SPEC Lab R Resources: Powersharing Groupwork for Data Management I

Alix Ziff, Benjamin A.T. Graham, Jasmine Chu, Annie Street, and Yuchen Gong

Summer 2022

Data Investigations: Reserved Government Positions

This groupwork uses the "IDC_training_2021.rds" data file. This is a country-year dataset gathered in the SPEC Lab. The data are already tidy. Each row is a unique observation identified by country code (gwno) and year; each column is a variable with information about each country's political institutions.

Getting Started

Write the header for your R script and save the script to your personal R script folder.

Note: Annotate the heck out of your code as you complete these exercises. LOTS of notes to yourself. A `#` at the start of a line tells R that what you are writing is a comment, not a line of code.

Set your working directory to the Training Data folder you have downloaded.

Use the `library()` function to load in the dplyr package.

```r
library(tidyverse)
library(readr)
```

Exercise 1.1: Load and view the dataset. This is a .rds file so you will want to use `readRDS`.

```r
# Note on this and future answer keys code: There are many ways to do things in R
    # so this is NOT the only correct way you can do things.
    # You may have written different code that works great!
    # This is ONE correct way to do things.

dt <- readRDS("./IDC_training_2022.rds")
# This code saves the object as "dt".
# You can name your data object whatever you like, but "dt" is nice and simple.
```

Exercise 1.2: What is the second variable in the dataset, reading from left to right? What is the eighth?

FYI

When you use the `select()` function, you will keep only the variables you mention. You can also use `rename()` to rename a variable without dropping all other variables.

There are several additional helper functions you can use inside `select()` such as `starts_with()`, `ends_with()`, `num_range()`, `matches()`.

You can use the `filter()` command to find rows/ cases based on certain conditions.

The `mutate()` function adds new variables while preserving old ones. When you use `mutate()`, make sure to make a unique variable name so you do not overwrite existing variables. You can use `transmute()` to add new variables and drop existing ones.

For this problem set, you will use piping with the notation `%>%` to make your code more efficient.

We will be focusing on variables relating to religion in the power sharing dataset. These variables have the prefix "rel" and include state establishment of religion, state restriction of religion, protection of religion, and protection against discrimination on the basis of religion.

DIY

Selecting Variables and Observations

Exercise 2.1: Select only the variables related to government positions reserved for minority groups and those related to subnational policy authority: "resseats": stands for Reserved seats (Mandated) "resman": stands for Reserved Executive Positions (Mandated) "resimp": stands for Reserved Executive Positions (Implemented) "subed": stands for Subnational Education Authority "subtax": stands for Subnational Tax Authority "subpolice": stands for Subnational Tax Authority and then also retain the three variables "gwno", "country", and "year".

Helpful Hint: Try using the `contains()` or `starts_with()` helper function to grab the reserved positions and subnational policy authority variables. Or you can use the concatenate function `c()` to create a vector, aka a list, of the variables you want to grab and then use the `all_of()` helper function to grab all the items in the vector.

```
vars_1 <- c("gwno", "country", "year", "resseats_IDC", "resman_IDC",
             "resimp_IDC", "subed_IDC", "subtax_IDC", "subpolice_IDC")
```

We separate each item in the vector with commas and we put quotes around the items in our vector since the items are character strings, not numbers. We have named this vector "vars_1", which is arbitrary. You can name yours whatever we want. I put gwno, country, and year first in the list so they show up on the left of my new tibble – gwno and year are my unit identifiers and country helps me understand what the gwno means – so I like to have them at the left when I look at my data.

```
# Option A by using select(all_of())
dt_revised <- dt %>%
  select(all_of(vars_1))

# Option B by using select(matches())
dt_vars <- dt %>%
  select("gwno", "country", "year", matches('sub|res'))%>%
  select(!resseatsimp_IDC)

# Option C by using select(starts_with))
dt_vars <- dt %>%
  select("gwno", "country", "year", starts_with(c("res", "sub"))) %>%
  select(!resseatsimp_IDC)
```

```r
# Option D by using select(contains))
dt_vars <- dt %>%
  select("gwno", "country", "year", contains(c("res", "sub"))) %>%
  select(!resseatsimp_IDC)

#Option E if we are allergic to helper functions
dt_revised2 <- dt %>%
  select(gwno, country, year, resseats_IDC, resman_IDC, resimp_IDC,
         subed_IDC, subtax_IDC, subpolice_IDC)
```

Exercise 3.1: using the `filter()` function, create a smaller dataframe that keeps only cases where subnational governments have either subnational tax authority *or* subnational education authority (i.e. subtax or subed takes a value of 1).

```r
df_sub <- dt %>%
    filter(subtax_IDC==1| subed_IDC==1) # filter for when subtax or
                                        # subed takes a value of 1.
```

Exercise 3.2: Name a country-year in which more than 20% of the legislative seats are reserved for members of minority groups (i.e. resseats is greater than 0.2).

```r
dt_resseats <- dt %>%
  filter(resseats_IDC>0.2) # filter for when resseats is greater than 0.2

View(dt_resseats)
# In the year 2012, the country Barbados saw more than 20% of the legislative
# seats reserved for members of minority groups.

#You could also just look for an example using: View(dt_resseats)
```

State Protection of Religion

Exercise 4.1 : Use the `mutate()` function to create a binary variable named "resseats_10" capturing whether at least 10% of legislative seats are reserved for minority groups.

```r
dt_revised <- dt_revised %>%
  mutate(resseats_10 = resseats_IDC>=0.1) #create binary variable through mutate
```

Exercise 5.1 We want to know whether reserved executive positions (resimp) have been getting more common over time. Use the `group_by()` and `summarise()` functions together to calculate the global proportion of countries that have this type of protection (i.e. the global average of resimp) by year. Is this type of provision present in a higher percentage of countries in 2010 or in 2019?

Note that you may have to remove missing values coded as NA or -44.

```r
dt_revised <- dt_revised %>%
  group_by(year)%>% #group by year
  select(resimp_IDC, year)%>%
  # take out -44 and NA values
  filter(resimp_IDC != -44 & resimp_IDC != "NA")%>%
  # find global average of resimp
  summarise(mean_resimp= mean(resimp_IDC, na.rm=TRUE))

# This type of provision present in a higher percentage of countries in 2010
```

Last thing: Put it all together. Use piping to combine exercises 1, 2, 4, and 5 into a single, elegant command. This means we don't filter subtax, subed, or resseats in this last exercise.

```
final <- dt%>%
  select("resseats_IDC", "resman_IDC", "resimp_IDC",
         "subed_IDC", "subtax_IDC", "subpolice_IDC", "gwno",
         "country", "year")%>%
  # create binary variable through mutate
  mutate(resseats_10= resseats_IDC>=0.1)%>%
  group_by(year)%>% # group by year
  # take out -44 and NA values
  filter(resimp_IDC != -44 & resimp_IDC != "NA")%>%
  # find global average of resimp
  summarise(mean_resimp= mean(resimp_IDC, na.rm=TRUE))
```