

```

install.packages("readxl")
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("readxl")

# Load necessary libraries
library(ggplot2)
library(dplyr)
library(tidyr)
library(readr)
library(readxl)

# Read the CSV file into a data frame
kiaHyundaiThefts <- read.csv("C:/Users/Chitramoy/Desktop/MS-DSC/DSC-640/Week-
6/Exercise/kiaHyundaiThefts.csv")

# Display the first 10 records
head(kiaHyundaiThefts, 10)

# Summarize percentKiaHyundai thefts by city
city_thefts <- kiaHyundaiThefts %>%
  group_by(city) %>%
  summarise(total_percentKiaHyundai = sum(percentKiaHyundai, na.rm = TRUE))

# Create a pie chart
pie(city_thefts$total_percentKiaHyundai,
    labels = city_thefts$city,
    main = "Percent of Kia/Hyundai Thefts by City",
    col = rainbow(length(city_thefts$city)),
    clockwise = TRUE)

# Optional: Add percentage labels to the pie chart
percent_labels <- paste0(round(city_thefts$total_percentKiaHyundai, 1), "%")
pie(city_thefts$total_percentKiaHyundai,
    labels = percent_labels,
    main = "Percent of Kia/Hyundai Thefts by City",
    col = rainbow(length(city_thefts$city)),
    clockwise = TRUE)

# Read the CSV file
file_path <- "C:/Users/Chitramoy/Desktop/MS-DSC/DSC-640/Week-
6/Exercise/KiaHyundaiMilwaukeeData.csv"
kiaHyundaiMilwaukeeData <- read.csv(file_path)

# Summarize data by year
year_data <- kiaHyundaiMilwaukeeData %>%
  group_by(year) %>%
  summarise(total_percentKiaHyundai = sum(percentKiaHyundai, na.rm = TRUE))

# Create a donut chart
ggplot(year_data, aes(x = 2, y = total_percentKiaHyundai, fill = factor(year))) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar(theta = "y") +
  xlim(0.5, 2.5) + # This adjusts the inner radius to create the donut effect
  theme_void() +
  theme(legend.position = "right") +
  labs(title = "Year-wise Percent Kia/Hyundai Thefts in Milwaukee", fill = "Year") +
  annotate("text", x = 0, y = 0, label = "Kia/Hyundai Thefts", size = 6)

# Read the data
data_path <- "C:/Users/Chitramoy/Desktop/MS-DSC/DSC-640/Week-

```

```

6/Exercise/KiaHyundaiMilwaukeeData.csv"
kia_hyundai_data <- read.csv(data_path)

# Summarize data by year
yearly_data <- kia_hyundai_data %>%
  group_by(year) %>%
  summarise(
    countKiaHyundaiThefts = sum(countKiaHyundaiThefts, na.rm = TRUE),
    countOtherThefts = sum(countOtherThefts, na.rm = TRUE)
  )

# Melt the data for ggplot
library(reshape2)
melted_data <- melt(yearly_data, id.vars = "year")

# Create stacked bar chart
ggplot(melted_data, aes(x = factor(year), y = value, fill = variable)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Kia and Hyundai Thefts vs. Other Thefts by Year",
    x = "Year",
    y = "Number of Thefts",
    fill = "Theft Type"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Read the data
data_path <- "C:/Users/Chitramoy/Desktop/MS-DSC/DSC-640/Week-6/Exercise/Motherboard VICE
News Kia Hyundai Theft Data.xlsx"
kia_hyundai_data <- read_excel(data_path)

# Convert KiaHyundais and All to numeric, handling any non-numeric entries
kia_hyundai_data$KiaHyundais <- as.numeric(kia_hyundai_data$KiaHyundais)
kia_hyundai_data$All <- as.numeric(kia_hyundai_data$All)

# Summarize the data by date (month-year)
kia_hyundai_data_summary <- kia_hyundai_data %>%
  group_by(Month) %>%
  summarise(
    KiaHyundais = sum(KiaHyundais, na.rm = TRUE),
    All = sum(All, na.rm = TRUE)
  )

# Melt the data for ggplot
kia_hyundai_data_long <- kia_hyundai_data_summary %>%
  pivot_longer(cols = c(KiaHyundais, All), names_to = "TheftType", values_to = "Count")

# Create stacked bar chart
ggplot(kia_hyundai_data_long, aes(x = Month, y = Count, fill = TheftType)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Kia/Hyundais vs All Thefts Over Time",
    x = "Date (Month/Year)",
    y = "Number of Thefts",
    fill = "Theft Type"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Remove rows with N/A values in any of the key columns
kia_hyundai_data <- kia_hyundai_data %>%
  filter(!is.na(Month), !is.na(KiaHyundais))

```

```

# Convert KiaHyundais to numeric, handling any non-numeric entries
kia_hyundai_data$KiaHyundais <- as.numeric(kia_hyundai_data$KiaHyundais)

# Summarize the data by date (month-year)
kia_hyundai_data_summary <- kia_hyundai_data %>%
  group_by(Month) %>%
  summarise(KiaHyundais = sum(KiaHyundais, na.rm = TRUE))

# Print summary data for inspection (optional)
print(kia_hyundai_data_summary)

# Create area chart
ggplot(kia_hyundai_data_summary, aes(x = Month, y = KiaHyundais)) +
  geom_area(fill = "blue", alpha = 0.6) +
  labs(
    title = "KiaHyundais Thefts Over Time",
    x = "Date (Month/Year)",
    y = "Number of KiaHyundai Thefts"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Load the dataset from the file path
file_path <- "C:/Users/Chitramoy/Desktop/MS-DSC/DSC-640/Week-6/Exercise/carTheftsMap.csv"
car_thefts_data <- read_csv(file_path)

# View the first few rows of the data
head(car_thefts_data)

# Check the structure of the data to identify categorical columns
str(car_thefts_data)

# Convert geo_name to a factor (categorical variable)
car_thefts_data$geo_name <- as.factor(car_thefts_data$geo_name)

# Filter top 20 geo_name by countCarThefts2022
top_20_geo <- car_thefts_data %>%
  arrange(desc(countCarThefts2022)) %>%
  slice_head(n = 20)

# --- Create Stacked Bar Plot for percentChange2019to2022 and geo_name ---

# Reshape data for stacked bar chart using percentChange2019to2022
stacked_data <- top_20_geo %>%
  select(geo_name, percentChange2019to2022, countCarThefts2022)

# Plot the stacked bar chart
ggplot(stacked_data, aes(x = reorder(geo_name, -countCarThefts2022), y =
percentChange2019to2022, fill = geo_name)) +
  geom_bar(stat = "identity") +
  labs(title = "Percent Change in Car Thefts (2019 to 2022) for Top 20 Geo Locations",
    x = "Geo Name",
    y = "Percent Change (2019 to 2022)",
    fill = "Geo Name") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3") # Optional: change color palette

```