Code Sample 1: Cleaning & Tidying Data

David I. Crabtree

2/19/2023

- Overview
- Importing FreedomHouse Data
- Cleaning & Tidying FreedomHouse Data
- Visualizing FreedomHouse Data

Overview

The FreedomHouse dataset is a common tool for researchers studying the strength of democracy around the world. However, the raw data provided on FreedomHouse's (https://freedomhouse.org/) website is obnoxiously messy, leaving researchers to do the work of cleaning and tidying the data before running analyses. Below I import FreedomHouse's data from their website and show how it can be cleaned using reproducible code. After cleaning the data, I build a few graphs to show background in ggplot.

A more extensive version of my work with FreedomHouse data, including 200+ procedurally generated country profiles, is available on my website (https://davidicrabtree.com/projects/democracy-reports/).

Importing FreedomHouse Data

Note that the FreedomHouse data directly downloaded from the FH website is very messy: the columns have empty variables ("X3, X4. . .") combined with years every third interval. The first row contains the names of the variables we are interested in. In the first row, "PR" represents an index of "Political Rights", "CL" represents an index of "Civil Liberties", and "Status" represents a categorical variable defining whether a country is "Free", "Partly Free", or "Not Free" according to FreedomHouse's coders. These variables need to eventually become the columns.

```
fh_url <- "https://freedomhouse.org/sites/default/files/2022-03/Country_and_Territory_Ra
tings_and_Statuses_FIW_1973-2022%20.xlsx"

fh <- read.xlsx(fh_url,
                sheet = 2,
                na.strings = "-",
                startRow = 2)


head(fh, c(5, 10))
```

```
##   Year(s).Under.Review 1972  X3     X4 1973  X6     X7 1974  X9    X10
## 1                 <NA>   PR CL  Status   PR CL  Status   PR CL  Status
## 2          Afghanistan    4  5      PF    7  6      NF    7  6      NF
## 3              Albania    7  7      NF    7  7      NF    7  7      NF
## 4              Algeria    6  6      NF    6  6      NF    6  6      NF
## 5              Andorra    4  3      PF    4  4      PF    4  4      PF
```

Cleaning & Tidying FreedomHouse Data

Note the pattern. The variables repeat every third column as follows:
PR, CL, Status, PR, CL, Status, PR, CL, Status
Also note that the year is currently where PR should be, X3 is where CL should be, X4 where status should be, and so on. Year also advances every third column

We can replicate the sequence that the variables repeat in the data with `seq()` . Note that below is the sequence for the CL variable.

```
head(seq(from = 2, to = 30135, by = 3))
```

```
## [1]  2  5  8 11 14 17
```

I start by renaming the first column to “country” and then pivoting so that all current columns except “country” become values for a new column called “Rate Type” and all current values in the selected columns become values for a new column called “Score”. I have noted the pattern that the important variables (PR, CL, and status) repeat with, so I remove the row with that data for now.

```
# Rename first column to "country"
names(fh)[1] <- "country"

fh_clean <- fh %>%
  filter(!is.na(country)) %>%
  pivot_longer(cols = 2:148,
               names_to = "RateType",
               values_to = "Score")

head(fh_clean)
```

```
## # A tibble: 6 x 3
##   country     RateType Score
##   <chr>       <chr>    <chr>
## 1 Afghanistan 1972     4
## 2 Afghanistan X3       5
## 3 Afghanistan X4       PF
## 4 Afghanistan 1973     7
## 5 Afghanistan X6       6
## 6 Afghanistan X7       NF
```

Next I access each variable’s proper location in the dataframe with the index operator `[]` and the `seq()` function.

```
fh_clean$RateType[seq(from = 1, to = 30135, by = 3)] = "pr"

fh_clean$RateType[seq(from = 2, to = 30135, by = 3)] = "cl"

fh_clean$RateType[seq(from = 3, to = 30135, by = 3)] = "status"
```

Now we have all scores attributed to their proper rating type, but we need to re-assign them to their proper years. Recall from earlier the data is in chronological order from 1975 to 2021 for all 205 countries, with the year advancing by 1 every third row.

```
head(fh_clean, 9)
```

```
## # A tibble: 9 x 3
##   country     RateType Score
##   <chr>       <chr>    <chr>
## 1 Afghanistan pr       4    
## 2 Afghanistan cl       5    
## 3 Afghanistan status   PF   
## 4 Afghanistan pr       7    
## 5 Afghanistan cl       6    
## 6 Afghanistan status   NF   
## 7 Afghanistan pr       7    
## 8 Afghanistan cl       6    
## 9 Afghanistan status   NF   
```

Repeat the years from 1973-2021 205 times, with each country having the same year three times. This will set it up to pivot back to wide format.

```
fh_clean <- fh_clean %>%
  mutate(year = rep(seq(from = 1973, to = 2021, by = 1), times = 205, each = 3))
```

Now all observations have a year attached to them corresponding to a given country's rating in a given year. There's still one last problem: the rating types should each be individual columns where the values in the "score" column are values for each rating type column. This is because rating types make more sense as variables than as values.

```
head(fh_clean)
```

```
## # A tibble: 6 x 4
##   country     RateType Score  year
##   <chr>       <chr>    <chr> <dbl>
## 1 Afghanistan pr       4      1973
## 2 Afghanistan cl       5      1973
## 3 Afghanistan status   PF     1973
## 4 Afghanistan pr       7      1974
## 5 Afghanistan cl       6      1974
## 6 Afghanistan status   NF     1974
```

```
# Tidy the dataframe: values in the current column "RateType" each become new columns wi
th values from the current column "Score".
fh_clean <- fh_clean %>%
  pivot_wider(names_from = "RateType",
              values_from = "Score")
```

The final result is dataframe where the unit of measurement is the country-year. The data is tidy, meaning each variable has its own column, each observation has its own row, each measurement has its own cell. The PR, CL, and Status variable include observations for every country-year in the data for a total of 30,135 country-year observations.

```
head(fh_clean)
```

```
## # A tibble: 6 x 5
##   country      year pr    cl    status
##   <chr>       <dbl> <chr> <chr> <chr>
## 1 Afghanistan  1973 4     5     PF
## 2 Afghanistan  1974 7     6     NF
## 3 Afghanistan  1975 7     6     NF
## 4 Afghanistan  1976 7     6     NF
## 5 Afghanistan  1977 7     6     NF
## 6 Afghanistan  1978 6     6     NF
```

```
tail(fh_clean)
```

```
## # A tibble: 6 x 5
##   country   year pr    cl    status
##   <chr>    <dbl> <chr> <chr> <chr>
## 1 Zimbabwe  2016 5     5     PF
## 2 Zimbabwe  2017 6     5     NF
## 3 Zimbabwe  2018 5     5     PF
## 4 Zimbabwe  2019 5     5     PF
## 5 Zimbabwe  2020 6     5     NF
## 6 Zimbabwe  2021 6     5     NF
```

```
head(unique(fh_clean$country))
```

```
## [1] "Afghanistan"         "Albania"             "Algeria"
## [4] "Andorra"             "Angola"              "Antigua and Barbuda"
```

```
tail(unique(fh_clean$country))
```

```
## [1] "Yemen"      "Yemen, N."  "Yemen, S."  "Yugoslavia" "Zambia"
## [6] "Zimbabwe"
```

Lastly, I recode the ratings so that lower scores correspond to less political rights and less civil liberties, respectively.

```
fh_clean$pr <- as.numeric(fh_clean$pr)
fh_clean$cl <- as.numeric(fh_clean$cl)

for (i in seq_along(fh_clean$pr)) {
  switch(fh_clean$pr[i],
         fh_clean$pr[i] <- 7,
         fh_clean$pr[i] <- 6,
         fh_clean$pr[i] <- 5,
         fh_clean$pr[i] <- 4,
         fh_clean$pr[i] <- 3,
         fh_clean$pr[i] <- 2,
         fh_clean$pr[i] <- 1)
}

for (i in seq_along(fh_clean$cl)) {
  switch(fh_clean$cl[i],
         fh_clean$cl[i] <- 7,
         fh_clean$cl[i] <- 6,
         fh_clean$cl[i] <- 5,
         fh_clean$cl[i] <- 4,
         fh_clean$cl[i] <- 3,
         fh_clean$cl[i] <- 2,
         fh_clean$cl[i] <- 1)
}

# Showing reverse coding worked
head(fh_clean)
```

```
## # A tibble: 6 x 5
##   country      year    pr    cl status
##   <chr>       <dbl> <dbl> <dbl> <chr>
## 1 Afghanistan  1973     4     3 PF
## 2 Afghanistan  1974     1     2 NF
## 3 Afghanistan  1975     1     2 NF
## 4 Afghanistan  1976     1     2 NF
## 5 Afghanistan  1977     1     2 NF
## 6 Afghanistan  1978     2     2 NF
```

Visualizing FreedomHouse Data

I visualize some relationships in the data below. First I import some data on each country’s continent and geographical data, since this is not included in FreedomHouse data. I then merge the continent data with FreedomHouse based on country name.

```
continents <- read_csv("https://raw.githubusercontent.com/davidicrabtree/democracy-repor
ts/main/data/continents.csv")
```

```
## Rows: 249 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): name, alpha-2, alpha-3, iso_3166-2, region, sub-region, intermediat...
## dbl (4): country-code, region-code, sub-region-code, intermediate-region-code
##
## 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.
```

```
fh_clean <- continents %>%
  rename("country" = "name",
         continent = region,
         region = `sub-region`) %>%
  select(country, continent, region) %>%
  right_join(fh_clean, by = "country")

head(fh_clean)
```

```
## # A tibble: 6 x 7
##   country     continent region          year    pr    cl status
##   <chr>       <chr>     <chr>          <dbl> <dbl> <dbl> <chr>
## 1 Afghanistan Asia      Southern Asia  1973     4     3 PF
## 2 Afghanistan Asia      Southern Asia  1974     1     2 NF
## 3 Afghanistan Asia      Southern Asia  1975     1     2 NF
## 4 Afghanistan Asia      Southern Asia  1976     1     2 NF
## 5 Afghanistan Asia      Southern Asia  1977     1     2 NF
## 6 Afghanistan Asia      Southern Asia  1978     2     2 NF
```

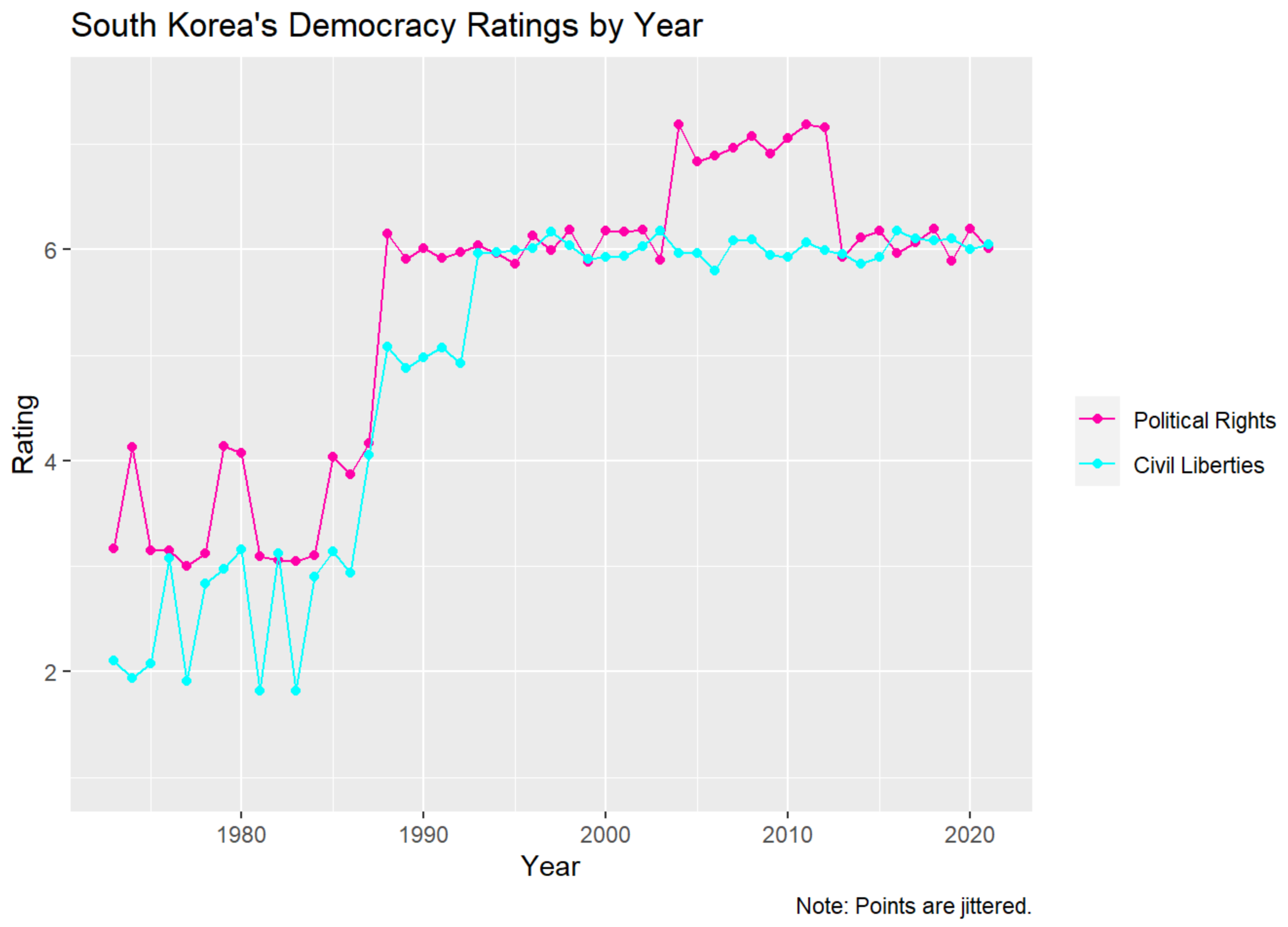
Now I create variables used in the rest of the ggplot graphs. This means if I want to re-create the graphics for different countries, all I have to do is change the country name stored below.

```
# Paramaterizing Code -- To see the below graphs for different countries, just enter a d
ifferent country name.

given_country <- "South Korea"
country_data <- fh_clean %>%
  filter(country == given_country)
most_recent <- country_data %>%
  drop_na(status) %>%
  summarize(max_year = max(year)) %>%
  pull()
country_continent <- country_data %>%
  filter(year == most_recent) %>%
  pull(continent)
```

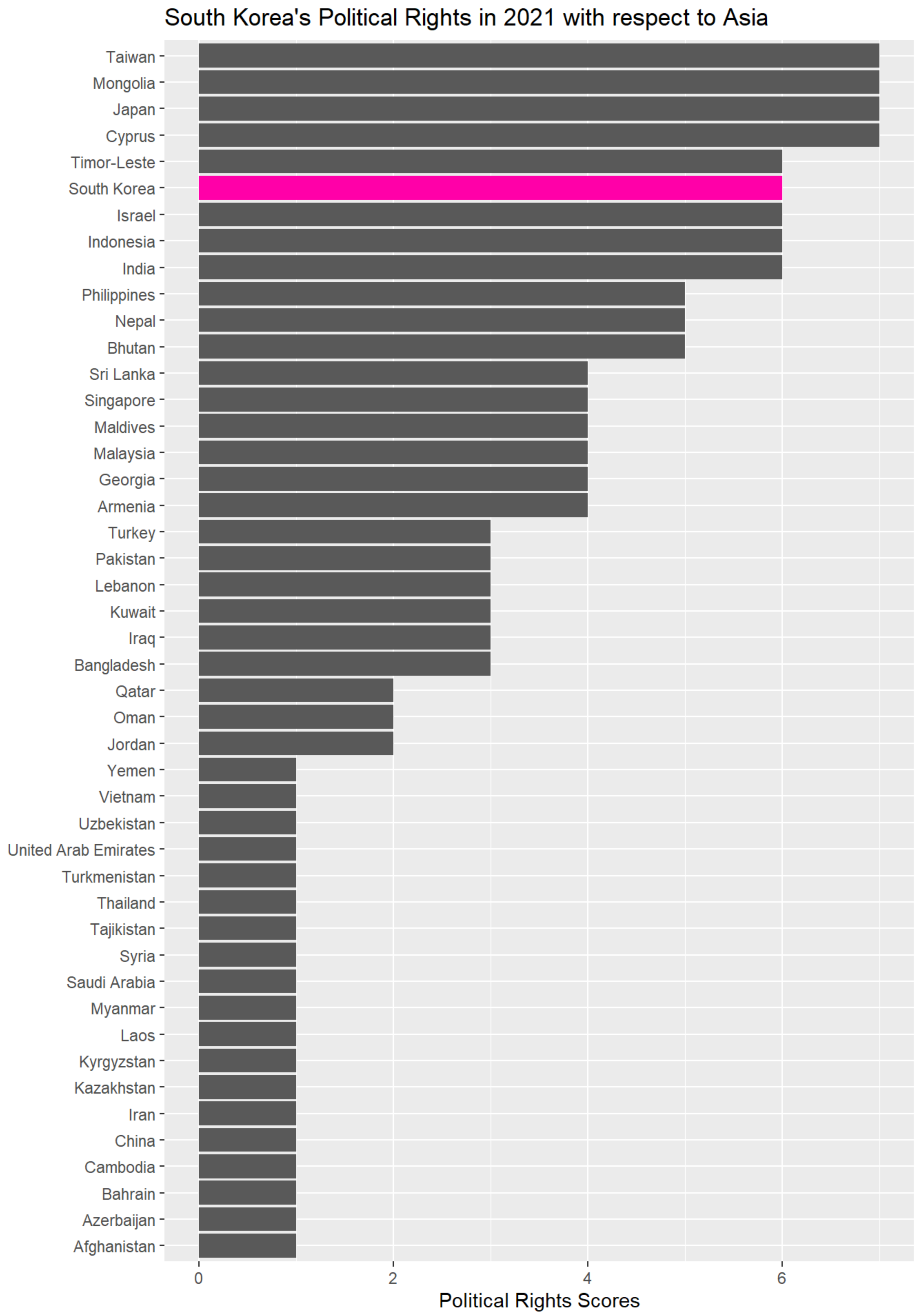
We can view how countries' political rights and civil liberties co-vary over time.

```r
country_data %>%
  drop_na(pr) %>%
  ggplot(aes(x = year)) +
  geom_point(aes(y = pr, color = "Political Rights"),
             position = position_jitter(width = 0, height = 0.2, seed = 72)) +
  geom_path(aes(y = pr, color = "Political Rights"),
            position = position_jitter(width = 0, height = 0.2, seed = 72)) +
  geom_point(aes(y = cl, color = "Civil Liberties"),
             position = position_jitter(width = 0, height = 0.2, seed = 60)) +
  geom_path(aes(y = cl, color = "Civil Liberties"),
            position = position_jitter(width = 0, height = 0.2, seed = 60)) +
  scale_y_continuous(limits = c(1, 7.5)) +
  scale_color_manual(breaks = c("Political Rights", "Civil Liberties"),
                     values=c("#FF00A8", "#00ffff")) +
  labs(
    title = glue("{given_country}'s Democracy Ratings by Year"),
    caption = "Note: Points are jittered.",
    x = "Year",
    y = "Rating",
    color = NULL
  )
```



We can also see how a given country compares to other countries in their continent on political rights.

```r
fh_clean %>%
  filter(year == most_recent,
         continent == country_continent) %>%
  ggplot(aes(x = fct_reorder(country, pr), y = pr)) +
  geom_col() +
  geom_col(data = country_data %>% filter(year == most_recent), fill = "#FF00A8") +
  coord_flip() +
  labs(
    title = glue("{given_country}'s Political Rights in {most_recent} with respect to {c
ountry_continent}"),
    x = NULL,
    y = "Political Rights Scores"
  )
```


South Korea's Political Rights in 2021 with respect to Asia
Taiwan
Mongolia
Japan
Cyprus
Timor-Leste
South Korea
Israel
Indonesia
India
Philippines
Nepal
Bhutan
Sri Lanka
Singapore
Maldives
Malaysia
Georgia
Armenia
Turkey
Pakistan
Lebanon
Kuwait
Iraq
Bangladesh
Qatar
Oman
Jordan
Yemen
Vietnam
Uzbekistan
United Arab Emirates
Turkmenistan
Thailand
Tajikistan
Syria
Saudi Arabia
Myanmar
Laos
Kyrgyzstan
Kazakhstan
Iran
China
Cambodia
Bahrain
Azerbaijan
Afghanistan
0
2
4
6
Political Rights Scores