 3.14 presents a heatmap of total convictions across various UK areas from 2014 to 2016. The Metropolitan area shows consistently high conviction levels, peaking above 8000. This indicates a significant concentration of criminal cases. Other areas like West Yorkshire, Greater Manchester, and West Midlands also exhibit moderate conviction rates around 2000. In contrast, countryside such as Dyfed Powys, Cumbria, and Wiltshire show notably lower figures. This underlines differences in population density and crime rates.

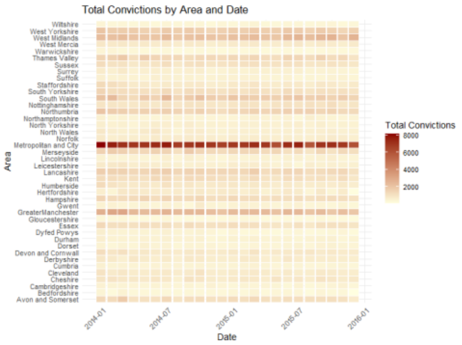


Figure 3.14. Total Convictions by Area and Date

Figure 3.15 displays the correlation heatmap of numeric variables related to convictions and unsuccessful cases. Most variables show a strong positive correlation. This indicates that increases in one crime type's convictions or unsuccessful cases are often mirrored by others. Particularly, variables like motoring convictions, drugs offences convictions, and theft handling convictions are highly correlated with their respective unsuccessful categories. This suggests consistent patterns in case outcomes across crime types.

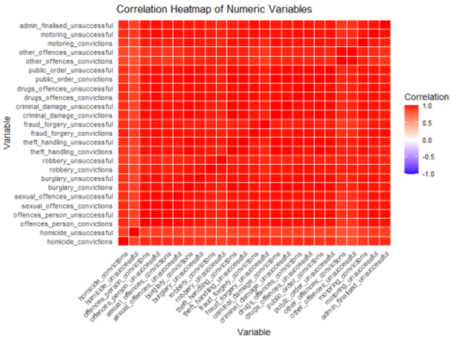


Figure 3.15. Correlation Heatmap of Numeric Variables

3.3.3 Regional and Overall Number Analysis

Figure 3.16 illustrates conviction trends over time for the top 10 areas. The Metropolitan area consistently leads with the highest conviction counts. This area showed periodic fluctuations but remained above 6000. Other areas like West Midlands, Greater Manchester, and West Yorkshire follow at a much lower scale, indicating regional disparities. Overall, most areas maintain relatively stable trends, suggesting persistent crime patterns throughout the observed period.

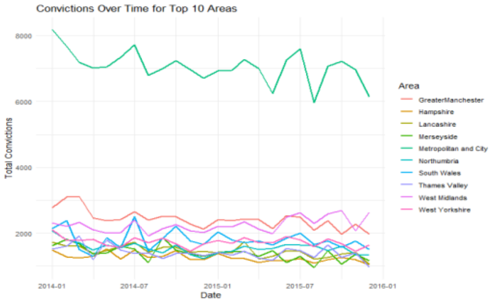


Figure 3.16. Convictions Over Time for Top 10 Areas

Figure 3.17 displays the top 10 crime categories by convictions within the Metropolitan and City area. This region was selected due to its high conviction density observed in prior density and distribution plots. Offences against the person and motoring lead in conviction counts. This situation reveals that the number of robbery convictions in this region is higher than national customs.

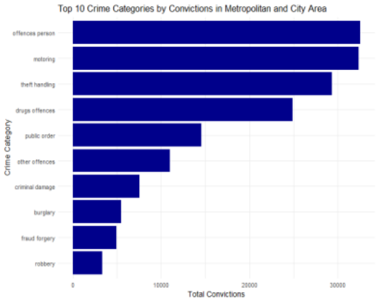
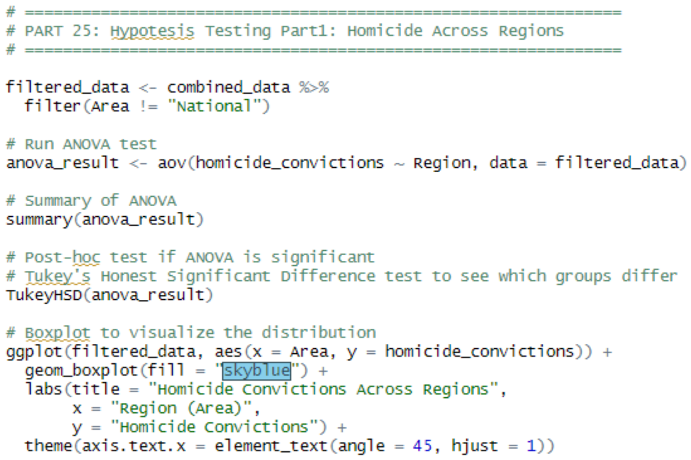


Figure 3.17. Top 10 Crime Categories by Convictions in Metropolitan and City Area

4 Hypothesis Testing

4.1 ANNOVA Hypothesis Test with Tukey's HSD Test – Homicide Across Regions

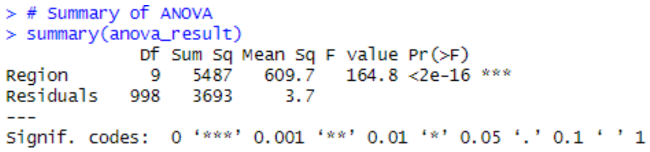
It is critical to understand regional differences in conviction rates. Because the criminal justice system strives for fairness and consistency. To test this, a one-way ANNOVA was conducted to compare mean homicide conviction rates across regions. The null hypothesis (*H_0*) states that there is no significant difference in mean conviction rates across regions, while the alternative hypothesis (*H_1*) assumes that at least one region is significantly different from the others. Figure 4.1 reflects the implementation of the ANNOVA hypothesis process.



```
# ================================================================
# PART 25: Hypotesis Testing Part1: Homicide Across Regions
# ================================================================

filtered_data <- combined_data %>%
  filter(Area != "National")

# Run ANOVA test
anova_result <- aov(homicide_convictions ~ Region, data = filtered_data)

# Summary of ANOVA
summary(anova_result)

# Post-hoc test if ANOVA is significant
# Tukey's Honest Significant Difference test to see which groups differ
TukeyHSD(anova_result)

# Boxplot to visualize the distribution
ggplot(filtered_data, aes(x = Area, y = homicide_convictions)) +
  geom_boxplot(fill = "skyblue") +
  labs(title = "Homicide Convictions Across Regions",
       x = "Region (Area)",
       y = "Homicide Convictions") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Figure 4.1. Implementation of the ANNOVA for Homicide Across Regions

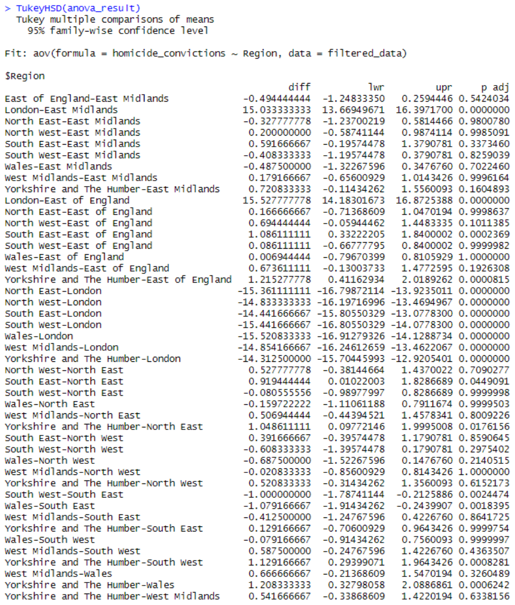
As a result of the ANNOVA analysis, the F value in the ANNOVA table given in Figure 4.2 is a very high value of 164,8. Conversely, the p-value was observed as lower than 2e^{-16}. Since this p-value is well below the typical significance level of 0.01, the *H_0* is rejected. In other words, there are statistically significant differences in terms of average homicide conviction rates between the regions. Thus, the *H_1* is accepted.



```
> # Summary of ANOVA
> summary(anova_result)
             Df Sum Sq Mean Sq F value Pr(>F)    
Region        9   5487   609.7   164.8 <2e-16 ***
Residuals   998   3693     3.7                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 4.2. ANNOVA Test Results of Homicide Across Regions

Based on ANNOVA test results, Tukey's Honest Significant Difference test was applied to detail the hypothesis. This test determined which regions had significant differences. The test results are given in Figure 4.3. Highly significant differences were found between London and all other regions. Moreover, the difference between London and North East was -15.36 and p-value lower than 0.0001. Significant differences were also observed between South East and South West, Wales and South East and Yorkshire and The Humber and some regions. These results showed that some regions had systematically different homicide conviction rates than others.



```
> TukeyHSD(anova_result)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = homicide_convictions ~ Region, data = filtered_data)

$Region
                                                  diff          lwr         upr     p adj
East of England-East Midlands              -0.494444444  -1.24833350   0.2594446 0.5424034
London-East Midlands                       15.033333333  13.66949671  16.3971700 0.0000000
North East-East Midlands                   -0.327777778  -1.23700219   0.5814466 0.9800780
North West-East Midlands                    0.200000000  -0.58741144   0.9874114 0.9985091
South East-East Midlands                    0.591666667  -0.19574478   1.3790781 0.3373460
South West-East Midlands                   -0.408333333  -1.19574478   0.3790781 0.8259039
Wales-East Midlands                        -0.487500000  -1.32267596   0.3476760 0.7022460
West Midlands-East Midlands                 0.179166667  -0.65600929   1.0143426 0.9996164
Yorkshire and The Humber-East Midlands      0.720833333  -0.11434262   1.5560093 0.1604893
London-East of England                     15.527777778  14.18301673  16.8725388 0.0000000
North East-East of England                  0.166666667  -0.71368609   1.0470194 0.9998637
North West-East of England                  0.694444444  -0.05944462   1.4483335 0.1011385
South East-East of England                  1.086111111   0.33222205   1.8400002 0.0002369
South West-East of England                  0.086111111  -0.66777795   0.8400002 0.9999982
Wales-East of England                       0.006944444  -0.79670399   0.8105929 1.0000000
West Midlands-East of England               0.673611111  -0.13003733   1.4772595 0.1926308
Yorkshire and The Humber-East of England    1.215277778   0.41162934   2.0189262 0.0000815
North East-London                         -15.361111111 -16.79872114 -13.9235011 0.0000000
North West-London                         -14.833333333 -16.19716996 -13.4694967 0.0000000
South East-London                         -14.441666667 -15.80550329 -13.0778300 0.0000000
South West-London                         -15.441666667 -16.80550329 -14.0778300 0.0000000
Wales-London                              -15.520833333 -16.91279326 -14.1288734 0.0000000
West Midlands-London                      -14.854166667 -16.24612659 -13.4622067 0.0000000
Yorkshire and The Humber-London           -14.312500000 -15.70445993 -12.9205401 0.0000000
North West-North East                       0.527777778  -0.38144664   1.4370022 0.7090277
South East-North East                       0.919444444   0.01022003   1.8286689 0.0449091
South West-North East                      -0.080555556  -0.98977997   0.8286689 0.9999998
Wales-North East                           -0.159722222  -1.11061188   0.7911674 0.9999503
West Midlands-North East                    0.506944444  -0.44394521   1.4578341 0.8009226
Yorkshire and The Humber-North East         1.048611111   0.09772146   1.9995008 0.0176156
South East-North West                       0.391666667  -0.39574478   1.1790781 0.8590645
South West-North West                      -0.608333333  -1.39574478   0.1790781 0.2975402
Wales-North West                           -0.687500000  -1.52267596   0.1476760 0.2140515
West Midlands-North West                   -0.020833333  -0.85600929   0.8143426 1.0000000
Yorkshire and The Humber-North West         0.520833333  -0.31434262   1.3560093 0.6152173
South West-South East                      -1.000000000  -1.78741144  -0.2125886 0.0024474
Wales-South East                           -1.079166667  -1.91434262  -0.2439907 0.0018395
West Midlands-South East                   -0.412500000  -1.24767596   0.4226760 0.8641725
Yorkshire and The Humber-South East         0.129166667  -0.70600929   0.9643426 0.9999754
Wales-South West                           -0.079166667  -0.91434262   0.7560093 0.9999997
West Midlands-South West                    0.587500000  -0.24767596   1.4226760 0.4363507
Yorkshire and The Humber-South West         1.129166667   0.29399071   1.9643426 0.0008281
West Midlands-Wales                         0.666666667  -0.21368609   1.5470194 0.3260489
Yorkshire and The Humber-Wales              1.208333333   0.32798058   2.0886861 0.0006242
Yorkshire and The Humber-West Midlands      0.541666667  -0.33868609   1.4220194 0.6338156
```

Figure 4.3. Tukey's Honest Significant Difference Test Results of Homicide Across Regions

Figure 4.4 shows the distribution of homicide convictions by region. In this context, the authorities need to consider homicide conviction assessments, especially in the London region, differently from other regions and increase precautions. Measures should be taken by increasing the number of police teams and tightening the sanctions for regions where the regional population density and countryside are less than the urban settlements.

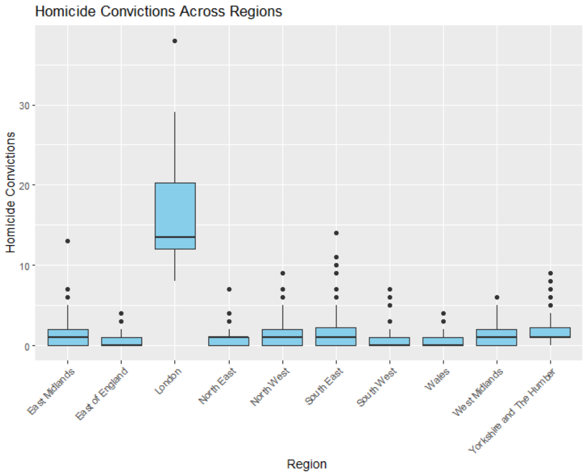
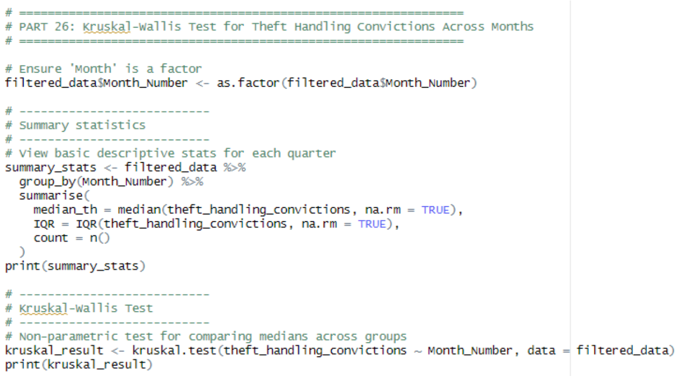


Figure 4.4. Distribution of Homicide Convictions Across Regions

4.2 Kruskal-Wallis Test- Theft Handling Convictions Across Months

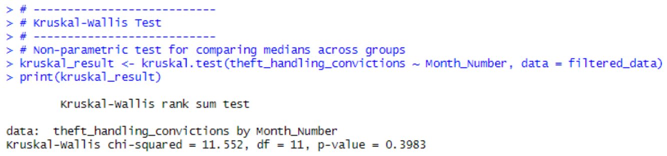
Understanding monthly variation in theft handling convictions is essential for identifying potential temporal trends in criminal activity. Since conviction rates may fluctuate due to seasonal patterns, a Kruskal-Wallis test was performed to examine differences in theft handling convictions across months. The H_0 posits that there is no significant difference in the distribution of convictions among the twelve months, while the H_1 assumes that at least one month differs significantly. Figure 4.5 illustrates the implementation of the Kruskal-Wallis hypothesis testing.



```
# ==============================================================
# PART 26: Kruskal-Wallis Test for Theft Handling Convictions Across Months
# ==============================================================

# Ensure 'Month' is a factor
filtered_data$Month_Number <- as.factor(filtered_data$Month_Number)

# ---------------------------
# Summary statistics
# ---------------------------
# View basic descriptive stats for each quarter
summary_stats <- filtered_data %>%
  group_by(Month_Number) %>%
  summarise(
    median_th = median(theft_handling_convictions, na.rm = TRUE),
    IQR = IQR(theft_handling_convictions, na.rm = TRUE),
    count = n()
  )
print(summary_stats)

# ---------------------------
# Kruskal-Wallis Test
# ---------------------------
# Non-parametric test for comparing medians across groups
kruskal_result <- kruskal.test(theft_handling_convictions ~ Month_Number, data = filtered_data)
print(kruskal_result)
```

Figure 4.5. Implementation of Kruskal-Wallis Hypothesis Testing

As seen in Figure 4.6, the test yielded a chi-squared value of 11,552 and a p-value of 0,398. This showed no statistically significant variation across months in terms of theft handling convictions. Since this p-value is above the typical significance level, the H_0 is accepted. Conversely, the H_1 is rejected.



```
> # ---------------------------
> # Kruskal-Wallis Test
> # ---------------------------
> # Non-parametric test for comparing medians across groups
> kruskal_result <- kruskal.test(theft_handling_convictions ~ Month_Number, data = filtered_data)
> print(kruskal_result)

	Kruskal-Wallis rank sum test

data:  theft_handling_convictions by Month_Number
Kruskal-Wallis chi-squared = 11.552, df = 11, p-value = 0.3983
```

Figure 4.6. Kruskal-Wallis Test Results

In addition, Figure 4.7 illustrates the distribution of theft handling convictions across months. This shows that monthly fluctuations in convictions are likely due to random variation rather than a systematic or seasonal effect. Besides, this means that theft handling convictions remain relatively constant throughout the year. Therefore, resource allocation or policy adjustments aimed at addressing theft handling monthly may not be necessary. Instead, decision makers must consider focusing long-term strategies rather than implementing month-by-month interventions. This finding highlights the importance of prioritizing data-driven approaches in law enforcement planning.

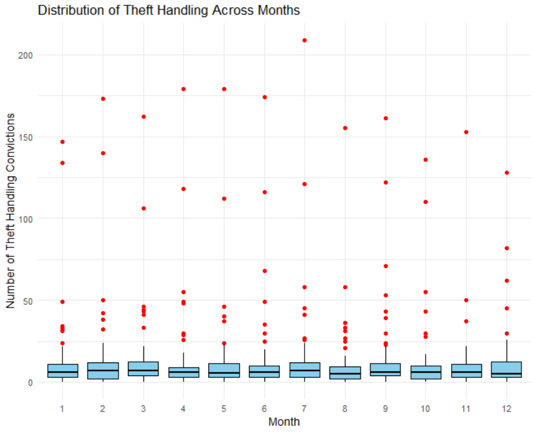
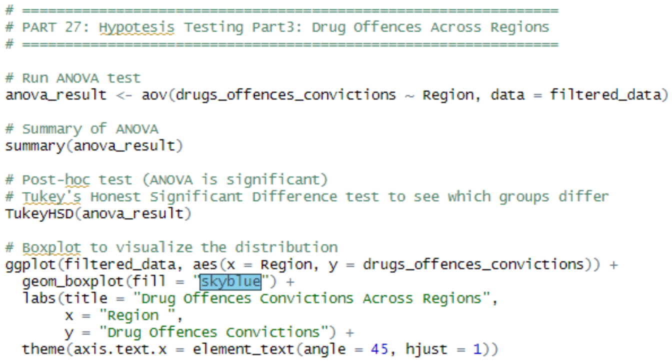


Figure 4.7. Distribution of Theft Handling Convictions Across Months

4.3 ANNOVA Hypothesis Test with Tukey's HSD Test – Drug Offences Across Regions

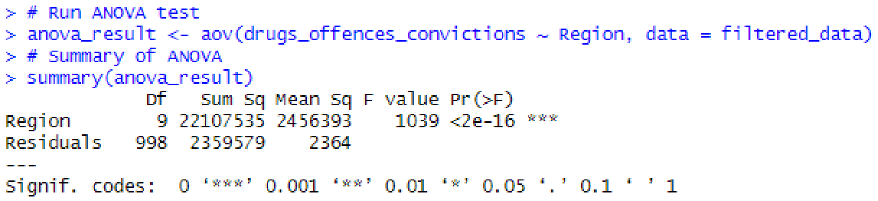
Drug offences convictions can be examined regionally because individuals’ access to drugs and the frequency of drug use may differ across regions. ANNOVA test was performed to test this hypothesis. The *H_0* states that there is no significant difference in the average conviction rates across regions, while the *H_1* assumes that at least one region is significantly different from the others. Figure 4.8 reflects the application of the ANNOVA hypothesis process.



```
# ==============================================================
# PART 27: Hypotesis Testing Part3: Drug Offences Across Regions
# ==============================================================

# Run ANOVA test
anova_result <- aov(drugs_offences_convictions ~ Region, data = filtered_data)

# Summary of ANOVA
summary(anova_result)

# Post-hoc test (ANOVA is significant)
# Tukey's Honest Significant Difference test to see which groups differ
TukeyHSD(anova_result)

# Boxplot to visualize the distribution
ggplot(filtered_data, aes(x = Region, y = drugs_offences_convictions)) +
  geom_boxplot(fill = "skyblue") +
  labs(title = "Drug Offences Convictions Across Regions",
       x = "Region ",
       y = "Drug Offences Convictions") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Figure 4.8. Implementation of the ANNOVA for Drug Offences Across Regions

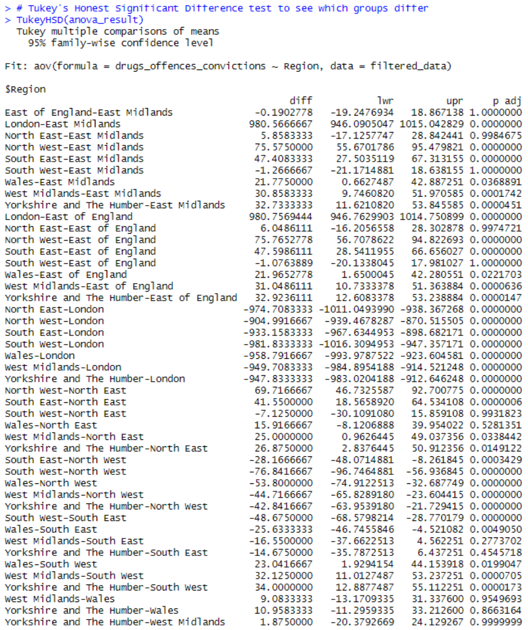
According to ANNOVA analysis, it was tested whether the average drug offences conviction rates were different between the regions. The F value in the ANNOVA table given in Figure 4.9 is a very high value of 1039. On the other hand, the p-value was observed lower then 2e^{-16}. Since this p-value is well below the typical significance level of 0.01, the *H_0* is rejected. In other words, there are statistically significant differences in terms of average drug offences conviction rates between the regions. Conversely, the *H_1* is accepted.



```
> # Run ANOVA test
> anova_result <- aov(drugs_offences_convictions ~ Region, data = filtered_data)
> # Summary of ANOVA
> summary(anova_result)
             Df   Sum Sq Mean Sq F value Pr(>F)    
Region        9 22107535 2456393    1039 <2e-16 ***
Residuals   998  2359579    2364                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 4.9. ANNOVA Test Results of Drug Offences Across Regions

Based on the ANOVA test results, a Tukey's HSD test was conducted to identify which regions had statistically significant differences in drug offence conviction rates. As shown in Figure 4.10, the results indicated regional differences. Notably, London demonstrated highly significant differences compared to all other regions, with mean differences ranging from approximately -904.99 (North West) to -981.83 (South West). In addition, North West region had significantly higher conviction rates than the North East. Besides, the South West had lower rates than the South East. Furthermore, Wales differed significantly from the East of England, and Yorkshire and The Humber showed consistent differences with regions such as the East Midlands and the South West.



```
> # Tukey's Honest Significant Difference test to see which groups differ
> TukeyHSD(anova_result)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = drugs_offences_convictions ~ Region, data = filtered_data)

$Region
                                                 diff           lwr         upr     p adj
East of England-East Midlands              -0.1902778   -19.2476934   18.867138 1.0000000
London-East Midlands                      980.5666667   946.0905047 1015.042829 0.0000000
North East-East Midlands                    5.8583333   -17.1257747   28.842441 0.9984675
North West-East Midlands                   75.5750000    55.6701786   95.479821 0.0000000
South East-East Midlands                   47.4083333    27.5035119   67.313155 0.0000000
South West-East Midlands                   -1.2666667   -21.1714881   18.638155 1.0000000
Wales-East Midlands                        21.7750000     0.6627487   42.887251 0.0368891
West Midlands-East Midlands                30.8583333     9.7460820   51.970585 0.0001742
Yorkshire and The Humber-East Midlands     32.7333333    11.6210820   53.845585 0.0000451
London-East of England                    980.7569444   946.7629903 1014.750899 0.0000000
North East-East of England                  6.0486111   -16.2056558   28.302878 0.9974721
North West-East of England                 75.7652778    56.7078622   94.822693 0.0000000
South East-East of England                 47.5986111    28.5411955   66.656027 0.0000000
South West-East of England                 -1.0763889   -20.1338045   17.981027 1.0000000
Wales-East of England                      21.9652778     1.6500045   42.280551 0.0221703
West Midlands-East of England              31.0486111    10.7333378   51.363884 0.0000636
Yorkshire and The Humber-East of England   32.9236111    12.6083378   53.238884 0.0000147
North East-London                        -974.7083333 -1011.0493990 -938.367268 0.0000000
North West-London                        -904.9916667  -939.4678287 -870.515505 0.0000000
South East-London                        -933.1583333  -967.6344953 -898.682171 0.0000000
South West-London                        -981.8333333 -1016.3094953 -947.357171 0.0000000
Wales-London                             -958.7916667  -993.9787522 -923.604581 0.0000000
West Midlands-London                     -949.7083333  -984.8954188 -914.521248 0.0000000
Yorkshire and The Humber-London          -947.8333333  -983.0204188 -912.646248 0.0000000
North West-North East                      69.7166667    46.7325587   92.700775 0.0000000
South East-North East                      41.5500000    18.5658920   64.534108 0.0000006
South West-North East                      -7.1250000   -30.1091080   15.859108 0.9931823
Wales-North East                           15.9166667    -8.1206888   39.954022 0.5281351
West Midlands-North East                   25.0000000     0.9626445   49.037356 0.0338442
Yorkshire and The Humber-North East        26.8750000     2.8376445   50.912356 0.0149122
South East-North West                     -28.1666667   -48.0714881   -8.261845 0.0003429
South West-North West                     -76.8416667   -96.7464881  -56.936845 0.0000000
Wales-North West                          -53.8000000   -74.9122513  -32.687749 0.0000000
West Midlands-North West                  -44.7166667   -65.8289180  -23.604415 0.0000000
Yorkshire and The Humber-North West       -42.8416667   -63.9539180  -21.729415 0.0000000
South West-South East                     -48.6750000   -68.5798214  -28.770179 0.0000000
Wales-South East                          -25.6333333   -46.7455846   -4.521082 0.0049050
West Midlands-South East                  -16.5500000   -37.6622513    4.562251 0.2773702
Yorkshire and The Humber-South East       -14.6750000   -35.7872513    6.437251 0.4545718
Wales-South West                           23.0416667     1.9294154   44.153918 0.0199047
West Midlands-South West                   32.1250000    11.0127487   53.237251 0.0000705
Yorkshire and The Humber-South West        34.0000000    12.8877487   55.112251 0.0000173
West Midlands-Wales                         9.0833333   -13.1709335   31.337600 0.9549693
Yorkshire and The Humber-Wales             10.9583333   -11.2959335   33.212600 0.8663164
Yorkshire and The Humber-West Midlands      1.8750000   -20.3792669   24.129267 0.9999999
```

Figure 4.10. Tukey's HSD Test Results of Drug Offences Across Regions

Figure 4.11 indicates the distribution of drug offences convictions by region. The ANOVA and Tukey HSD results provide strong evidence of significant regional differences in drug convictions. London consistently shows significantly higher convictions than all other regions. This is due to its high population density and easy access to drugs. Therefore, policy makers and criminal justice authorities should address the problem of drug offences regionally and try to solve the problem. This problem, especially in crowded areas, requires meticulous inspections and dedicated police teams.

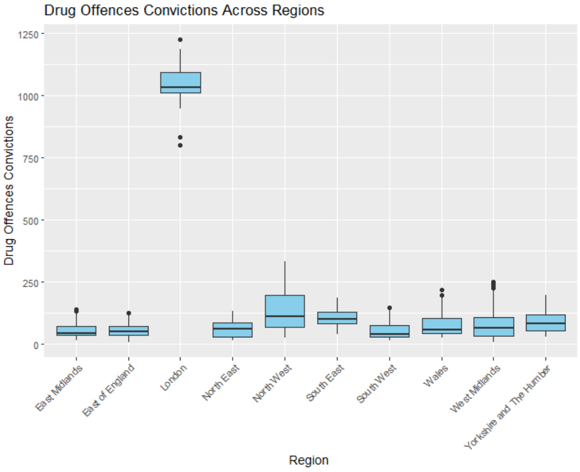


Figure 4.11. Distribution of Drug Offences Across Regions

5 Predictive Modelling

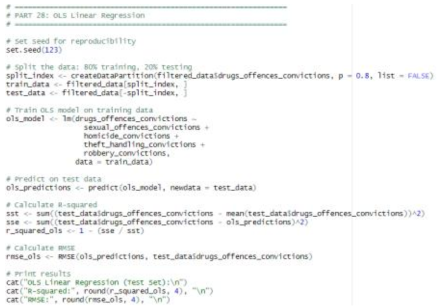
In the predictive modelling section, and 9 different models were created for regression clustering and classification tasks. The aim of this section was to compare the hypotheses and the performance of the models and create statistical interpretations.

5.1 Regression Analysis

Understanding the predictive power of different regression models in crime datasets is crucial for accurate prediction and decision making. This study develops and evaluates three regression models to determine their effectiveness in predicting drug crime convictions based on convictions. Moreover, performance metrics such as R^2 and Root Mean Square Error (RMSE) are used to evaluate the predictive capability of each model. In this section, a 20% test dataset is used to evaluate the performance of the models on data that they have not seen.

5.1.1 Ordinary Least Squares (OLS) Linear Regression

The H_0 assumes that the use of homicide, robbery, burglary, and sexual offense convictions to explain drug offense convictions is not explanatory, while the H_1 assumes that at least one variable is significantly associated. Figure 5.1 indicates the OLS model implementation process, including setting seed constant for reproducibility, split process of train-test data and model creation.



```
# ------------------------------------------------------------------
# PART 28: OLS Linear Regression
# ------------------------------------------------------------------

# Set seed for reproducibility
set.seed(123)

# Split the data: 80% training, 20% testing
split_index <- createDataPartition(filtered_data$drugs_offences_convictions, p = 0.8, list = FALSE)
train_data <- filtered_data[split_index, ]
test_data <- filtered_data[-split_index, ]

# Train OLS model on training data
ols_model <- lm(drugs_offences_convictions ~
                  sexual_offences_convictions +
                  homicide_convictions +
                  theft_handling_convictions +
                  robbery_convictions,
                data = train_data)

# Predict on test data
ols_predictions <- predict(ols_model, newdata = test_data)

# Calculate R-squared
sst <- sum((test_data$drugs_offences_convictions - mean(test_data$drugs_offences_convictions))^2)
sse <- sum((test_data$drugs_offences_convictions - ols_predictions)^2)
r_squared_ols <- 1 - (sse / sst)

# Calculate RMSE
rmse_ols <- RMSE(ols_predictions, test_data$drugs_offences_convictions)

# Print results
cat("OLS Linear Regression (Test Set):\n")
cat("R-squared:", round(r_squared_ols, 4), "\n")
cat("RMSE:", round(rmse_ols, 4), "\n")
```

Figure 5.1. OLS Model Implementation

As a result of the OLS model, the R^2 value in the test data was found as 0.845 and the RMSE value was 44.3. This shows that the model can explain 84.5% of the variability on the target variable.

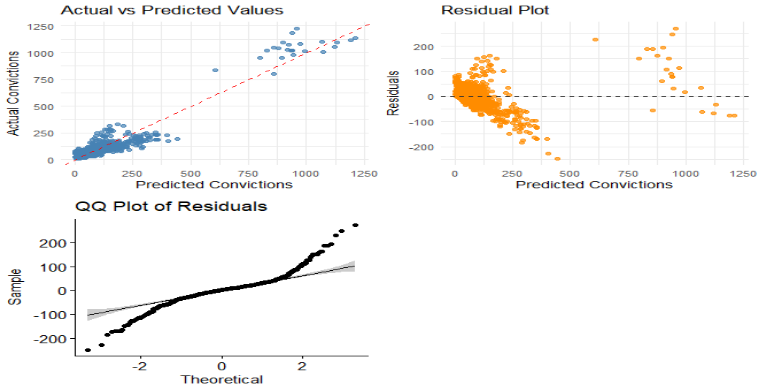


Figure 5.2. OLS Model Performance Visualizations

Figure 5.2 illustrates the performance diagnostics for the OLS regression model. The top-left plot compares actual drug offense convictions against the predicted values. A strong clustering of data points along the reference line indicates a generally good fit, suggesting the model captures the trend. However, some deviation is observed at higher conviction values, where the model tends to underpredict the actual outcomes. The top-right residual plot visualizes the residuals against the predicted values. Furthermore, curved pattern is evident, and this suggests that prediction errors are not equally distributed across all prediction levels. Lastly, the bottom-left Q-Q plot of residuals reveals that the residuals deviate from the diagonal line this indicates that the residuals are not perfectly normally distributed. In summary, OLS model demonstrates reasonable predictive performance for lower to mid-range values.

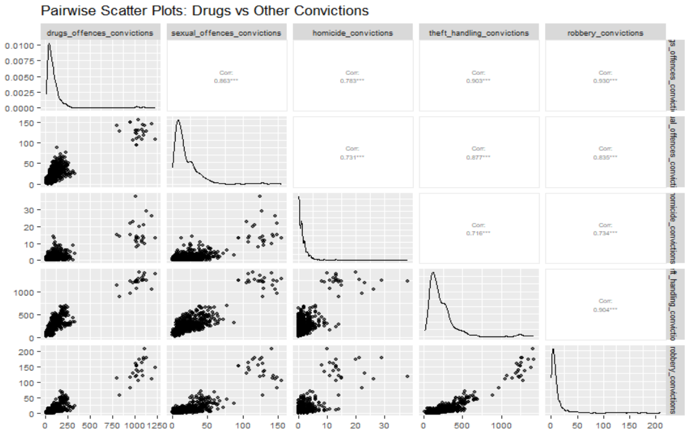


Figure 5.3. Pairwise Scatter Plots: Drugs vs Other Convictions

Figure 5.3 presents pairwise scatter plots and correlation coefficients between drug offense convictions and other conviction types. Strong positive correlations are observed, especially with *theft_handling*, suggesting these variables are effective predictors. The density plots indicate skewed distributions, while the scatter plots confirm linear associations.

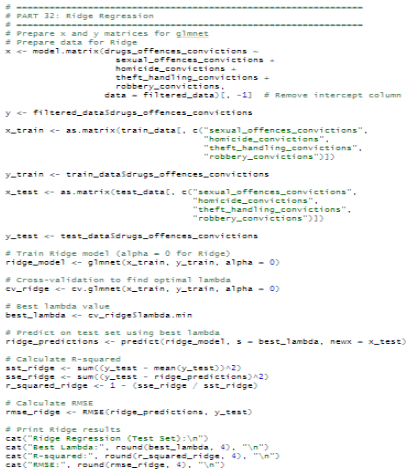
Analysis Type	R^2	Adjusted R^2	Estimated Coefficient	Std. Error	p_value
Homicide vs Drug	0.6127	0.6123	0.0152	0.0004	<0.001
Robbery vs Drug	0.8651	0.8650	0.1351	0.0017	<0.001
Theft Handling vs Drug	0.8147	0.8145	11.493	0.0173	<0.001
Sexual Offences vs Drug	0.7447	0.7445	0.1188	0.0022	<0.001

Table 5.1. OLS Model Summary for Different Scenarios

Table 5.1 summarizes the results of simple linear regression models assessing the predictive power of individual crime categories on drug offense convictions. All sub-models exhibit strong explainability. For example, robbery showed the highest R^2 value, indicating it explains approximately 86.5% of the variance in drug convictions. Besides, theft handling and sexual offenses also show strong explanatory power (R^2 = 0.8147 and 0.7447, respectively). These findings validate the strong linear relationship between certain crime types and drug-related offenses. Additionally, the p-value was observed as lower than 0,001. Thus, the H_0 is rejected. In other words, chosen variables are statistically significant to explain drug offences convictions. Conversely, the H_1 is accepted.

5.1.2 Ridge Regression

Second regression model was Ridge model. Using same test data model implemented as shown in Figure 5.4. As a result of the Ridge model, the R^2 value in the test data was found as 0,827 and the RMSE value was 44,7. This shows that the model can explain 84,3% of the variability on the drug offences convictions.



```r
# ==================================================================
# PART 32: Ridge Regression
# ==================================================================
# Prepare x and y matrices for glmnet
# Prepare data for Ridge
x <- model.matrix(drugs_offences_convictions ~
                    sexual_offences_convictions +
                    homicide_convictions +
                    theft_handling_convictions +
                    robbery_convictions,
                  data = filtered_data)[, -1]  # Remove intercept column

y <- filtered_data$drugs_offences_convictions

x_train <- as.matrix(train_data[, c("sexual_offences_convictions",
                                    "homicide_convictions",
                                    "theft_handling_convictions",
                                    "robbery_convictions")])

y_train <- train_data$drugs_offences_convictions

x_test <- as.matrix(test_data[, c("sexual_offences_convictions",
                                  "homicide_convictions",
                                  "theft_handling_convictions",
                                  "robbery_convictions")])

y_test <- test_data$drugs_offences_convictions

# Train Ridge model (alpha = 0 for Ridge)
ridge_model <- glmnet(x_train, y_train, alpha = 0)

# Cross-validation to find optimal lambda
cv_ridge <- cv.glmnet(x_train, y_train, alpha = 0)

# Best lambda value
best_lambda <- cv_ridge$lambda.min

# Predict on test set using best lambda
ridge_predictions <- predict(ridge_model, s = best_lambda, newx = x_test)

# Calculate R-squared
sst_ridge <- sum((y_test - mean(y_test))^2)
sse_ridge <- sum((y_test - ridge_predictions)^2)
r_squared_ridge <- 1 - (sse_ridge / sst_ridge)

# Calculate RMSE
rmse_ridge <- RMSE(ridge_predictions, y_test)

# Print Ridge results
cat("Ridge Regression (Test Set):\n")
cat("Best Lambda:", round(best_lambda, 4), "\n")
cat("R-squared:", round(r_squared_ridge, 4), "\n")
cat("RMSE:", round(rmse_ridge, 4), "\n")
```

Figure 5.4. Ridge Model Implementation

Figure 5.5 illustrates the performance diagnostics for the Ridge regression model. According the first plot, it is seen that model captures the underlying trend. However, some deviation is observed at higher conviction values, where the model tends to underpredict the actual outcomes. In second plot, curved pattern is evident, and this suggests that prediction errors are not equally distributed across all prediction levels. Lastly, the Q-Q plot of residuals reveals that the residuals deviate from the diagonal line this indicates that the residuals are not perfectly normally distributed. In summary, Ridge model demonstrates reasonable predictive performance for lower to mid-range values.

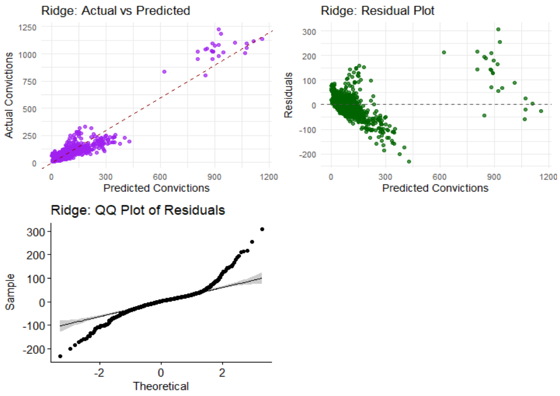


Figure 5.5. Ridge Model Performance Visualizations

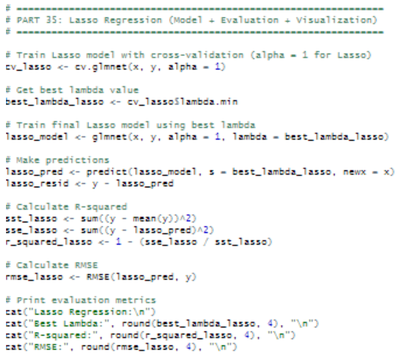
Table 5.2 presents the estimated coefficients from the Ridge Regression model. The homicide convictions variable has the largest positive coefficient (9.34). This means that this variable has strong influence on predicting drug-related convictions. Robbery convictions also maintain a notable effect, while sexual offenses show a moderate impact.

Variable	Coefficient
sexual_offences_convictions	1,40
homicide_convictions	9,34
theft_handling_convictions	0,19
robbery_convictions	2,72

Table 5.2. Estimated Ridge Model Parameters

5.1.3 Lasso Regression

Third regression model was Lasso model. Using same test data model implemented as shown in Figure 5.6. As a result, the R^2 value in the test data was found as 0,905 and the RMSE value was 47,9. This shows that the model can explain 90,5% of the variability on the drugs offences convictions.



```
# ==============================================================
# PART 35: Lasso Regression (Model + Evaluation + Visualization)
# ==============================================================

# Train Lasso model with cross-validation (alpha = 1 for Lasso)
cv_lasso <- cv.glmnet(x, y, alpha = 1)

# Get best lambda value
best_lambda_lasso <- cv_lasso$lambda.min

# Train final Lasso model using best lambda
lasso_model <- glmnet(x, y, alpha = 1, lambda = best_lambda_lasso)

# Make predictions
lasso_pred <- predict(lasso_model, s = best_lambda_lasso, newx = x)
lasso_resid <- y - lasso_pred

# Calculate R-squared
sst_lasso <- sum((y - mean(y))^2)
sse_lasso <- sum((y - lasso_pred)^2)
r_squared_lasso <- 1 - (sse_lasso / sst_lasso)

# Calculate RMSE
rmse_lasso <- RMSE(lasso_pred, y)

# Print evaluation metrics
cat("Lasso Regression:\n")
cat("Best Lambda:", round(best_lambda_lasso, 4), "\n")
cat("R-squared:", round(r_squared_lasso, 4), "\n")
cat("RMSE:", round(rmse_lasso, 4), "\n")
```

Figure 5.6. Lasso Model Implementation

Figure 5.7 illustrates the performance plots of Lasso model. In first plot, it is seen that model captures the underlying trend. However, some deviation is observed at higher conviction values, where the model tends to underpredict the actual outcomes. In second plot, curved pattern is evident, and this suggests that prediction errors are not equally distributed across all prediction levels. Finally, the Q-Q plot of residuals reveals that the residuals deviate from the diagonal line this indicates that the residuals are not perfectly normally distributed.

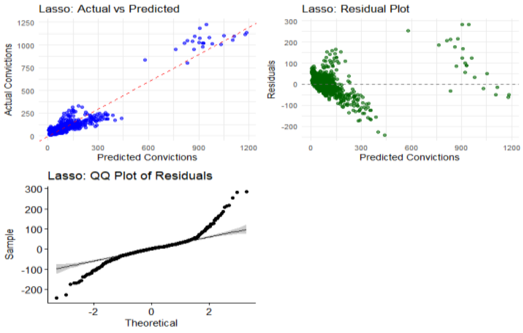


Figure 5.7. Lasso Model Performance Visualizations

Table 5.3 presents the estimated coefficients from the Lasso model. The homicide convictions variable has the largest positive coefficient (7.96), indicating a strong influence on predicting drug-related convictions. Robbery convictions also maintain a notable effect. Briefly, when predicting drug offense convictions, it is quite instructive to analyse the homicide convictions.

Variable	Coefficient
sexual_offences_convictions	1,05
homicide_convictions	7,96
theft_handling_convictions	0,15
robbery_convictions	3,55

Table 5.3. Estimated Lasso Model Parameters

5.1.4 Model Comparison

Figure 5.8 illustrates a comparison of applied models using R-squared and RMSE metrics. In Panel A, Lasso regression achieves the highest R-squared value (0.905). Thus, Lasso explains the greatest proportion of variance in drug offense convictions among the models. Besides, OLS and Ridge models show similar and slightly lower explanatory power (0.845 and 0.843, respectively). However, Panel B reveals that OLS yields the lowest RMSE. This means that, it has the most accurate predictions in terms of error magnitude. These results suggest that while Lasso predicts better, OLS delivers the best fit on the test data in terms of prediction accuracy, reflecting a trade-off between bias and variance. To sum up, a combined model will be more successful for real life application.

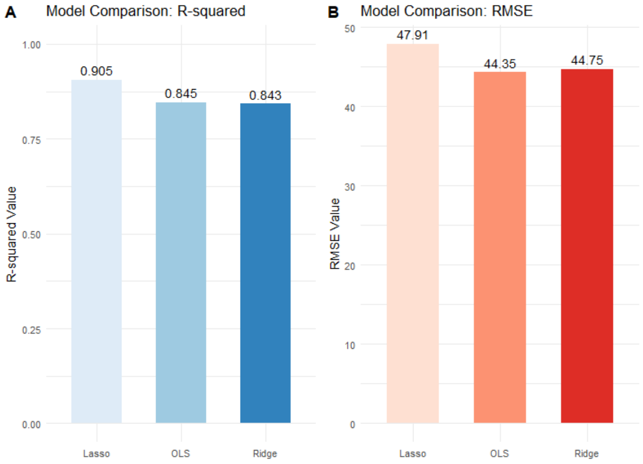


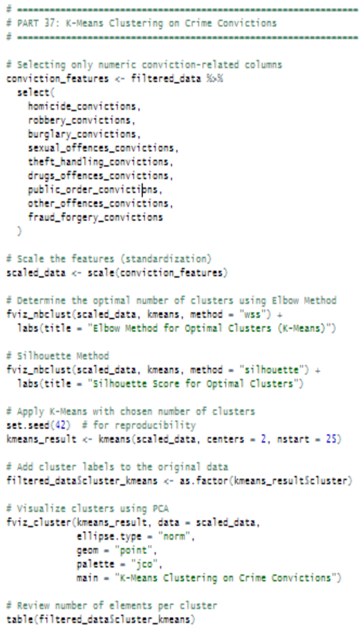
Figure 5.8. Comparison of Regression Models

5.2 Clustering

In this study, K-Means, Hierarchical and DBSCAN clustering methods were applied. The aim of this section was to examine the data in more meaningful clusters using the convictions numbers. Only convictions numbers were selected as clustering data to satisfy methods' requirements, and the same data was used for all algorithms.

5.2.1 K-Means Clustering

The elbow method was used to determine the optimum cluster number for the K-Means algorithm and the clusters were visualized by reducing the results to 2 dimensions with PCA. Figure 5.9 indicates the implementation of K-Means clustering.



```r
# ===================================================================
# PART 37: K-Means Clustering on Crime Convictions
# ===================================================================

# Selecting only numeric conviction-related columns
conviction_features <- filtered_data %>%
  select(
    homicide_convictions,
    robbery_convictions,
    burglary_convictions,
    sexual_offences_convictions,
    theft_handling_convictions,
    drugs_offences_convictions,
    public_order_convictions,
    other_offences_convictions,
    fraud_forgery_convictions
  )

# Scale the features (standardization)
scaled_data <- scale(conviction_features)

# Determine the optimal number of clusters using Elbow Method
fviz_nbclust(scaled_data, kmeans, method = "wss") +
  labs(title = "Elbow Method for Optimal Clusters (K-Means)")

# Silhouette Method
fviz_nbclust(scaled_data, kmeans, method = "silhouette") +
  labs(title = "Silhouette Score for Optimal Clusters")

# Apply K-Means with chosen number of clusters
set.seed(42)  # for reproducibility
kmeans_result <- kmeans(scaled_data, centers = 2, nstart = 25)

# Add cluster labels to the original data
filtered_data$cluster_kmeans <- as.factor(kmeans_result$cluster)

# Visualize clusters using PCA
fviz_cluster(kmeans_result, data = scaled_data,
             ellipse.type = "norm",
             geom = "point",
             palette = "jco",
             main = "K-Means Clustering on Crime Convictions")

# Review number of elements per cluster
table(filtered_data$cluster_kmeans)
```

Figure 5.9. Implementation of K-Means Clustering

Based on the elbow method illustrated in Figure 5.10, the optimal number of clusters appears to be 2. At this point, there is a significant reduction in the within-cluster inertia, and the rate of decrease slows down beyond this value. Therefore, selecting two clusters is likely to provide meaningful separation without overfitting the data.

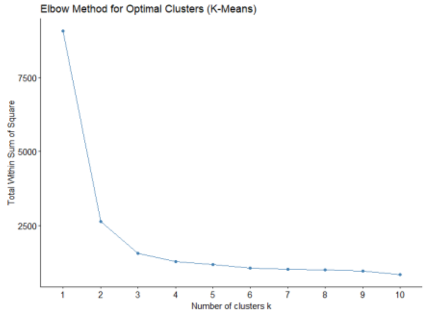


Figure 5.10. Elbow Method for Optimal Clusters

In this study, the Silhouette Score was used to evaluate clustering performance across different values of *k*. As shown in Figure 5.11, the score drops significantly after *k* equal to 2. This suggests that a two-cluster solution maintains good cluster separation while improving stability with achieving 0,85 silhouette score.

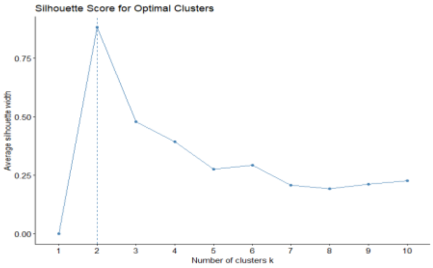


Figure 5.11. Silhouette Score for Optimal Clusters

Figure 5.12 illustrates the results of K-Means clustering applied to crime conviction data, visualized using PCA. The two-dimensional plot captures 91.7% of the total variance, with Dimension 1 explaining 87.2%. Data points are grouped into two distinct clusters, shown by different colours and confidence ellipses. Cluster 1 contains a denser distribution, while Cluster 2 is more dispersed along Dimension 1. This indicates significant underlying differences in conviction-related metrics.

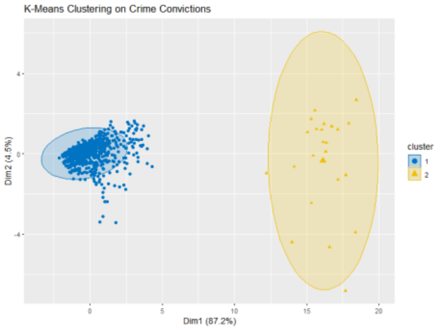


Figure 5.12. K-Means Cluster Visualization using PCA

Figure 5.13 presents the distribution of various crime conviction types across the two K-Means clusters. Boxplots display clear differences, with Cluster 2 consistently associated with higher conviction counts in all offence categories. Cluster 1 exhibits lower and more variable distributions. Moreover, this reinforces the clustering outcome observed in Figure 5.12, indicating that the dataset naturally divides into two distinct groups based on conviction volumes. The figure highlights how specific offence categories contribute to the separation between clusters, particularly for high-volume crimes such as drugs and theft-related offences.

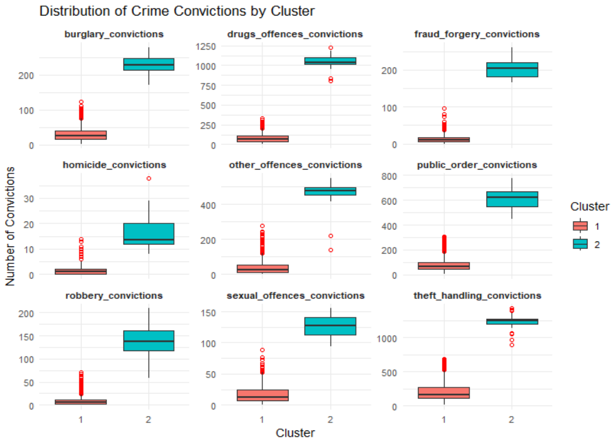
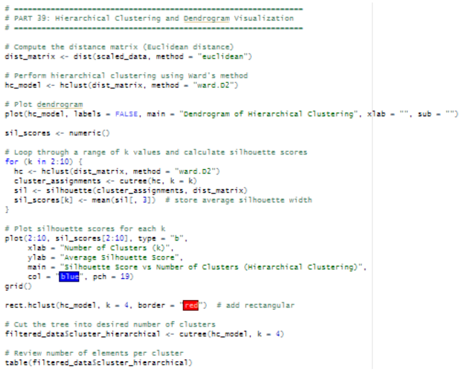


Figure 5.13. Distribution of Crime Convictions by K-Means Cluster

5.2.2 Hierarchical Clustering

The dendrogram plot was used to determine the optimum cluster number for the Hierarchical algorithm and the clusters were visualized with using PCA. Figure 5.14 indicates the implementation of Hierarchical Clustering.



```
# ---------------------------------------------------------------
# PART 39: Hierarchical Clustering and Dendrogram Visualization
# ---------------------------------------------------------------

# Compute the distance matrix (Euclidean distance)
dist_matrix <- dist(scaled_data, method = "euclidean")

# Perform hierarchical clustering using Ward's method
hc_model <- hclust(dist_matrix, method = "ward.D2")

# Plot dendrogram
plot(hc_model, labels = FALSE, main = "Dendrogram of Hierarchical Clustering", xlab = "", sub = "")

sil_scores <- numeric()

# Loop through a range of k values and calculate silhouette scores
for (k in 2:10) {
  hc <- hclust(dist_matrix, method = "ward.D2")
  cluster_assignments <- cutree(hc, k = k)
  sil <- silhouette(cluster_assignments, dist_matrix)
  sil_scores[k] <- mean(sil[, 3])  # store average silhouette width
}

# Plot silhouette scores for each k
plot(2:10, sil_scores[2:10], type = "b",
     xlab = "Number of Clusters (k)",
     ylab = "Average Silhouette Score",
     main = "Silhouette Score vs Number of Clusters (Hierarchical Clustering)",
     col = "blue", pch = 19)
grid()

rect.hclust(hc_model, k = 4, border = "red")  # add rectangular

# Cut the tree into desired number of clusters
filtered_data$cluster_hierarchical <- cutree(hc_model, k = 4)

# Review number of elements per cluster
table(filtered_data$cluster_hierarchical)
```

Figure 5.14. Implementation of Hierarchical Clustering

Figure 5.15 illustrates the hierarchical clustering dendrogram. This plot visually represents the nested grouping of data points according to their pairwise distances. In the figure, vertical lines indicate the distance at which clusters merge, and longer vertical lines reflect greater dissimilarity. Observe the largest vertical distances that are not crossed by horizontal lines, and a natural division into three main clusters is determined. This visual gap, also known as the *"distance threshold"* supports the selection of the optimal number of clusters without requiring prior assumptions. So, the optimal number of clusters was determined as four.

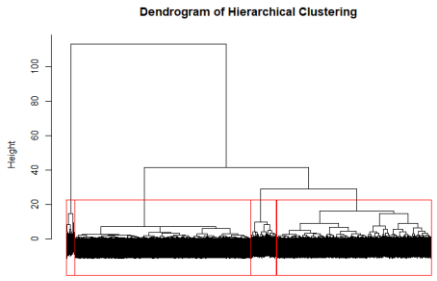


Figure 5.15. Dendrogram of Hierarchical Clustering

Figure 5.16 illustrates the Silhouette scores for different cluster numbers. According to the dendrogram, the optimal number of clusters is 4, with a Silhouette Score of 0,35. This suggests that a four-cluster solution maintains moderate cluster separation.

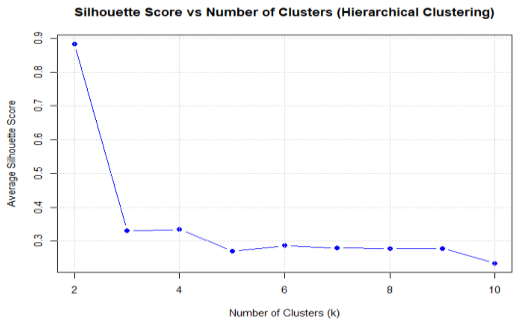


Figure 5.16. Silhouette Score vs Number of Clusters (Hierarchical Clustering)

Figure 5.17 presents a PCA-based visualization of hierarchical clustering results. The plot reduces high-dimensional data into two principal components, revealing distinct groupings. Clusters 1, 2, and 3 appear closely grouped with some overlap, indicating similarity in their feature space. Conversely, Cluster 4 is clearly separated along the first principal component, suggesting it contains substantially different observations. This separation highlights the higher conviction numbers within the dataset.

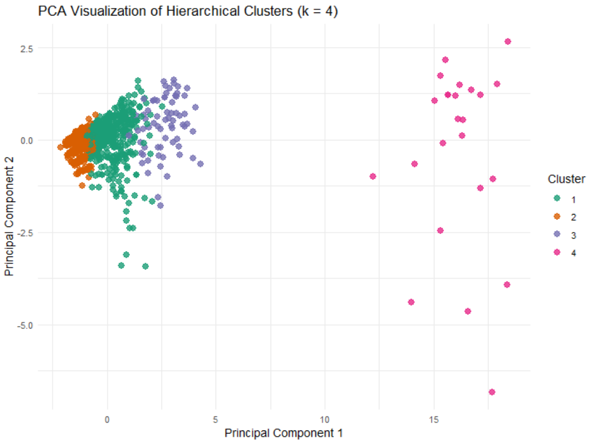


Figure 5.17. Hierarchical Cluster Visualization using PCA

Figure 5.18 reveals the distribution of various crime convictions across four hierarchical clusters. In general, Cluster 4 shows higher conviction counts across all crime categories, suggesting areas in this group experience elevated crime rates. Clusters 1 and 2 generally exhibit lower conviction levels this indicates safer regions. Conversely, Cluster 3 shows moderate conviction levels, especially for theft and sexual offences. The boxplots reveal substantial variability within some clusters, especially Cluster 4, highlighting intra-cluster differences in crime severity. Finally, these patterns suggest meaningful distinctions among clusters in terms of criminal activity and can inform targeted policy strategies.

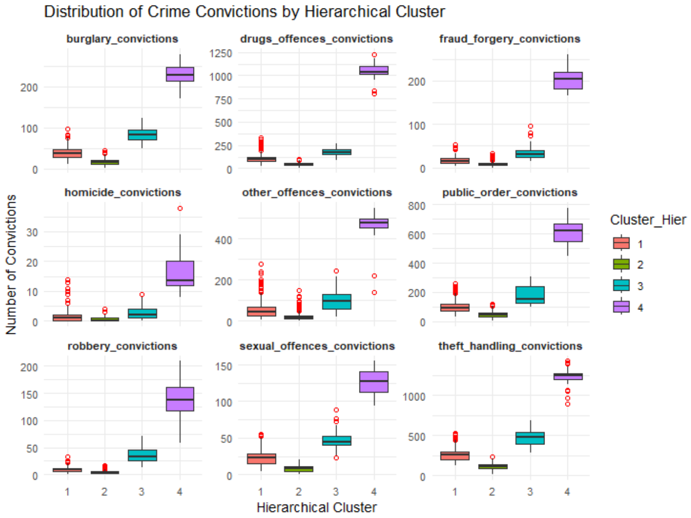
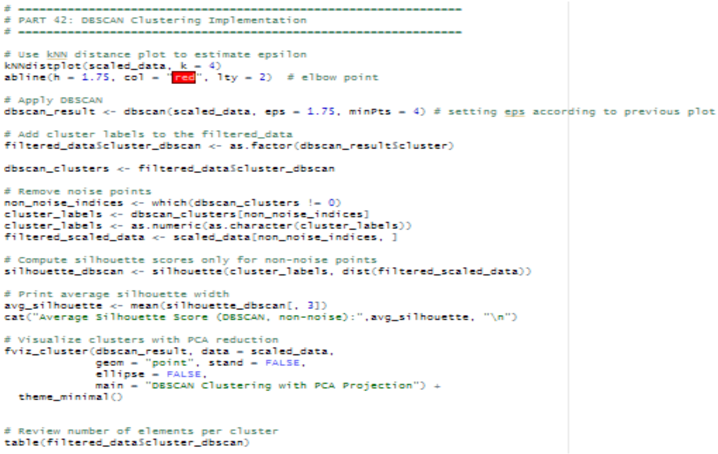


Figure 5.18. Distribution of Crime Convictions by Hierarchical Cluster

5.2.3 DBSCAN Clustering

The k-distance graph method was used to determine the optimal eps value for the DBSCAN algorithm. The clusters were visualized by reducing the results to two dimensions using PCA. Figure 5.19 illustrates the implementation of DBSCAN clustering.



```
# ================================================================
# PART 42: DBSCAN Clustering Implementation
# ================================================================

# Use kNN distance plot to estimate epsilon
kNNdistplot(scaled_data, k = 4)
abline(h = 1.75, col = "red", lty = 2)  # elbow point

# Apply DBSCAN
dbscan_result <- dbscan(scaled_data, eps = 1.75, minPts = 4) # setting eps according to previous plot

# Add cluster labels to the filtered_data
filtered_data$cluster_dbscan <- as.factor(dbscan_result$cluster)

dbscan_clusters <- filtered_data$cluster_dbscan

# Remove noise points
non_noise_indices <- which(dbscan_clusters != 0)
cluster_labels <- dbscan_clusters[non_noise_indices]
cluster_labels <- as.numeric(as.character(cluster_labels))
filtered_scaled_data <- scaled_data[non_noise_indices, ]

# Compute silhouette scores only for non-noise points
silhouette_dbscan <- silhouette(cluster_labels, dist(filtered_scaled_data))

# Print average silhouette width
avg_silhouette <- mean(silhouette_dbscan[, 3])
cat("Average Silhouette Score (DBSCAN, non-noise):",avg_silhouette, "\n")

# Visualize clusters with PCA reduction
fviz_cluster(dbscan_result, data = scaled_data,
             geom = "point", stand = FALSE,
             ellipse = FALSE,
             main = "DBSCAN Clustering with PCA Projection") +
  theme_minimal()


# Review number of elements per cluster
table(filtered_data$cluster_dbscan)
```

Figure 5.19. Implementation of DBSCAN Clustering

Figure 5.20 illustrates the k-distance graph used to determine the optimal eps parameter for DBSCAN clustering. The plot shows the 4-nearest neighbour distances for each data point. A noticeable inflection point appears around a distance value of 1.75. According to this, eps value was selected as 1.75. Detailly, this value balances the inclusion of core points while excluding noise. Besides, the minPts parameter was set to 4, based on common practice and the dimensionality of the data. Using these variables, DBSCAN achieved 0,886 silhouette score excluding data those flagged as noise.

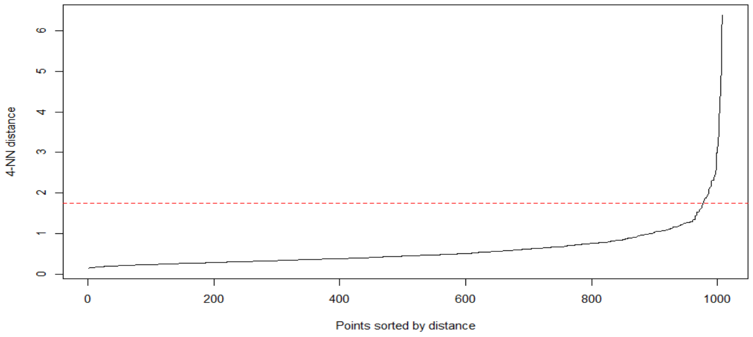


Figure 5.20. Finding optimum eps value

Figure 5.21 illustrates the DBSCAN clustering results visualized using a PCA projection. Two main clusters were identified. In detail, Cluster 1 represents most observations grouped densely, while Cluster 2 is a distinct segment. Besides, the black points flagged as Cluster 0 are labelled by DBSCAN as outlier, as they do not fit the density requirements of any cluster. The PCA axes capture 87.2% and 4.5% of the variance respectively, allowing for a meaningful low-dimensional representation of the clustering structure.

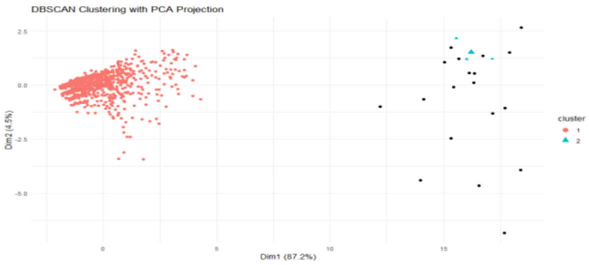


Figure 5.21. DBSCAN Clustering Visualization using PCA

Figure 5.22 displays the distribution of various crime conviction types across DBSCAN-identified clusters. Generally, Cluster 1 shows lower conviction counts across all offence categories, suggesting these records represent less severe or lower-activity regions. Cluster 2 exhibits the highest conviction levels, particularly in drug, theft handling, and sexual offences, indicating high-crime density areas. Cluster 0 represents outliers with extreme values. These clustering results shows decision makers must organize their policies according to convictions levels.

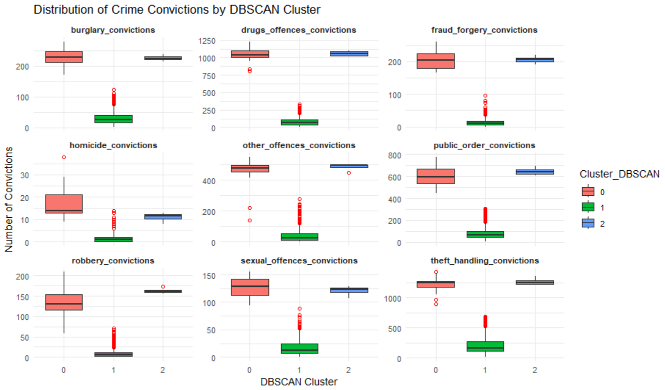


Figure 5.22. Distribution of Crime Convictions by DBSCAN Cluster

5.2.4 Model Comparison

This section compares the performance and interpretability of K-Means, Hierarchical and DBSCAN clustering methods applied to criminal conviction data. Firstly, K-Means achieved a silhouette score of 0,85 with 2 optimal clusters that effectively separated low and high conviction profiles. Although the implementation is simple, this algorithm suffers from the problem of capturing complex shapes or outliers. Furthermore, hierarchical clustering produced 4 clusters and a silhouette score of 0,35. Although its performance was lower, it provided a more detailed view of the data by revealing the varying levels of crime severity across regions. Finally, DBSCAN provided the highest silhouette score of 0,886 and identified 2 main clusters with Cluster 0 as outliers. Unlike other methods, DBSCAN handled non-linear cluster shapes and anomalies more effectively. It provided a more reliable model by addressing outliers within the conviction numbers.

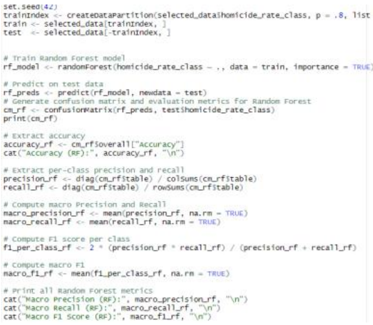
In summary, DBSCAN outperformed other methods in terms of robustness and cluster quality. However, K-Means provided more interpretable segmentation. With DBSCAN method, decision makers can make more goal-oriented and managerial decisions.

5.3 Classification

In this section, Random Forest, Logistic Regression and XGBoost classification models were implemented to predict homicide convictions. Data was created using conviction percentages. 80% of the data was used for training and 20% for testing. Since the homicide convictions column is numeric, records above the median were flagged as 1 and records below as 0 to create a classification target. Also, the seed was set to 42 to ensure reproducibility and the macro f1 score was accepted as the metric considering the structure of the target.

5.3.1 Random Forest

Figure 5.23 indicates the implementation of random forest classification model including calculation of evaluation metrics.



```
set.seed(42)
trainIndex <- createDataPartition(selected_data$homicide_rate_class, p = .8, list = FALSE)
train <- selected_data[trainIndex, ]
test  <- selected_data[-trainIndex, ]


# Train Random Forest model
rf_model <- randomForest(homicide_rate_class ~ ., data = train, importance = TRUE)

# Predict on test data
rf_preds <- predict(rf_model, newdata = test)
# Generate confusion matrix and evaluation metrics for Random Forest
cm_rf <- confusionMatrix(rf_preds, test$homicide_rate_class)
print(cm_rf)

# Extract accuracy
accuracy_rf <- cm_rf$overall["Accuracy"]
cat("Accuracy (RF):", accuracy_rf, "\n")

# Extract per-class precision and recall
precision_rf <- diag(cm_rf$table) / colSums(cm_rf$table)
recall_rf <- diag(cm_rf$table) / rowSums(cm_rf$table)

# Compute macro Precision and Recall
macro_precision_rf <- mean(precision_rf, na.rm = TRUE)
macro_recall_rf <- mean(recall_rf, na.rm = TRUE)

# Compute F1 score per class
f1_per_class_rf <- 2 * (precision_rf * recall_rf) / (precision_rf + recall_rf)

# Compute macro F1
macro_f1_rf <- mean(f1_per_class_rf, na.rm = TRUE)

# Print all Random Forest metrics
cat("Macro Precision (RF):", macro_precision_rf, "\n")
cat("Macro Recall (RF):", macro_recall_rf, "\n")
cat("Macro F1 Score (RF):", macro_f1_rf, "\n")
```

Figure 5.23. Implementation of Random Forest

Figure 5.24 presents the confusion matrix of the Random Forest classifier. The model correctly classified 66 true negatives and 61 true positives. However, it also misclassified 39 instances as false negatives and 34 as false positives. Since the test data consist of 210 records model shows moderate predictive power achieving 0,619 macro f1 score. This can be improved by adding synthetic data.

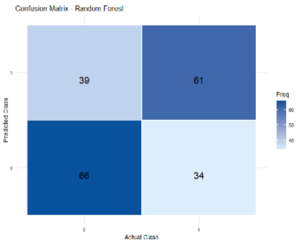


Figure 5.24. Confusion Matrix of Random Forest

Figure 5.25 indicates the feature importance rankings derived from the Random Forest model, measured by mean decrease in gini. Area emerges as the most discriminative variable. This suggest that regional disparities, socioeconomic conditions and law enforcement distribution significantly influence homicide conviction rates. Besides, *total_convictions* and *total_unsuccessful* variables also hold substantial predictive power, reinforcing the connection between broader crime patterns and homicide rates. Conviction percentages for offences against persons and public order crimes exhibit moderate importance. Temporal variables contribute negligibly, supporting the hypothesis that homicide rates are largely independent of seasonal trends.

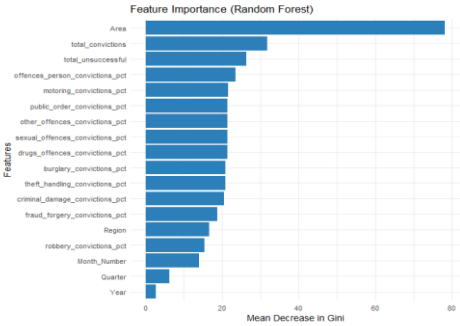
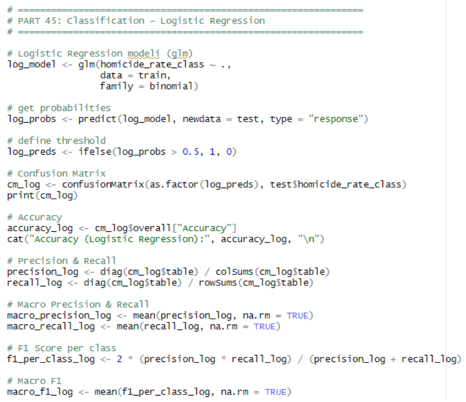


Figure 5.25. Feature Importance of Random Forest

5.3.2 Logistic Regression



```
# ============================================================
# PART 45: Classification - Logistic Regression
# ============================================================

# Logistic Regression modeli (glm)
log_model <- glm(homicide_rate_class ~ .,
                 data = train,
                 family = binomial)

# get probabilities
log_probs <- predict(log_model, newdata = test, type = "response")

# define threshold
log_preds <- ifelse(log_probs > 0.5, 1, 0)

# Confusion Matrix
cm_log <- confusionMatrix(as.factor(log_preds), test$homicide_rate_class)
print(cm_log)

# Accuracy
accuracy_log <- cm_log$overall["Accuracy"]
cat("Accuracy (Logistic Regression):", accuracy_log, "\n")

# Precision & Recall
precision_log <- diag(cm_log$table) / colSums(cm_log$table)
recall_log <- diag(cm_log$table) / rowSums(cm_log$table)

# Macro Precision & Recall
macro_precision_log <- mean(precision_log, na.rm = TRUE)
macro_recall_log <- mean(recall_log, na.rm = TRUE)

# F1 Score per class
f1_per_class_log <- 2 * (precision_log * recall_log) / (precision_log + recall_log)

# Macro F1
macro_f1_log <- mean(f1_per_class_log, na.rm = TRUE)
```

Figure 5.26. Implementation of Logistic Regression

Figure 5.27 presents the confusion matrix of the Logistic Regression classifier. The model correctly classified 63 true negatives and 62 true positives. However, it also misclassified 38 instances as false negatives and 37 as false positives. Additionally, model achieved 0,625 macro f1 score.

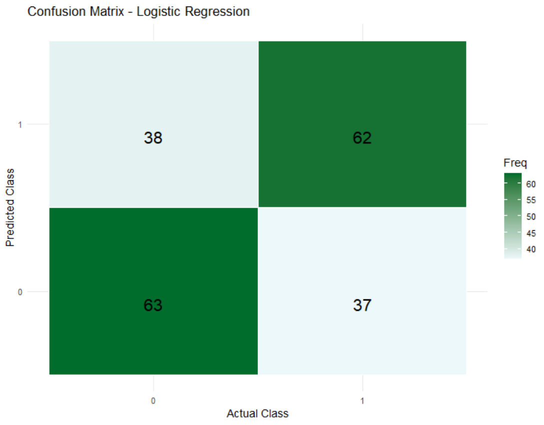


Figure 5.27. Confusion Matrix of Logistic Regression

Figure 5.28 presents the histogram of predicted probabilities from the logistic regression model, revealing a distinct bimodal distribution. This indicates strong confidence in binary classifications. The clear separation suggests that selected features effectively distinguish between classes.

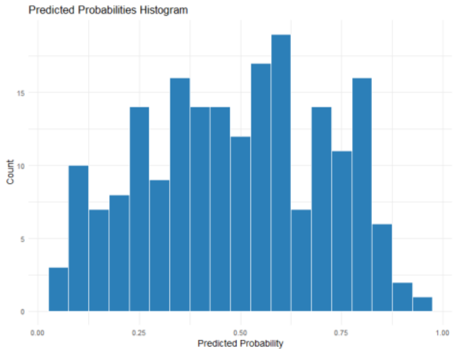


Figure 5.28. Predicted Probabilities Histogram of Logistic Regression

Figure 5.29 indicates the top 20 logistic regression coefficients. *Offences_person_convictions_pct* emerges as the most discriminative variable. This suggest that person offences significantly influence homicide conviction rates. Besides, regional and area-based variables are notably stands in higher rank. So, it can also say that regional differences contribute to classification.

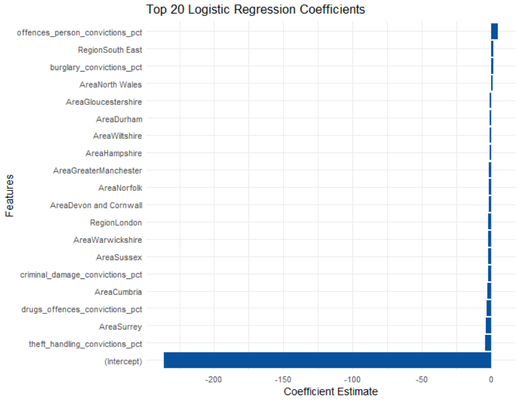
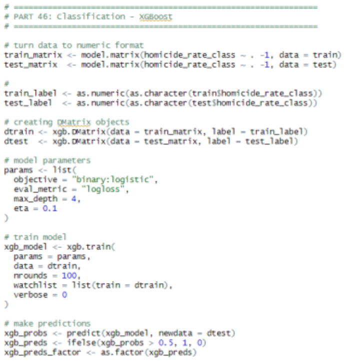


Figure 5.29. Top 20 Logistic Regression Coefficients

5.3.3 XGBoost

While performing the XGBoost model, the *eta* parameter was selected as 0.1 and the *max_depth* parameter as 4 to prevent the model from being complex and the risk of overfitting. It was also given that the objective was binary. Figure 5.30 indicates the implementation of XGBoost classification.



```
# ================================================================
# PART 46: Classification - XGBoost
# ================================================================

# turn data to numeric format
train_matrix <- model.matrix(homicide_rate_class ~ . -1, data = train)
test_matrix  <- model.matrix(homicide_rate_class ~ . -1, data = test)

#
train_label <- as.numeric(as.character(train$homicide_rate_class))
test_label  <- as.numeric(as.character(test$homicide_rate_class))

# creating DMatrix objects
dtrain <- xgb.DMatrix(data = train_matrix, label = train_label)
dtest  <- xgb.DMatrix(data = test_matrix, label = test_label)

# model parameters
params <- list(
  objective = "binary:logistic",
  eval_metric = "logloss",
  max_depth = 4,
  eta = 0.1
)

# train model
xgb_model <- xgb.train(
  params = params,
  data = dtrain,
  nrounds = 100,
  watchlist = list(train = dtrain),
  verbose = 0
)

# make predictions
xgb_probs <- predict(xgb_model, newdata = dtest)
xgb_preds <- ifelse(xgb_probs > 0.5, 1, 0)
xgb_preds_factor <- as.factor(xgb_preds)
```

Figure 5.30. Implementation of XGBoost

Figure 5.31 presents the confusion matrix of the XGBoost classifier. The model correctly classified 60 true negatives and 64 true positives. However, it also misclassified 36 instances as false negatives and 40 as false positives. Additionally, model achieved 0,620 macro f1 score.

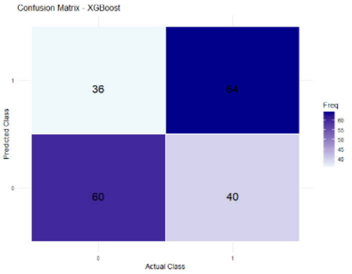


Figure 5.31. Confusion Matrix of XGBoost

As seen in Figure 5.32, when the feature importance of the XGBoost model was examined, it was seen that the total conviction numbers were the most important feature in distinguishing homicide rates. Afterwards, it was seen that the percentages of person offenses and public offenses were effective. Unlike other models, the XGBoost model did not learn much from area information.

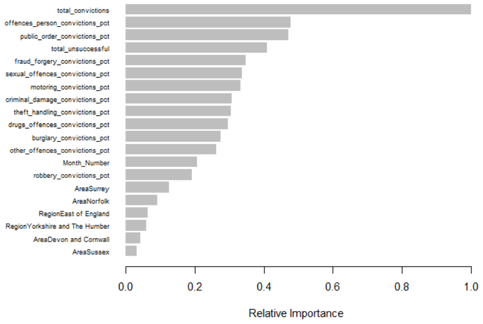
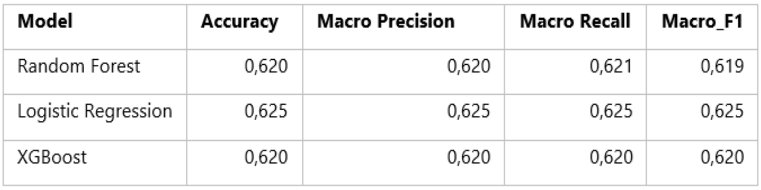


Figure 5.32. Feature Importance of XGBoost

5.3.4 Model Comparison

The performance comparison of Random Forest, Logistic Regression, and XGBoost reveals that Logistic Regression delivered the most consistent and balanced results. It achieved the highest accuracy, macro precision, recall, and F1-score, each reaching a value of 0.625. Random Forest and XGBoost both followed with slightly lower but equal scores of 0.620 across all metrics. Besides, Random Forest and Logistic Regression models learned a lot from the area feature and benefited a lot from homicide prediction. On the contrary, XGBoost model found the total conviction and total unsuccessful columns more important. Table 5.4 gives the results of the models in different valuation metrics.



Model	Accuracy	Macro Precision	Macro Recall	Macro_F1
Random Forest	0,620	0,620	0,621	0,619
Logistic Regression	0,625	0,625	0,625	0,625
XGBoost	0,620	0,620	0,620	0,620

Table 5.4. Results of Classification Models

Lastly, Figure 5.33 illustrates the radar chart of classifications models over 4 different performance metrics and there are slight differences between models’ performance.

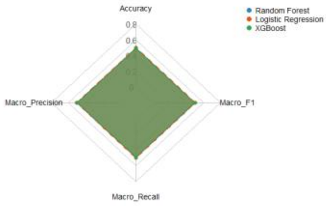


Figure 5.33. Radar Chart of Classification Models

6 Critical Evaluation of Tools and Techniques

This section provides a critical evaluation of the tools, techniques, and performance metrics applied throughout the study. The aim is to assess the suitability, strengths, and limitations of the selected modelling approaches, and evaluation metrics in relation to the dataset and research objectives.

6.1 Critical Evaluation of Applied Techniques

In this study 9 different modelling techniques were used. In regression step, first technique was OLS Model. The model aims to determine the linear relationship that best fits the observed data points (Kuchibhotla et al., 2018). Detailly, Riveros (2020) found that the OLS estimator produced reliable and consistent estimates even with small sample sizes, provided that the classical assumptions of the linear regression model were satisfied. Second regression model was Ridge Model. In this context, Pellegrini et al. (2025) analysed the performance of Ridge regression under small sample sizes and varying correlation coefficients. The results showed that Ridge regression provides more stable and reliable estimates compared to OLS, especially in cases of multicollinearity. Moreover, third model was Lasso Model. In Lasso context, Lee et al. (2022) evaluated the estimation and variable selection performance of the Lasso method at different sample sizes. The results revealed that Lasso can provide effective estimations even at small sample sizes.

In clustering step, first technique was K-means. K-means is a clustering algorithm that groups data into k clusters. It finds the centre of each cluster and assign data points to the nearest centre (Hartigan and Wong, 1979). Ahmed et al. (2020) conducted a study explaining the easy implementation of the K-means algorithm. In this study, they also stated the performance degradation of the algorithm in non-numerical features and the limitations of the model with the need for a predefined cluster number. Second clustering method was Hierarchical Clustering. Ran et al. (2023) emphasized in their study on hierarchical clustering methods that these methods are time and space consuming, and drew attention to their easy implementation in languages such as R. Last clustering method was DBSCAN. The algorithm provides a density-based clustering approach and does not require a predetermined number of clusters. It aims to discover clusters of different shapes and sizes. It can also effectively identify outliers (Ester et al., 1996). Similarly, Wang et al. (2022) demonstrated the capability of the DBSCAN algorithm in variable density datasets. They also revealed that using the correct *eps* and *minpts* variables improved the classification performance. In this study, this algorithm outperformed the others with the correct parameter usage.

In classification step, first two techniques were Random Forest and Logistic Regression. The Random Forest algorithm, which tries to predict the problem by creating multiple decision trees, has been examined in many studies. For instance, Schonlau and Zou (2020) revealed the advantages of RF such as high accuracy rates, ability to determine variable importance, and robustness against over-fitting. They also stated that RF is effective in dealing with complex data structures thanks to its non-parametric structure. On the other hand, Logistic Regression is particularly suitable for binary classification tasks due to its probabilistic output and linear decision boundaries (Zaidi and Al Luhayb, 2023). Additionally, Kirasich et al. (2018) compared the performance of Random Forest and Logistic Regression algorithms on imbalanced data sets and the results show that Random Forest provides higher accuracy on imbalanced data sets. In summary, these two algorithms are among the most preferred models in classification problems due to the accuracy success of Random Forest and the easy implementation of Logistic Regression. Lastly, XGBoost model was applied. Unlike traditional methods like Random Forest and Logistic Regression, XGBoost is an ensemble learning method based on the principle of "gradient boosting". This model builds successive decision trees to minimize errors, and each new tree tries to correct the errors of the previous one (Bentéjac et al., 2020). Furthermore, Djon et al. (2023) performed a comparative analysis on theft prediction in Chicago and the XGBoost model outperformed other models with an F1 score of 86%.

6.2 Critical Evaluation of Evaluation Metrics

In this study, silhouette score was used in the evaluation phase of clustering models. The Silhouette Score measures how well data points fit within their assigned clusters, balancing cohesion and separation, with values closer to 1 indicating better clustering (Rousseeuw, 1987). Shutaywi and Kachouie (2021) emphasized that silhouette score is a suitable method for evaluating the quality of clustering results in unsupervised learning scenarios, especially in cases where there is no training data.

In the classification step, macro F1 score was selected as the evaluation metric. Although it is recommended to use macro F1 score in unbalanced distributions, it was seen as a reasonable evaluation metric because our target, which is numerical, was changed to binary in the study. For example, Opitz (2024) states that Macro F1 score is a popular choice due to its sensitivity to class prevalence and the need for high accuracy for all classes. Additionally, Hinojosa Lee et al. (2024) state that Macro F1 score reflects the performance of minority classes more accurately by giving equal weight to all classes, especially in unbalanced data sets, and therefore should be preferred.

7 Conclusion

This study meticulously explored the Crown Prosecution Service's case outcomes dataset using data analysis and predictive modelling techniques. Initially, data preparation step performed to enhance analysis and modelling results. After that, descriptive analysis performed and this revealed patterns in conviction rates across offence categories, regions, and time. For instance, theft handling, offences against the person, and motoring offences convictions were dominated with reaching over 60% of cases. In other context, conviction success rates were generally higher than 50%. However, sexual offences exhibited a notably lower success rate, coupled with a comparatively higher proportion of unsuccessful outcomes. This indicated the evidentiary and procedural complexities inherent in prosecuting such crimes. Additionally, regional disparities were found in metropolitan areas, particularly London, consistently reporting higher conviction volumes and broader distribution ranges. This indicated the influence of demographic and socio-economic factors on crime and justice outcomes.

Cluster analyses provided different perspective by identifying groups in the data. K-Means clustering with a silhouette score of 0,85, examined cases in two clusters based on their conviction volume. On the other hand, DBSCAN, which showed superior performance with a silhouette score of 0,886, provided a detailed understanding of anomalous conviction records by identifying outliers. Lastly, hierarchical clustering yielded 0,35 silhouette score. As a result, these unsupervised learning methods highlighted the existence of both high-activity crime areas and low-crime areas. They also highlighted potential area-specific policy strategies.

Regression modelling was performed to predict drug offence convictions based on crime categories. Among the applied models, the Lasso regression achieved the highest explanatory power with an R^2 value of 0,905. In other perspective, OLS delivered the most precise predictions in terms of RMSE. Across models, homicide and robbery convictions consistently emerged as significant predictors of drug offences. These findings confirm that certain crime categories are statistically interlinked, offering predictive opportunities for resource allocation and early intervention strategies within the criminal justice system.

Finally, classification models employed to predict homicide conviction outcomes based on conviction percentages. Logistic Regression demonstrated the highest macro F1 score and Random Forest and XGBoost closely followed. All models identified percentage convictions for crimes against the person and total convictions as significant predictors. The moderate but consistent performance of the classification models highlighted the complex nature of predicting murder convictions. It also demonstrated that data-driven approaches can support decision-making frameworks. All in all, this study provides a robust analytical framework for understanding conviction outcomes, providing evidence-based recommendations for policy improvement, operational planning, and targeted crime reduction initiatives.

7.1 Social Perspective

This study has revealed regional differences and crime-specific conviction patterns, and highlighted potential disparities in prosecution. The significantly lower conviction rates and higher unsuccessful rates for sensitive categories such as sexual offences highlight the systemic challenges that vulnerable victims face in seeking justice. Moreover, the consistent dominance of certain types of crime, particularly theft and driving, reflects broader social issues such as urban crime density, socioeconomic disparities and the demand for focused prevention policies.

From a policy perspective, the findings suggest that the administration of justice is not experienced uniformly across communities, particularly in metropolitan areas. These findings highlight the need for regionally adaptive policing strategies and equitable legal resources. Moreover, the moderate predictive success of classification models for homicide cases demonstrates the complexity of criminal behaviour, which is influenced by broader social, economic and cultural factors. Overall, this study highlights the importance of data-driven justice reform by advocating for evidence-based policies that not only address crime but also reduce regional disparities and increase public trust in legal systems.

7.2 Future Works

Future studies could expand this analysis by integrating socio-economic and policing resource variables. This could enhance model interpretability and predictive power. In addition, real-time data streams could be incorporated to develop dynamic predictive models. With their capability of forecasting crime trends, this would enable authorities to anticipate crime surges and allocate resources more effectively.

Additionally, hypothesis testing techniques are omitted in this report to comply with report length restrictions. Critically evaluating these techniques is suggested as another possible future work. Additionally, classification models could be expanded using advanced ensemble techniques or deep learning architectures, improving predictive performance for complex outcomes like homicide convictions.

Lastly, deploying the developed models within operational justice systems could offer decision support for court case management and targeted intervention programs. This allows findings from past analyses to be used to predict the future. Also, these predictions can help public safety, and the justice system operate more efficiently.

8 References

Castro-Toledo, F.J., Miró-Llinares, F. and Aguerri, J.C. (2023) ‘Data-driven criminal justice in the age of algorithms: epistemic challenges and practical implications’, *Criminal Law Forum*, 34, pp. 295–316. doi:10.1007/s10609-023-09454-y.

Lavorgna, A. and Ugwudike, P. (2021). ‘The datafication revolution in criminal justice: An empirical exploration of frames portraying data-driven technologies for crime prevention and control.’ *Big Data & Society*, 8(2). doi: 10.1177/20539517211049670.

Kuchibhotla, A.K., Brown, L.D. and Buja, A. (2018) ‘Model-free study of ordinary least squares linear regression’, *arXiv preprint*, 1809.10538. doi: arxiv.org/abs/1809.10538.

Riveros, J. (2020) ‘Low sample size and regression: A Monte Carlo approach’, *Journal of Applied Economic Sciences*, XV, pp. 22–44. doi: /10.14505/jaes.v15.1(67).02.

Pellegrini, P., Leuci, E., Pellegrini, C., Pupo, S., & Pelizza, L. (2025) “Suggestions on early psychosis: towards a new psychopathology?,” *Academia Mental Health and Well-Being. Academia.edu Journals*, 2(1). doi: 10.20935/MHealthWellB7584.

Lee, J.H., Shi, Z. and Gao, Z. (2022) ‘On LASSO for predictive regression’, *Journal of Econometrics*, 229(2), pp. 322–349. doi: 10.1016/j.jeconom.2021.02.002.

Hartigan, J.A. and Wong, M.A. (1979) ‘Algorithm AS 136: A K-means clustering algorithm’, *Journal of the Royal Statistical Society: Series C (Applied Statistics),* 28(1), pp. 100–108. doi:10.2307/2346830.

Ahmed, M., Seraj, R. and Islam, S.M.S. (2020) ‘The k-means algorithm: A comprehensive survey and performance evaluation’, *Electronics*, 9, p. 1295. doi: /10.3390/electronics9081295.

Ran, X., Xi, Y., Lu, Y. et al. (2023) ‘Comprehensive survey on hierarchical clustering algorithms and the recent developments’, *Artificial Intelligence Review*, 56, pp. 8219–8264. doi: 10.1007/s10462-022-10366-3.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996) ‘A density-based algorithm for discovering clusters in large spatial databases with noise’, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231. doi: /10.5555/3001460.3001507.

Wang, Z., Ye, Z., Du, Y., Mao, Y., Liu, Y., Wu, Z. and Wang, J. (2022) ‘AMD-DBSCAN: An adaptive multi-density DBSCAN for datasets of extremely variable density’, *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10. doi: /10.1109/DSAA54385.2022.10032412.

Schonlau, M. and Zou, R.Y. (2020) ‘The random forest algorithm for statistical learning’, *The Stata Journal*, 20(1), pp. 3–29. doi: 10.1177/1536867X20909688.

Zaidi, A. and Al Luhayb, A. (2023) ‘Two statistical approaches to justify the use of the logistic function in binary logistic regression’, *Mathematical Problems in Engineering*, 2023, pp. 1–11. doi:10.1155/2023/5525675.

Kirasich, K., Smith, T. and Sadler, B. (2018) ‘Random Forest vs Logistic Regression: Binary Classification for Heterogeneous Datasets’, *SMU Data Science Review*, 1(3), Article 9. Available at: https://scholar.smu.edu/datasciencereview/vol1/iss3/9.

Bentéjac, C., Csörgő, A. and Martínez-Muñoz, G. (2020) ‘A comparative analysis of gradient boosting algorithms’, *Artificial Intelligence Review*, 54(3), pp. 1937–1967. doi: 10.1007/s10462-020-09896-5.

Djon, D., Jhawar, J., Drumm, K. and Tran, V. (2023) ‘A comparative analysis of multiple methods for predicting a specific type of crime in the city of Chicago’, *arXiv preprint*, 2304.13464. Available at: https://arxiv.org/abs/2304.13464.

Rousseeuw, P.J. (1987) 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *Journal of Computational and Applied Mathematics*, 20, pp. 53–65. doi: 10.1016/0377-0427(87)90125-7.

Shutaywi, M. and Kachouie, N.N. (2021) ‘Silhouette analysis for performance evaluation in machine learning with applications to clustering’, *Entropy*, 23(6), p. 759. doi: 10.3390/e23060759.

Opitz, J. (2024) ‘A closer look at classification evaluation metrics and a critical reflection of common evaluation practice’, *Transactions of the Association for Computational Linguistics*, 12, pp. 820–836.doi: 10.1162/tacl_a_00675.

Hinojosa Lee, M.C., Braet, J. and Springael, J. (2024) ‘Performance metrics for multilabel emotion classification: comparing micro, macro, and weighted F1-scores’, *Applied Sciences*, 14, p. 9863. doi: 10.3390/app14219863.