BST270 Congress Trends 538 Reproduce

Daniel Paydarfar

2025-01-22

Packages

In the cell below, we load the necessary packages for reproducing the article figures. The commented out code are for libraries used for preprocessing. Note that all preprocessing steps in this .rmd are commented out and we instead provide .csv files for processed data to save you a step.

```
library(ggplot2)
# library(dplyr)
# library(lubridate)
```

Loading the Data

In the cell below, we load data from our local directory downloaded from https://data.fivethirtyeight.com/. The data was used in the following FiveThirtyEight article: https://fivethirtyeight.com/features/aging-congress-boomers/

```
df = read.csv('data_aging_congress.csv')
```

Replicating 'Median age of the U.S. Senate and U.S. House by Congress, 1919 to 2023'

Data Preprocessing

Explanation of the Steps: Step 1: Load the Dataset Load the dataset into R using read.csv(). Replace the file name with the correct path to your dataset. Step 2: Extract Year from start_date Convert the start_date column to a date format using as.Date() and extract the year using lubridate::year(). Step 3: Filter for Senate and House Focus on records where the chamber column is either "Senate" or "House." Select only the necessary columns: start_year (year), chamber (Senate/House), and age_years (age in years). Step 4: Calculate Median Age Group the data by start_year and chamber using group_by() and calculate the median age for each group using summarize(). Step 5: Reshape for Plotting Reshape the data into a wide format using pivot_wider() so that Senate and House ages are in separate columns. This makes it easier to plot multiple lines in a single graph. Step 6: Save Processed Data Save the processed data to a CSV file (processed_congress_data.csv) using write.csv(). This file can then be used to generate the plot.

The cell below contains code that performs these preprocessing steps. However, we have commented it out and instead provide the processed .csv file for your convenience. Use the processed data"processed_congress_data_time.csv" to generate the figure.

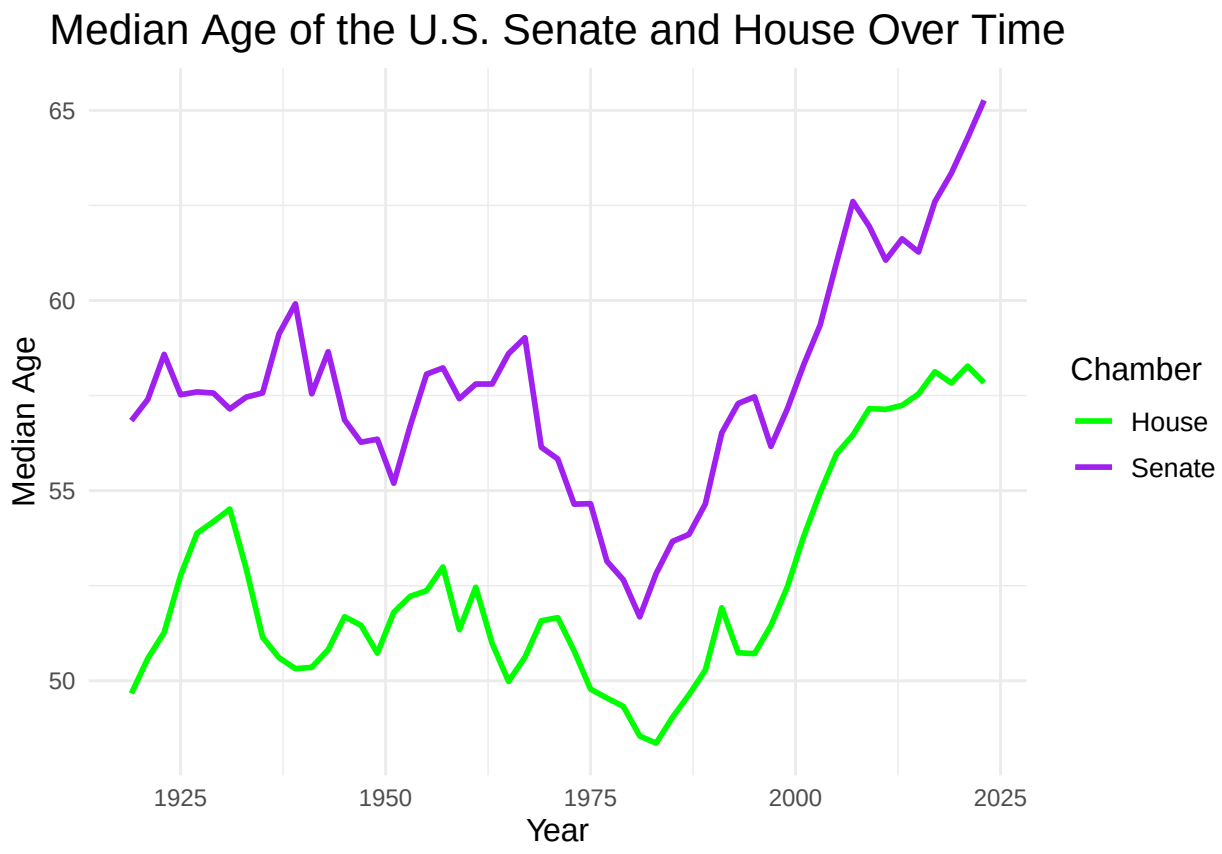
```
# # Load necessary libraries
#
# # Step 1: Load the original dataset
# # Replace 'data_aging_congress.csv' with the correct path to your dataset
# data <- read.csv("data_aging_congress.csv")
#
# # Step 2: Convert relevant columns to proper date formats
# # Convert the 'start_date' column to a Date object and extract the year
# data <- data %>%
#   mutate(start_year = year(as.Date(start_date)))
#
# # Step 3: Filter data for Senate and House
# # Keep only relevant columns and filter for chambers "Senate" and "House"
# filtered_data <- data %>%
#   select(start_year, chamber, age_years) %>%  # Select necessary columns
#   filter(chamber %in% c("Senate", "House"))  # Filter for Senate and House
#
# # Step 4: Calculate the median age for Senate and House by year
# median_age_data <- filtered_data %>%
#   group_by(start_year, chamber) %>%  # Group by year and chamber
#   summarize(median_age = median(age_years, na.rm = TRUE)) %>%  # Calculate median age
#   ungroup()
#
# # Step 5: Reshape the data for easier plotting
# # Spread the data to wide format with columns for Senate and House
# library(tidyr)
# wide_data <- median_age_data %>%
#   pivot_wider(names_from = chamber, values_from = median_age)
#
# # Step 6: Save the processed data to a CSV file
# write.csv(wide_data, "processed_congress_data.csv", row.names = FALSE)
#
# # Output a message indicating where the file was saved
# cat("Processed data saved to: processed_congress_data.csv\n")
```

```
# Load the processed data
data <- read.csv("processed_congress_data_time.csv")

# Plot the data
ggplot(data, aes(x = start_year)) +
  geom_line(aes(y = Senate, color = "Senate"), size = 1) +
  geom_line(aes(y = House, color = "House"), size = 1) +
  labs(
    title = "Median Age of the U.S. Senate and House Over Time",
    x = "Year",
    y = "Median Age",
    color = "Chamber"
  ) +
  scale_color_manual(values = c("Senate" = "purple", "House" = "green")) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title = element_text(size = 12),
```

```
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )
```

```
## 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.
```



Replicating 'Median age of the largest generation in each Congress, 1937 to 2023'

Data Preprocessing

Explanation of the Steps: Define Generations:

The classify_generation function assigns a generation based on the birth year. Extract Birth Year:

The birth_year column is created using the lubridate::year function to extract the year from the birthday column. Filter Data:

Filter out rows with start_year before 1937 and include only the four specified generations. Group and Summarize:

Use group_by and summarize to calculate the median age and count for each generation by year. Select Largest Generation:

For each start_year, use slice_max to select the generation with the largest count. Save Processed Data:

Save the filtered and summarized data to a CSV file, ready for plotting in R.

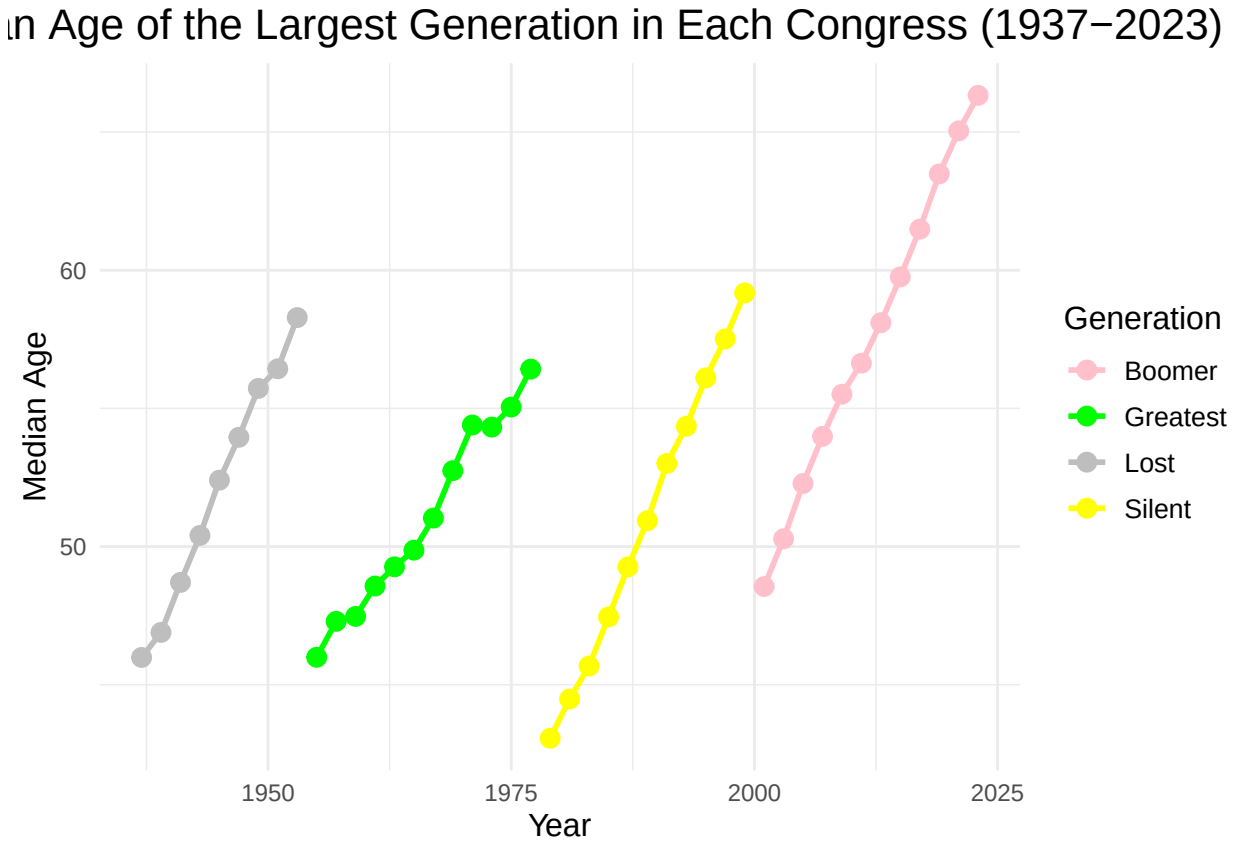
The cell below contains code that performs these preprocessing steps. However, we have commented it out and instead provide the processed .csv file for your convenience. Use the processed data "updated_largest_generation_data.csv" to generate the figure.

```
#
# # Step 1: Load the original dataset
# # Replace 'data_aging_congress.csv' with the correct path to your data file
# data <- read.csv("data_aging_congress.csv")
#
# # Step 2: Define generation categories based on the provided birth year ranges
# classify_generation <- function(birth_year) {
#   if (birth_year >= 1883 & birth_year <= 1900) {
#     return("Lost")
#   } else if (birth_year >= 1901 & birth_year <= 1927) {
#     return("Greatest")
#   } else if (birth_year >= 1928 & birth_year <= 1945) {
#     return("Silent")
#   } else if (birth_year >= 1946 & birth_year <= 1964) {
#     return("Boomer")
#   } else {
#     return(NA)  # Exclude individuals outside these ranges
#   }
# }
#
# # Step 3: Extract birth year from the birthday column and classify generations
# data <- data %>%
#   mutate(
#     birth_year = year(as.Date(birthday)),  # Extract birth year from the 'birthday' column
#     generation = sapply(birth_year, classify_generation)  # Apply the classification function
#   )
#
# # Step 4: Filter the data to include only records from 1937 onward and specified generations
# filtered_data <- data %>%
#   filter(
#     start_year >= 1937,  # Keep only records starting from 1937
#     generation %in% c("Lost", "Greatest", "Silent", "Boomer")  # Include only the specified generatio
#   )
#
# # Step 5: Group by start_year and generation to calculate the median age and count
# generation_summary <- filtered_data %>%
#   group_by(start_year, generation) %>%
#   summarize(
#     median_age = median(age_years, na.rm = TRUE),  # Median age for each generation per year
#     count = n()  # Count of members in each generation per year
#   ) %>%
#   ungroup()
```

```r
#
# # Step 6: Select the largest generation for each year based on count
# largest_generation_per_year <- generation_summary %>%
#   group_by(start_year) %>%
#   slice_max(order_by = count, n = 1) %>%  # Select the generation with the highest count for each yea
#   ungroup()
#
# # Step 7: Save the processed data to a CSV file
# write.csv(
#   largest_generation_per_year %>% select(start_year, generation, median_age),
#   "updated_largest_generation_data.csv",
#   row.names = FALSE
# )
#
# # Output a message indicating where the file was saved
# cat("Processed data saved to: updated_largest_generation_data.csv\n")
```

```r
# Load the processed data
data <- read.csv("updated_largest_generation_data.csv")

# Plot the data
ggplot(data, aes(x = start_year, y = median_age, color = generation)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  labs(
    title = "Median Age of the Largest Generation in Each Congress (1937-2023)",
    x = "Year",
    y = "Median Age",
    color = "Generation"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  ) +
  scale_color_manual(values = c(
    "Lost" = "gray",
    "Greatest" = "green",
    "Silent" = "yellow",
    "Boomer" = "pink"
  ))
```


ın Age of the Largest Generation in Each Congress (1937−2023)
Median Age
60
50
1950
1975
2000
2025
Year
Generation
Boomer
Greatest
Lost
Silent