A2 Assignment - Refugee Analysis

MBAN - Group 7

Team Members:

Abi Joshua GEORG / Eri Yoshimoto / Hakeem GARCIA
Nattida TAVAROJN / Neha NAGABHUSHAN / Weikang YANG

Introduction

Refugee migration has been a significant global issue, with various countries experiencing fluctuations in the number of people seeking asylum. This report analyzes refugee data from multiple countries over a span of years to identify trends, patterns, and key insights. The study involves data cleaning, transformation, visualization, and statistical analysis to gain a deeper understanding of global refugee movements.

Data Preparation and Cleaning

The dataset initially contained raw refugee statistics, which required significant cleaning before analysis.

```
library(tidyverse)
library(readr)
# read data from database
raw_df <- read_csv("data/A2_refugee_status.csv", col_types = cols(.default = "c"))
```

Handling Missing and Inconsistent Values

The dataset contained placeholders for missing values, such as “D,” “X,” and “-”, which were converted to “0.” Numeric values were reformatted by removing commas and converting them into numerical data types.

```
# Set NULL value("D", "X", "-")as "0"
raw_df[raw_df == "D" | raw_df == "X" | raw_df == "-"] <- "0"
# set the value column as numbers value and delete comma and transfer to numeric value
raw_df[ , -1] <- lapply(raw_df[ , -1], function(x) as.numeric(gsub(",", "", x)))
# set cleaned data frame as df and use df in later operation
# combine "Congo, Democratic Republic" and "Congo, Republic"
congo_rows <- raw_df %>% filter(`Continent/Country of Nationality` %in%
                                  c("Congo, Democratic Republic", "Congo, Republic"))
congo_sum <- colSums(congo_rows[, -1], na.rm = TRUE)

#  remove the original duplicated Congo data
raw_df <- raw_df %>% filter(!`Continent/Country of Nationality` %in%
                              c("Congo, Democratic Republic", "Congo, Republic"))

# add the new combined Congo data to the data frame
```

```
raw_df <- raw_df %>%
  add_row(`Continent/Country of Nationality` = "Congo", !!!as.list(congo_sum))
```

Standardizing Country Names

Countries with alternative naming conventions, such as “China, People’s Republic” and “Korea, North,” were renamed to “China” and “North Korea” for consistency.

```
# Replace the Countries' name with formal format
country_df <- raw_df %>%
  mutate(`Continent/Country of Nationality` = case_when(
    `Continent/Country of Nationality` == "China, People's Republic" ~ "China",
    `Continent/Country of Nationality` == "Korea, North" ~ "North Korea",
    TRUE ~ `Continent/Country of Nationality`
  ))
df <- country_df
```

```
# create the new dataframe 'contry_df',and remove the un-country row from the dataframe
# defined the name list that will be removed from the dataframe
non_countries <- c("Africa", "Asia", "Europe", "North America",
                   "Oceania", "South America", "Unknown", "Other", "Total")

# use filter to get the row that will be delete
removed_countries <- df %>%
  filter(`Continent/Country of Nationality` %in% non_countries)

# create a new dataframe that only containt the 'countries'
country_df <- df %>%
  filter(!`Continent/Country of Nationality` %in% non_countries)

# print the row that be deleted to make sure all of them are 'non-countries'
print("delete rows with non-contry:")
```

```
## [1] "delete rows with non-contry:"
```

```
print(removed_countries$`Continent/Country of Nationality`)
```

```
##  [1] "Africa"        "Asia"          "Europe"        "North America"
##  [5] "Oceania"       "South America" "Unknown"       "Other"
##  [9] "Unknown"       "Total"
```

```
# check the new dataframe
head(country_df)
```

```
## # A tibble: 6 x 11
##   Continent/Country of~1 `2006` `2007` `2008` `2009` `2010` `2011` `2012` `2013`
##   <chr>                   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Afghanistan               651    441    576    349    515    428    481    661
## 2 Angola                    13      4      0      8      0      0      0      6
## 3 Armenia                   87     29      9      4      0     15      8      3
```

```
## 4 Azerbaijan                 77     78     30     38     18     16     10      3
## 5 Belarus                   350    219    111    146    103     66     83     10
## 6 Bhutan                      3      0   5320  13452  12363  14999  15070   9134
## # i abbreviated name: 1: 'Continent/Country of Nationality'
## # i 2 more variables: '2014' <dbl>, '2015' <dbl>
```

Trends in Refugee Migration

1. Overall Refugee Trends by Country

A line chart visualizes the refugee numbers from each country over time.

```
# Calculate the number of refugees for each state by year
continent_df <- df %>%
  filter(`Continent/Country of Nationality` %in% c("Africa", "Asia", "Europe",
                                                   "North America", "Oceania", "South America")) %>%
  pivot_longer(cols = -`Continent/Country of Nationality`,
               names_to = "Year", values_to = "Value") %>%
  group_by(`Continent/Country of Nationality`, Year) %>%
  summarise(Total_Refugees = sum(Value, na.rm = TRUE))

# make sure the 'year' value is in Number data format
continent_df$Year <- as.numeric(continent_df$Year)
```

Refugee Trends Across Presidential Terms (2006–2015)

The analysis of refugee data from 2006 to 2015 provides valuable insights into global migration patterns and highlights the influence of leadership periods on refugee numbers. By combining stacked bar charts with trend lines and presidential term overlays, the visualization effectively tells a compelling story of refugee distribution across continents during the George W. Bush and Barack Obama administrations.

Key Insights: Presidential Term Influence:

The data is segmented into two distinct presidential periods:

- George W. Bush (2006–2008): This period shows a steady increase in refugee numbers, indicating global conflicts or crises that led to heightened migration.
- Barack Obama (2009–2015): The refugee numbers initially declined but later stabilized at higher levels, reflecting the impact of sustained geopolitical challenges during this time.

Stacked Bar Chart for Continental Trends:

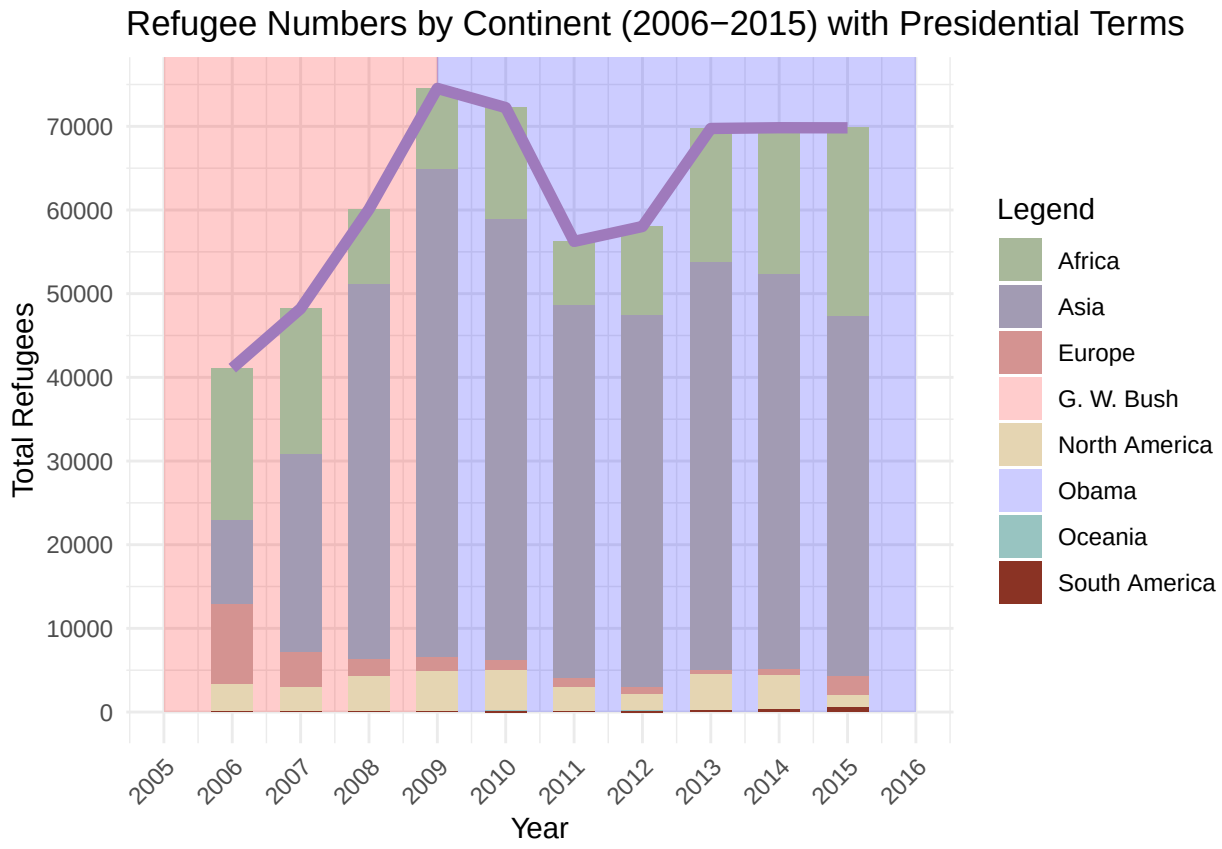
The stacked bar chart provides a breakdown of refugee numbers by continent. Notably:

- Asia and Africa consistently contributed the highest numbers of refugees, driven by ongoing conflicts and socio-political instability.
- Europe and the Americas saw comparatively smaller contributions, reflecting differences in regional refugee patterns.

```r
library(ggplot2)
library(dplyr)

presidents <- data.frame(
  President = c("G. W. Bush", "Obama"),
  Start_Year = c(2005, 2009),
  End_Year = c(2009, 2016)
)

continent_df <- continent_df %>%
  mutate(
    Year = as.numeric(Year),
    Total_Refugees = as.numeric(Total_Refugees)
  )


yearly_refugees <- continent_df %>%
  group_by(Year) %>%
  summarise(Total_Refugees = sum(Total_Refugees, na.rm = TRUE))


ggplot() +

  geom_rect(data = presidents, aes(xmin = Start_Year, xmax = End_Year, ymin = 0, ymax = Inf, fill = Pres

  geom_bar(data = continent_df, aes(x = Year, y = Total_Refugees, fill = `Continent/Country of Nationali
           stat = "identity", position = "stack", width = 0.6) +
  geom_line(data = yearly_refugees, aes(x = Year, y = Total_Refugees, group = 1),
            color = "#9f7abc", size = 2) +
  scale_fill_manual(
    name = "Legend",
    values = c(
      "G. W. Bush" = "red",  # color the president
      "Obama" = "blue",
      "Africa" = "#a8b89a",
      "Asia" = "#a29bb3",
      "Europe" = "#d49391",
      "North America" = "#e5d5b2",
      "Oceania" = "#98c4c1",
      "South America" = "#8A3324"
    )
  ) +

  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(limits = c(2005, 2016), breaks = 2005:2016) +

  theme_minimal() +
  labs(
    title = "Refugee Numbers by Continent (2006-2015) with Presidential Terms",
    x = "Year",
    y = "Total Refugees"
  ) +
  theme(
```

```
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "right"
  )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

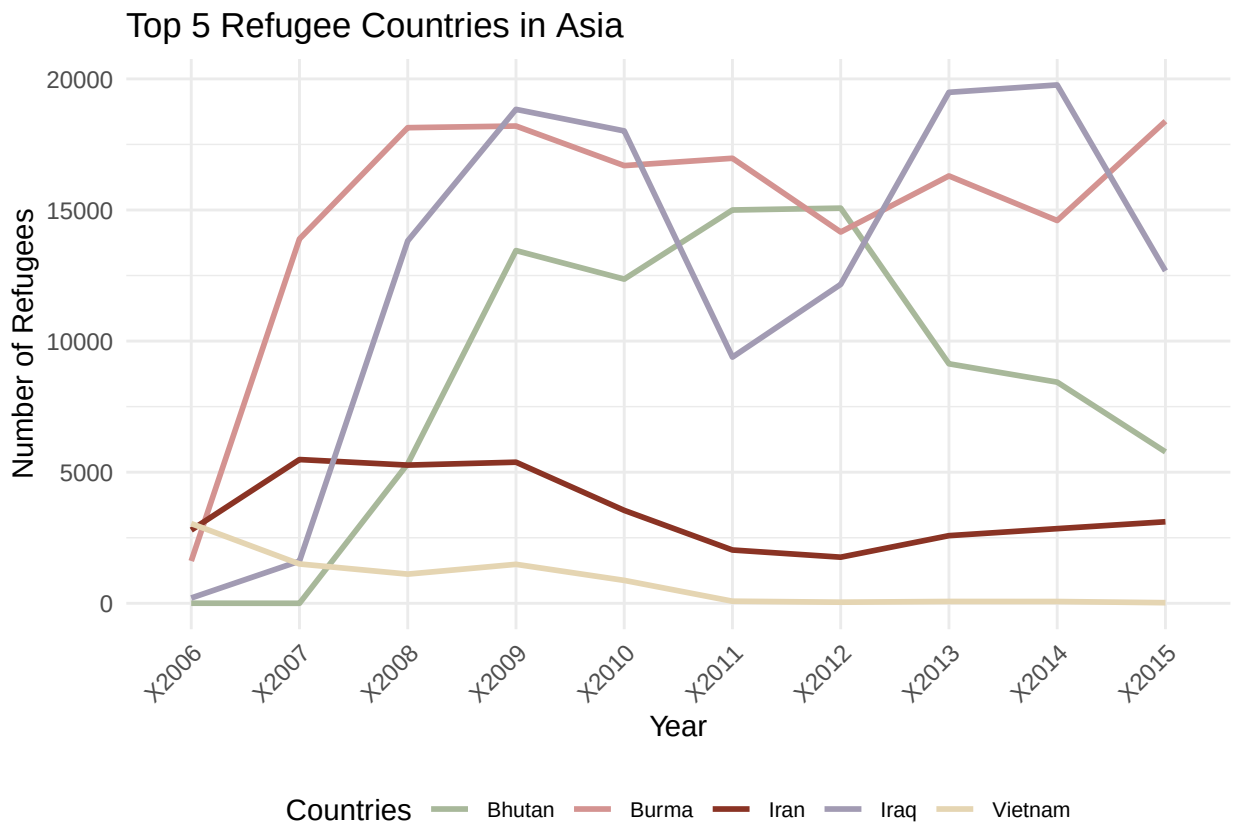



```
# Load required libraries
library(ggplot2)
library(dplyr)
library(tidyr)
library(patchwork)
library(countrycode)  # For country to continent mapping
theme_set(
  theme_minimal() +  # Base theme
    theme(
      plot.background = element_rect(fill = "#e5d5b2", color = NA),  # Set background color
      panel.background = element_rect(fill = "#e5d5b2", color = NA),  # Set panel background color
      legend.background = element_rect(fill = "#e5d5b2", color = NA)  # Set legend background color
    )
)
# Load the dataset
```

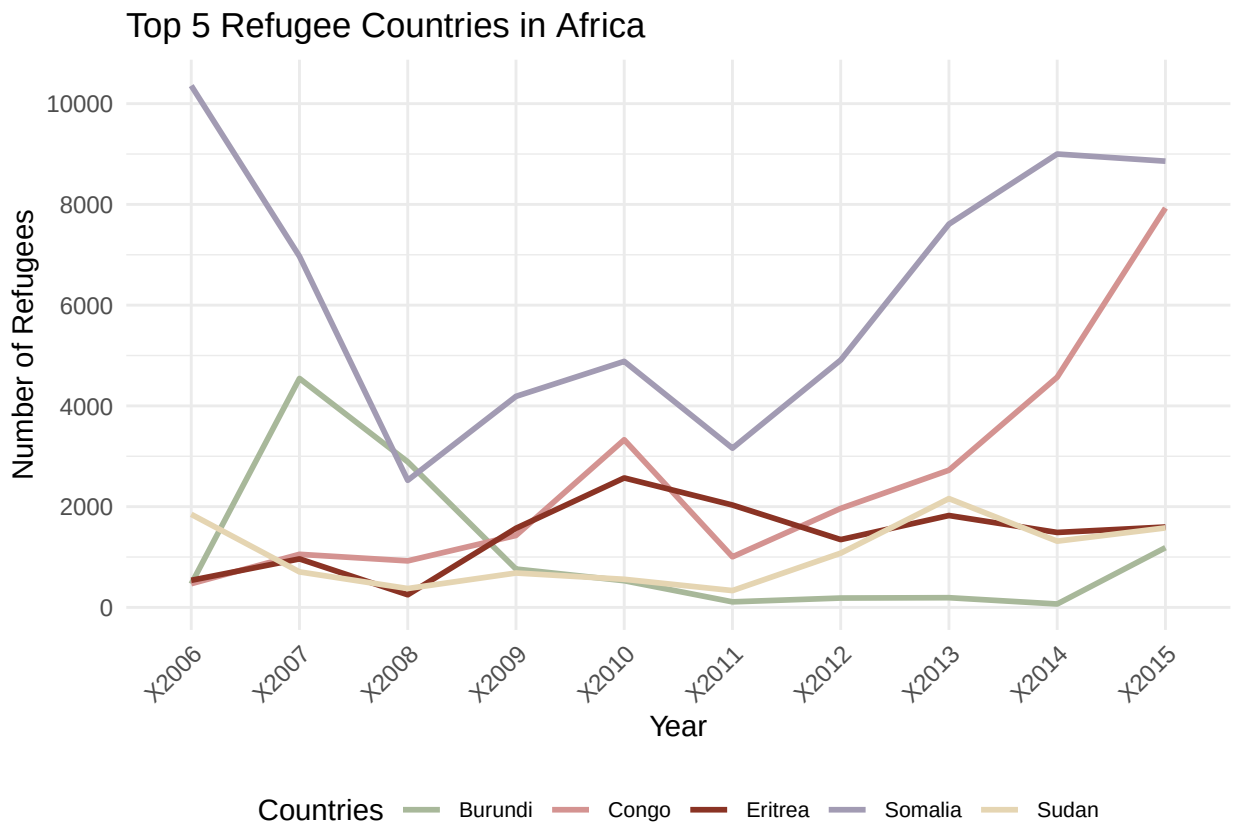
```r
data <- read.csv("country_data.csv")

# Convert the dataset to long format
data_long <- data %>%
  pivot_longer(cols = -Country,
               names_to = "Year",
               values_to = "Value")

# Step 1: Map countries to continents using countrycode package
data_long <- data_long %>%
  mutate(
    Continent = countrycode(Country, origin = "country.name", destination = "continent")
  )

# Step 2: Manually classify certain countries as North America
data_long <- data_long %>%
  mutate(
    Continent = ifelse(Country %in% c("Haiti", "Cuba", "Honduras"), "North America", Continent)
  )

# Step 3: Exclude countries from Oceania and South America
data_long <- data_long %>%
  filter(!Continent %in% c("Oceania", "South America"))

# Step 4: Define a function to get top 5 countries by total refugees for each continent
top_5_per_continent <- function(data, continent, rank) {
  data %>%
    filter(Continent == continent) %>%
    group_by(Country) %>%
    summarise(Total_Refugees = sum(Value, na.rm = TRUE)) %>%
    arrange(desc(Total_Refugees)) %>%
    slice_head(n = rank) %>%
    inner_join(data, by = "Country")
}

# Step 5: Get the top 5 countries for each continent
asia_data <- top_5_per_continent(data_long, "Asia", 5)
africa_data <- top_5_per_continent(data_long, "Africa", 5)
north_america_data <- top_5_per_continent(data_long, "North America", 3)  # Now includes manually class
europe_data <- top_5_per_continent(data_long, "Europe", 5)

# Step 6: Define a function to create line plots for each continent
plot_continent <- function(data, title) {
  ggplot(data, aes(x = Year, y = Value, color = Country, group = Country)) +
    geom_line(size = 1) +
    theme_minimal() +
    labs(title = title, x = "Year", y = "Number of Refugees", color = "Countries") +
    theme(
      legend.position = "bottom",
      legend.text = element_text(size = 8)
    ) +
    guides(color = guide_legend(ncol = 2)) +
    scale_y_continuous(
```

```
      limits = c(0, max(data$Value, na.rm = TRUE) * 1.2),  # Extend Y-axis by 20%
      breaks = scales::pretty_breaks(n = 10)  # Add more breaks for readability
    )
}


# Step 7: Create individual line plots for each continent
plot_asia <- plot_continent(asia_data, "Top 5 Refugee Countries in Asia")
plot_africa <- plot_continent(africa_data, "Top 5 Refugee Countries in Africa")
plot_north_america <- plot_continent(north_america_data, "Top 5 Refugee Countries in North America")
plot_europe <- plot_continent(europe_data, "Top 5 Refugee Countries in Europe")

# Define custom colors for countries
custom_colors <- c(
  # Asia
  "Bhutan" = "#a8b89a", "Iraq" = "#a29bb3", "Burma" = "#d49391",
  "Vietnam" = "#e5d5b2", "Iran" = "#8A3324",

  # Africa
  "Burundi" = "#a8b89a", "Somalia" = "#a29bb3", "Congo" = "#d49391",
  "Sudan" = "#e5d5b2", "Eritrea" = "#8A3324",

  # North America
  "Cuba" = "#a8b89a", "Honduras" = "#a29bb3", "Haiti" = "#d49391",

  # Europe
  "Belarus" = "#a8b89a", "Russia" = "#a29bb3", "Latvia" = "#d49391",
  "Ukraine" = "#e5d5b2", "Moldova" = "#8A3324"
)

# Adjust plot dimensions for a single column layout
options(repr.plot.width = 15, repr.plot.height = 30)

# Modify the plot_continent function to include custom line colors
plot_continent <- function(data, title) {
  ggplot(data, aes(x = Year, y = Value, color = Country, group = Country)) +
    geom_line(size = 1) +
    theme_minimal() +
    labs(title = title, x = "Year", y = "Number of Refugees", color = "Countries") +
    theme(
      legend.position = "bottom",
      legend.text = element_text(size = 8),
      axis.text.x = element_text(angle = 45, hjust = 1)
    ) +
    scale_y_continuous(
      breaks = scales::pretty_breaks(n = 5)  # Adjust Y-axis breaks dynamically
    ) +
    scale_color_manual(values = custom_colors)  # Apply custom colors
}

# Create updated plots for each continent
plot_asia <- plot_continent(asia_data, "Top 5 Refugee Countries in Asia")
plot_africa <- plot_continent(africa_data, "Top 5 Refugee Countries in Africa")
```

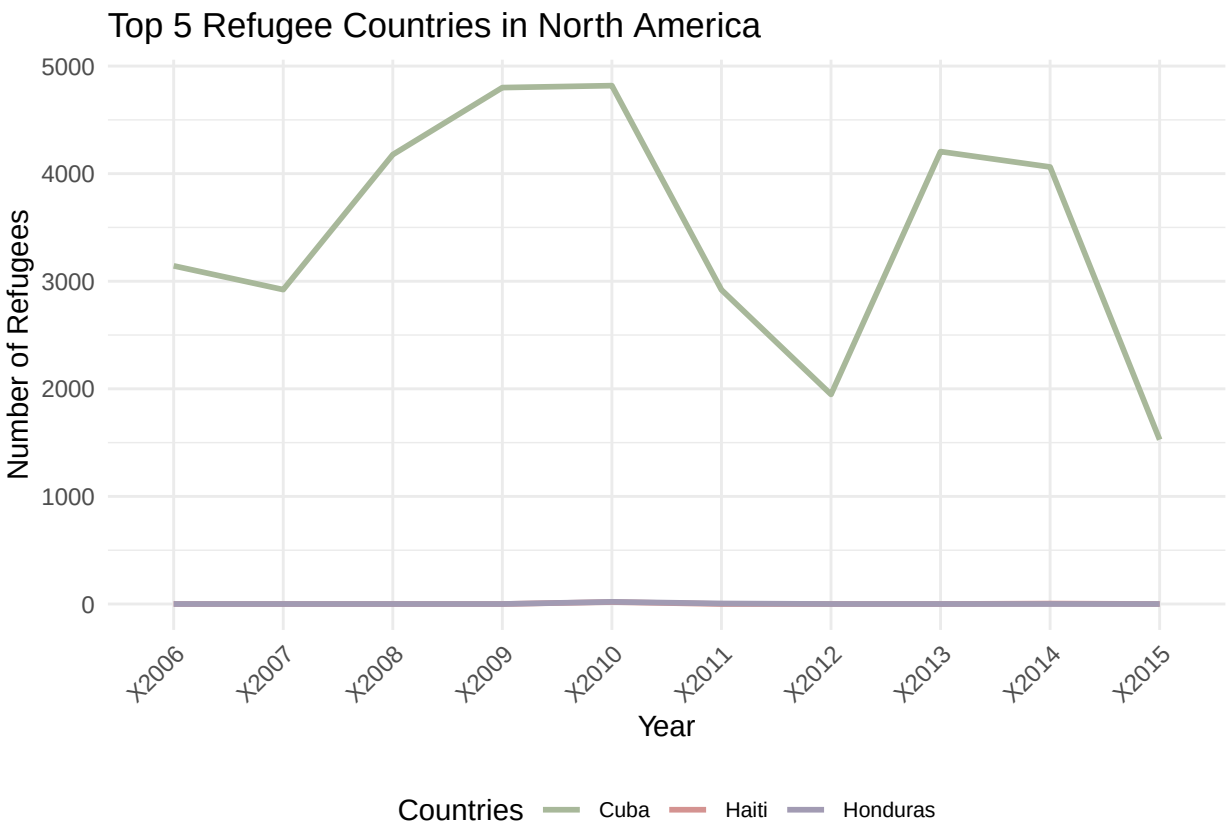
```
plot_north_america <- plot_continent(north_america_data, "Top 5 Refugee Countries in North America")
plot_europe <- plot_continent(europe_data, "Top 5 Refugee Countries in Europe")

# Combine plots with a single column layout
plot_asia
```



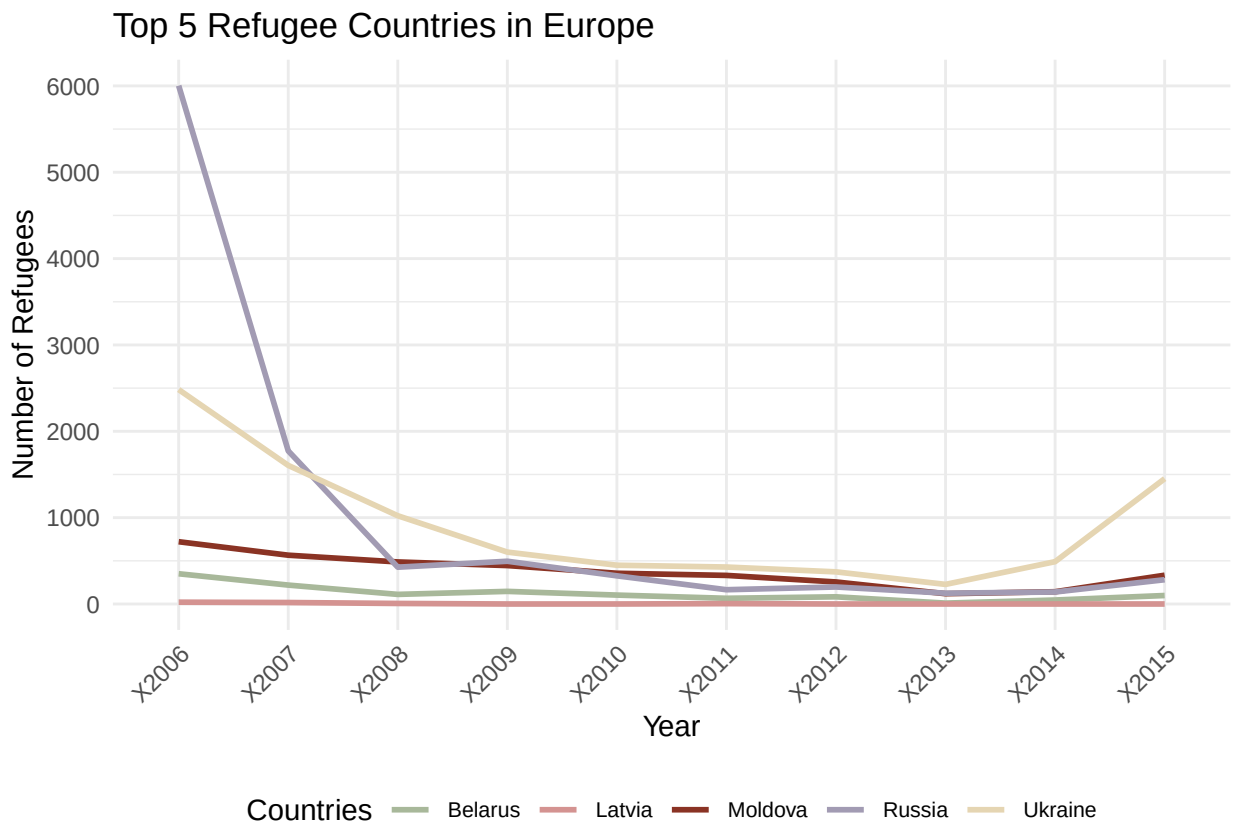

```
plot_africa
```




```
plot_north_america
```


Top 5 Refugee Countries in North America
Number of Refugees
5000
4000
3000
2000
1000
0
X2006
X2007
X2008
X2009
X2010
X2011
X2012
X2013
X2014
X2015
Year
Countries
Cuba
Haiti
Honduras

```
plot_europe
```

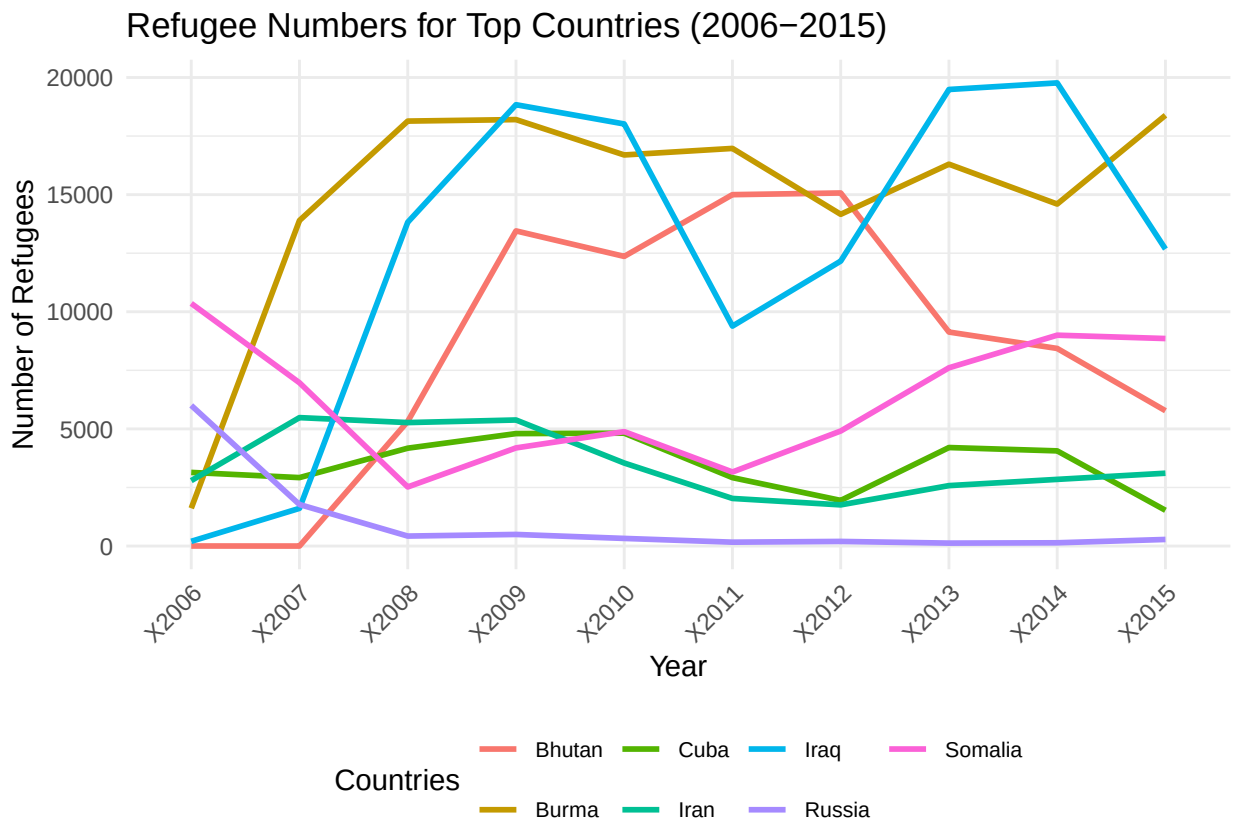



```r
# Load required libraries
library(dplyr)
library(tidyr)

# Load the dataset
data <- read.csv("country_data.csv")

# Convert the dataset to long format
data_long <- data %>%
  pivot_longer(cols = -Country, names_to = "Year", values_to = "Refugees")
# Step 1: Find the top 10 countries by refugees for each year
top10_per_year <- data_long %>%
  group_by(Year) %>%
  slice_max(order_by = Refugees, n = 3) %>%
  ungroup()

# Step 2: Extract all years' data for these top countries
top_countries <- unique(top10_per_year$Country)  # Get unique top countries
all_data_top_countries <- data_long %>%
  filter(Country %in% top_countries)  # Filter all years for top countries

# View the data
print(all_data_top_countries)
```

```
## # A tibble: 70 x 3
##    Country Year  Refugees
```

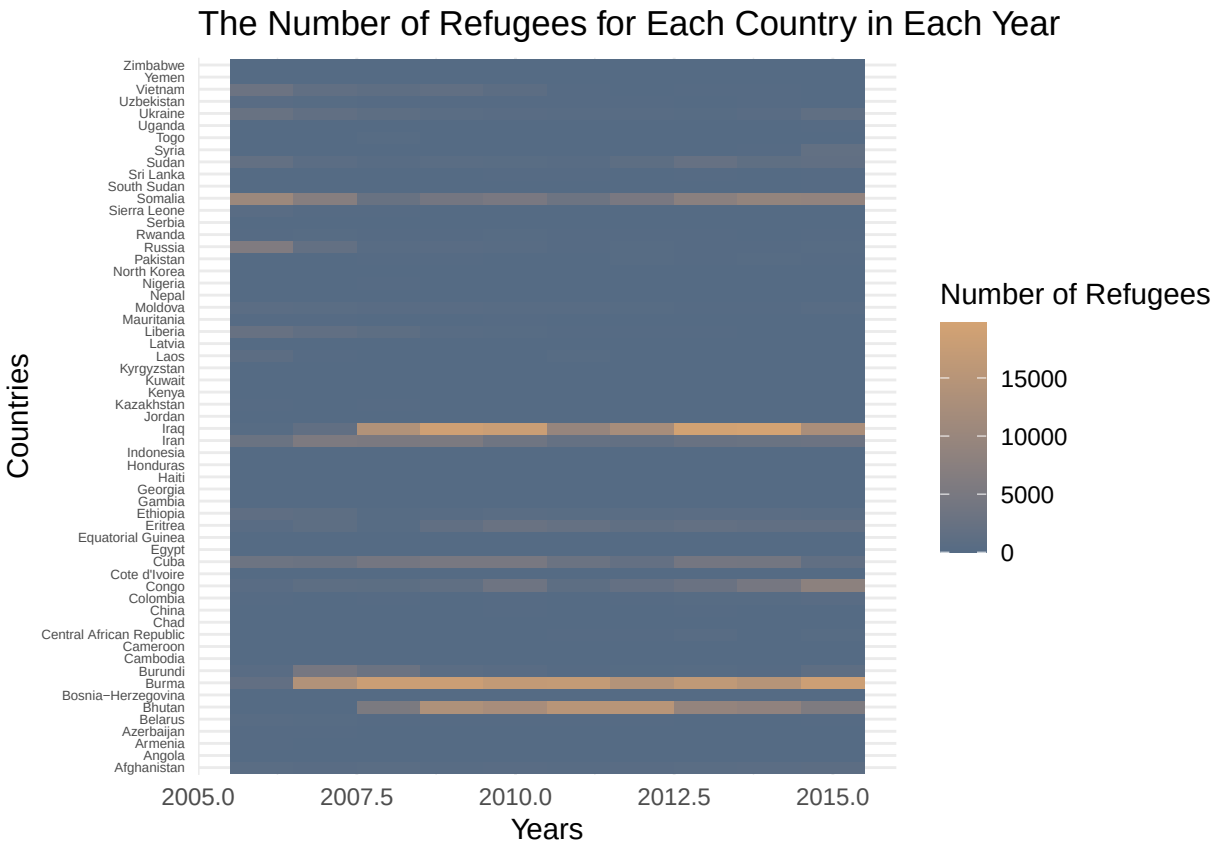
```
##    <chr>   <chr>    <int>
##  1 Bhutan  X2006        3
##  2 Bhutan  X2007        0
##  3 Bhutan  X2008     5320
##  4 Bhutan  X2009    13452
##  5 Bhutan  X2010    12363
##  6 Bhutan  X2011    14999
##  7 Bhutan  X2012    15070
##  8 Bhutan  X2013     9134
##  9 Bhutan  X2014     8434
## 10 Bhutan  X2015     5775
## # i 60 more rows
```

```r
# Save to a new CSV file if needed
write.csv(all_data_top_countries, "all_top_countries_refugees.csv", row.names = FALSE)


# Load required libraries
library(ggplot2)
library(dplyr)

# Ensure data is loaded into all_data_top_countries
# If not already loaded, reload the dataset
all_data_top_countries <- read.csv("all_top_countries_refugees.csv")

# Create a line plot for all top countries over the years
ggplot(all_data_top_countries, aes(x = Year, y = Refugees, color = Country, group = Country)) +
  geom_line(size = 1) +  # Add lines for each country
  theme_minimal() +  # Use a minimal theme
  labs(
    title = "Refugee Numbers for Top Countries (2006-2015)",
    x = "Year",
    y = "Number of Refugees",
    color = "Countries"
  ) +
  theme(
    legend.position = "bottom",  # Move legend to the bottom
    legend.text = element_text(size = 8),  # Adjust legend text size
    axis.text.x = element_text(angle = 45, hjust = 1)  # Rotate x-axis labels
  )
```



3. Heatmap of Refugee Numbers

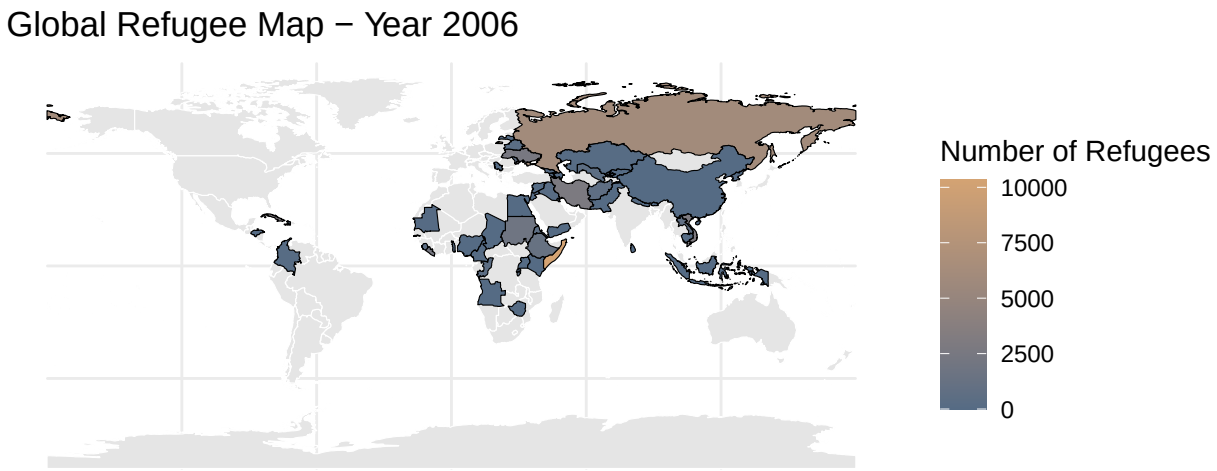
```
country_long <- country_df %>%
  pivot_longer(cols = -`Continent/Country of Nationality`,
               names_to = "Year", values_to = "Value") %>%
  mutate(
    Year = as.numeric(Year),
    Value = as.numeric(Value)
  )

ggplot(country_long, aes(x = as.numeric(Year), y = `Continent/Country of Nationality`, fill = Value)) +
  geom_tile() +
  scale_fill_gradient(low = "#556b84", high = "#d4a373") +
  theme_minimal() +
  labs(title = "The Number of Refugees for Each Country in Each Year",
       x = "Years",
       y = "Countries",
       fill = "Number of Refugees") +
  theme(legend.position = "right",
  axis.text.y = element_text(size = 5))
```



The World Map with Refugees number

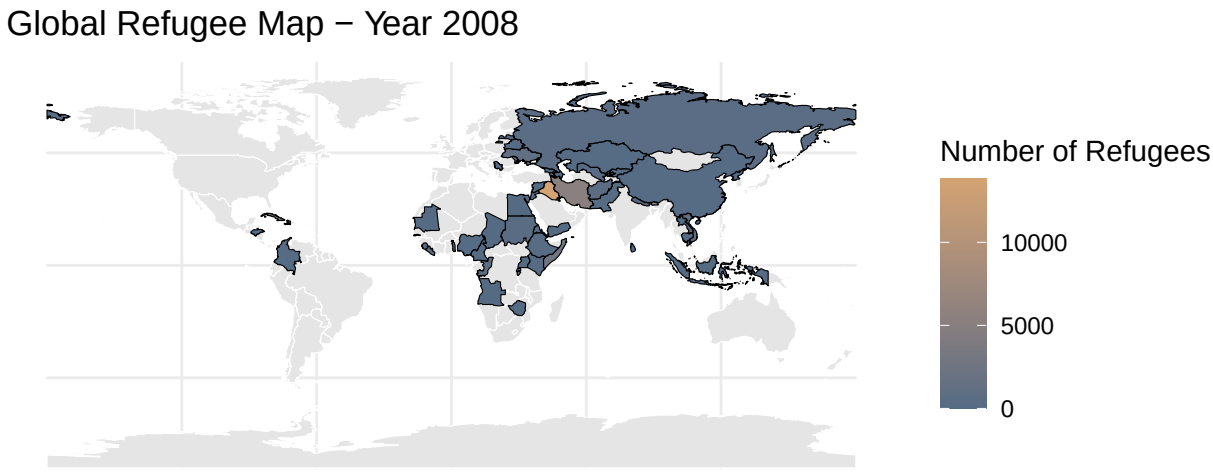
```
library(gganimate)
library(sf)
library(rnaturalearth)
library(rnaturalearthdata)
library(gifski)
library(transformr)

country_long <- country_df %>%
  pivot_longer(cols = -`Continent/Country of Nationality`,
               names_to = "Year", values_to = "Value")
country_long$Year <- as.numeric(country_long$Year)


# load the world map
world_map <- ne_countries(scale = "medium", returnclass = "sf")

# adjust the name of countries
country_long <- country_long %>%
  rename(country = `Continent/Country of Nationality`)

# combine the map data and refugees data
map_data <- world_map %>%
```

```
  left_join(country_long, by = c("name" = "country"))

country_long$Year <- as.numeric(country_long$Year)

#
plot_yearly_maps <- function(year) {
  yearly_data <- map_data %>% filter(Year == year)

  ggplot() +
    geom_sf(data = world_map, fill = "gray90", color = "white") +
    geom_sf(data = yearly_data, aes(fill = Value), color = "black") +
    scale_fill_gradient(low = "#556b84", high = "#d4a373", na.value = "gray90") +
    theme_minimal() +
    labs(title = paste("Global Refugee Map - Year", year), fill = "Number of Refugees")
}

#
plot_yearly_maps(2006)
```

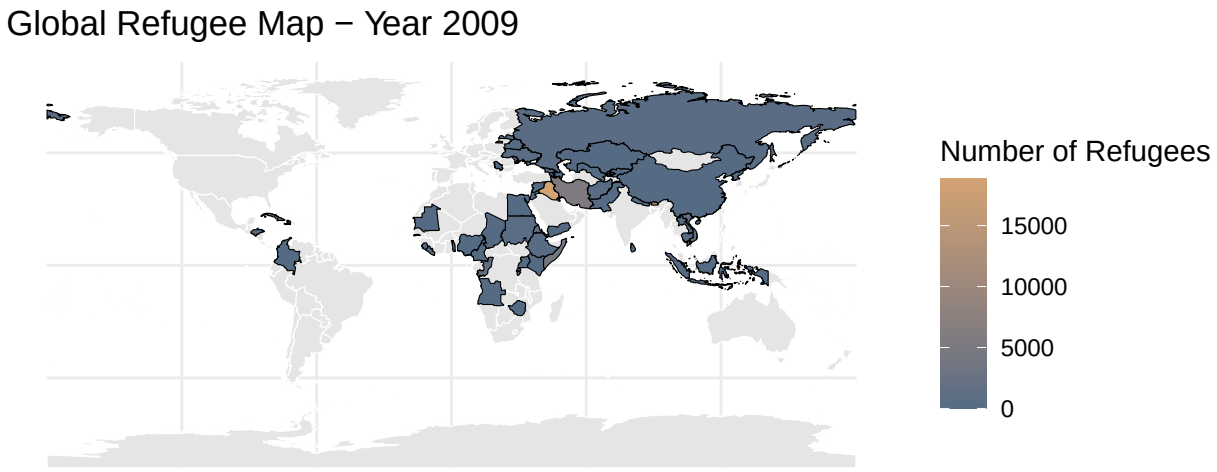



```
plot_yearly_maps(2007)
```

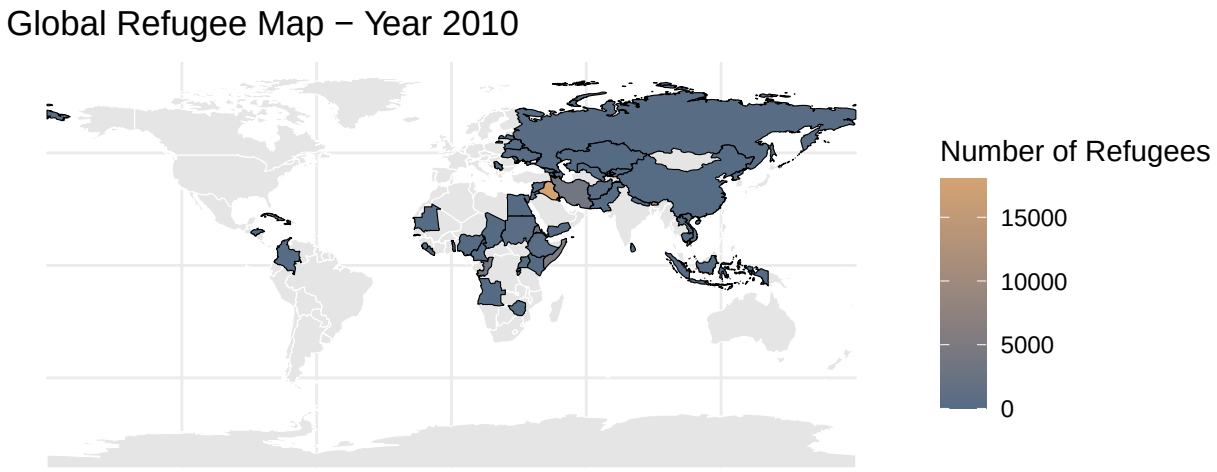

```
plot_yearly_maps(2008)
```




```
plot_yearly_maps(2009)
```

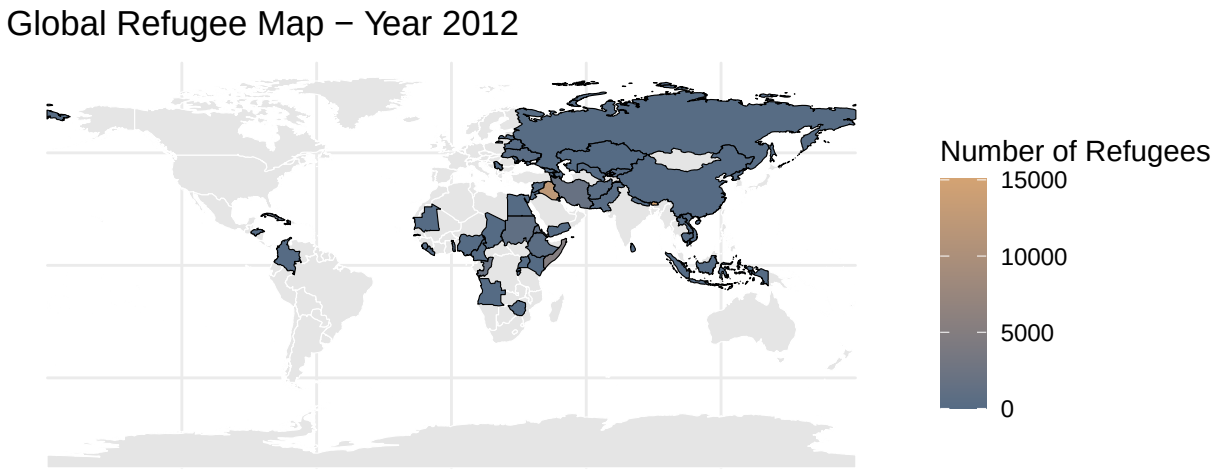



```
plot_yearly_maps(2010)
```

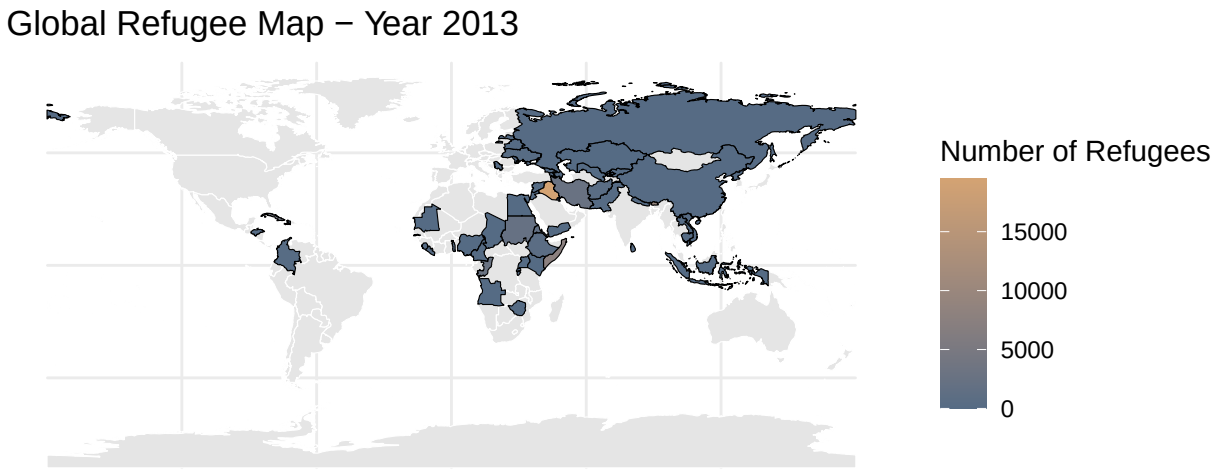



```
plot_yearly_maps(2011)
```

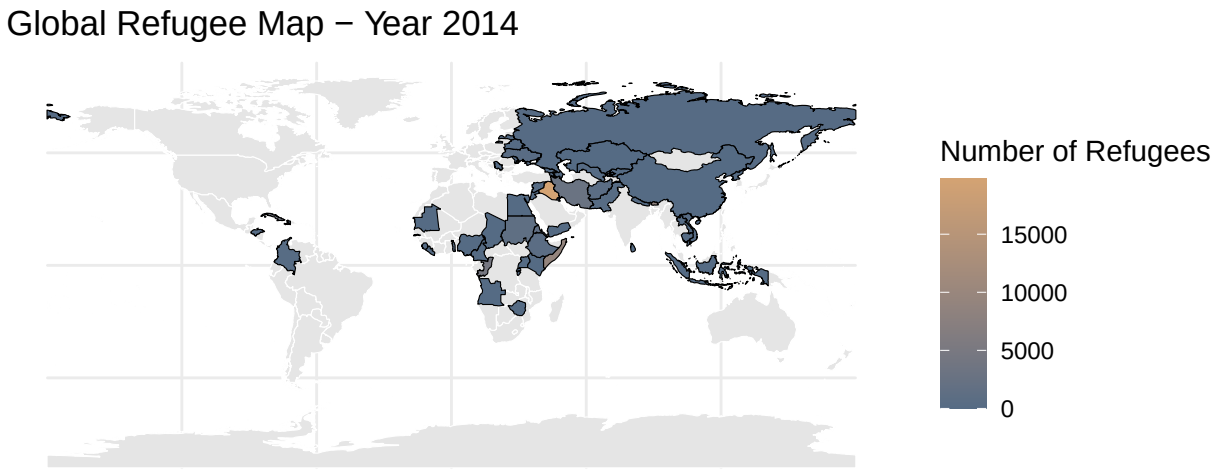

```
plot_yearly_maps(2012)
```



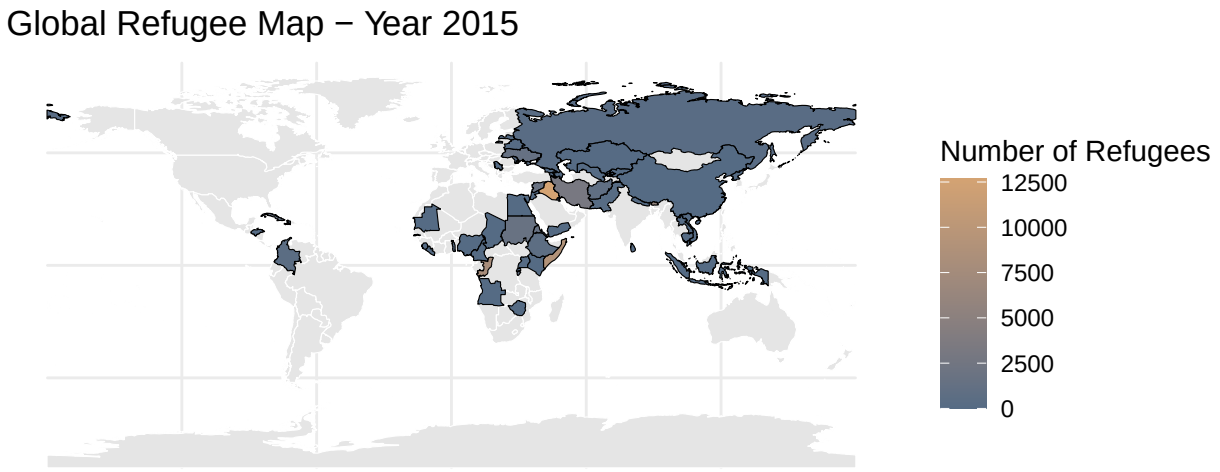

```
plot_yearly_maps(2013)
```




```
plot_yearly_maps(2014)
```




```
plot_yearly_maps(2015)
```



below code is for generate the GIF library(gganimate)

p <- ggplot() + geom_sf(data = world_map, fill = "gray90", color = "white") + # the world map geom_sf(data = map_data, aes(fill = Value), color = "black") + # data refugees scale_fill_gradient(low = "blue", high = "yellow", na.value = "gray90") + theme_minimal() + labs(title = "Global Refugees data Map (Year: {frame_time})", fill = "Number of Refugees", x = " ", y =" ") + transition_time(Year) + # change the plot by year ease_aes('linear') # change smoothly

generate the GIF

anim <- animate(p, duration = 10, fps = 40, width = 800, height = 500, renderer = gifski_renderer()) anim_save("refugees_map_smooth.gif", animation = anim)

Conclusion

This analysis provides a comprehensive look at global refugee movements, highlighting key trends, high-risk regions, and changes over time. The combination of data cleaning, visualization, and statistical analysis allows for an in-depth understanding of the factors driving refugee migration, which can support future policy-making and humanitarian efforts.