Democracy and Redistribution

Instructions

A long-standing debate in the social sciences is whether democracies redistribute more to the poor than autocracies. Research on this topic is challenging, however, due to the prevalence of missing data. Information about particular countries (and variables) are often missing, and this absence of data is not random. For example, autocratic countries are less likely to report their data to international institutions like the World Bank. Also, starting in the 1990s, countries have become better at both collecting and reporting data on different indicators such as economic growth or infant mortality. So if we just analyze data without taking these factors into account, we might bias our results. This exercise is based on the following prominent paper:

Ross, Michael (2006), "Is Democracy Good for the Poor", *American Journal of Political Science*, Vol. 50, No. 4, pp. 860 - 874.

Prior to Ross' work, the prevailing belief was that democracies redistributed to the poor more than autocracies. Ross's work challenged this belief. Specifically, Ross argued that previous studies had paid insufficient attention to differences between countries and time trends. Further, Ross argued that their analysis did not address the problem of missing data.

Below you will find a dictionary with the main variables in two datasets we analyze:

World Bank: `world_bank.csv`

Name	Description
`country_name`	Country name.
`country_code`	Country abbreviation.
`year`	Year.
`gdp_growth`	GDP growth rate (percentage).
`gdp_per_capita`	GDP per capita (2000 US$).
`inf_mort`	Infant mortality (deaths per 1000 children under 5).
`pop_density`	Population density (per sq. km).

Polity IV: `polity.csv`

Name	Description
`scode`	Country abbreviation.
`year`	Year.
`polity`	Polity Score. Ranges from -10 (most autocratic) to 10 (most democratic)

Question 1: Wide to Long Data

Before we do anything with these two datasets, you'll notice that the `polity.csv` data is in wide format rather than long. It has columns that correspond to countries and the values of those columns are the polity for that country in a particular year. As is often the case, we want this data in long format, where the country is recorded in its own column rather than across many different columns. Read in the data and use `pivot_longer()` to turn it into long format.

Question 2: Joining

Now that we have two long datasets, let's join them together. Two of the most common types of joins are `left_join()`, which keeps all rows in the "left-hand" data regardless of whether there is a match in the data being joined (the "right-hand" data), and `inner_join()`, which keeps only the rows that are present in both datasets.

Read both datasets into R and inspect them (you probably already have `polity.csv` read into R from the last question). Before attempting either type of join, decide which type you think you should use and why, supposing that the World Bank data is the "left-hand" dataset.

Question 3: Joining Continued

Perform your chosen join and inspect the resulting data. How many rows does it have? Do you notice any missing data? Do you think you chose the right join?

Question 4: Scatterplot

Now that we have joined the two datasets together, we can examine the relationship between variables in them. We want to know how GDP per capita is related to Polity score. Create a scatterplot with Polity score on the x-axis and **log** GDP per capita on the y-axis. Can you tell what the relationship is between these two variables just by looking?

Question 5: Correlation

Sometimes it's difficult to use plots to learn the relationship between two variables visually. Luckily, we can calculate a correlation! Calculate the correlation between Polity and **log** GDP per capita and interpret your result.

Question 6: Z-Score Function

Next, let's write a function that will take a column of our data and return the standardized version of the column. Recall the formula for the Z-score for some variable x:

$$Z = \frac{x - \text{mean}(x)}{\text{sd}(x)}$$

You can also use the following template for the general composition of a user-defined function. You get to decide how many arguments (also called inputs), so you may not need the same number as are in the template. Make sure you include the `return(...)` statement to tell your function what you want to output.

```
my_function <- function(arg1, arg2, arg3, arg4, ...) {
  result <- ...do something...
  return(result)
}
```

Once your create your function, use it on the `polity` variable and check to see that the standardized variable has mean 0 and standard deviation 1.