UNIVERSITY OF GLOUCESTERSHIRE

Data Analysis and Visualisation Principles Report

Student Name: Maral Zarafshan
Student Code: S4218518
Module Title: CT7202 - Data Analysis and Visualisation Principles
Module Tutor: Dr. Bhupesh Mishra
School of Computing and Technology
University of Gloucestershire
June 2023

Introduction

The Crown Prosecution Service (CPS) plays a vital role in the criminal justice system, acting as the principal prosecuting authority in England and Wales. The CPS is an independent organization. They are responsible for deciding whether to charge individuals with criminal offenses and for conducting prosecutions in courts across the country. In this report, we will explore the dataset obtained from the CPS, specifically focusing on the outcomes of criminal cases categorized by principal offense.

The dataset, sourced from the data.gov.uk website, comprises information on the outcomes of cases handled by the CPS over a specific period. It provides valuable insights into convictions, acquittals, and other case outcomes related to various offense categories. This dataset spans a time frame from April 2015 to March 2018 and covers a range of offenses, including homicide, drugs offenses, fraud, forgery, theft, handling, and robbery etc.

By analyzing this dataset, we aim to gain a deeper understanding of the patterns and trends associated with different types of criminal offenses, focusing particularly on Drugs, Homicide, Fraud and Forgery and Against person. These offenses have significant implications for society, and exploring their outcomes can shed light on the effectiveness of the criminal justice system in dealing with such crimes.

In the following sections, we will delve into the details of the dataset, explain the methodologies employed, and present the analysis and findings. By examining the data on criminal case outcomes, we aim to provide valuable insights into the performance of the CPS and the criminal justice system's response to Drugs, Homicide, Fraud and forgery offenses during the specified period.

Objective of the Report:

The primary objective of this report is to analyse the relationships between drug use and homicide as well as patterns in fraud and forgeries. We intend to achieve the following goals by analysing the available data and doing a thorough analysis:

- Study the Connection Between Drug Use and Homicide: Look at the link between crimes involving drugs and homicides.
- Analyse the frequency of drug use among those who take part in homicides.
- Examine the details of drug-related homicides, including its causes and circumstances.
- Find any patterns or trends that might appear between drug use and the frequency of killings.
- Examine the forgeries and fraud throughout time.

Hypothesis one:

There is a positive association between the number of drug offences and homicide convictions, suggesting that drug-related criminal activities contribute to an increased risk of homicides."

This hypothesis suggests that there is a relationship between drug offences and homicides, implying that areas or time periods with higher drug-related criminal activities may also experience higher rates of homicides. The hypothesis assumes that involvement in drug-related activities, either as a perpetrator or a victim, increases the likelihood of engaging in violent behaviour, leading to an increased number of homicide convictions. This hypothesis can be tested by analyzing the correlation or patterns between drug offences and homicide convictions in the dataset.

Hypothesis two:

related to fraud and forgery during the period of April 2015 to March 2018 could be:

The incidence of fraud and forgery cases increased over time, indicating a rise in fraudulent activities during the specified period.

This hypothesis suggests that there might have been a temporal trend of increasing fraud and forgery cases during the selected time frame. It assumes that factors such as advancements in technology, changes in economic conditions, or evolving criminal tactics may have contributed to a higher occurrence of fraudulent activities. The hypothesis can be tested by examining the frequency or rate of fraud and forgery cases over time using the dataset. Statistical analysis or visualization techniques can help identify any patterns or trends in the occurrence of these offenses.

Data Cleaning and Integration Techniques

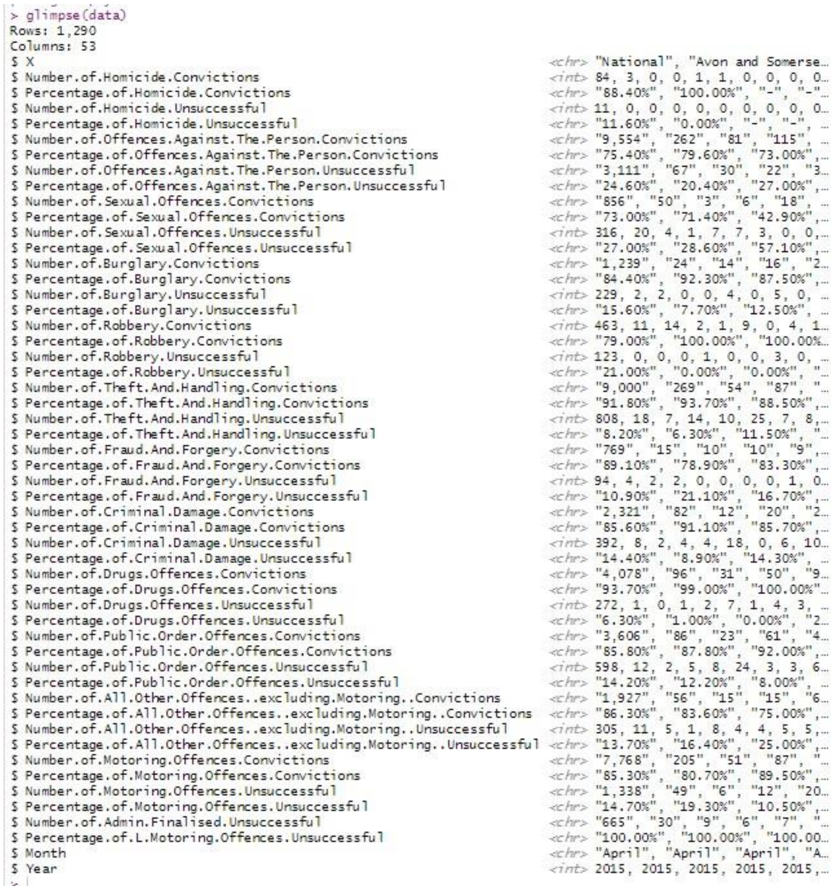
My initial focus was on combining multiple CSV files, standardizing column names, and sorting the data using the "Year" column. The directory is first set in the code, and then all CSV files are shown there. After that, it creates an empty list to store the data frames and sequentially reads and stores each CSV file in the list by looping through each one. The columns in each data frame are renamed to the common names after identifying the common column names shared by all data frames.

The combined data is sent to a CSV file when the data frames have been merged into one. The last column's name is then changed to "Year" in the code after reading the combined data from the CSV file. Changing the "Year" column to numeric, and the data frame is arranged in ascending order according to the "Year" column. The sorted data frame is then saved to a fresh CSV file.

To provide consistent column names and to sort the data by year, the algorithm aggregates and standardizes the data from different CSV files. This makes it simpler to compare and analyse data from various sources. The merging and sorting of the data is made easier using loops and methods like order and rbind. However, it would be advantageous to provide error-handling features in case any CSV files are absent or if the column names are inconsistent. Furthermore, the code could benefit from including data cleaning and validation procedures to deal with missing or inaccurate variables before performing the merging and sorting operations.

Data cleaning:

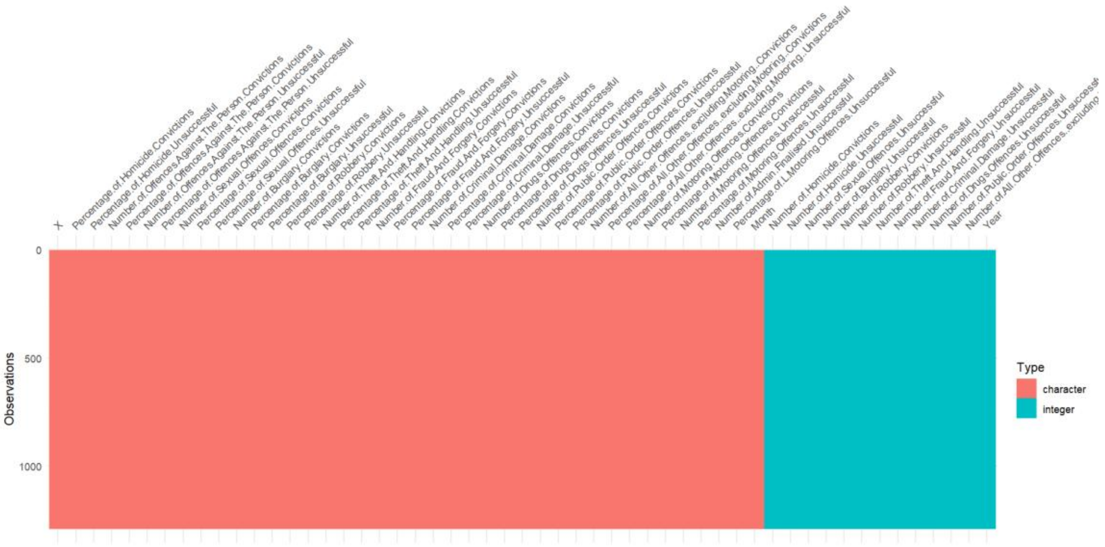
To begin, we are going to run the head () function, which allows us to see the first 6 rows by default. And for displaying the type and a preview of all columns as a row using glimpse from “dplyr” library:



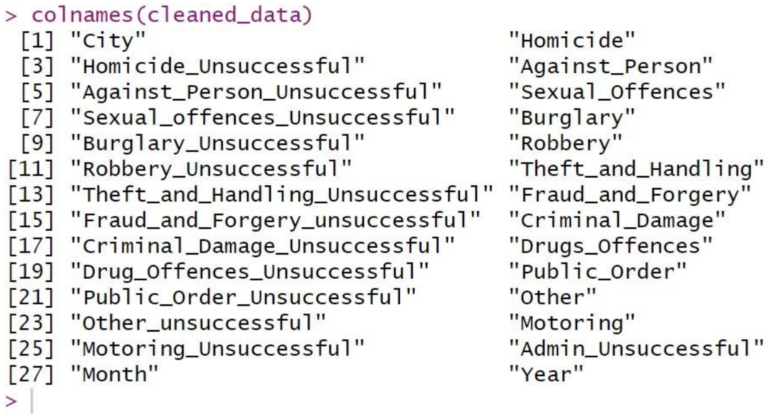
```
> glimpse(data)
Rows: 1,290
Columns: 53
$ X                                                              <chr> "National", "Avon and Somerse…
$ Number.of.Homicide.Convictions                                 <int> 84, 3, 0, 0, 1, 1, 0, 0, 0, 0…
$ Percentage.of.Homicide.Convictions                             <chr> "88.40%", "100.00%", "-", "-"…
$ Number.of.Homicide.Unsuccessful                                <int> 11, 0, 0, 0, 0, 0, 0, 0, 0, 0…
$ Percentage.of.Homicide.Unsuccessful                            <chr> "11.60%", "0.00%", "-", "-", …
$ Number.of.Offences.Against.The.Person.Convictions              <chr> "9,554", "262", "81", "115", …
$ Percentage.of.Offences.Against.The.Person.Convictions          <chr> "75.40%", "79.60%", "73.00%",…
$ Number.of.Offences.Against.The.Person.Unsuccessful             <chr> "3,111", "67", "30", "22", "3…
$ Percentage.of.Offences.Against.The.Person.Unsuccessful         <chr> "24.60%", "20.40%", "27.00%",…
$ Number.of.Sexual.Offences.Convictions                          <chr> "856", "50", "3", "6", "18", …
$ Percentage.of.Sexual.Offences.Convictions                      <chr> "73.00%", "71.40%", "42.90%",…
$ Number.of.Sexual.Offences.Unsuccessful                         <int> 316, 20, 4, 1, 7, 7, 3, 0, 0,…
$ Percentage.of.Sexual.Offences.Unsuccessful                     <chr> "27.00%", "28.60%", "57.10%",…
$ Number.of.Burglary.Convictions                                 <chr> "1,239", "24", "14", "16", "2…
$ Percentage.of.Burglary.Convictions                             <chr> "84.40%", "92.30%", "87.50%",…
$ Number.of.Burglary.Unsuccessful                                <int> 229, 2, 2, 0, 0, 4, 0, 5, 0, …
$ Percentage.of.Burglary.Unsuccessful                            <chr> "15.60%", "7.70%", "12.50%", …
$ Number.of.Robbery.Convictions                                  <int> 463, 11, 14, 2, 1, 9, 0, 4, 1…
$ Percentage.of.Robbery.Convictions                              <chr> "79.00%", "100.00%", "100.00%…
$ Number.of.Robbery.Unsuccessful                                 <int> 123, 0, 0, 0, 1, 0, 0, 3, 0, …
$ Percentage.of.Robbery.Unsuccessful                             <chr> "21.00%", "0.00%", "0.00%", "…
$ Number.of.Theft.And.Handling.Convictions                       <chr> "9,000", "269", "54", "87", "…
$ Percentage.of.Theft.And.Handling.Convictions                   <chr> "91.80%", "93.70%", "88.50%",…
$ Number.of.Theft.And.Handling.Unsuccessful                      <int> 808, 18, 7, 14, 10, 25, 7, 8,…
$ Percentage.of.Theft.And.Handling.Unsuccessful                  <chr> "8.20%", "6.30%", "11.50%", "…
$ Number.of.Fraud.And.Forgery.Convictions                        <chr> "769", "15", "10", "10", "9",…
$ Percentage.of.Fraud.And.Forgery.Convictions                    <chr> "89.10%", "78.90%", "83.30%",…
$ Number.of.Fraud.And.Forgery.Unsuccessful                       <int> 94, 4, 2, 2, 0, 0, 0, 0, 1, 0…
$ Percentage.of.Fraud.And.Forgery.Unsuccessful                   <chr> "10.90%", "21.10%", "16.70%",…
$ Number.of.Criminal.Damage.Convictions                          <chr> "2,321", "82", "12", "20", "2…
$ Percentage.of.Criminal.Damage.Convictions                      <chr> "85.60%", "91.10%", "85.70%",…
$ Number.of.Criminal.Damage.Unsuccessful                         <int> 392, 8, 2, 4, 4, 18, 0, 6, 10…
$ Percentage.of.Criminal.Damage.Unsuccessful                     <chr> "14.40%", "8.90%", "14.30%", …
$ Number.of.Drugs.Offences.Convictions                           <chr> "4,078", "96", "31", "50", "9…
$ Percentage.of.Drugs.Offences.Convictions                       <chr> "93.70%", "99.00%", "100.00%"…
$ Number.of.Drugs.Offences.Unsuccessful                          <int> 272, 1, 0, 1, 2, 7, 1, 4, 3, …
$ Percentage.of.Drugs.Offences.Unsuccessful                      <chr> "6.30%", "1.00%", "0.00%", "2…
$ Number.of.Public.Order.Offences.Convictions                    <chr> "3,606", "86", "23", "61", "4…
$ Percentage.of.Public.Order.Offences.Convictions                <chr> "85.80%", "87.80%", "92.00%",…
$ Number.of.Public.Order.Offences.Unsuccessful                   <int> 598, 12, 2, 5, 8, 24, 3, 3, 6…
$ Percentage.of.Public.Order.Offences.Unsuccessful               <chr> "14.20%", "12.20%", "8.00%", …
$ Number.of.All.Other.Offences..excluding.Motoring..Convictions  <chr> "1,927", "56", "15", "15", "6…
$ Percentage.of.All.Other.Offences..excluding.Motoring..Convictions <chr> "86.30%", "83.60%", "75.00%",…
$ Number.of.All.Other.Offences..excluding.Motoring..Unsuccessful <int> 305, 11, 5, 1, 8, 4, 4, 5, 5,…
$ Percentage.of.All.Other.Offences..excluding.Motoring..Unsuccessful <chr> "13.70%", "16.40%", "25.00%",…
$ Number.of.Motoring.Offences.Convictions                        <chr> "7,768", "205", "51", "87", "…
$ Percentage.of.Motoring.Offences.Convictions                    <chr> "85.30%", "80.70%", "89.50%",…
$ Number.of.Motoring.Offences.Unsuccessful                       <chr> "1,338", "49", "6", "12", "20…
$ Percentage.of.Motoring.Offences.Unsuccessful                   <chr> "14.70%", "19.30%", "10.50%",…
$ Number.of.Admin.Finalised.Unsuccessful                         <chr> "665", "30", "9", "6", "7", "…
$ Percentage.of.L.Motoring.Offences.Unsuccessful                 <chr> "100.00%", "100.00%", "100.00…
$ Month                                                          <chr> "April", "April", "April", "A…
$ Year                                                           <int> 2015, 2015, 2015, 2015, 2015,…
```

The code begins by loading necessary packages such as readr, tidyverse, and dplyr. It then uses functions like head, glimpse, and summary to get an overview of the dataset, including the column names, dimensions, and summary statistics.

Next, the code utilizes the skimr package to generate more detailed summary statistics, including missing values and inline histograms for each variable. The visdat package is also used to visualize missing data in the dataset:



Data cleaning operations are performed to remove irrelevant or incomplete data. Columns deemed irrelevant are excluded from the dataset. The code also uses grep() function to remove columns containing the word "Percentage" and renames the remaining columns using a predefined set of names.

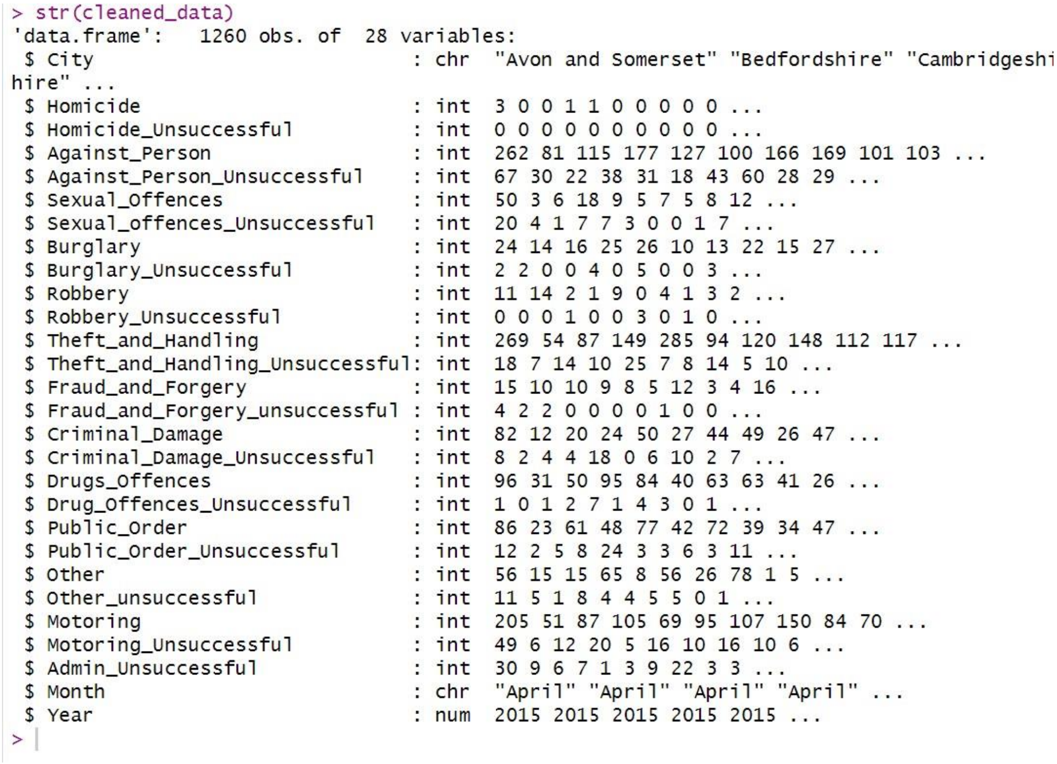


```
> colnames(cleaned_data)
 [1] "City"                             "Homicide"
 [3] "Homicide_Unsuccessful"            "Against_Person"
 [5] "Against_Person_Unsuccessful"      "Sexual_Offences"
 [7] "Sexual_offences_Unsuccessful"     "Burglary"
 [9] "Burglary_Unsuccessful"            "Robbery"
[11] "Robbery_Unsuccessful"             "Theft_and_Handling"
[13] "Theft_and_Handling_Unsuccessful"  "Fraud_and_Forgery"
[15] "Fraud_and_Forgery_unsuccessful"   "Criminal_Damage"
[17] "Criminal_Damage_Unsuccessful"     "Drugs_Offences"
[19] "Drug_Offences_Unsuccessful"       "Public_Order"
[21] "Public_Order_Unsuccessful"        "Other"
[23] "Other_unsuccessful"               "Motoring"
[25] "Motoring_Unsuccessful"            "Admin_Unsuccessful"
[27] "Month"                            "Year"
>
```

Further data cleaning involves removing specific rows based on the city name and converting certain columns to the integer data type. After checking data summary, it was revealed that there were certain NA values present. To address this issue, a closer examination was carried out, leading to the identification of the rows associated with metropolitan and city areas as the sources of these missing values. Consequently, a decision was made to exclude these rows from the dataset. By eliminating the metropolitan and city rows, we aimed to ensure the integrity and accuracy of the data, enabling us to proceed with further analysis and interpretation with confidence.

Duplicate rows are also checked and removed if found.

Finally, the cleaned dataset is saved as a CSV file named "cleaned_data.csv" for further analysis.



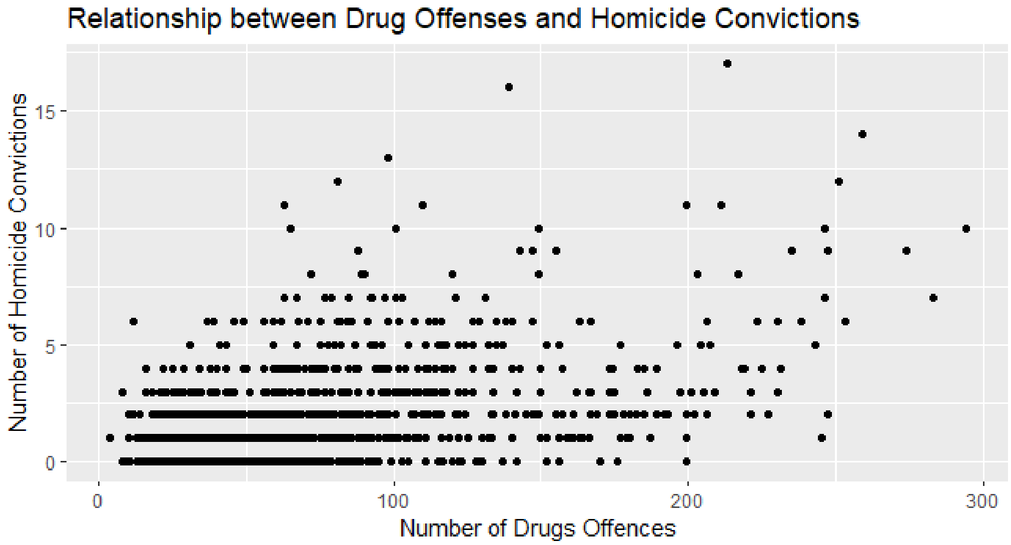
```
> str(cleaned_data)
'data.frame':	1260 obs. of  28 variables:
 $ City                           : chr  "Avon and Somerset" "Bedfordshire" "Cambridgeshi
hire" ...
 $ Homicide                       : int  3 0 0 1 1 0 0 0 0 0 ...
 $ Homicide_Unsuccessful          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ Against_Person                 : int  262 81 115 177 127 100 166 169 101 103 ...
 $ Against_Person_Unsuccessful    : int  67 30 22 38 31 18 43 60 28 29 ...
 $ Sexual_Offences                : int  50 3 6 18 9 5 7 5 8 12 ...
 $ Sexual_offences_Unsuccessful   : int  20 4 1 7 7 3 0 0 1 7 ...
 $ Burglary                       : int  24 14 16 25 26 10 13 22 15 27 ...
 $ Burglary_Unsuccessful          : int  2 2 0 0 4 0 5 0 0 3 ...
 $ Robbery                        : int  11 14 2 1 9 0 4 1 3 2 ...
 $ Robbery_Unsuccessful           : int  0 0 0 1 0 0 3 0 1 0 ...
 $ Theft_and_Handling             : int  269 54 87 149 285 94 120 148 112 117 ...
 $ Theft_and_Handling_Unsuccessful: int  18 7 14 10 25 7 8 14 5 10 ...
 $ Fraud_and_Forgery              : int  15 10 10 9 8 5 12 3 4 16 ...
 $ Fraud_and_Forgery_unsuccessful : int  4 2 2 0 0 0 0 1 0 0 ...
 $ Criminal_Damage                : int  82 12 20 24 50 27 44 49 26 47 ...
 $ Criminal_Damage_Unsuccessful   : int  8 2 4 4 18 0 6 10 2 7 ...
 $ Drugs_Offences                 : int  96 31 50 95 84 40 63 63 41 26 ...
 $ Drug_Offences_Unsuccessful     : int  1 0 1 2 7 1 4 3 0 1 ...
 $ Public_Order                   : int  86 23 61 48 77 42 72 39 34 47 ...
 $ Public_Order_Unsuccessful      : int  12 2 5 8 24 3 3 6 3 11 ...
 $ Other                          : int  56 15 15 65 8 56 26 78 1 5 ...
 $ Other_unsuccessful             : int  11 5 1 8 4 4 5 5 0 1 ...
 $ Motoring                       : int  205 51 87 105 69 95 107 150 84 70 ...
 $ Motoring_Unsuccessful          : int  49 6 12 20 5 16 10 16 10 6 ...
 $ Admin_Unsuccessful             : int  30 9 6 7 1 3 9 22 3 3 ...
 $ Month                          : chr  "April" "April" "April" "April" ...
 $ Year                           : num  2015 2015 2015 2015 2015 ...
> 
```

Descriptive Analytics

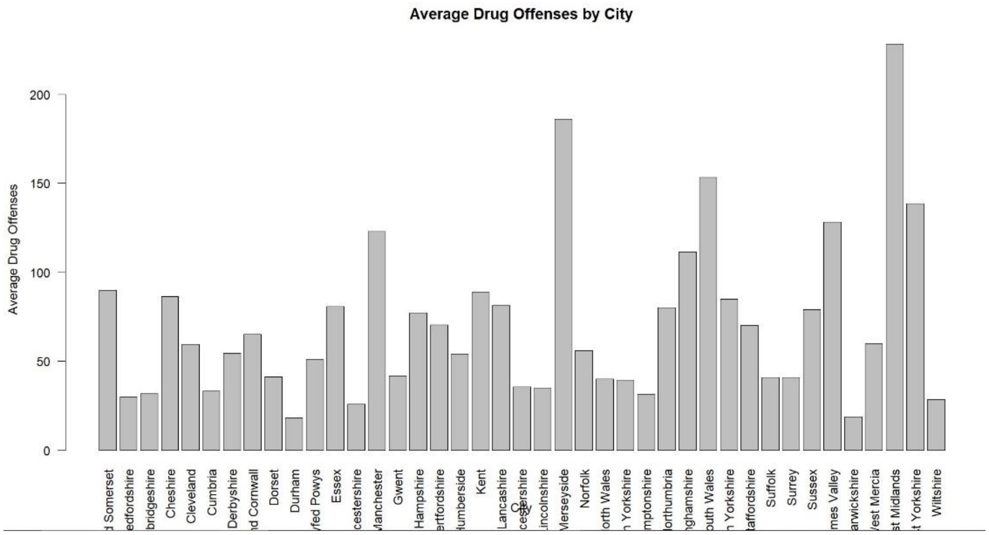
Then for EDA part we focus on data preprocessing and exploratory data analysis (EDA) for Hypothesis First, the code converts the "Month" and "Year" columns into a date format by mapping the month names to numeric values and formatting the date as "YYYY-MM-01". The cleaned data is then saved to a CSV file.

For the EDA part, the code calculates summary statistics for the "Drugs_Offences" and "Homicide" variables, including measures such as mean, median, and standard deviation. The code also calculates the correlation coefficient between these two variables using both the Pearson and Spearman methods.

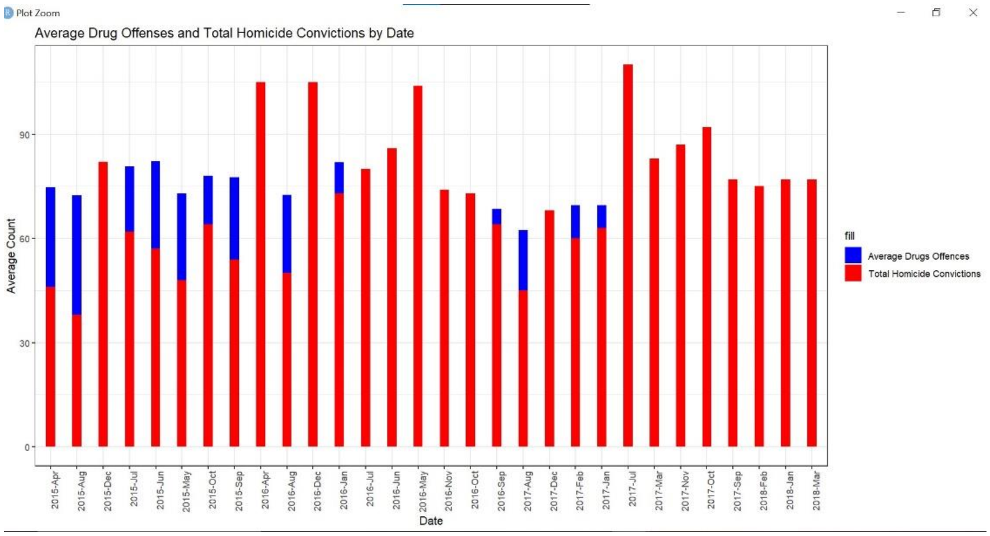
In terms of data visualization, the code creates two types of plots. First, a scatter plot is generated to visualize the relationship between "Drugs_Offences" and "Homicide" variables. This plot helps understand if there is any linear association or pattern between these variables.



Second, a bar plot is created to compare the average number of drug offenses across different cities. The plot provides a visual comparison of drug offenses among cities.



Additionally, the code generates a bar plot to analyze the average count of drug offenses and total numbers of homicide convictions over time. This plot uses the formatted date as the x-axis and displays the average count of drug offenses and homicide convictions as separate bars. The plot provides insights into any temporal trends or patterns in the data.

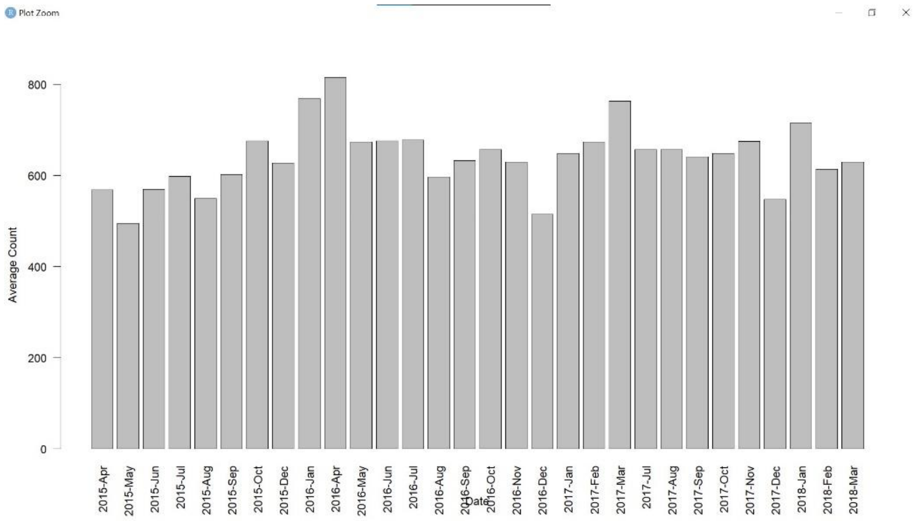


EDA for Hypothesis two:

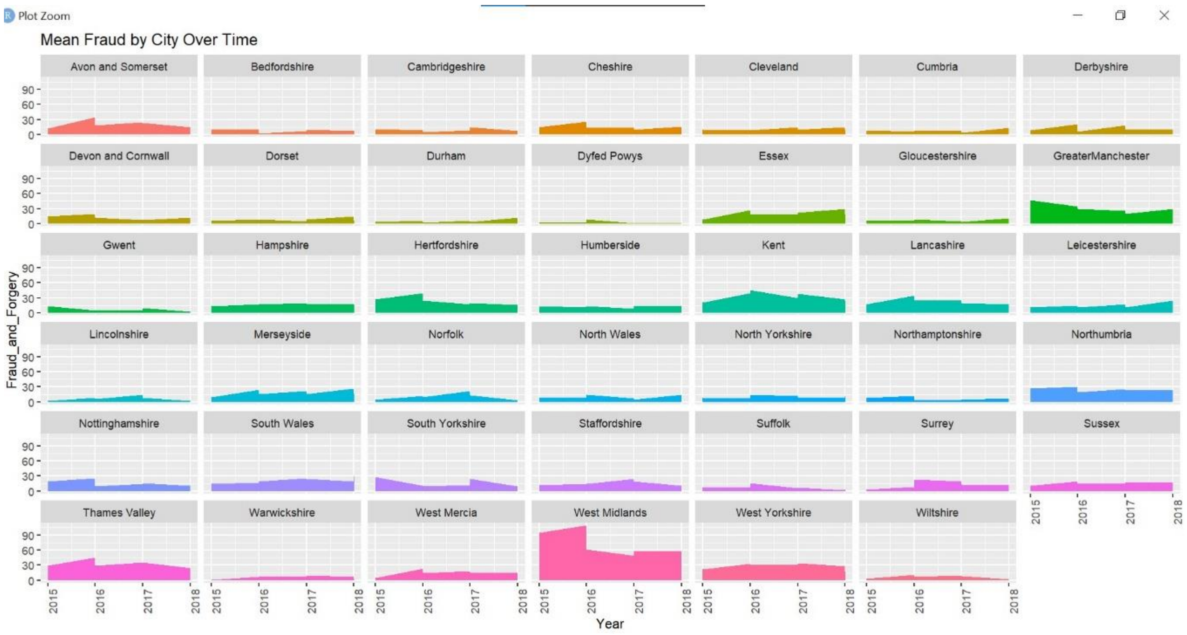
For Hypothesis two, the code focuses on data visualization for analyzing the trends and patterns of Fraud and Forgery over time. It starts by extracting the unique months and years from the dataset. Then, it calculates the mean value of Fraud and Forgery for each month and year combination and stores them in the mean_values list.

Next, the code utilizes the ggplot2 library to create a line plot showing the trend of Fraud and Forgery over time. It uses the na.omit() function to remove any missing values from the mean_df data frame, which contains the calculated mean values. The line plot is generated using the geom_line() and geom_point() functions, with the x-axis representing the month and year and the y-axis representing the mean value of Fraud and Forgery. The plot is labeled with appropriate axis labels and a title.

For the hypothesis related to Fraud and Forgery, the code further analyzes the average count and total count of Fraud and Forgery for each date. It creates two separate bar plots to visualize these measures. The mean_df list contains the average count of Fraud and Forgery for each ordered date, and the sum_df list contains the total count of Fraud and Forgery for each ordered date. Both bar plots display the date on the x-axis and the average or total count on the y-axis.In general, by using visualisation approaches, it is possible to gain a better understanding of the patterns and variations in fraud and forgery throughout time, as well as data about their frequency and intensity.



Then we used facet_wrap function. The plot consists of multiple area plots, each representing a specific city, filled with different colors to differentiate between them. The x-axis represents the years, while the y-axis represents the mean values of fraud and forgery. By facet wrapping the plot by city, it allows for easy comparison and identification of patterns specific to each city. The title "Mean Fraud by City Over Time" provides a clear context for the plot. To enhance readability, the x-axis labels have been rotated by 90 degrees. The legend is intentionally removed to reduce clutter and focus on the main insights. This visualization effectively communicates the mean fraud and forgery trends across cities over time, aiding in identifying variations and potential areas of concern.



The provided visualization depicts the mean values of fraud and forgery incidents across different cities over time. It reveals that the West Midlands had the highest rate of fraud and forgery among all cities.

However, upon analyzing the plot, we observe that there is no discernible trend in the occurrence of fraud and forgery incidents over time. Despite variations among cities, no consistent pattern emerges across the dataset. This suggests that the incidents of fraud and forgery do not exhibit a clear temporal trend.

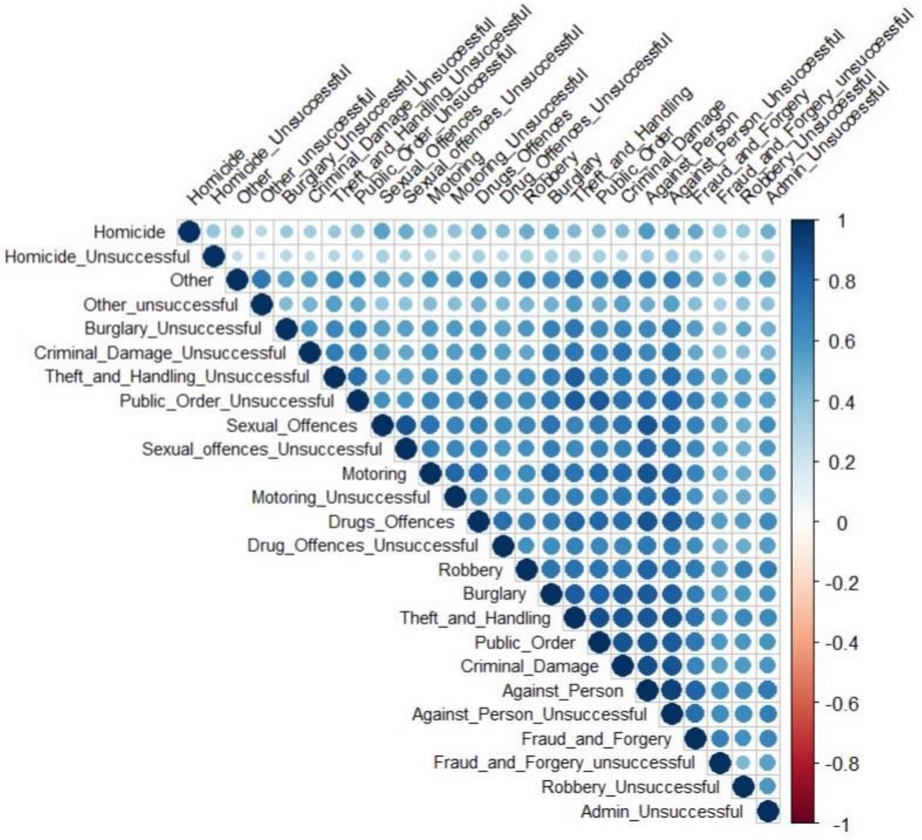
CORRELATION AND COVARIANCE:

The correlation matrix provides insights into the relationships between different variables in the dataset. Each cell in the matrix represents the correlation coefficient between two variables. The coefficient ranges from -1 to 1, indicating the strength and direction of the relationship. Positive values indicate a positive correlation, negative values indicate a negative correlation, and values close to zero indicate a weak or no correlation. By examining the correlation matrix, we can identify variables that are strongly correlated, which can provide valuable insights into potential patterns or dependencies in the data.

The correlation plot visualizes the correlation matrix using colors and allows for a more intuitive understanding of the relationships between variables. In the plot, each cell is color-coded based on the correlation coefficient, where warmer colors (e.g., red) represent positive correlations, cooler colors (e.g., blue) represent negative correlations, and lighter colors represent weaker correlations. The plot helps in identifying clusters of variables that are strongly correlated with each other. By examining the correlation plot, we can quickly identify patterns and relationships within the dataset.

The cov() function is used to compute the covariance matrix, data_cov, using the Pearson correlation method. The resulting covariance matrix provides insights into the linear relationship between pairs of variables in the dataset. Similarly, the cor() function is applied to compute the correlation matrix, data_cor, using the Pearson correlation method. The correlation matrix represents the pairwise correlation coefficients between variables in the dataset. It provides a measure of the strength and direction of the linear relationship between variables.

To visualize the correlation matrix, the code utilizes the corrplot() function from the corrplot package. This function generates a correlation plot that displays the correlation coefficients as colored squares. The plot is arranged in an upper triangular form to avoid redundancy, and the correlation coefficients are color-coded to indicate their magnitude. Additionally, the code employs the rcorr() function from the Hmisc package to calculate the significance levels of the correlation coefficients. The resulting res2 object contains the correlation coefficients (res2$r) and the corresponding p-values (res2$p), which indicate the statistical significance of the observed correlations.

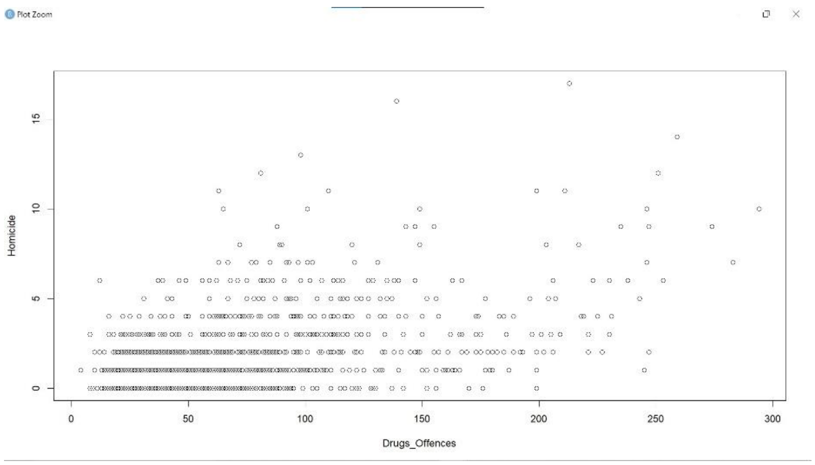


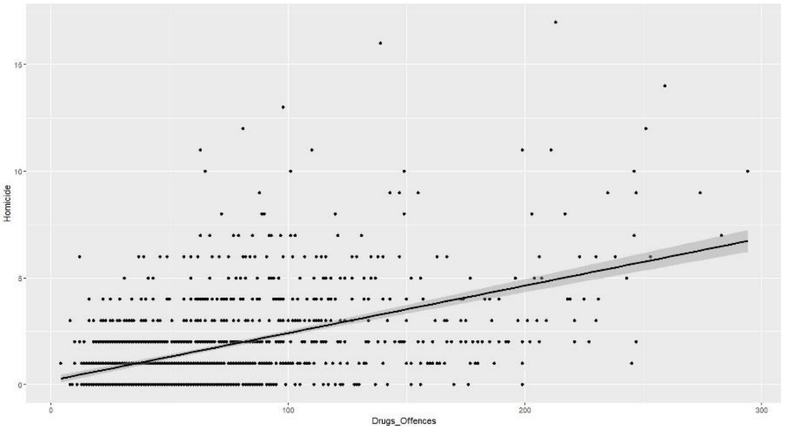
The correlation matrix plot was generated to explore the relationships between variables in the dataset. The plot used shades of blue to represent the strength and direction of the correlations. The consistent blue color across the plot suggests that the variables in the dataset exhibit weak or no significant linear associations with each other. This indicates that changes in one variable are not consistently accompanied by proportional changes in another variable. The lack of distinct color patterns or clusters implies that the variables are largely independent of each other. However, it is important to note that the absence of significant correlations does not necessarily mean there are no relationships between the variables. There may still exist non-linear or complex relationships that are not captured by this particular analysis.

Prediction Model Implementation

Linear Regression Technique:

The first part of the code demonstrates simple linear regression by fitting a linear model to the relationship between Drugs_Offences and Homicide. The lm() function is used to create the linear regression model, and the summary() function provides a summary of the regression results, including the coefficients, standard errors, t-values, and p-values. This information helps in assessing the significance of the relationship between the variables.

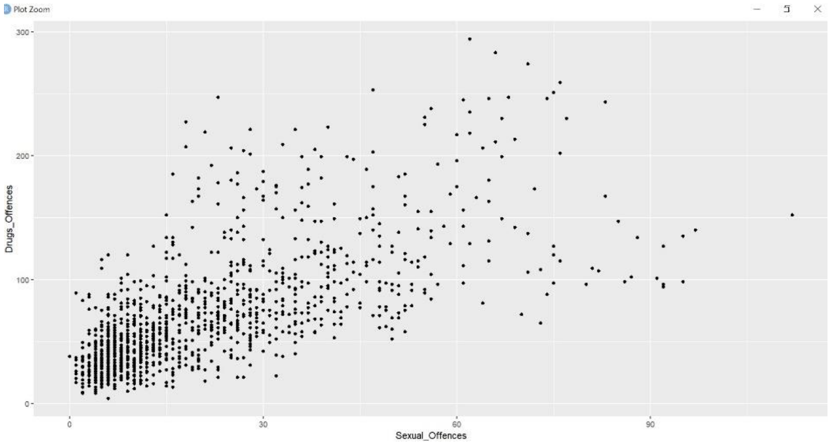




The code then extends the analysis to multiple regression by examining the relationship between Drugs_Offences, Sexual_Offences, and Against_Person. The lm() function is used again to create a multiple regression model, and the summary() function provides the regression summary. This allows us to assess the individual and combined effects of the predictors on the response variable.

To visualize the regression results, the code utilizes ggplot2. The first graph displays a scatter plot of the relationship between Drugs_Offences and Homicide, with a fitted regression line. The geom_smooth() function is used to add the regression line to the plot. The equation for the regression line is also included using the stat_regline_equation() function.

In the second graph, the relationship between Sexual_Offences and Drugs_Offences is plotted at different levels of Against_Person. The plotting.data dataframe is created to generate a sequence of values for Sexual_Offences and three levels of Against_Person. The predicted values of Drugs_Offences based on the multiple regression model are then added as a line plot using the geom_line() function. This visualization allows for a clearer understanding of the relationship between Sexual_Offences, Drugs_Offences, and Against_Person.



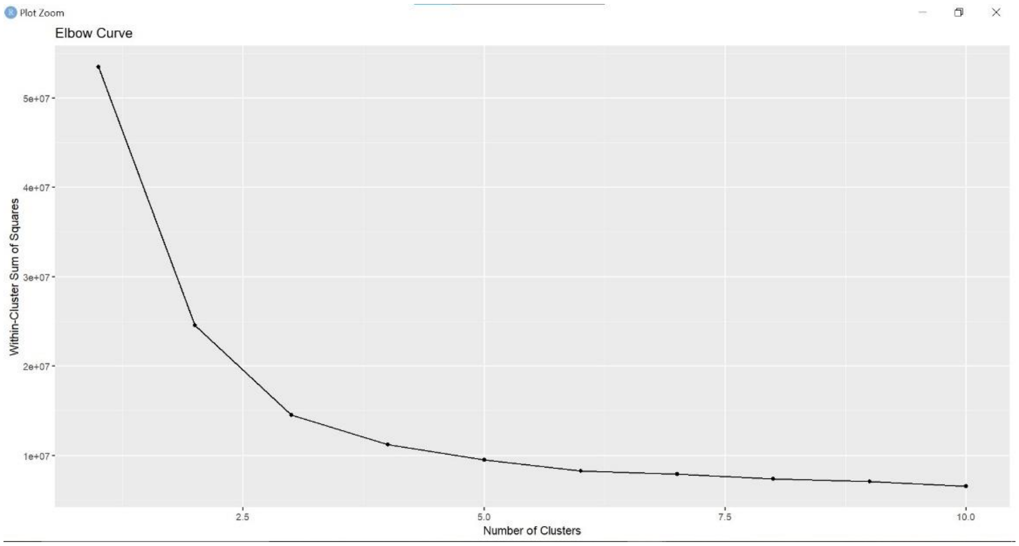
Clustering Technique:

For clustering methpod first the code performs the K-means clustering algorithm on a dataset named 'new_data' for different values of the number of clusters (k). It calculates the within-cluster sum of squares (WCSS) for each value of k, which represents the sum of squared distances between each data point and its assigned cluster centroid. The purpose of this code is to generate an elbow curve, which is a plot of the number of clusters versus the WCSS.

The elbow curve helps determine the optimal number of clusters by identifying the point where adding more clusters does not significantly reduce the WCSS. In this code, the values of k range from 1 to 10. For each value of k, the K-means algorithm is applied, and the WCSS is stored in the 'wcss' vector.

The resulting elbow curve is plotted using ggplot2, with the number of clusters (k) on the x-axis and the WCSS on the y-axis. The curve is represented by a line connecting the data points and individual points are shown as well. The x-axis is labeled as "Number of Clusters," the y-axis is labeled as "Within-Cluster Sum of Squares," and the plot is titled "Elbow Curve."

By examining the elbow curve, analysts can identify the point where the WCSS starts to level off, suggesting diminishing returns from increasing the number of clusters. This point can be considered as the optimal number of clusters for the dataset.



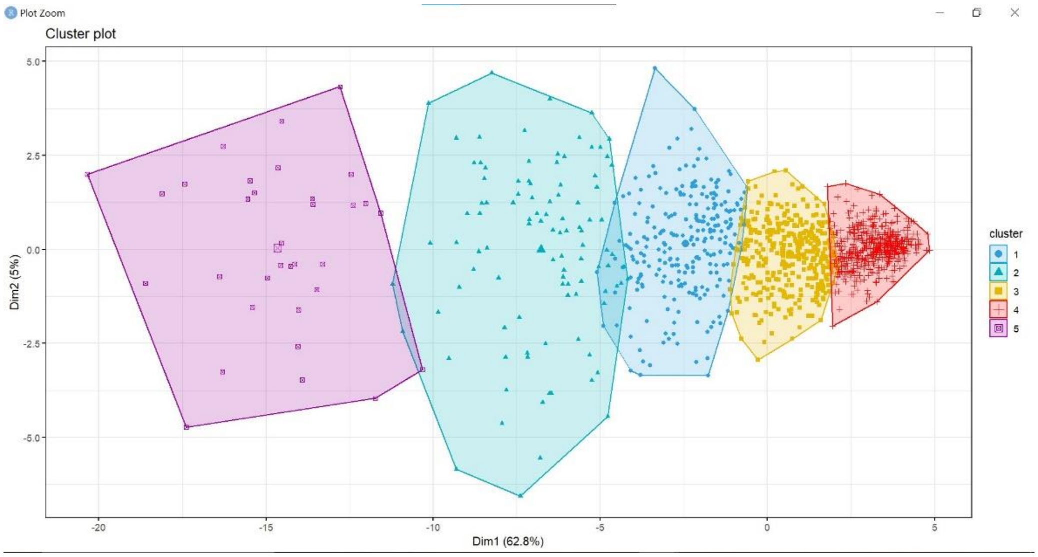
The provided code performs K-means clustering on the dataset and visualizes the resulting clusters using ggplot2 and factoextra packages.

First, the ggplot2 and factoextra packages are loaded to access the required functions for clustering and visualization. The k-means algorithm is applied to the dataset with k=5 clusters using the `kmeans` function. The `set.seed` function is used to set a seed value (123) for reproducibility of results. The result of the k-means clustering is stored in the variable `km.res`. The `print(km.res)` statement is used to print the detailed information about the k-means clustering results. It provides the cluster centers, size of each cluster, and within-cluster sum of squares. The cluster number assigned to each observation in the dataset can be accessed using `km.res$cluster`. The first four cluster assignments are displayed using the `head` function.

Next, another k-means clustering is performed on the scaled version of the dataset using `scale(new_data)` to standardize the variables. The number of clusters is set to 3, and the seed value is set to 123 for reproducibility. The result of this clustering is stored in the variable `res.km`.

The cluster assignments for each individual can be accessed using `res.km$cluster`. This provides the cluster number for each observation in the dataset.

To visualize the clusters, the `fviz_cluster` function from factoextra is used. It takes the `res.km` object as input and uses the original `new_data` dataset. The function creates a scatter plot with points representing the observations and different colors indicating the clusters. The `palette` argument specifies the colors for each cluster. Additional options such as ellipse type, theme, and plot appearance are also set.



The resulting plot shows the clusters obtained from the k-means clustering, allowing for visual interpretation of the grouping of individuals in the dataset.

Classification Technique:

For the last technique, the code performs K-Nearest Neighbors (KNN) classification on the dataset.

First, a random sample comprising 90% of the total rows in the dataset is generated using the `sample` function. The length of the sample is then calculated, which helps in determining the size of the training and testing sets.

To ensure that the variables are on a similar scale, a normalization function (`nor`) is defined. This function scales the values of each variable in the dataset to a range of 0 to 1. The normalization is applied to columns 2, 4, 6, and 18, as these are the predictor variables.

The dataset is then split into training and testing sets using the generated sample. The training set (`cleaned_data_train`) consists of rows corresponding to the sample, while the testing set (`cleaned_data_test`) comprises the remaining rows.

Next, the target category variable (16th column) is extracted from the training set (`cleaned_data_target_category`). This variable will be used as the 'cl' argument in the KNN function for classification. Similarly, the target category variable is extracted from the testing set (`cleaned_data_test_category`) to evaluate the accuracy of the model.

The `class` package is loaded to access the KNN function. The KNN algorithm is then applied to the training and testing sets using the `knn` function, with a specified value of k (13 in this case). The function predicts the categories of the testing set based on the training set and stores the results in the variable `pr`.

A confusion matrix (`tab`) is created by comparing the predicted categories (`pr`) with the actual categories (`cleaned_data_test_category`). The confusion matrix provides an overview of the classification results, showing the counts of correct and incorrect predictions for each category.

Finally, the `accuracy` function calculates the accuracy of the KNN classification model. It divides the sum of the diagonal elements (correct predictions) of the confusion matrix by the sum of all predictions to determine the overall accuracy. The accuracy is then multiplied by 100 to obtain the percentage.

Critical Review of Data Analytics Tools and Techniques

Descriptive Statistics and Correlation Analysis:

The report provides an overview of the dataset and a thorough analysis of its key features using descriptive statistics. Descriptive statistics enable a deeper understanding of the data by helping in identifying trends, outliers, and variable distributions. Correlation analysis evaluates the relationships between variables and highlights any possible connections. Finding significant predictors and learning about the data structure are both facilitated by this strategy. However, it should be noted that correlation does not inevitably imply causation.

Linear Regression Analysis:

The report employs linear regression analysis to imitate the relationships between predictor factors and the target variable. Using linear regression makes prediction easier and assists in determining the importance of variables by evaluating their impacts on the target variable. It is simpler to understand the results because visuals and a regression summary are included. The method facilitates understanding of the direction and magnitude of the interactions. It is crucial to consider the linear regression assumptions and run model diagnostics into consideration the linear regression assumptions and run model diagnostics to confirm the accuracy of the results.

Classification using K-Nearest Neighbors (KNN):

The use of the KNN algorithm for classification tasks in this report demonstrates the application of supervised machine learning techniques. KNN is a straightforward and intuitive algorithm for classification, and its implementation in the report showcases the ability to categorize data based on similarity. The inclusion of a confusion matrix allows for the evaluation of the classification accuracy. However, the effectiveness of KNN could be further enhanced by conducting hyperparameter tuning and assessing other performance metrics, such as precision, recall, or F1 score.

In evaluating the effectiveness of the techniques used in the report, it is important to consider the alignment between the chosen techniques and the research objectives. The techniques employed should directly contribute to answering the research questions or addressing the problem at hand. Furthermore, the interpretation and communication of the results are crucial to effectively convey the findings and support decision-making processes.

It should be noted that the effectiveness of data analytics techniques also depends on the quality of the dataset, appropriate preprocessing steps, and the analyst's expertise in selecting and applying the techniques. Adequate consideration should be given to the limitations and assumptions associated with each technique to ensure the reliability of the results the effectiveness of the techniques used in the report.

When considering alternative solutions to the data analytics techniques used in the report, several options come to mind. Each alternative has its own strengths and weaknesses, which should be carefully evaluated based on the specific requirements of the analysis. Let's discuss some alternative solutions and compare their merits:

1. Decision Trees:

Decision trees are a popular alternative for both descriptive and predictive analytics tasks. They provide a graphical representation of decisions and their possible consequences. Decision trees are advantageous as they are easy to understand and interpret. They can handle both categorical and numerical variables and can capture nonlinear relationships. However, decision trees are prone to overfitting and can become complex when dealing with large datasets or high-dimensional data.

2. Random Forest:

Random Forest is an ensemble learning technique that combines multiple decision trees to improve predictive accuracy and handle overfitting. It offers several benefits, including robustness to outliers, feature importance ranking, and automatic handling of missing values. Random Forest can handle large datasets and high-dimensional data effectively. However, it may be computationally expensive and can be challenging to interpret due to the complexity of the model.

3. Support Vector Machines (SVM):

SVM is a powerful technique for classification and regression tasks. It aims to find an optimal hyperplane that separates data points into different classes. SVM is effective when dealing with high-dimensional data and can handle both linear and nonlinear relationships. It is less prone to overfitting and can handle datasets with fewer samples. However, SVM can be sensitive to the choice of hyperparameters and may require careful tuning. It may also face challenges with scalability when dealing with large datasets.

Critical Review of Visualization Tools:

The use of visualization tools in data analytics is crucial for effectively communicating insights and patterns hidden within the data. In the codes provided, several visualization tools were utilized, including ggplot2, corrplot, and factoextra. Let's critically review these visualization tools and assess their effectiveness:

1. ggplot2:

ggplot2 is a powerful and versatile visualization tool that provides a wide range of plot types and customization options. It allows for the creation of visually appealing and informative plots, enabling users to effectively communicate insights from the data. The code snippet showcases the use of ggplot2 for scatter plots with regression lines and equations, which aids in understanding the relationship between variables. The ability to layer different elements and customize aesthetics provides flexibility in designing meaningful visualizations. However, ggplot2 does require a learning curve and a good understanding of its syntax, which can be challenging for beginners. Nevertheless, once mastered, ggplot2 offers great flexibility and control over plot design.

2. corrplot:

The corrplot package offers a convenient and effective way to visualize correlation matrices. The code snippet demonstrates the use of corrplot to visualize the correlation matrix of the cleaned data. Correlation matrices are essential for understanding the relationships between variables, and corrplot provides clear and concise visual representations of these relationships. It allows for customization of color schemes, labels, and other visual elements, enhancing the interpretability of the correlation matrix. However, it is important to note that corrplot may not be suitable for large correlation matrices, as the visual representation can become cluttered and difficult to interpret. Additionally, corrplot focuses solely on correlation values and may not capture other important aspects of the data relationships.

3. factoextra:

The factoextra package provides visualization tools specifically designed for exploratory multivariate data analysis, including clustering results. The code snippet utilizes the fviz_cluster() function to visualize K-means clustering results. The package offers various plot types and customization options, allowing users to effectively present and interpret clustering outcomes. Visualizations, such as scatter plots with cluster assignments, centroid plots, and biplots, facilitate the understanding of cluster patterns and enable meaningful comparisons. However, it is worth noting that factoextra is primarily focused on specific analyses like PCA and clustering, and may not have the same level of versatility as other visualization tools for general-purpose data exploration.

When exploring alternative visualization techniques, it's important to consider a variety of options to choose the most suitable one for specific data analysis needs. Here are some alternative visualization techniques and a comparison of their strengths and weaknesses:

1. Heatmaps:

Heatmaps are effective for visualizing relationships and patterns in large datasets. They use color gradients to represent values, allowing for easy identification of high and low values. Heatmaps are particularly useful for displaying correlation matrices, gene expression data, and categorical data. Their main strength lies in their ability to reveal patterns and clusters within complex datasets. However, heatmaps can become visually overwhelming for extremely large datasets and may not be suitable for datasets with a limited range of values.

2. Box Plots:

Box plots provide a visual summary of numerical data through quartiles, medians, and outliers. They are particularly useful for comparing distributions and identifying outliers. Box plots are effective in representing the spread and skewness of data, making them valuable for identifying data distribution characteristics. They are also compact and allow for easy comparison between multiple categories. However, box plots may not provide a detailed view of the underlying data distribution and may not be suitable for datasets with a large number of categories.

3. Network Graphs:

Network graphs are ideal for visualizing relationships and connections between entities. They consist of nodes (representing entities) and edges (representing relationships). Network graphs are particularly useful for social network analysis, transportation networks, and biological networks. They offer a visual representation of the complex interconnections within the data, enabling the identification of central nodes, clusters, and patterns. However, network graphs can become visually cluttered and challenging to interpret when dealing with a large number of nodes and edges.

4. Treemaps:

Treemaps are effective for visualizing hierarchical data structures. They use nested rectangles to represent categories, with the size of each rectangle proportional to a specific attribute or value. Treemaps are useful for displaying the hierarchical relationships and relative sizes of different categories within a dataset. They allow for easy comparison of category sizes and are particularly useful for visualizing hierarchical data such as file systems, organizational structures, and market segmentation. However, treemaps may not be suitable for displaying deep hierarchical structures with many levels, and the readability of labels can become an issue when dealing with small rectangles.

5. Word Clouds:

Word clouds visually represent text data by displaying words in varying sizes based on their frequency or importance. They are useful for quickly identifying the most common terms or themes in a text corpus. Word clouds offer a visually engaging and intuitive representation of textual data, making them suitable for text analysis, sentiment analysis, and topic modeling. However, word clouds do not provide quantitative information about word frequencies, and the importance of individual words can be subjective. They are best used as a supplementary visualization technique rather than a primary analysis tool.

The choice depends on the data, the analysis objectives, and the intended audience. Each alternative visualisation technique has advantages and disadvantages of its own. It is advised to experiment with many strategies before selecting the best one for conveying the needed data insights.

Conclusion

In conclusion, the analysis of the dataset, which covers 30 different months, sheds light on the connection between drugs and homicide as well as the evolution of fraud and forgeries.

The results show that there is a positive correlation between the variables in the first hypothesis, which looked at the connection between drugs and homicide. An important positive connection coefficient was found by the correlation analysis, indicating that as drug offences rose, so did the frequency of convictions for homicide. The visualizations, which showed that the overall number of homicide convictions was larger than the typical number of drug use, provide additional evidence for this conclusion. These findings suggest that drug usage and violent crime, including homicide, may be associated, highlighting the significance of addressing drug-related issues in attempts to lower violent crime.

Moving on to the second hypothesis, which looked at the rise of fraud and forgeries, the study failed to find a discernible pattern within the period under consideration. There was no consistent rising or decreasing trend in the visualisations or analysis of mean values across time for fraud and forgery events. It is important to remember that this conclusion is based on the data that is available for the months that were analysed. Instances of fraud and forgeries may exhibit various trends or patterns over a longer period or successive years. To fully understand the trend of fraud and forgeries and possibly find any underlying reasons contributing to these, more investigation and analysis over a longer time frame would be appealing.

In general, the analysis supports the positive correlation between drugs and homicide, highlighting the need for targeted interventions to address drug-related issues as a means of reducing violent crimes. However, the trend of fraud and forgery over time did not show any clear pattern within the examined period, suggesting the importance of conducting more extensive analyses over longer periods to capture potential trends and factors influencing these types of criminal activities.

Discussion of the implications and limitations of the study:

The findings of the research have some implications for understanding and responding to crimes involving drugs, homicide, fraud, and forgeries.

First, the link between drugs and homicide highlights the need for strong anti-drug policies to potentially lessen the linked violent crimes. This research shows that initiatives that address drug trafficking, reduce substance misuse, and support rehabilitation programs may help to lower the rate of homicide. These findings can be used by politicians and law enforcement to prioritize resources and create evidence-based strategies to combat drug-related crimes.

Additionally, the lack of a clear trend in fraud and forgery over the examined period highlights the complexity of these criminal activities. While no significant pattern was observed within the limited timeframe, it indicates that fraud and forgery incidents may be influenced by various factors that are not captured in the dataset. Future research should consider expanding the temporal scope and include a broader range of variables, such as economic conditions, technological advancements, and legal frameworks, to better understand the dynamics and potential trends in fraud and forgery. Such insights can guide the development of targeted prevention measures, law enforcement strategies, and public awareness campaigns to combat these financial crimes effectively.

Recognizing the study's weaknesses is crucial, though. Firstly, the analysis was based on a specific dataset covering only 30 months, which might not capture long-term trends or account for seasonal variations or outliers. The findings may not be generalizable to different regions or periods.

Moreover, the study focused on a limited set of variables, and other relevant factors influencing drugs, homicide, fraud, and forgeries, such as socio-economic factors, cultural aspects, and law enforcement policies, were not considered. Future studies should aim to incorporate a more comprehensive range of variables to provide a more nuanced understanding of these criminal activities.

Furthermore, the analysis relied on correlational analyses, which demonstrate associations but do not establish causation. While the positive correlation between drugs and homicide suggests a relationship, it does not imply a direct causal link. Establishing causality requires more in-depth research designs, including longitudinal studies or experimental approaches.

In summary, the study's implications highlight the importance of addressing drug-related issues to reduce violent crimes and the need for further investigation into the dynamics and underlying factors of fraud and forgery. However, the limitations of the study should be considered when interpreting the findings, emphasizing the need for future research to overcome these limitations and provide more robust insights into criminal activities and their prevention strategies.

References

Bhupesh, Mishra, 2023 Lectures of Data Analysis and Visualisation Principles course, University of Gloucestershire.

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). Multivariate Data Analysis (8th ed.). Cengage Learning.

Field, A., Miles, J., & Field, Z. (2012). Discovering Statistics Using R. SAGE Publications.
James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning: with Applications in R. Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.

Raschka, S., & Mirjalili, V. (2017). Python Machine Learning. Packt Publishing.
Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.
Hastie, T., Tibshirani, R., & Friedman, J. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Baesens, B., Roesch, D., & Scheule, H. (2018). Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS. John Wiley & Sons.

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). Pattern Classification (2nd ed.). Wiley-Interscience

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM Computing Surveys (CSUR), 31(3), 264-323.

Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques. Morgan Kaufmann.

Zaki, M. J., & Meira Jr, W. (2014). Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

The Crown Prosecution Service2022, https://www.cps.gov.uk/.

Shrashti,Singhal, 2020 All about Heatmaps. Retrieved on: 10th June 2023
Available at:https://towardsdatascience.com/all-about-heatmaps-bb7d97f099d7