EES2019 Stacking Process

Generic Dichotomous Variables Estimation

Giuseppe Carteny

31.08.2021

Contents

1 Introduction

Our next tasks consist in (1) estimating a set of generic variables and (2) creating the codebook of the EES 2019 voter study SDM. We will start with the generic variables estimation, then we will start also preparing the codebook, and for a while the two tasks will run parallel.

As you might remember from our first call, **generic variables** concern the relationship between each voter and each relevant party in a given party system. Such variables can be categorized according to different criteria, but in our case we will distinguish them according to the routine for creating them.

Among the EES 2019 voter study variables previously stacked, only the propensity to vote (PTV) variable (`Q10`; See the EES2019 Codebook) can be reshaped in a long format without additional changes. All the others need more or less straightforward procedures for being estimated.

In the following lines I will show you how to compute a particular kind of generic variables: namely *dichotomous* generic variables. Then I will ask to one of you to prepare the codebook for the variables that are going to be estimated.

The **workflow** will be based, again, on a series of country-specific scripts. As mentioned earlier, this is probably not the most efficient procedure, but will help us to keep under control all the critical steps, perhaps allowing us in the future to create a more succint workflow. As done previously, I provide you an exemplary script, again based on the EES 2019 Italian voter study data.

2 Estimating Generic Dichotomous Variables and Preparing the Codebook

The procedure for estimating generic dichotomous variables is rather straightforward, and can be applied to all the variables that are listed below (see Sect. 2.), all relying on values that refer to (at least some of) the parties in the `party` variable that we created when stacking the EES 2019 observations. There are few cases in which such procedure should be slightly different, but overall we will use an auxiliary function, named `gendic.fun`, that I coded for this task.

2.1 The `gendic.fun` function

Details about this function can be found in the 'EES2019_stack_funs.R' script, in the 'Scripts' subdirectory, that includes also functions drafted for other tasks but that are still under revision. In short, the `gendic.fun` takes the values of a specific variable (say, `Q7`, the vote choice variable) and compares them with the `party` variable. When the value of the target variable equals the party code then the function assigns value `1` to the generic variable, and `0` otherwise. Missing values or other specific values, then, are coded according the values assigned in the original EES 2019 variable that has been used. The output of the function is a data frame with one column, namely the generic variable.

2.2 Procedure for computing generic dichotomous variables

For developing your scripts you just have to run the 'EES2019_stack.R' script. After this passage we can start working on our script(s).

As usual we first filter the main datasets according to the country that we are interested in:

```
cntry = 'IT'

EES2019_it <- EES2019 %>% filter(countryshort==cntry)
EES2019_stckd_it <- EES2019_stckd %>% filter(countryshort==cntry)
EES2019_cdbk_it <- EES2019_cdbk %>% filter(countryshort==cntry)

rm(cntry)
```

Then, check the values of the variable of interest. All the categorical variables that we are going to use should have approximately three values smaller than 100 (0, 96, 98) and then a limited set of values referring to the parties of the party system under investigation. If there are values that are not in this pattern then you should check carefully and perhaps send me an email in order to evaluate together what to do.

Once we checked the variables, then we can apply the function as shown below:

```
EES2019_it_stack <-
  cbind(EES2019_stckd_it,
        gendic.fun(data = EES2019_stckd_it,
                   var = 'Q7',
                   stack_var = 'party')) %>%
  as_tibble()
```

Of course, we can add more variables to the chunk above, just copy-pasting the `gendic.fun` function and changing the argument `var`. Alternatively, we can apply such function to a vector containing the variables of interest, then binding the results, and then the SDM[1].

Returning to our tutorial, once finished the list of variables then we just have to clean the environment.

2.3 Codebook development

Once finished the estimation of the variable(s) of interest, then the next step is to insert a new entry in the SDM codebook. You can find the R Markdown script on which we will work ('Codebook.Rmd') in the '~/EESstacked/Docs/docs_scripts' subdirectory.

The entry should look, more or less, as the following one, that comes from the codebook I sent you a few days ago:

```
## Q7_gen {-}
Dichotomous vote choice variable, measuring whether the respondent voted for the stack
party (`party`). Recoded from the EES 2019 original vote choice variable (`votech`).

Values:

0 - Did not vote for the stack party
1 - Did vote for the stack party

99 - Missing
```

[1] In my case, when I will code my scripts I guess that I will resort to the latter solution. But in any case, this is up to you.

3 Which Variables

The list of variables to be transformed in dichotomous generic variables is the following:

- `Q2`: Which party the respondent believes would be best at dealing with the problem specified in `Q1`;
- `Q7`: Which party the respondent (recalls to have) voted for at the 2019 European Parliament elections;
- `Q9_rec`: Which party the respondent (recalls to have) voted for at the last (national) general election. This variable consists in a recoded version of the original EES 2019 `Q9` variable;
- `Q25_rec`: Which party the respondent feels close to (aka party identification). This variable consists in a recoded version of the original EES 2019 `Q25` variable.

4 Who Does What

I will take care of Belgium, Bulgaria, Cyprus, and Italy, while you will take care of the following countries:

- **Willie**: Denmark, Estonia, Germany, Luxembourg, Malta, Netherlands, Spain, United Kingdom;
- **Julian**: Czech Rep., Finland, Greece, Hungary, Lithuania, Slovakia, Poland, Sweden;
- **Matthias**: Austria, Croatia, France, Ireland, Latvia, Portugal, Romania, Slovenia.

Since the development of the codebook cannot be divided as the tasks aforementioned, then you will take turns. This time **Julian** will fill the codebook with a set of entries referring to the generic variables generated from the variables mentioned above.

5 The Deadline

I believe that a few days should be enough to finish the tasks listed above (deadline: **06.09.2021**).