A2 Assignment - Refugee Analysis

MBAN - Group 7

Team Members:

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

Load and Clean Data

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

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

# 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))

# Replace the Countries' name with formal format
country_df <- raw_df %>%
```

```r
  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 <- raw_df
head(df,85)
```

```
## # A tibble: 69 x 11
##    Continent/Country of Natio~1 ‘2006‘ ‘2007‘ ‘2008‘ ‘2009‘ ‘2010‘ ‘2011‘ ‘2012‘
##    <chr>                         <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 Africa                        18129  17486   8943   9678  13325   7693  10629
##  2 Asia                          10086  23564  44819  58309  52695  44583  44416
##  3 Europe                         9615   4192   2059   1693   1238    996    908
##  4 North America                  3145   2922   4177   4800   4856   2930   1948
##  5 Oceania                           0      0      0      0      0      0      0
##  6 South America                   119     54    100     57    126     46    130
##  7 Unknown                           0      0      9     65   1053    136    148
##  8 Afghanistan                     651    441    576    349    515    428    481
##  9 Angola                           13      4      0      8      0      0      0
## 10 Armenia                          87     29      9      4      0     15      8
## # i 59 more rows
## # i abbreviated name: 1: ‘Continent/Country of Nationality‘
## # i 3 more variables: ‘2013‘ <dbl>, ‘2014‘ <dbl>, ‘2015‘ <dbl>
```

```r
# 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:"
```

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

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

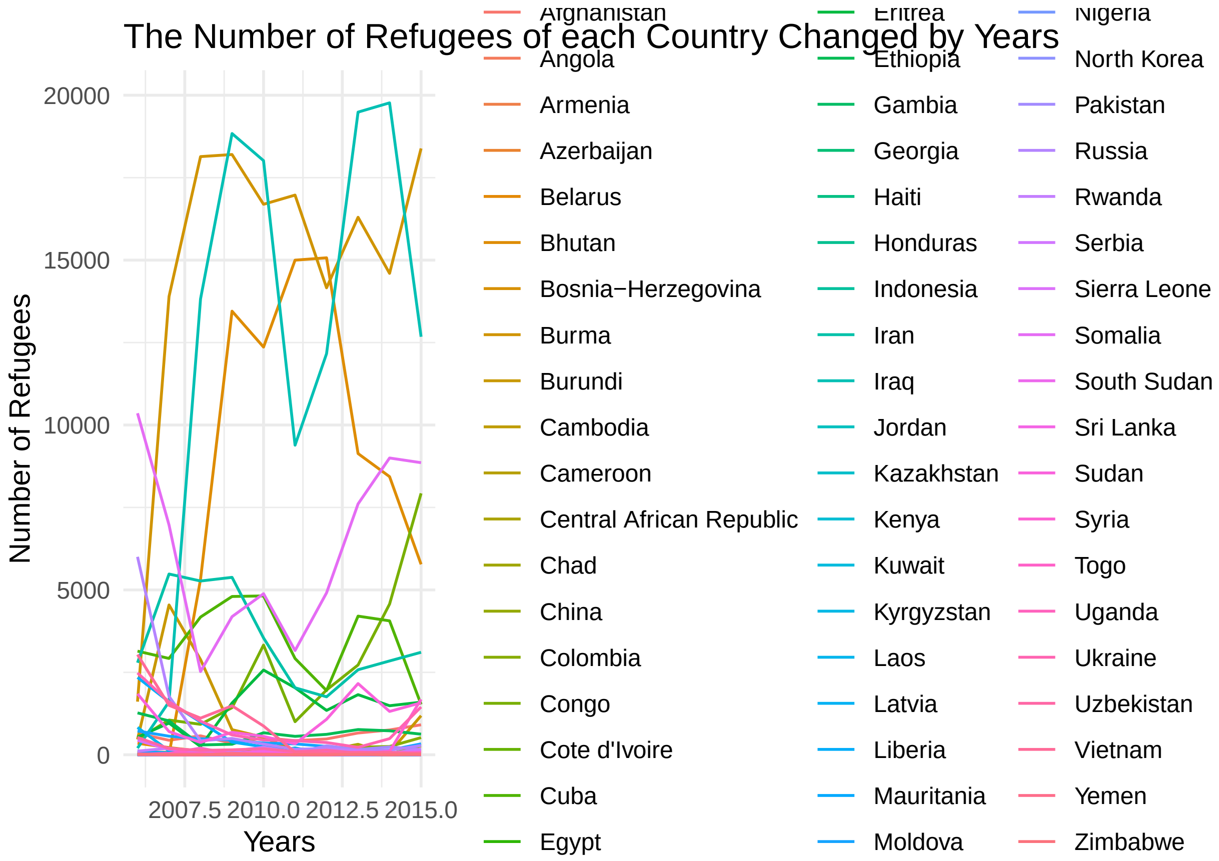
```r
# check the new dataframe
head(country_df)
```

```
## # 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>
```

```r
library(tidyverse)

# read data
country_df <- read_csv("country_data.csv")
```

```
## Rows: 59 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): Continent/Country of Nationality
## dbl (10): 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
library(ggplot2)

# convert the data to the 'long' varibale type, which will be easy to make the plots
country_long <- country_df %>%
  pivot_longer(cols = -`Continent/Country of Nationality`,
               names_to = "Year", values_to = "Value")

# plot the line chart
ggplot(country_long, aes(x = as.numeric(Year), y = Value, color = `Continent/Country of Nationality`)) +
  geom_line() +
  theme_minimal() +
  labs(title = "The Number of Refugees of each Country Changed by Years",
       x = "Years",
       y = "Number of Refugees",
       color = "Countries") +
  theme(legend.position = "right")
```



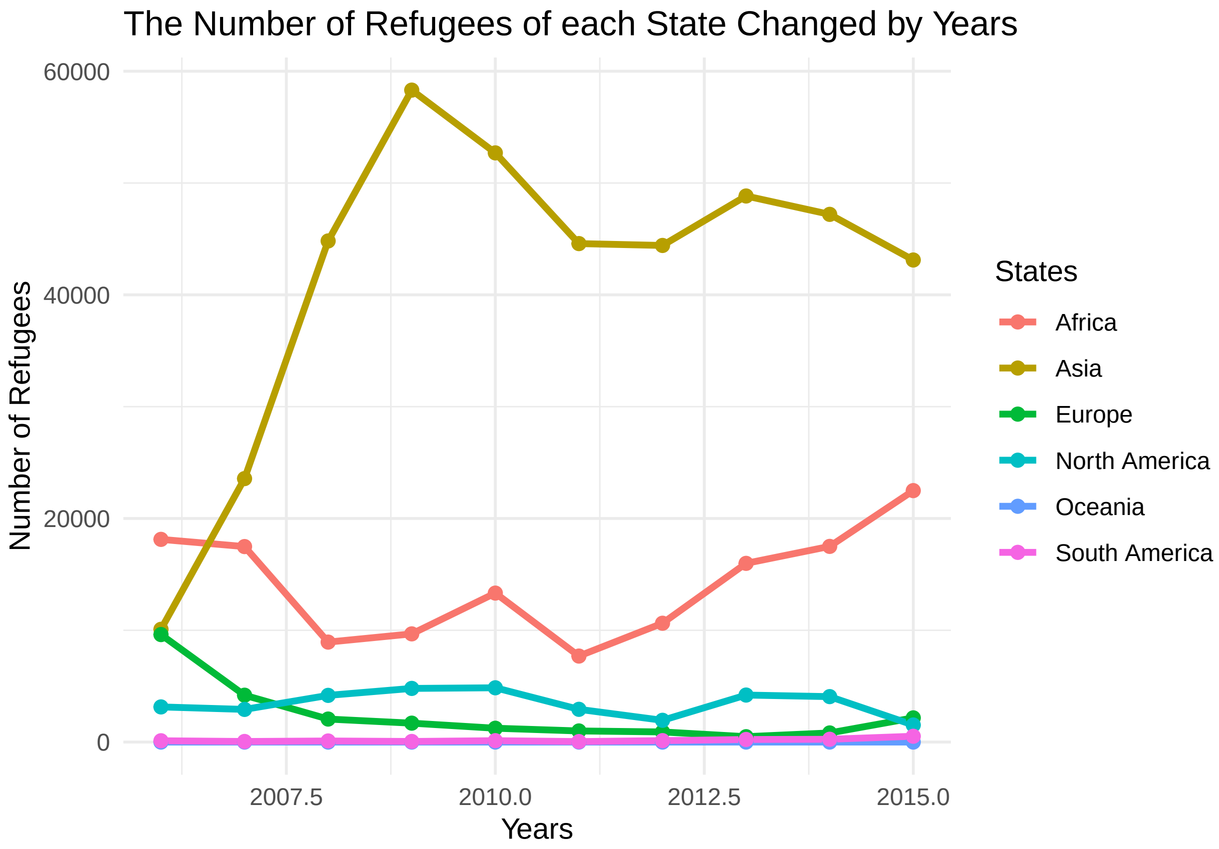
```
# Calculate the number of refugees for each state by year
continent_df <- df %>%
  filter(`Continent/Country of Nationality` %in% c("Africa", "Asia", "Europe",
                                                   "North America", "Oceania", "South America")) %>%
  pivot_longer(cols = -`Continent/Country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

```
## 'summarise()' has grouped output by 'Continent/Country of Nationality'. You can
## override using the '.groups' argument.
```

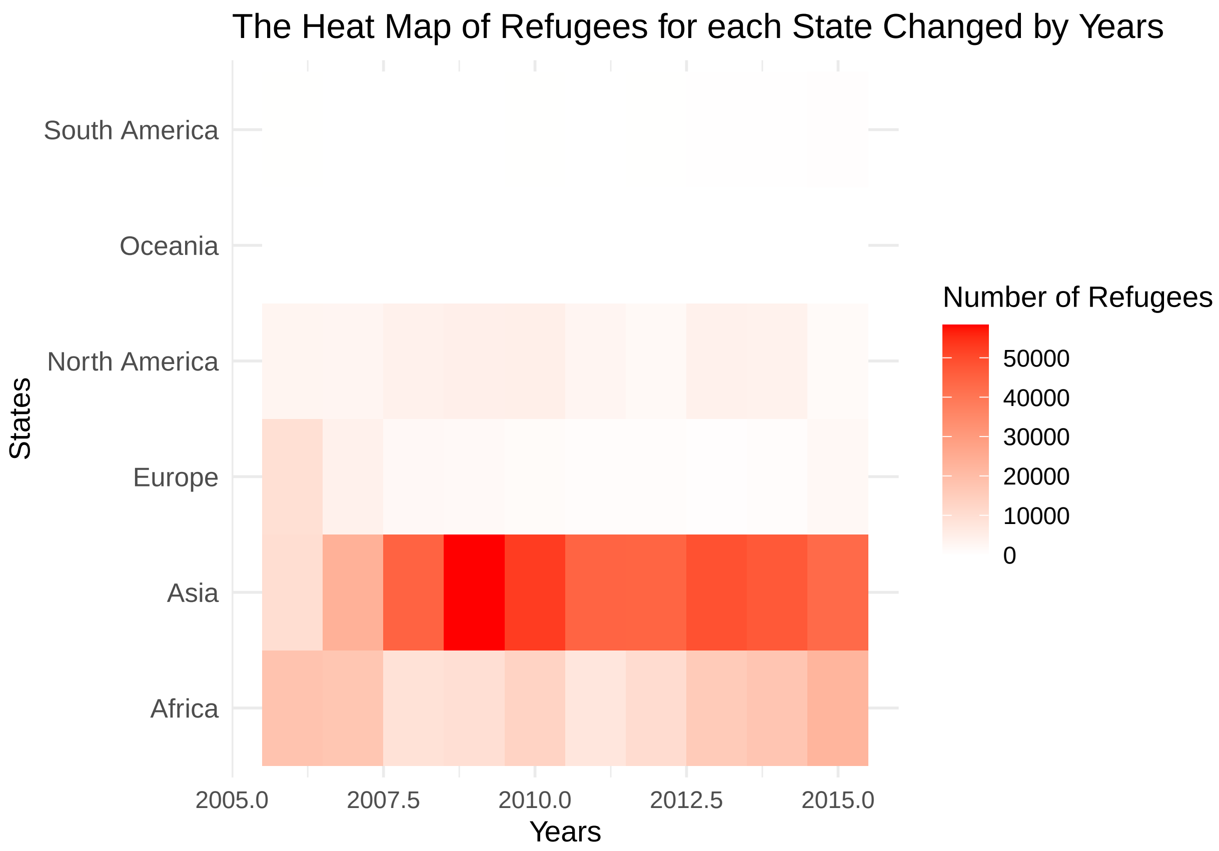
```
# make sure the 'year' value is in Number data format
continent_df$Year <- as.numeric(continent_df$Year)
```

```
library(ggplot2)

ggplot(continent_df, aes(x = Year, y = Total_Refugees, color = `Continent/Country of Nationality`)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  theme_minimal() +
  labs(title = "The Number of Refugees of each State Changed by Years",
       x = "Years",
       y = "Number of Refugees",
      color = "States") +
  theme(legend.position = "right")
```

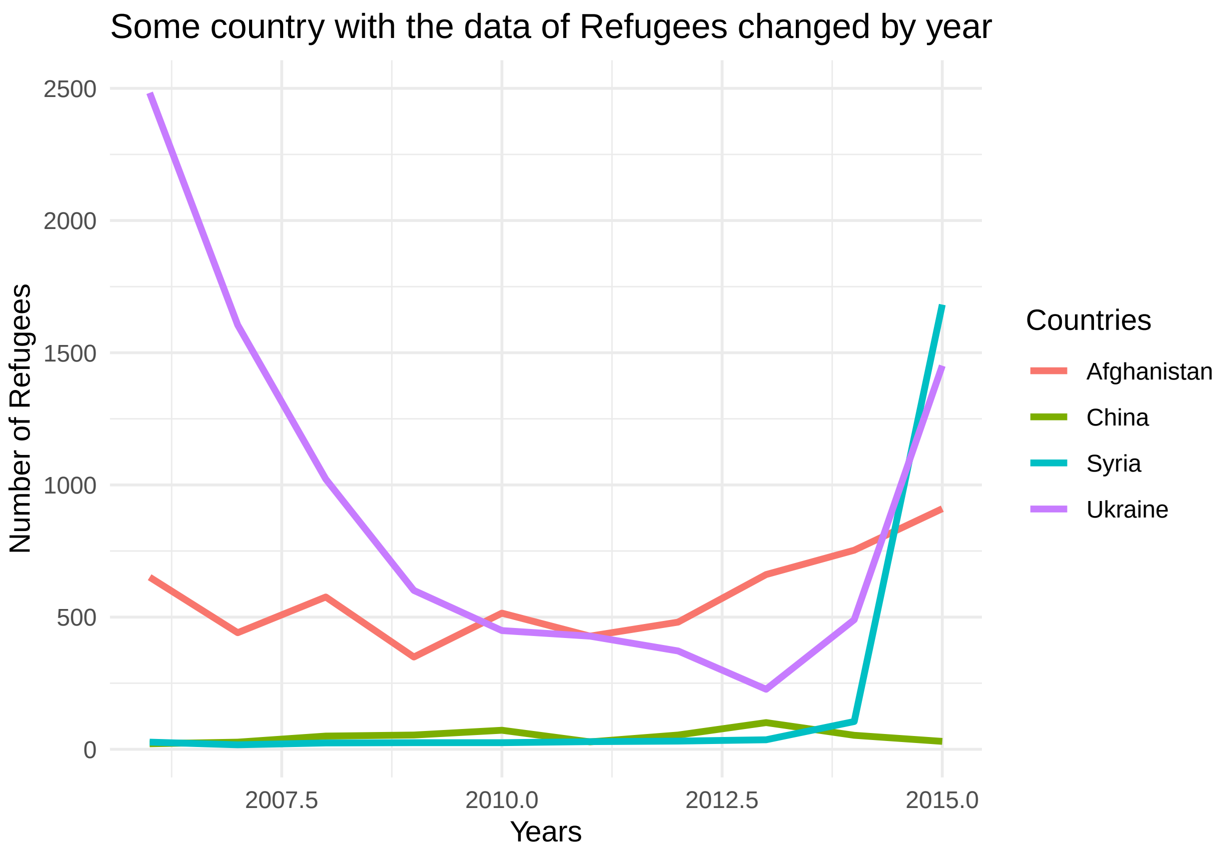
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



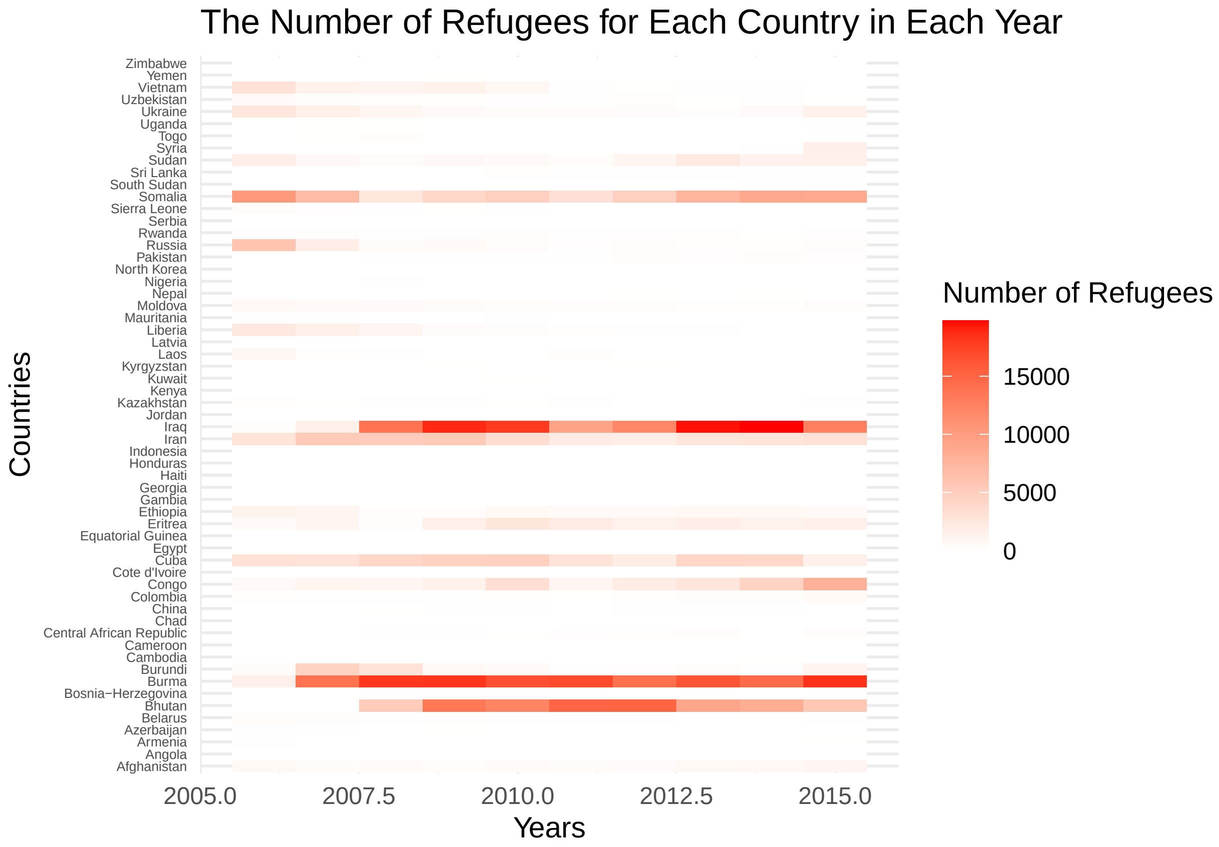
```
ggplot(continent_df, aes(x = Year, y = `Continent/Country of Nationality`, fill = Total_Refugees)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +  # set the color
  theme_minimal() +
  labs(title = "The Heat Map of Refugees for each State Changed by Years",
      x = "Years",
      y = "States",
      fill = "Number of Refugees") +
  theme(legend.position = "right",
       axis.text.y = element_text(size = 10))
```



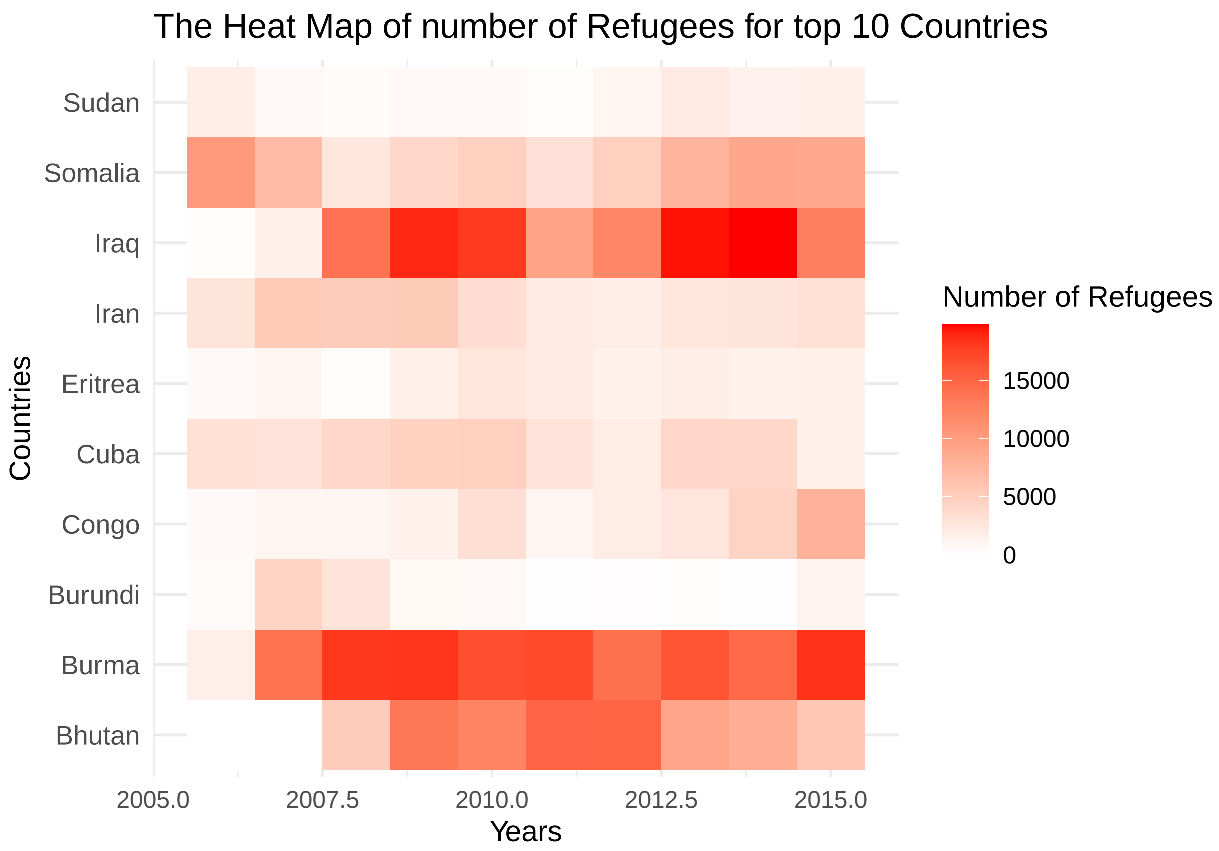
```
selected_countries <- c("China", "India", "Syria", "Afghanistan", "Ukraine")

filtered_df <- country_long %>% filter(`Continent/Country of Nationality` %in% selected_countries)

ggplot(filtered_df, aes(x = as.numeric(Year), y = Value, color = `Continent/Country of Nationality`)) +
  geom_line(size = 1.2) +
  theme_minimal() +
  labs(title = "Some country with the data of Refugees changed by year",
       x = "Years",
       y = "Number of Refugees",
       color = "Countries") +
  theme(legend.position = "right")
```



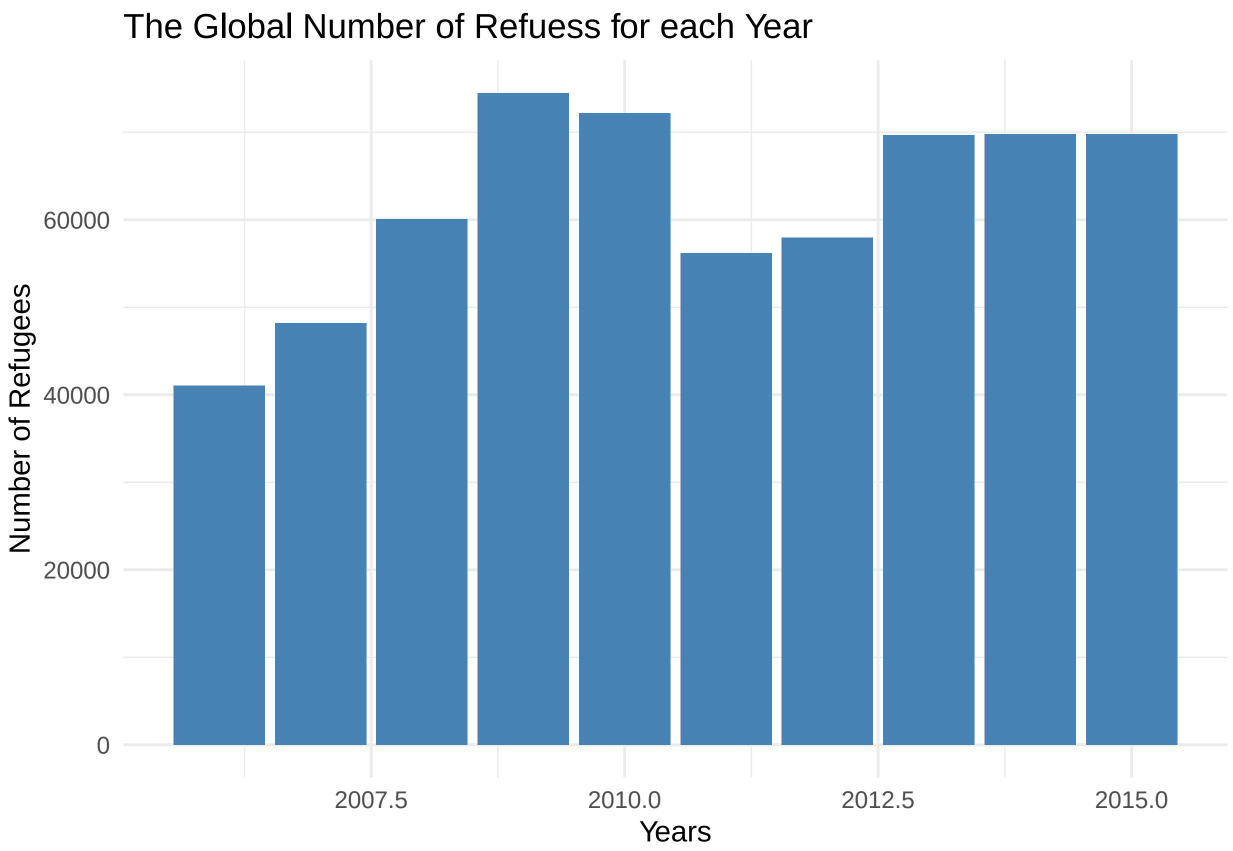
```r
ggplot(country_long, aes(x = as.numeric(Year), y = `Continent/Country of Nationality`, fill = Value)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  theme_minimal() +
  labs(title = "The Number of Refugees for Each Country in Each Year",
       x = "Years",
       y = "Countries",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



```r
# Calculate the number of Refugees for each country in Total
top_countries <- country_long %>%
  group_by(`Continent/Country of Nationality`) %>%
  summarise(Total_Refugees = sum(Value, na.rm = TRUE)) %>%
  arrange(desc(Total_Refugees)) %>%
  slice_head(n = 10)  #  pick top 10

# get the name of country in top 10
top_country_names <- top_countries$`Continent/Country of Nationality`

# filter the data with top 10
filtered_df <- country_long %>%
  filter(`Continent/Country of Nationality` %in% top_country_names)

ggplot(filtered_df, aes(x = as.numeric(Year), y = `Continent/Country of Nationality`, fill = Value)) +
  geom_tile() +
  scale_fill_gradient(low = "white", high = "red") +
  theme_minimal() +
  labs(title = "The Heat Map of number of Refugees for top 10 Countries",
       x = "Years",
       y = "Countries",
       fill = "Number of Refugees") +
  theme(legend.position = "right",
        axis.text.y = element_text(size = 10))
```



```r
# Caculate the total number of each year
yearly_totals <- country_long %>%
  group_by(Year) %>%
  summarise(Total = sum(Value, na.rm = TRUE))

# Draw the bar chart
ggplot(yearly_totals, aes(x = as.numeric(Year), y = Total)) +
  geom_col(fill = "steelblue") +
  theme_minimal() +
  labs(title = "The Global Number of Refuess for each Year",
       x = "Years",
       y = "Number of Refugees")
```



```r
library(tidyverse)
library(ggplot2)
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"))
```

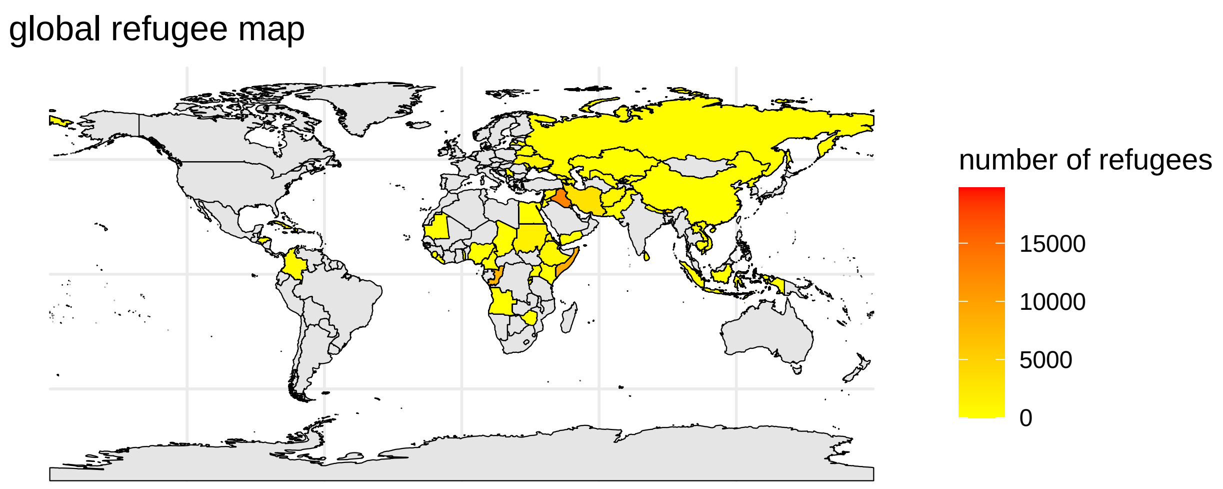
```
# check the complete of the dataframe
summary(country_long$Value)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     0.0     5.0    28.0  1049.9   313.2 19769.0
```

```
table(is.na(country_long$Value))  # check is there NULL in the data set
```

```
## 
## FALSE 
##   590
```

```
ggplot() +
  geom_sf(data = world_map, fill = "gray90", color = "white") +
  geom_sf(data = map_data, aes(fill = Value), color = "black") +
  scale_fill_gradient(low = "yellow", high = "red", na.value = "gray90") +
  theme_minimal() +
  labs(title = "global refugee map ", fill = "number of refugees")
```



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 = “yellow”, high = “red”, 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)
```