Working with XML Data

Rebecca Weiss

10/7/2021

```r
#load all necessary libraries
library(RCurl)
library(XML)
library(tidyverse)
library(stringr)
```

Clear the workspace:

```r
rm(list = ls())
```

1. Load the XML data directly from the URL below into a data frame (or tibble) and display the dimensions of the data.

```r
# load XML data into df
url <- getURL("https://www.senate.gov/general/contact_information/senators_cfm.xml")
senate_df <- xmlToDataFrame(url)

# display dims
dim(senate_df)
```

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

After loading the XML and putting into a dataframe, we see from running `dim(senate_df)` that there are 13 variables/columns and 101 rows/observations.

use this XML data to answer 2 and 3

2. Construct a regular expression (regex) to extract only the first and last names of each senator; the regex should remove their middle initial and/or suffix e.g. remove Jr. III, etc. Ensure that the updated names are reflected in the dataframe.

```r
# get rid of anything after first/last name and print to confirm
senate_df <- senate_df %>%  drop_na(first_name)
(senate_df$first_name <- sub("[[:space:]].*", "", senate_df$first_name))
```

```
##   [1] "Tammy"       "John"        "Michael"     "Marsha"      "Richard"
##   [6] "Roy"         "Cory"        "John"        "Mike"        "Sherrod"
##  [11] "Richard"     "Maria"       "Shelley"     "Benjamin"    "Thomas"
##  [16] "Robert"      "Bill"        "Susan"       "Christopher" "John"
##  [21] "Catherine"   "Tom"         "Kevin"       "Mike"        "Ted"
##  [26] "Steve"       "Tammy"       "Richard"     "Joni"        "Dianne"
##  [31] "Deb"         "Kirsten"     "Lindsey"     "Chuck"       "Bill"
##  [36] "Margaret"    "Josh"        "Martin"      "John"        "Mazie"
##  [41] "John"        "Cindy"       "James"       "Ron"         "Tim"
##  [46] "Mark"        "John"        "Angus"       "Amy"         "James"
##  [51] "Patrick"     "Mike"        "Ben"         "Cynthia"     "Joe,"
##  [56] "Edward"      "Roger"       "Mitch"       "Robert"      "Jeff"
##  [61] "Jerry"       "Lisa"        "Christopher" "Patty"       "Jon"
##  [66] "Alex"        "Rand"        "Gary"        "Rob"         "Jack"
##  [71] "James"       "Mitt"        "Jacky"       "Mike"        "Marco"
##  [76] "Bernard"     "Ben"         "Brian"       "Charles"     "Rick"
##  [81] "Tim"         "Jeanne"      "Richard"     "Kyrsten"     "Tina"
##  [86] "Debbie"      "Dan"         "Jon"         "John"        "Thom"
##  [91] "Patrick"     "Tommy"       "Chris"       "Mark"        "Raphael"
##  [96] "Elizabeth"   "Sheldon"     "Roger"       "Ron"         "Todd"
```

```
(senate_df$last_name <-sub("[[:space:]].*", "", senate_df$last_name))
```

```
##   [1] "Baldwin"      "Barrasso"     "Bennet"       "Blackburn"    "Blumenthal"
##   [6] "Blunt"        "Booker"       "Boozman"      "Braun"        "Brown"
##  [11] "Burr"         "Cantwell"     "Capito"       "Cardin"       "Carper"
##  [16] "Casey"        "Cassidy"      "Collins"      "Coons"        "Cornyn"
##  [21] "Cortez"       "Cotton"       "Cramer"       "Crapo"        "Cruz"
##  [26] "Daines"       "Duckworth"    "Durbin"       "Ernst"        "Feinstein"
##  [31] "Fischer"      "Gillibrand"   "Graham"       "Grassley"     "Hagerty"
##  [36] "Hassan"       "Hawley"       "Heinrich"     "Hickenlooper" "Hirono"
##  [41] "Hoeven"       "Hyde-Smith"   "Inhofe"       "Johnson"      "Kaine"
##  [46] "Kelly"        "Kennedy"      "King"         "Klobuchar"    "Lankford"
##  [51] "Leahy"        "Lee"          "LujÃ¡n"       "Lummis"       "Manchin"
##  [56] "Markey"       "Marshall"     "McConnell"    "Menendez"     "Merkley"
##  [61] "Moran"        "Murkowski"    "Murphy"       "Murray"       "Ossoff"
##  [66] "Padilla"      "Paul"         "Peters"       "Portman"      "Reed"
##  [71] "Risch"        "Romney"       "Rosen"        "Rounds"       "Rubio"
##  [76] "Sanders"      "Sasse"        "Schatz"       "Schumer"      "Scott"
##  [81] "Scott"        "Shaheen"      "Shelby"       "Sinema"       "Smith"
##  [86] "Stabenow"     "Sullivan"     "Tester"       "Thune"        "Tillis"
##  [91] "Toomey"       "Tuberville"   "Van"          "Warner"       "Warnock"
##  [96] "Warren"       "Whitehouse"   "Wicker"       "Wyden"        "Young"
```

From question 1, we know that there is an NA value that we drop first. Then we modify the dataframe to get rid of any cases with space in them to get only first and last name.

3. Create a function called senatorsByState() which takes the two letter abbreviation for a US State as its input argument and displays the first name, last name and party of each senator for the selected state. For example, if the user enters 'MA' your function should display a message that is similar to the following: "The senators for MA are: Edward Markey, Democratic Party and Elizabeth Warren, Democratic Party

```
senatorsByState <- function(x)  {
  df <- senate_df %>%
    select(state, first_name, last_name, party) %>%
    filter(state == x)

  print(paste0("The senators for ", df$state[1], " are: ",
           df$first_name[1], " ",df$last_name[1], ", " ,df$party[1],
           " and ", df$first_name[2], " ", df$last_name[2], ", " ,df$party[2]))
}

# test function
senatorsByState("MA")
```

```
## [1] "The senators for MA are: Edward Markey, D and Elizabeth Warren, D"
```

We created a function that selects the rows needed, filters so state is == to input, and prints the information as requested. We see that when we run `senatorsByState("MA")`, the function satisfied our goal.

Answer the questions below using the attached dataset on the Ratio Of Female To Male Youth Unemployment Rate.

4. Download the attached csv file from Canvas and load it in your R environment. Perform steps to tidy the data and the prepared data should be divided across two tibbles named country_name and indicator_data. The country_name tibble should contain the country name and country code (ensure that you remove duplicates), and the indicator_data tibble should include the country_code, year, and value. Note: Tidy the data using pivot_longer(), pivot_wider() and separate(), where applicable.

```
# load entire data set and skip first 3 rows where there is no data of use:
unemp_df <- read_csv("Ratio Of Female To Male Youth Unemployment Rate .csv", skip = 3)
```

```
## Warning: Missing column names filled in: 'X66' [66]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_logical(),
##   `Country Name` = col_character(),
##   `Country Code` = col_character(),
##   `Indicator Name` = col_character(),
##   ‘Indicator Code‘ = col_character(),
##   ‘1991‘ = col_double(),
##   ‘1992‘ = col_double(),
##   ‘1993‘ = col_double(),
##   ‘1994‘ = col_double(),
##   ‘1995‘ = col_double(),
##   ‘1996‘ = col_double(),
##   ‘1997‘ = col_double(),
##   ‘1998‘ = col_double(),
##   ‘1999‘ = col_double(),
##   ‘2000‘ = col_double(),
##   ‘2001‘ = col_double(),
##   ‘2002‘ = col_double(),
##   ‘2003‘ = col_double(),
##   ‘2004‘ = col_double(),
##   ‘2005‘ = col_double(),
##   ‘2006‘ = col_double()
##   # ... with 14 more columns
## )
## i Use ‘spec()‘ for the full column specifications.
```

```
# create tibble for country_name, remove duplicates, view:
(country_name <- as_tibble(unemp_df) %>%
  select(c("Country Name", "Country Code")) %>%
  distinct())
```

```
## # 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
## # ... with 253 more rows
```

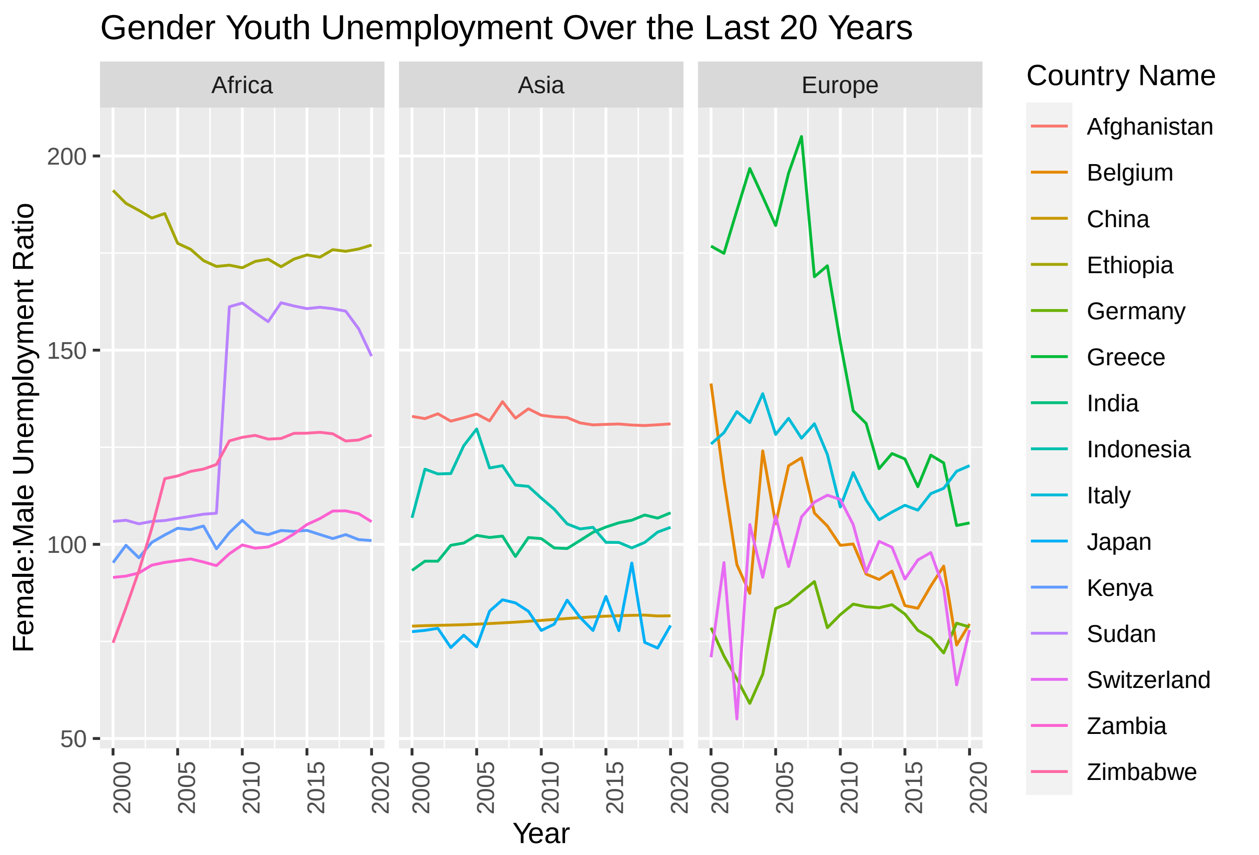
```
# create tibble for indicator_data, tidy data as needed, view:
(indicator_data <- as_tibble(unemp_df) %>%
      pivot_longer(cols = 5:65,
               names_to = "year",
                values_to = "value",
                values_drop_na = TRUE) %>%
  select(c("Country Code", "year", "value")))
```

```
## # A tibble: 6,990 x 3
##    ‘Country Code‘ year  value
##    <chr>          <chr> <dbl>
##  1 AFG            1991   131.
##  2 AFG            1992   131.
##  3 AFG            1993   130.
##  4 AFG            1994   131.
##  5 AFG            1995   133.
##  6 AFG            1996   131.
##  7 AFG            1997   133.
##  8 AFG            1998   133.
##  9 AFG            1999   133.
## 10 AFG            2000   133.
## # ... with 6,980 more rows
```

The data was loaded and tidied so that the years were pivoted into one variable, *year* and the unemployment ratio was filled in for their values, recoded as *value* in the dataframe. The indicator_data and country_name tibbles were then created and displayed based on the data asked for in the question. We see that there are 263 individual countries from the country_name data.

5. Select five countries from each of the following continents: Africa, Asia and Europe. Visualize their respective data from the last 20 years using a line chart; use facet_wrap to display the data for each continent in its own chart. Explain your observations about the ratio of female to male youth unemployment rate in the selected regions.

```r
# join indicator_data and country_name to combine country code from last 20 years
joined <- left_join(indicator_data, country_name, by = "Country Code") %>%
  filter(year > 1999)
joined$year <- as.integer(joined$year)

# select 5 countries from each continent and put into their own variable
afr <- joined[joined$`Country Name` %in% c(
  "Zambia", "Kenya", "Zimbabwe", "Ethiopia", "Sudan"), ] %>%
  mutate(Continent = "Africa")

eur <- joined[joined$`Country Name` %in% c(
  "Italy", "Greece", "Germany", "Belgium", "Switzerland"), ] %>%
  mutate(Continent = "Europe")

asia <- joined[joined$`Country Name` %in% c(
  "Japan", "Indonesia", "China", "India", "Afghanistan"), ] %>%
  mutate(Continent = "Asia")

# bind all data together to one dataframe
alldf <- rbind(afr, eur, asia)

# plot results
ggplot(data = alldf, aes(x=year, y=value, color = `Country Name`,
                         group = `Country Name`)) +
  geom_line() +
  scale_x_continuous(breaks = seq(2000, 2020, by = 5)) +
  theme(axis.text.x = element_text(angle=90)) +
  labs(title = "Gender Youth Unemployment Over the Last 20 Years",
       y = "Female:Male Unemployment Ratio",
       x = "Year") +
  facet_wrap(~Continent)
```



From the graph of this data separated by continent, we see many discrepancies by country in addition to by continent for ratio of female: male unemployment. For example, in Asia, while there remained different values by country, we see that the data is relatively flat over time compare to the other two continents, with Afghanistan remaining consistently highest. In Europe, we see many jumps over time,but what is most striking is that jump that occurs in Greece in the early 2000s, then drastic fall recently. In Africa we see that Ethiopia is consistently the highest, but there is a massive jump in the numbers ~2006 where the ratio went up and stayed higher.