Assignment 3 - Weeks 5 & 6 Exercise - Kia Thefts

Rohit Patil

2025-07-06

The Viral Vulnerability: A Story of Stolen Kias and Hyundais

1. Load the required Libraries to perform analysis

```
# Load necessary libraries
# dplyr: for data manipulation (filtering, grouping, summarizing, renaming, mutating)
# tidyr: for reshaping data (pivot_longer, pivot_wider)
# ggplot2: for creating various types of plots
# maps: for geographical data to create maps
# treemapify: for creating treemap visualizations
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(treemap)
library(maps)
library(sf)
```

```
## Linking to GEOS 3.13.0, GDAL 3.8.5, PROJ 9.5.1; sf_use_s2() is TRUE
```

```
library(tidyr)
library(readxl)
options(warn = -1)
```

2. Load datasets for analysis

```
# carTheftsMap.csv: Contains aggregated car theft data by agency and year, including geographic coordin
# Motherboard VICE News Kia Hyundai Theft Data.xlsx: Contains monthly Kia/Hyundai and all car theft dat
# kiaHyundaiThefts.csv: Contains Kia Hyundai theft for each city by month and year.
# KiaHyundaiMilwaukeeData.csv: Contains Kia Hyundai theft for Milwaukee, WI city by month and year.
car_thefts_map <- read.csv("carTheftsMap.csv")
file_path <- "Motherboard VICE News Kia Hyundai Theft Data.xlsx"
```

```
motherboard_data_xlsx <- read_excel(file_path, n_max = 5, col_names = FALSE, .name_repair = "minimal")
kia_hyundai_thefts <- read.csv("kiaHyundaiThefts.csv")
kia_hyundai_milwaukee_data <- read.csv("KiaHyundaiMilwaukeeData.csv")
```

3. Convert the excel format to CSV

```
suppressWarnings({
# Step 1: Determine the maximum number of columns by reading a few rows
max_cols <- ncol(motherboard_data_xlsx)

# Step 2: Read the two header rows, explicitly setting the number of columns.
city_headers_df <- read_excel(file_path, n_max = 1, col_names = FALSE, .name_repair = "minimal", range =
metric_headers_df <- read_excel(file_path, skip = 1, n_max = 1, col_names = FALSE, .name_repair = "minim

# Step 3: Read the main data, explicitly setting the number of columns.
all_data <- read_excel(file_path, skip = 2, col_names = FALSE, .name_repair = "minimal", range = cell_c
num_data_cols <- ncol(all_data)

# Step 4: Manually perform a forward-fill for the city names.
filled_cities_for_data_cols <- character(num_data_cols - 1)
current_city <- ""
for (i in 2:num_data_cols) {
    if (!is.na(city_headers_df[1, i])) {
        current_city <- as.character(city_headers_df[1, i])
    }
    filled_cities_for_data_cols[i - 1] <- current_city
}

# Step 5: Generate the repeating metric names for pasting
metric_names_pattern <- c("Kia/Hyundais", "All", "Percent")
metric_names_for_pasting <- rep(metric_names_pattern, length.out = num_data_cols - 1)

# Step 6: Combine the filled city names with the generated metric names
combined_data_headers <- paste(filled_cities_for_data_cols, metric_names_for_pasting, sep = "_")

# Step 7: Create the final, complete list of headers.
final_headers <- c("Date", combined_data_headers)

# Step 8: Assign the robustly generated headers to the dataframe.
names(all_data) <- final_headers

# --- Data Processing ---

# FINAL CORRECTED DATE CONVERSION: Directly convert POSIXct to Date.
all_data <- all_data %>%
    mutate(Date = as.Date(Date))

# --- Reshaping Data (Tidying) ---

tidy_data <- all_data %>%
  pivot_longer(
    cols = -Date,
```

```
    names_to = "City_Metric",
    values_to = "Value",
    values_transform = list(Value = as.numeric)
  ) %>%
  separate(City_Metric, into = c("City", "Metric"), sep = "_", extra = "merge") %>%
  filter(!is.na(Value) & !is.na(Date))

final_df <- tidy_data %>%
  pivot_wider(
    names_from = Metric,
    values_from = Value
  )

# --- DIAGNOSTIC: Print names of final_df before renaming ---
print("Names of final_df before renaming:")
print(names(final_df))

# Rename columns to be more analysis-friendly
final_df <- final_df %>%
  rename(
    Kia_Hyundai_Thefts = `Kia/Hyundais`,
    All_Thefts = `All`,
    Percent_Kia_Hyundai = `Percent`
  )

# --- Diagnostic: Print Summary of final_df ---
print("Summary of final_df before writing to CSV:")
print(summary(final_df))

# --- Output to CSV ---

# Write the final, tidy data frame to a new CSV file
output_path <- "Motherboard_VICE_News_Kia_Hyundai_Theft_Data_Cleaned.csv"
write.csv(final_df, output_path, row.names = FALSE)
})
```

```
## [1] "Names of final_df before renaming:"
## [1] "Date"         "City"         "Kia/Hyundais" "All"          "Percent"
## [1] "Summary of final_df before writing to CSV:"
##       Date                City           Kia_Hyundai_Thefts   All_Thefts
##  Min.   :2019-12-01   Length:3081        Min.   :   0.00    Min.   :   0.0
##  1st Qu.:2020-11-01   Class :character   1st Qu.:   3.00    1st Qu.:  62.0
##  Median :2021-10-01   Mode  :character   Median :   7.00    Median : 107.0
##  Mean   :2021-10-10                      Mean   :  42.09    Mean   : 257.1
##  3rd Qu.:2022-09-01                      3rd Qu.:  22.00    3rd Qu.: 326.0
##  Max.   :2023-08-01                      Max.   :1431.00    Max.   :3182.0
##                                          NA's   :145        NA's   :143
##  Percent_Kia_Hyundai
##  Min.   :0.0000
##  1st Qu.:0.0299
##  Median :0.0563
##  Mean   :0.1078
##  3rd Qu.:0.1207
##  Max.   :0.8179
```

```
##  NA's   :350
```

4. Load the transformed dataset for analysis

```r
motherboard_data <- read.csv("Motherboard_VICE_News_Kia_Hyundai_Theft_Data_Cleaned.csv")
```

5. Data Cleaning and Preparation

```r
# Clean motherboard_data:
# Convert the 'Date' column to a proper Date object for time-series analysis.
motherboard_data$Date <- as.Date(motherboard_data$Date)
# Create new columns for consistency and clarity in plotting.
# countKiaHyundaiThefts: Directly uses the 'Kia_Hyundai_Thefts' column.
# countOtherThefts: Calculated by subtracting 'Kia_Hyundai_Thefts' from 'All_Thefts'. This isolates non
# percentKiaHyundai: Directly uses the 'Percent_Kia_Hyundai' column.
motherboard_data <- motherboard_data %>%
  mutate(
    countKiaHyundaiThefts = Kia_Hyundai_Thefts,
    countOtherThefts = All_Thefts - Kia_Hyundai_Thefts,
    percentKiaHyundai = Percent_Kia_Hyundai
  ) %>%
  # Filter out rows where 'All_Thefts' is missing (NA) or zero.
  # This is crucial because a zero or missing total theft count would lead to meaningless percentages o
  filter(!is.na(All_Thefts) & All_Thefts > 0)

# Handle NA values in 'countOtherThefts'.
# If 'All_Thefts' was present but 'Kia_Hyundai_Thefts' was NA, 'countOtherThefts' could become NA.
# This line ensures that any such NA values are treated as 0, assuming no other thefts if Kia/Hyundai t
motherboard_data$countOtherThefts[is.na(motherboard_data$countOtherThefts)] <- 0

# Clean car_thefts_map data:
# Convert car theft count columns (e.g., countCarThefts2019) to numeric.
# The gsub function removes commas from numbers (e.g., "1,000" becomes "1000") before conversion.
car_thefts_map$countCarThefts2019 <- as.numeric(gsub(",", "", car_thefts_map$countCarThefts2019))
car_thefts_map$countCarThefts2020 <- as.numeric(gsub(",", "", car_thefts_map$countCarThefts2020))
car_thefts_map$countCarThefts2021 <- as.numeric(gsub(",", "", car_thefts_map$countCarThefts2021))
car_thefts_map$countCarThefts2022 <- as.numeric(gsub(",", "", car_thefts_map$countCarThefts2022))

# Filter out rows where 'percentChange2019to2022' is NA.
# These rows do not provide meaningful percentage change data for the map visualization.
car_thefts_map_cleaned <- car_thefts_map %>%
  filter(!is.na(percentChange2019to2022))
```

6. Visualizations

6.1 Visual 1: Stacked Area Chart - Monthly Kia/Hyundai vs. Other Thefts in Milwaukee

```r
# Purpose: To visually demonstrate the dramatic shift in the proportion of Kia/Hyundai thefts
# relative to other car thefts over time in Milwaukee, a city heavily impacted by the issue.
# Justification: A stacked area chart is ideal for showing how the composition of a total
# (total thefts) changes over time, highlighting the increasing dominance of Kia/Hyundai thefts.
milwaukee_data_filtered <- motherboard_data %>%
  filter(City == "Milwaukee, WI") # Filter data specifically for Milwaukee, WI.

# Debugging prints (can be removed after verification):
# These lines help confirm that data is correctly filtered before plotting.
print("Milwaukee Data Head:")
```

```
## [1] "Milwaukee Data Head:"
```

```r
print(head(milwaukee_data_filtered))
```

```
##         Date          City Kia_Hyundai_Thefts All_Thefts Percent_Kia_Hyundai
## 1 2019-12-01 Milwaukee, WI                 22        339          0.06489676
## 2 2020-01-01 Milwaukee, WI                 23        312          0.07371795
## 3 2020-02-01 Milwaukee, WI                 24        290          0.08275862
## 4 2020-03-01 Milwaukee, WI                 27        267          0.10112360
## 5 2020-04-01 Milwaukee, WI                 17        237          0.07172996
## 6 2020-05-01 Milwaukee, WI                 12        230          0.05217391
##   countKiaHyundaiThefts countOtherThefts percentKiaHyundai
## 1                    22              317        0.06489676
## 2                    23              289        0.07371795
## 3                    24              266        0.08275862
## 4                    27              240        0.10112360
## 5                    17              220        0.07172996
## 6                    12              218        0.05217391
```

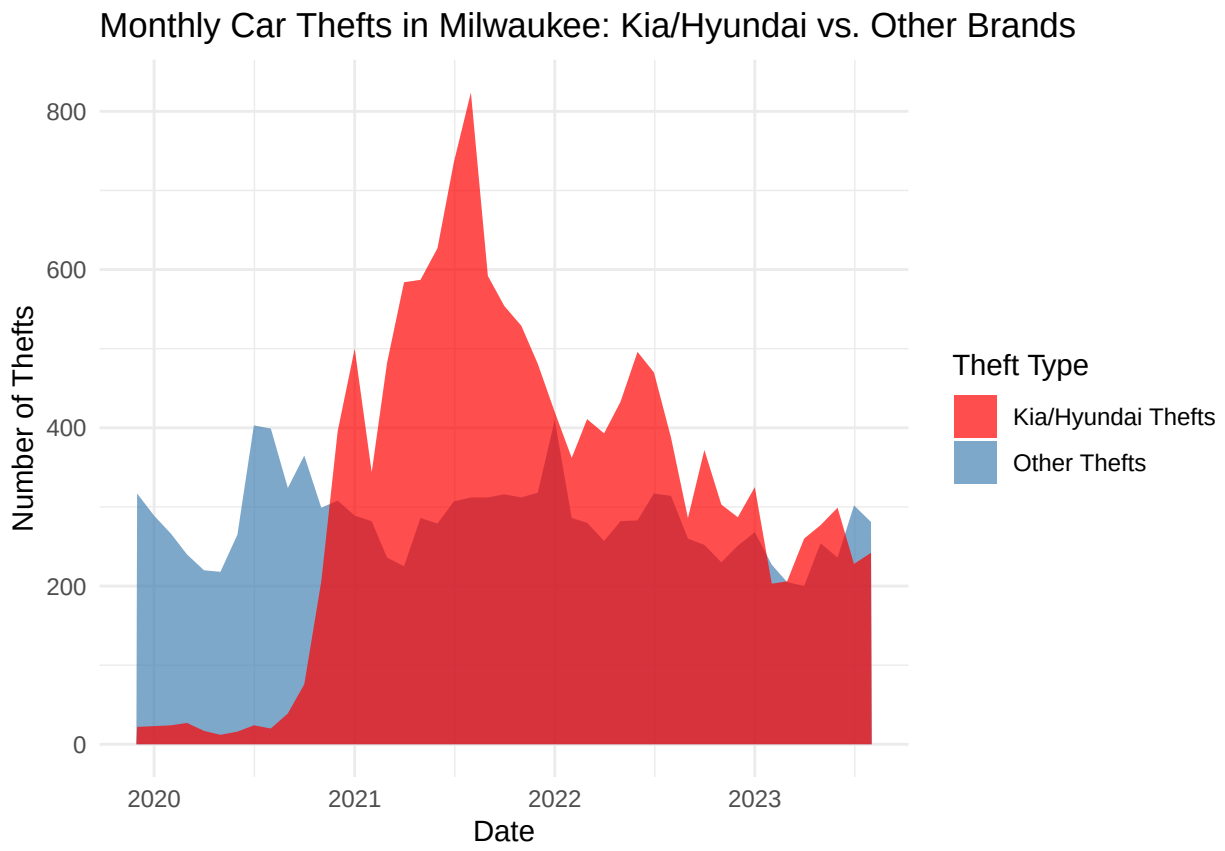
```r
print("Number of rows in Milwaukee Data:")
```

```
## [1] "Number of rows in Milwaukee Data:"
```

```r
print(nrow(milwaukee_data_filtered))
```

```
## [1] 45
```

```r
ggplot(milwaukee_data_filtered, aes(x = Date)) + # Map 'Date' to the x-axis.
  # Add area layer for 'countOtherThefts'.
  geom_area(aes(y = countOtherThefts, fill = "Other Thefts"), alpha = 0.7) +
  # Add area layer for 'countKiaHyundaiThefts'. Alpha for transparency.
  geom_area(aes(y = countKiaHyundaiThefts, fill = "Kia/Hyundai Thefts"), alpha = 0.7) +
  # Manually set colors for consistency and impact (red for Kia/Hyundai, blue for others).
  scale_fill_manual(values = c("Kia/Hyundai Thefts" = "red", "Other Thefts" = "steelblue")) +
  # Define chart labels and title for clarity.
  labs(title = "Monthly Car Thefts in Milwaukee: Kia/Hyundai vs. Other Brands",
       x = "Date",
       y = "Number of Thefts",
       fill = "Theft Type") +
  theme_minimal() # Use a clean, minimalist theme.
```




```
ggsave("milwaukee_thefts_stacked_area_chart.png", width = 10, height = 6)
```

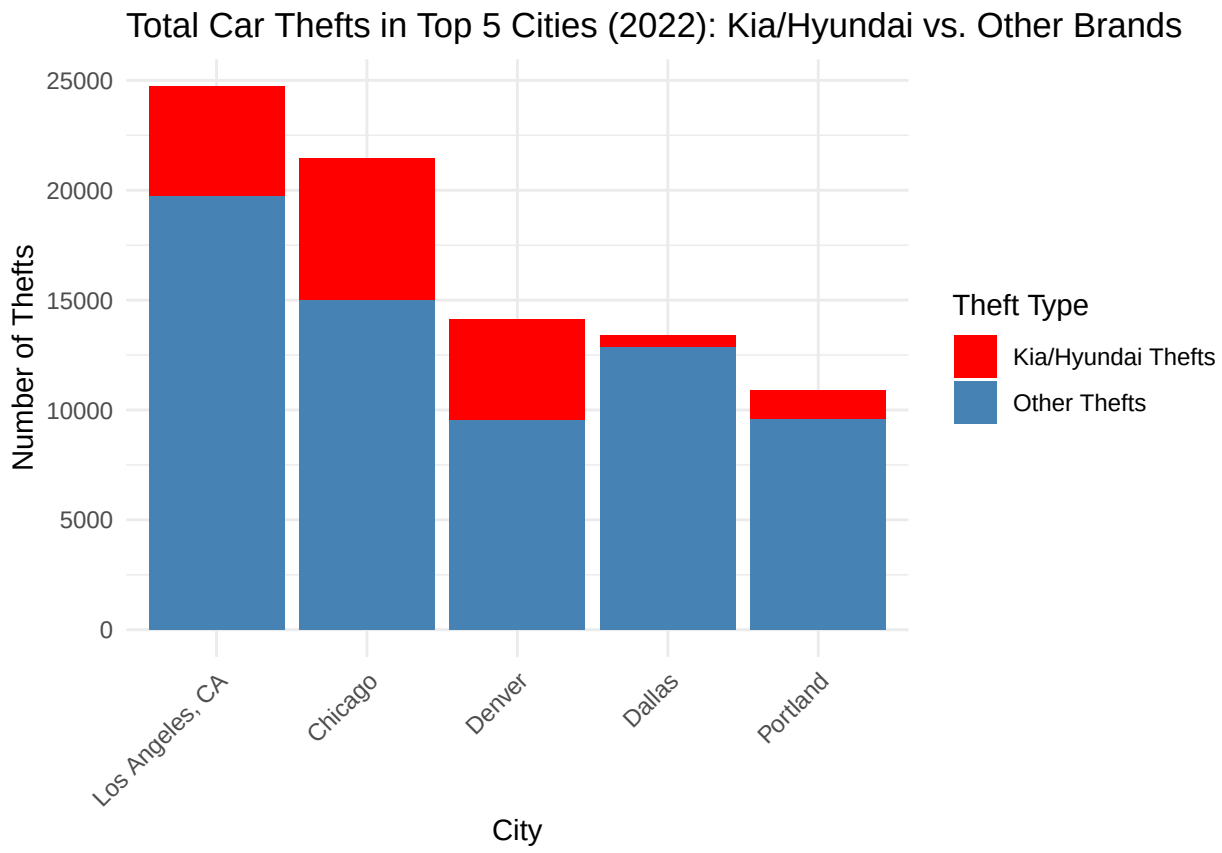
6.2 Visual 2: Stacked Bars - Total Thefts by Type for Top 5 Cities in 2022

```
# Purpose: To compare the total number of Kia/Hyundai thefts versus other thefts in the top 5 most affe
# Justification: Stacked bar charts are excellent for comparing parts of a whole across different categ
# It clearly shows the absolute volume and the proportion of Kia/Hyundai thefts within each city's tota
top_cities_2022 <- motherboard_data %>%
  filter(format(Date, "%Y") == "2022") %>% # Filter data for the year 2022.
  group_by(City) %>% # Group data by city.
  # Summarize total Kia/Hyundai and other thefts for each city.
  summarise(total_kia_hyundai = sum(countKiaHyundaiThefts, na.rm = TRUE),
            total_other = sum(countOtherThefts, na.rm = TRUE)) %>%
  mutate(total_all = total_kia_hyundai + total_other) %>% # Calculate total thefts for ranking.
  arrange(desc(total_all)) %>% # Order cities by total thefts in descending order.
  head(5) # Select the top 5 cities.

# Reshape data from wide to long format for ggplot2's stacked bar chart.
# This creates a 'theft_type' column (Kia/Hyundai or Other) and a 'count' column.
top_cities_2022_long <- top_cities_2022 %>%
  pivot_longer(cols = c(total_kia_hyundai, total_other),
                       names_to = "theft_type",
                       values_to = "count") %>%
```

```r
  # Rename theft types for better readability in the legend.
  mutate(theft_type = ifelse(theft_type == "total_kia_hyundai", "Kia/Hyundai Thefts", "Other Thefts"))

ggplot(top_cities_2022_long, aes(x = reorder(City, -total_all), y = count, fill = theft_type)) +
  geom_bar(stat = "identity", position = "stack") + # Create stacked bars.
  # Use consistent colors.
  scale_fill_manual(values = c("Kia/Hyundai Thefts" = "red", "Other Thefts" = "steelblue")) +
  labs(title = "Total Car Thefts in Top 5 Cities (2022): Kia/Hyundai vs. Other Brands",
       x = "City",
       y = "Number of Thefts",
       fill = "Theft Type") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for readability.
```




```r
ggsave("top5_cities_stacked_bar_chart.png", width = 10, height = 6)
```

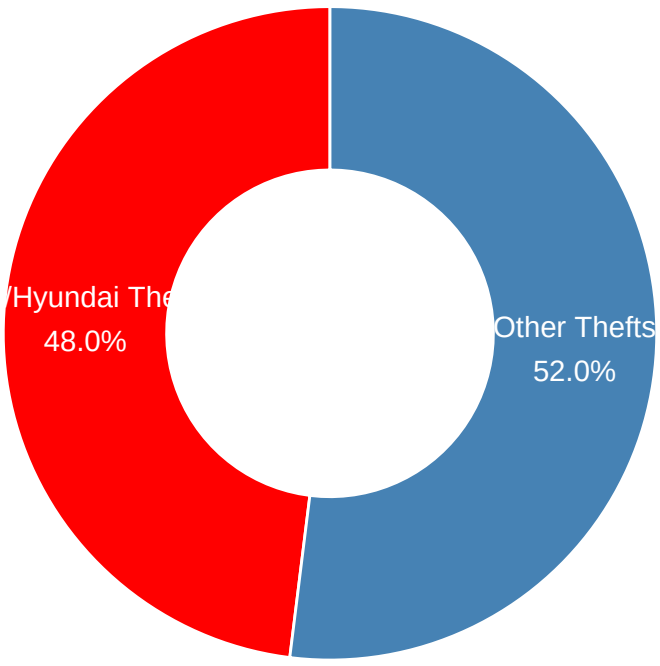
6.3 Visual 3: Donut Chart - Proportion of Kia/Hyundai Thefts in Chicago (Peak Month/Year)

```r
# Purpose: To visually represent the overwhelming proportion of Kia/Hyundai thefts during Chicago's pea
# Justification: A donut chart effectively shows parts of a whole, and by focusing on a single peak mon
# it delivers a powerful message about the severity of the issue in a specific context.
chicago_peak <- motherboard_data %>%
  filter(City == "Chicago") %>% # Filter data for Chicago.
```

```r
  arrange(desc(percentKiaHyundai)) %>% # Find the month with the highest percentage of Kia/Hyundai thef
  head(1) # Select only that peak month.

# Prepare data for the donut chart.
chicago_peak_data <- data.frame(
  theft_type = c("Kia/Hyundai Thefts", "Other Thefts"),
  count = c(chicago_peak$countKiaHyundaiThefts, chicago_peak$countOtherThefts)
) %>%
  mutate(percentage = count / sum(count),
         label = paste0(theft_type, "\n", scales::percent(percentage, accuracy = 0.1))) # Calculate per

ggplot(chicago_peak_data, aes(x = 2, y = percentage, fill = theft_type)) +
  geom_bar(stat = "identity", width = 1, color = "white") + # Create the bar for the donut chart.
  coord_polar(theta = "y") + # Convert to polar coordinates to make it a donut.
  xlim(0.5, 2.5) + # Adjust x-axis limits to create the donut hole.
  # Use consistent colors.
  scale_fill_manual(values = c("Kia/Hyundai Thefts" = "red", "Other Thefts" = "steelblue")) +
  # Add text labels inside the donut segments.
  geom_text(aes(label = label), position = position_stack(vjust = 0.5), color = "white", size = 4) +
  labs(title = paste0("Proportion of Car Thefts in Chicago (Peak: ", format(chicago_peak$Date, "%b %Y")
       fill = "Theft Type") +
  theme_void() + # Remove all non-data ink.
  theme(plot.title = element_text(hjust = 0.5),
        legend.position = "none") # Center title and remove legend as labels are direct.
```



```r
ggsave("chicago_donut_chart.png", width = 8, height = 8)
```

6.4 Visual 4: Treemap - Total Kia/Hyundai Thefts by City (Overall)

```r
# Purpose: To show the overall distribution of Kia/Hyundai thefts across different cities,
# with larger rectangles representing cities with more thefts.
# Justification: Treemaps are effective for displaying hierarchical data or, in this case,
# proportional data where size directly corresponds to a value (total thefts), making
# it easy to identify major contributors at a glance.
total_kia_hyundai_thefts_by_city_overall <- motherboard_data %>%
  group_by(City) %>% # Group by city.
  summarise(total_thefts = sum(countKiaHyundaiThefts, na.rm = TRUE)) %>% # Sum Kia/Hyundai thefts for e
  arrange(desc(total_thefts)) # Order by total thefts.

# Filter for cities with significant thefts to make the treemap readable.
# This threshold helps to focus on the most impactful cities and prevents clutter.
treemap_data <- total_kia_hyundai_thefts_by_city_overall %>%
  filter(total_thefts > 1000)

# Save the treemap as a PNG image.
png("kia_hyundai_thefts_treemap.png", width = 800, height = 600)
treemap(treemap_data,
        index = "City", # Column used for labels within the treemap.
        vSize = "total_thefts", # Column used to determine the size of each rectangle.
        type = "index", # Type of treemap (hierarchical structure).
        title = "Total Kia/Hyundai Thefts by City (Overall 2019-2022)", # Chart title.
        palette = "Reds", # Color palette, using shades of red to indicate intensity.
        fontsize.title = 16,
        fontsize.labels = 12,
        fontcolor.labels = "white", # Label color for readability against red background.
        bg.labels = "transparent", # Transparent background for labels.
        align.labels = list(c("left", "top"), c("right", "bottom")), # Label alignment.
        overlap.labels = 0.5, # Controls how much labels can overlap.
        border.col = "white", # Color of borders between rectangles.
        border.lwds = 2) # Line width of borders.
dev.off() # Close the PNG device.
```

```
## pdf
##   2
```

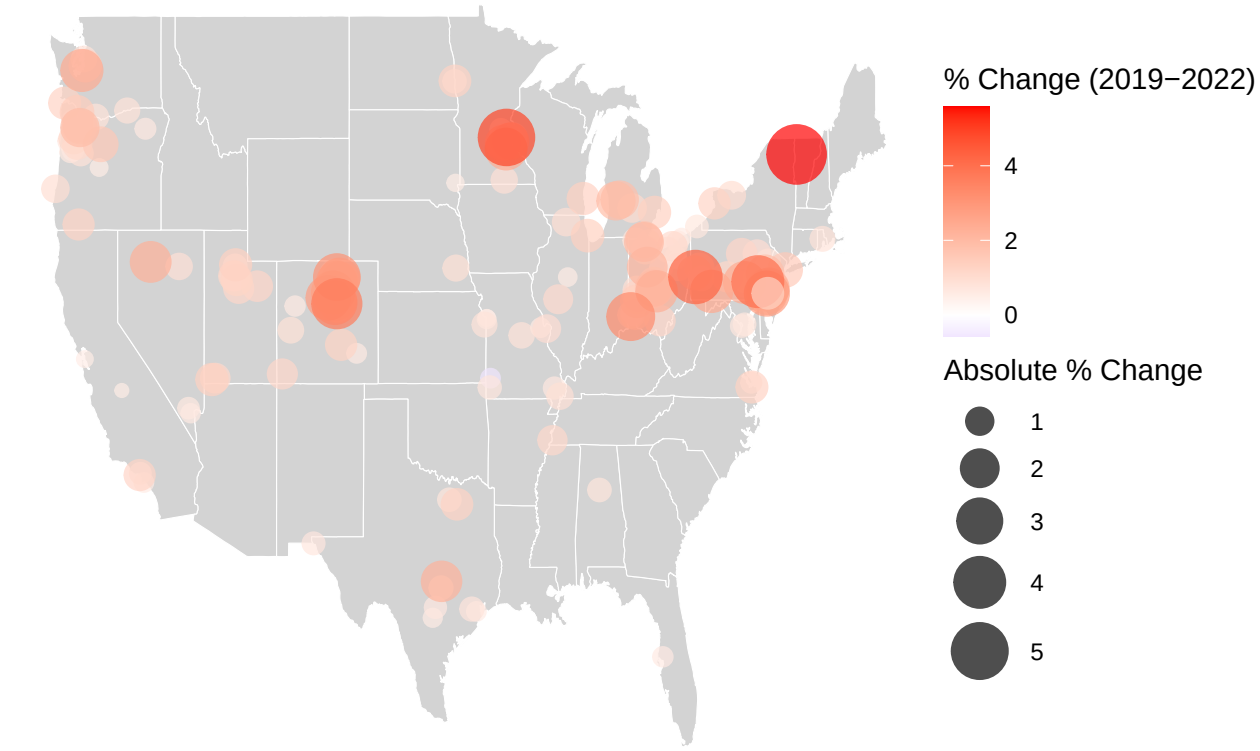
6.5 Visual 5: Geographic Map - Percent Change in Car Thefts (2019-2022)

```r
# Purpose: To visualize the geographical spread and magnitude of changes in car thefts across different
# Justification: A scatter plot on a map effectively shows spatial patterns.
# Color and size aesthetics are used to convey the direction and intensity of change.
# Filter for cities with significant change for better visualization.
# This helps to highlight areas with notable increases or decreases, making the map more informative.
map_data <- car_thefts_map_cleaned %>%
```

```r
  filter(abs(percentChange2019to2022) > 0.5)

# Get US states map data from the 'maps' package.
us_states <- map_data("state")

ggplot() +
  # Draw the base map of US states.
  geom_map(data = us_states, map = us_states,
           aes(x = long, y = lat, map_id = region),
           fill = "lightgray", color = "white", size = 0.2) +
  # Add points for each city, with size and color mapped to theft change.
  geom_point(data = map_data, aes(x = longitude, y = latitude, size = abs(percentChange2019to2022), col
  # Scale point size based on the absolute percentage change.
  scale_size_continuous(range = c(2, 10), name = "Absolute % Change") +
  # Use a diverging color gradient: blue for decrease, red for increase, white for no change.
  scale_color_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0, name = "% Change (2019
  labs(title = "Geographic Distribution of Car Theft % Change (2019-2022)",
       x = "Longitude",
       y = "Latitude") +
  theme_void() + # Remove all non-data ink from the map.
  theme(legend.position = "right", plot.title = element_text(hjust = 0.5)) # Position legend and center
```

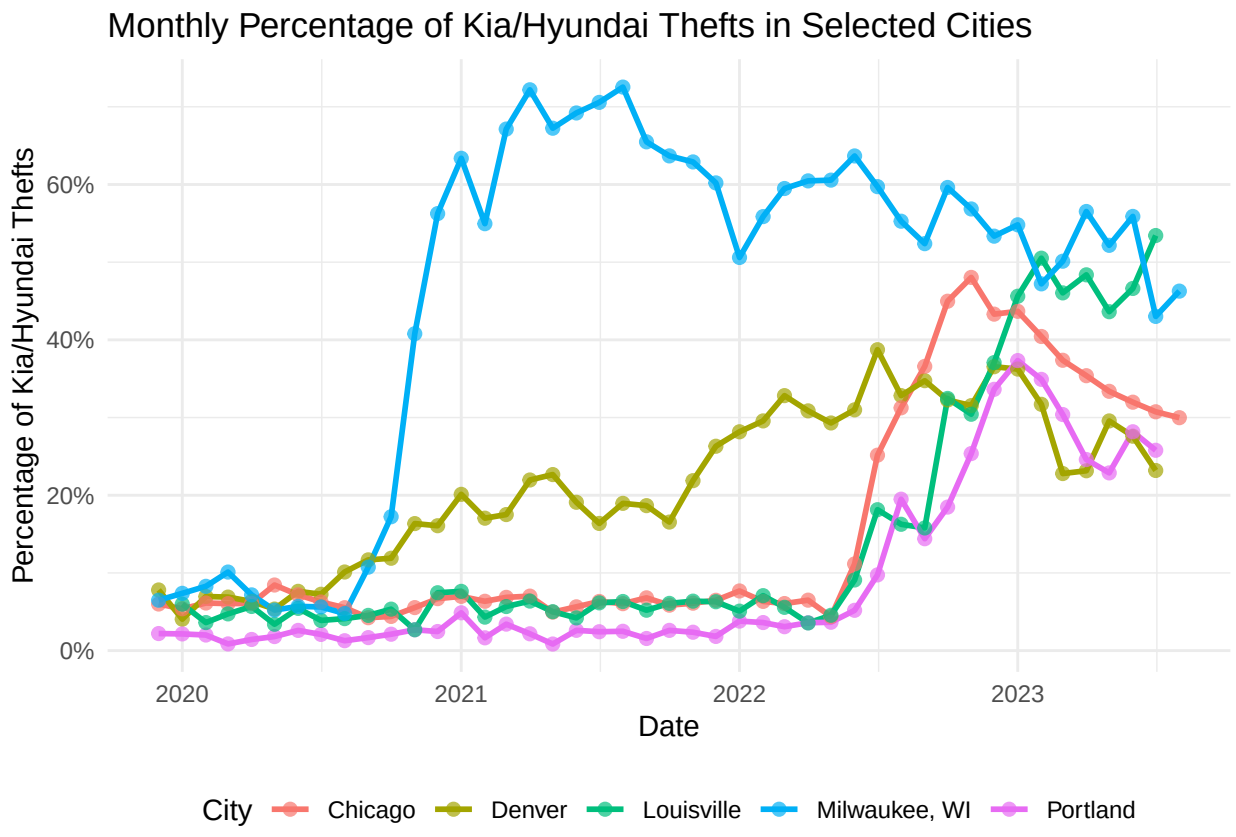



```r
ggsave("car_theft_percent_change_map.png", width = 12, height = 8)
```

6.6 Visual 6: Line Chart - Monthly Percentage of Kia/Hyundai Thefts for Selected Cities

```
# Purpose: To illustrate the trend of Kia/Hyundai thefts as a percentage of total thefts over time
# in several key cities, allowing for direct comparison of how the issue evolved in different locations
# Justification: Line charts are excellent for showing trends over time.
# Plotting percentages helps normalize for varying total theft volumes across cities.
# Select a few cities to show trends. These cities were chosen for their relevance and to show diverse
selected_cities <- c("Milwaukee, WI", "Chicago", "Portland", "Louisville", "Cleveland", "Denver", "Los
filtered_kia_hyundai_thefts_percent <- motherboard_data %>%
  filter(City %in% selected_cities) # Filter data for the selected cities.

ggplot(filtered_kia_hyundai_thefts_percent, aes(x = Date, y = percentKiaHyundai, color = City, group = C
  geom_line(size = 1) + # Draw lines connecting data points over time.
  geom_point(size = 2, alpha = 0.7) + # Add points for individual data observations.
  labs(title = "Monthly Percentage of Kia/Hyundai Thefts in Selected Cities",
       x = "Date",
       y = "Percentage of Kia/Hyundai Thefts",
       color = "City") +
  scale_y_continuous(labels = scales::percent) + # Format y-axis labels as percentages.
  theme_minimal() +
  theme(legend.position = "bottom") # Position legend at the bottom for better use of space.
```




```
ggsave("kia_hyundai_percent_line_chart.png", width = 10, height = 6)
```