

# Data Visualisation In R programming

Atharva Rodge

## Introduction

In the age of data and AI it is important for us to extract the meaningful insights from number of data-set to understand the relation between the variable and for decision making across various domain. Data visualization is a necessary aspect of EDA to look for patterns, trends, missing values, relation between the variables, finding outliers and most importantly for easier understand for non tech people and stakeholders.

This project focus on Using R programming to explore, analyze, and visualize the data-set. Through interactive and visually compelling representations. In this project we will employ various range of visualization techniques, including bar plots, histograms, two\_way table , heat map, Time series plots and interactive plots. By selecting appropriate tool we will convey the information clearly and accurately.

This study will thoroughly examine the Colchester policing data set in this project, employing a variety data visualization techniques. We want to provide helpful insights into the dynamics of crime in the Colchester region in 2023 by examining the distribution of crime categories, using maps to analyse geographical patterns, analyzing trends using time series plots, and investigating correlations between elements. To improve the presentation of our findings and give an entertaining and instructional look at the dataset, we will also incorporate advanced visualizations and interactive components.

## Crime Data

### Dataset

```
project_data <- read.csv("crime23.csv")
project_data <- project_data[,c(1,3:7,9,10,12)]

dim(project_data)

## [1] 6878     9
```

The given data set consists of 6878 rows & 9 columns after removing unwanted columns. Providing precise information on criminal incidences in Colchester in the year 2023. These include Type of crime, Date, outcome status etc for our understanding.

This dataset provides a complete perspective of Colchester's criminal records, enabling for analysis and investigation of different elements of crime episodes that occurred in 2023.

## Bar Plot for count of crime by category

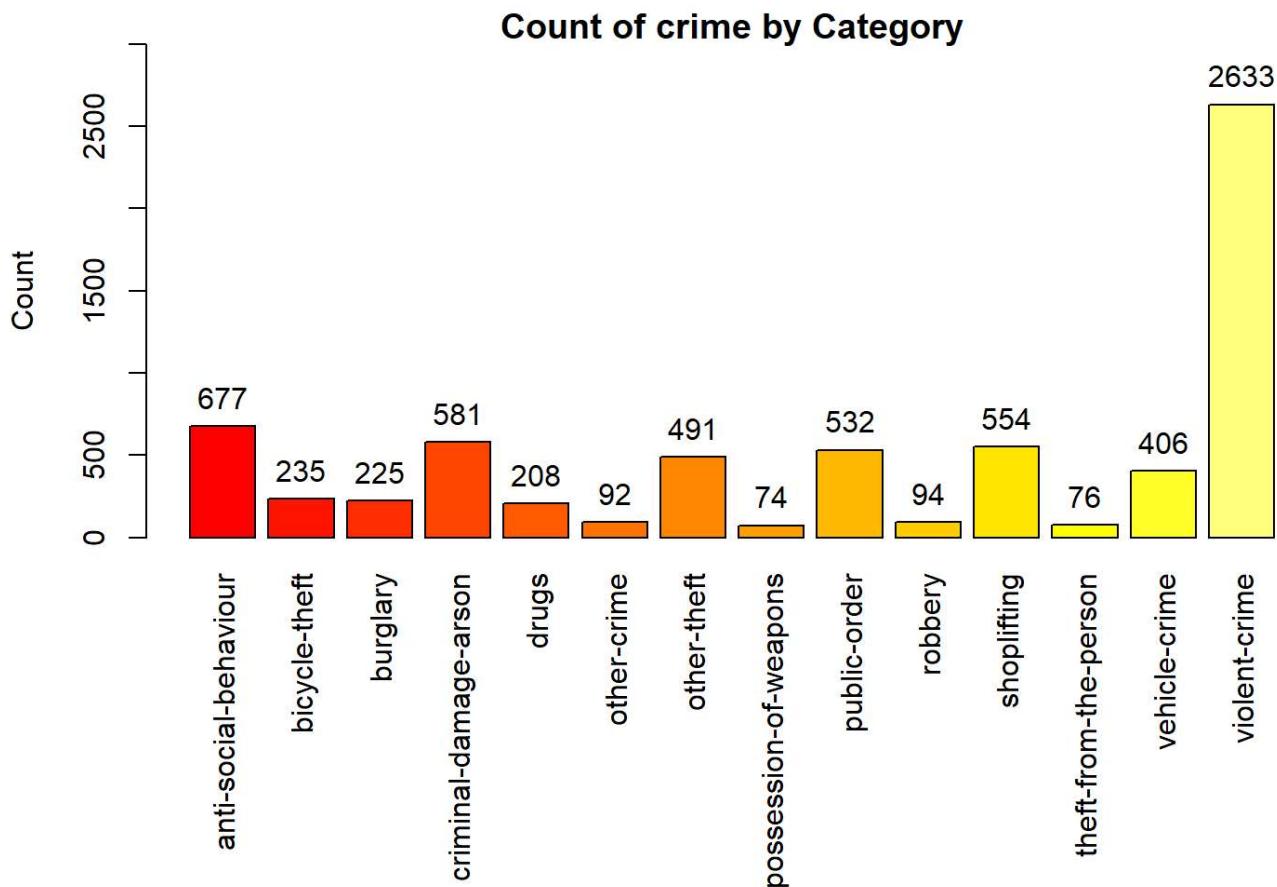
```
crime_category <- project_data %>% group_by(category) %>% count()

par(mar = c(10, 5, 1, 2) + 0.1)
options(repr.plot.width=6, repr.plot.height=4)

# bar plot

bar_p <- barplot(crime_category$n, names.arg = crime_category$category, las = 3, col=heat.colors(n=15), ylim = c(0,3000), main = "Count of crime by Category", ylab = "Count" )

text(x = barplot(crime_category$n, names.arg = crime_category$category, plot = FALSE),
y = crime_category$n + 0.5,
labels = crime_category$n,
pos = 3)
```



The above bar plot is used to check the count of crime by their category. The most crime happened in Colchester are “violent-crime” followed by “anti-social-behaviour” and “criminal\_damage\_arson”. The least amount of crimes are the one with “possession-of-weapons” i.e most crimes which takes place doesn’t include weapons. The other most occurring of crimes are “vehicle-crime”, “shoplifting”, “public\_order” etc.

## Bar Plot for count of crime by category

```

outcome_category <- project_data %>% group_by(outcome_status) %>% count()

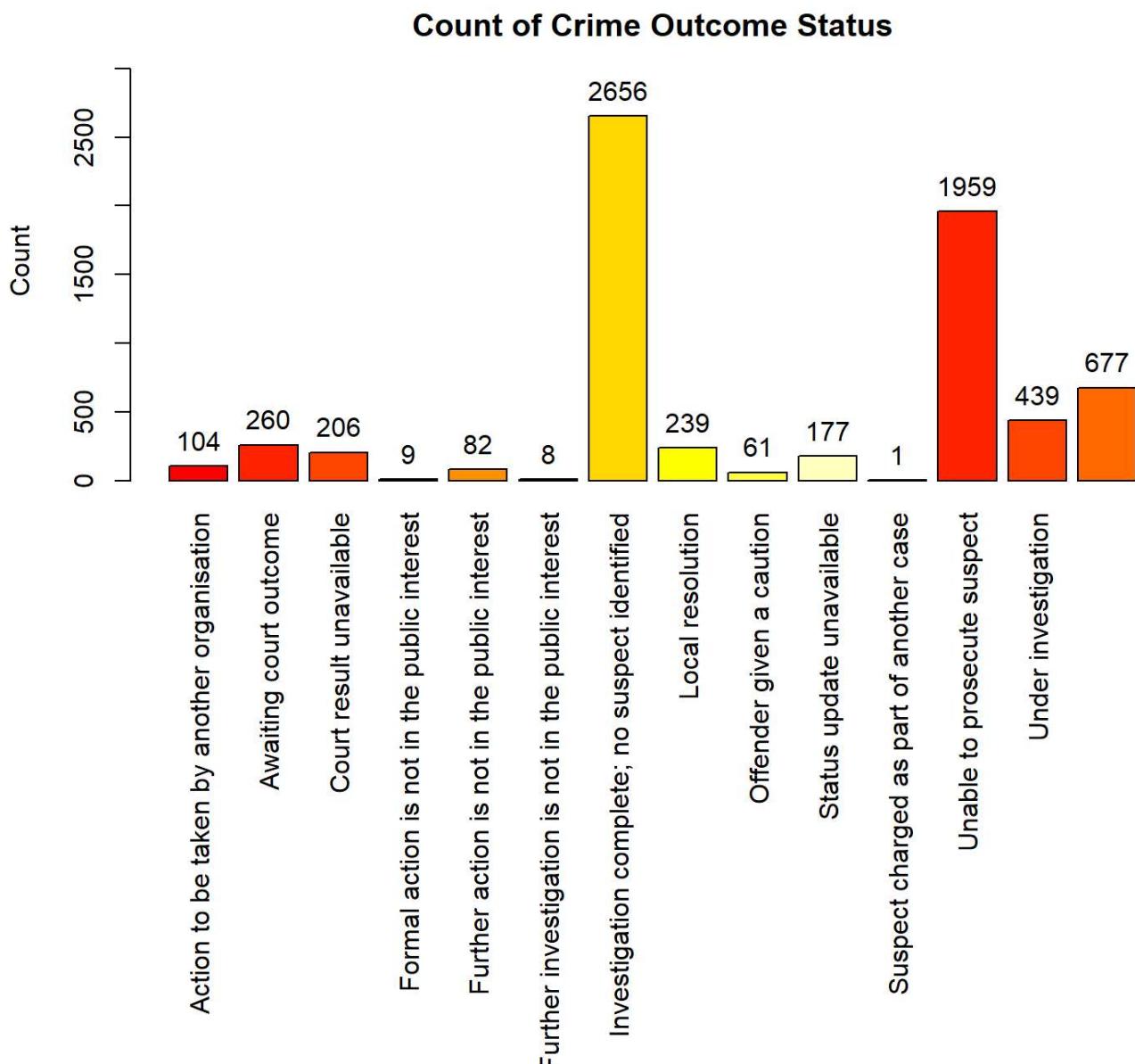
par(mar = c(20, 5, 1, 2) + 0.1)
options(repr.plot.width=6, repr.plot.height=4)

# the bar plot

bar_p <- barplot(outcome_category$n, names.arg = outcome_category$outcome_status, las = 3, col=heat.colors(n=10), ylim = c(0,3200), main = "Count of Crime Outcome Status", ylab = "Count")

text(x = barplot(outcome_category$n, names.arg = outcome_category$outcome_status, plot = FALSE),
     y = outcome_category$n + 0.5,
     labels = outcome_category$n,
     pos = 3)

```



To see how many crimes have been committed and their outcomes, utilise the bar plot above. The majority of criminal cases have been closed with no suspects found. We may infer the type of status and the conclusion from this pub plot. For instance, most instances did not allow for the suspect to be prosecuted, while other crimes were resolved locally.

## Two way table

```
two_way <- project_data %>%
  select(date, category) %>%
  mutate(Month_Name = format(as.Date(paste(date, "01", sep = "-")), "%B")) %>%
  mutate(Month_Name = factor(Month_Name, levels = month.name)) # Specify the order of months

two_way_table <- table(two_way$category, two_way$Month_Name)
two_way_table
```

```
##
##                                     January February March April May June July August
## anti-social-behaviour          46      49    21    53   67   52   76   71
## bicycle-theft                  20      14    19    16   16   14   15   21
## burglary                        17      22    14    22   15   26   14   20
## criminal-damage-arson         59      37    52    63   64   42   42   33
## drugs                           14      17    21    21   22   15   17   7
## other-crime                     7       5     6    15   3    11   12   9
## other-theft                      48      37    35    38   42   41   51   41
## possession-of-weapons           3       3     11    5    7   3    8    5
## public-order                     45      42    58    51   37   36   40   41
## robbery                          8       7     8    7    7   17   6    5
## shoplifting                      76      31    51    40   51   59   33   57
## theft-from-the-person            6       7     12    7    5   6    9    5
## vehicle-crime                   65      15    21    29   24   45   25   16
## violent-crime                   237     181   226   207  226  196  236  219
##
##                                     September October November December
## anti-social-behaviour          90      68    39    45
## bicycle-theft                  37      26    27    10
## burglary                        18      31    11    15
## criminal-damage-arson         47      45    53    44
## drugs                           25      19    13    17
## other-crime                     7       6     5    6
## other-theft                      34      49    37    38
## possession-of-weapons           8       6     8    7
## public-order                     45      52    45    40
## robbery                          8       9     5    7
## shoplifting                      33      43    39    41
## theft-from-the-person            7       3     4    5
## vehicle-crime                   20      26    56    64
## violent-crime                   263     209   221   212
```

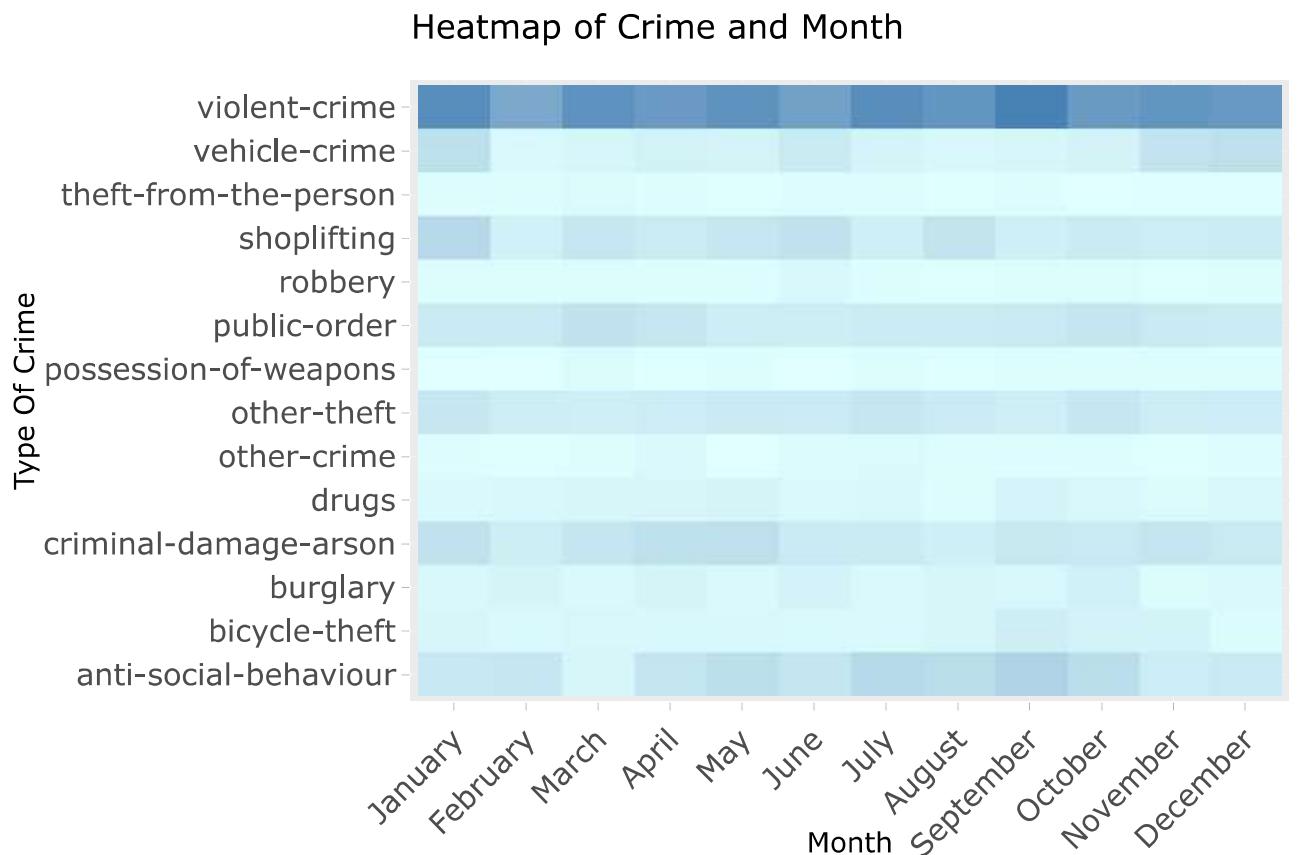
```
two_way_df <- as.data.frame(two_way_table)
```

This two-way table displays information on the various sorts of crimes recorded throughout different months of the year. Each row refers to a distinct sort of crime, and each column represents a month. We can see the distribution of different sorts of crimes throughout the year. For example, some sorts of crimes may have seasonal fluctuations, with larger rates occurring during specific months. for example “Anti-socialBehaviour crime” suddenly increases after march and decreases after october as shown in the above table.

## Heat map of crime and Month

```
two_way_heat <- ggplot(two_way_df, aes(x = Var2, y = Var1 , fill = Freq)) +
  geom_bin2d() +
  scale_fill_gradient(low = "#E0FFFF", high = "#4682B4") + # Customize the color gradient
  labs(x = "Month", y = "Type Of Crime", title = "Heatmap of Crime and Month", legend = "Frequency") +
  theme(axis.text = element_text(size = 12),
        axis.text.x = element_text(size = 12 ,angle = 45),)

ggplotly(two_way_heat)
```

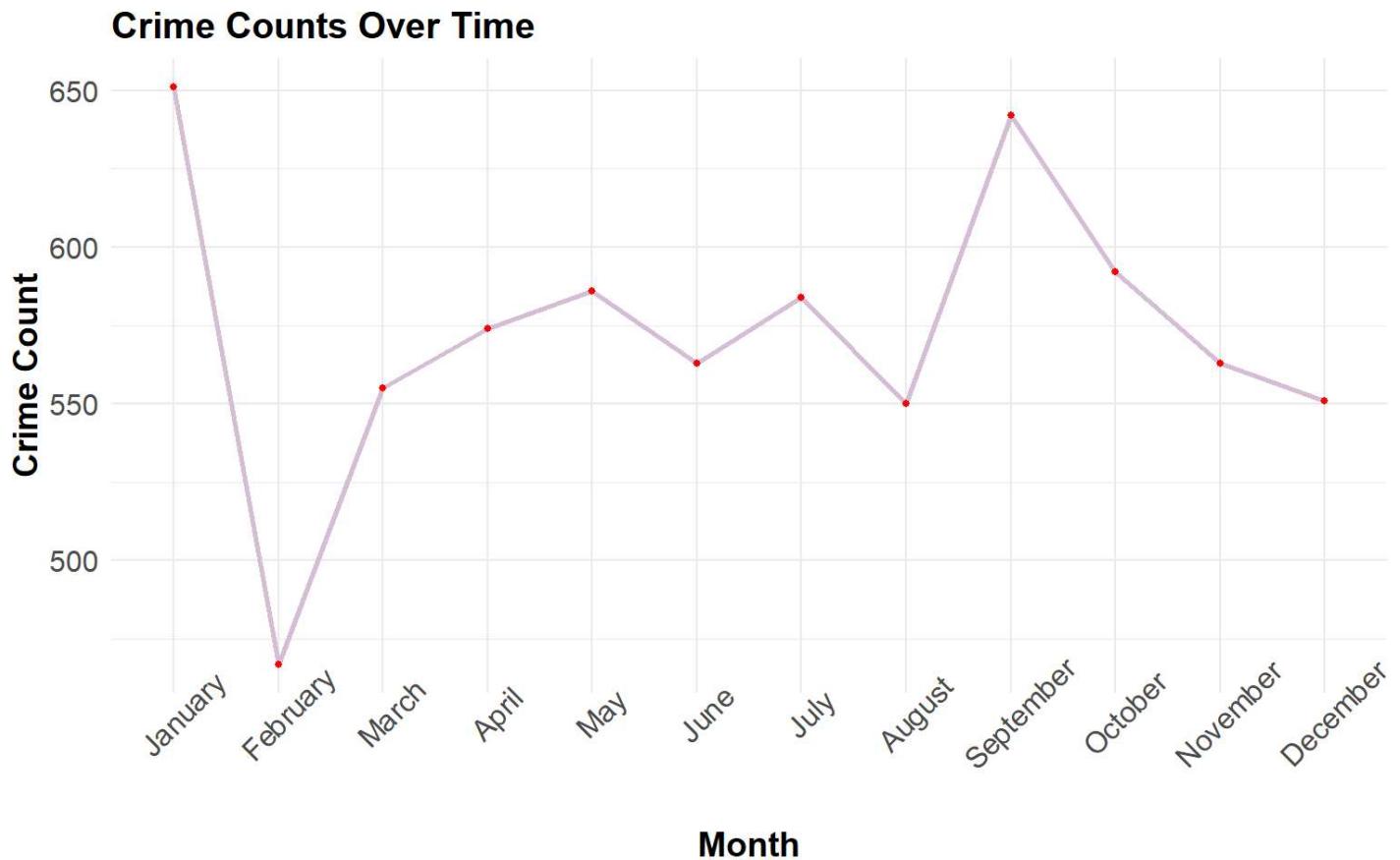


The heatmap above depicts the frequency of various sorts of crimes recorded in different months. Each cell in the heatmap represents a combination of a certain sort of crime (y-axis) and a given month (x-axis), with the colour intensity of each cell representing the frequency of occurrence. The heatmap allows us to identify temporal patterns in crime occurrences throughout the year. By examining the variations in color intensity across different months, we can discern whether certain types of crimes exhibit seasonal trends or are more prevalent during specific times of the year.

## Time series plot of crime count over time

```
crime_month <- project_data %>%
  group_by(date) %>%
  count() %>%
  mutate(Month_Name = format(as.Date(paste(date, "01", sep = "-")), "%B")) %>%
  mutate(Month_Name = factor(Month_Name, levels = month.name)) # Specify the order of months

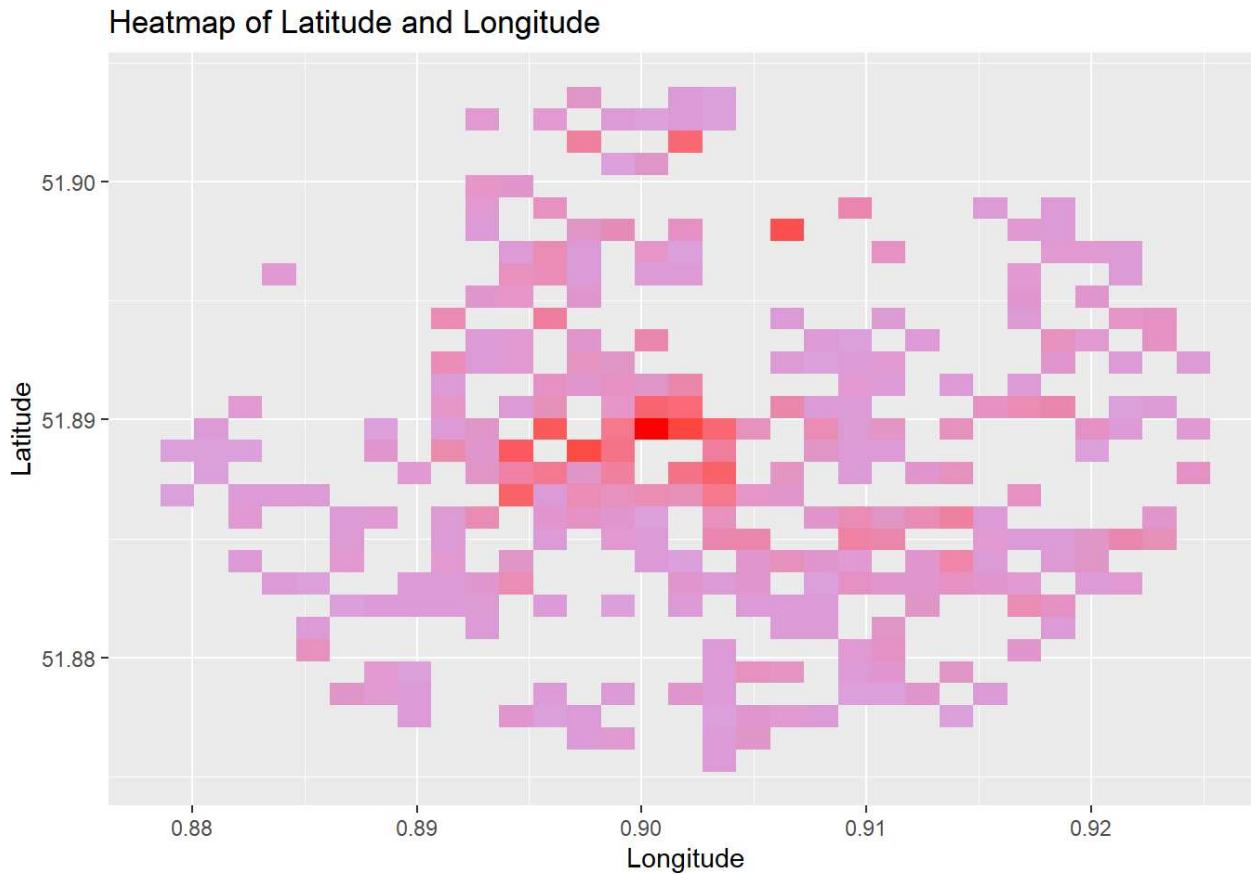
# Create a time series plot
ggplot(crime_month, aes(x = Month_Name, y = n, group = 1)) +
  geom_line(color = "#D8BFD8", size = 1) +
  geom_point(color = "red", size = 1) +
  labs(x = "Month", y = "Crime Count", title = "Crime Counts Over Time") +
  theme_minimal() +
  theme(plot.title = element_text(size = 15, face = "bold"),
        axis.text.x = element_text(size = 12, angle = 45),
        axis.text = element_text(size = 12),
        axis.title = element_text(size = 14, face = "bold"))
```



The time series line graph showing the number of crimes committed in a city over a one-year period. The x-axis, or horizontal axis, of the graph is labeled “Month” and lists the twelve months of the year. The y-axis, or vertical axis, is labeled “Crime Count” and shows the number of crimes reported. The above time series chart shows that there was a decrease in a crime rate from the month of January to February and the the crime rates in the city started to increase by month on month. The most count of crimes occurred are in the month of January And September.

## Latitude & Longitude Heatmap

```
ggplot(project_data, aes(x = project_data$long, y = project_data$lat)) +
  geom_bin2d() +
  scale_fill_gradient(low = "#DDA0DD", high = "red") +
  labs(x = "Longitude", y = "Latitude", title = "Heatmap of Latitude and Longitude")
```



The Above heatmap implies that the most number of crime occurred at long-0.90 and lat-51.89. this heat map helps us to identify the count of crime based on the location the darker the location the number of count of crime are more at that location.

## Pie Chart

```
project_data$street_name <- gsub("^On or near $", "Unknown Locatoin", project_data$street_name)

top_fifteen_crime_streets <- project_data %>%
  group_by(street_name) %>%
  count() %>%
  arrange(desc(n) ) %>%
  head(15)

top_fifteen_crime_streets$street_name <- gsub("^On or near ", "", top_fifteen_crime_streets$street_name)

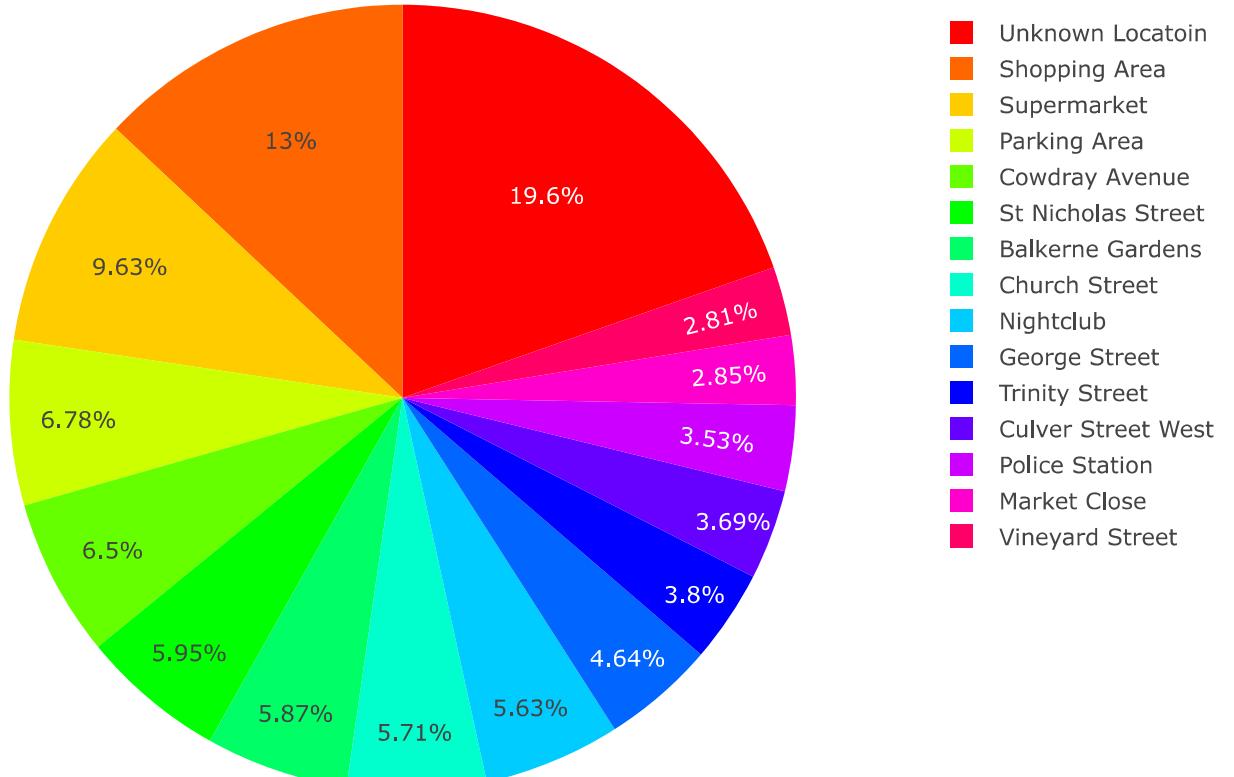
# top_ten_crime_streets
percentages <- round(top_fifteen_crime_streets$n / sum(top_fifteen_crime_streets$n) * 100, 1)

# Create a color palette
colors <- rainbow(length(top_fifteen_crime_streets$n))

# Create a pie chart
pie_chart <- plot_ly(
  labels = top_fifteen_crime_streets$street_name,
  values = top_fifteen_crime_streets$n,
  type = "pie",
  marker = list(colors = colors)
) %>% layout(title = "Pie Chart of Top Fifteen Crime Streets" , x = 2)

# pie chart
pie_chart
```

Pie Chart of Top Fifteen Crime Streets



“Pie Chart of Top Fifteen Crime Streets.” It shows the proportion of crime on the top fifteen crime streets in Colchester. The pie chart is broken into 15 slices, with each reflecting a separate criminal street. The slices are coloured in different colours and labelled with the name of the street. Each slice has a label next to it indicating the proportion of crimes committed on that specific street. Here are some of the streets included in the pie chart, along with their respective percentages: Shopping Area: 13% Super Market: 9.63% Parking Area: 6.78%

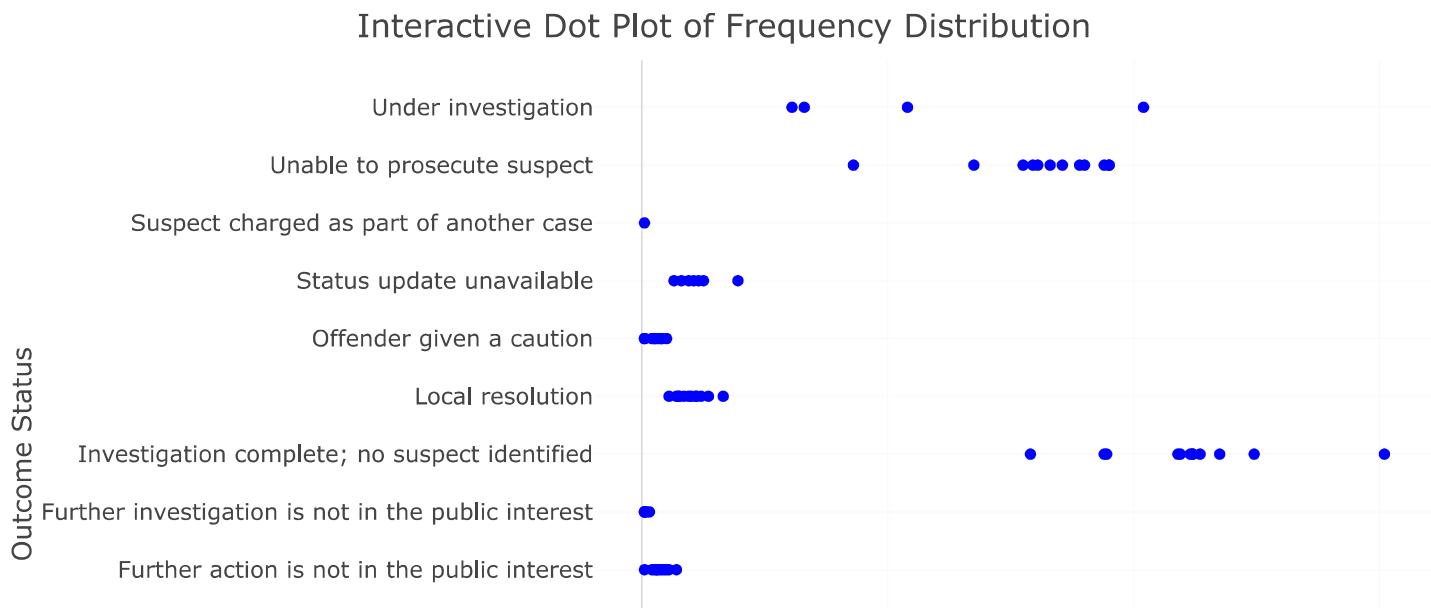
These are the streets with most number of crime among the top 15 streets followed by these street with average number of crimes. Cowdray Avenue Culver Street West Vineyard Street Balkerne Gardens St Nicholas Street Trinity Street

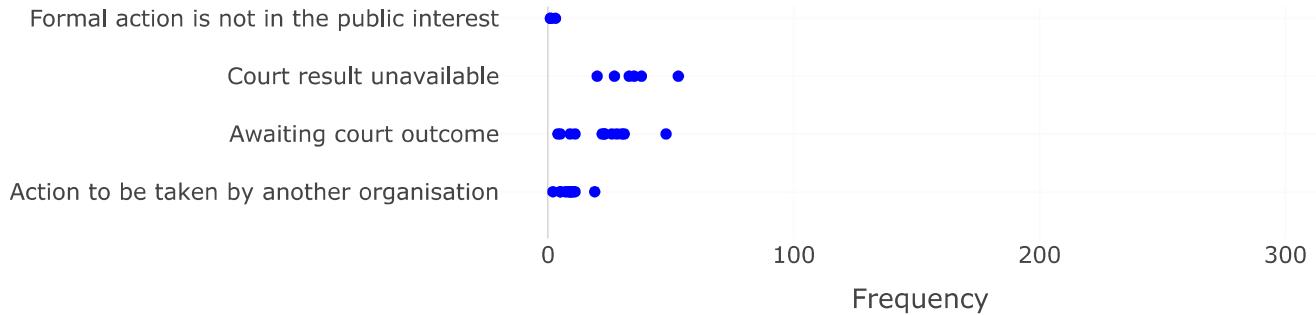
## Dot plot for Outcome status and its count

```
dot_ch <- project_data %>%
  select(date , outcome_status) %>%
  group_by(outcome_status, date) %>%
  count() %>%
  mutate(Month_Name = format(as.Date(paste(date, "01", sep = "-")), "%B")) %>%
  mutate(Month_Name = factor(Month_Name, levels = month.name)) # Specify the order of months

# Create the interactive dot chart
dot_chart <- plot_ly(dot_ch, x = ~n, y = ~outcome_status, type = "scatter",
                      mode = "markers", marker = list(color = "blue", symbol = "circle"),
                      text = ~paste("Frequency:", n, "<br>Category:", Month_Name),
                      hoverinfo = "text") %>%
  layout(title = "Interactive Dot Plot of Frequency Distribution",
         xaxis = list(title = "Frequency"),
         yaxis = list(title = "Outcome Status"))

# interactive dot chart
dot_chart
```





The dot plot depicts the frequency of each result condition. Graphing the frequency distribution of various result statuses over time. Analysing the placements of the dots allows one to see trends and changes in outcomes over time. This visualisation helps to understand the distribution of outcomes, offering significant insights on patterns and probable links between the result status of crimes committed in Colchester, frequency, and dates. This is an interactive graphic that displays the frequency and timing of the condition.

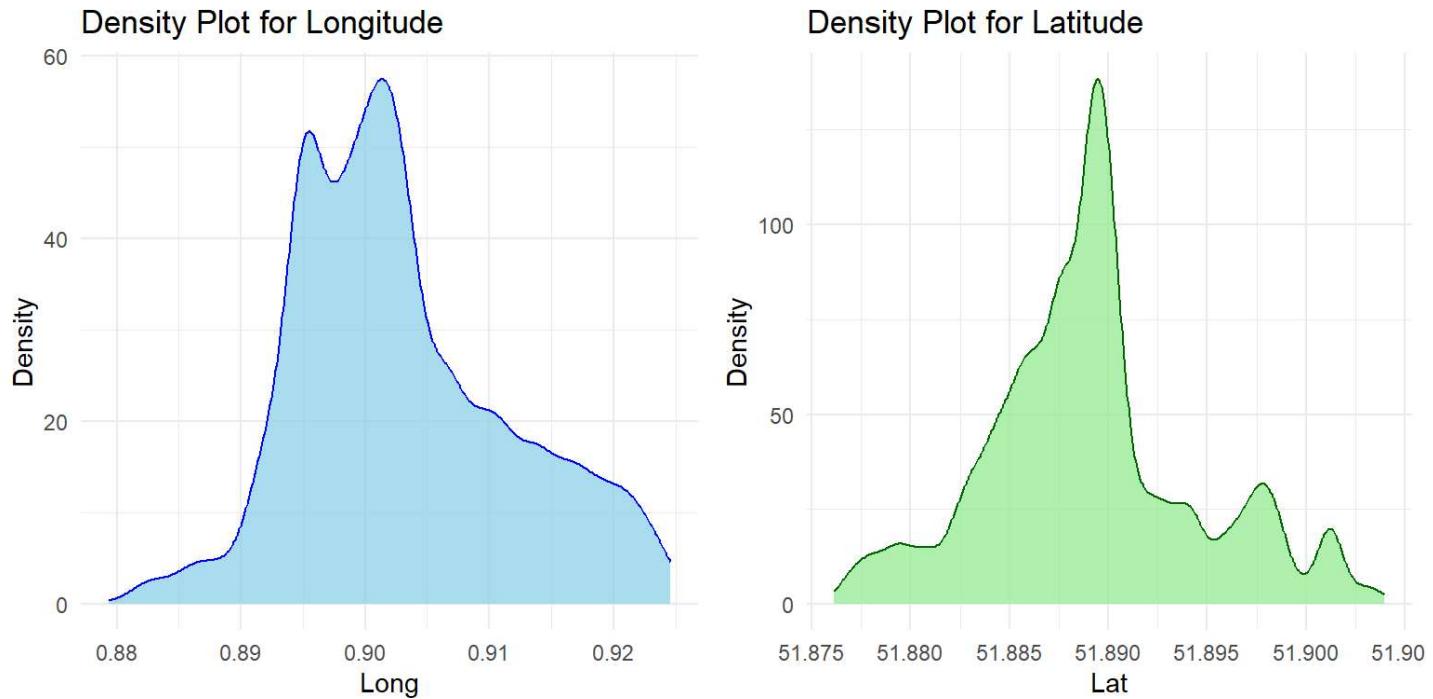
## Density plot for latitude and Longitude

```
library(gridExtra)

# Create the first density plot
plot1 <- ggplot(project_data, aes(x = long)) +
  geom_density(fill = "skyblue", color = "blue", alpha = 0.7) +
  labs(x = "Long", y = "Density", title = "Density Plot for Longitude") +
  theme_minimal() # Use a minimal theme

# Create the second density plot
plot2 <- ggplot(project_data, aes(x = lat)) +
  geom_density(fill = "lightgreen", color = "darkgreen", alpha = 0.7) +
  labs(x = "Lat", y = "Density", title = "Density Plot for Latitude") +
  theme_minimal() # Use a minimal theme

# Combine the two plots into a single plot
grid.arrange(plot1, plot2, ncol = 2)
```



This R function creates two density charts that show the distribution of longitude and latitude in the project data set. The first map shows the density of longitude values in a blue color scheme, while the second plot shows latitude values in green. These visualizations provide insight into the spatial distribution of data points, which aids in analyzing geographical trends within the data set. As we can see in the density plot the height of the plot shows the occurrence of crime at the particular location. interpret

## Time Series plot for crime by category

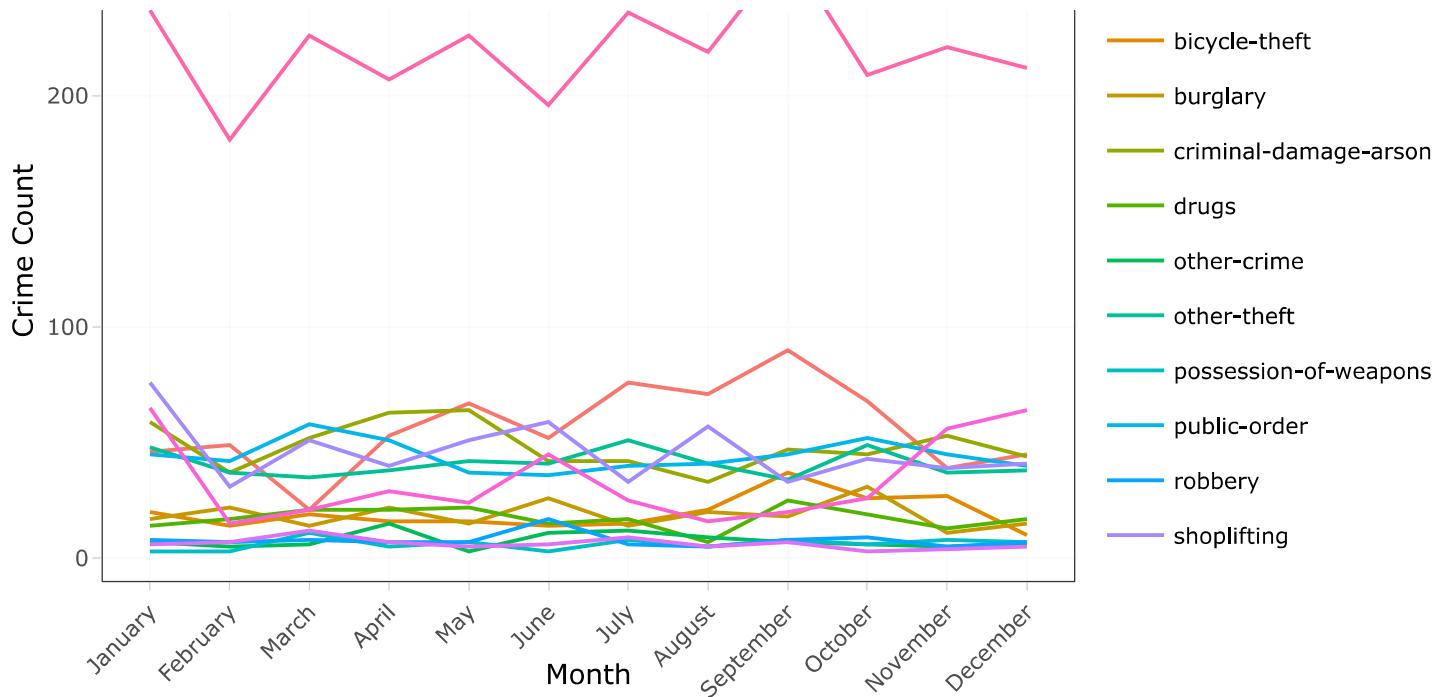
```
# Create a grouped data frame with crime counts by category and month
crime_cat_group <- project_data %>%
  group_by(category, date) %>%
  count() %>%
  mutate(Month_Name = format(as.Date(paste(date, "01", sep = "-")), "%B")) %>%
  mutate(Month_Name = factor(Month_Name, levels = month.name)) # Specify the order of months

# Create the time series plot
grouped_ts <- ggplot(crime_cat_group, aes(x = Month_Name, y = n, group = category, color = category)) +
  geom_line(size = 0.5) + # Add a Line
  labs(x = "Month", y = "Crime Count", title = "Crime Counts of Different Crime Category Over Time") +
  # Add Labels and title
  theme_bw() + # Use a minimal theme
  theme(legend.position = "bottom", legend.title = element_blank(),
        axis.text.x = element_text(angle = 45, vjust = 0.5, hjust = 1))

ggplotly(grouped_ts)
```

Crime Counts of Different Crime Category Over Time





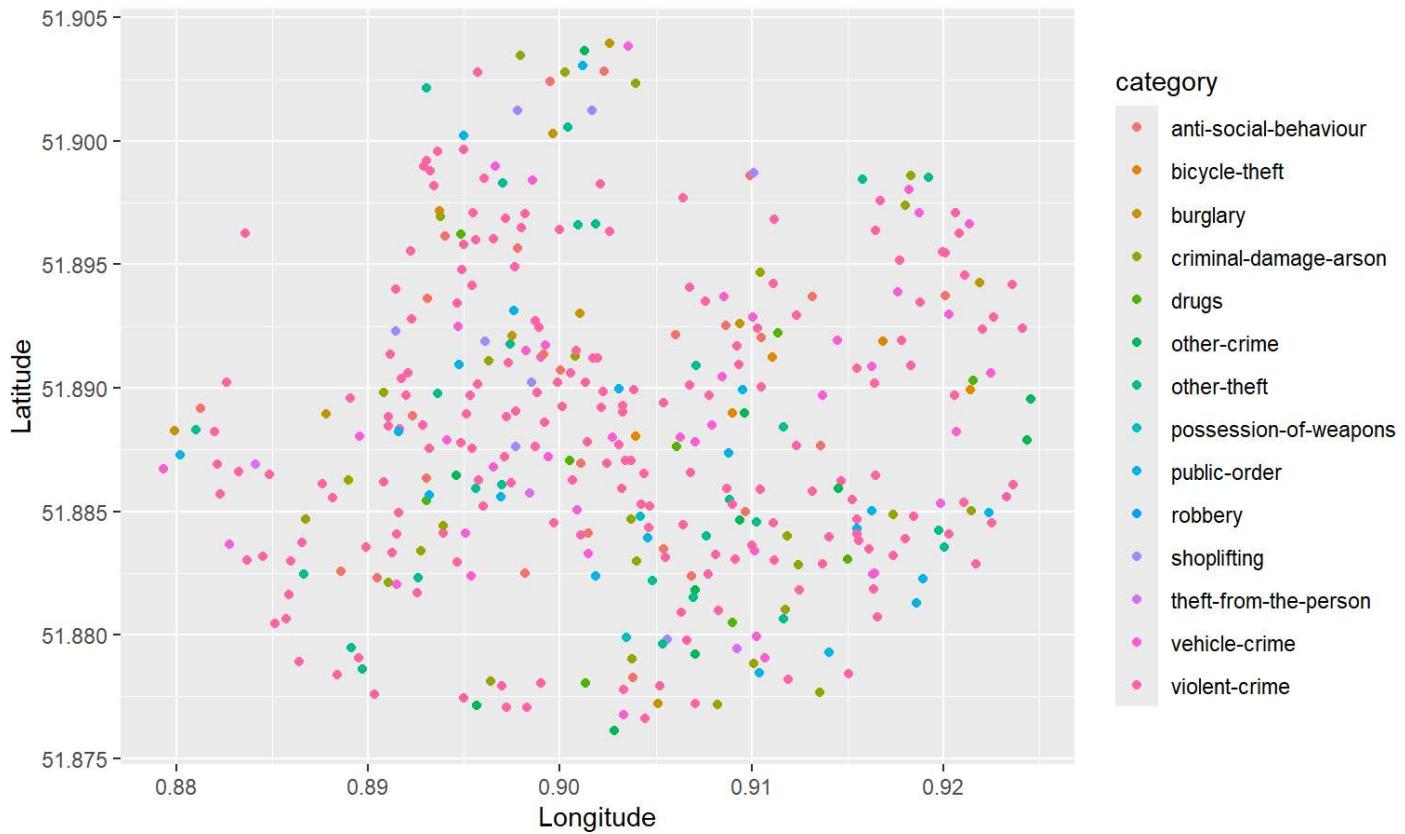
This R code builds a time series plot that depicts the trend of crime counts over time, organised by crime type. The visualisation, which aggregates data from the project data set and plots incident counts against months, provides insights into the temporal trends of various types of crimes. The interactive aspect enhances the exploration, enabling users to hover over the plot for detailed information on specific categories and months. This visualization helps to understand how different crime categories evolve over time, which allows for more informed decision-making and resource allocation for crime prevention strategies. Each line depicts the count of crime by category over the year.

## Scatter plot of Latitude and Longitude

```
library(ggplot2)

# Scatter plot with different colors for each category
ggplot(project_data, aes(x = long, y = lat, color = category)) +
  geom_point() +
  labs(x = "Longitude", y = "Latitude", title = "Scatter Plot with Different Colors for Each Crime Category")
```

### Scatter Plot with Different Colors for Each Crime Category



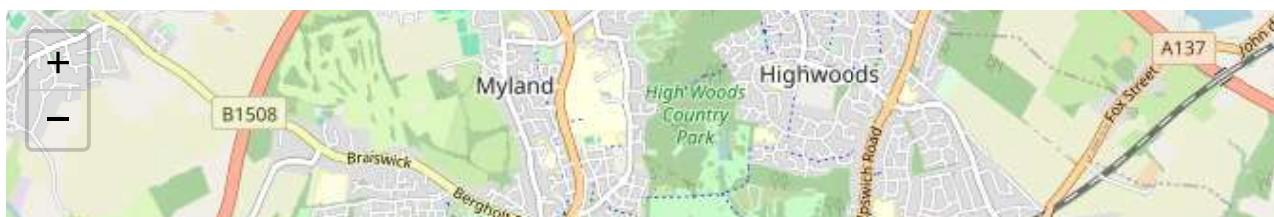
This R code makes a scatter plot of geographical data, namely longitude and latitude, with dots coloured by crime type. By showing the distribution of crimes and categorising them, the visualisation provides insights into the geographic patterns of criminal activity. This enables us to identify hotspots or clusters of certain types of crimes, the higher the cluster the more crimes happened at that location which helps with resource allocation and strategic planning crime prevention operations.

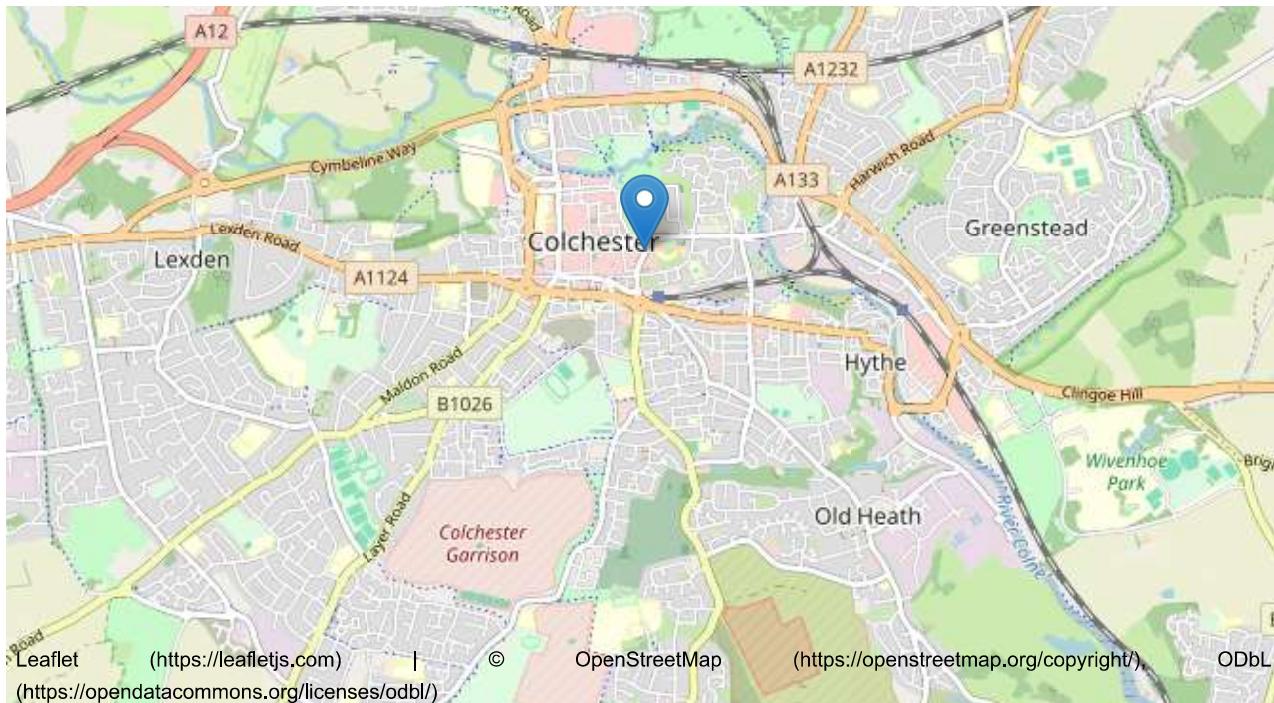
### Leaflet 'Colchester Map'

```
library(leaflet)
# Colchester location
colchester <- c(51.8891, 0.9040)

# Create map
map <- leaflet() %>%
  addTiles() %>%
  setView(lng = colchester[2], lat = colchester[1], zoom = 13) %>%
  addMarkers(lng = colchester[2], lat = colchester[1], popup = "Colchester")

# Display map
map
```





This map depicts the location of the city of Colchester where the crimes occurred in the year 2023.

## Map with crime scenes

```

# Define a color palette for different categories
# Creating a new data frame having only neccessary columns.
df_location <- data.frame(
  inc_type = project_data$category,
  street = project_data$street_name,
  street_id = project_data$street_id,
  latitude = as.numeric(project_data$lat),
  longitude = as.numeric(project_data$long)
)

category_colors <- c(
  "anti-social-behaviour" = "#FF5733", "bicycle-theft" = "#3498DB", "burglary" = "#28B463", "criminal-damage-arson" = "#F4D03F", "drugs" = "#AF7AC5", "other-crime" = "#A569BD", "other-theft" = "#FFC300", "possession-of-weapons" = "#EC7063", "public-order" = "#5DADE2",
  "robbery" = "#F39C12", "shoplifting" = "#85929E", "theft-from-the-person" = "#CB4335", "vehicle-crime" = "#1ABC9C", "violent-crime" = "#E74C3C"
)

# Create a factor variable for category with corresponding colors
df_location$inc_type <- factor(df_location$inc_type, levels = names(category_colors))
pal <- colorFactor(palette = category_colors, domain = df_location$inc_type)

# Create a Leaflet map with circle markers for each incident location in Colchester
map_colchester <- leaflet(df_location) %>%
  addTiles() %>%
  addCircleMarkers(radius = 1,
                  color = ~pal(inc_type), # Assign color based on category
                  popup = paste0("Street: ", df_location$street,
                                "<br>Incident Type: ", df_location$inc_type)) %>%
  setView(lng = 0.9040, lat = 51.8891, zoom = 13.5)

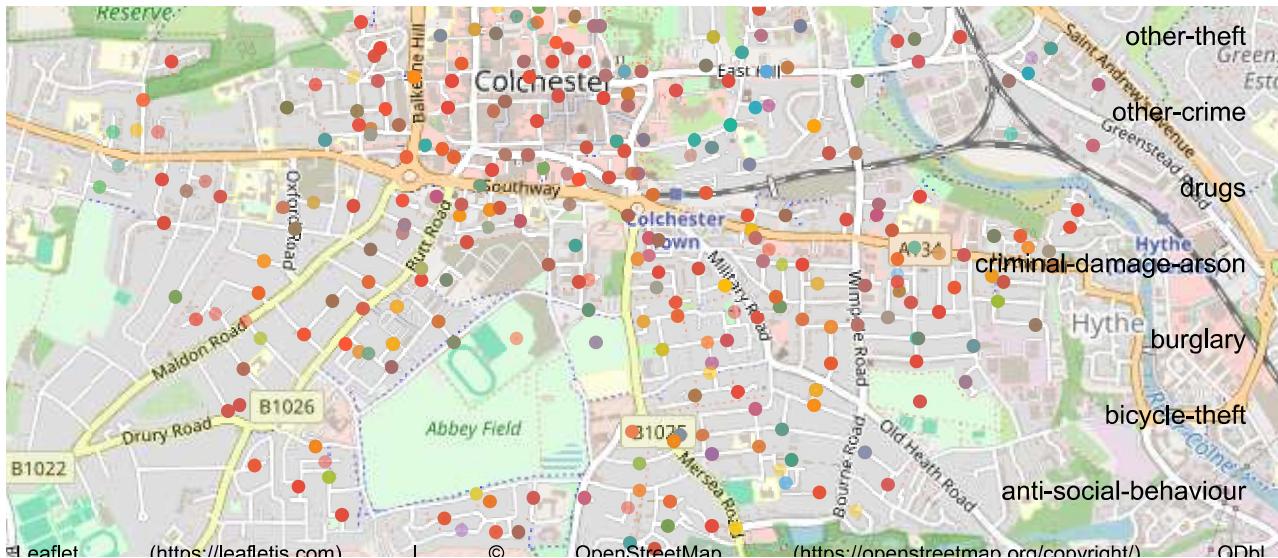
# Legend Labels
legend_labels <- names(category_colors)

for (i in seq_along(legend_labels)) {
  map_colchester <- map_colchester %>%
    addLegend("bottomright", colors = category_colors[legend_labels[i]],
              labels = legend_labels[i], opacity = 0.5)
}

# Display
map_colchester

```





Interactive leaflet map of crime occurrence sites in Colchester, England. Categorised by different crime type. Each occurrence is represented by a circular marker, with different colours given according to the criminal type. This visualisation gives a thorough overview of crime distribution in the area, allowing us to discover geographical trends and hotspots for various types of crimes. The interactive capabilities enable users to investigate the exact location and obtain insights into crime patterns.

## Conclusion

In summary, a variety of approaches have been used in this R programming data visualization project to successfully convey the insights gained from our data set. We have developed a dynamic and captivating experience for examining geographical and temporal data by utilizing Plotly for interactive graphs and the Leaflet library for mapping latitude and longitude coordinates.

Plot types that we have used include tables, bar plots, pie charts, dot plots, density plots, box plots, scatter plots, pair plots, histograms and time series plots. Every graphic has a distinct function in helping to visualize various data elements, such as trends over time or categorical distributions.

To further improve the visualization of patterns and trends and provide a better understanding of the underlying data dynamics, we have also implemented smoothing methods.

Overall, through the thoughtful selection and implementation of visualization techniques, this project has effectively communicated key insights and trends present in the data set, facilitating informed decision-making and deeper exploration of the data.