R-Notebook

November 29, 2024

[25]:
```r
# Simple string assignment
file_path <- "../Datasets/cw_r.xlsx"
```

[26]:
```r
# Install and load necessary packages (if not already installed)
if (!require(readxl)) install.packages("readxl")
if (!require(dplyr)) install.packages("dplyr")
# Load ggplot2
if (!require(ggplot2)) install.packages("ggplot2")
# Load libraries
library(readxl)
library(dplyr)
library(ggplot2)
```

[27]:
```r
# Read the Excel file
dataset <- read_excel(file_path, sheet = "Table 3d")
```

```
New names:
* `` -> `...2`
* `` -> `...3`
* `` -> `...4`
* `` -> `...5`
* `` -> `...6`
* `` -> `...7`
* `` -> `...8`
* `` -> `...9`
* `` -> `...10`
* `` -> `...11`
* `` -> `...12`
```

[28]:
```r
# Preview the dataset to understand its structure
head(dataset)
str(dataset)
```

	Table 3d: Fraud and computer misuse offences referred to the National Fraud Intelligence Burea
	<chr>
A tibble: 6 x 12	England and Wales, year ending March 2013 to year ending March 2022
	Fraud data are not designated as National Statistics.
	This worksheet contains one table. Some cells refer to notes which can be found on the notes wo
	Link to Notes worksheet
	Some shorthand is used in this table. [u] Indicates the percentage change are not reported becau
	Link to Table of contents

```
tibble [75 x 12] (S3: tbl_df/tbl/data.frame)
 $ Table 3d: Fraud and computer misuse offences referred to the National Fraud
Intelligence Bureau (NFIB), with percentage change [note 1, 2, 23, 25]: chr
[1:75] "England and Wales, year ending March 2013 to year ending March 2022"
"Fraud data are not designated as National Statistics." "This worksheet contains
one table. Some cells refer to notes which can be found on the notes worksheet."
"Link to Notes worksheet" …
 $ …2
: chr [1:75] NA NA NA NA …
 $ …3
: chr [1:75] NA NA NA NA …
 $ …4
: chr [1:75] NA NA NA NA …
 $ …5
: chr [1:75] NA NA NA NA …
 $ …6
: chr [1:75] NA NA NA NA …
 $ …7
: chr [1:75] NA NA NA NA …
 $ …8
: chr [1:75] NA NA NA NA …
 $ …9
: chr [1:75] NA NA NA NA …
 $ …10
: chr [1:75] NA NA NA NA …
 $ …11
: chr [1:75] NA NA NA NA …
 $ …12
: chr [1:75] NA NA NA NA …
```

Explanation:

- `read_excel()`: This function is used to read in the specific sheet (**Table 3d**) from the provided Excel file.
- `head()` and `str()`: These functions give you an overview of the dataset.

[29]:
```
# Check the names of the columns in the dataset
colnames(dataset)

# Inspect the first few rows of the dataset
```

```r
head(dataset)

# Get a detailed view of the dataset's structure
str(dataset)
```

1. 'Table 3d: Fraud and computer misuse offences referred to the National Fraud Intelligence Bureau (NFIB), with percentage change [note 1, 2, 23, 25]' 2. '...2' 3. '...3' 4. '...4' 5. '...5' 6. '...6' 7. '...7' 8. '...8' 9. '...9' 10. '...10' 11. '...11' 12. '...12'

A tibble: 6 x 12

Table 3d: Fraud and computer misuse offences referred to the National Fraud Intelligence Burea <chr>
England and Wales, year ending March 2013 to year ending March 2022
Fraud data are not designated as National Statistics.
This worksheet contains one table. Some cells refer to notes which can be found on the notes wo
Link to Notes worksheet
Some shorthand is used in this table. [u] Indicates the percentage change are not reported becau
Link to Table of contents

```
tibble [75 x 12] (S3: tbl_df/tbl/data.frame)
 $ Table 3d: Fraud and computer misuse offences referred to the National Fraud
Intelligence Bureau (NFIB), with percentage change [note 1, 2, 23, 25]: chr
[1:75] "England and Wales, year ending March 2013 to year ending March 2022"
"Fraud data are not designated as National Statistics." "This worksheet contains
one table. Some cells refer to notes which can be found on the notes worksheet."
"Link to Notes worksheet" …
 $ …2
: chr [1:75] NA NA NA NA …
 $ …3
: chr [1:75] NA NA NA NA …
 $ …4
: chr [1:75] NA NA NA NA …
 $ …5
: chr [1:75] NA NA NA NA …
 $ …6
: chr [1:75] NA NA NA NA …
 $ …7
: chr [1:75] NA NA NA NA …
 $ …8
: chr [1:75] NA NA NA NA …
 $ …9
: chr [1:75] NA NA NA NA …
 $ …10
: chr [1:75] NA NA NA NA …
 $ …11
: chr [1:75] NA NA NA NA …
 $ …12
: chr [1:75] NA NA NA NA …
```

[30]:
```
# Read the dataset from the Excel file.
# skipping metadata rows (data starts at row 9)
dataset <- read_excel(file_path, sheet = "Table 3d", skip = 8)

# Inspect the dataset after skipping the metadata
head(dataset)
str(dataset)
```

A tibble: 6 x 12

Reporting body and fraud and computer misuse type [note 3, 4] <chr>	Apr 2012 to Mar 2013 <dbl>	A <dbl>
Banking and credit industry fraud	306641	2
Cheque, plastic card and online bank accounts (not PSP) [note 5, 6]	222272	2
Application fraud (excluding mortgages)	80082	5
Mortgage related fraud	2977	4
Mandate fraud	1300	2
Dishonestly retaining a wrongful credit	10	1

```
tibble [67 x 12] (S3: tbl_df/tbl/data.frame)
 $ Reporting body and fraud and computer misuse type [note 3, 4]
: chr [1:67] "Banking and credit industry fraud" "Cheque, plastic card and
online bank accounts (not PSP) [note 5, 6]" "Application fraud (excluding
mortgages)" "Mortgage related fraud" …
 $ Apr 2012 to
Mar 2013                                                        : num
[1:67] 306641 222272 80082 2977 1300 …
 $ Apr 2013 to
Mar 2014                                                        : num
[1:67] 286234 226087 53390 4337 2288 …
 $ Apr 2014 to
Mar 2015                                                        : num
[1:67] 320495 254479 58277 3832 3738 …
 $ Apr 2015 to
Mar 2016                                                        : num
[1:67] 367714 293683 64166 3544 6193 …
 $ Apr 2016 to
Mar 2017                                                        : num
[1:67] 380390 284607 84752 3095 7800 …
 $ Apr 2017 to
Mar 2018                                                        : num
[1:67] 352064 266538 74027 2708 8618 …
 $ Apr 2018 to
Mar 2019                                                        : num
[1:67] 386326 297234 78845 2324 7752 …
 $ Apr 2019 to
Mar 2020                                                        : num
[1:67] 433336 338130 87033 2085 5929 …
 $ Apr 2020 to
Mar 2021                                                        : num
```

```
[1:67] 384886 298815 80209 1381 4390 …
 $ Apr 2021 to
Mar 2022
[note 28]                                                    : num [1:67] 552024
475038 71475 1297 4084 …
 $ Apr 2021 to Mar 2022
 compared with
 previous year total
 % change
[note 28]: chr [1:67] "43.425325940668145" "58.973947091009492"
"-10.889052350733708" "-6.0825488776249053" …
```

Explanation:

`skip = 8`: This argument skips the first 8 rows (assuming that the data starts from row 9). Adjust this number based on the actual structure.

[31]:
```r
# Rename columns for better readability
colnames(dataset) <- c(
  "Fraud_Type",
  "Year_2012_2013", "Year_2013_2014", "Year_2014_2015",
  "Year_2015_2016", "Year_2016_2017", "Year_2017_2018",
  "Year_2018_2019", "Year_2019_2020", "Year_2020_2021",
  "Year_2021_2022", "Percentage_Change"
)

# View the renamed dataset
head(dataset)
```

A tibble: 6 x 12

Fraud_Type <chr>	Year_2012_2013 <dbl>	Year_ <dbl>
Banking and credit industry fraud	306641	28623
Cheque, plastic card and online bank accounts (not PSP) [note 5, 6]	222272	22608
Application fraud (excluding mortgages)	80082	53390
Mortgage related fraud	2977	4337
Mandate fraud	1300	2288
Dishonestly retaining a wrongful credit	10	132

[32]:
```r
# Filter for "Banking and Credit Industry Fraud"
fraud_df_1 <- dataset %>%
  filter(Fraud_Type == "Banking and credit industry fraud")

# View the basic structure of the first data frame
head(fraud_df_1)
str(fraud_df_1)
```

A tibble: 1 x 12

Fraud_Type <chr>	Year_2012_2013 <dbl>	Year_2013_2014 <dbl>	Year_2014_2015 <dbl>	
Banking and credit industry fraud	306641	286234	320495	

```
tibble [1 x 12] (S3: tbl_df/tbl/data.frame)
 $ Fraud_Type       : chr "Banking and credit industry fraud"
 $ Year_2012_2013   : num 306641
 $ Year_2013_2014   : num 286234
 $ Year_2014_2015   : num 320495
 $ Year_2015_2016   : num 367714
 $ Year_2016_2017   : num 380390
 $ Year_2017_2018   : num 352064
 $ Year_2018_2019   : num 386326
 $ Year_2019_2020   : num 433336
 $ Year_2020_2021   : num 384886
 $ Year_2021_2022   : num 552024
 $ Percentage_Change: chr "43.425325940668145"
```

[33]:
```r
# Filter for "Cheque, Plastic Card, and Online Bank Accounts Fraud"
fraud_df_2 <- dataset %>%
  filter(Fraud_Type == "Cheque, plastic card and online bank accounts (not PSP)␣
 ↪[note 5, 6]") # nolint

# View the basic structure of the second data frame
head(fraud_df_2)
str(fraud_df_2)
```

A tibble: 1 x 12

Fraud_Type	Year_2012_2013	Year_
<chr>	<dbl>	<dbl>
Cheque, plastic card and online bank accounts (not PSP) [note 5, 6]	222272	22608

```
tibble [1 x 12] (S3: tbl_df/tbl/data.frame)
 $ Fraud_Type       : chr "Cheque, plastic card and online bank accounts (not
PSP) [note 5, 6]"
 $ Year_2012_2013   : num 222272
 $ Year_2013_2014   : num 226087
 $ Year_2014_2015   : num 254479
 $ Year_2015_2016   : num 293683
 $ Year_2016_2017   : num 284607
 $ Year_2017_2018   : num 266538
 $ Year_2018_2019   : num 297234
 $ Year_2019_2020   : num 338130
 $ Year_2020_2021   : num 298815
 $ Year_2021_2022   : num 475038
 $ Percentage_Change: chr "58.973947091009492"
```

[34]:
```r
# Display a preview of the data frames
print(fraud_df_1)
print(fraud_df_2)

# Check if the data frames have the correct structure and content
str(fraud_df_1)
```

```
str(fraud_df_2)
```

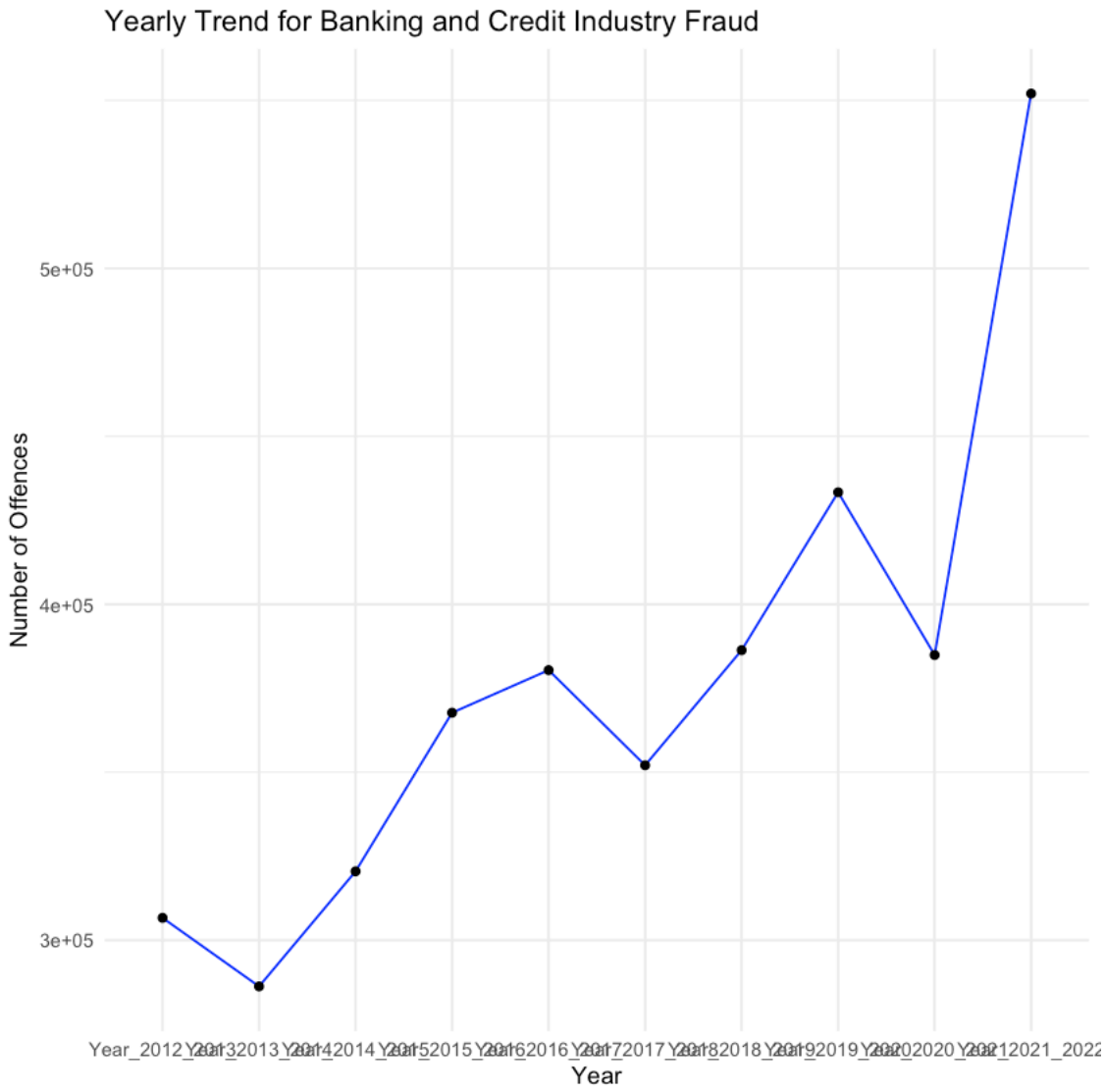
```
# A tibble: 1 x 12
  Fraud_Type         Year_2012_2013 Year_2013_2014 Year_2014_2015 Year_2015_2016
  <chr>                       <dbl>          <dbl>          <dbl>          <dbl>
1 Banking and credi~         306641         286234         320495         367714
# i 7 more variables: Year_2016_2017 <dbl>, Year_2017_2018 <dbl>,
#   Year_2018_2019 <dbl>, Year_2019_2020 <dbl>, Year_2020_2021 <dbl>,
#   Year_2021_2022 <dbl>, Percentage_Change <chr>
# A tibble: 1 x 12
  Fraud_Type         Year_2012_2013 Year_2013_2014 Year_2014_2015 Year_2015_2016
  <chr>                       <dbl>          <dbl>          <dbl>          <dbl>
1 Cheque, plastic c~         222272         226087         254479         293683
# i 7 more variables: Year_2016_2017 <dbl>, Year_2017_2018 <dbl>,
#   Year_2018_2019 <dbl>, Year_2019_2020 <dbl>, Year_2020_2021 <dbl>,
#   Year_2021_2022 <dbl>, Percentage_Change <chr>
tibble [1 x 12] (S3: tbl_df/tbl/data.frame)
 $ Fraud_Type       : chr "Banking and credit industry fraud"
 $ Year_2012_2013   : num 306641
 $ Year_2013_2014   : num 286234
 $ Year_2014_2015   : num 320495
 $ Year_2015_2016   : num 367714
 $ Year_2016_2017   : num 380390
 $ Year_2017_2018   : num 352064
 $ Year_2018_2019   : num 386326
 $ Year_2019_2020   : num 433336
 $ Year_2020_2021   : num 384886
 $ Year_2021_2022   : num 552024
 $ Percentage_Change: chr "43.425325940668145"
tibble [1 x 12] (S3: tbl_df/tbl/data.frame)
 $ Fraud_Type       : chr "Cheque, plastic card and online bank accounts (not
PSP) [note 5, 6]"
 $ Year_2012_2013   : num 222272
 $ Year_2013_2014   : num 226087
 $ Year_2014_2015   : num 254479
 $ Year_2015_2016   : num 293683
 $ Year_2016_2017   : num 284607
 $ Year_2017_2018   : num 266538
 $ Year_2018_2019   : num 297234
 $ Year_2019_2020   : num 338130
```

```
 $ Year_2020_2021   : num 298815
 $ Year_2021_2022   : num 475038
 $ Percentage_Change: chr "58.973947091009492"
```

[35]:
```r
install.packages("tidyr", repos = "https://cran.rstudio.com/")
library(tidyr)
```

```
The downloaded binary packages are in
/var/folders/pb/_45838w13ds5fcrv51k520zh0000gn/T//Rtmps03vVk/downloaded_packages
```

[36]:
```r
# Create a long format version of fraud_df_1 to use in ggplot
fraud_df_1_long <- fraud_df_1 %>%
  pivot_longer(cols = starts_with("Year"), names_to = "Year", values_to =␣
 ↪"Count")  # nolint

# Plot a line graph showing trend over the years for the first fraud type
ggplot(fraud_df_1_long, aes(x = Year, y = Count, group = 1)) +
  geom_line(color = "blue") +
  geom_point() +
  theme_minimal() +
  ggtitle("Yearly Trend for Banking and Credit Industry Fraud") +
  xlab("Year") +
  ylab("Number of Offences")
```

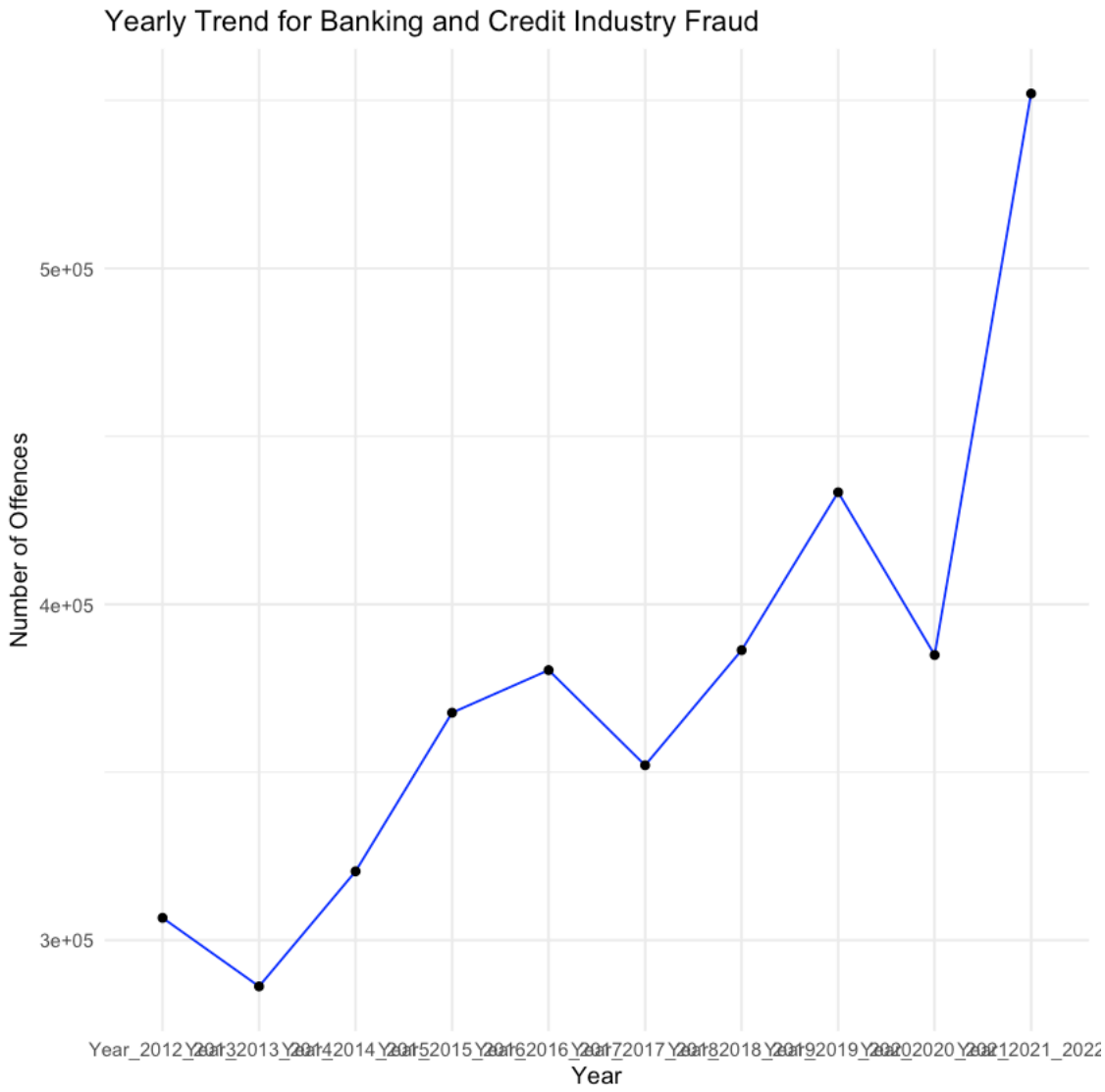


[37]:
```
# Create a long format version of fraud_df_1 to use in ggplot
fraud_df_1_long <- fraud_df_1 %>%
  pivot_longer(cols = starts_with("Year"), names_to = "Year", values_to =␣
 ↪"Count")  # nolint

# View the first few rows of the transformed data
head(fraud_df_1_long)
```

A tibble: 6 x 4

Fraud_Type <chr>	Percentage_Change <chr>	Year <chr>	Count <dbl>
Banking and credit industry fraud	43.425325940668145	Year_2012_2013	306641
Banking and credit industry fraud	43.425325940668145	Year_2013_2014	286234
Banking and credit industry fraud	43.425325940668145	Year_2014_2015	320495
Banking and credit industry fraud	43.425325940668145	Year_2015_2016	367714
Banking and credit industry fraud	43.425325940668145	Year_2016_2017	380390
Banking and credit industry fraud	43.425325940668145	Year_2017_2018	352064

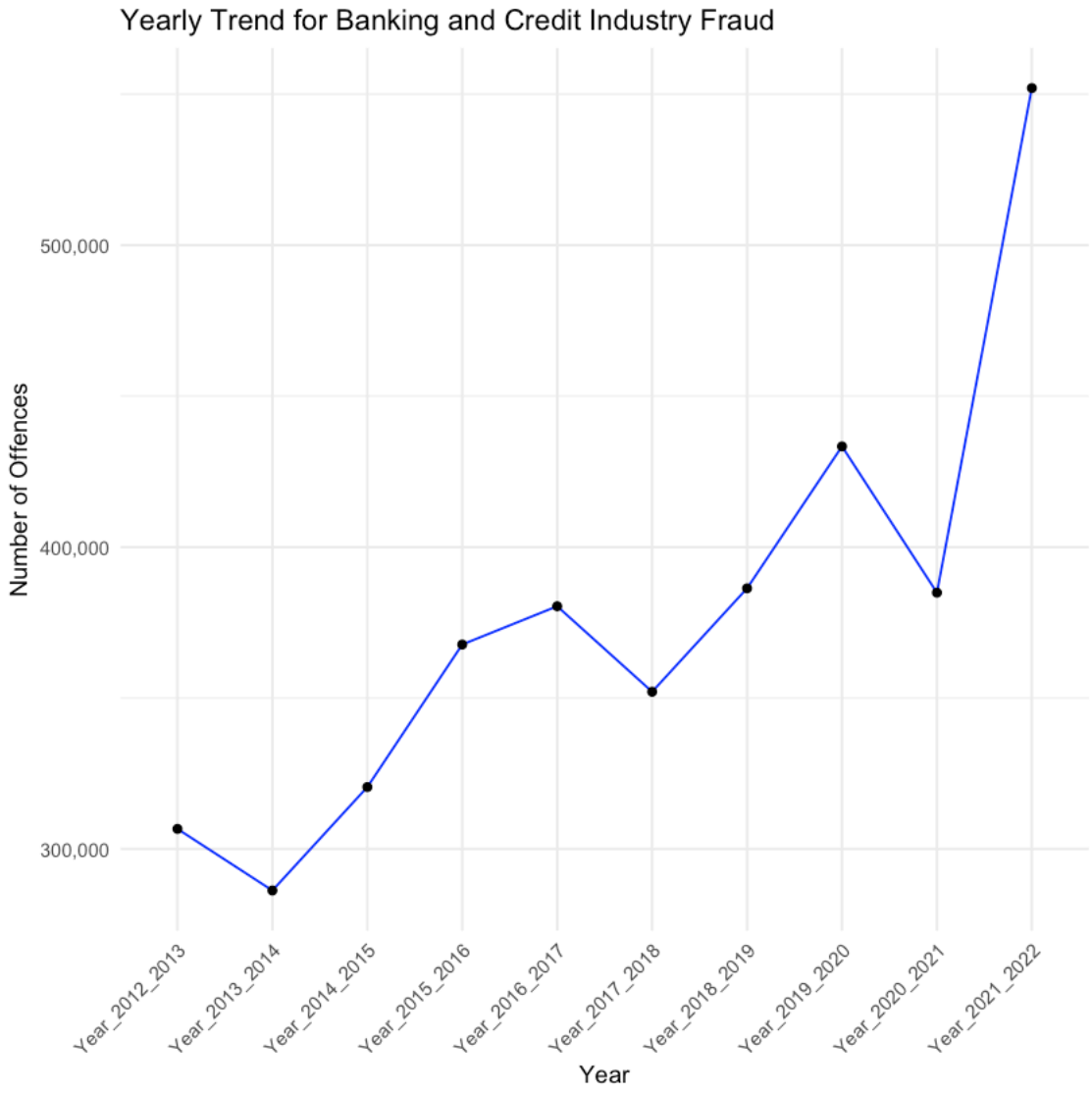
[38]:
```
# Plot a line graph showing the trend over the years for the first fraud type
ggplot(fraud_df_1_long, aes(x = Year, y = Count, group = 1)) +
  geom_line(color = "blue") +
  geom_point() +
  theme_minimal() +
  ggtitle("Yearly Trend for Banking and Credit Industry Fraud") +
  xlab("Year") +
  ylab("Number of Offences")
```



[39]:
```
# Install the tidyverse package (if not already installed)
if (!require(tidyverse)) install.packages("tidyverse")

# Load the tidyverse package
library(tidyverse)
```

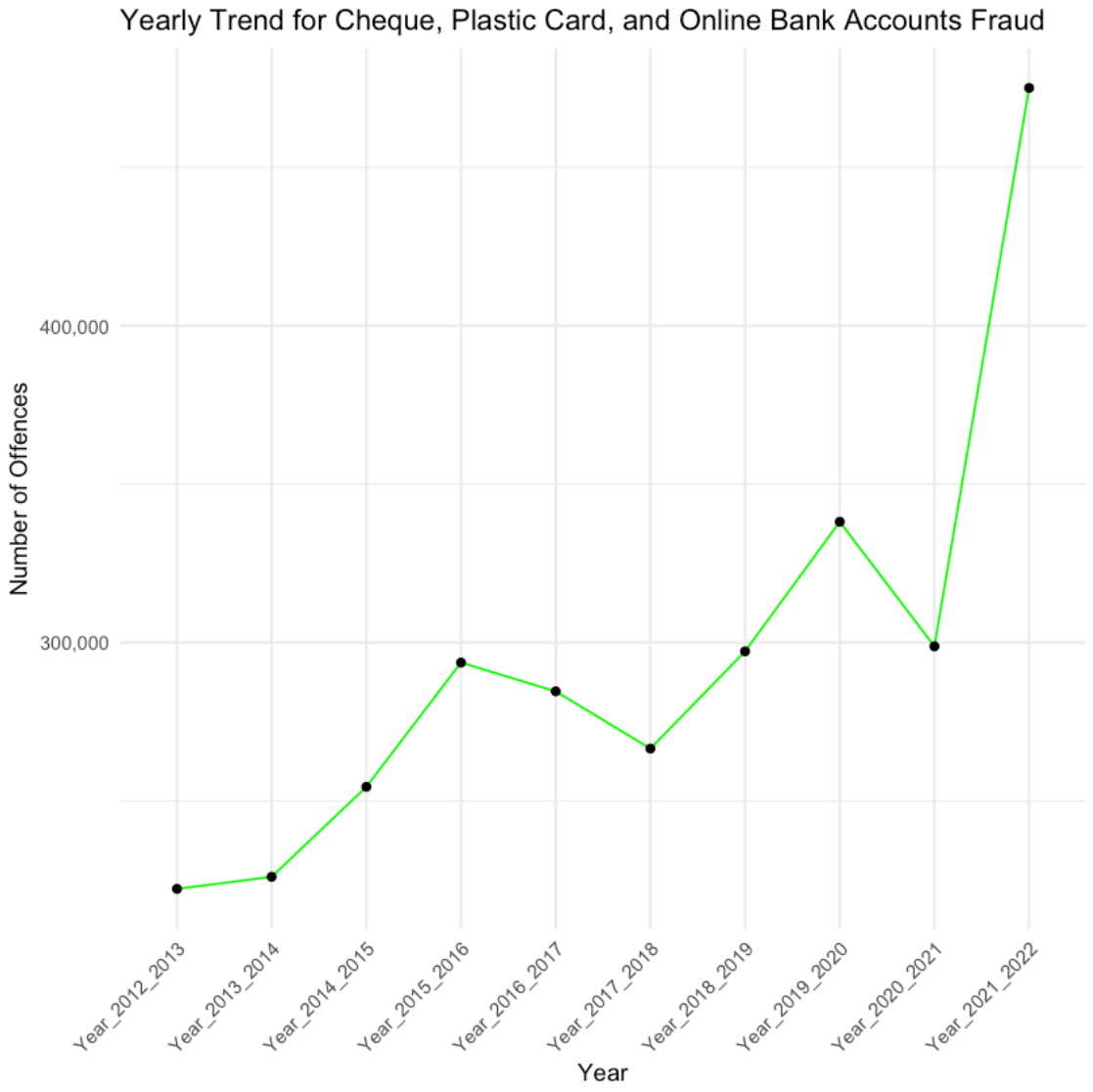
[40]:
```
# Load the scales package for better formatting
if (!require(scales)) install.packages("scales")
library(scales)

# Improved Line Plot with Adjustments for Better Readability
ggplot(fraud_df_1_long, aes(x = Year, y = Count, group = 1)) +
  geom_line(color = "blue") +
```

```
  geom_point() +
  theme_minimal() +
  ggtitle("Yearly Trend for Banking and Credit Industry Fraud") +
  xlab("Year") +
  ylab("Number of Offences") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +  # Rotate X-axis␣
↪labels  # nolint
  scale_y_continuous(labels = comma)  # Use comma formatting for Y-axis
```



[41]:
```
# Convert fraud_df_2 to long format for visualization
fraud_df_2_long <- fraud_df_2 %>%
```

```r
  pivot_longer(cols = starts_with("Year"), names_to = "Year", values_to =␣
 ↪"Count")  # nolint

# Plot a line graph showing trend over the years for fraud_df_2
ggplot(fraud_df_2_long, aes(x = Year, y = Count, group = 1)) +
  geom_line(color = "green") +
  geom_point() +
  theme_minimal() +
  ggtitle("Yearly Trend for Cheque, Plastic Card, and Online Bank Accounts␣
 ↪Fraud") +  # nolint
  xlab("Year") +
  ylab("Number of Offences") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +  # Rotate X-axis␣
 ↪labels  # nolint
  scale_y_continuous(labels = comma)  # Format Y-axis
```



[42]:
```r
# Summary statistics for fraud_df_2 (Cheque, Plastic Card, and Online Bank Accounts Fraud)  # nolint
fraud_summary_2 <- fraud_df_2 %>%
  select(starts_with("Year")) %>%
  summarise(
    Mean = mean(c_across(everything()), na.rm = TRUE),
    Median = median(c_across(everything()), na.rm = TRUE),
    SD = sd(c_across(everything()), na.rm = TRUE),
    Min = min(c_across(everything()), na.rm = TRUE),
    Max = max(c_across(everything()), na.rm = TRUE)
  )

print(fraud_summary_2)
```

```
# A tibble: 1 x 5
     Mean Median     SD    Min    Max
    <dbl>  <dbl>  <dbl>
<dbl>  <dbl>
1 295688. 293683
65291. 65291.
475038
```

[43]:
```r
# Load the knitr package for neat printing
if (!require(knitr)) install.packages("knitr")
library(knitr)

# Summarize fraud_df_1 by year to calculate total offenses over time
fraud_summary_by_year_1 <- fraud_df_1_long %>%
  group_by(Year) %>%
  summarise(Total_Offences = sum(Count, na.rm = TRUE))
print("Summary of fraud_df_1 by year:")
kable(fraud_summary_by_year_1)

# Repeat similar summarization for fraud_df_2
fraud_summary_by_year_2 <- fraud_df_2_long %>%
  group_by(Year) %>%
  summarise(Total_Offences = sum(Count, na.rm = TRUE))
print("Summary of fraud_df_2 by year:")
kable(fraud_summary_by_year_2)
```

```
[1] "Summary of fraud_df_1 by year:"
```

Year	Total_Offences

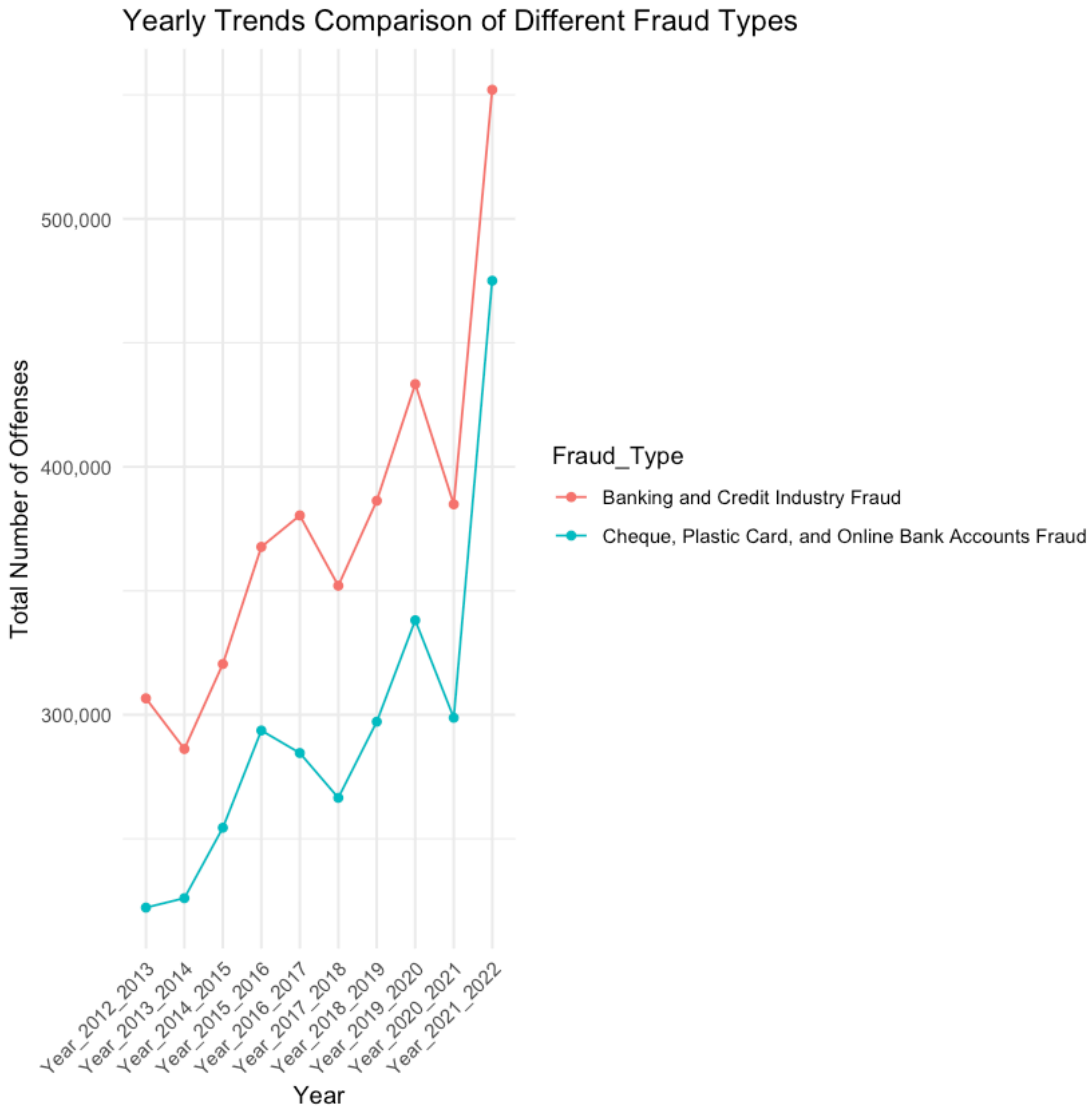
```
|Year_2012_2013 |         306641|
|Year_2013_2014 |         286234|
|Year_2014_2015 |         320495|
|Year_2015_2016 |         367714|
|Year_2016_2017 |         380390|
|Year_2017_2018 |         352064|
|Year_2018_2019 |         386326|
|Year_2019_2020 |         433336|
|Year_2020_2021 |         384886|
|Year_2021_2022 |         552024|
```

```
[1] "Summary of fraud_df_2 by year:"
```

```
|Year           | Total_Offences|
|:--------------|--------------:|
|Year_2012_2013 |         222272|
|Year_2013_2014 |         226087|
|Year_2014_2015 |         254479|
|Year_2015_2016 |         293683|
|Year_2016_2017 |         284607|
|Year_2017_2018 |         266538|
|Year_2018_2019 |         297234|
|Year_2019_2020 |         338130|
|Year_2020_2021 |         298815|
|Year_2021_2022 |         475038|
```

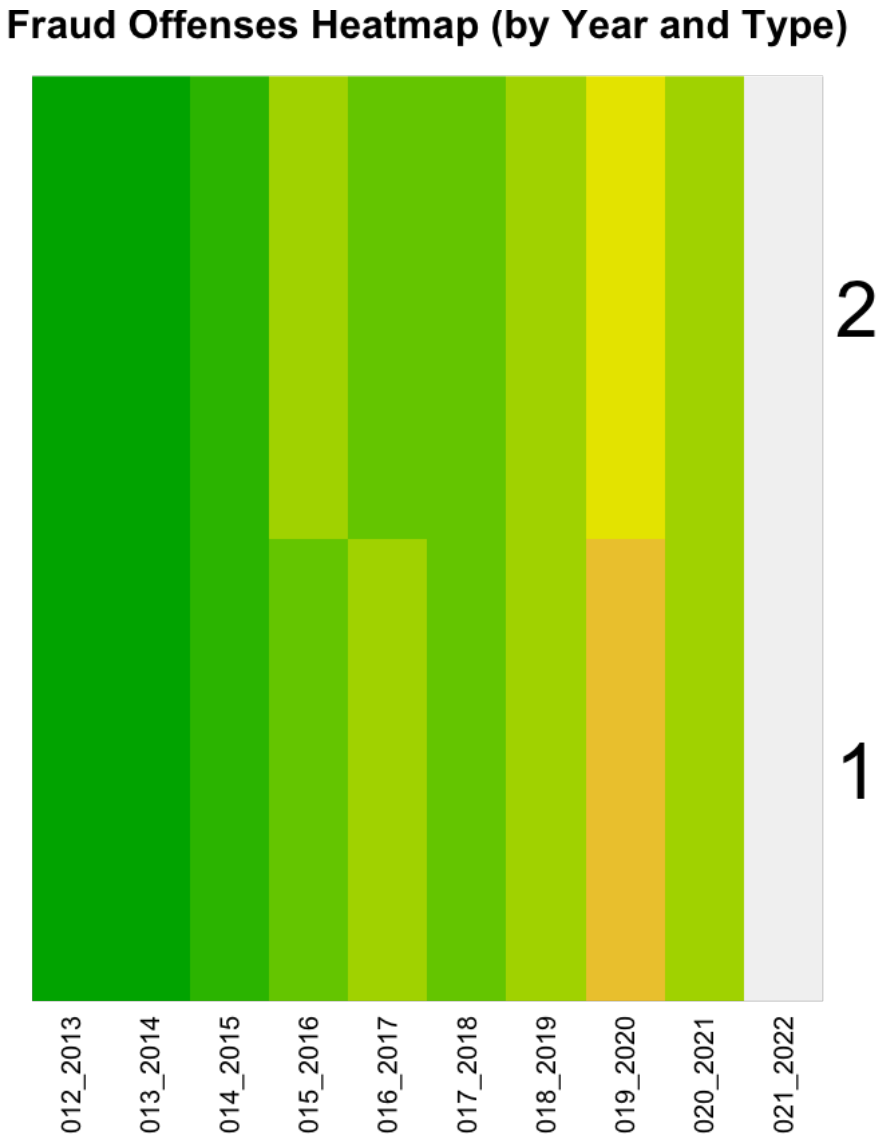
[44]:
```r
# Combine summaries for both fraud types into one data frame
fraud_combined_summary <- bind_rows(
  fraud_summary_by_year_1 %>% mutate(Fraud_Type = "Banking and Credit Industry␣
 ↪Fraud"),  # nolint
  fraud_summary_by_year_2 %>% mutate(Fraud_Type = "Cheque, Plastic Card, and␣
 ↪Online Bank Accounts Fraud")  # nolint
)

# Plotting trends for both fraud types on the same graph
ggplot(fraud_combined_summary, aes(x = Year, y = Total_Offences, color =␣
 ↪Fraud_Type, group = Fraud_Type)) +  # nolint
  geom_line() +
  geom_point() +
  theme_minimal() +
  ggtitle("Yearly Trends Comparison of Different Fraud Types") +
  xlab("Year") +
  ylab("Total Number of Offenses") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +  # Rotate X-axis␣
 ↪labels for better readability  # nolint
```

```
  scale_y_continuous(labels = comma)  # Add commas to Y-axis values for␣
↪readability  # nolint
```



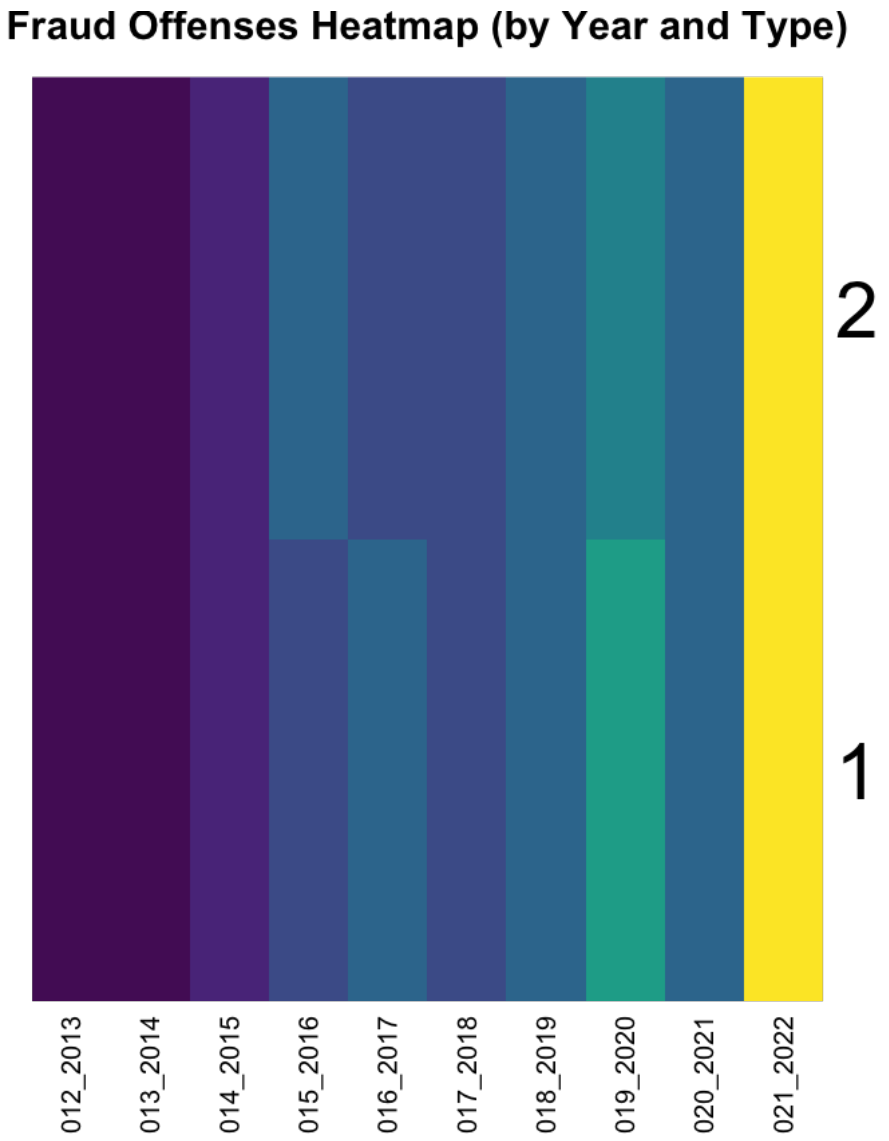
[45]:
```
# Heatmap showing offenses over years for both fraud types
fraud_heatmap_data <- pivot_wider(
  fraud_combined_summary,
  names_from = Year,
  values_from = Total_Offences
)

fraud_heatmap <- as.matrix(fraud_heatmap_data %>% select(-Fraud_Type))
```

```
heatmap(fraud_heatmap, Rowv = NA, Colv = NA, col = terrain.colors(10), scale =␣
↪"row",  # nolint
        margins = c(5, 10), main = "Fraud Offenses Heatmap (by Year and Type)")
```



[46]:
```
# Adjusting the color palette to provide better visualization
if (!require(viridis)) install.packages("viridis")
library(viridis)

# Generate heatmap with improved color settings
fraud_heatmap_data <- pivot_wider(
  fraud_combined_summary,
  names_from = Year,
```

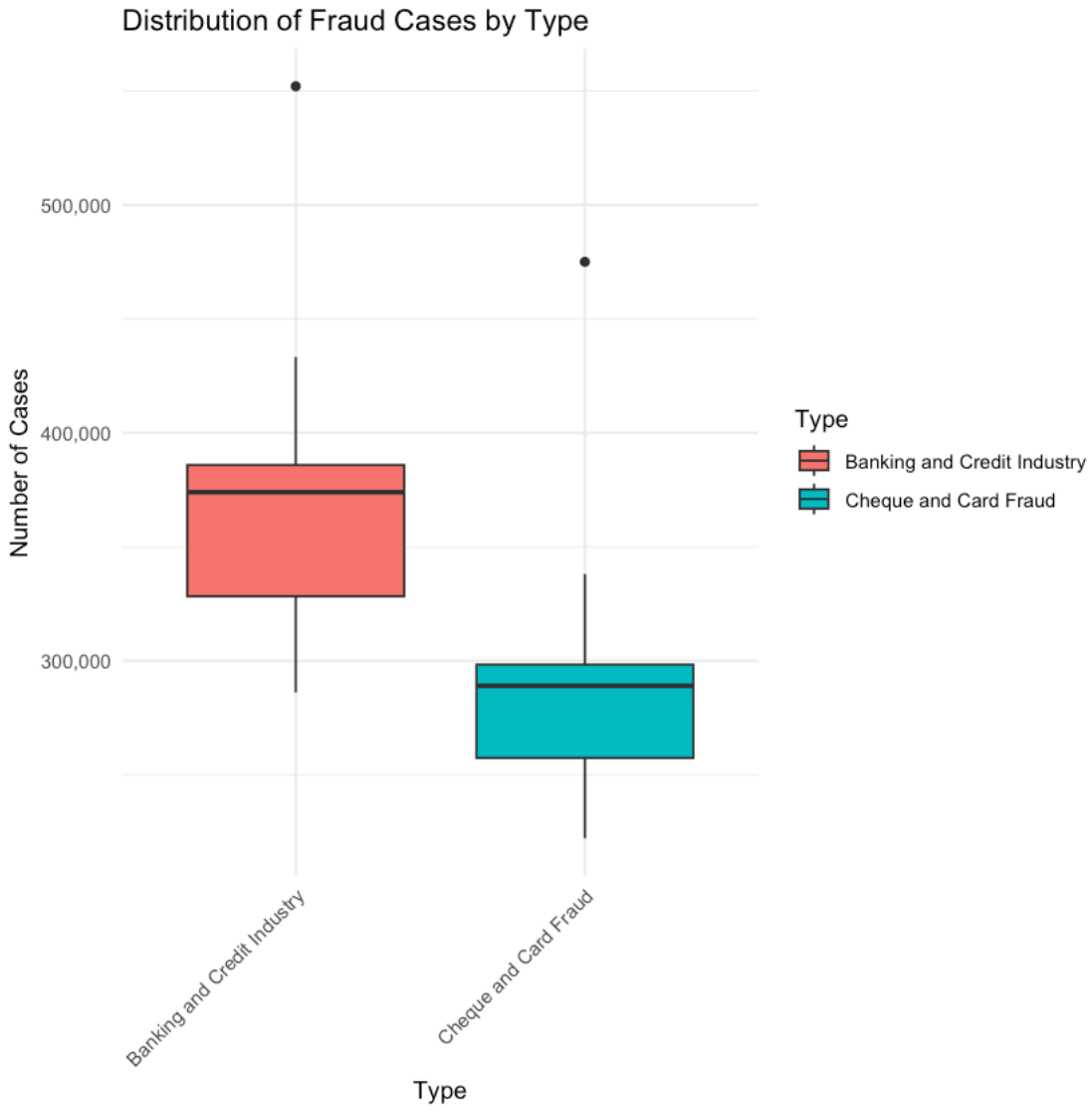
```
  values_from = Total_Offences
)

fraud_heatmap <- as.matrix(fraud_heatmap_data %>% select(-Fraud_Type))

heatmap(fraud_heatmap, Rowv = NA, Colv = NA, col = viridis::viridis(10), scale␣
↪= "row",  # nolint
        margins = c(5, 10), main = "Fraud Offenses Heatmap (by Year and Type)")
```

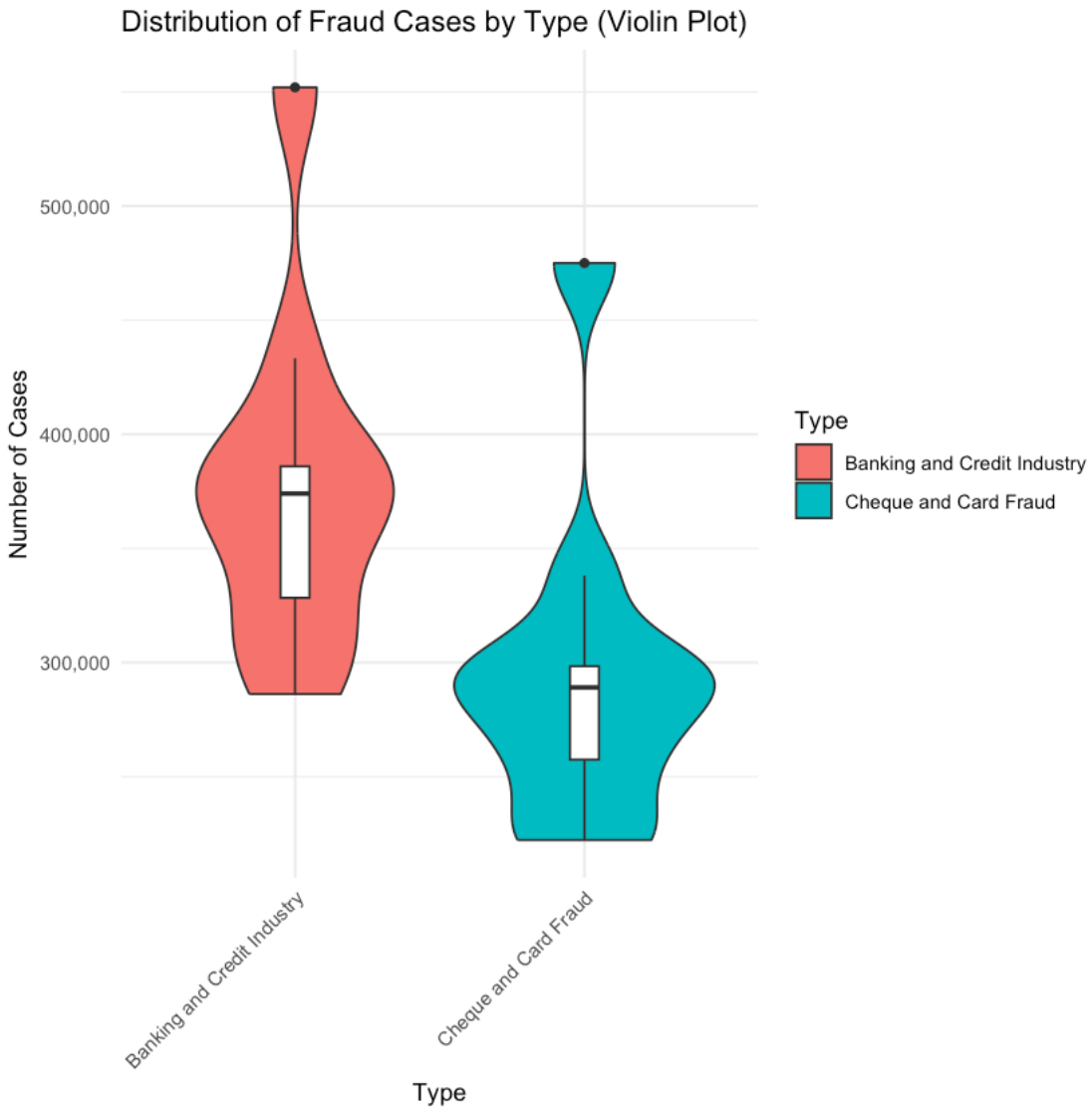


[47]:
```
# Calculate central tendency and dispersion metrics for both datasets
# Banking and Credit Industry Fraud
fraud_df_1_stats <- fraud_df_1 %>%
```

```r
 select(starts_with("Year")) %>%
 summarise(
   Mean = mean(c_across(everything()), na.rm = TRUE),
   Median = median(c_across(everything()), na.rm = TRUE),
   SD = sd(c_across(everything()), na.rm = TRUE),
   IQR = IQR(c_across(everything()), na.rm = TRUE),
   Min = min(c_across(everything()), na.rm = TRUE),
   Max = max(c_across(everything()), na.rm = TRUE)
 )

# Cheque, Plastic Card and Online Bank Accounts Fraud
fraud_df_2_stats <- fraud_df_2 %>%
 select(starts_with("Year")) %>%
 summarise(
   Mean = mean(c_across(everything()), na.rm = TRUE),
   Median = median(c_across(everything()), na.rm = TRUE),
   SD = sd(c_across(everything()), na.rm = TRUE),
   IQR = IQR(c_across(everything()), na.rm = TRUE),
   Min = min(c_across(everything()), na.rm = TRUE),
   Max = max(c_across(everything()), na.rm = TRUE)
 )

# Create box plots for both fraud types
# Convert data to long format for boxplot
fraud_df_1_long <- fraud_df_1 %>%
 select(starts_with("Year")) %>%
 pivot_longer(cols = everything(), names_to = "Year", values_to = "Count")

fraud_df_2_long <- fraud_df_2 %>%
 select(starts_with("Year")) %>%
 pivot_longer(cols = everything(), names_to = "Year", values_to = "Count")

# Add fraud type identifier
fraud_df_1_long$Type <- "Banking and Credit Industry"
fraud_df_2_long$Type <- "Cheque and Card Fraud"

# Combine datasets
combined_long <- bind_rows(fraud_df_1_long, fraud_df_2_long)

# Create boxplot
ggplot(combined_long, aes(x = Type, y = Count, fill = Type)) +
 geom_boxplot() +
 theme_minimal() +
 ggtitle("Distribution of Fraud Cases by Type") +
 ylab("Number of Cases") +
 scale_y_continuous(labels = comma) +
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

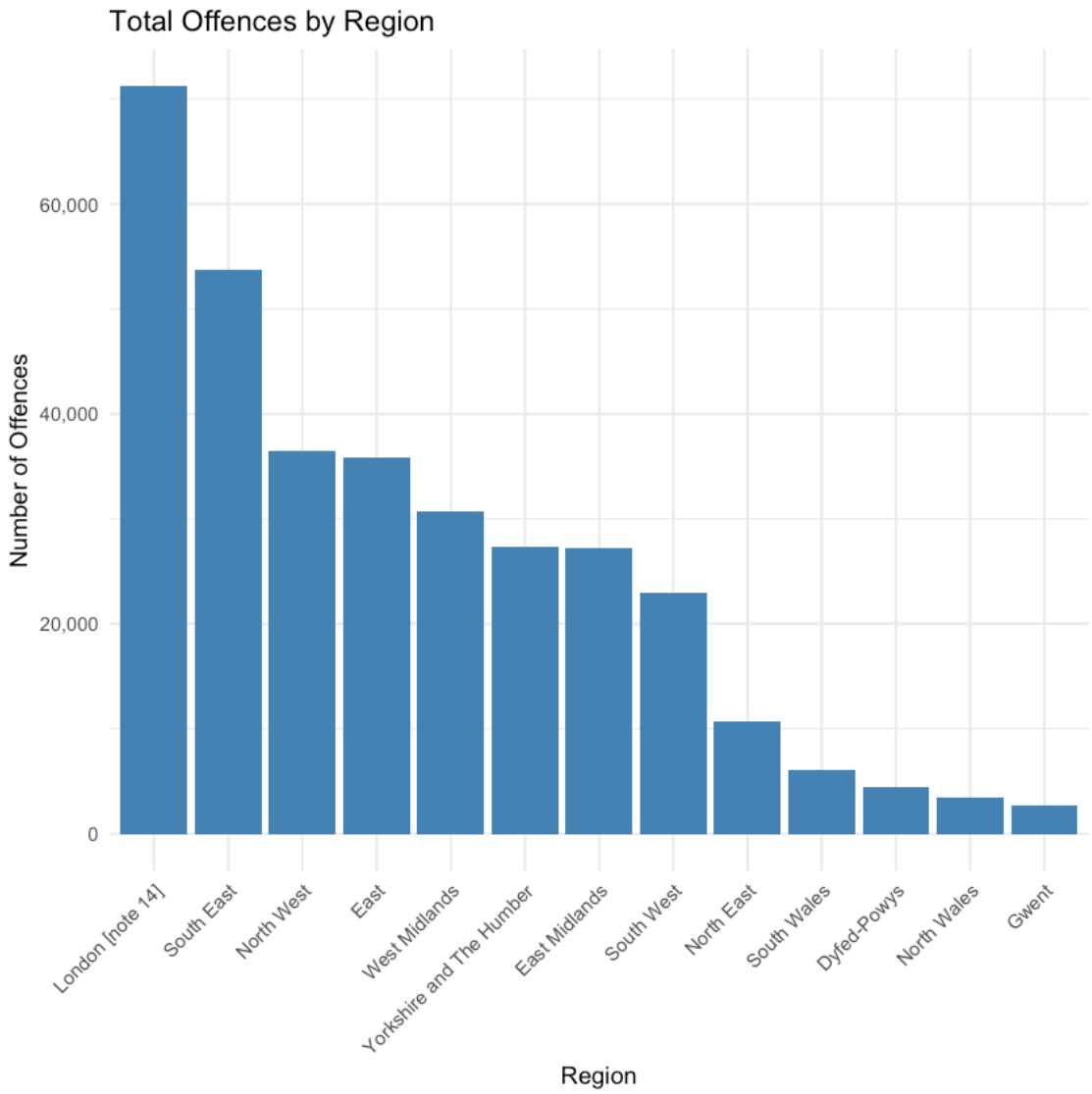
```r
# Create violin plot
ggplot(combined_long, aes(x = Type, y = Count, fill = Type)) +
 geom_violin() +
 geom_boxplot(width = 0.1, fill = "white") +
 theme_minimal() +
 ggtitle("Distribution of Fraud Cases by Type (Violin Plot)") +
 ylab("Number of Cases") +
 scale_y_continuous(labels = comma) +
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

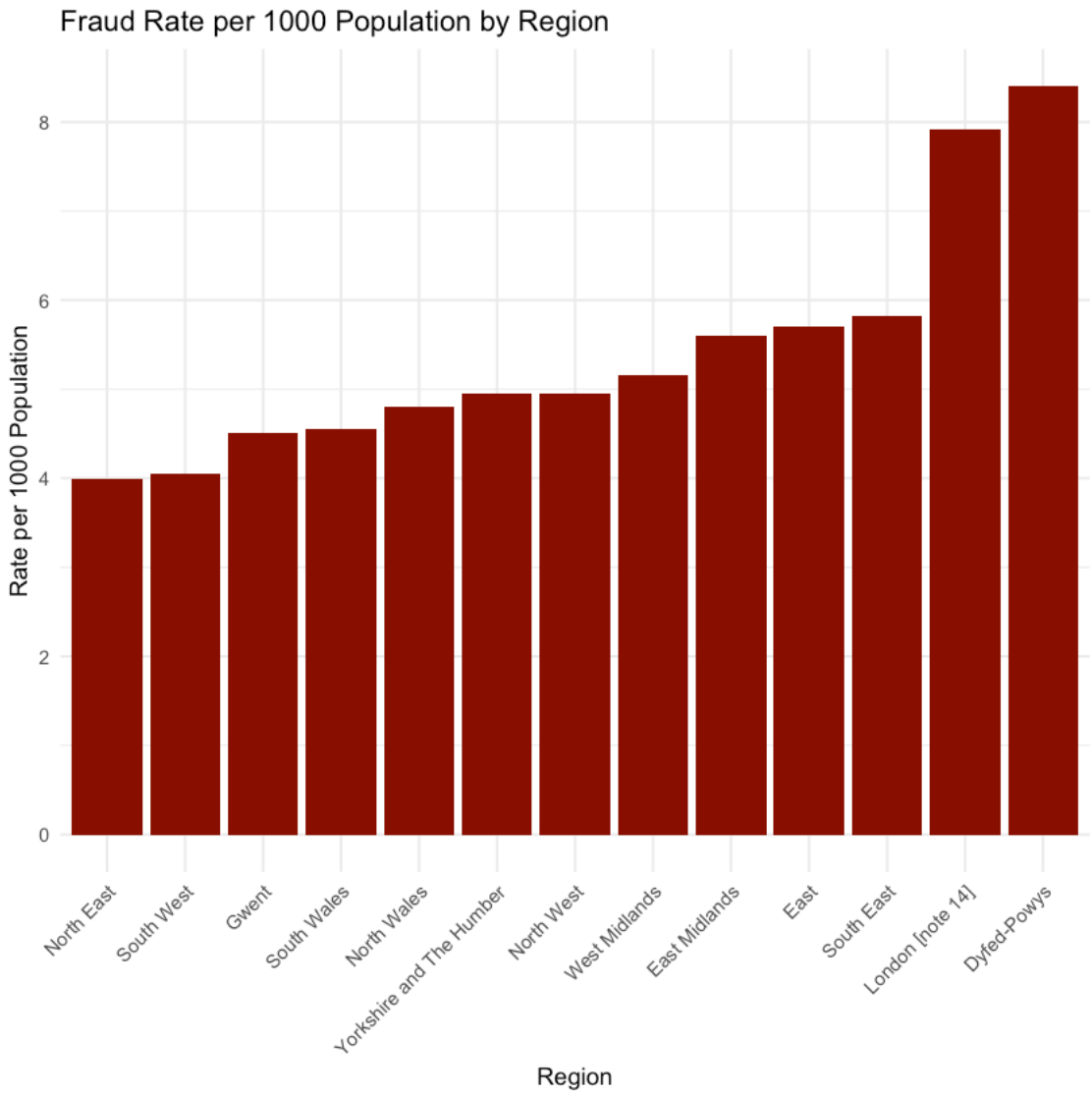


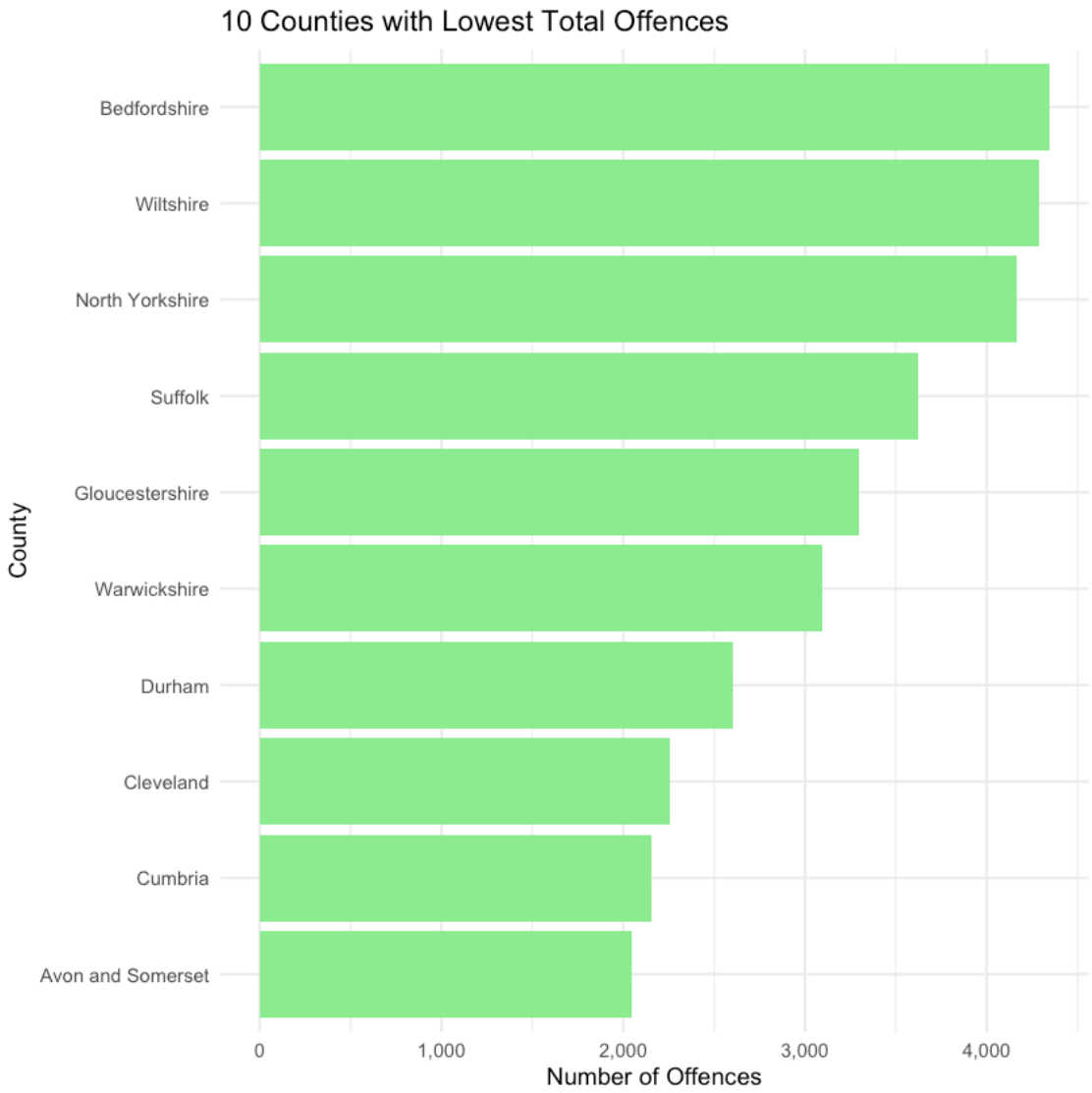


[56]:
```
# Load Table 5 with correct skip
table5_data <- read_excel(file_path, sheet = "Table 5", skip = 9, range = "A10:
 ↪E66")

# Rename columns for clarity
table5_df <- table5_data %>%
  rename(
    Area_Code = `Area Code`,
    Area_Name = `Area Name`,
    Offences = `Number of offences`,
    Rate = `Rate per 1,000 population`,
    Change = `% change from\r\n previous year`
  ) %>%
```

```r
  filter(!Rate %in% c("[z]", "[u1]")) %>%  # Filter first
  mutate(Rate = as.numeric(Rate))          # Then convert to numeric

# Create region-level visualization
regions_df <- table5_df %>%
  filter(str_detect(Area_Code, "^E12|^W15")) %>%
  arrange(desc(Offences))

# Plot for regional offences and rates
p1 <- ggplot(regions_df, aes(x = reorder(Area_Name, -Offences), y = Offences)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Total Offences by Region",
       x = "Region",
       y = "Number of Offences") +
  scale_y_continuous(labels = comma)

p2 <- ggplot(regions_df, aes(x = reorder(Area_Name, Rate), y = Rate)) +
  geom_bar(stat = "identity", fill = "darkred") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Fraud Rate per 1000 Population by Region",
       x = "Region",
       y = "Rate per 1000 Population")

# Create county-level visualization
counties_df <- table5_df %>%
  filter(str_detect(Area_Code, "^E23")) %>%
  arrange(Offences)

# Plot lowest offence counties
p3 <- counties_df %>%
  head(10) %>%
  ggplot(aes(x = reorder(Area_Name, Offences), y = Offences)) +
  geom_bar(stat = "identity", fill = "lightgreen") +
  coord_flip() +
  theme_minimal() +
  labs(title = "10 Counties with Lowest Total Offences",
       x = "County",
       y = "Number of Offences") +
  scale_y_continuous(labels = comma)

# Display plots
print(p1)
print(p2)
print(p3)
```


Total Offences by Region
Number of Offences
60,000
40,000
20,000
0
London [note 14]
South East
North West
East
West Midlands
Yorkshire and The Humber
East Midlands
South West
North East
South Wales
Dyfed-Powys
North Wales
Gwent
Region


Fraud Rate per 1000 Population by Region
Rate per 1000 Population
0
2
4
6
8
North East
South West
Gwent
South Wales
North Wales
Yorkshire and The Humber
North West
West Midlands
East Midlands
East
South East
London [note 14]
Dyfed-Powys
Region


10 Counties with Lowest Total Offences
Bedfordshire
Wiltshire
North Yorkshire
Suffolk
Gloucestershire
Warwickshire
Durham
Cleveland
Cumbria
Avon and Somerset
County
0
1,000
2,000
3,000
4,000
Number of Offences