Reproducing the visualisations using R programming language

Email: ravanasamuthram_krishnan.pooja@stud.hs-fresenius.de

Name: Pooja Ravanasamuthram Krishnan ID: 400376705

Last compiled on 15 January, 2025

Contents

Abstract

This paper analyzes the origin and trends of economic sanctions (in international law also referred to as restrictive measures), 1950–2022, drawing from the Global Sanctions Database. We reproduce the graphs in the study using the R programming language (original code provided in Stata). The dataset needed for the graphs is however not provided, which creates a problem for replication. The consequences of sanctions on political goals are central to this

paper. The types of graphs are line graph, bar graph, scatter plot, stacked area chart and composite graph showing different trend in data. As observed, despite the overall complexity and variety of visual trends present in the data, all graphs have similarities in the high customisation in order to further make the data easily understandable, the use of the same colour schemes to distinguish different categories, and the concentration of time as a common factor as the X-axis, illustrating trends and shifts over decades. This project demonstrates the utility of R in replicating complex visualizations and offers insights into the process of analyzing economic sanctions data.

Project Overview

The project explores converting raw data into meaningful visualizations through R, a statistical programming language, in terms of how well they reflect findings in the paper. This project is also quite useful in understanding how the methodologies were done in the study while it has practical lessons on the challenges and fine points of data analysis and visualization in R.This project focuses on replicating visualizations from the paper "Economic Sanctions: Evolution, Consequences, and Challenges." The paper discusses the history, objectives, and effectiveness of economic sanctions, with an emphasis on their evolution over time and their political and economic goals.

Significance of using R

R is widely used in the data science job market and can be run on any operating system, such as Windows, macOS, Linux, and many other environments. It can be integrated with various programming languages, including C, C#, and Python. The number of R users and those interested in learning it is significantly increasing, largely because it is free to use, eliminating the need for financial investment and licensing. As an open-source programming language, R is accessible to anyone. With the growing number of users, many skilled professionals use R

as part of their profession, while others learn it out of personal interest.

Replication fo stata code and plot in R

The capabilities of this language are extended by the Comprehensive R Archive Network (CRAN), which contains numerous libraries and packages, such as ggplot2 and dplyr. These resources can be applied in various fields, including machine learning and bioinformatics, further enhancing the versatility and utility of R.

```
options(repos = c(CRAN = "https://cran.rstudio.com/"))


install.packages("dplyr")
```

```
## Installing package into 'C:/Users/pooja/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'dplyr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\pooja\AppData\Local\Temp\RtmpqMGCIu\downloaded_packages
```

```
install.packages("haven")
```

```
## Installing package into 'C:/Users/pooja/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'haven' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\pooja\AppData\Local\Temp\RtmpqMGCIu\downloaded_packages
```

```
install.packages("ggplot2")
```

```
## Installing package into 'C:/Users/pooja/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\pooja\AppData\Local\Temp\RtmpqMGCIu\downloaded_packages
```

```
install.packages("reshape2")
```

```
## Installing package into 'C:/Users/pooja/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

```
## package 'reshape2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\pooja\AppData\Local\Temp\RtmpqMGCIu\downloaded_packages
```

To replicate the first graph, which is a stacked area graph, necessary libraries such as 'dplyr', 'tidyr', 'ggplot2', and 'readxl' are loaded. Since the dataset is in Excel format, the 'readxl' library is used as it supports older Excel file formats like '.xls' and '.xlsx'. The Excel file is loaded, and the sheet = 1 argument is specified to indicate that the first sheet of the Excel file should be loaded.

```
#Loading necessary libraries
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(tidyr)
library(ggplot2)
library(readxl)



#loading csv file
gsdb_data <- read_excel("C:/Users/pooja/OneDrive/Data Science- Economic Sanctions/GSDB_V
```

The column 'other_sanct' is renamed to 'other' for clarity and to prevent potential conflicts with the generic name 'other'. The 'objective' column is split into four separate columns: 'objective1', 'objective2', 'objective3', and 'objective4', with missing values filled automatically. A for loop is used to check which variables belong to each objective. Similarly, the 'success' column is also split into separate columns.

```
gsdb_data <- gsdb_data %>% rename(other_sanct = other)


gsdb_data <- gsdb_data %>%
  separate(objective, into = c("objective1", "objective2", "objective3",
  "objective4"), sep = ",", fill = "right")


objectives <- c("democracy", "destab_regime", "end_war", "human_rights",
                "terrorism", "territorial_conflict", "prevent_war",
                "policy_change", "other")
```

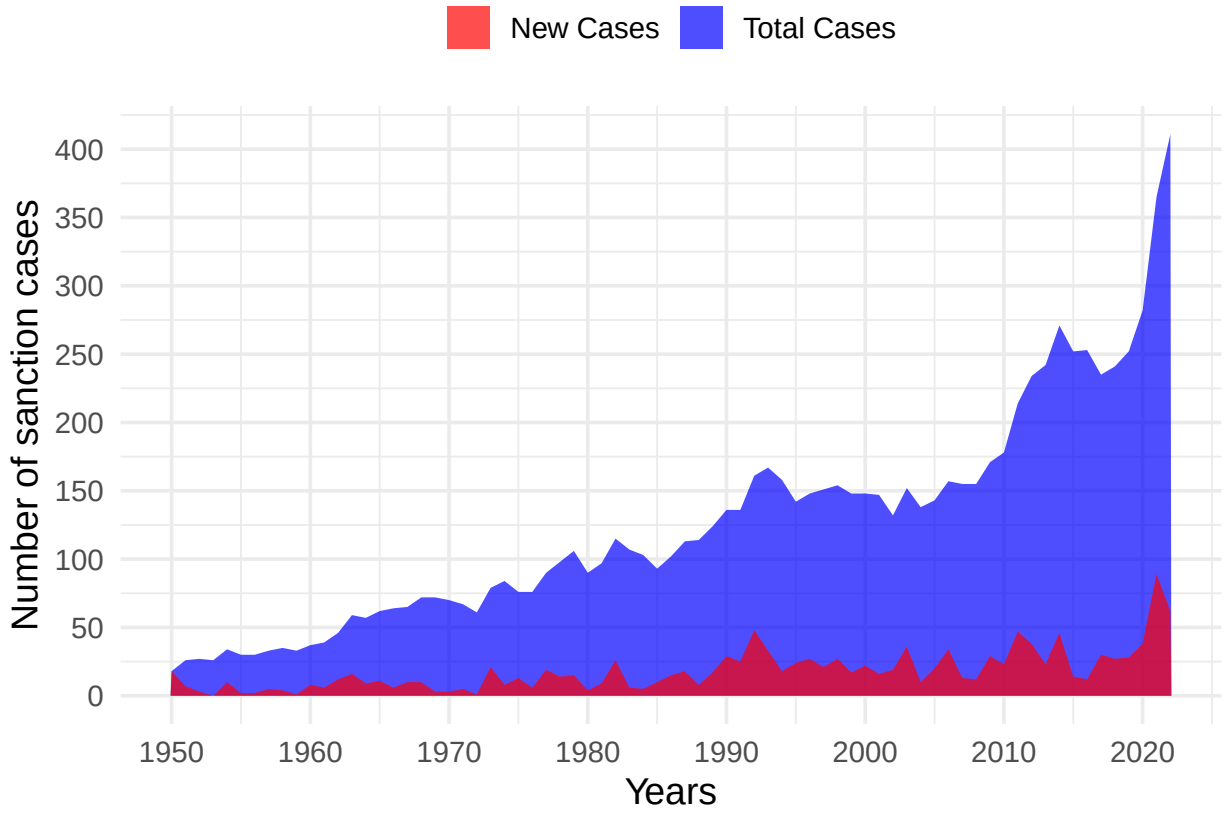
```
for (obj in objectives) {
  gsdb_data[[obj]] <- ifelse(gsdb_data$objective1 == obj | gsdb_data$objective2
                             == obj | gsdb_data$objective3 == obj |
                               gsdb_data$objective4 == obj, 1, 0)
}

gsdb_data <- gsdb_data %>%
  separate(success, into = c("success1", "success2", "success3", "success4"),
           sep = ",", fill = "right")

success_types <- c("success_total", "success_part", "nego_settlement", "failed",
                   "ongoing")

for (succ in success_types) {
  gsdb_data[[succ]] <- ifelse(gsdb_data$success1 == succ | gsdb_data$success2 ==
                                succ | gsdb_data$success3 == succ |
                                gsdb_data$success4 == succ, 1, 0)
}

sanct_begin <- gsdb_data %>%
  select(case_id, begin) %>%
  rename(year = begin)

sanct_end <- gsdb_data %>%
  select(case_id, end) %>%
  rename(year = end)
```

```r
gsdb_case <- bind_rows(sanct_begin, sanct_end) %>%
  distinct() %>%
  arrange(case_id, year) %>%
  filter(year >= 1950)

expanded_data <- gsdb_case %>%
  group_by(case_id) %>%
  complete(year = full_seq(year, 1)) %>%
  left_join(gsdb_data, by = "case_id")

expanded_data <- expanded_data %>%
  filter(year >= 1950)

expanded_data <- expanded_data %>%
  mutate(id_new = ifelse(year %in% sanct_begin$year[sanct_begin$case_id ==
  case_id], 1, 0))
```

```
## Warning: There were 1027 warnings in `mutate()`.
## The first warning was:
## i In argument: `id_new = ifelse(...)`.
## i In group 3: `case_id = 3`.
## Caused by warning in `sanct_begin$case_id == case_id`:
## ! longer object length is not a multiple of shorter object length
## i Run `dplyr::last_dplyr_warnings()` to see the 1026 remaining warnings.
```

```r
expanded_data <- expanded_data %>%
  mutate(id = 1)
```

```r
collapsed_data <- expanded_data %>%
  group_by(year) %>%
  summarise(total_cases = sum(id, na.rm = TRUE),
            new_cases = sum(id_new, na.rm = TRUE))


ggplot(collapsed_data, aes(x = year)) +
  geom_area(aes(y = total_cases, fill = "Total Cases"), alpha = 0.7) +
  geom_area(aes(y = new_cases, fill = "New Cases"), alpha = 0.7) +
  scale_fill_manual(values = c("Total Cases" = "blue", "New Cases" = "red")) +
  labs(y = "Number of sanction cases", x = "Years") +
  scale_y_continuous(breaks = seq(0, 400, by = 50)) +
  scale_x_continuous(breaks = seq(1950, 2022, by = 10)) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "top", legend.title = element_blank())
```

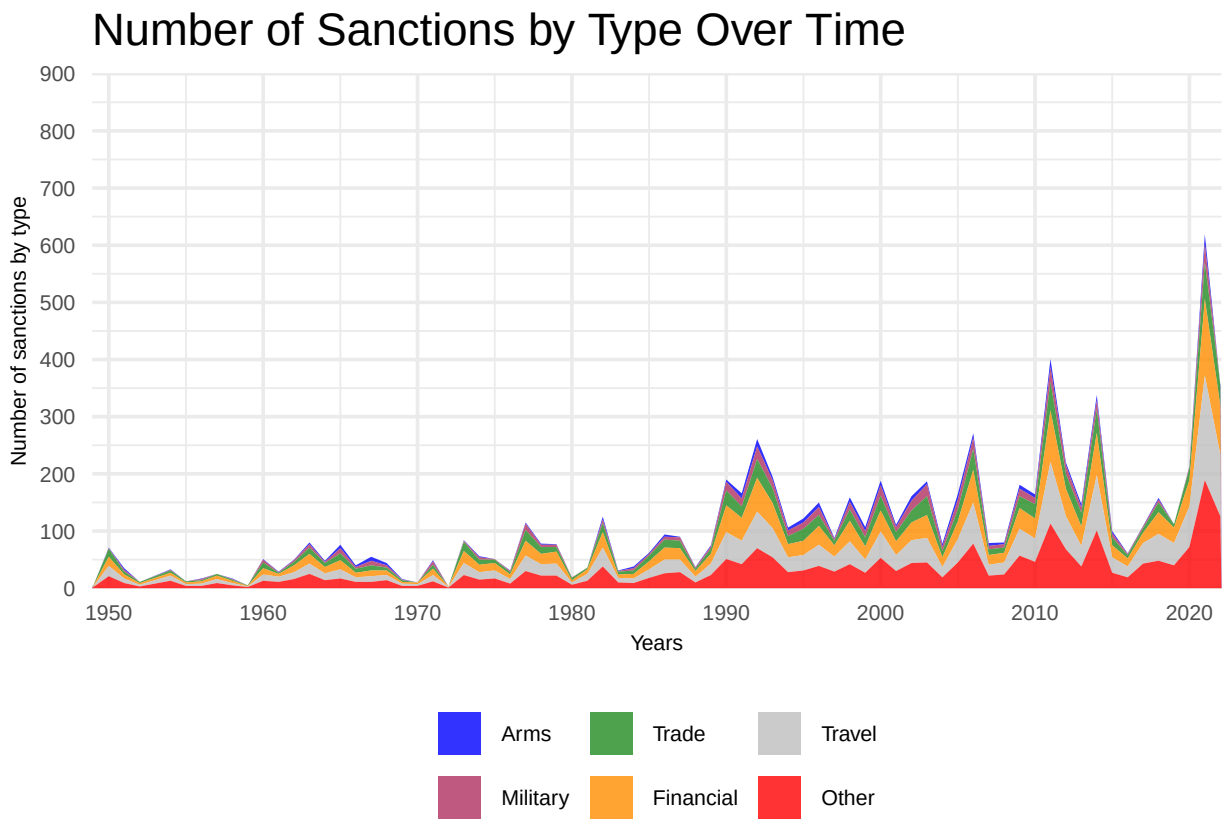



```
ggsave("evol_sanct.png", width = 8, height = 6, dpi = 300)

ggsave("figure_1.pdf", width = 8, height = 6)
```

```
#Loading necessary libraries

library(dplyr)

library(ggplot2)

library(readr)


gsdb_v3_case <- read_excel("C:/Users/pooja/OneDrive/Data Science- Economic Sanctions/GSD


gsdb_v3_case <- gsdb_v3_case %>%

  mutate(year = begin)
```

```r
collapsed_data <- gsdb_v3_case %>%
  group_by(year) %>%
  summarise(
    arms = sum(arms, na.rm = TRUE),
    military = sum(military, na.rm = TRUE),
    trade = sum(trade, na.rm = TRUE),
    financial = sum(financial, na.rm = TRUE),
    travel = sum(travel, na.rm = TRUE),
    other_sanct = sum(other, na.rm = TRUE)  # Use 'other' instead of 'other_sanct'
  )



collapsed_data <- collapsed_data %>%
  mutate(
    sum2 = arms + military,
    sum3 = arms + military + trade,
    sum4 = arms + military + trade + financial,
    sum5 = arms + military + trade + financial + travel,
    sum6 = arms + military + trade + financial + travel + other_sanct
  )


plot_data <- collapsed_data %>%
  select(year, arms, sum2, sum3, sum4, sum5, sum6) %>%
  pivot_longer(cols = c("arms", "sum2", "sum3", "sum4", "sum5", "sum6"),
               names_to = "category", values_to = "value") %>%
  mutate(category = factor(category, levels = c("arms", "sum2", "sum3", "sum4",
                                                "sum5", "sum6")))
```

```r
#defining color palette for the categories
colors <- c("blue", "maroon", "forestgreen", "darkorange", "grey", "red")


#plotting the graph
ggplot(plot_data, aes(x = year, y = value, fill = category)) +
  geom_area(alpha = 0.8, color = NA) +
  scale_fill_manual(
    values = colors,
    labels = c("Arms", "Military", "Trade", "Financial", "Travel", "Other")
  ) +
  labs(y = "Number of sanctions by type", x = "Years") +
  scale_y_continuous(breaks = seq(0, 900, by = 100), limits = c(0, 900), expand
                     = c(0, 0)) +
  scale_x_continuous(breaks = seq(1950, 2022, by = 10), expand = c(0, 0)) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "bottom",
    legend.title = element_blank(),
    legend.text = element_text(size = 8),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 8),
    plot.background = element_rect(fill = "white", color = "white")
  ) +
  ggtitle("Number of Sanctions by Type Over Time")
```



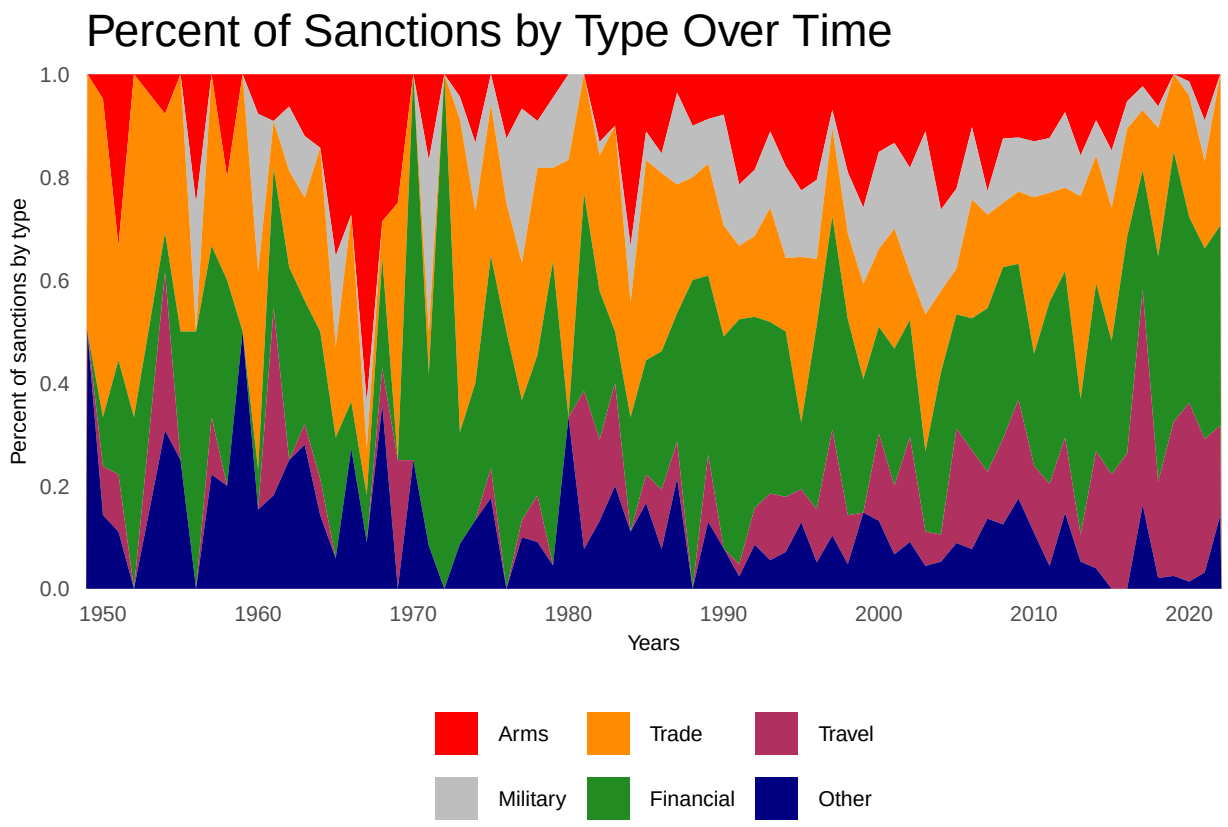

```
#Saving the plot
ggsave("evol_type_levl.png", width = 10, height = 6, dpi = 300)
ggsave("figure_2a.pdf", width = 10, height = 6)
```

```
##Figure 2(b)


#Loading necessary libraries
library(dplyr)
library(ggplot2)
library(readr)
library(tidyr)


#loading the csv file
```

```r
gsdb_v3_case <- read_excel("C:/Users/pooja/OneDrive/Data Science- Economic Sanctions/GSD


gsdb_v3_case <- gsdb_v3_case %>%
  mutate(year = begin)


collapsed_data <- gsdb_v3_case %>%
  group_by(year) %>%
  summarise(
    arms = sum(arms, na.rm = TRUE),
    military = sum(military, na.rm = TRUE),
    trade = sum(trade, na.rm = TRUE),
    financial = sum(financial, na.rm = TRUE),
    travel = sum(travel, na.rm = TRUE),
    other_sanct = sum(other, na.rm = TRUE)  # Use 'other' instead of 'other_sanct'
  )


collapsed_data <- collapsed_data %>%
  mutate(
    total_sanctions = arms + military + trade + financial + travel + other_sanct,
    perc_arms = arms / total_sanctions,
    perc_military = military / total_sanctions,
    perc_trade = trade / total_sanctions,
    perc_financial = financial / total_sanctions,
    perc_travel = travel / total_sanctions,
    perc_other = other_sanct / total_sanctions
  )
```

```r
plot_data <- collapsed_data %>%
  select(year, perc_arms, perc_military, perc_trade, perc_financial, perc_travel
         , perc_other) %>%
  pivot_longer(cols = starts_with("perc_"), names_to = "category", values_to =
                 "value") %>%
  mutate(category = factor(category,
                           levels = c("perc_arms", "perc_military", "perc_trade"
                                      ,"perc_financial", "perc_travel",
                                      "perc_other"),
                           labels = c("Arms", "Military", "Trade", "Financial",
                                      "Travel", "Other")))


#defining color palette for the categories
colors <- c("red", "grey", "darkorange", "forestgreen", "maroon", "navy")


#Plotting the graph
ggplot(plot_data, aes(x = year, y = value, fill = category)) +
  geom_area(position = "stack", color = NA) +  # Ensure proper stacking
  scale_fill_manual(values = colors) +
  labs(y = "Percent of sanctions by type", x = "Years") +
  scale_y_continuous(breaks = seq(0, 1, by = 0.2), limits = c(0, 1),
                     expand = c(0, 0)) +
  scale_x_continuous(breaks = seq(1950, 2022, by = 10), expand = c(0, 0)) +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "bottom",
    legend.title = element_blank(),
```

```
    legend.text = element_text(size = 8),
    axis.text = element_text(size = 8),
    axis.title = element_text(size = 8),
    plot.background = element_rect(fill = "white", color = "white")
  ) +
  ggtitle("Percent of Sanctions by Type Over Time")
```

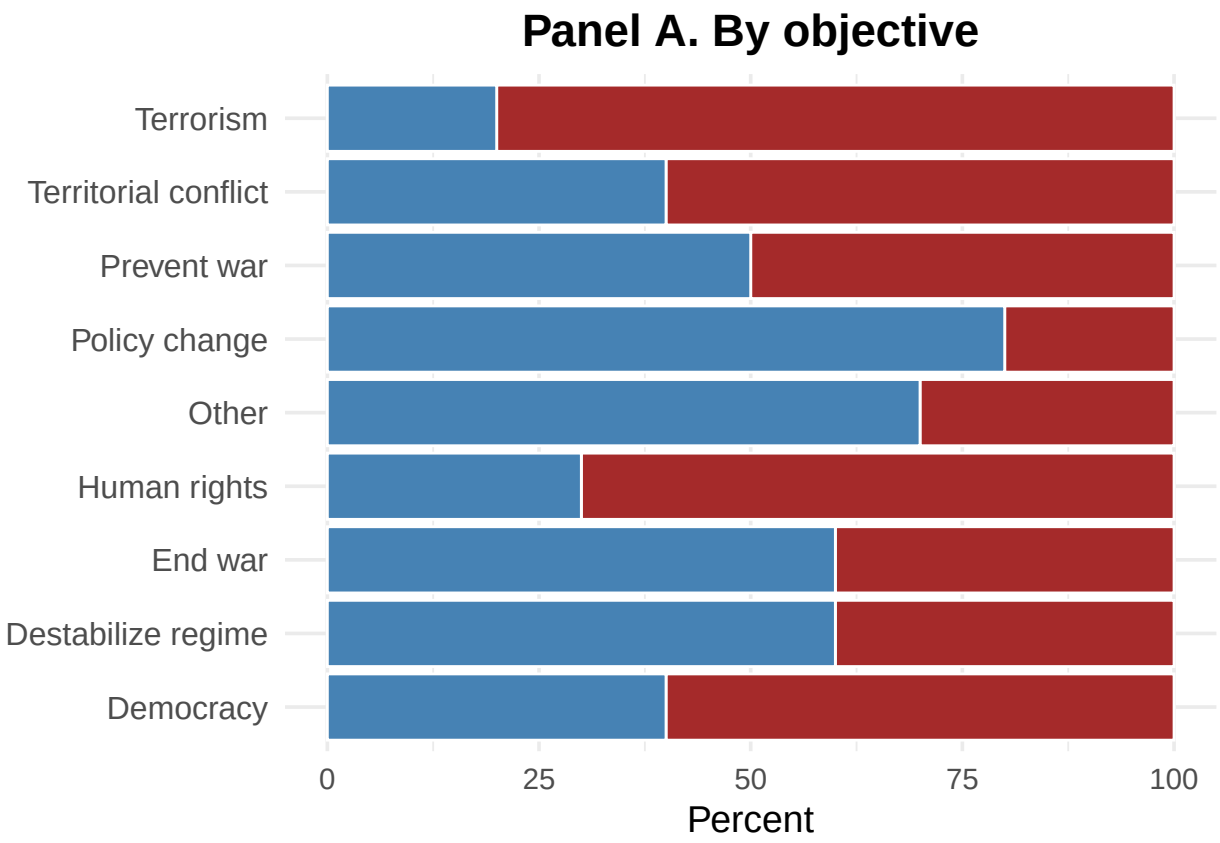



```
#Saving the plot
ggsave("evol_type_perc_fixed.png", width = 10, height = 6, dpi = 300)
ggsave("figure_2b_fixed.pdf", width = 10, height = 6)
```

```
  # Load necessary libraries
  library(ggplot2)
library(dplyr)
```

```r
# Step 1: Create a sample dataset (replace with your data)
data <- data.frame(
  Objective = c("Terrorism", "Territorial conflict", "Destabilize regime",
                "Policy change", "Other", "Prevent war", "End war",
                "Human rights", "Democracy"),
  Category1 = c(80, 60, 40, 20, 30, 50, 40, 70, 60),# Values for blue bars

  Category2 = c(20, 40, 60, 80, 70, 50, 60, 30, 40)  # Values for red bars
)



# Step 2: Reshape the data into long format
data_long <- data %>%
  tidyr::pivot_longer(cols = c("Category1", "Category2"),
                      names_to = "Category", values_to = "Percent")

# Step 3: Plot the stacked horizontal bar chart
ggplot(data_long, aes(x = Percent, y = Objective, fill = Category)) +
  geom_bar(stat = "identity", position = "stack", color = "white") +
  scale_fill_manual(values = c("Category1" = "brown", "Category2" = "steelblue")) +
  labs(x = "Percent", y = NULL, title = "Panel A. By objective") +
  theme_minimal(base_size = 14) +
  theme(
    legend.position = "none",
    axis.text.y = element_text(size = 12),
    plot.title = element_text(hjust = 0.5, face = "bold")
```

```
)
```


Panel A. By objective
Terrorism
Territorial conflict
Prevent war
Policy change
Other
Human rights
End war
Destabilize regime
Democracy
0
25
50
75
100
Percent