DA5020.A5.Hsiao-Yu.Peng

Hsiao-Yu Peng

2023-10-10

Q1

```
url <- "https://www.senate.gov/general/contact_information/senators_cfm.xml"

# Use RCurl to fetch the data from the URL
xml_data <- getURL(url)

# Parse the XML data
parsed_data <- xmlParse(xml_data)

# Convert the XML data into a data fram
senator_df <- xmlToDataFrame(parsed_data)

# Check the data frame information
dim(senator_df)
```

```
## [1] 101  13
```

```
glimpse(senator_df)
```

```
## Rows: 101
## Columns: 13
## $ member_full         <chr> "Baldwin (D-WI)", "Barrasso (R-WY)", "Bennet (D-CO~
## $ last_name           <chr> "Baldwin", "Barrasso", "Bennet", "Blackburn", "Blu~
## $ first_name          <chr> "Tammy", "John", "Michael F.", "Marsha", "Richard"~
## $ party               <chr> "D", "R", "D", "R", "D", "D", "R", "R", "R", "D", ~
## $ state               <chr> "WI", "WY", "CO", "TN", "CT", "NJ", "AR", "IN", "A~
## $ address             <chr> "141 Hart Senate Office Building Washington DC 205~
## $ phone               <chr> "(202) 224-5653", "(202) 224-6441", "(202) 224-585~
## $ email               <chr> "https://www.baldwin.senate.gov/feedback", "https:~
## $ website             <chr> "https://www.baldwin.senate.gov/", "https://www.ba~
## $ class               <chr> "Class I", "Class I", "Class III", "Class I", "Cla~
## $ bioguide_id         <chr> "B001230", "B001261", "B001267", "B001243", "B0012~
## $ leadership_position <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ text                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

```
summary(senator_df)
```

```
##  member_full         last_name          first_name          party
##  Length:101         Length:101         Length:101         Length:101
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##     state              address            phone              email
##  Length:101         Length:101         Length:101         Length:101
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##    website             class           bioguide_id        leadership_position
##  Length:101         Length:101         Length:101         Length:101
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##      text
##  Length:101
##  Class :character
##  Mode  :character
```

```
sum(is.na(senator_df))
```

```
## [1] 207
```

```
# head(senator_df)
```

The dataset has dimension 101x13, all variables are character type. The are some missing values in the dataset.

Q2

```
# Extract only the first names
pattern <- "\\s|,"
senator_df$first_name <- sapply(str_split(senator_df$first_name, pattern), function(x) x[1])

# Select first name, last name and the party
df <- senator_df %>%
  select(first_name, last_name, party)

head(df)
```

```
##   first_name  last_name party
## 1      Tammy    Baldwin     D
## 2       John   Barrasso     R
## 3    Michael     Bennet     D
## 4     Marsha  Blackburn     R
## 5    Richard Blumenthal     D
## 6       Cory     Booker     D
```

The data frame's variable "first_name" has been removed the middle initial or prefix. It extracts only first, last names and party.

Q3

```
senatorsByState <- function(state_abbreviation) {
  selected_senators <- senator_df %>%
    filter(state == state_abbreviation) %>%
    select(first_name, last_name, party) %>%
    mutate(party = str_replace_all(party, c("D" = "Democratic Party", "R" = "Republican Party", "I" = "

  # Check if there are senators from the selected state
  if (nrow(selected_senators) > 0) {
    # Initialize senator_info as NULL
    senator_info <- NULL

    # Iterate through selected senators
    for (i in 1:nrow(selected_senators)) {
      senator_name <- paste(selected_senators[i, "first_name"], selected_senators[i, "last_name"])
      senator_party <- selected_senators[i, "party"]
      # Concatenate each row to senator_info
      senator_info <- c(senator_info, paste(senator_name, senator_party, sep = ", "))
    }

    # Create a message with senator information by joining senator_info with ", "
    message <- paste("The senators for", state_abbreviation, "are:", toString(senator_info))

    # Display the message
    cat(message, "\n")
  } else {
    cat("No senators found for the specified state (", state_abbreviation, ")\n")
  }
}

# Example:
senatorsByState("MA")
```

```
## The senators for MA are: Edward Markey, Democratic Party, Elizabeth Warren, Democratic Party
```

Q4

```
# load data set
df <- read_csv("~/Desktop/2023Fall_Syllabus/DA5020/week5/Ratio Of Female To Male Youth Unemployment Rat
```

```
## New names:
## Rows: 263 Columns: 66
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (4): Country Name, Country Code, Indicator Name, Indicator Code dbl (30): 1991,
## 1992, 1993, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, ... lgl (32): 1960,
## 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ...
## 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.
## * ‘‘ -> ‘...66‘
```

```
head(df)
```

```
## # A tibble: 6 x 66
##   ‘Country Name‘ ‘Country Code‘ ‘Indicator Name‘  ‘Indicator Code‘ ‘1960‘ ‘1961‘
##   <chr>          <chr>          <chr>             <chr>            <lgl>  <lgl>
## 1 Aruba          ABW            Ratio of female ~ SL.UEM.1524.FM.~ NA     NA
## 2 Afghanistan    AFG            Ratio of female ~ SL.UEM.1524.FM.~ NA     NA
## 3 Angola         AGO            Ratio of female ~ SL.UEM.1524.FM.~ NA     NA
## 4 Albania        ALB            Ratio of female ~ SL.UEM.1524.FM.~ NA     NA
## 5 Andorra        AND            Ratio of female ~ SL.UEM.1524.FM.~ NA     NA
## 6 Arab World     ARB            Ratio of female ~ SL.UEM.1524.FM.~ NA     NA
## # i 60 more variables: ‘1962‘ <lgl>, ‘1963‘ <lgl>, ‘1964‘ <lgl>, ‘1965‘ <lgl>,
## #   ‘1966‘ <lgl>, ‘1967‘ <lgl>, ‘1968‘ <lgl>, ‘1969‘ <lgl>, ‘1970‘ <lgl>,
## #   ‘1971‘ <lgl>, ‘1972‘ <lgl>, ‘1973‘ <lgl>, ‘1974‘ <lgl>, ‘1975‘ <lgl>,
## #   ‘1976‘ <lgl>, ‘1977‘ <lgl>, ‘1978‘ <lgl>, ‘1979‘ <lgl>, ‘1980‘ <lgl>,
## #   ‘1981‘ <lgl>, ‘1982‘ <lgl>, ‘1983‘ <lgl>, ‘1984‘ <lgl>, ‘1985‘ <lgl>,
## #   ‘1986‘ <lgl>, ‘1987‘ <lgl>, ‘1988‘ <lgl>, ‘1989‘ <lgl>, ‘1990‘ <lgl>,
## #   ‘1991‘ <dbl>, ‘1992‘ <dbl>, ‘1993‘ <dbl>, ‘1994‘ <dbl>, ‘1995‘ <dbl>, ...
```

```
# Create country_name tibble
country_name <- df %>%
  select(`Country Name`, `Country Code`) %>%
  distinct() # remove duplicate rwows

print(country_name)
```

```
## # A tibble: 263 x 2
##    ‘Country Name‘       ‘Country Code‘
##    <chr>                <chr>
##  1 Aruba                ABW
##  2 Afghanistan          AFG
##  3 Angola               AGO
##  4 Albania              ALB
##  5 Andorra              AND
##  6 Arab World           ARB
##  7 United Arab Emirates ARE
##  8 Argentina            ARG
##  9 Armenia              ARM
## 10 American Samoa       ASM
## # i 253 more rows
```

The tibble named “country_name” has 263 rows and 2 columns.

```
# tidy the data frame by pivot_longer()
tidy_data <- df %>%
  pivot_longer(cols = -c("Country Name", "Country Code", "Indicator Name", "Indicator Code"), names_to =
  rename(country_code = `Country Code`)
```

```r
# Create indicator_data tibble
indicator_data <- tidy_data %>%
  select(country_code, year, value)

print(indicator_data)
```

```
## # A tibble: 16,306 x 3
##    country_code year  value
##    <chr>        <chr> <dbl>
##  1 ABW          1960     NA
##  2 ABW          1961     NA
##  3 ABW          1962     NA
##  4 ABW          1963     NA
##  5 ABW          1964     NA
##  6 ABW          1965     NA
##  7 ABW          1966     NA
##  8 ABW          1967     NA
##  9 ABW          1968     NA
## 10 ABW          1969     NA
## # i 16,296 more rows
```

The tibble named "indicator_data" has 16,306 rows and 3 columns.

Q5

```r
country_data <- read_csv("~/Desktop/2023Fall_Syllabus/DA5020/week5/Country Meta-Data.csv")
```

```
## New names:
## Rows: 263 Columns: 6
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (5): Country Code, Region, IncomeGroup, SpecialNotes, TableName lgl (1): ...6
## 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.
## * `` -> `...6`
```

```r
country_data <- country_data %>%
  rename(country_code = `Country Code`)

indicator_data_20 <- indicator_data %>%
  filter(year >= 2000 & !is.na(value)) %>%
  inner_join(country_data, by = "country_code")
```

We filter the last 20 year information and remove missing value in the "value" column.

```r
# Write a function to get unique 5 countries
get_unique_country_codes <- function(region) {
  indicator_data_20 %>%
    filter(Region == region) %>%
```

```
    select(country_code) %>%
    unique()
    # slice(1:5)
}

# Get unique country code from the continents
unique_asia <- get_unique_country_codes("South Asia")
unique_america <- get_unique_country_codes("North America")
unique_mdEast <- get_unique_country_codes("Middle East & North Africa")


print(unique_asia)
```

```
## # A tibble: 8 x 1
##   country_code
##   <chr>
## 1 AFG
## 2 BGD
## 3 BTN
## 4 IND
## 5 LKA
## 6 MDV
## 7 NPL
## 8 PAK
```

```
unique_asia <- c("AGF", "BGD", "BTN", "IND", "LKA")
print(unique_america)
```

```
## # A tibble: 2 x 1
##   country_code
##   <chr>
## 1 CAN
## 2 USA
```

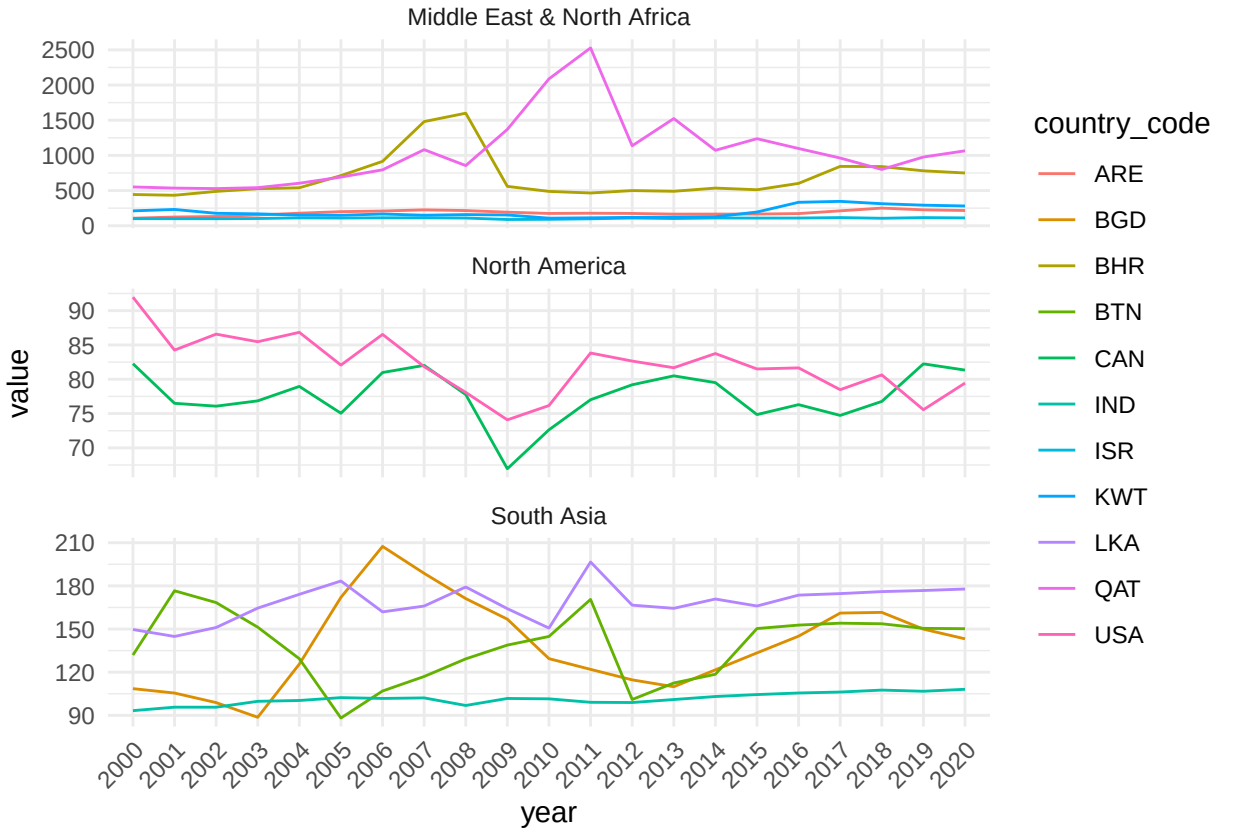
```
unique_america <- c("CAN", "USA")
print(unique_mdEast)
```

```
## # A tibble: 21 x 1
##    country_code
##    <chr>
##  1 ARE
##  2 BHR
##  3 DJI
##  4 DZA
##  5 EGY
##  6 IRN
##  7 IRQ
##  8 ISR
##  9 JOR
## 10 KWT
## # i 11 more rows
```

```r
unique_mdEast <- c("ARE", "BHR", "QAT", "KWT", "ISR")
```

```r
# Select countries
selected_countries <- indicator_data_20 %>%
  filter(country_code %in% c(unique_asia, unique_america, unique_mdEast))
```

```r
# Create the line plots
ggplot(selected_countries, aes(x = year, y= value, group = country_code, color = country_code)) +
  geom_line() +
  facet_wrap(~Region, ncol=1, scales="free_y") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



We compared the ratio of female to male youth unemployment rates in North America, South Asia, and the Middle East over the last 20 years. We found that the lowest ratio is in North America, ranging from 70 to 90. The second lowest ratio is in South Asia, which ranges from 90 to 210. The highest ratio is in the Middle East, ranging from 250 to 500, with QATAR (QAT) reaching even higher levels (2500) in 2011.

North America comprises three countries in the dataset: the USA, Canada (CAN), and Bermuda (BMU). However, Bermuda (BMU) lacks unemployment information, so we only display data for the USA and Canada in the line plots.

The Middle East exhibits a relatively high ratio of female to male youth unemployment rates, possibly influenced by cultural factors. In some Middle Eastern countries, women may face restrictions on working outside the home, which could explain the higher ratio compared to the other three regions.

We also conducted a survey of income group information within the dataset. North America falls into the high-income group category, whereas South Asia is categorized as a low-income group. High-income groups tend to have relatively lower ratios of female to male youth unemployment rates, which may indicate that higher income levels encourage women to participate in the workforce.