Assessment - CT7202 Data Analytics & Visualisation Principles

Hamza Bin Riaz

S4215630

Assessment: CT7202 - Data Analytics And Visualisation Principles

Course Instructor: Dr. Bhupesh Mishra

Title: Implementation of Data Analytics Techniques on Crown Prosecution Dataframe.

Executive Summary

In this project we have performed various data analytics techniques on the Crown Prosecution data from the United Kingdom(UK) for the years 2014, 2015, 2016, and 2017. The analysis includes data cleaning, visualization, data summary, hypothesis testing, regression analysis, clustering, and classification.

Data cleaning involved handling missing values, removing irrelevant variables such as (percentage columns), adding new variables as per need of hypothesis clustering and classification techniques, and ensuring data consistency by updating names and values of the columns). Visualization techniques such as bar plots, heatmaps, scatter plots, and line graphs were employed to explore the relationships between different variables and identify trends.

The data summary provided an overview of key statistics and distributions of the variables, enabling a better understanding of the dataset. Hypothesis testing was conducted to evaluate specific claims and determine if there were significant differences or associations between variables.

Regression analysis was performed to model the relationship between crime rates and predictors such as homicide convictions, burglary convictions, and region. The analysis revealed significant associations between these predictors and crime rates, providing insights into the factors influencing crime in different areas.

Clustering analysis was conducted to group similar records based on selected crime variables. K-means clustering was employed to identify distinct clusters, allowing for a deeper understanding of crime patterns and potentially assisting in resource allocation and targeted interventions.

Finally, classification analysis was utilized to predict the future trend of crime rates in rural areas. A linear regression model was trained using historical data, and the model was then used to make predictions for future years. The predicted total crimes in rural areas were plotted over the years, aiding in forecasting and proactive decision-making.

Overall, this project provides a comprehensive analysis of the Crown Prosecution data, enabling valuable insights into crime trends, factors influencing crime rates, and predictive modeling for future crime rates in rural areas. The findings from this analysis can be utilized by law enforcement agencies, policymakers, and other stakeholders to better understand, monitor, and address crime-related challenges.

Data Overview

The dataset used for this analysis is sourced from the Crown Prosecution Service Case Outcomes by Principal Offense Category (POC) dataset, obtained from the data.gov.uk website. The dataset provides information on case outcomes from 2014 to 2017, collected on a monthly basis across forty-two (42) counties in England, where applicable.

The case outcomes are categorized into convictions and unsuccessful verdicts. Convictions include guilty pleas, trial convictions, and verdicts against defendants who did not appear in court. Unsuccessful verdicts encompass various outcomes such as discontinuances, withdrawals, discharged committals, dismissals or acquittals, and administrative finalizations.

The offenses recorded in the dataset cover a range of categories including homicide, offenses against the person such as sexual assault, burglary, robbery, theft, handling fraud or forgery, as well as criminal damage to public places and automobiles. All offenses not related to motoring are included in this category.

Data Integration

Loading and Merging Data

For this project we have taken the crown prosecution data of 4 years 2014, 2015, 2016, and 2017. In the below mentioned code, we are doing three things:

1. Defining path to the directory where our dataframe is saved in the form of multiple CSV files.
2. Merging all CSV files into a single merged CSV file.
3. Getting Year and Month from the name of files and adding them as 2 new columns into the merged dataframe.

```
# Defining path to the directory containing the csv files
data_dir <- "C:\\Users\\L380\\OneDrive\\Desktop\\DA-Assignment\\Dataset"

# Merging all files into 1 dataframe and adding 2 new columns of Year and Month
dfs <- list()
for (year in dir(data_dir)) {
  year_dir <- file.path(data_dir, year)
  for (csv_file in dir(year_dir, pattern = ".csv")) {
    month <- str_split(csv_file, "_")[[1]][4]
    year <- as.character(str_split(csv_file, "_")[[1]][5])
    year <- sub(".csv$", "", year)
    print(year)
    df <- read_csv(file.path(year_dir, csv_file))
   df <- df %>% mutate(Year = year, Month = month)
   dfs[[length(dfs) + 1]] <- df
  }
}
```

Data Pre-Processing

Data Cleaning

Sorting Rows of Dataframe on the basis of Years and Months

In the code below, we are sorting the rows(observations) of the merged dataframe in ascending order on the basis of Years and Months i.e for years, 2014, 2015, 2016, and 2017. And for months, it starts from January and goes till December.

Code

```
# Sorting All the rows of merged dataframe into ascending order as per Years and Months
final_df <- bind_rows(dfs)
final_df <- final_df %>% arrange(Year, factor(Month, levels = c("january", "february",
final_df
```

Output

	...1	Number of Homicide Convictions	Percentage of Homicide Convictions	Number of Homicide Unsuccessful	Percentage of Homicide Unsuccessful	Number of Offences Against The Person Convictions	Percentage of Offences Against The Person Convictions	Number of Offences Against The Person Unsuccessful
1	National	51	82.3%	11	17.7%	9087	75.6%	29
2	Avon and Somerset	0	-	0	-	228	78.6%	
3	Bedfordshire	0	0.0%	1	100.0%	68	70.1%	
4	Cambridgeshire	0	-	0	-	101	82.8%	
5	Cheshire	0	-	0	-	170	81.0%	
6	Cleveland	2	40.0%	3	60.0%	119	73.0%	
7	Cumbria	0	-	0	-	113	89.7%	
8	Derbyshire	0	0.0%	1	100.0%	135	69.9%	
9	Devon and Cornwall	0	-	0	-	220	77.5%	
10	Dorset	0	-	0	-	87	75.0%	

Verifying the number of rows and columns.

Code

```
#verifying the number of rows and columns

nrow(final_df)
ncol(final_df)
```

Output

```
> nrow(final_df)
[1] 1806
> ncol(final_df)
[1] 53
```

Note: Throughout the code, we have added some checks to ensure the no. of row, no. of column, and variables of the dataframe, but we will not be mentioning them over and over again in the report.

Shifting Year and Month Column in the Beginning

In the code below we are rearranging the columns of Year and Month and placing them in the beginning for the convenience in checking and analysing the dataframe.

```
#Now rearranging the columns by shifting the columns of Year and Month at the beginning

final_df <- final_df %>%
  select(1, 52, 53, 2:51)

View(final_df)
```

Removing Percentage Columns

In the code below we are removing the columns of percentage, at this point we don't need them, and the other thing is that, even if we need them afterwards, we can regenerate them.

```r
#removing the columns with percentages

final_df <- final_df %>%
  select(-matches("Percentage"))

View(final_df)
```

Fine-tuning Names of Columns

In the code below, we are replacing spaces with dots in column names because in programming it's convenient to have the attribute values without spaces. Also we are trimming any extra spaces at the start or end of column names.

```r
#replacing spaces with dots in column names + trimming any extra spaces at start or end

colnames(final_df) <- gsub("Number of", "", colnames(final_df))
colnames(final_df) <- gsub(" ", ".", trimws(colnames(final_df)))
```

Changing name of first column from …1 to County

```r
#changing name of first column to COunty

names(final_df)[names(final_df) == "...1"] <- "County"
colnames(final_df)
```

Checking Unique Values of All Variables

```
#checking unique values of each variable

colnames(final_df)
unique(final_df$Year)
unique(final_df$Month)
unique(final_df$Homicide.Convictions)
unique(final_df$Homicide.Unsuccessful)
unique(final_df$Offences.Against.The.Person.Convictions)
unique(final_df$Offences.Against.The.Person.Unsuccessful)
unique(final_df$Sexual.Offences.Convictions)
unique(final_df$Sexual.Offences.Unsuccessful)
unique(final_df$Burglary.Convictions)
unique(final_df$Burglary.Unsuccessful)
unique(final_df$Robbery.Convictions)
unique(final_df$Theft.And.Handling.Convictions)
unique(final_df$Theft.And.Handling.Unsuccessful)
unique(final_df$Fraud.And.Forgery.Convictions)
unique(final_df$Fraud.And.Forgery.Unsuccessful)
unique(final_df$Criminal.Damage.Unsuccessful)
unique(final_df$Criminal.Damage.Convictions)
unique(final_df$Drugs.Offences.Convictions)
unique(final_df$Drugs.Offences.Unsuccessful)
unique(final_df$Public.Order.Offences.Convictions)
unique(final_df$Public.Order.Offences.Unsuccessful)
unique(final_df$`All.Other.Offences.(excluding.Motoring).Convictions`)
unique(final_df$`All.Other.Offences.(excluding.Motoring).Unsuccessful`)
unique(final_df$Motoring.Offences.Convictions)
unique(final_df$Motoring.Offences.Unsuccessful)
unique(final_df$Admin.Finalised.Unsuccessful)
```

Standardizing Values in Month Column

After checking unique values of each variable, it occurred to us that values in the column if month are inconsistent, so below mentioned code we are standardising the values in the column of Month.

Code

```
# standardizing values from Months variable

month_mapping <- c(
  "january" = "January",
  "february" = "February",
  "march" = "March",
  "april" = "April",
  "may" = "May",
  "june" = "June",
  "july" = "July",
  "august" = "August",
  "september" = "September",
  "october" = "October",
  "november" = "November",
  "december" = "December",
  "Aug" = "August",
  "Dec" = "December",
  "Jul" = "July",
  "Nov" = "November",
  "Oct" = "October",
  "Sep" = "September"
)

# cleaning and standardizing the month values
final_df <- final_df %>% mutate(Month = recode(Month, !!!month_mapping))

unique(final_df$Month)
```

Output

```
> unique(final_df$Month)
 [1] "January"   "February"  "March"     "April"     "May"       "June"      "July"
 [8] "August"    "September" "October"   "November"  "December"
```

Removing Rows with "National" Crimes Data

In the code below, we are removing rows with data of national crimes as it will create a problem afterwards, like when we will calculate mean or do any other calculation on the basis of crimes and counties.

Also checking all the columns that we have at the moment, just a check to analyze if the process is going smooth.

Code

```
#removing rows with the data of national crimes
final_df <- subset(final_df, !grepl("National", County))

nrow(final_df)
ncol(final_df)

View(final_df)
```

Output

After removing the percentage columns, and removing the rows with the data of crimes taking place at national level, we are left with 1764 rows and 28 columns. Initially we had 1806 rows and 53 columns.

```
> nrow(final_df)
[1] 1764
> ncol(final_df)
[1] 28
```

Adding a New Column of Region

In the code, below, we are adding a new column named Region, in which we are marking all the counties as Rural or Urban. This will help us in the upcoming techniques like hypothesis testing, regression analysis, clustering, and classification. Apart from adding this new column, we are shifting it to the beginning of the dataframe so that it's visible and easy to analyze.

Code

```
# defining the list of rural counties
rural_counties <- c("Cumbria", "Dorset", "Durham", "Dyfed Powys", "Gloucestershire",
                    "Lincolnshire", "Norfolk", "North Wales", "North Yorkshire",
                    "South Wales", "Suffolk", "Warwickshire", "Wiltshire")

# adding a column named "Region" indicating rural or urban areas based on the county
final_df$Region <- ifelse(final_df$County %in% rural_counties, "Rural", "Urban")

View(final_df)

#shifting "Region" column to second index after "County"

final_df <- final_df %>%
  select(1, 29, 2:28)

View(final_df)
```

Output

County	Region	Year	Month	Homicide.Convictions	Homicide.Unsuccessful	Offences.Against.The.Person.Convictions
Avon and Somerset	Urban	2014	January	0	0	228
Bedfordshire	Urban	2014	January	0	1	68
Cambridgeshire	Urban	2014	January	0	0	101
Cheshire	Urban	2014	January	0	0	170
Cleveland	Urban	2014	January	2	3	119
Cumbria	Rural	2014	January	0	0	113
Derbyshire	Urban	2014	January	0	1	135
Devon and Cornwall	Urban	2014	January	0	0	220
Dorset	Rural	2014	January	0	0	87
Durham	Rural	2014	January	1	0	105
Dyfed Powys	Rural	2014	January	0	0	69
Essex	Urban	2014	January	3	0	331
Gloucestershire	Rural	2014	January	0	0	71
GreaterManchester	Urban	2014	January	2	1	478
Gwent	Urban	2014	January	0	0	85
Hampshire	Urban	2014	January	0	0	303
Hertfordshire	Urban	2014	January	3	0	113
Humberside	Urban	2014	January	0	1	217
Kent	Urban	2014	January	0	0	230

column 6: numeric with range 0 - 14

Data Exploration and Visualisation

Data Summary

Using the defaul;t function of summary() we have checked the mean, median, modes and quartiles of all of our variables, but we will only discuss those in the report that are used afterwards. Our mainly used variables are:

- Homicide.Convictions
- Burglary.Convictions
- Offences.Against.The.Person.Convictions

```
Homicide.Convictions
Min.    : 0.000
1st Qu.: 0.000
Median : 1.000
Mean    : 1.882
3rd Qu.: 2.000
Max.    :38.000
```

```
Burglary.Convictions
Min.    :  1.00
1st Qu.: 14.00
Median : 23.00
Mean    : 31.81
3rd Qu.: 37.00
Max.    :278.00
```

```
Offences.Against.The.Person.Convictions
Min.    :  29.0
1st Qu.: 114.0
Median : 175.0
Mean    : 234.0
3rd Qu.: 265.2
Max.    :1904.0
```

Glimpse

Glimpse is a default function that tells us about the data type and values of each variable, below is the output for that function.

```
> glimpse(final_df)
Rows: 1,764
Columns: 29
$ County                                                <chr> "Avon and Somerset", "Bedfor…
$ Region                                                <chr> "Urban", "Urban", "Urban", "…
$ Year                                                  <chr> "2014", "2014", "2014", "201…
$ Month                                                 <chr> "January", "January", "Janua…
$ Homicide.Convictions                                  <dbl> 0, 0, 0, 0, 2, 0, 0, 0, 0, 1…
$ Homicide.Unsuccessful                                 <dbl> 0, 1, 0, 0, 3, 0, 1, 0, 0, 0…
$ Offences.Against.The.Person.Convictions               <dbl> 228, 68, 101, 170, 119, 113,…
$ Offences.Against.The.Person.Unsuccessful              <dbl> 62, 29, 21, 40, 44, 13, 58, …
$ Sexual.Offences.Convictions                           <dbl> 35, 2, 10, 15, 11, 4, 22, 16…
$ Sexual.Offences.Unsuccessful                          <dbl> 17, 1, 3, 1, 6, 3, 7, 3, 7, …
$ Burglary.Convictions                                  <dbl> 49, 7, 18, 38, 36, 16, 36, 2…
$ Burglary.Unsuccessful                                 <dbl> 1, 4, 4, 5, 2, 0, 4, 4, 1, 4…
$ Robbery.Convictions                                   <dbl> 8, 16, 6, 10, 3, 1, 5, 10, 6…
$ Robbery.Unsuccessful                                  <dbl> 0, 7, 4, 0, 2, 0, 2, 0, 0, 0…
$ Theft.And.Handling.Convictions                        <dbl> 338, 75, 148, 205, 334, 115,…
$ Theft.And.Handling.Unsuccessful                       <dbl> 32, 4, 15, 5, 31, 7, 17, 23,…
$ Fraud.And.Forgery.Convictions                         <dbl> 18, 17, 10, 14, 11, 6, 15, 1…
$ Fraud.And.Forgery.Unsuccessful                        <dbl> 0, 3, 4, 1, 3, 0, 0, 2, 0, 2…
$ Criminal.Damage.Convictions                           <dbl> 93, 22, 30, 39, 46, 38, 53, …
$ Criminal.Damage.Unsuccessful                          <dbl> 14, 8, 3, 3, 13, 3, 12, 9, 4…
$ Drugs.Offences.Convictions                            <dbl> 148, 31, 47, 64, 65, 52, 86,…
$ Drugs.Offences.Unsuccessful                           <dbl> 4, 3, 1, 3, 2, 1, 9, 5, 5, 1…
$ Public.Order.Offences.Convictions                     <dbl> 123, 30, 37, 77, 123, 78, 59…
$ Public.Order.Offences.Unsuccessful                    <dbl> 28, 9, 2, 8, 27, 2, 15, 11, …
$ `All.Other.Offences.(excluding.Motoring).Convictions` <dbl> 63, 13, 28, 50, 34, 52, 28, …
$ `All.Other.Offences.(excluding.Motoring).Unsuccessful` <dbl> 9, 2, 9, 5, 14, 5, 12, 14, 1…
$ Motoring.Offences.Convictions                         <dbl> 256, 171, 103, 264, 228, 112…
$ Motoring.Offences.Unsuccessful                        <dbl> 40, 13, 16, 16, 16, 7, 9, 31…
$ Admin.Finalised.Unsuccessful                          <dbl> 20, 12, 14, 13, 3, 12, 13, 1…
>
```

Data Exploration Report

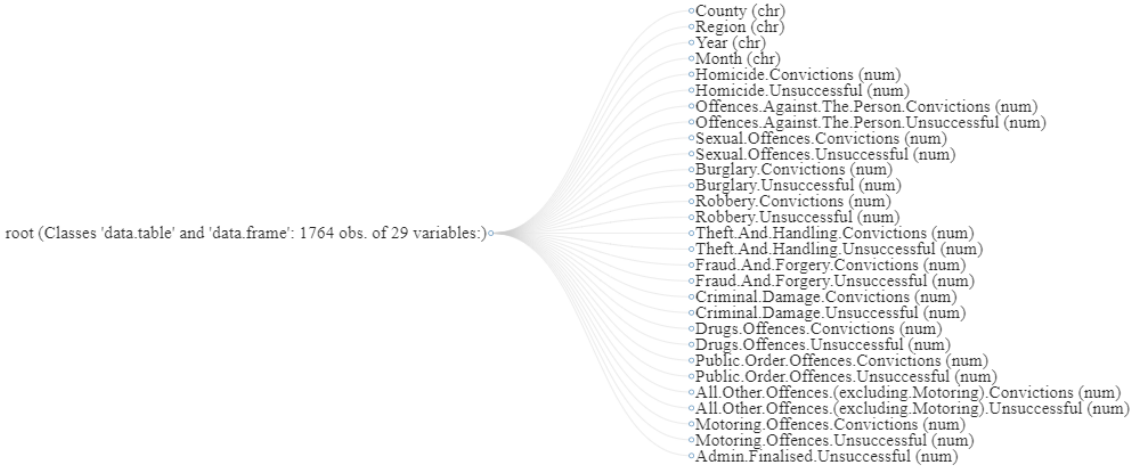
Now we are generating data exploration report using a default function and the code for that is below:

Code

```
DataExplorer::create_report(final_df)
```

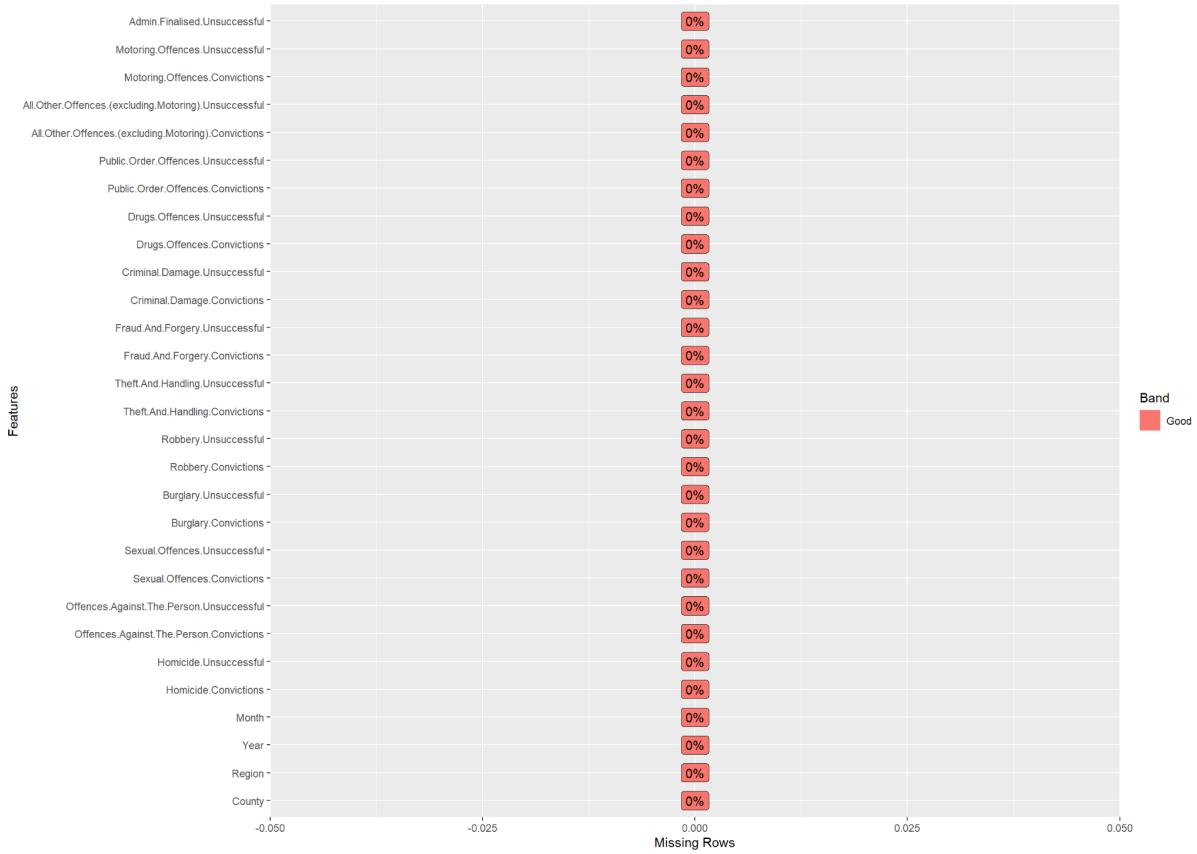
Output

Checking all the Variables and their Types



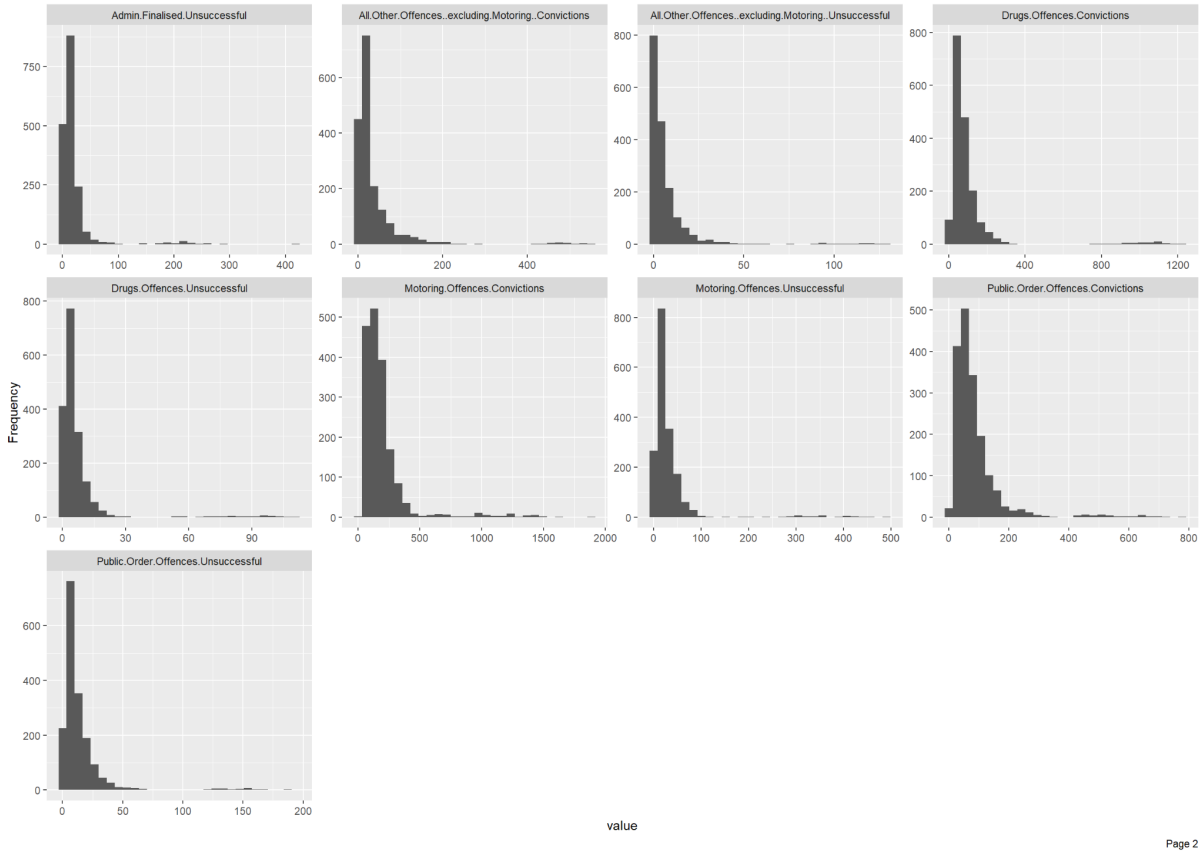
Checking Missing Values in All Variables

Image below represents that there are no missing values in any row of the dataset.



Univariate Distribution of All Variables Using Histogram

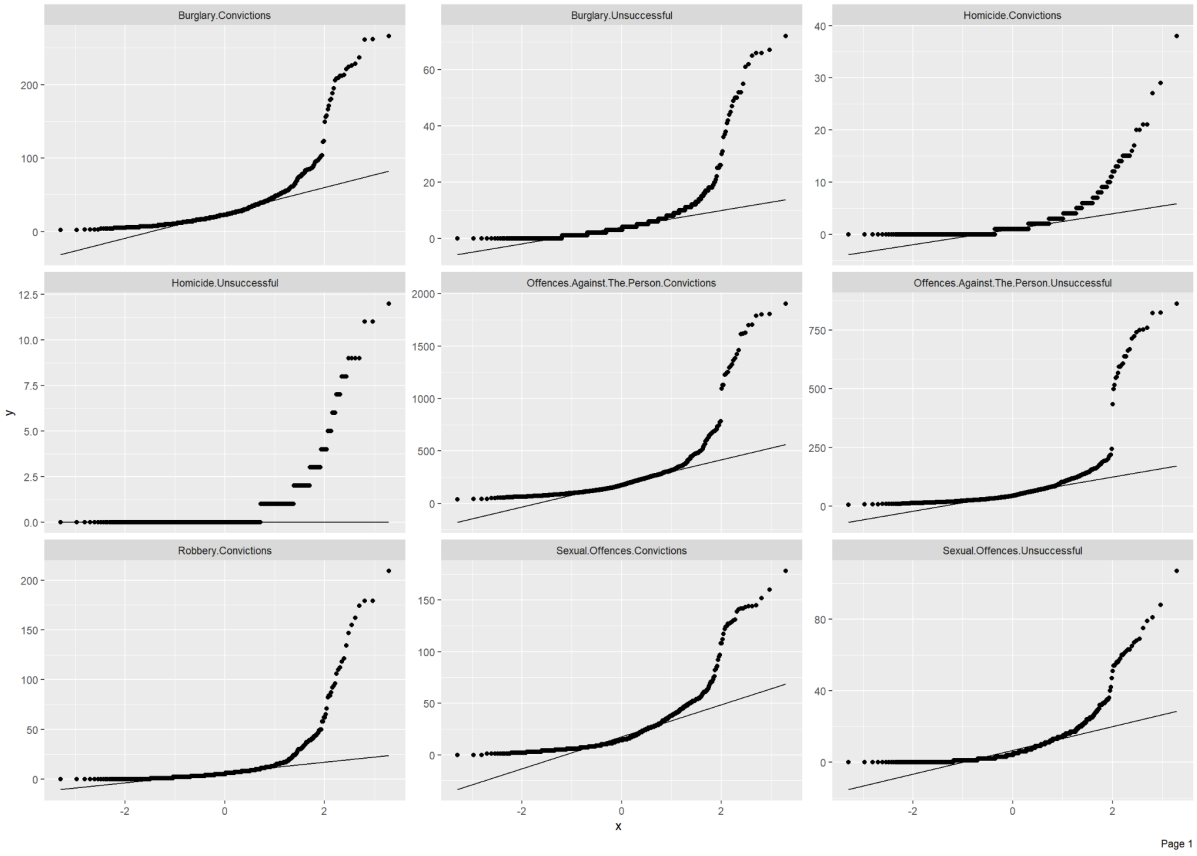
Univariate distribution refers to the distribution of a single variable or feature in a dataset. It provides insights into the distributional properties of that variable, such as its central tendency, spread, and shape. Analyzing the univariate distribution helps understand the characteristics and patterns of the variable independently of other variables.



Theoretical Distribution of Variables Using QQ Plot

We have used QQ plots (Quantile-Quantile plot) to assess if a dataset follows a particular theoretical distribution, such as the normal distribution. It is commonly used in statistical analysis to compare the quantiles of the observed data against the quantiles expected under a specific theoretical distribution. We have analysed the following variables like Burglary, Homicide, Offences against the person, Robbery and Sexual Offences. From the graphs below we can depict following:

- We don’t have straight lines so the variables have values that are skewed either negatively or positively.
- Most of the points are close to the straight line, so the majority of the values are normal and have a similar trend.
- Some values are far away from the straight line, which means that we do have some outliers in each variable.



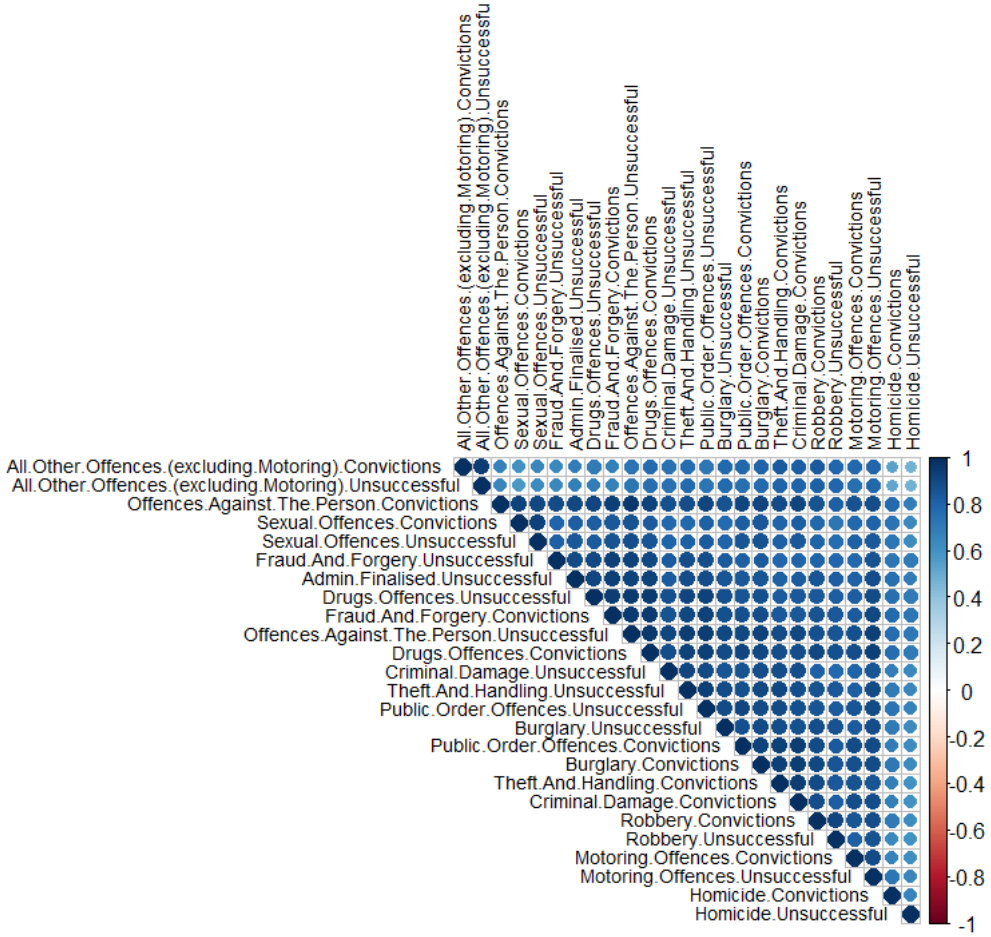
Correlation of All Variables of Dataframe Using Heatmap

The correlation heatmap of your criminal convictions dataframe depicts the pairwise correlations between different variables in the dataset. It shows the strength and direction of the relationships between the variables, as the correlation value lies between -1 to +1. If the value is from 0 to 1 it means that the correlation is positive and if the value is from 0 to -1, it shows that the correlation between the variables is negative. +1 means perfect correlation in positive direction and -1 means fully correlated variables but in opposite direction as one is increasing and other is decreasing. The heatmap helps in identifying which variables are positively or negatively correlated with each other. This information can be useful in understanding the interdependencies between different types of criminal convictions and exploring potential patterns or associations in the data.

Code

```
#finding thr correlation for all the crime columns of the dataframe

cor_matrix <- cor(final_df[5:29])

  corrplot(cor_matrix, type = "upper", order = "hclust", tl.cex = 0.7, tl.col = "black",
           is.corr = TRUE, mar = c(0, 0, 0, 0))
  heatmap(cor_matrix,
          col = colorRampPalette(c("blue", "white", "red"))(100),
          main = "Correlation Heatmap")
```

Output



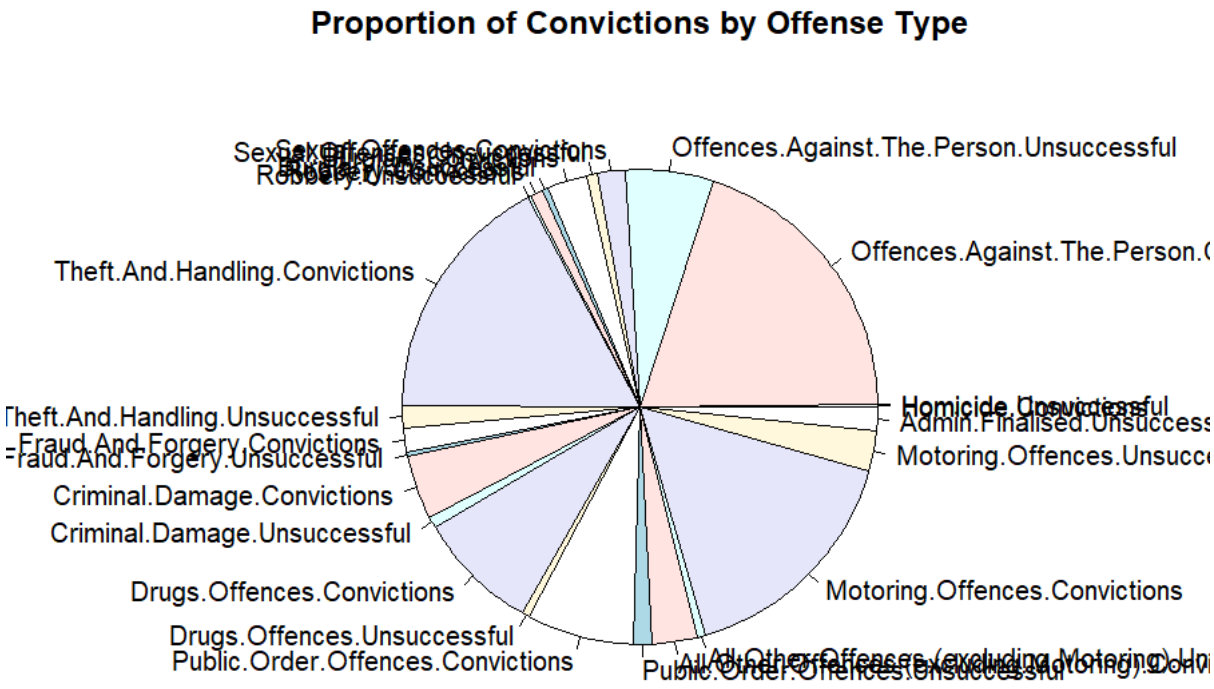
Showing Crime Proportion Using Pie Chart

Mentioned below is the pie chart that depicts the proportion of each crime, as we can clearly see the top 3 crimes are: Offences against the person, Motoring Offence Convictions, and Theft and Handling Convictions.

Code

```
# Showing proportion of convictions by Offense Type
convictions_by_offense <- final_df[, c(5:29)]
convictions_total <- colSums(convictions_by_offense)
pie(convictions_total, labels = names(convictions_total),
    main = "Proportion of Convictions by Offense Type")
```

Output



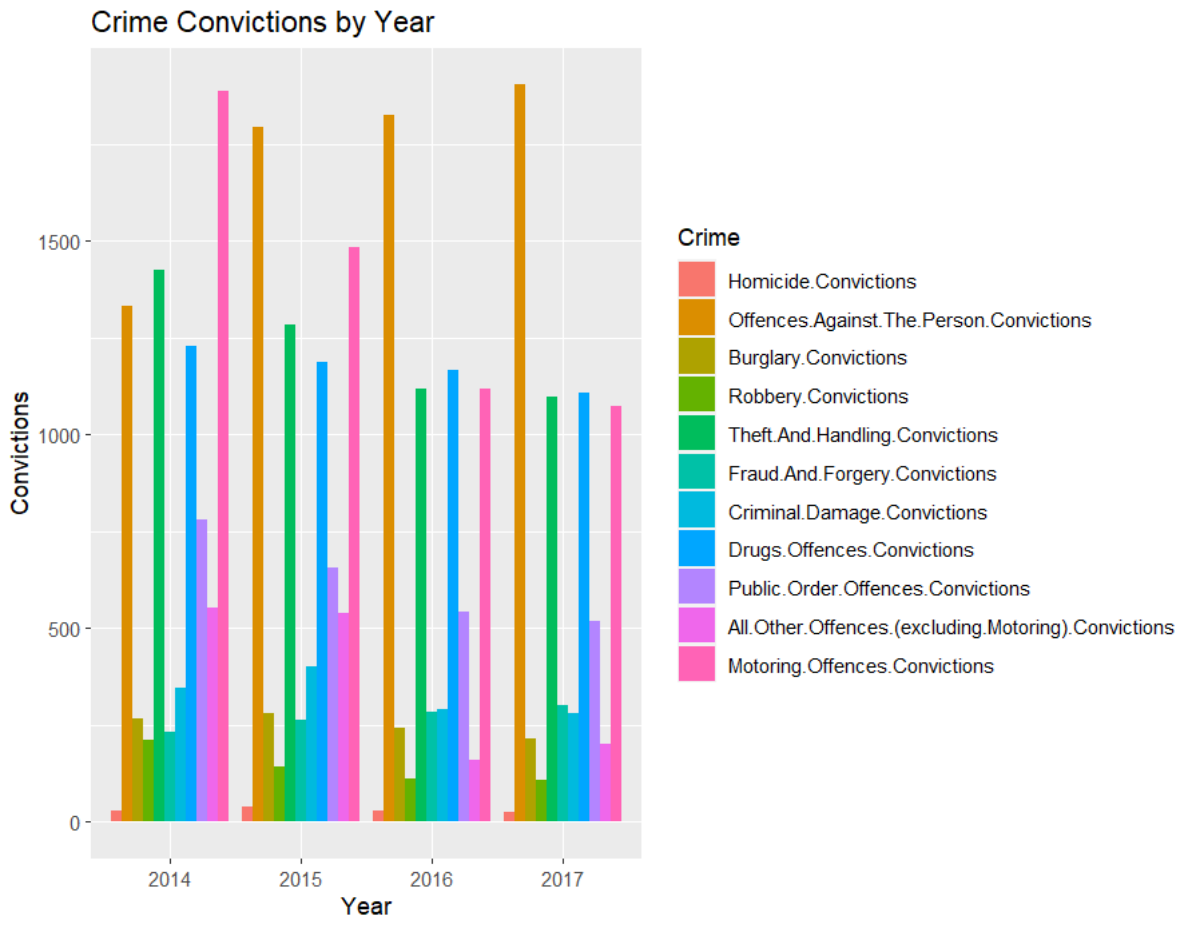
Successful Crimes Trend Over the Years

In the code and the histogram below, we are showing the Successful convictions over the years, 2014, 2015, 2016, and 2017. By the use of this histogram we can depict the following:

Top crimes are Motoring offence, offence against the person, theft and handling, and drugs offences.
Trend of crimes is not completely same but to some extent its same as during all 4 years, each crime is taking place at a certain pace, there is no big increase or decrease.

By use of this data concerned individuals can make strategies and prioritise the efforts against the crimes.

Output- Histogram



Code

```r
# successful crimes' graph over the years

# selecting the relevant variables for the years 2014, 2015, 2017, and 2018
selected_years <- c("2014", "2015", "2016", "2017")
s_crimes_df <- c("Homicide.Convictions", "Offences.Against.The.Person.Convictions",
                 "Burglary.Convictions", "Robbery.Convictions", "Theft.And.Handling.Cor
                 "Fraud.And.Forgery.Convictions", "Criminal.Damage.Convictions",
                 "Drugs.Offences.Convictions", "Public.Order.Offences.Convictions",
                 "All.Other.Offences.(excluding.Motoring).Convictions", "Motoring.Offer

# cresting subset of dataframe for selected years and variables
subset_df <- final_df[final_df$Year %in% selected_years, c("Year", s_crimes_df)]

# reshaping the data from wide to long format for plotting
subset_df_long <- reshape2::melt(subset_df, id.vars = "Year", variable.name = "Crime",
                                 value.name = "Convictions")

# creating a grouped bar chart
ggplot(subset_df_long, aes(x = Year, y = Convictions, fill = Crime)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Crime Convictions by Year",
       x = "Year", y = "Convictions", fill = "Crime") +
  theme(legend.position = "right")
```

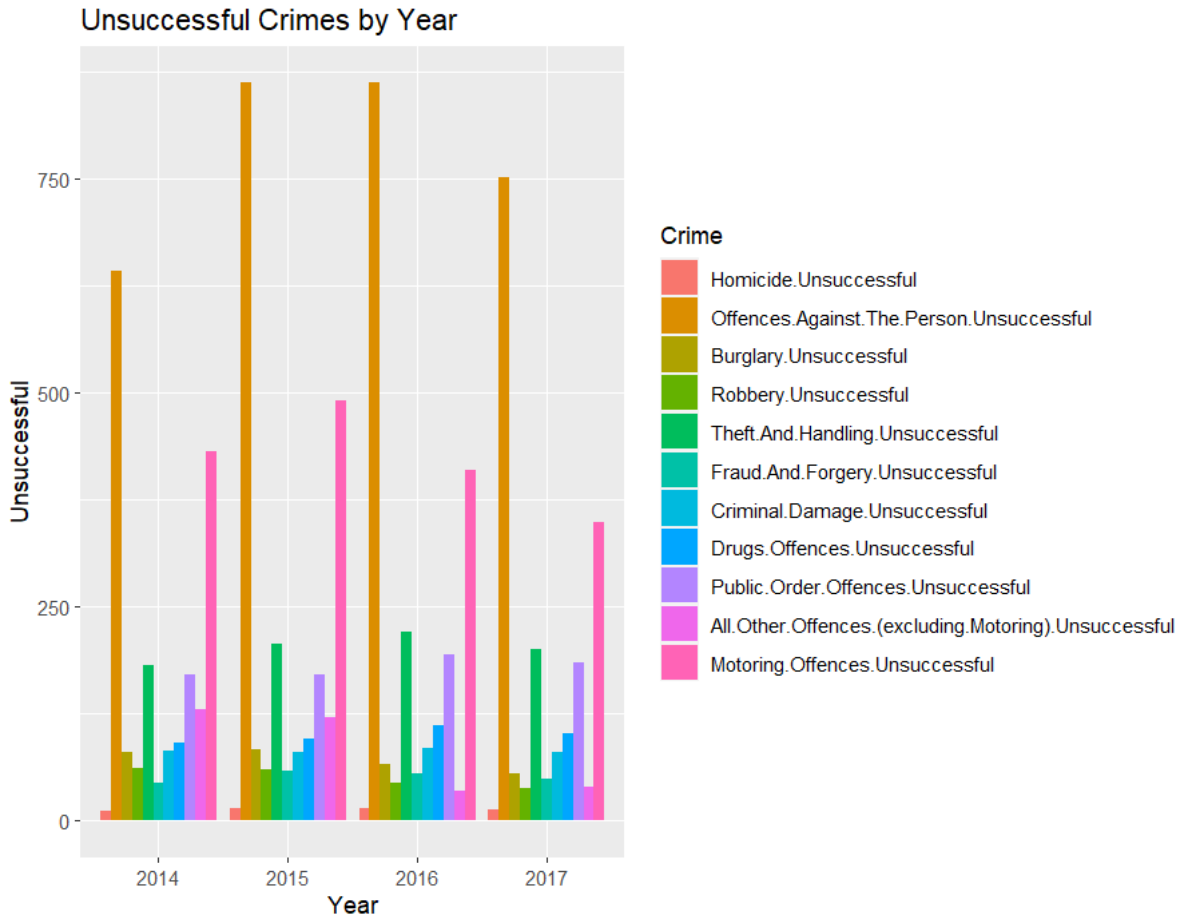
Unsuccessful Crime Trend Over The Years

In the code and the histogram below, we are showing the unsuccessful convictions over the years, 2014, 2015, 2016, and 2017. By the use of this histogram we can depict the following:

Most unsuccessful crimes are Motoring offense, offense against the person
Trend of unsuccessful crimes is not completely same but to some extent it's the same as during all 4 years, each crime is taking place at a certain pace, there is no big increase or decrease.

By use of this data concerned individuals can make strategies and prioritize the efforts against the crimes.

Output - Histogram



Code

```
# unsuccessful crimes over the years

us_crimes_df <- c("Homicide.Unsuccessful", "Offences.Against.The.Person.Unsuccessful",
                  "Burglary.Unsuccessful", "Robbery.Unsuccessful", "Theft.And.Handling.U
                  "Fraud.And.Forgery.Unsuccessful", "Criminal.Damage.Unsuccessful",
                  "Drugs.Offences.Unsuccessful", "Public.Order.Offences.Unsuccessful",
                  "All.Other.Offences.(excluding.Motoring).Unsuccessful", "Motoring.Offe

subset_df <- final_df[, c("Year", us_crimes_df)]

subset_df_long <- reshape2::melt(subset_df, id.vars = "Year", variable.name = "Crime", va

ggplot(subset_df_long, aes(x = Year, y = Unsuccessful, fill = Crime)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Unsuccessful Crimes by Year",
       x = "Year", y = "Unsuccessful", fill = "Crime") +
  theme(legend.position = "right")
```

Creating Separate Dataframes for Successful & UnSuccessful Crimes

Creating separate dataframes for successful and unsuccessful ccrimes

Code

```
# Creating a dataframe for successful crimes
successful_crimes_df <- final_df[, c("County", "Year", "Month", "Homicide.Convictions", "Offences.Against.The.Person.Convictions",
                                     "Sexual.Offences.Convictions","Burglary.Convictions", "Robbery.Convictions",
                                     "Theft.And.Handling.Convictions", "Fraud.And.Forgery.Convictions", "Criminal.Damage.Convictions",
                                     "Drugs.Offences.Convictions", "Public.Order.Offences.Convictions",
                                     "All.Other.Offences.(excluding.Motoring).Convictions",
                                     "Motoring.Offences.Convictions")]
View(successful_crimes_df)

colnames(successful_crimes_df)
```

Output

```
> colnames(successful_crimes_df)
 [1] "County"
 [2] "Year"
 [3] "Month"
 [4] "Homicide.Convictions"
 [5] "Offences.Against.The.Person.Convictions"
 [6] "Sexual.Offences.Convictions"
 [7] "Burglary.Convictions"
 [8] "Robbery.Convictions"
 [9] "Theft.And.Handling.Convictions"
[10] "Fraud.And.Forgery.Convictions"
[11] "Criminal.Damage.Convictions"
[12] "Drugs.Offences.Convictions"
[13] "Public.Order.Offences.Convictions"
[14] "All.Other.Offences.(excluding.Motoring).Convictions"
[15] "Motoring.Offences.Convictions"
```

Code

```
# Creating a dataframe for unsuccessful crimes
unsuccessful_crimes_df <- final_df[, c("County", "Year", "Month", "Homicide.Unsuccessful", "Offences.Against.The.Person.Unsuccessful",
                                       "Sexual.Offences.Unsuccessful",
                                       "Burglary.Unsuccessful", "Robbery.Unsuccessful", "Theft.And.Handling.Unsuccessful",
                                       "Fraud.And.Forgery.Unsuccessful", "Criminal.Damage.Unsuccessful", "Drugs.Offences.Unsuccessful",
                                       "Public.Order.Offences.Unsuccessful", "All.Other.Offences.(excluding.Motoring).Unsuccessful",
                                       "Motoring.Offences.Unsuccessful")]


View(unsuccessful_crimes_df)
colnames(unsuccessful_crimes_df)
```

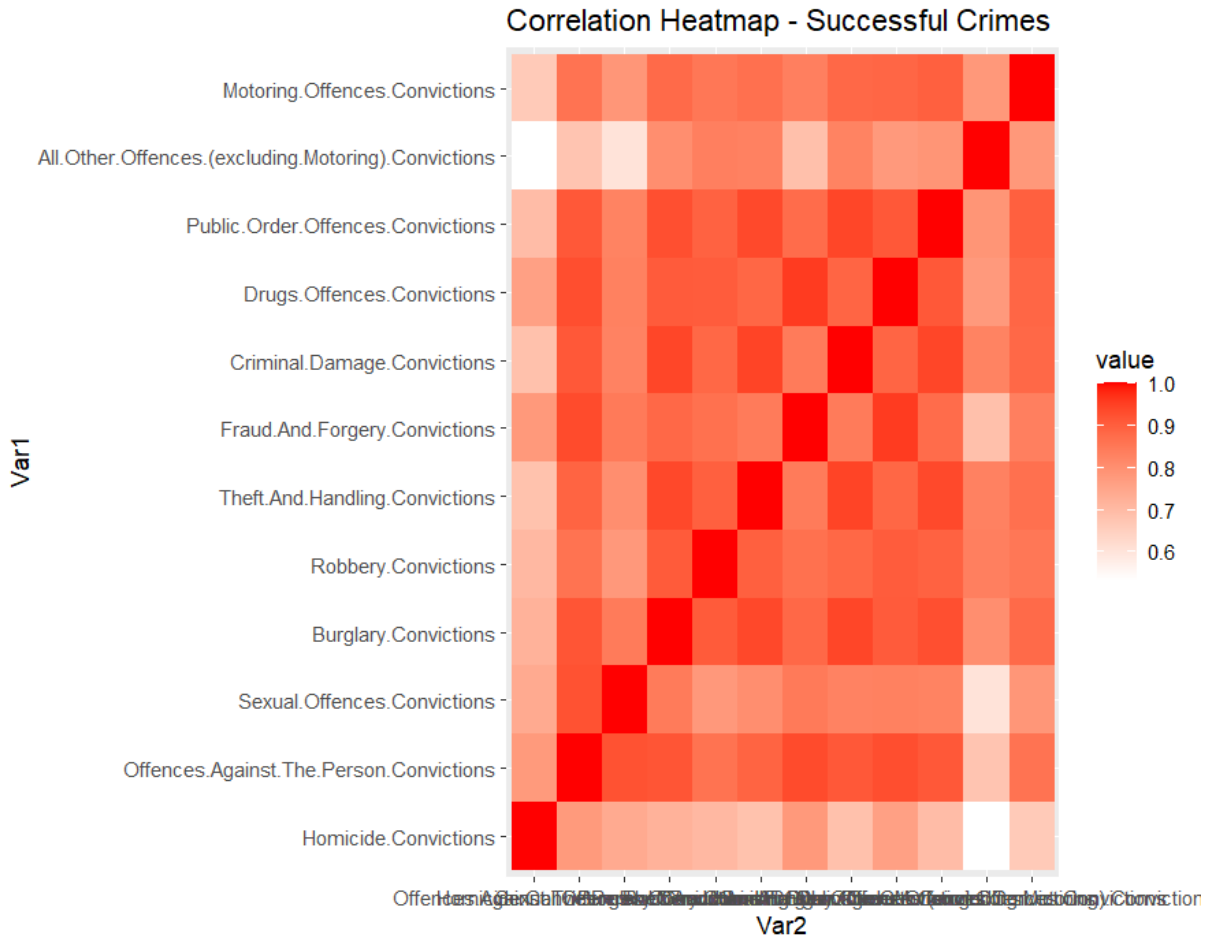
Output

```
> colnames(unsuccessful_crimes_df)
 [1] "County"
 [2] "Year"
 [3] "Month"
 [4] "Homicide.Unsuccessful"
 [5] "Offences.Against.The.Person.Unsuccessful"
 [6] "Sexual.Offences.Unsuccessful"
 [7] "Burglary.Unsuccessful"
 [8] "Robbery.Unsuccessful"
 [9] "Theft.And.Handling.Unsuccessful"
[10] "Fraud.And.Forgery.Unsuccessful"
[11] "Criminal.Damage.Unsuccessful"
[12] "Drugs.Offences.Unsuccessful"
[13] "Public.Order.Offences.Unsuccessful"
[14] "All.Other.Offences.(excluding.Motoring).Unsuccessful"
[15] "Motoring.Offences.Unsuccessful"
```

Computing and plotting correlation for successful crimes

```
#computing correlation matrix for successful crimes
successful_corr <- cor(successful_crimes_df[, 4:15])
print("Correlation Between Successful Crimes: ")
print(successful_corr)
```

```
#plotting correlation heatmap for successful crimes
ggplot(successful_corr_long, aes(Var2, Var1, fill = value)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Correlation Heatmap - Successful Crimes")
```



Computing Covariance for Successful Crimes DF

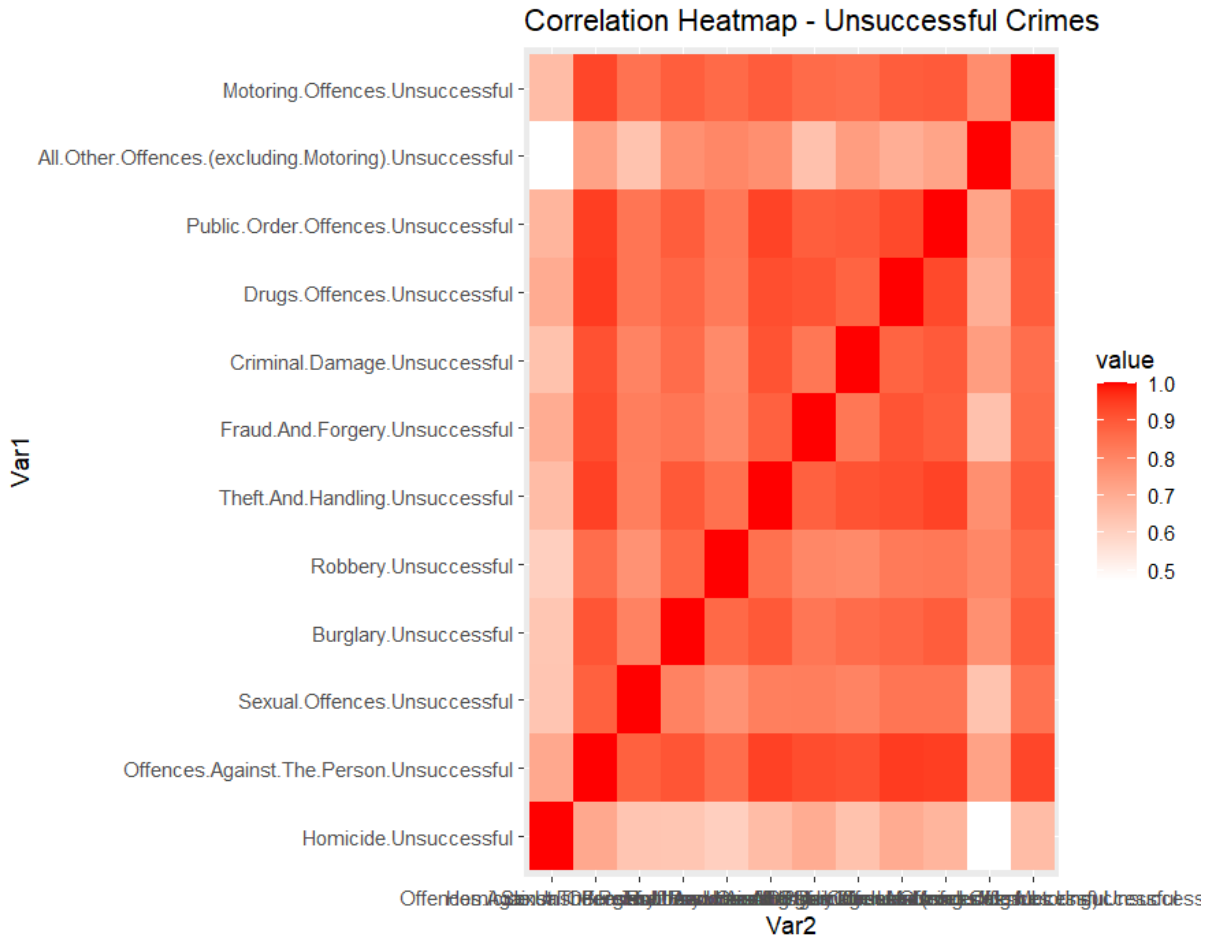
```
#computing covariance matrix for successful crimes
successful_cov <- cov(successful_crimes_df[, 4:15])
print("Corvariance Between Successful Crimes: ")
print(successful_cov)
```

```
#plotting covariance heatmap for successful crimes
ggplot(successful_cov_long, aes(Var2, Var1, fill = value)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(title = "Covariance Heatmap - Successful Crimes")
```

Computing and Plotting Correlation Heatmap for Unsuccessful Crimes DF

```
#computing correlation matrix for unsuccessful crimes
unsuccessful_corr <- cor(unsuccessful_crimes_df[, 4:15])
print("Correlation Between Unsuccessful Crimes: ")
print(unsuccessful_corr)
```

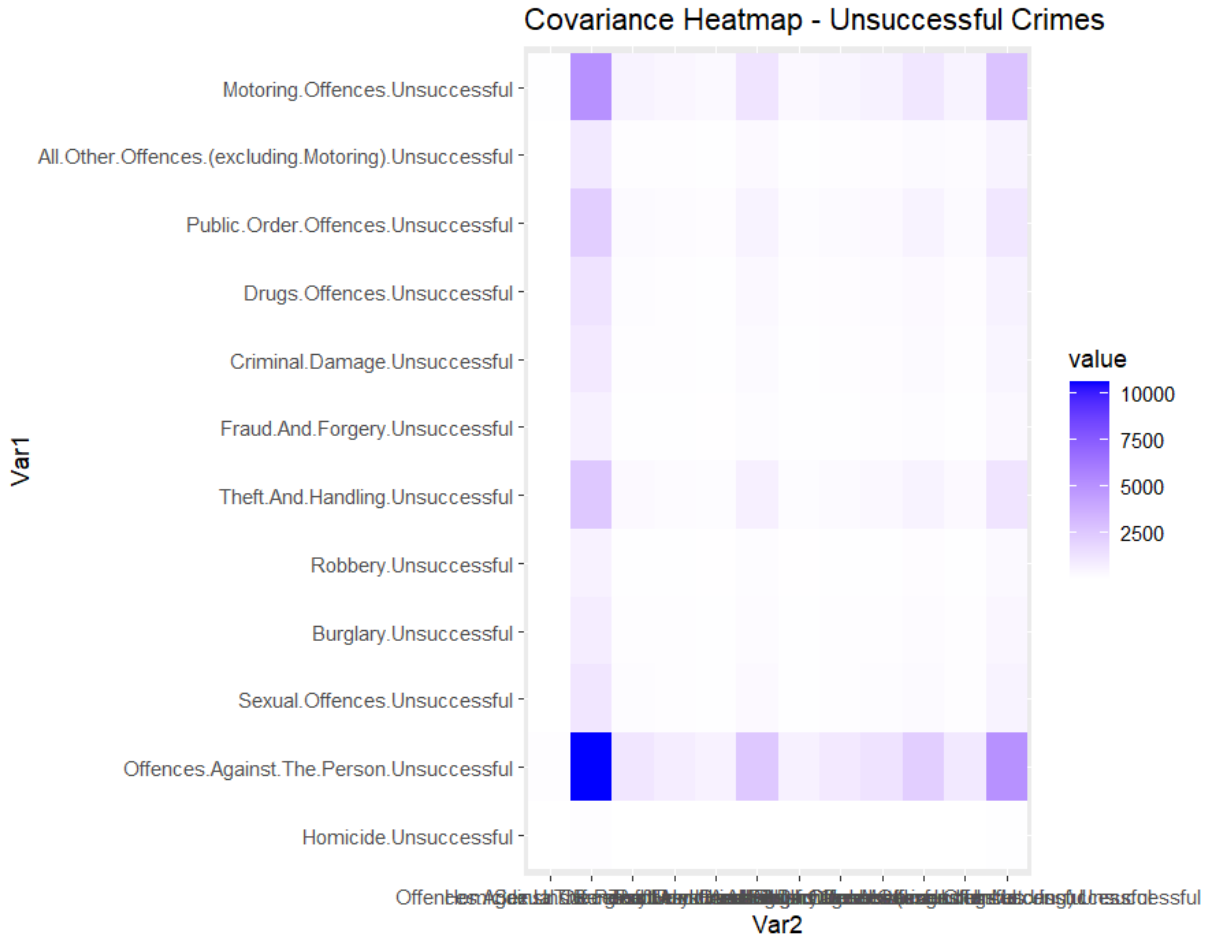
```
#plotting correlation heatmap for unsuccessful crimes
ggplot(unsuccessful_corr_long, aes(Var2, Var1, fill = value)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  labs(title = "Correlation Heatmap - Unsuccessful Crimes")
```



Computing and Plotting Covariance Heatmap for Unsuccessful Crimes DF

```r
#computing covariance matrix for unsuccessful crimes
unsuccessful_cov <- cov(unsuccessful_crimes_df[, 4:15])
print("Correlation Between Unsuccessful Crimes: ")
print(unsuccessful_cov)
```

```r
#plotting covariance heatmap for unsuccessful crimes
ggplot(unsuccessful_cov_long, aes(Var2, Var1, fill = value)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(title = "Covariance Heatmap - Unsuccessful Crimes")
```



Predictive Analysis

In the predictive analysis we have performed various techniques including, hypothesis testing, regression analysis, classification, and clustering.

Hypothesis Testing

We have done 2 hypotheses in this project and the details are below.

Hypothesis Testing #1

Hypothesis Statement: If the crime rate is increasing or decreasing in the rural and urban areas separately.

Code

```r
#HYPOTHESIS 1 - If the crime rate is increasing or decreasing in the rural areas

# Subsetting the relevant variables
data_subset <- final_df[, c("Year", "TotalCrimes", "Region")]

# Grouping the data by year and region
data_grouped <- aggregate(TotalCrimes ~ Year + Region, data_subset, mean)

# Creating a new variable for rural regions
data_grouped$Rural <- ifelse(data_grouped$Region == "Rural", 1, 0)

# Performing regression analysis
model <- lm(TotalCrimes ~ Year + Rural, data = data_grouped)
summary(model)
```

The regression analysis was performed for this hypothesis testing. The summary of the regression model provides several key insights:

1. Intercept (Baseline TotalCrimes): The estimated intercept value of 1527.92 represents the baseline or starting point for TotalCrimes when all other variables are held constant.

2. Year Coefficients: The coefficients for the Year variables (Year2015, Year2016, Year2017) indicate the estimated change in TotalCrimes for each corresponding year, compared to the reference year. However, none of the Year coefficients are statistically significant ($p > 0.05$), suggesting that the Year variable may not have a significant impact on the TotalCrimes.

3. Rural Coefficient: The coefficient for the Rural variable (-794.40) indicates the estimated change in TotalCrimes when the region is rural compared to when it is not rural (urban). This coefficient is statistically significant ($p < 0.001$), indicating that the variable "Rural" has a significant impact on the TotalCrimes. The negative sign of the coefficient suggests that, on average, rural regions have lower TotalCrimes compared to urban regions.

4. R-squared: The multiple R-squared value of 0.9948 indicates that the model explains approximately 99.48% of the variance in the TotalCrimes. This suggests that the Year and Rural variables together explain a large portion of the variation in the TotalCrimes.

5. F-statistic: The F-statistic of 143 and its associated p-value (0.0009388) suggest that the overall regression model is statistically significant, indicating that at least one of the predictors (Year or Rural) is significantly related to the TotalCrimes.

To sum it up, the regression analysis reveals that the variable "Rural" has a significant impact on the TotalCrimes, indicating that rural regions generally have lower total crime rates compared to urban regions. However, the year on year trend of crimes is decreasing.

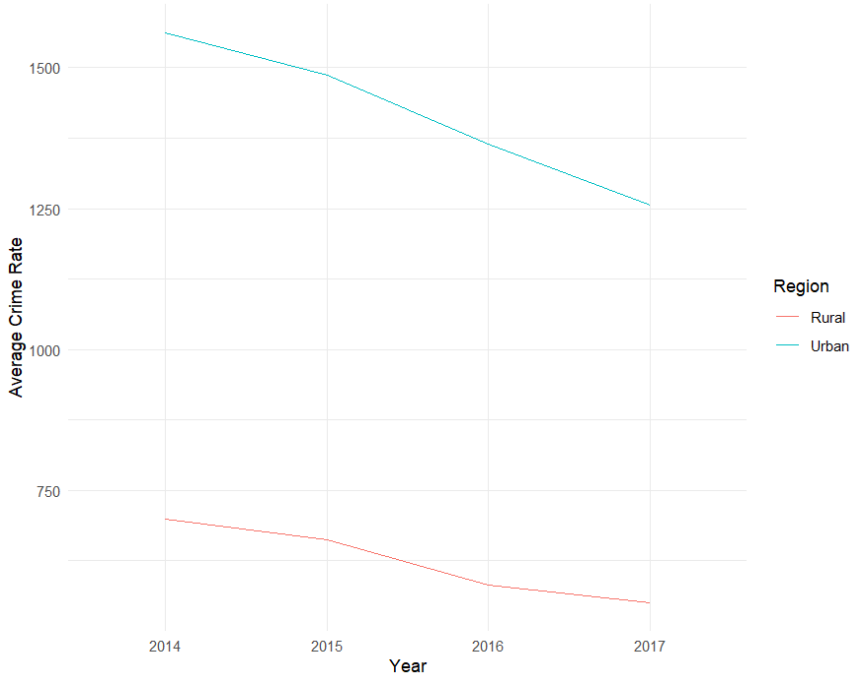
Output

```
Residuals:
      1       2       3       4       5       6       7       8
-34.729 -15.506   5.593  44.643  34.729  15.506  -5.593 -44.643

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1527.92      38.03  40.178 3.39e-05 ***
Year2015      -56.38      48.10  -1.172 0.325764
Year2016     -157.66      48.10  -3.278 0.046509 *
Year2017     -226.83      48.10  -4.715 0.018059 *
Rural        -794.40      34.01 -23.355 0.000172 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 48.1 on 3 degrees of freedom
Multiple R-squared:  0.9948,    Adjusted R-squared:  0.9878
F-statistic:   143 on 4 and 3 DF,  p-value: 0.0009388
```

Code for Plot

```
# Creating a line plot
ggplot(data_grouped, aes(x = Year, y = TotalCrimes, color = Region, group = Region)) +
  geom_line() +
  labs(x = "Year", y = "Average Crime Rate", color = "Region") +
  theme_minimal()
```

Output for plot



Hypothesis Testing #2

Hypothesis Statement: Whether the number of Homicide Convictions is significantly influenced by the number of Burglary Convictions.

Code

```
#HYPOTHESIS 2:whether the number of Homicide Convictions
#is significantly influenced by the number of Burglary Convictions.


crime_data <- final_df[, c("Homicide.Convictions", "Burglary.Convictions")]

crime_data <- na.omit(crime_data)

# Spliting the data into training and test sets (e.g., 70% training and 30% test)
set.seed(123)
train_index <- sample(1:nrow(crime_data), 0.7 * nrow(crime_data))
train_data <- crime_data[train_index, ]
test_data <- crime_data[-train_index, ]

model <- lm(Homicide.Convictions ~ Burglary.Convictions, data = train_data)

predictions <- predict(model, newdata = test_data)

mse <- mean((predictions - test_data$Homicide.Convictions)^2)

summary(model)
```

Output

```
Residuals:
    Min      1Q  Median      3Q     Max
-7.0687 -1.0829 -0.3471  0.6886 24.9241

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         -0.167142   0.088360  -1.892   0.0588 .
Burglary.Convictions  0.064286   0.001988  32.338   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.227 on 1232 degrees of freedom
Multiple R-squared:  0.4591,    Adjusted R-squared:  0.4587
F-statistic:  1046 on 1 and 1232 DF,  p-value: < 2.2e-16
```

We conducted regression analysis for this hypothesis testing. The summary of the regression model provides the following insights:

1. Coefficients: The coefficient for the Burglary Convictions variable is estimated as 0.064286. The coefficient is statistically significant ($p < 0.001$), suggesting a strong relationship between the two variables.

2. R-squared: The multiple R-squared value of 0.4591 indicates that approximately 45.91% of the variance in Homicide Convictions can be explained by the Burglary Convictions variable. This suggests a moderate level of predictability.

3. F-statistic: The F-statistic of 1046 and its associated p-value ($p < 2.2e-16$) indicate that the overall regression model is highly significant. This suggests that the Burglary Convictions variable significantly contributes to explaining the variability in Homicide Convictions.

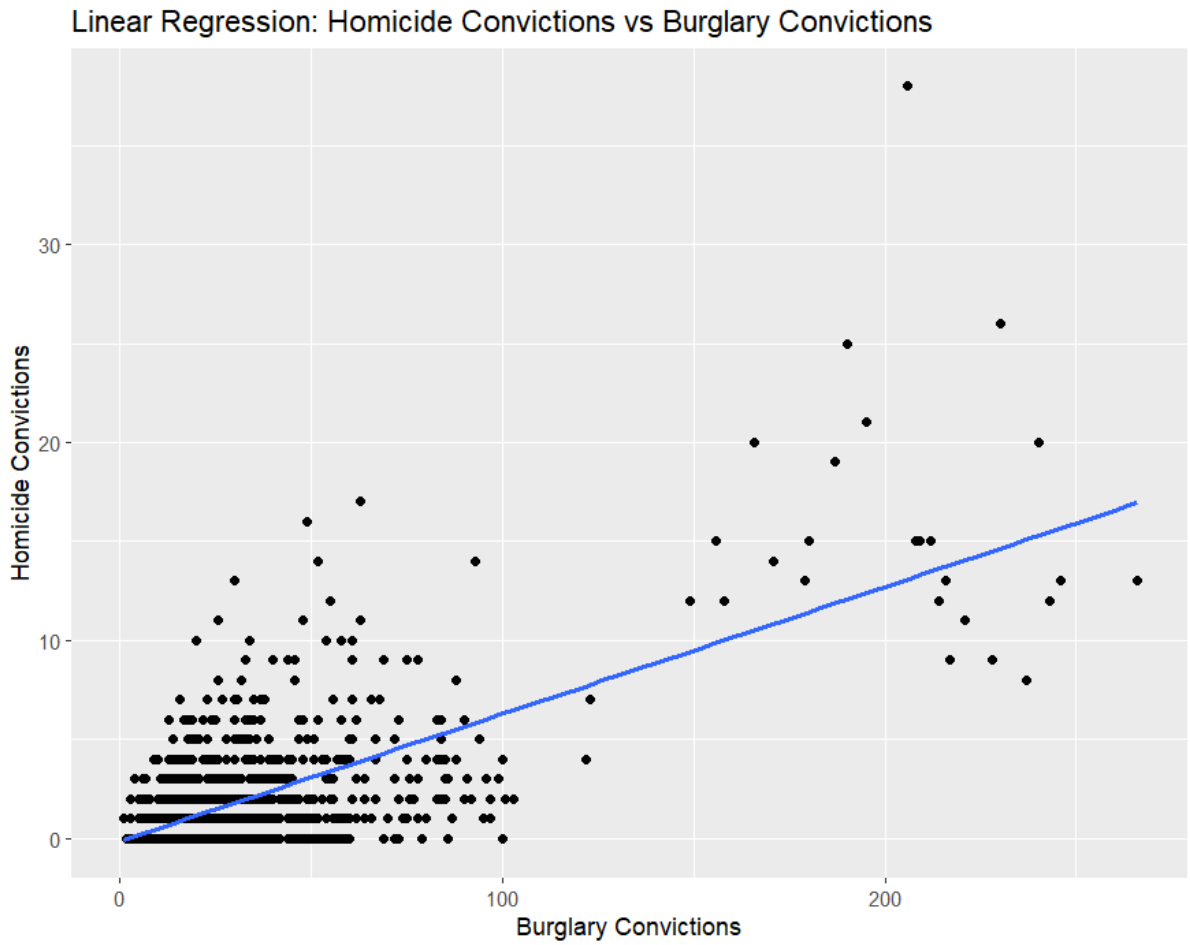
4. Residuals: The residuals (i.e., the differences between the observed and predicted values) show a relatively small spread, with the majority falling within the range of -7.0687 to 24.9241. This indicates that the model's predictions are generally close to the actual values.

To conclude our hypothesis, this analysis reveals a significant and positive relationship between Homicide Convictions and Burglary Convictions. As the number of Burglary Convictions increases, there is an expected increase in Homicide Convictions. The model accounts for approximately 45.91% of the variability in Homicide Convictions (Biswal, 2023).

Code for Plotting

```
#visualising the findings

ggplot(train_data, aes(x = Burglary.Convictions, y = Homicide.Convictions)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Burglary Convictions", y = "Homicide Convictions") +
  ggtitle("Linear Regression: Homicide Convictions vs Burglary Convictions")
```

Output for pllotting



Clustering

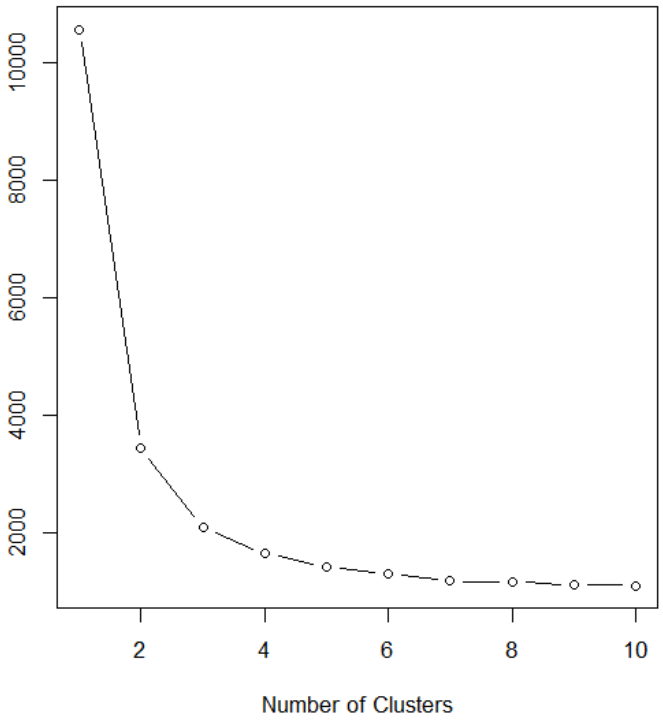
Clustering is an unsupervised machine learning technique to group similar data points together based on their characteristics or features. It aims to discover inherent patterns or structures within the data without any predefined labels or target variables. In the code below, we are using this techniques on the following variables of our dataframe "Homicide.Convictions", "Offences.Against.The.Person.Convictions", "Burglary.Convictions", "Robbery.Convictions", "Theft.And.Handling.Convictions", and "Drugs.Offences.Convictions" (*What Is Cluster Analysis & When Should You Use It?*, n.d.).

We are using elbow method for creating the number of clusters being made, and based on the elbow method analysis, the code is applying the k-means clustering algorithm with the chosen number of clusters (in this case, 4).

Code

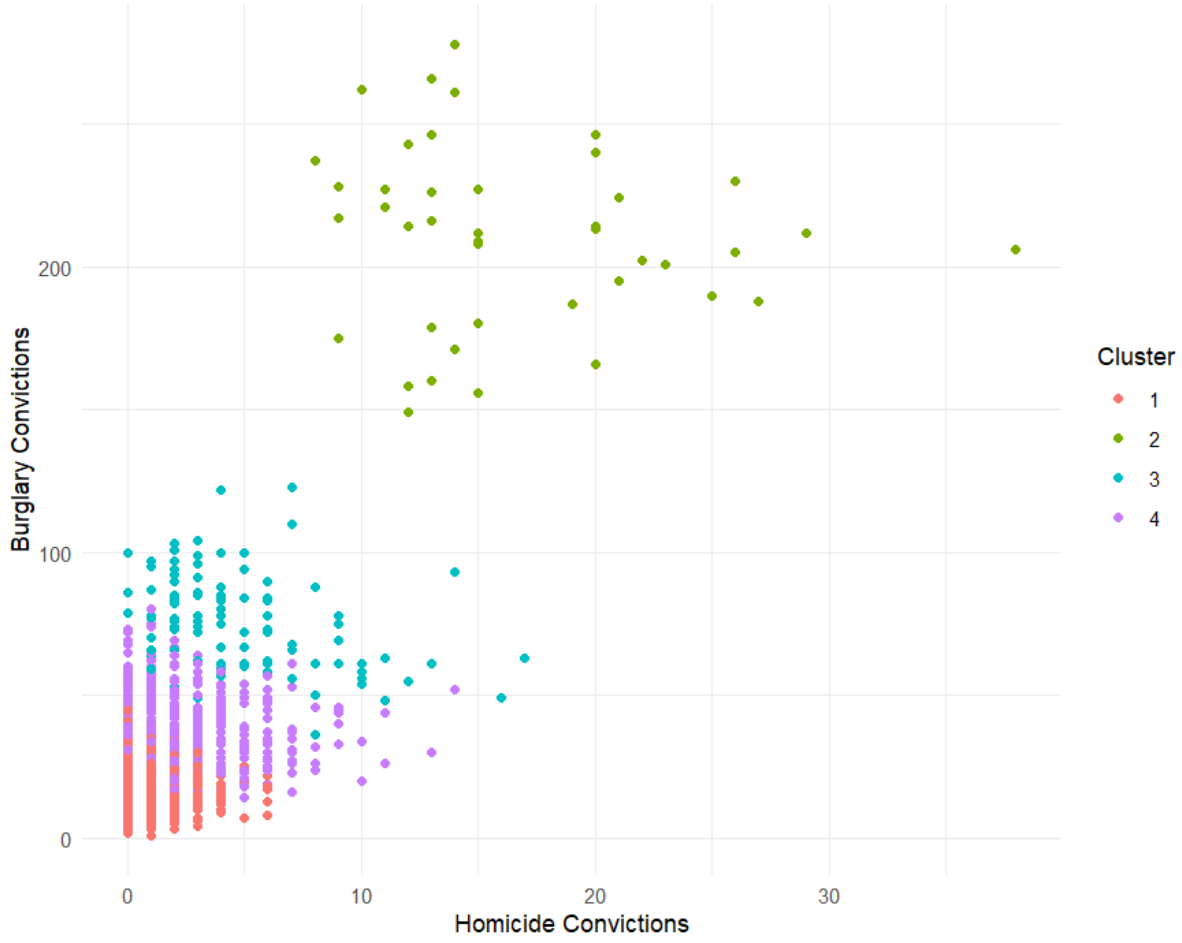
```r
#CLUSTERING

# Choosing the relevant variables for clustering
clustering_data <- final_df[, c("Homicide.Convictions", "Offences.Against.The.Person.Convic
                                "Burglary.Convictions", "Robbery.Convictions",
                                "Theft.And.Handling.Convictions", "Drugs.Offences.Convictic

# standardizing the data
standardized_data <- scale(clustering_data)

#choosing no of clusters using elbow method
wcss <- numeric(length = 10)

for (k in 1:10) {
  kmeans_model <- kmeans(standardized_data, centers = k)
  wcss[k] <- kmeans_model$tot.withinss
}

plot(1:10, wcss, type = "b", xlab = "Number of Clusters", ylab = "WCSS")

chosen_number_of_clusters <- 4

#applying k-means clustering with the chosen number of clusters
kmeans_model <- kmeans(standardized_data, centers = chosen_number_of_clusters)
cluster_labels <- kmeans_model$cluster

#adding labels with clusters
final_df$Cluster <- as.factor(cluster_labels)
```

Output



Visualizing the clusters

```
#visualizing the clusters using a scatter plot
ggplot(final_df, aes(x = Homicide.Convictions, y = Burglary.Convictions, color = Cluster))
  geom_point() +
  labs(x = "Homicide Convictions", y = "Burglary Convictions", color = "Cluster") +
  theme_minimal()
```


Burglary Convictions
200
100
0
0
10
20
30
Homicide Convictions
Cluster
1
2
3
4

Regression Analysis

- Forecasting Crimes in “Gloucestershire” county

In this section, we are trying to predict the crime rate in the county named Gloucestershire. The analysis utilized the ARIMA model to generate future predictions. The accuracy of the predictions was evaluated using various metrics, and the forecasted values were obtained for the specified time period, in this case, 2028 to 2066.

Code

```
#PREDICTIVE ANALYSIS - Crimes in Gloucestershire


#choosing required variables from the dataframe
gloucestershire_data <- subset(final_df, County == "Gloucestershire" & Year >= 2014 & Year

#converting "Year" variable to a proper date format
gloucestershire_data$Year <- as.Date(gloucestershire_data$Year, format = "%Y")

#creating a time series object
crime_ts <- ts(gloucestershire_data$TotalCrimes, frequency = 1, start = c(2014, 1))

#splitting the data into training and testing
train_data <- window(crime_ts, end = c(2016, 12))
test_data <- window(crime_ts, start = c(2017, 1))

# training the data using the ARIMA model
arima_model <- auto.arima(train_data)

#Predicting future values
forecast_values <- forecast(arima_model, h = length(test_data))

#extracting the forecasted values
forecasted_crime <- forecast_values$mean

#calculating accuracy of predictions
accuracy(forecast_values, test_data)

#viewing the forecasted values
forecasted_crime
```

Output

```
                      ME     RMSE      MAE        MPE     MAPE      MASE        ACF1
Training set  1.624048e-14 49.15640 44.22449  -1.702105 11.84425 0.9068113  0.05158081
Test set     -3.521429e+01 63.13272 51.62245 -12.969205 16.60023 1.0585045 -0.02932448
             Theil's U
Training set        NA
Test set     0.8474618
>
> #viewing the forecasted values
> forecasted_crime
Time Series:
Start = 2028
End = 2066
Frequency = 1
 [1] 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714
[10] 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714
[19] 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714
[28] 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714 378.0714
[37] 378.0714 378.0714 378.0714
```

Analysis Results

The analysis produced the following results:

1. Accuracy Measures:
   - Mean Error (ME): The average forecast bias was negligible, indicating that, on average, the forecasts were unbiased.
   - Root Mean Squared Error (RMSE): The average magnitude of the forecast errors was 63.13, suggesting a moderate level of forecast accuracy.
   - Mean Absolute Error (MAE): The average absolute forecast error was 51.62, providing an estimate of the average forecast deviation.
   - Mean Percentage Error (MPE): The average percentage deviation of the forecasts from the actual values was -12.97%, indicating a tendency to slightly overestimate the crime rates.
   - Mean Absolute Percentage Error (MAPE): The average absolute percentage deviation of the forecasts from the actual values was 16.60%, representing the average magnitude of forecast errors relative to the actual values.
   - Mean Absolute Scaled Error (MASE): The forecast accuracy, relative to a naïve forecast, was 1.06, suggesting a moderate level of forecast improvement.
   - Autocorrelation of Forecast Errors (ACF1): The autocorrelation of the forecast errors at lag 1 was -0.029, indicating no significant residual autocorrelation.

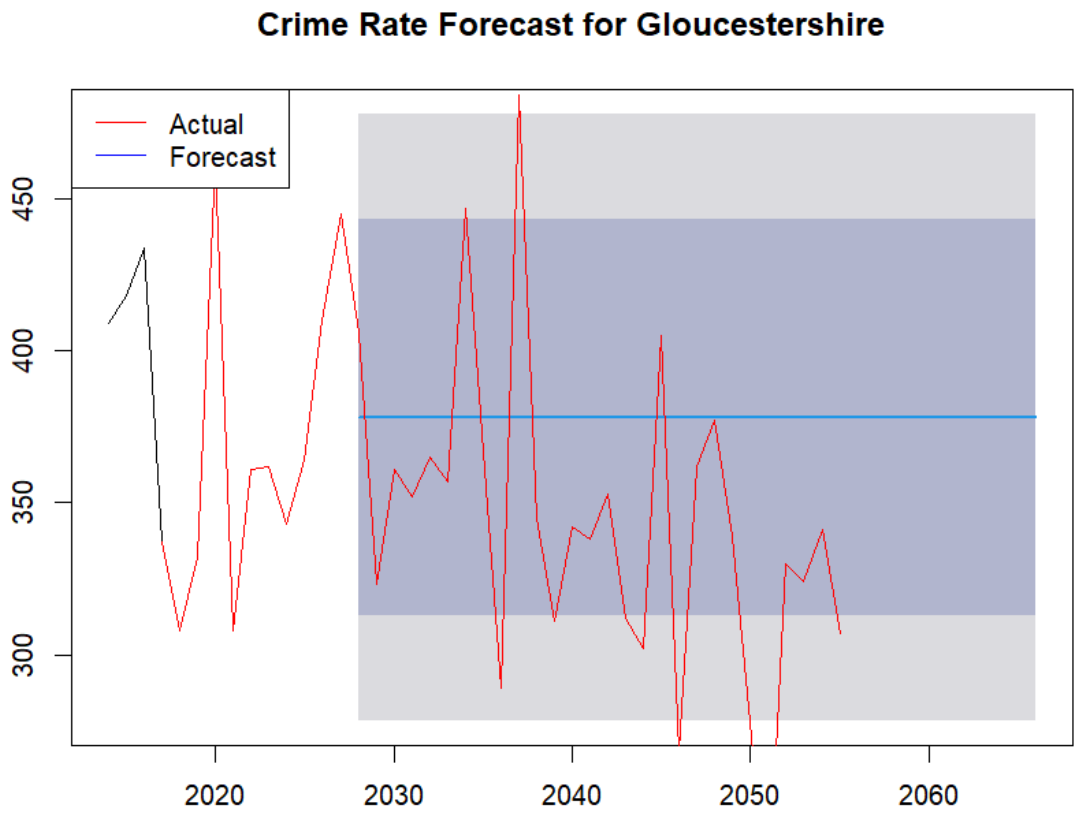
2. Theil's U:
   - The Theil's U statistic, which measures the forecast accuracy relative to a naïve forecast, was 0.85, suggesting a reasonable level of improvement over the naïve forecast.

Forecasted Values:
The forecasted values for crimes in Gloucestershire are as follows:
- The forecasted total crimes range from 378.07 to 378.07 for each future time period from 2028 to 2066. These values indicate the expected level of crimes in Gloucestershire based on the ARIMA model's predictions (Gallo, 2015).

Plotting-Code

```
# plotting the actual and forecasted values
plot(forecast_values, main = "Crime Rate Forecast for Gloucestershire")
lines(test_data, col = "red")
legend("topleft", legend = c("Actual", "Forecast"), col = c("red", "blue"), lty = 1)
```

Plot-Output



Findings

The predictive analysis provided insights into the future trends of crimes in Gloucestershire. Although the forecasted value suggests relatively stable crime rates, it is important to note that forecasting crime is a challenging task due to various dynamic factors. The accuracy measures indicate a moderate level of forecast accuracy, and improvements can be explored to enhance the predictive modeling approach. These results can assist law enforcement agencies and policymakers in understanding potential crime patterns and inform decision-making processes related to resource allocation and crime prevention strategies.

Classification

Classification is a supervised machine learning technique that involves categorizing or assigning predefined labels or classes to input data based on the features or attributes. It is an approach where the algorithm learns from labeled training data to make predictions or classifications on new, unseen data (//, n.d.).

In the below provided code, we are taking the variables Homicide.Convictions and Burglary.Convictions as input features. We are using these input features to predict the Region variable, which indicates whether an area is classified as a high crime area or a low crime area.

Code

```
#CLASSIFICATION


# Convert the response variable to a factor
classification_df <- final_df[c("Homicide.Convictions", "Burglary.Convictions")]

target_variable <- final_df$Region

# Split the data into training and testing sets (adjust the ratio as needed)
set.seed(123)  # for reproducibility
train_indices <- sample(1:nrow(classification_data), 0.7 * nrow(classification_data))
train_data <- classification_data[train_indices, ]
train_target <- target_variable[train_indices]
test_data <- classification_data[-train_indices, ]
test_target <- target_variable [-train_indices]


multinom_model <- multinom (train_target ~ ., data = train_data)

# predictions on the test data
predictions <- predict(multinom_model, newdata = test_data, type = "class")

# model performance evaluation
accuracy <- sum (predictions == test_target) / length(test_target)
print (paste("Accuracy: ", accuracy))
```

Output

```
> multinom_model <- multinom (train_target ~ ., data = train_data)
# weights:  6 (5 variable)
initial  value 855.343621
iter  10 value 561.664757
iter  10 value 561.664757
iter  10 value 561.664757
final  value 561.664757
converged
>
> # predictions on the test data
> predictions <- predict(multinom_model, newdata = test_data, type = "class")
>
> # model performance evaluation
> accuracy <- sum (predictions == test_target) / length(test_target)
> print (paste("Accuracy: ", accuracy))
[1] "Accuracy:  0.771698113207547"
>
```

Findings

The output above shows the convergence of the multinomial logistic regression model, indicating that the optimization algorithm successfully converged.

The printed output indicates the accuracy of the model on the test data, which is approximately 0.7717 (or 77.17%).

References

/ /. (n.d.). / / - Wiktionary. Retrieved June 21, 2023, from https://www.ibm.com/docs/en/iis/11.7?topic=dcao-data-classification-analysis

Biswal, A. (2023, June 7). *What is Hypothesis Testing in Statistics? Types and Examples*. Simplilearn. Retrieved June 21, 2023, from https://www.simplilearn.com/tutorials/statistics-tutorial/hypothesis-testing-in-statistics

Gallo, A. (2015, November 4). *A Refresher on Regression Analysis*. Harvard Business Review. Retrieved June 21, 2023, from https://hbr.org/2015/11/a-refresher-on-regression-analysis

What is Cluster Analysis & When Should You Use It? (n.d.). Qualtrics. Retrieved June 21, 2023, from https://www.qualtrics.com/uk/experience-management/research/cluster-analysis/

0.1000/182

Lastname, W. (2009). If there is no DOI use the URL of the main website referenced. *Article Without DOI Reference*, Vol#(Issue#), 166-212.