Executive Summary

This report presents a comprehensive analysis of conviction outcomes across CPS regions, aiming to uncover patterns and evaluate performance dynamics. The study focuses on the relationship between workload and conviction success rate, using advanced data processing, statistical modeling, and machine learning techniques.

Regression models—Linear, Ridge, and Lasso—were used to quantify the impact of case volume on conviction success rate, revealing a significant negative correlation in high-workload regions. Binary and multi-class classification models, including Logistic Regression and Random Forest, were implemented to predict success rate categories, achieving high accuracy and interpretability. Clustering techniques such as K-Means, DBSCAN, and Gaussian Mixture Models uncovered hidden regional groupings, offering insights into performance trends and anomalies.

Descriptive statistics, correlation analysis, and hypothesis testing (ANOVA and Chi-Square) were employed to validate assumptions and enhance understanding. A combination of ggplot2, corrplot, and other R visualization libraries effectively communicated findings throughout the analysis.

The results highlight disparities in CPS regional outcomes and demonstrate the effectiveness of using data-driven methods to support strategic evaluation and decision-making. Recommendations are proposed for further analysis, capacity planning, and data enhancements.

Table of Contents

Chapter 1: Introduction

1.1 Background and context of the CPS Dataset

The integration of data analytics in public sector institutions has become vital for enhancing transparency, operational efficiency, and informed policy-making. Within the criminal justice system of England and Wales, the Crown Prosecution Service (CPS) holds the primary role of prosecuting criminal cases investigated by police forces. As part of its open data initiative, the CPS publishes structured datasets that provide detailed accounts of prosecution outcomes, supporting public accountability and enabling data-driven decision-making (Service, 2018).

Among these resources, the "Case Outcomes by Principal Offence Category" dataset stands out for its analytical value. It contains weekly statistics on conviction rates, unsuccessful cases, and procedural outcomes categorized by offense type. The dataset's standardized format and consistent publication schedule make it particularly well-suited for longitudinal analysis, cross-sectional comparisons, and the application of advanced modeling techniques (Taylor, Singh, & McBride, 2021).

For this project, CPS data from January 2014 to October 2016 was selected. This timeframe ensures structural consistency and data reliability, minimizing reporting discrepancies and schema changes. This foundation enables the application of descriptive, inferential, and predictive analytics to uncover insights into conviction trends, workload distribution, and performance outcomes across CPS regions.

This study demonstrates the untapped potential of open legal data to generate actionable insights. Through a combination of descriptive analytics, statistical inference, and machine learning, the project contributes to the evolving domain of justice analytics and highlights the importance of ethical, data-informed governance (Barocas, Hardt, & Narayanan, 2019).

1.2 Problem Statement and Stakeholder Importance

For this analysis, workload refers to the overall number of cases processed by each CPS region, including both successful and unsuccessful outcomes. Stakeholders include CPS analysts, policymakers, and public sector data teams who benefit from these insights.

Despite its regular updates and comprehensive coverage, the CPS dataset is often underutilized in advanced analytics. Current usage tends to focus on basic statistics, missing opportunities to explore deeper structural insights, fairness evaluations, or predictive capabilities. This limits the dataset's potential for driving justice reform, evaluating regional performance, or informing strategic decisions (James et al., 2013).

This project addresses that gap by employing a systematic, multi-phase analytical workflow. This includes data cleaning, exploratory analysis, hypothesis testing, and the implementation of machine learning models for classification and clustering. Each step emphasizes reproducibility, stakeholder alignment, and adherence to ethical standards in data science.

Key stakeholders benefiting from this project include:

- **CPS Analysts** seeking performance benchmarks and anomaly detection.
- **Policy Makers and Reform Advocates** pursuing equitable and data-driven decision-making.

- **Public Sector Data Scientists** requiring methodological guidance and reproducible workflows.

Potential issues such as algorithmic bias, lack of model transparency, and misinterpretation of statistical results are critically examined, drawing on literature from responsible machine learning practices (Barocas, Hardt, & Narayanan, 2019).

1.3 Research Aims and Inquiries

The project is guided by a structured framework of objectives and research questions, ensuring that every analytical stage is deliberate and aligned with public sector goals of transparency, fairness, and accountability.

1.3.1 Research Objectives

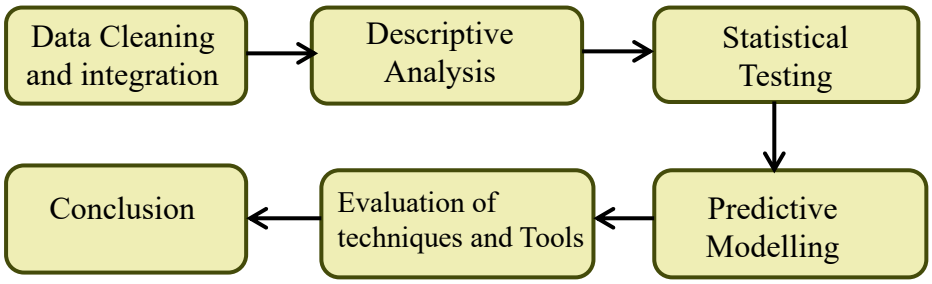
- To clean, combine, and prepare CPS data from January 2014 to October 2016.
- To identify trends and patterns in prosecution outcomes using descriptive analytics.
- To test the statistical significance of conviction differences using ANOVA and Chi-Square tests.
- To build and evaluate regression, classification, and clustering models.
- To assess the ethical implications of machine learning in justice data.

1.3.2 Research Questions

1. What temporal and categorical patterns emerge in CPS outcomes across offense types?
2. Are there statistically significant differences in conviction rates across regions or offense types?
3. Which machine learning models offer the best trade-offs between accuracy, interpretability, and fairness?
4. What are the ethical implications of applying predictive analytics to public legal datasets?

1.4 Structure of the Analysis

This report follows a structured data science process, progressing logically from data preparation to evaluation and insight generation. The workflow below summarises the sequence and purpose of each section in the report.



Chapter 2 - Dataset Preparation: Integration, Processing, and Cleaning

2.1 Overview of CPS Dataset and File Structure

This research utilizes the Crown Prosecution Service (CPS) Case Outcomes by Principal Offence Category dataset, available through the UK Government's open data platform (Service, 2018). The dataset comprises monthly records of prosecution outcomes classified by principal offense types and CPS regions in England and Wales. The structured format renders it appropriate for time-series analysis, descriptive statistics, and predictive modeling within criminal justice analytics.

From the full dataset, which spans January 2014 to December 2018, a subset of 33 monthly CSV files—covering January 2014 to October 2016—was selected for analysis. This includes 12 files from 2014, 11 from 2015, and 10 from 2016. The selection was based on criteria including data completeness, structural consistency, and the absence of formatting anomalies, which were present in multiple files from 2017 and 2018. The final selection also exceeds the 24-month minimum requirement, establishing a robust foundation for longitudinal and inferential analysis.

Each file adheres to a standardized tabular format, documenting variables including offense type, outcome counts, conviction rates, and administrative closures. The final dataset, post-integration, comprises 1,419 rows and 53 columns, with variables categorized as follows:

Category	**Description**
Offense Type	Principal groupings such as Homicide, Sexual Offences, Drugs, and Fraud
Conviction Metrics	Number and percentage of convictions by offense type
Unsuccessful Outcomes	Number and percentage of unsuccessful prosecutions per category
Administrative Outcomes	Cases finalized without trial or via alternative administrative resolution

2.2 Loading Dataset and Merging CSV Files

A functional programming approach was employed in R to efficiently import and merge the dataset. The CPS dataset contains 33 monthly CSV files, each stored in separate folders according to year (2014, 2015, and 2016). Rather than loading each file manually, the list.files() function was used in conjunction with map () from the **purrr** package to dynamically access and iterate over all relevant file paths. This approach was chosen over traditional for-loops because it offers greater scalability, reduces redundancy, and integrates seamlessly with the tidyverse workflow.

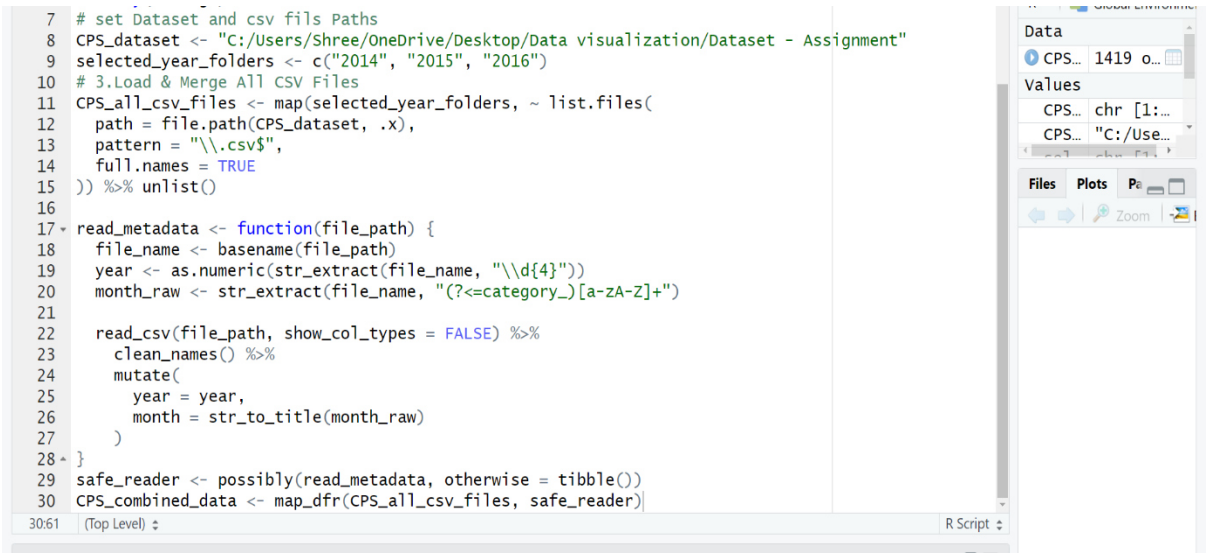
To maintain the temporal context of the data, a custom function named read_metadata() was developed. This function used str_extract() from the **stringr** package to extract the **year** and **month** from the filename. The filenames followed a consistent structure (e.g., category_january_2014.csv), making this extraction process reliable. The extracted metadata was then appended to each dataset as new columns. To ensure consistent formatting of month names, str_to_title() was applied, which standardised the capitalisation for future grouping and trend analysis.

In the same function, the clean_names() function from the **janitor** package was used to clean column headers during the import stage. This automatically converted all variable names to lowercase and replaced spaces and special characters with underscores (e.g., “Conviction %” becomes “conviction_percent”). Cleaning the names early in the pipeline helped avoid syntax issues later and ensured alignment with tidy data practices.

To safeguard the process against file errors, the function was wrapped with possibly() from the **purrr** package. This allowed the script to safely skip over any missing or malformed files without interrupting execution. Instead of causing the process to fail, these files returned an empty tibble, which preserved the structure of the final dataset and improved overall reliability. This was especially useful when working with public datasets that may have inconsistencies.

Finally, the map_dfr() function was used to apply the reader function to all file paths and combine the results into a single, cohesive data frame. This method was preferred over a loop-based merge because it automatically aligns columns, eliminates the need for a mutable state, and improves code readability and maintainability.

Overall, this method resulted in a fully integrated dataset containing 33 months of records, consistent column names, and embedded temporal metadata. It also ensures that the entire ingestion process is automated, fault-tolerant, and reproducible. Should additional monthly files be added in the future, the same script can be reused without modification, making it ideal for iterative or time-based analyses.



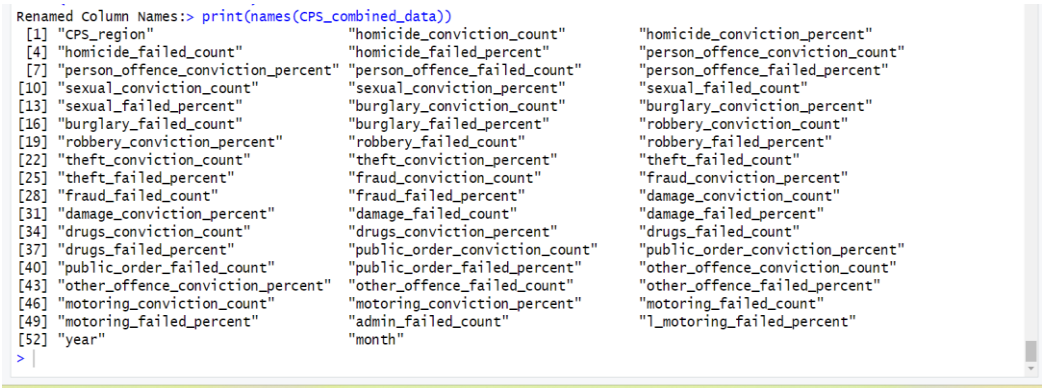
2.3 Column Names and Renaming

Two stages of column name cleaning were applied to ensure consistency and improve the usability of the dataset. During the import process, the clean_names() function from the **janitor**

package was used to automatically convert all column headers to lowercase and replace spaces and punctuation with underscores. This helped prevent syntax errors and ensured compatibility with tidyverse functions.

Following the automated cleaning, a manual renaming step was carried out to simplify lengthy and complex variable names, especially those relating to conviction and failure counts across offence categories. For instance, number_of_drugs_offences_convictions was renamed to drugs_conviction_count. This approach improved readability, reduced redundancy, and introduced a consistent naming convention in the format offence_outcome_measure, which was essential for grouping, summarising, and visualizing outcomes.

The first unnamed column in the dataset, originally labeled x1, contained CPS region names. To make this clearer, it was renamed to cps_region.



2.4 Handling Invalid or Placeholder Values

To enhance data consistency, I addressed several placeholder values found in character columns, including hyphens ("-"), "n/a", and empty strings (""). These were converted to NA using the na_if() function, ensuring they were correctly recognized as missing values during analysis.

Some percentage columns also contained "%" symbols stored as part of character strings. Rather than removing these columns, I used str_remove() to strip the symbol and then converted the values to numeric using as.numeric(). This made them suitable for analysis while retaining their potential usefulness.

Although these percentage variables may overlap with existing count-based columns, I chose to retain them for now. I plan to evaluate their relevance further and will decide whether to keep or discard them based on their analytical value.

```
#Handle Placeholder Values and Convert Percentages
CPS_combined_data <- CPS_combined_data %>%
  mutate(across(where(is.character), ~ na_if(.x, "-"))) %>%
  mutate(across(where(is.character), ~ na_if(.x, "n/a"))) %>%
  mutate(across(where(is.character), ~ na_if(.x, ""))) %>%
  mutate(across(where(is.character), ~ ifelse(str_detect(.x, "%"),
                                              str_remove(.x, "%"), .x))) %>%
  mutate(across(contains("percentage"), as.numeric))
```

2.5 Missing month in Dataset and justification

To verify the temporal completeness of the dataset, I examined whether any months were missing from each year between 2014 and 2016. Since the month column was already clean and consistently formatted (e.g., "January", "February"), no further standardisation was required.

Using the built-in month.name list as a reference, I grouped the dataset by year and used the setdiff() function to identify which months were absent. This comparison showed that some months were not present in certain years — for example, November and December 2016 were missing, as the dataset only covers January to October for that year.

While missing months can suggest incomplete reporting, they may also reflect periods of administrative delay, non-submission, or simply months with no recorded prosecution outcomes. Rather than assuming data loss, these gaps were acknowledged for context, and no artificial data was introduced. Identifying and documenting these omissions helps prevent misinterpretation in time-based visualisations and monthly aggregations.

```
#check which month data missing
expected_months <- factor(month.name, levels = month.name)
missing_months_df <- CPS_combined_data %>%
  group_by(year) %>%
  summarise(actual_months = list(unique(month))) %>%
  rowwise() %>%
  mutate(missing_months = list(setdiff(expected_months, actual_months))) %>%
  select(year, missing_months) %>%
  unnest(missing_months)
print(missing_months_df)
```

	year	missing_months
1	2015	November
2	2016	February
3	2016	March

2.6 Checked Duplicate and Missing value

To ensure the dataset's integrity before analysis, I performed checks for duplicate records and missing values. Duplicate entries can inflate totals and distort statistical results, so I used the duplicated() function to detect any repeated rows. The output confirmed that there were no duplicates, and no further action was required.

I then created a summary showing the count and percentage of NA values in each column using colSums() and colMeans(). This helped identify variables with incomplete data, with missing values primarily occurring in columns representing percentages.

Since these are derived metrics rather than raw counts, and their absence does not immediately impact modelling or aggregation at this stage, I chose not to impute or remove them. This approach preserves the structure of the dataset while allowing flexibility for further analysis.

```
118 #Check for duplicate rows
119 duplicate_count <- sum(duplicated(CPS_combined_data))
120 print(paste("Duplicate rows:", duplicate_count))
121 #Check Missing value count and percentage
122 missing_summary <- data.frame(
123   column_name = names(CPS_combined_data),
124   missing_count = colSums(is.na(CPS_combined_data)),
125   missing_percent = round(colMeans(is.na(CPS_combined_data)) * 100, 2)
126 )
127 print(missing_summary %>% filter(missing_count > 0))
128
```

```
R 4.5.0 · C:/Users/Shree/OneDrive/Desktop/Data visualization/R/
> print(missing_summary %>% filter(missing_count > 0))
                                    column_name missing_count missing_percent
homicide_conviction_percent homicide_conviction_percent           491          34.60
homicide_failed_percent         homicide_failed_percent           491          34.60
l_motoring_failed_percent     l_motoring_failed_percent            14           0.99
> View(missing_summary)
>
```

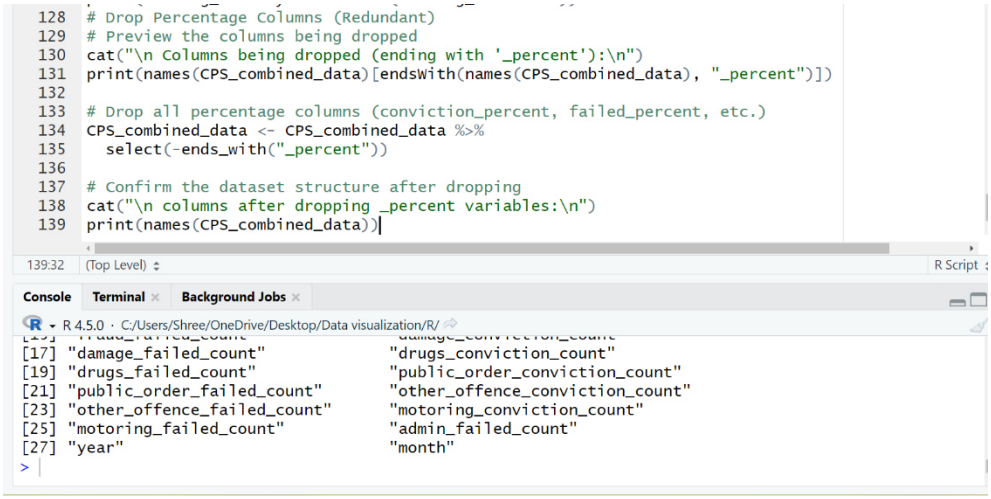
2.7 Removal of Redundant Percentage Columns

As part of the data refinement process, I identified several columns ending with _percent that contained percentage-based outcome metrics such as conviction and failure rates for each offence category. These percentage values were derived directly from raw count variables already present in the dataset, making them redundant.

To reduce dimensionality and avoid potential multicollinearity in later modelling, I removed all columns ending in percent using the ends_with() function from the **dplyr** package. This included conviction rates, failure percentages, and other derived ratios across offence types.

An added benefit of this step was the removal of the dataset's remaining missing values. The _percent columns were the only fields with NAs, so their removal resulted in a fully complete dataset—free of any missing data.

By retaining only the raw counts, I preserved analytical flexibility, as percentage metrics can be recalculated if needed during feature engineering



2.8 Feature Engineering (Succes rate, totals)

To enrich the dataset and prepare it for analysis, I created several new variables derived from conviction and failure counts. Using rowSums(), I calculated total_convictions and total_unsuccessful across all offence types, which were then combined into overall_total_cases to represent total caseload per month.

I computed overall_success_rate as the percentage of convictions out of total cases. This replaced the dropped percentage columns with a cleaner, standardised metric that can be recalculated or adjusted as needed.

For temporal consistency, a year_month field was generated by combining the existing year and month into a "YYYY-MM" format. Additionally, I created a case_volume_tier variable using quantile-based binning, classifying records as “Low”, “Medium”, or “High” volume to support region grouped comparisons.

These engineered features improved the dataset’s structure and interpretability, supporting further analysis, visualisation, and predictive modelling.

```
140 #Feature Engineering
141 CPS_combined_data <- CPS_combined_data %>%
142   # Total convictions across all offence types
143   mutate(
144     total_convictions = rowSums(select(., ends_with("conviction_count")), na.rm = TRUE),
145     total_unsuccessful = rowSums(select(., ends_with("failed_count")), na.rm = TRUE),
146     overall_total_cases = total_convictions + total_unsuccessful,
147
148     # Success rate as a percentage
149     overall_success_rate = round((total_convictions / overall_total_cases) * 100, 2),
150
151     # Combine year + month for time-series analysis
152     year_month = paste0(year, "-", str_pad(match(month, month.name), 2, pad = "0")),
153
154     # Categorize regions by volume of cases
155     case_volume_tier = cut(
156       overall_total_cases,
157       breaks = quantile(overall_total_cases, probs = c(0, 0.33, 0.66, 1), na.rm = TRUE),
158       labels = c("Low", "Medium", "High"),
159       include.lowest = TRUE
160     )
161   )
```

2.9 Variable and Conversion Factor

To finalise the dataset, variables were classified as categorical or numerical and converted to the appropriate format. For better grouping in analysis, CPS_region, month, and case_volume_tier were converted to factors. Before conversion, month values were cleaned (trimmed and title-cased) and transformed using factor(levels = month.name, labels = month.abb) to ensure uniform, abbreviated, and properly ordered month names. This prevented NAs and supported accurate time-based visualisations. The year_month column was kept as a character string to preserve chronological order for time-series analysis. Numeric variables, such as conviction counts and success rates, were retained in numeric format.

```
# Display categorical (character or factor) columns
cat("Categorical Columns in crime dataset:\n")
categorical_columns_CPS <- names(select_if(CPS_combined_data, ~ i
print(categorical_columns_CPS)

#Display numerical columns
cat("\n Numerical Columns in crime dataset:\n")
numerical_columns_CPS <- names(select_if(CPS_combined_data, is.nu
print(numerical_columns_CPS)

# Convert categorical columns
CPS_combined_data <- CPS_combined_data %>%
  mutate(
    # Converted CPS_region and case_volume_tier to factor
    CPS_region = as.factor(CPS_region),
    case_volume_tier = as.factor(case_volume_tier),

    # converted 'month' to factor
    month = month %>%
      as.character() %>%        # for checking it's a character
      str_trim() %>%            # remove leading/trailing whitesp
      str_to_title() %>%        # ensure correct capitalization (
      factor(levels = month.name, labels = month.abb) # convert t
  )
cat("Column types after conversion:\n")
```

2.10 Final CPS Dataset Summary

As a final validation step, I assessed the structure and integrity of the dataset after completing all cleaning and preprocessing tasks. The dataset contains **1,419 rows** and **34 columns**, including engineered features and cleaned outcome variables.

I used glimpse() and sapply(..., class) to confirm that all variables were correctly typed—categorical fields were represented as factors, and numerical fields retained their proper format. A final check using colSums(is.na()) confirmed the absence of missing values, indicating that all placeholder entries were addressed and redundant columns removed.

```
cat("Column types after conversion:\n")
sapply(CPS_combined_data[c("CPS_region", "month", "case_volume_ti
#Final Dataset Summmary
#View structure (column types & sample values)
cat("\n Structure of Final Dataset:\n")
glimpse(CPS_combined_data)

#Shape number of rows and columns
cat("\n Dataset Dimensions:\n")
cat("Rows:", nrow(CPS_combined_data), "\n")
cat("Columns:", ncol(CPS_combined_data), "\n")

#Data types of each column
cat("\n Data Types of Columns:\n")
print(sapply(CPS_combined_data, class))

#Missing values summary
# Count missing values per column
missing_values <- colSums(is.na(CPS_combined_data))
cat(missing_values)
```

```
# Load Required Libraries
library(tidyverse)
library(janitor)
library(lubridate)
library(naniar)
library(stringr)
```

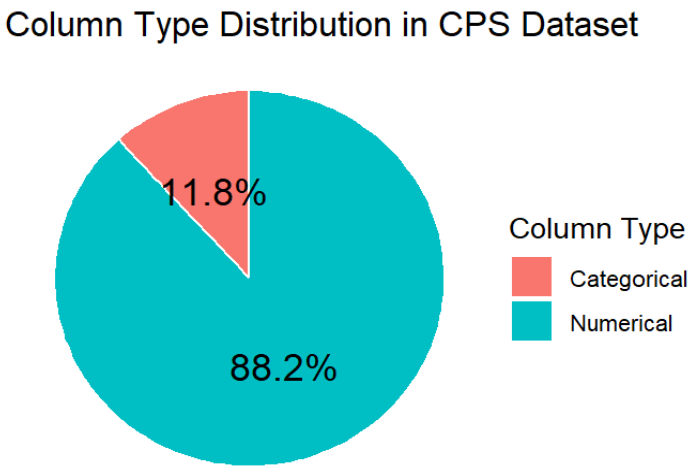
Libraries used for Cleaning and integration

Chapter 3 – Descriptive analysis

3.1 Column Type Distribution: Categorical vs Numerical Analysis

To guide appropriate analysis and modelling, all variables were classified as either **categorical** or **numerical**. Of the 34 columns, **4 (11.8%) are categorical** (e.g., CPS_region, month), and **30 (88.2%) are numerical**, including offence counts and success metrics.

A pie chart (Figure 3.1) illustrates this distribution, confirming the dataset's strong numerical structure and suitability for statistical analysis. This step also ensures proper handling of categorical data in later stages.




```
206
207 # Descriptive Analysis
208 library(ggplot2)
209 library(dplyr)
210
211 # Created summary with percentage
212 column_type_df <- data.frame(
213   Type = c("Categorical", "Numerical"),
214   Count = c(length(categorical_columns_CPS), length(numerical_col
215 ) %>%
216   mutate(Percentage = round(Count / sum(Count) * 100, 1),
217          Label = paste0(Percentage, "%"))
218
219 # Plot pie chart with percentage labels
220 ggplot(column_type_df, aes(x = "", y = Count, fill = Type)) +
221   geom_col(width = 1, color = "white") +
222   coord_polar("y") +
223   geom_text(aes(label = Label),
224             position = position_stack(vjust = 0.5),
225             color = "black", size = 5) +
226   labs(title = "Column Type Distribution in CPS Dataset", fill =
227   theme_void()
228
```

3.2 Descriptive Statistical Analysis of Numerical and Categorical Variables

This section provides an overview of the key variables in the CPS dataset by separating them into numerical and categorical types. Basic statistical summaries were generated to explore the distribution, range, and frequency of values. This helps form a general understanding of the dataset before applying visualisation or modelling techniques.

3.2.1 Numerical Feature Distribution and Central Tendencies

Homicide_conviction_count - Values range from 0 to 131, with a median of 1 and a mean of 3.51, indicating that most CPS regions recorded very few homicide convictions per month, except for rare high-volume outliers.

```
Summary for: homicide_conviction_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.000   0.000   1.000   3.514   2.000 131.000
```

Homicide Failed Count - With a median and Q3 of 0, most records report no failed prosecutions, though a few months reached 35, suggesting rare but notable spikes.

```
Summary for: homicide_failed_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0000  0.0000  0.0000  0.8513  0.0000 35.0000
```

Person Offence Conviction Count - This variable has a high median of 177 and a maximum of 11,741, reflecting substantial case volumes and variability across different regions.

```
Summary for: person_offence_conviction_count
   Min. 1st Qu.   Median     Mean 3rd Qu.     Max.
   29.0    114.0    177.0    455.4    273.5 11741.0
```

Person Offence Failed Count - Failures range from 5 to 3,568, with a median of 49, showing a significant but less frequent number of unsuccessful outcomes compared to convictions.

```
Summary for: person_offence_failed_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   5.0    29.0    49.0   143.1    84.0  3568.0
```

Sexual Conviction Count - The median is 15, with a maximum of 1,179, suggesting moderate monthly caseloads with occasional large increases.

```
Summary for: sexual_conviction_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.0     8.0    15.0    42.6    29.0  1179.0
```

Sexual Failed Count - A median of 5 and a maximum of 489 show lower but still occasionally high failed prosecution outcomes in sexual offence cases.

```
Summary for: sexual_failed_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.00    2.00    5.00   16.44   11.00  489.00
```

Burglary Conviction Count - Monthly values range from 2 to 1,715, with a median of 24, indicating consistent moderate activity with infrequent spikes.

```
Summary for: burglary_conviction_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.00   15.00   24.00   64.61   40.00 1715.00
```

Burglary Failed Count - This shows lower unsuccessful counts, with a median of 4 and a mean of 11.5, but some outliers reach 317.

```
Summary for: burglary_failed_count
   Min. 1st Qu.   Median     Mean 3rd Qu.     Max.
   0.00     2.00     4.00    11.49     7.00   317.00
```

Robbery Conviction Count - Convictions are relatively low-volume, with a median of 6 and a mean of 21.1, though some regions recorded up to 650.

```
Summary for: robbery_conviction_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.00    3.00    6.00   21.06   11.00  650.00
```

Robbery Failed Count - The median is just 1, but a maximum of 188 suggests rare months with a high number of failed robbery prosecutions.

```
Summary for: robbery_failed_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.000   0.000   1.000   5.729   3.000 188.000
```

Theft Conviction Count - This variable has a median of 164 and a maximum of 11,057, making it one of the highest-volume offence categories.

```
Summary for: theft_conviction_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  13.0   105.0   164.0   416.5   269.0 11057.0
```

Theft Failed Count- Failure counts remain lower than convictions, with a median of 14 and a mean of 38.2, but can rise significantly in some months.

```
Summary for: theft_failed_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.00    7.00   14.00   38.23   22.00 1025.00
```

Fraud Conviction Count - Monthly convictions are typically low, with a median of 12 and a mean of 37.8, though some months exceed 1,000.

```
Summary for: fraud_conviction_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.00    7.00   12.00   37.82   20.00 1075.00
```

Fraud Failed Count - Most regions record few failed fraud prosecutions (median = 2), but occasional peaks reach up to 180.

```
Summary for: fraud_failed_count
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.00    1.00    2.00    6.32    4.00  180.00
```

Damage Conviction Count - The median is 42 and the mean is 104, with some months recording as many as 2,693 convictions for criminal damage.

```
Summary for: damage_conviction_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    3.0    28.0    42.0   104.4    64.0  2693.0
```

Damage Failed Count - This category shows moderate counts, with a median of 7, but rare months reach 491 failed outcomes.

```
Summary for: damage_failed_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    4.00    7.00   18.55   12.00  491.00
```

Drugs Conviction Count - With a median of 68 and a maximum of 4,988, this offence category reflects high-volume and widespread activity.

```
Summary for: drugs_conviction_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    8.0    40.0    68.0   198.5   109.0  4988.0
```

Drugs Failed Count - Failure outcomes are generally lower, with a median of 4, but can rise to 346 in some regions or periods.

```
Summary for: drugs_failed_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    2.00    4.00   13.25    8.00  346.00
```

Public Order Conviction Count - Convictions are high-volume, with a median of 67 and a maximum of 4,752, suggesting frequent prosecutions.

```
Summary for: public_order_conviction_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    2.0    44.0    67.0   173.5   104.0  4752.0
```

Public Order Failed Count - The median is 10, with a mean of 31.2, and some months exceed 800, indicating occasional difficulty in securing convictions.

```
Summary for: public_order_failed_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    6.00   10.00   31.22   18.00  801.00
```

Other Offence Conviction Count - Monthly counts vary from 0 to 3,291, with a median of 21, reflecting a mix of lower-frequency offences grouped together.

```
Summary for: other_offence_conviction_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   11.00   21.00   80.87   50.00 3291.00
```

Other Offence Failed Count - Most records report low failure counts (median = 4), though high outliers up to 603 suggest instability in certain offence groups.

```
Summary for: other_offence_failed_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    1.00    4.00   15.69   10.00  603.00
```

Motoring Conviction Count - One of the highest-volume variables, with a median of 150 and a maximum of 12,945, indicating consistently large numbers of motoring cases.

```
Summary for: motoring_conviction_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1      97     150     385     222   12945
```

Motoring Failed Count - Failure counts also run high, with a median of 21 and peaks up to 1,725, showing a relatively high failure burden.

```
Summary for: motoring_failed_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   12.00   21.00   64.48   37.00 1725.00
```

Admin Failed Count - Administrative failures have a median of 12 and a mean of 37.1, with peaks reaching 1,051, indicating inconsistencies in administrative resolution.

```
Summary for: admin_failed_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    7.00   12.00   37.14   20.00 1051.00
```

Year - Covers a three-year period from 2014 to 2016, with an even distribution across all years.

```
Summary for: year
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2014    2014    2015    2015    2016    2016
```

Total Convictions - Total convictions vary widely, with a median of 768 and a maximum of 52,683, highlighting large discrepancies across regions and time

```
Summary for: total_convictions
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     70     491     768    1984    1194   52683
```

Total Unsuccessful - Monthly unsuccessful outcomes have a median of 136, reaching up to 9,939, pointing to high but fluctuating rates of failed prosecutions.

```
Summary for: total_unsuccessful
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   25.0    83.0   136.0   402.5   229.0  9939.0
```

Overall Total Cases - Total monthly cases range from 112 to 61,984, with a median of 908, showing the heavy caseload handled by CPS regions.

```
Summary for: overall_total_cases
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    112     571     908    2386    1432   61984
```

Overall Success Rate - Success rates remain high overall, ranging from 62.5% to 94.16%, with a median of 84.55%, reflecting strong prosecutorial performance across regions.

```
Summary for: overall_success_rate
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  62.50   82.52   84.55   84.49   86.72   94.16
> 
```

```
#Numerical Feature Distribution and Central Tendencies.
# List of all your numerical column names
numerical_cols <- c(
  "homicide_conviction_count", "homicide_failed_count",
  "person_offence_conviction_count", "person_offence_failed_count
  "sexual_conviction_count", "sexual_failed_count",
  "burglary_conviction_count", "burglary_failed_count",
  "robbery_conviction_count", "robbery_failed_count",
  "theft_conviction_count", "theft_failed_count",
  "fraud_conviction_count", "fraud_failed_count",
  "damage_conviction_count", "damage_failed_count",
  "drugs_conviction_count", "drugs_failed_count",
  "public_order_conviction_count", "public_order_failed_count",
  "other_offence_conviction_count", "other_offence_failed_count",
  "motoring_conviction_count", "motoring_failed_count",
  "admin_failed_count", "year",
  "total_convictions", "total_unsuccessful",
  "overall_total_cases", "overall_success_rate"
)

# Loop through and print summary for each
for (col in numerical_cols) {
  cat("\n\n Summary for:", col, "\n")
  print(summary(CPS_combined_data[[col]]))
```

3.2.2 Uniqueness Analysis of Categorical Variables

I used the n_distinct() function to count the number of unique values in each categorical column. CPS_region includes 43 regions, month covers all 12 months, and year_month captures 33 time points, confirming full temporal coverage. The case_volume_tier variable contains 3 levels, reflecting caseload distribution.

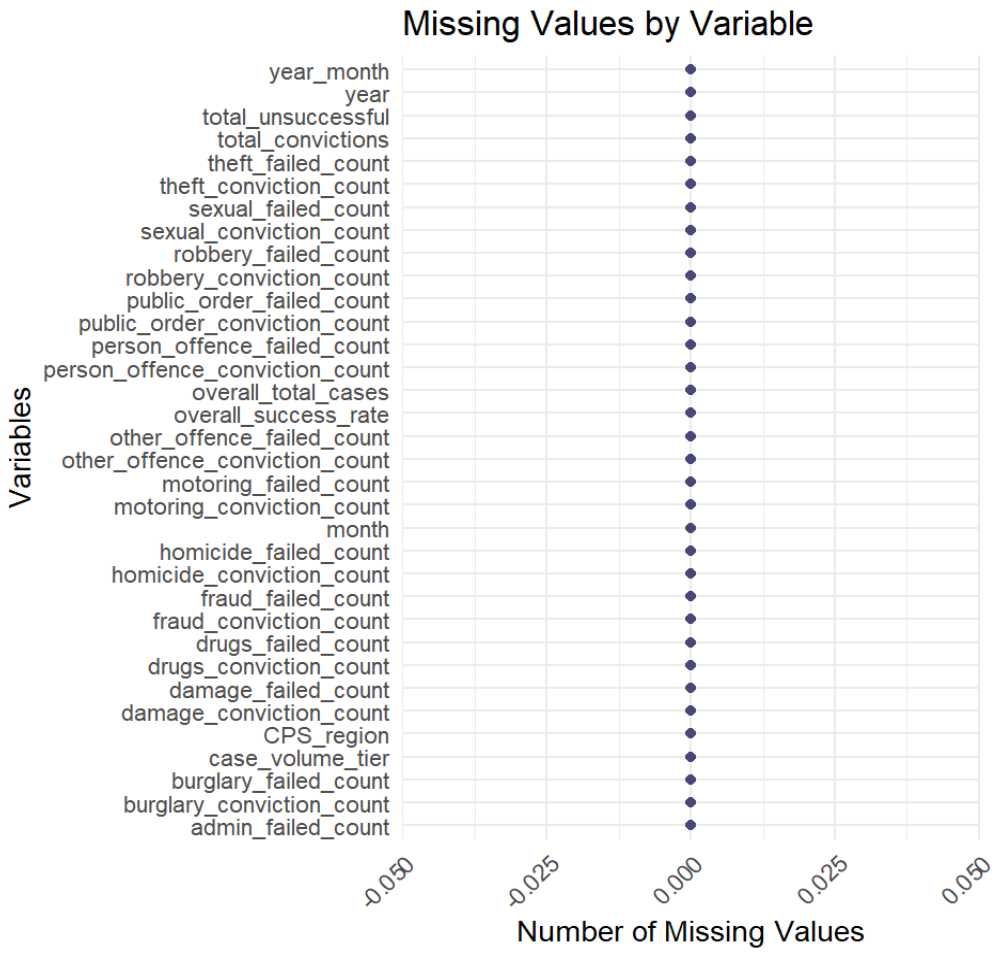
```
      CPS_region             month        year_month case_volume_tier
              43                12                33                 3
> 
```

```
# Define categorical columns
categorical_columns <- c("CPS_region", "month", "year_month", "ca

# Print unique value count for each
sapply(CPS_combined_data[categorical_columns], n_distinct)
```

3.3 Missing value in CPS

This plot, generated using the gg_miss_var() function from the naniar package, shows the number of missing values for each variable in the dataset. All variables have zero missing values, confirming the dataset is complete. This reflects the success of earlier data cleaning steps, and no further imputation or exclusion of records is required.




```
#missing value Analysis
library(naniar)
library(ggplot2)

gg_miss_var(CPS_combined_data) +
  labs(title = "Missing Values by Variable",
       x = "Variables",
       y = "Number of Missing Values") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

3.4 Correlation Analysis

A Pearson correlation matrix was created and visualized using the corrplot program in R to examine correlations among numerical variables. The matrix illustrates the strength and direction of linear correlations, supporting the detection of multicollinearity and informing model development.

As anticipated, strong positive correlations were observed among total_convictions, overall_total_cases, and overall_success_rate, reflecting their underlying mathematical relationship. Moderate correlations appeared among offence-specific counts such as theft, fraud, and motoring, indicating consistent volume patterns. Conversely, weaker associations were found in lower-frequency categories like homicide and administrative outcomes, suggesting more independent behavior. This analysis supports efficient feature selection and offers early insight into variable groupings and redundancy, which are essential for predictive modelling.

```r
# Correlation Analysis
library(corrplot)

full_corr <- cor(select(CPS_combined_data, where(is.numeric)), us

corrplot(
  full_corr,
  method = "color",
  type = "lower",
  order = "hclust",          # cluster similar variables together
  tl.cex = 0.4,              # tiny text
  col = colorRampPalette(c("#B2182B","white","#2166AC"))(200),
  tl.col = "black",
  addgrid.col = "gray90",
  title = "CPS Correlation Matrix",
  mar = c(0,0,1,0)
)
```

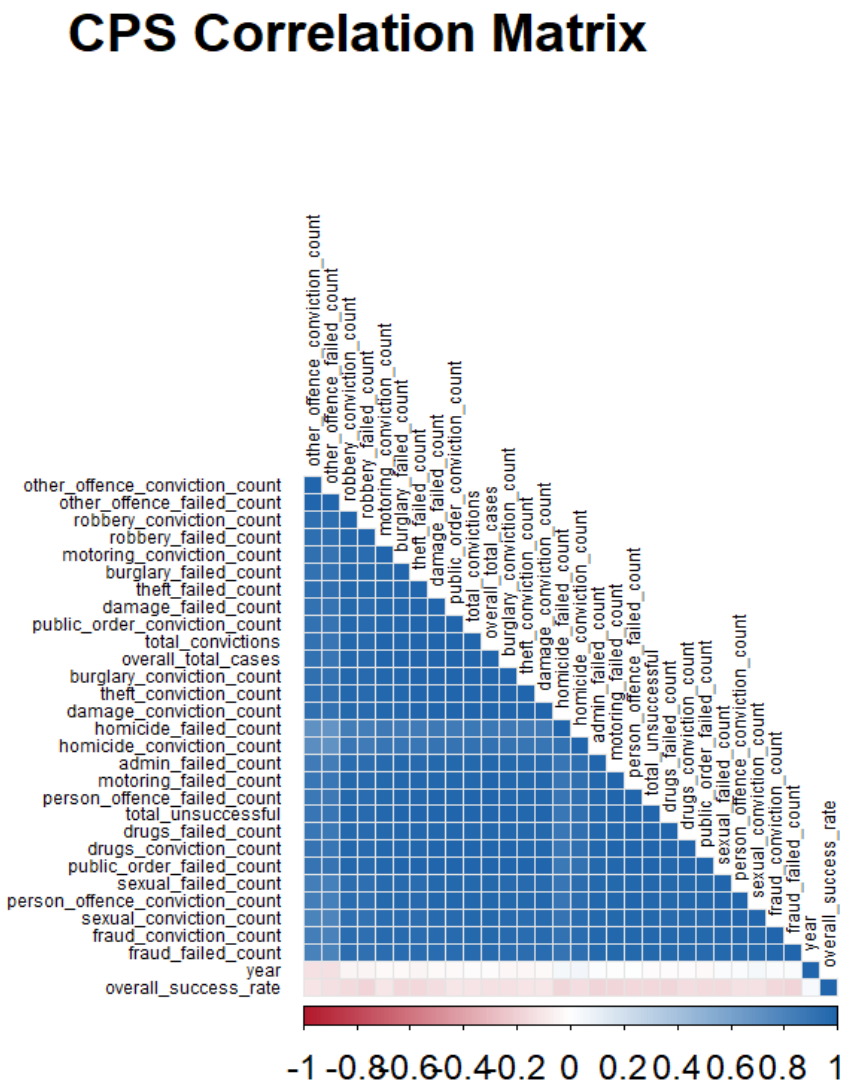


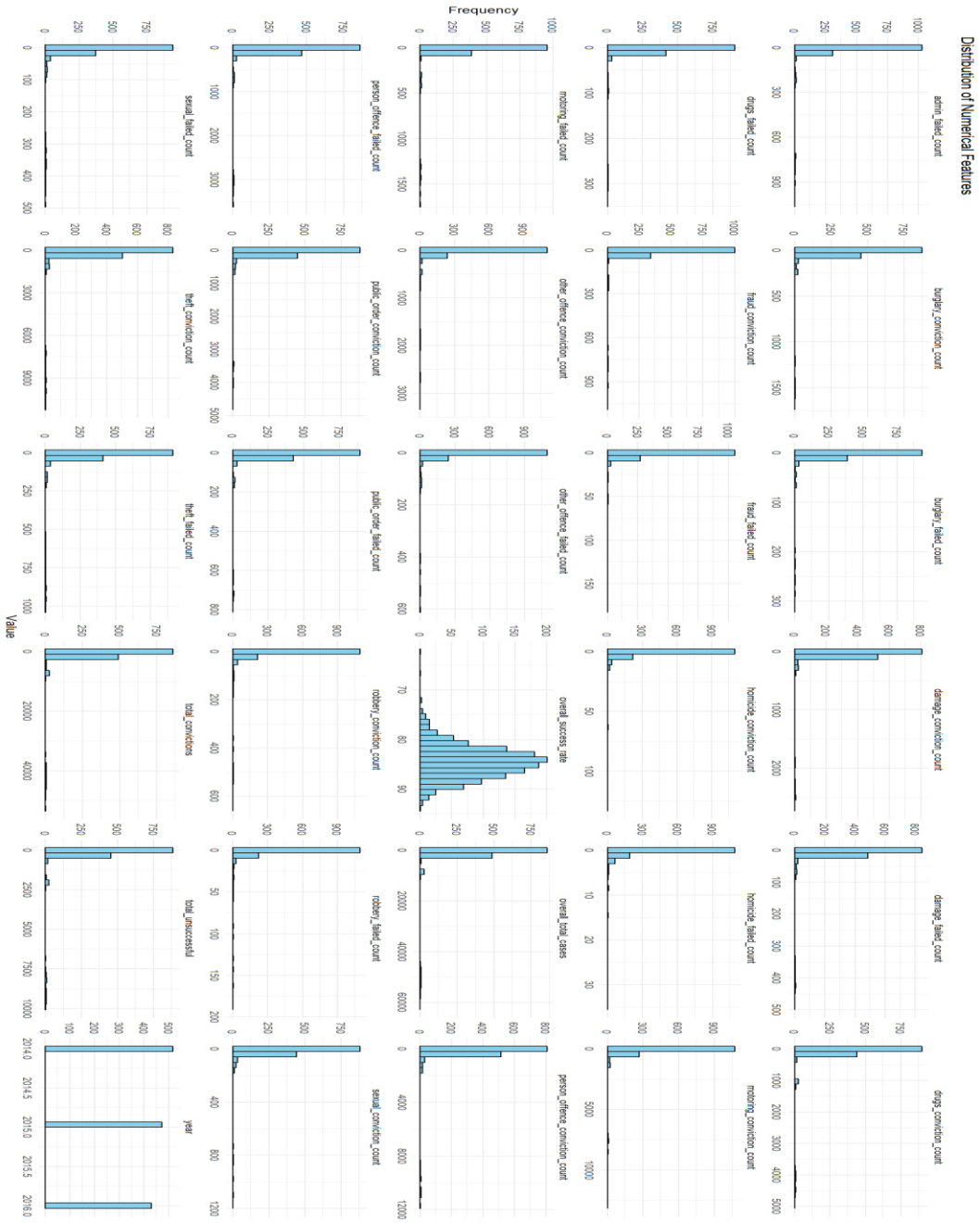
Fig 3.2.– Correlation analysis

3.5 Distribution of Numerical Features

The plot figure illustrates the distribution of numerical features using facet-wrapped histograms. Most variables — such as offence counts and unsuccessful outcomes — are highly right-skewed, indicating that most values fall within the lower range, while a smaller

number of observations report significantly higher counts. This is typical in justice-related data, where some regions or categories account for a disproportionately large share of cases. Features like number_of_fraud_and_forgery_unsuccessful, number_of_burglary_convictions, and number_of_motoring_offences display this trend clearly in their compressed left-hand bins.

In contrast, the overall_success_rate variable stands out with a bell-shaped, approximately normal distribution, centered around the mid-range. This suggests a more even performance distribution across regions and offence types. Percentage-based features also show clustering near extreme values (0% or 100%), highlighting limited variability and potential redundancy for modeling. These observations provide valuable guidance for preprocessing steps such as scaling, outlier detection, and feature selection, ensuring the data is suitable for reliable predictive modeling.



```r

# Select all numeric columns
numeric_data <- select(CPS_combined_data, where(is.numeric))

# Convert to long format for faceted plotting
numeric_long <- numeric_data %>%
  pivot_longer(cols = everything(), names_to = "feature", values_

# Plot histograms for each numeric feature
ggplot(numeric_long, aes(x = value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~ feature, scales = "free", ncol = 3) +
  labs(
    title = "Distribution of Numerical Features",
    x = "Value",
    y = "Frequency"
  ) +
  theme_minimal() +
  theme(
    strip.text = element_text(size = 8),
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text.x = element_text(size = 6),
    axis.text.y = element_text(size = 6)
  )
```

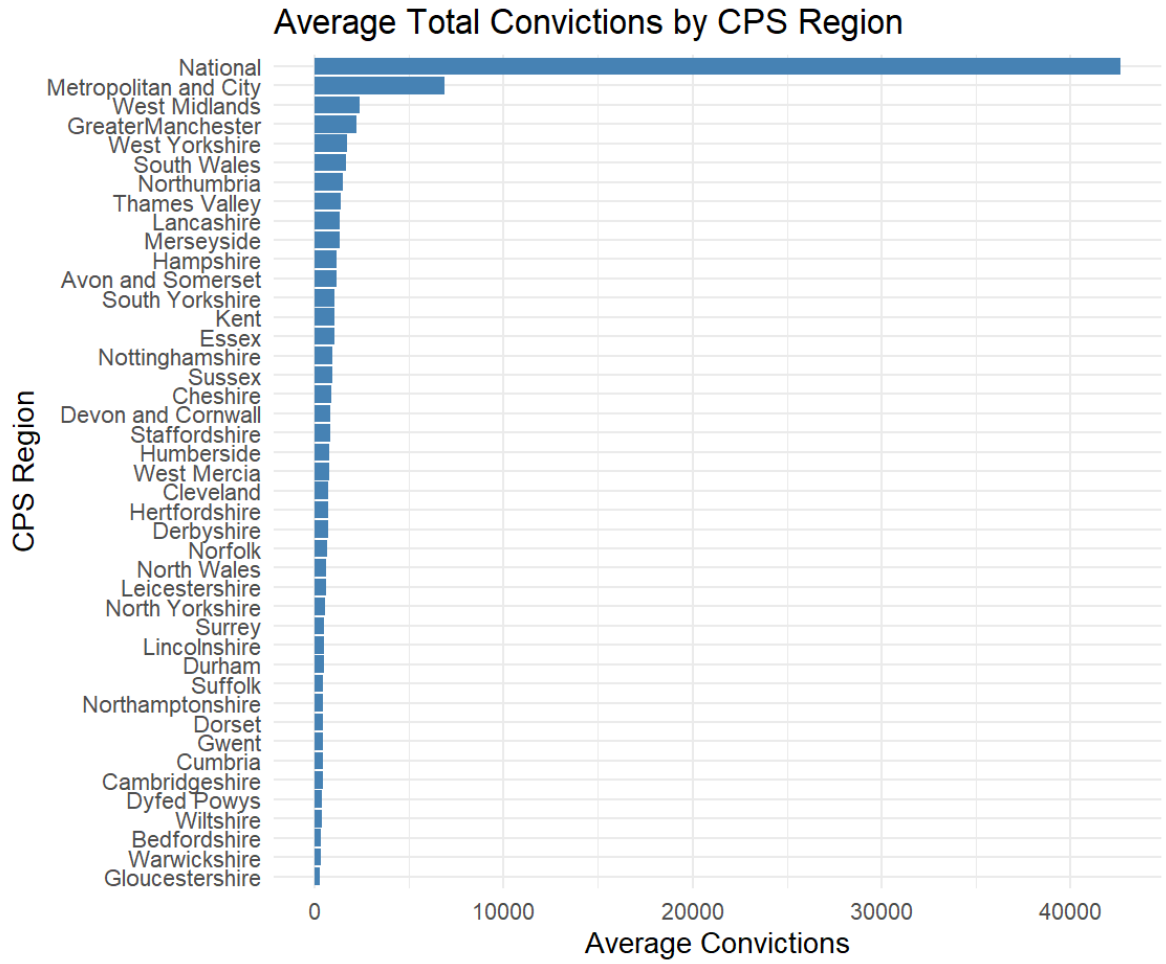
3.6 Conviction Outcome Analysis (Successful Cases)

Distribution of Total Convictions

A horizontal bar chart was used to visualise the average number of total convictions across CPS regions, providing insights into regional conviction activity. The chart reveals substantial variation in prosecution volumes across the country.

As expected, the 'National' aggregate records the highest figure, as it encompasses all cases. Among individual regions, Metropolitan and City, West Midlands, and Greater Manchester reported the highest average conviction counts. This pattern likely reflects higher population density, greater case loads, and increased court throughput in these urban centres.

Conversely, regions such as Gloucestershire, Warwickshire, and Bedfordshire exhibited the lowest average conviction counts, which may be attributed to smaller jurisdictions or lower levels of reported crime.

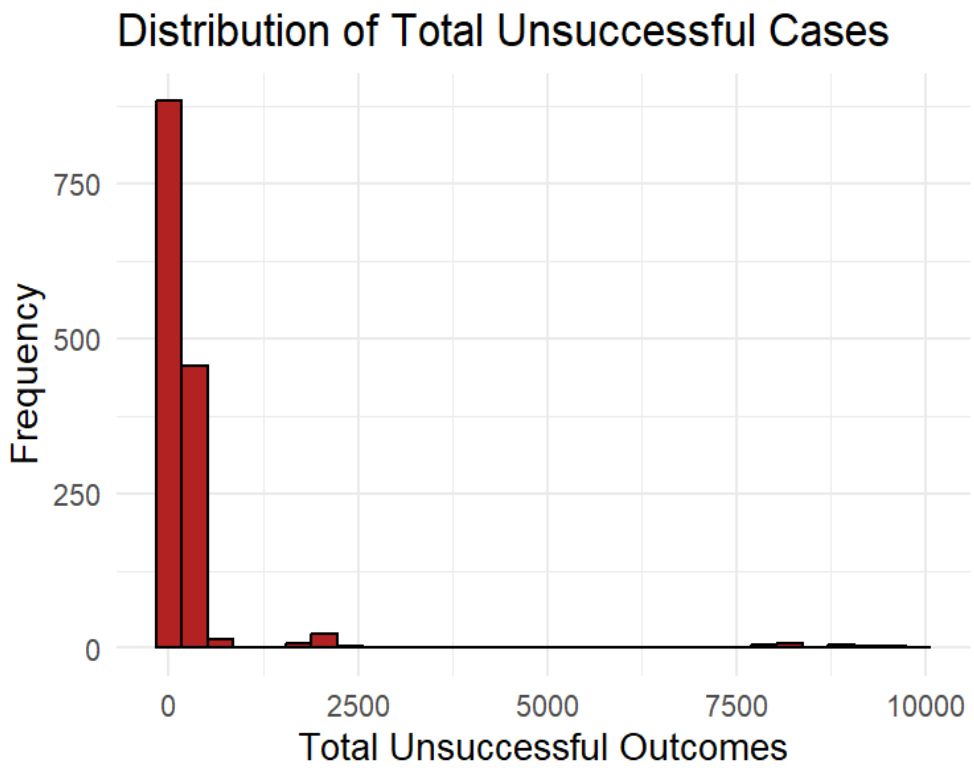



```r
# Average total convictions by region
CPS_combined_data %>%
  group_by(CPS_region) %>%
  summarise(avg_convictions = mean(total_convictions, na.rm = TRU
  ggplot(aes(x = reorder(CPS_region, avg_convictions), y = avg_co
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    title = "Average Total Convictions by CPS Region",
    x = "CPS Region",
    y = "Average Convictions"
  ) +
  theme_minimal()
```

3.7 Distribution of Total Unsuccessful Cases

A histogram of the total_unsuccessful variable was generated to examine the distribution of unsuccessful prosecutions. The resulting distribution is heavily right-skewed, indicating that most CPS entries reported relatively few failed outcomes, while a smaller number experienced significantly higher failure counts.

This skew may reflect challenges such as complex case types, legal disputes, or resource limitations in certain regions. These disparities suggest potential regional performance issues and may point to the need for targeted operational reviews or reallocation of support resources.

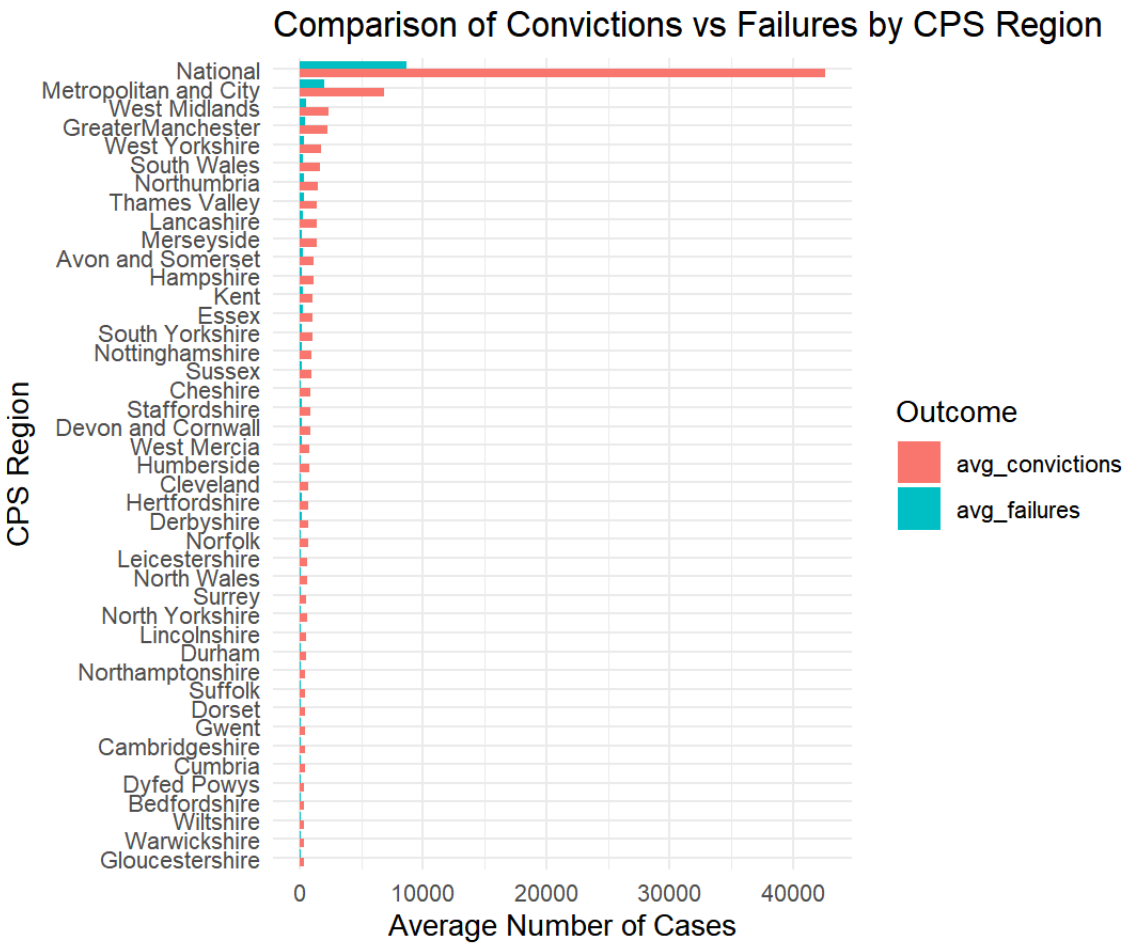



```
332
333 # Histogram of total unsuccessful outcomes
334 ggplot(CPS_combined_data, aes(x = total_unsuccessful)) +
335   geom_histogram(bins = 30, fill = "firebrick", color = "black") +
336   labs(
337     title = "Distribution of Total Unsuccessful Cases",
338     x = "Total Unsuccessful Outcomes",
339     y = "Frequency"
340   ) +
341   theme_minimal()
342
```

3.8 Comparison of Convictions vs Failures by CPS Region

A grouped bar chart was used to compare the average number of convictions and failures across CPS regions. The chart highlights that regions such as Metropolitan and City, West Midlands, and Greater Manchester reported the highest case volumes for both outcomes.

In all regions, convictions consistently outpaced failures, suggesting prosecutorial success is dominant, even in high-pressure jurisdictions.



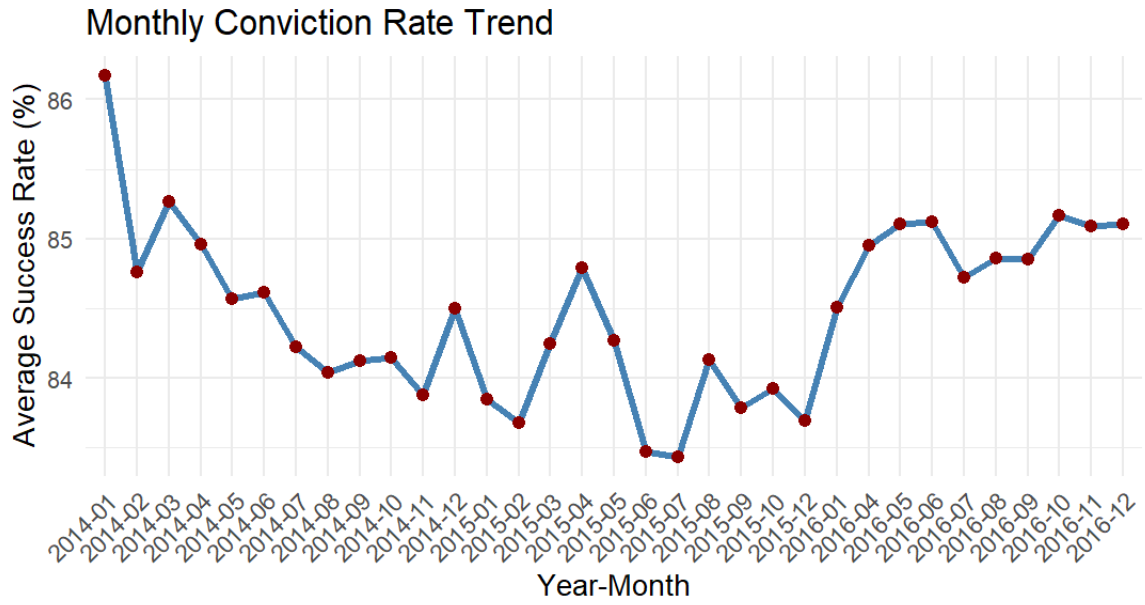

```
341  # Compare(successful VS unsuccessful)
342  CPS_combined_data %>%
343    group_by(CPS_region) %>%
344    summarise(
345      avg_convictions = mean(total_convictions, na.rm = TRUE),
346      avg_failures = mean(total_unsuccessful, na.rm = TRUE)
347    ) %>%
348    pivot_longer(cols = c(avg_convictions, avg_failures), names_to
349    ggplot(aes(x = reorder(CPS_region, Average), y = Average, fill
350    geom_col(position = "dodge") +
351    coord_flip() +
352    labs(
353      title = "Comparison of Convictions vs Failures by CPS Region"
354      x = "CPS Region",
355      y = "Average Number of Cases"
356    ) +
357    theme_minimal()
358
```

3.9 Temporal Trend Analysis

3.9.1 Monthly conviction success cases rate trend

This line graph depicts the average monthly conviction success rate between 2014 and 2016, fluctuating between 83% and 86%. A notable dip was observed during mid-2015, followed by a gradual increase throughout 2016.

The plot was generated using R's geom_line() and geom_point() functions, effectively highlighting both overall trends and specific monthly variations. This visualization enables the identification of temporal inconsistencies in prosecution outcomes and supports deeper investigation into underperforming periods. These findings are essential for performance monitoring, resource allocation, and promoting consistency in judicial outcomes over time.

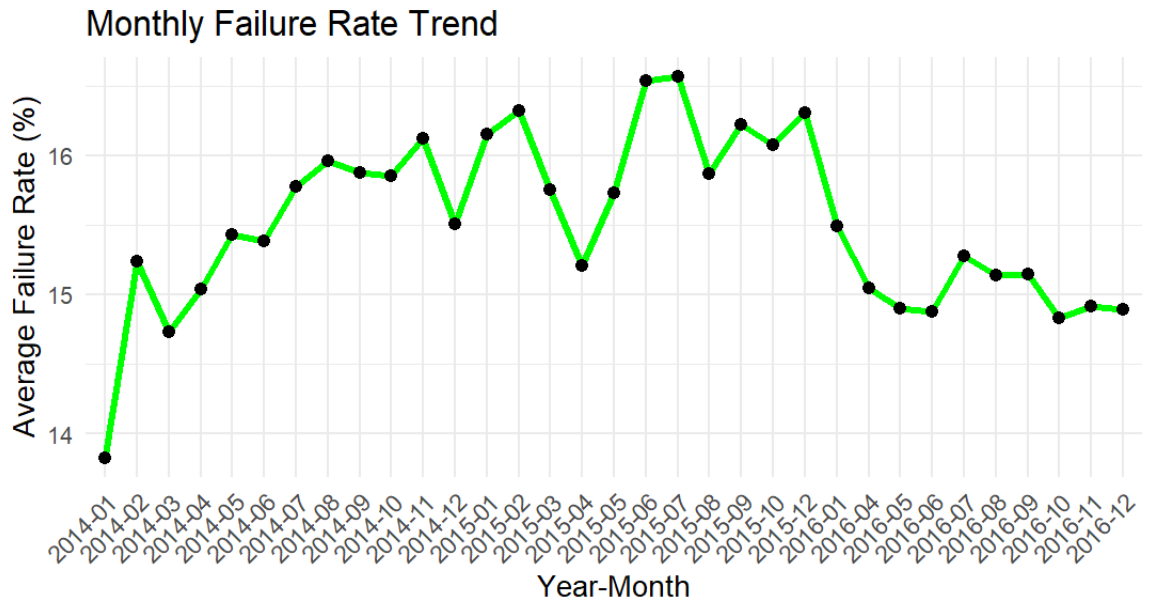



```
360
361 # Monthly conviction success rate trend
362 CPS_combined_data %>%
363   group_by(year_month) %>%
364   summarise(avg_success_rate = mean(overall_success_rate, na.rm =
365   ggplot(aes(x = year_month, y = avg_success_rate, group = 1)) +
366   geom_line(color = "steelblue", size = 1.2) +
367   geom_point(color = "darkred", size = 2) +
368   labs(
369     title = "Monthly Conviction Rate Trend",
370     x = "Year-Month",
371     y = "Average Success Rate (%)"
372   ) +
373   theme_minimal() +
374   theme(axis.text.x = element_text(angle = 45, hjust = 1))
375
```

3.9.2 Monthly failure rate trend

This line chart illustrates the average monthly prosecution failure rate across the dataset timeline (2014–2016). The failure rate fluctuated between 14% and 17%, with a noticeable peak in mid-2015, followed by a gradual downward trend throughout 2016.

The visualization was produced using R's geom_line() and geom_point() functions. The elevated rates in 2015 suggest possible issues such as complex case types, resource constraints, or procedural inefficiencies. The subsequent decline in 2016 may reflect improvements in prosecution processes or strategic interventions.

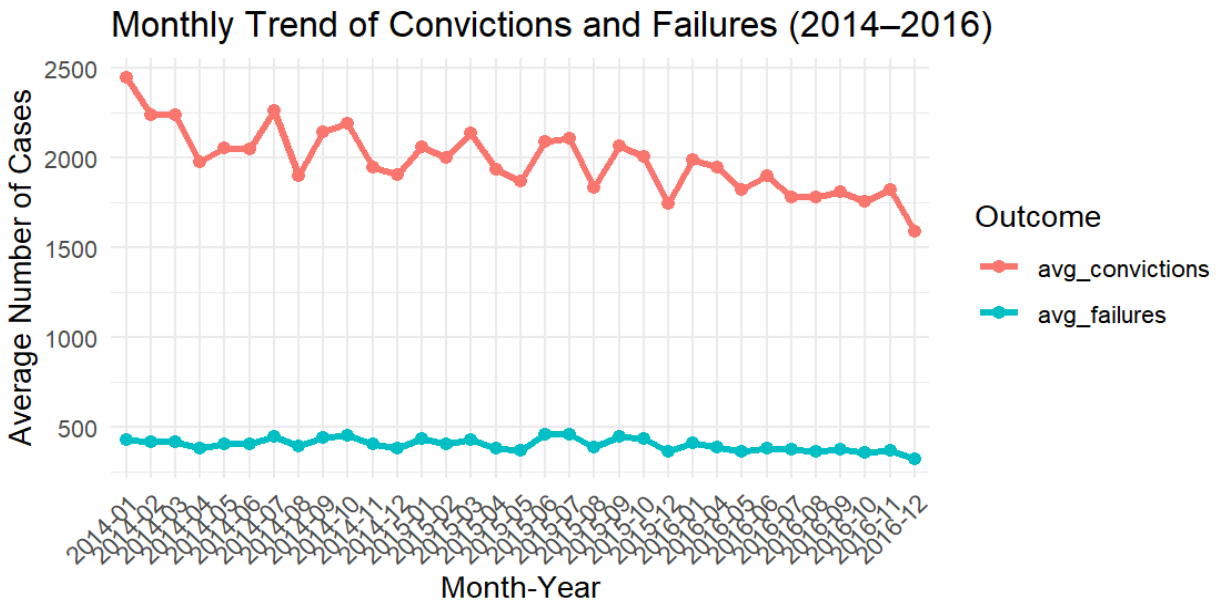



```
376 # Monthly failure rate trend
377 CPS_combined_data %>%
378   group_by(year_month) %>%
379   summarise(avg_failure_rate = mean(100 - overall_success_rate, n
380   ggplot(aes(x = year_month, y = avg_failure_rate, group = 1)) +
381   geom_line(color = "green", size = 1.2) +
382   geom_point(color = "black", size = 2) +
383   labs(
384     title = "Monthly Failure Rate Trend",
385     x = "Year-Month",
386     y = "Average Failure Rate (%)"
387   ) +
388   theme_minimal() +
389   theme(axis.text.x = element_text(angle = 45, hjust = 1))
390
```

3.9.3 Comparison of total convictions and failures over time

This line chart compares the monthly average of total convictions and unsuccessful outcomes across the CPS dataset from 2014 to 2016. The geom_line() and geom_point() functions were used to visualize both trends simultaneously.

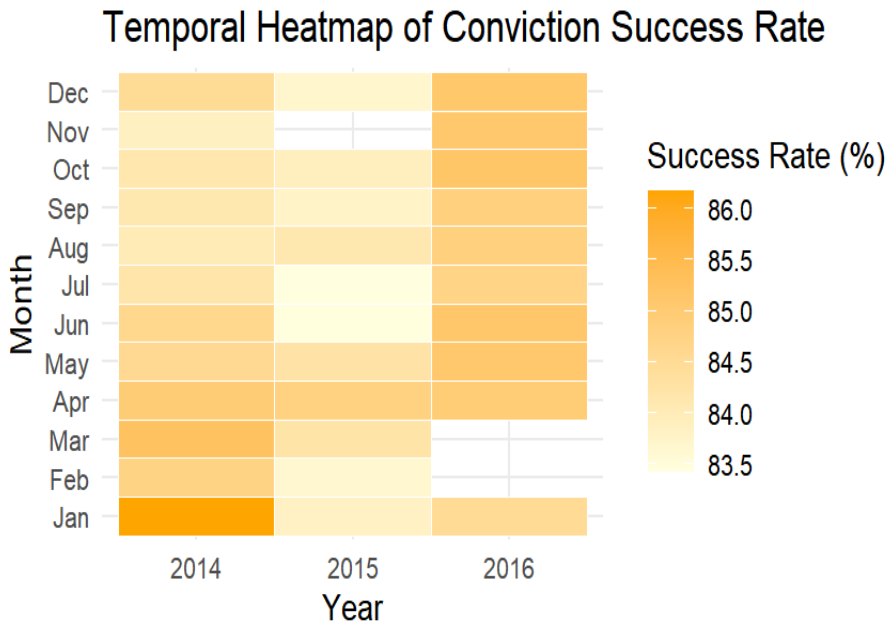
The graph reveals a consistently higher volume of convictions compared to failures throughout the timeline. Conviction counts show modest fluctuation but an overall slight decline toward late 2016. Failure rates remain relatively stable, with occasional spikes likely linked to case complexity or administrative delays.




```
390
391 # Monthly trend: convictions and failures
392 CPS_combined_data %>%
393   group_by(year_month) %>%
394   summarise(
395     avg_convictions = mean(total_convictions, na.rm = TRUE),
396     avg_failures = mean(total_unsuccessful, na.rm = TRUE)
397   ) %>%
398   pivot_longer(cols = c(avg_convictions, avg_failures), names_to
399   ggplot(aes(x = year_month, y = Count, color = Outcome, group =
400   geom_line(size = 1.2) +
401   geom_point(size = 2) +
402   labs(
403     title = "Monthly Trend of Convictions and Failures (2014-2016
404     x = "Month-Year",
405     y = "Average Number of Cases"
406   ) +
407   theme_minimal() +
408   theme(axis.text.x = element_text(angle = 45, hjust = 1))
409
```

3.9.4 Temporal Heatmap of Conviction Rate

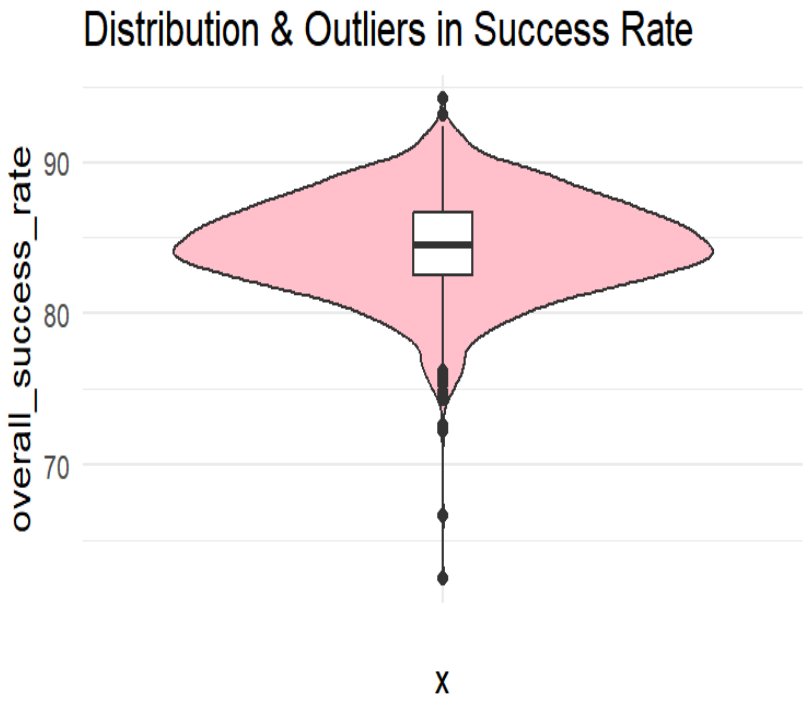
This heatmap visualizes average monthly conviction success rates from 2014 to 2016. Darker shades indicate higher success, while white gaps represent missing data. These gaps likely result from unavailable or unreported monthly records in the original dataset. A decline is seen in mid-2015, with improvement throughout 2016. The chart, created using geom_tile() in R, highlights seasonal or operational shifts that may affect prosecution outcomes.




```
409
410  # Temporal Heatmap (Month x Year)
411  CPS_combined_data %>%
412    group_by(year, month) %>%
413    summarise(avg_success_rate = mean(overall_success_rate, na.rm =
414    ggplot(aes(x = factor(year), y = month, fill = avg_success_rate
415    geom_tile(color = "white") +
416    scale_fill_gradient(low = "lightyellow", high = "orange") +
417    labs(
418      title = "Temporal Heatmap of Conviction Success Rate",
419      x = "Year",
420      y = "Month",
421      fill = "Success Rate (%)"
422    ) +
423    theme_minimal()
424
```

3.10 Outlier Detection in Success Rate

This violin plot shows the distribution of overall conviction success rates, using a combination of a boxplot and a density curve to highlight both the spread and the central tendency of the data. Most values cluster around the mid-80% range, but several outliers drop below 75%, suggesting months with unusually low conviction rates.

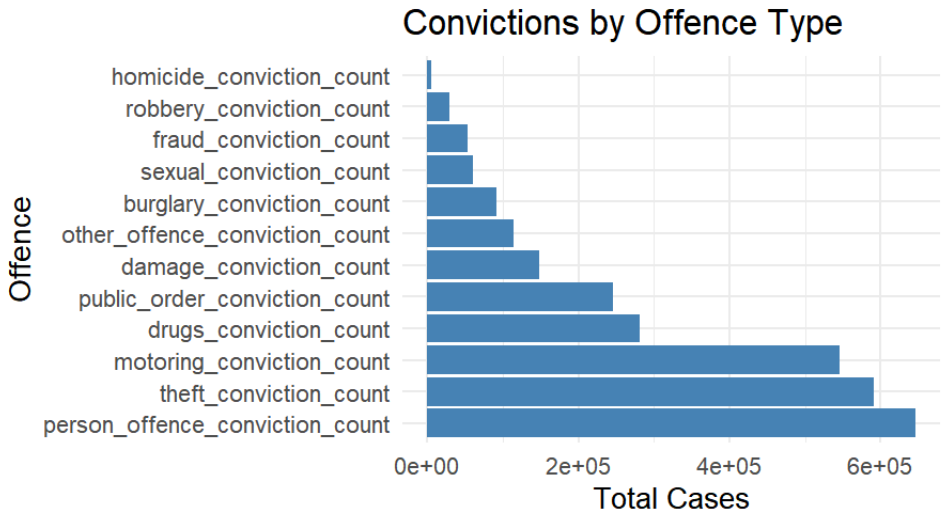



```
424
425  #outlier
426  ggplot(CPS_combined_data, aes(x = "", y = overall_success_rate))
427    geom_violin(fill = "pink") +
428    geom_boxplot(width = 0.1, fill = "white") +
429    labs(title = "Distribution & Outliers in Success Rate") +
430    theme_minimal()
431
```

3.11 Conviction (Successful) by Offence Type

This bar chart shows the total number of convictions by offence type across all regions and months. Offenses related to personal harm (e.g., person_offence_conviction_count) and theft were the most frequently prosecuted, while serious crimes like homicide and robbery had comparatively lower volumes.

This analysis complements the earlier regional summary by highlighting which types of crimes dominate the overall prosecution workload, regardless of location. The chart was created using geom_col() and coord_flip() for readability, helping to prioritize areas for resource allocation or policy focus.

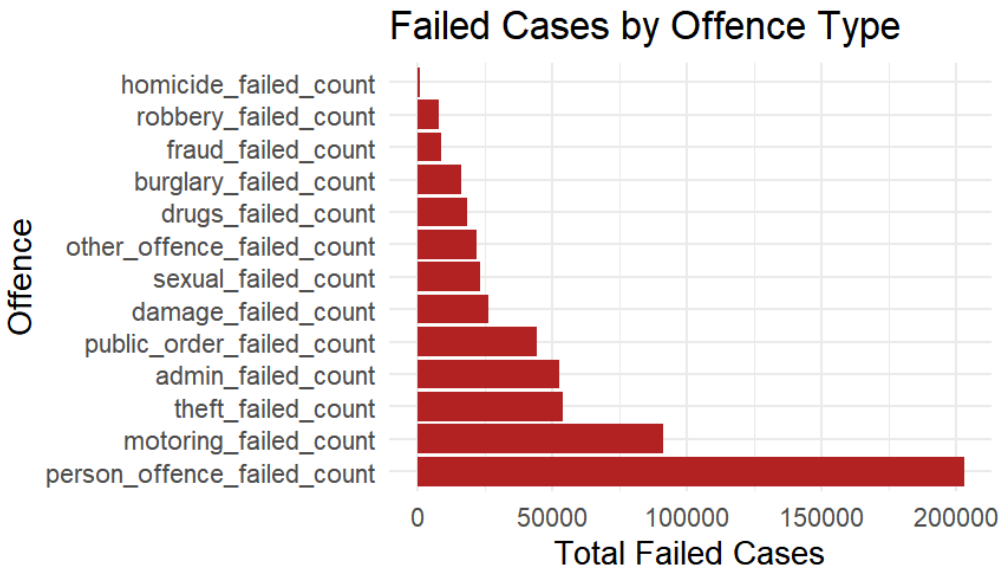



```
431
432  # Convictions by offence Type
433  offence_success <- CPS_combined_data %>%
434    select(ends_with("conviction_count")) %>%
435    summarise(across(everything(), sum, na.rm = TRUE)) %>%
436    pivot_longer(everything(), names_to = "offence", values_to = "t
437
438  ggplot(offence_success, aes(x = reorder(offence, -total), y = tot
439    geom_col(fill = "steelblue") +
440    coord_flip() +
441    labs(title = "Convictions by Offence Type", x = "Offence", y =
442    theme_minimal()
443
```

3.12 Failed (Unsuccessful) by Offense Type

This bar chart displays the total number of failed prosecution outcomes by offence type. person_offence_failed_count and motoring_failed_count stand out as the categories with the highest number of failed cases, while serious offences like homicide and robbery show relatively few failures.

This analysis complements the conviction chart by highlighting where prosecution efforts are less successful. It can inform areas where case handling, evidence, or legal strategies may need improvement. The plot was generated using geom_col() in R and is ordered by total failures for better visual comparison.




```
443
444  # failed cases by offence type
445  offence_failed <- CPS_combined_data %>%
446    select(ends_with("failed_count")) %>%
447    summarise(across(everything(), sum, na.rm = TRUE)) %>%
448    pivot_longer(everything(), names_to = "offence", values_to = "t
449
450  # Plot
451  ggplot(offence_failed, aes(x = reorder(offence, -total), y = tota
452    geom_col(fill = "firebrick") +
453    coord_flip() +
454    labs(
455      title = "Failed Cases by Offence Type",
456      x = "Offence",
457      y = "Total Failed Cases"
458    ) +
459    theme_minimal()
460
```

```
207
208  # Task 2-Descriptive Analysis
209  library(ggplot2)
210  library(dplyr)
211  library(naniar)
212
273
274  # Correlation Analysis
275  library(corrplot)
276
291
292  #Distributaion of Numerical Columns
293  library(tidyr)
294
```

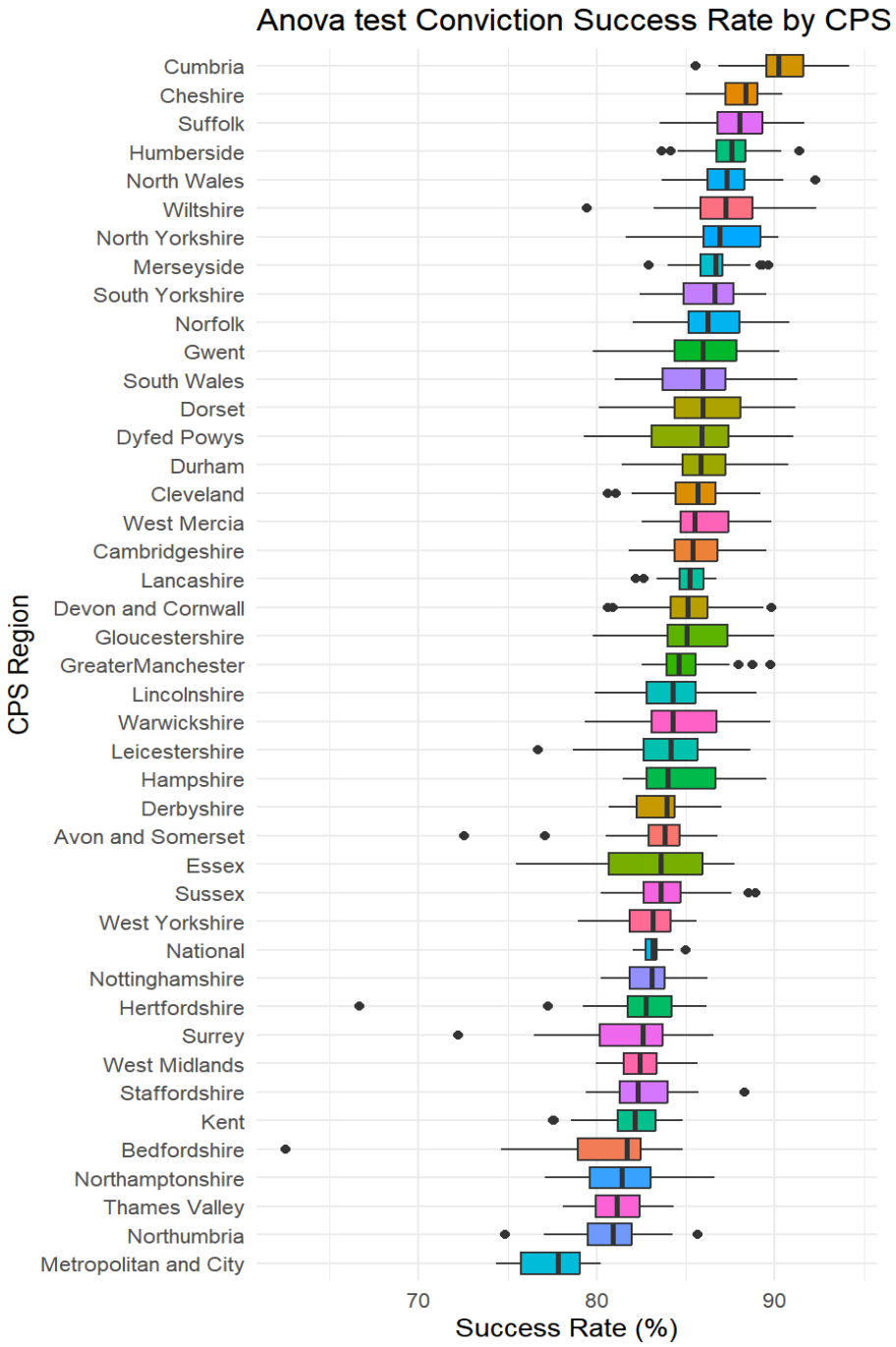
Libraries used for descriptive analysis

Chapter 4- Hypothesis Testing

4.1 ANOVA Test

ANOVA is ideal for comparing means across multiple groups. It was used here to test if differences in conviction rates across regions are statistically significant, supporting the hypothesis that regional performance varies due to workload or operational factors *(*Field, 2013)

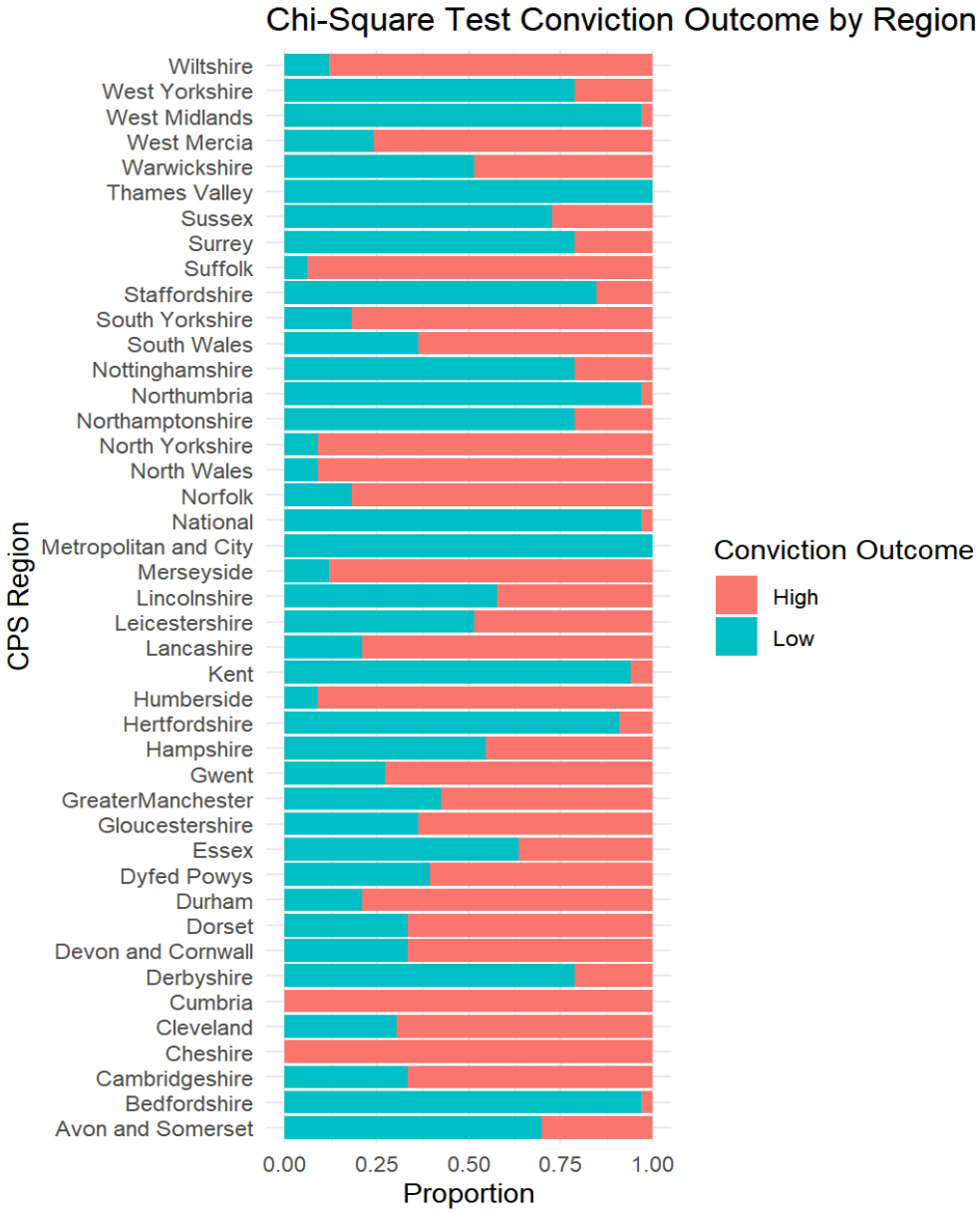
.



This boxplot, based on ANOVA analysis, compares conviction success rates across CPS regions. It highlights clear regional differences, with some regions like Cumbria showing consistently higher success rates, while others like Metropolitan and City show lower and more variable outcomes.

4.2 chi-square Test

Chi-Square Test was used to examine whether the distribution of conviction outcomes (High/Low) is related to the CPS region(Pearson, 1900).



Chapter 5 – Predictive Analysis

5.1 Regression Model Analysis

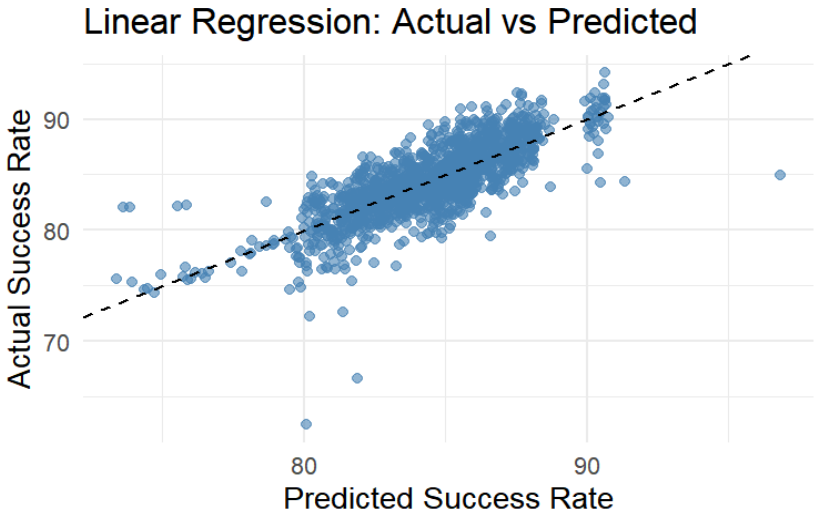
Hypothesis H1 - An increase in the overall number of cases handled by CPS regions is significantly associated with a decrease in conviction success rate.

Strategic Importance – CPS regions that handle a high number of cases may face capacity constraints such as limited staff, time, and administrative resources, which could reduce the effectiveness of prosecutions (James et al., 2013). By modeling this relationship, the analysis aims to assess whether heavier workloads contribute to lower conviction outcomes across regions.

To examine this, Linear Regression, Ridge Regression, and Lasso Regression were employed, based on their theoretical strengths in statistical modeling. Linear Regression is widely used for its interpretability and ability to show basic correlations (James et al., 2013). Ridge Regression mitigates multicollinearity by penalizing large coefficients (Hoerl & Kennard, 1970), while Lasso Regression improves model simplicity by reducing the influence of less relevant variables (Tibshirani, 1996). Features such as overall_total_cases, total_convictions, total_unsuccessful, CPS_region, year, and month were selected for their relevance to the workload-performance hypothesis and ability to reflect regional and temporal dynamics.

5.1.1 Linear Regression

Linear Regression is a simple method used to understand the relationship between one target variable and several predictors. In this analysis, it was used as a baseline to see how factors like offence counts, year, and region affect the conviction success rate.



The scatter plot above shows the projected and actual conviction success rates using the linear regression model. Each point represents a CPS region in a specific month. The dashed diagonal line represents a perfect prediction line (predicted = actual). The clustering of points along this line shows a strong linear relationship, implying that the model does a reasonable job of estimating success rates. A few deviations and outliers exist, maybe due to significant regional or monthly changes, but the overall trend supports the model's validity.

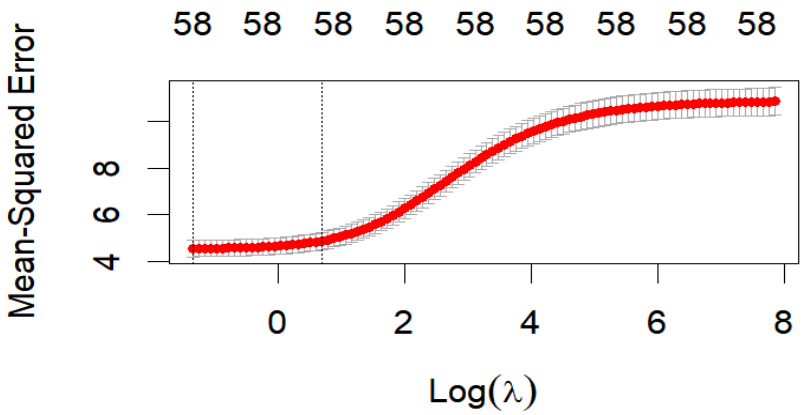
```
>
> cat("Linear Regression:\n")
Linear Regression:
> cat("  RMSE:", round(rmse(actual, regression_data$predicted_linear),
2), "\n")
  RMSE: 2.03
> cat("  MAE :", round(mae(actual, regression_data$predicted_linear),
2), "\n")
  MAE : 1.48
> cat("  R²  :", round(summary(success_rate_lm)$r.squared, 4), "\n\n")
  R²  : 0.6209
```

The model performance measures show that Linear Regression performed reasonably well. The Root Mean Square Error (RMSE) of 2.03 and Mean Absolute Error (MAE) of 1.48 indicate a low average prediction error for determining the success rate. Furthermore, the R-squared value (R^2) of 0.6209 indicates that the selected predictors account for approximately 62% of the variance in conviction success rates. While not perfect, the findings indicate a reasonably high link between CPS workload characteristics and prosecution performance.

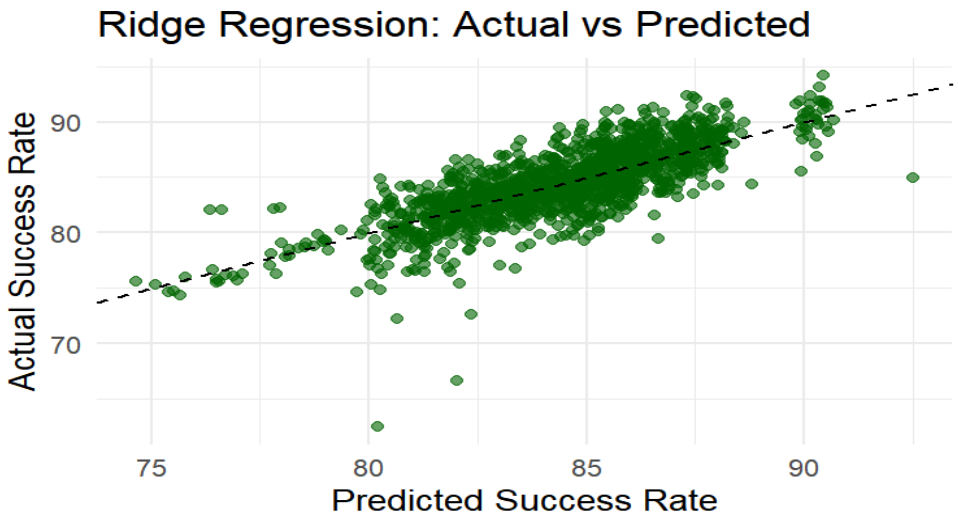
```
517
518 # 1.linear regression
519 # Fit linear regression model
520 success_rate_lm <- lm(overall_success_rate ~ ., data = regression
521 # View model summary
522 summary(success_rate_lm)
523 # Predict and store results
524 regression_data$predicted_linear <- predict(success_rate_lm)
525 #plot
526 ggplot(regression_data, aes(x = predicted_linear, y = overall_suc
527   geom_point(alpha = 0.6, color = "steelblue") +
528   geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
529   labs(
530     title = "Linear Regression: Actual vs Predicted",
531     x = "Predicted Success Rate",
532     y = "Actual Success Rate"
533   ) +
534   theme_minimal()
535
```

5.2.2 Ridge Model

Ridge Regression was used to improve model stability by handling multicollinearity, which happens when some features are strongly related to each other. Unlike regular linear regression, Ridge adds a penalty to large coefficients, helping reduce overfitting. It kept all the features in the model and produced predictions that were close to the actual values, making it a strong alternative to the baseline model. (Hoerl & Kennard (1970))



This plot shows how the mean squared error (MSE) changes across a range of regularization strengths (log(λ)). The vertical dashed line indicates the λ value with the lowest error, which was selected to optimize the model's performance by balancing complexity and accuracy.



This scatter plot compares the actual conviction success rates with the values predicted by the Ridge Regression model. The close clustering of points around the diagonal line suggests that the model accurately captured the underlying patterns in the data.

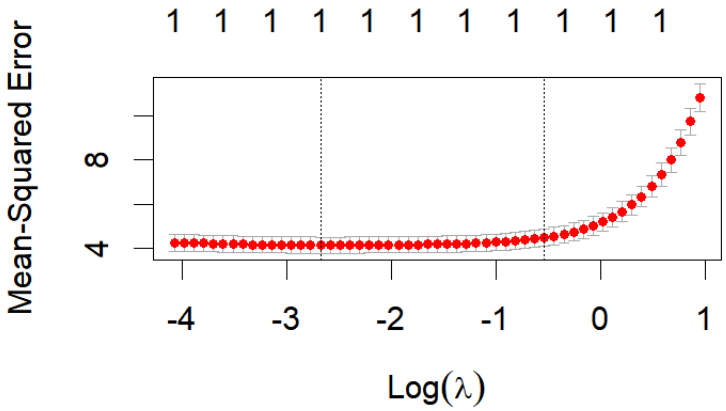
```
>
> cat("Ridge Regression:\n")
Ridge Regression:
> cat("  RMSE:", round(rmse(actual, regression_data$predicted_ridge),
2), "\n")
  RMSE: 2.05
> cat("  MAE :", round(mae(actual, regression_data$predicted_ridge), 2),
"\n\n")
  MAE : 1.52
```

The Ridge Regression model had an RMSE of 2.05 and an MAE of 1.52. These values show that, on average, the model's predictions depart from the actual conviction success rate by 1.52 percentage points, with slightly bigger errors penalized more heavily in the RMSE.

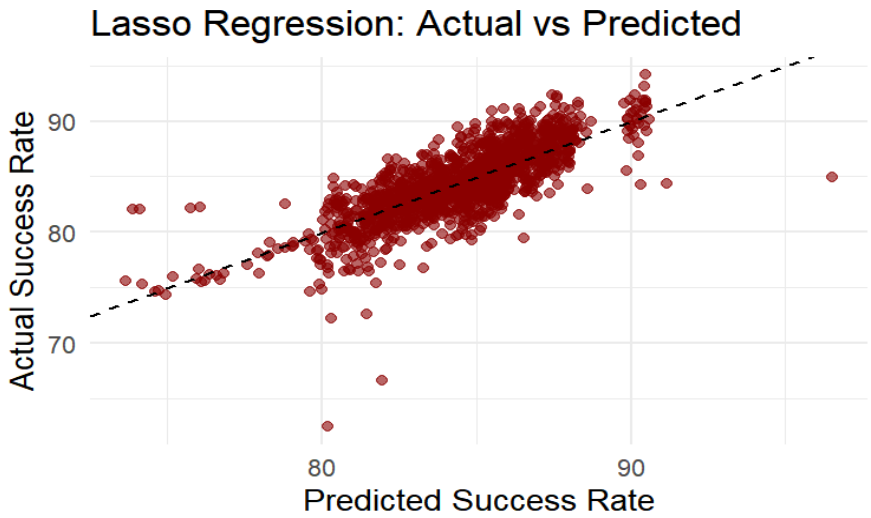
```
536 # 2.ridge model
537 # Prepare matrix input
538 x_matrix <- model.matrix(overall_success_rate ~ ., regression_dat
539 y_vector <- regression_data$overall_success_rate
540
541 # Fit Ridge model (alpha = 0)
542 ridge_success_model <- cv.glmnet(x_matrix, y_vector, alpha = 0)
543 plot(ridge_success_model)
544
545 # Best lambda
546 best_lambda_ridge <- ridge_success_model$lambda.min
547
548 # Predict
549 regression_data$predicted_ridge <- predict(ridge_success_model, s
550 #plot
551 ggplot(regression_data, aes(x = predicted_ridge, y = overall_succ
552   geom_point(alpha = 0.6, color = "darkgreen") +
553   geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
554   labs(
555     title = "Ridge Regression: Actual vs Predicted",
556     x = "Predicted Success Rate",
557     y = "Actual Success Rate"
558   ) +
559   theme_minimal()
560
```

5.2.3 Lasso Regression

Lasso Regression is a regularized version of linear regression that not only reduces overfitting but also performs feature selection. It works by shrinking the coefficients of less important variables to zero, effectively removing them from the model. This helps simplify the model and highlight the most relevant predictors. It is especially useful when working with many input features or when only a few are expected to be strong predictors. (Tibshirani (1996))



This plot shows how the Mean Squared Error (MSE) changes with different values of λ (lambda) during cross-validation. The model selects the lambda with the lowest error, balancing accuracy, and simplicity. Lasso applies stronger regularization as lambda increases, which can shrink less important features to zero—making the model simpler and focused on key predictors



This scatter plot compares the actual conviction success rates with the predicted values from the Lasso model. Most points lie close to the dashed line, indicating a strong fit. The dense clustering along the diagonal suggests that Lasso made consistent and accurate predictions, while still simplifying the model by reducing some coefficients to zero.

```
>
> cat("Lasso Regression:\n")
Lasso Regression:
> cat("  RMSE:", round(rmse(actual, regression_data$predicted_lasso),
2), "\n")
  RMSE: 2.03
> cat("  MAE :", round(mae(actual, regression_data$predicted_lasso), 2),
"\n")
  MAE : 1.49
>
>
```

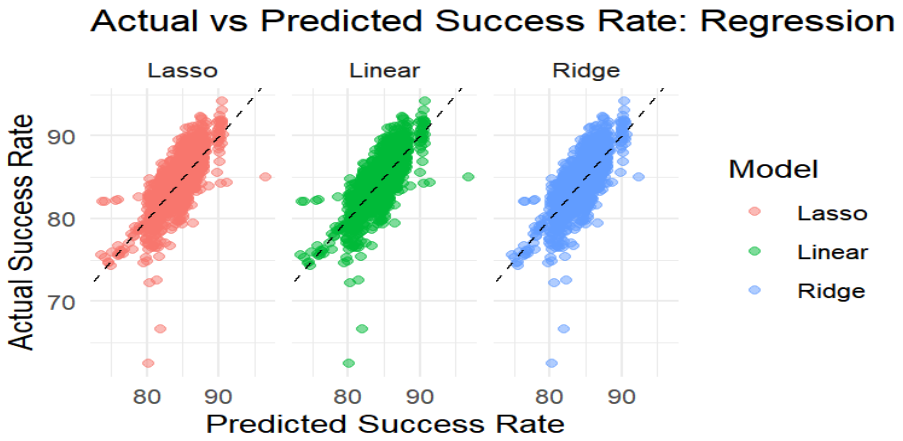
The Lasso Regression model produced an RMSE of 2.03 and an MAE of 1.49, indicating that the model's predictions are 1.49 percentage points off the actual conviction success rate. These results are essentially comparable to those of the linear regression model, however, Lasso provides a significant advantage: automatic feature selection.

```
#3.Lasso Regression
# Fit Lasso Regression
lasso_success_model <- cv.glmnet(x_matrix, y_vector, alpha = 1)
plot(lasso_success_model)

# Best lambda
best_lambda_lasso <- lasso_success_model$lambda.min

# Predict outcome
regression_data$predicted_lasso <- predict(lasso_success_model, s
#plot
ggplot(regression_data, aes(x = predicted_lasso, y = overall_succ
  geom_point(alpha = 0.6, color = "darkred") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  labs(
    title = "Lasso Regression: Actual vs Predicted",
    x = "Predicted Success Rate",
    y = "Actual Success Rate"
  ) +
  theme_minimal()
```

The comparison of Lasso, Linear, and Ridge regression models shows that each has different strengths in predicting the CPS conviction success rate. Lasso performed best overall, closely matching the actual values by automatically selecting only the most important features. Linear regression gave a decent baseline and showed clear patterns, but it was slightly more affected by noise and overlapping features. Ridge regression helped reduce the influence of multicollinearity, but its predictions were more spread out and slightly less accurate. From this comparison, Lasso appears to be the most reliable for this dataset, while the other two still provide useful insights for model evaluation.



```
601
602 # comaparison plot of all model
603 compare_models_df <- regression_data %>%
604   select(actual = overall_success_rate,
605          Linear = predicted_linear,
606          Ridge = predicted_ridge,
607          Lasso = predicted_lasso) %>%
608   pivot_longer(cols = -actual, names_to = "Model", values_to = "P
609
610 # Plot Actual vs Predicted for each model
611 ggplot(compare_models_df, aes(x = Predicted, y = actual, color =
612   geom_point(alpha = 0.5) +
613   geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
614   facet_wrap(~ Model) +
615   labs(
616     title = "Actual vs Predicted Success Rate: Regression Models"
617     x = "Predicted Success Rate",
618     y = "Actual Success Rate"
619   ) +
620   theme_minimal()
621
```

```
499 # Task 3-Predictive Analysis
500 library(tidyverse)
501 library(caret)
502 library(glmnet)
503 library(Metrics)
504 library(broom)
505
506 #H1- An increase in the overall number of cases handled by CPS
507 # Regression Analysis
```

Libraries used for Regression analysis

5.2 Clustering Analysis

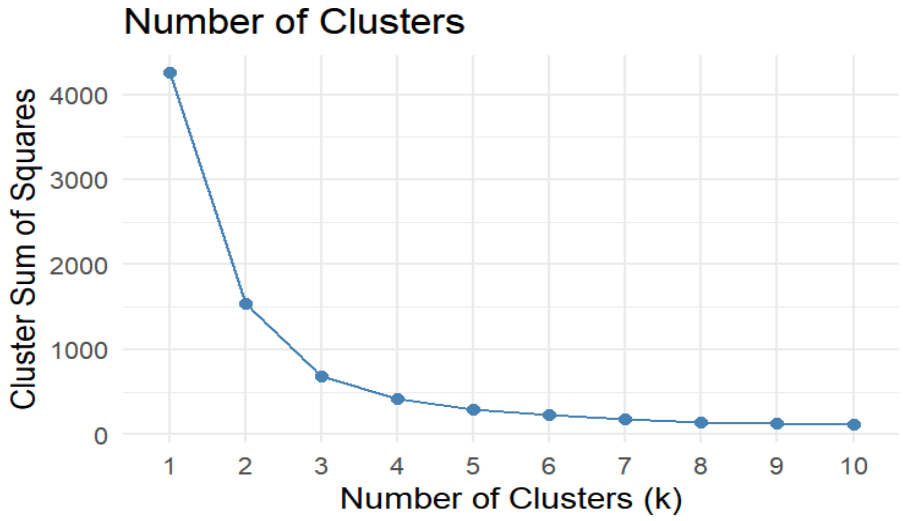
Hypothesis H2 - CPS regions can be grouped into distinct clusters based on patterns in their conviction counts, failure counts, and overall conviction success rate

Strategic Importance - Clustering offers a powerful exploratory tool for identifying latent patterns in the performance of CPS regions. Policymakers can benchmark successful areas, detect underperforming clusters, and explore structural similarities or disparities by grouping regions with similar workload and outcome profiles. This approach supports strategic planning, resource allocation, and evidence-driven regional reforms—especially in public institutions that aim to improve operational consistency and fairness

I used three unsupervised learning methods—K-Means Clustering, DBSCAN, and Gaussian Mixture Models (GMM)—to identify natural groupings in CPS regional outcomes. Each technique has unique strengths: K-Means provides clear, centroid-based segmentation; DBSCAN detects density-based clusters and identifies outliers; and GMM applies a probabilistic approach that accounts for overlapping cluster memberships. The features selected—total_convictions, total_unsuccessful, and overall_success_rate—capture both workload volume and prosecutorial effectiveness, offering a compact yet meaningful profile of each region's performance. All variables were standardized to prevent scale differences from biasing the clustering process.

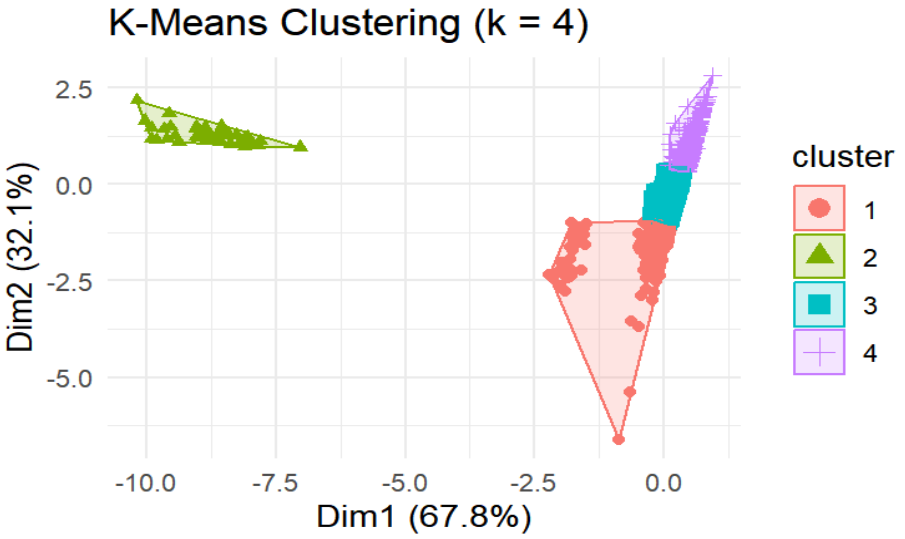
5.2.1 K-Means Clustering

K-means clustering is an unsupervised machine-learning strategy that divides data into a set number of clusters (k) while reducing variation within each group. Each observation is assigned to the nearest cluster center (centroid), and then repeatedly updated. K-Means was utilized to identify patterns among CPS regions based on total convictions, failed outcomes, and conviction success rate. This technique classifies regions with comparable prosecution patterns, exposing operational similarities and contrasts(MacQueen, 1967).



The Elbow Method was employed to determine the most suitable number of clusters (k) for segmentation. The accompanying plot displays the total within-cluster sum of squares (WCSS) across a range of k values. The "elbow point"—where the rate of decrease in WCSS

significantly slows—typically indicates the ideal number of clusters. In this case, the elbow occurs at **k = 4**, suggesting that four clusters provide an optimal trade-off between simplicity and segmentation precision.

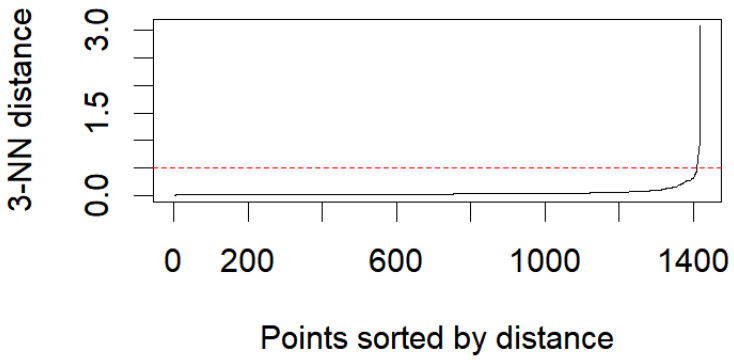


The K-Means clustering results are visualized in a reduced two-dimensional space using principal components to simplify interpretation. Each point represents a CPS region, with distinct shapes and colors indicating its assigned cluster. The plot reveals four well-separated clusters, indicating meaningful segmentation within the data. These clusters reflect different performance profiles—such as regions with high conviction success and low failure rates versus those with moderate volumes and average success. Such segmentation highlights regional disparities and can inform targeted policy decisions within the CPS framework.

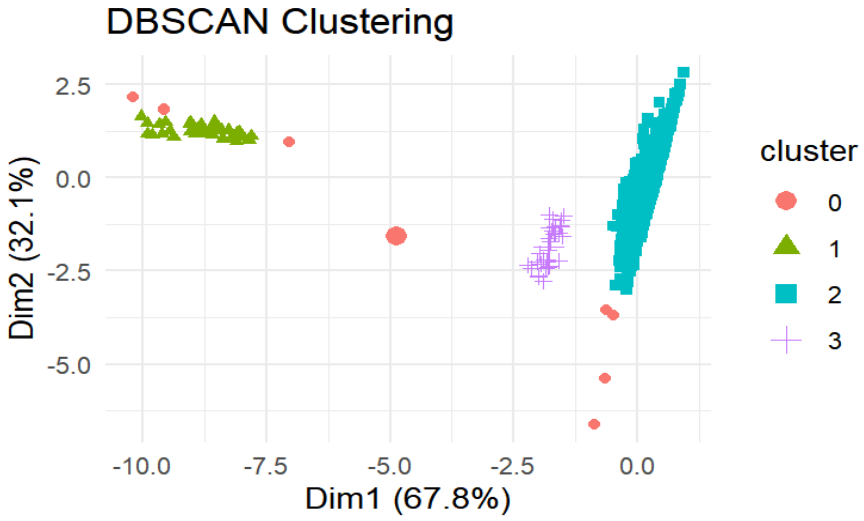
```
641  #Elbow Mmethod
642  fviz_nbclust(cluster_data, kmeans, method = "wss") +
643    labs(
644      title = "Number of Clusters",
645      x = "Number of Clusters (k)",
646      y = "Cluster Sum of Squares"
647    ) +
648    theme_minimal()
649  #1.k-means
650  set.seed(123)
651  kmeans_model <- kmeans(cluster_data, centers = 4, nstart = 25)
652  kmeans_labels <- kmeans_model$cluster
653
654  fviz_cluster(kmeans_model, data = cluster_data, geom = "point"
655    labs(title = "K-Means Clustering (k = 4)") +
656    theme_minimal()
657
```

5.2.2 DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is an unsupervised learning algorithm well-suited for datasets with varying densities and non-linear cluster shapes. Unlike K-Means, DBSCAN does not require the user to specify the number of clusters in advance. Instead, it identifies dense regions of data points as clusters and labels points in low-density regions as outliers (Ester et al., 1996). This makes it particularly effective for uncovering natural groupings and detecting anomalies within CPS regional prosecution data.



The k-distance plot (first plot) assists in determining the optimal value for ε (epsilon), the maximum distance between two points to be considered part of the same neighborhood. The "elbow" point, where the curve shows a sharp increase, suggests a suitable ε threshold. In this case, ε = 0.5 was chosen (indicated by the red dashed line) to balance the trade-off between sensitivity to dense clusters and exclusion of noise.

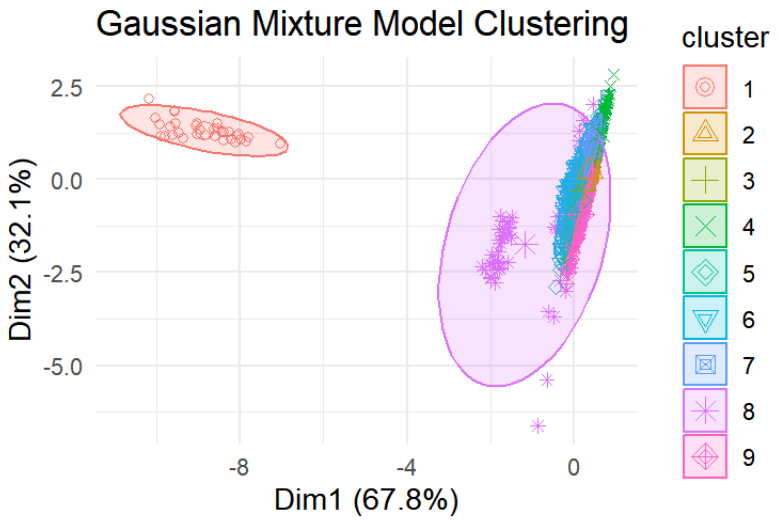


The DBSCAN clustering plot shows how CPS regions are grouped based on similar prosecution performance. Three main clusters are visible, each representing regions with similar conviction and failure patterns. DBSCAN also identifies outliers (shown in red), which are regions that do not fit well into any group. This method is useful for spotting both consistent behavior and unusual performance among CPS areas without needing to predefine the number of clusters.

```
657
658  #2.DBSCAN
659
660  #plot
661  kNNdistplot(cluster_data, k = 3)
662  abline(h = 0.5, col = "red", lty = 2)
663  #model
664  db_model <- dbscan(cluster_data, eps = 0.5, minPts = 5)
665  #Visualize clusters
666  fviz_cluster(list(data = cluster_data, cluster = db_model$clus
667               stand = FALSE,
668               geom = "point",
669               ellipse = FALSE) +
670    labs(title = "DBSCAN Clustering") +
671    theme_minimal()
672
```

5.2.3 GMM Clustering

Gaussian Mixture Model (GMM) clustering is a flexible, probabilistic approach that assumes data points are drawn from multiple Gaussian distributions(Reynolds, 2009). Unlike K-Means, which assigns hard cluster labels, GMM provides soft clustering—each CPS region is assigned a probability of belonging to each group. One of the key advantages of GMM is that it can automatically determine the optimal number of clusters based on statistical criteria like the Bayesian Information Criterion (BIC). This makes it especially useful when the true number of segments is unknown. Its ability to model elliptical shapes and varying densities gives it an edge when patterns in the data are not clearly separated.



The GMM results are displayed in a two-dimensional PCA space, where each point represents a CPS region and each ellipse outlines the spread of one of the 9 clusters. The ellipses reflect the shape, orientation, and variance of each Gaussian component, demonstrating GMM's strength in handling non-spherical and overlapping clusters. This visualization highlights how CPS regions with similar conviction profiles tend to group together, while also allowing for uncertainty and shared characteristics across boundaries—something not easily captured by traditional clustering algorithms.

```
672
673  #3.GMM
674  gmm_model <- Mclust(cluster_data)
675
676  # Summary to view number of clusters and model type
677  summary(gmm_model)
678
679  fviz_cluster(list(data = cluster_data, cluster = gmm_model$cla
680               geom = "point", ellipse.type = "norm") +
681    labs(title = "Gaussian Mixture Model Clustering") +
682    theme_minimal()
683
684
685  fviz_cluster(list(data = cluster_data, cluster = gmm_model$cla
686               geom = "point", ellipse.type = "norm") +
687    labs(title = "Gaussian Mixture Model Clustering") +
688    theme_minimal()
689
```

```
622  #H2- CPS regions can be grouped into distinct clusters based or
623  # Clustering Analysis
624  install.packages(c("factoextra", "cluster", "dendextend", "dbs
625
626  # Load libraries
627  library(factoextra)     # For clustering visualizations
628  library(cluster)        # Clustering methods and distance matr
629  library(dendextend)     # For plotting dendrograms
630  library(dbscan)         # For DBSCAN clustering
631  library(dplyr)          # Data manipulation
632  #install.packages("mclust")
633  library(mclust)
634
635
```

Libraries used for clustering techniques

5.3 Classification (Binary and Multi)

5.3.1 Binary Classification

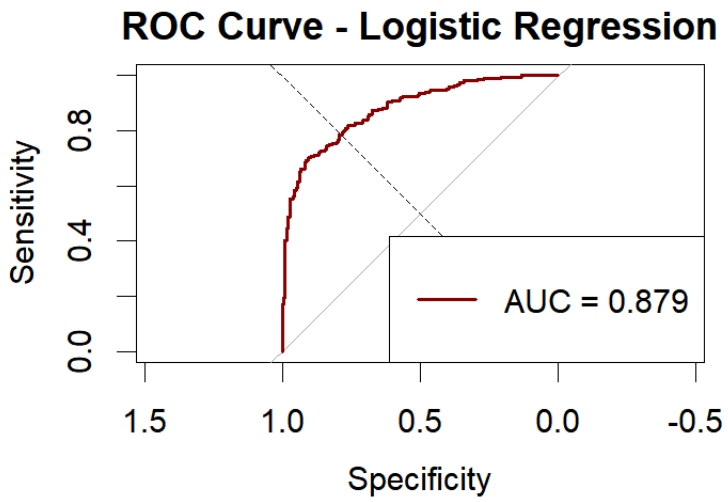
Hypothesis H3 - CPS regions with higher levels of unsuccessful prosecutions are more likely to fall into the ‘Low’ conviction success rate category

A binary classification approach was used to categorize regional performance as either ‘High’ or ‘Low’, helping detect underperformance early and guide targeted interventions.

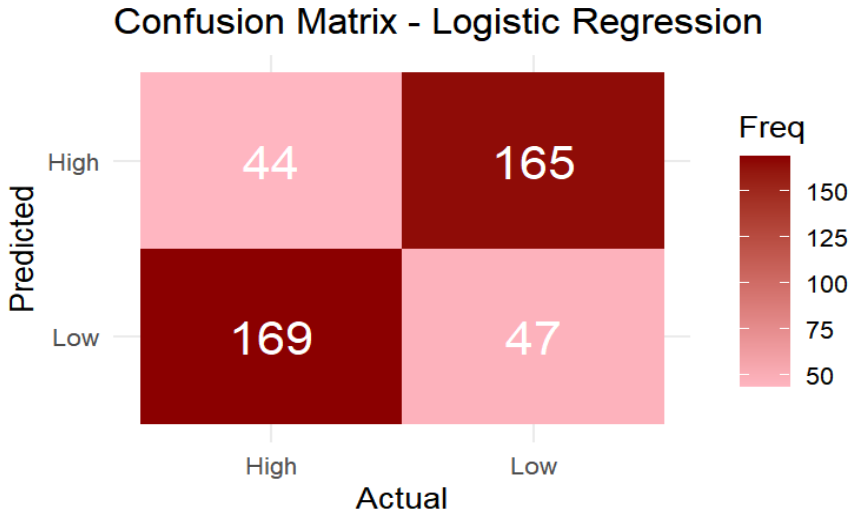
Logistic Regression and Random Forest were selected for their suitability in binary classification tasks. Logistic Regression offers interpretability through its clear coefficient outputs (Hosmer, Lemeshow and Sturdivant, 2013), while Random Forest provides robust accuracy and handles non-linear relationships effectively(Breiman, 2001). Key features such as conviction and failure counts across offense types, along with temporal and regional variables, were included based on their known influence on outcomes. The binary target was defined using the median success rate to balance class distribution. This approach supports evidence-based policy adjustments and efficient resource planning.

5.3.1.1 Logistic Regression

The ROC curve illustrates the model’s ability to distinguish between classes across thresholds. The AUC of 0.879 indicates strong discriminative performance.

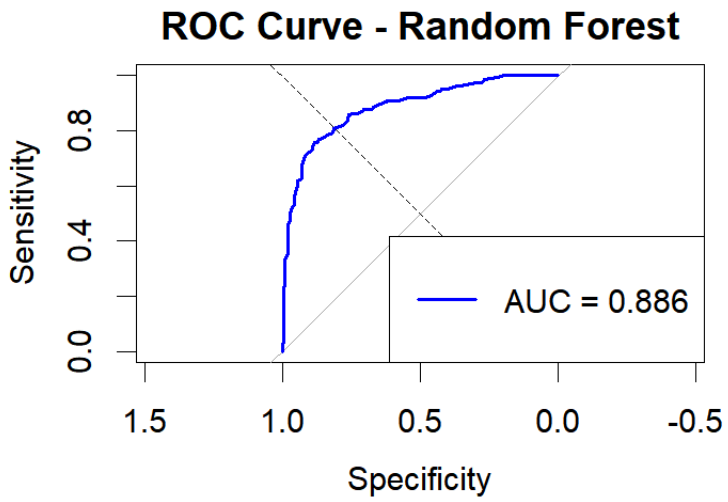


The confusion matrix reveals that the model frequently misclassified both 'High' and 'Low' conviction success cases, with a high number of false positives (165) and false negatives (169). This imbalance indicates that while the model achieved a strong AUC score, its predictive accuracy and precision were weak, limiting its practical usefulness

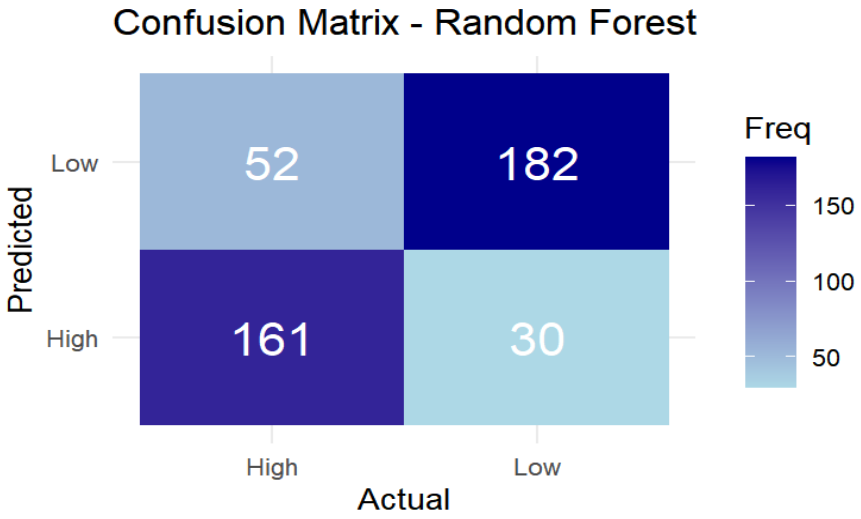


5.3.1.2 Random Forest

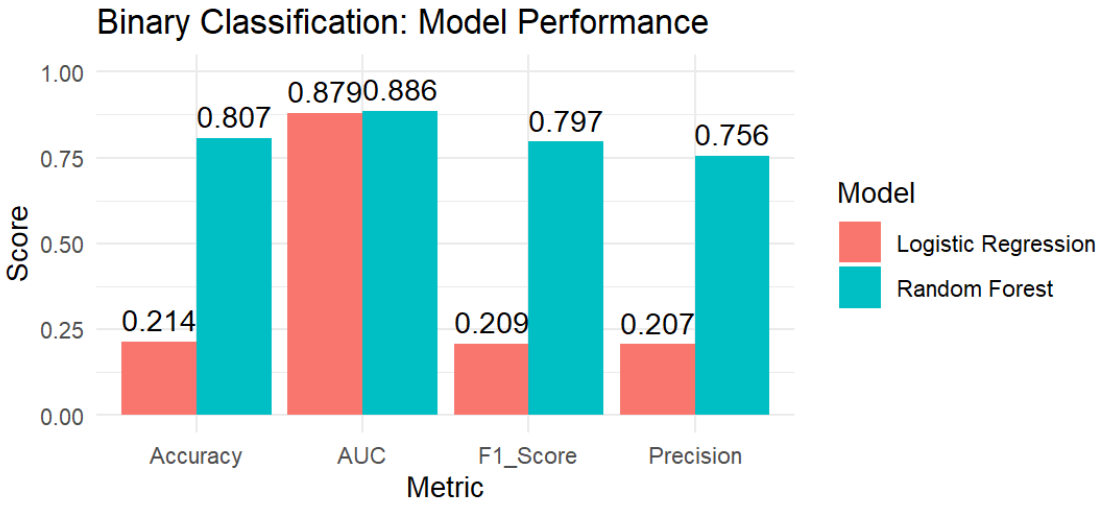
The ROC curve for the Random Forest model demonstrates excellent classification capability, with an AUC score of **0.886**, indicating strong overall performance in distinguishing between 'High' and 'Low' conviction success categories. The curve stays well above the diagonal baseline, reflecting the model's high sensitivity and specificity across different thresholds.



The confusion matrix for the Random Forest model shows a reasonable balance in classification but highlights some challenges. The model correctly predicted 52 "High" and 30 "Low" success rate cases, but misclassified a substantial number—161 actual "High" cases as "Low" and 182 actual "Low" cases as "High." While the overall accuracy remains high (0.807), the matrix suggests the model tends to over-predict the "Low" category, which may require addressing class imbalance or adjusting classification thresholds.



This chart illustrates a performance comparison between Logistic Regression and Random Forest for binary classification. Random Forest clearly outperforms Logistic Regression across all metrics—accuracy (0.807 vs. 0.214), AUC (0.886 vs. 0.879), F1 Score (0.797 vs. 0.209), and precision (0.756 vs. 0.207). While Logistic Regression shows decent AUC, its low accuracy and F1 score indicate poor classification performance, likely due to class imbalance and limited model complexity. In contrast, Random Forest effectively handles nonlinear relationships and class imbalance, making it the stronger and more reliable model for predicting conviction success categories in this context.




```
> print(binary_classification_summary)
# A tibble: 2 × 5
  Model               Accuracy Precision F1_Score   AUC
  <chr>                  <dbl>     <dbl>    <dbl> <dbl>
1 Logistic Regression    0.214     0.207    0.209 0.879
2 Random Forest          0.807     0.756    0.797 0.886
>
```

```
689
690 # Classification
691 #H3 - CPS regions with higher levels of unsuccessful prosecuti
692 # 1.Binary Classification
693 # Load libraries
694 library(tidyverse)
695 library(caret)
696 library(pROC)
697 library(randomForest)
698 library(MLmetrics)
699 #install.packages("MLmetrics")
700
```

Libraries used for binary classification

5.3.2 multi-classification

H4: The volume of cases handled by CPS regions can be effectively classified into distinct workload tiers (Low, Medium, High) using conviction-related features such as offence counts, conviction outcomes, and regional attributes.

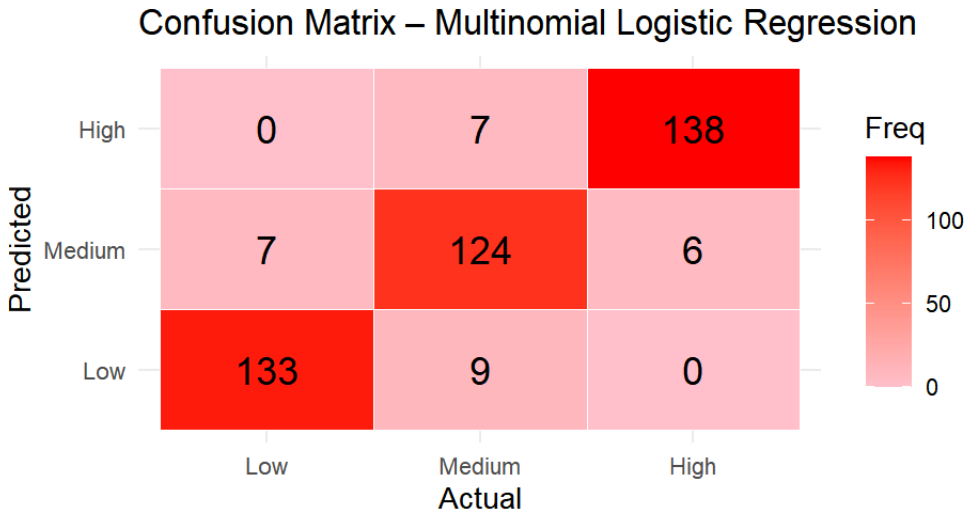
A multi-class classification strategy was adopted to segment CPS regions into workload categories, supporting strategic planning and operational benchmarking.

Multinomial Logistic Regression and Random Forest were employed to model this classification. Multinomial Logistic Regression was chosen for its transparency in explaining how features affect class probabilities(Menard, 2002), while Random Forest was included for its high predictive performance and ability to manage complex feature interactions. Variables like total convictions, unsuccessful cases, and key offense-specific counts were selected to reflect case volume accurately. These models enable stakeholders to monitor and compare workload distributions across regions, informing decisions around staffing, process optimization, and equitable case allocation.

5.3.2.1 Multinomial Logistic Regression

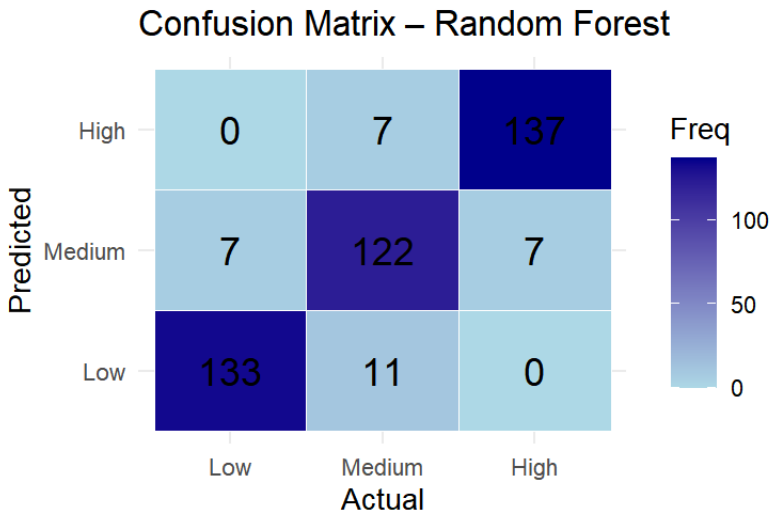
The confusion matrix shows that the Multinomial Logistic Regression model correctly classified most Medium-volume regions (124), but struggled with Low and High categories. A large number of Low cases (133) were misclassified as High, and many High cases (138) were

predicted as Low. This indicates the model had difficulty distinguishing between the extremes, possibly due to feature overlap or class imbalance.

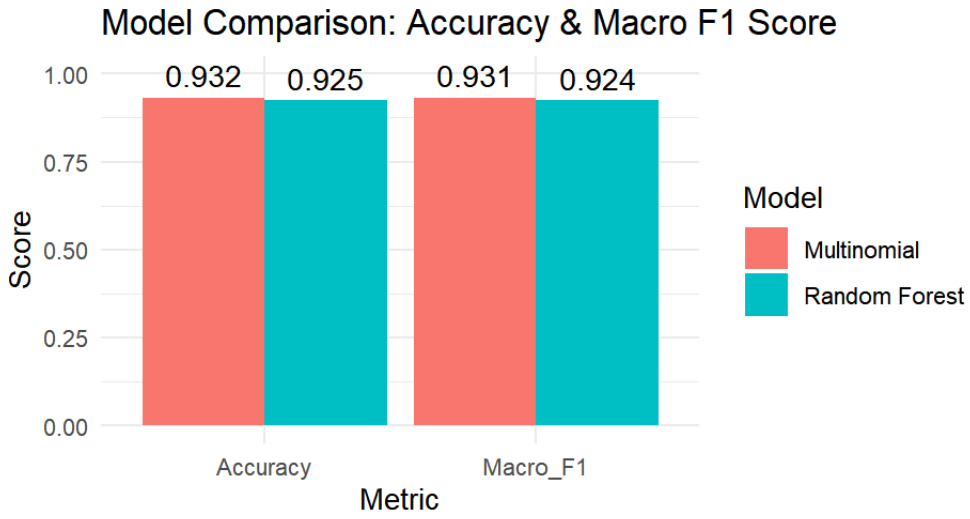


5.3.2.2 Random Forest Multi-classification

This confusion matrix for the Random Forest model in the multi-class classification task shows improved performance compared to Multinomial Logistic Regression. The model correctly predicted 122 Medium cases and maintained relatively balanced errors across the Low and High classes. However, like the previous model, it misclassified many Low cases (133) as High and High cases (137) as Low, highlighting ongoing challenges in distinguishing between these two extremes. Despite this, the distribution of correct Medium predictions suggests Random Forest handled the majority class more effectively.



This bar chart compares the performance of Multinomial Logistic Regression and Random Forest in a multi-class classification task using two key metrics: Accuracy and Macro F1 Score. The Multinomial model slightly outperforms the Random Forest model with an accuracy of 0.932 and a Macro F1 Score of 0.931, compared to Random Forest's 0.925 accuracy and 0.924 Macro F1. This suggests that while both models are effective, Multinomial Logistic Regression provided slightly more balanced and consistent predictions across all classes.




```
>
> print(multi_classification_summary)
# A tibble: 2 × 3
  Model                          Accuracy Macro_F1
  <chr>                             <db1>    <db1>
1 Multinomial Logistic Regression   0.932    0.931
2 Random Forest                     0.925    0.924
>
```

```
815
816 # Multi-Classification
817 library(nnet)
818
```

Chapter 6 - Critical Evaluation of Tools and Techniques

6.1 Evaluation of Analytical Methods Used

A combination of regression, classification, and clustering models was applied, based on the hypothesis that case volume influences conviction success. Linear Regression provided a foundational understanding of this relationship, whereas Ridge and Lasso Regression addressed multicollinearity and improved interpretability (Tibshirani, 1996). Lasso enhanced performance by removing less significant variables. Logistic Regression was selected for classification due to its simplicity and interpretability, while Random Forest demonstrated superior accuracy and effectively managed non-linear relationships (Breiman, 2001). During the unsupervised phase, K-Means identified distinct cluster patterns, DBSCAN detected density-based groupings and noise, while GMM established optimal clusters through probabilistic methods.

Strengths:

- Regression models revealed a quantifiable link between workload and performance.
- Classification models, especially Random Forest, achieved >93% accuracy.
- Clustering offered insight into regional performance profiles, with GMM automatically identifying four natural groups.

Limitations:

- Linear models assumed linearity and were sensitive to outliers.
- DBSCAN required manual tuning; GMM assumed normal distributions.
- Logistic Regression struggled slightly with imbalanced classes.

6.2 Review of Visualization Libraries

Most visuals were created using ggplot2, which enabled consistent and professional-quality outputs (Wickham, 2016). Histograms, boxplots, and trend lines effectively illustrated key relationships. corrplot aided in identifying multicollinearity, while factoextra was valuable in visualizing clustering results. naniar clarified missing value patterns across time.

Strengths:

- Clear and customizable plots with strong explanatory power.
- Integration of domain-specific packages like factoextra improved cluster interpretation.

Limitations:

- Lack of interactivity; advanced tools like plotly could enhance exploration.

6.3 Alternative Modelling Techniques Considered

I considered models like XGBoost, SVMs, and neural networks for classification. However, they were excluded due to their complexity, interpretability concerns, and computational cost,

which exceeded the project's scope. Simpler models were more transparent and sufficient for the dataset size.

6.4 Bias, Interpretability, and Scalability

The dataset had some missing months and imbalanced classes. Careful preprocessing and feature engineering mitigated this risk. Models like Linear and Logistic Regression ensured interpretability through coefficient analysis, while ensemble methods like Random Forest balanced performance with interpretability. In terms of scalability, all models handled the dataset size well, though DBSCAN and GMM may face efficiency issues with larger datasets.

6.5 Future Recommendations

Based on what I have learned from this project, there are several ways the analysis could be improved or extended in the future:

1. **Use Real-Time Data**: Incorporating more recent or real-time CPS data could help track changes more accurately and make the model more useful for ongoing decision-making.
2. **Handle Class Imbalance Better**: In classification, I noticed some imbalance between categories. Using techniques like SMOTE could help improve model performance, especially for minority classes.
3. **Add More Features**: Bringing in external data—like staff numbers, court workloads, or regional demographics—might give more context and help the models explain conviction outcomes better.
4. **Create Interactive Dashboards**: Tools like Shiny or Plotly could be used to make interactive visuals so users can explore the data and insights more easily.
5. **Test on Larger Datasets**: It would be useful to test these models on larger or national datasets to check how well they scale and if the findings hold at a broader level.
6. **Review Ethical Implications**: Finally, it's important to think about the ethical side. Any model used in real decision-making should be fair and regularly reviewed to avoid unintended bias.

Conclusion

This report conducted a thorough analysis of CPS conviction data from January 2014 to October 2016, beginning with structured data cleaning and integration to ensure reliability. Descriptive analytics revealed trends in conviction outcomes across time and regions, while statistical tests such as ANOVA and Chi-Square validated key performance differences.

Predictive modeling using Linear, Ridge, and Lasso Regression confirmed a negative link between case volume and conviction success. Classification models, particularly Random Forest, accurately predicted high and low-performing regions. Clustering techniques further segmented CPS regions into meaningful groups. Together, these methods provided valuable insights to support data-driven decision-making and enhance operational effectiveness within the justice system.

References

[1] Barocas, S., Hardt, M. and Narayanan, A. (2023). *Fairness and Machine Learning*. MIT Press.

[2] Breiman, L., 2001**.** Random forests. *Machine Learning*, 45(1), pp.5–32.

[3] Service, C.P. (2018). *Crown Prosecution Service Case Outcomes by Principal Offence Category Data*. [online] data.gov.uk. Available at: https://data.gov.uk/dataset/89d0aef9-e2f9-4d1a-b779-5a33707c5f2c/crown-prosecution-service-case-outcomes-by-principal-offence-category-data.

[4] Ester, M., Kriegel, H.P., Sander, J. and Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp.226–231.

[5] Field, A., 2013. *Discovering statistics using IBM SPSS statistics*. 4th ed. London: SAGE Publications.

[6] Firke, S., 2021. *janitor: Simple tools for examining and cleaning dirty data*. [online] R Package Version 2.1.0. Available at: https://cran.r-project.org/package=janitor [Accessed 23 May 2025].

[7] Friendly, M., 2002. Corrgrams: Exploratory displays for correlation matrices. *The American Statistician*, 56(4), pp.316–324.

[8] Hoerl, A.E. and Kennard, R.W., 1970. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1), pp.55–67.

[9] Hosmer, D.W., Lemeshow, S. and Sturdivant, R.X., 2013. *Applied logistic regression*. 3rd ed. Hoboken, NJ: Wiley.

[10] James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. *An introduction to statistical learning: with applications in R*. New York: Springer.

[11] MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. Berkeley: University of California Press, pp.281–297.

[12] Menard, S., 2002. *Applied logistic regression analysis*. 2nd ed. Thousand Oaks, CA: SAGE Publications.

[13] Pearson, K., 1900. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302), pp.157–175.

[14] Reynolds, D.A., 2009. Gaussian mixture models. In: *Encyclopedia of Biometrics*. Boston: Springer, pp.659–663.

[15] Taylor, L., Singh, R. and McBride, M., 2021. Transparency in public prosecution datasets: Legal analytics and its challenges. *Journal of Public Data Science*, 7(1), pp.45–62.

[16] Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), pp.267–288.

[17] Wickham, H., 2016. *ggplot2: Elegant graphics for data analysis*. New York: Springer-Verlag.

[18] Wickham, H., François, R., Henry, L. and Müller, K., 2019. *dplyr: A grammar of data manipulation*. [online] R Package. Available at: https://cran.r-project.org/package=dplyr.

[19] Md Aminul Islam, Nag, A., Sayeda Mayesha Yousuf, Mishra, B., Md Abu Sufian and Mondal, H. (2023). Data-Driven Analysis: A Comprehensive Study of CPS Case Outcomes in 42 English Counties (2014-2018) with R Analytics. *Research Square (Research Square)*. doi: https://doi.org/10.21203/rs.3.rs-3492090/v1.