EES2019 Stacking Process

First steps tutorial

Giuseppe Carteny

02.08.2021

Contents

1 Introduction

This document is a short guide introducing the first steps of your job as student assistants for the ProConEU project. Read it carefully and send a feedback about any additional information you might need.

Some of the information about the Git repository, the workflow, and the scripts might change during the following weeks.

If you have *any* doubt, issue, comment, suggestion, feel free to contact me via email.

2 The Git/GitHub Workflow

2.1 The 'EESstacked' Repository Structure

The repository that we will work with is named 'EESstacked'. This repo is constituted by three main directories, and a set of subdirectories for each of the main ones. The current structure is the following:

- **Data**: A folder containing the EES 2019 voter study data, as well as additiona/auxiliary data sets;
  - *EES2019*: This folder contains the EES 2019 voter study data, as well as the EES 2019 codebook;
  - *EP2019*: Folder containing the 2019 European elections results.
- **Scripts**: Folder containing all the scripts that we will work with, except those for creating the documentation. The main scripts will be stored in this folder. The scripts sourced by the latter are split in two subdirectories:
  - *aux_data_scripts*: Folder containing R scripts for loading and mutating the auxiliary datasets;
  - *country_spec_scripts*: Folder containing country-specific script (As specified below, this will be the main folder that you will use for the first part of your job. See Sect. 3).
- **Docs**: Folder containing all the relevant documentation (including this guide). This folder has only one subdirectory:
  - *docs_scripts*: This folder contains the RMarkdown scripts (and related files) used to produce the documentation.

2.2 Preliminary Steps: Fork and Clone

In order to fork the 'EESstacked' repository you must go on the repository page, and then click on the '**Fork**' button in the top-right corner of the page.

Once you forked the repo your next step is **cloning** it. In RStudio you can achieve this following the steps below:

1. Go to "File > New Project"
2. Click on "Version Control: Checkout a project from a version control repository"
3. Click on "Git: Clone a project from a repository"

4. Fill in the info:

    a. URL: use the repo HTTPS address
    b. Create as a subdirectory of: Browse to where you would like to create this folder

At this point RStudio should open a new working session related to your new R project, and you should be able to work on the local version of the repository.

2.3 Workflow

Once concluded the preliminary steps you should be able to start working on the project. In order to create our stacked data matrix (hereinafter, SDM) in a tidy and efficient way we will use the following workflow:

1. Modify the R scripts of interest (see Sect. 3);
2. Save your files and **commit** the changes in R studio or in your Git client. Try to write synthetic but informative commit titles/notes;
3. Once selected all the commits **push** them on your remote repository on GitHub;
4. Once pushed, go on the GitHub page of your forked repository (‘`https://github.com/.../EESstacked`’), then go on the ‘Pull requests’ tab (see Figure 1), and click on **‘New pull request’** (see Figure 2);
5. If there are no conflicts between your forked repo and the upstream one, then you can **create a new pull request** (see Figure 3). Leave a comment and the create the pull request;
6. After that I will receive a notification on my GitHub repository and I will check the scripts. If everything is fine I will merge the new scripts in the upstream repository.

Figure 1: Pull requests tab

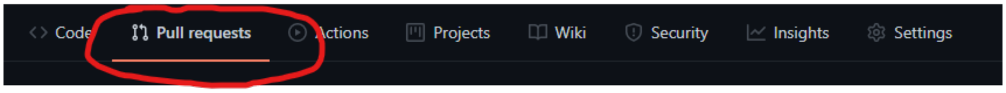
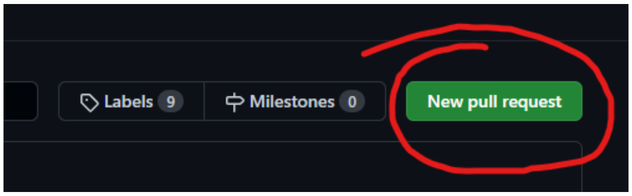


Figure 2: New pull request

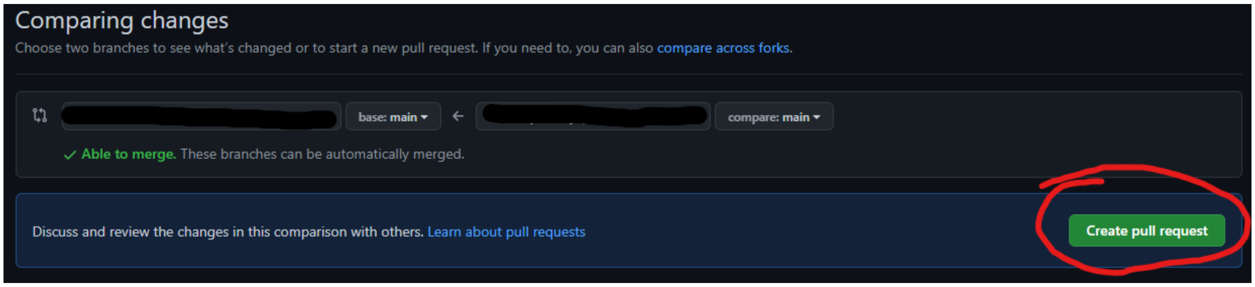


2.4 Update your GitHub and Local Repositories

You might be interested in updating your local repository with the latest version of the original/upstream repo. In order to do so...

1. Go on the GitHub page of your forked repository (‘`https://github.com/.../EESstacked`’);

Figure 3: Create pull request



Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

[redacted] base: main ← [redacted] compare: main

✓ Able to merge. These branches can be automatically merged.

Discuss and review the changes in this comparison with others. Learn about pull requests

2. **Fetch** the upstream using the dedicated button (see Figure 4), then click on the button 'Fetch and merge' (see Figure 5). If there are no conflicts between the upstream repository and yours, then your remote repo on GitHub should be synced with the usptream;
3. Then go on Rstudio or your Git client and **pull** the repository.

Figure 4: Fetch upstream

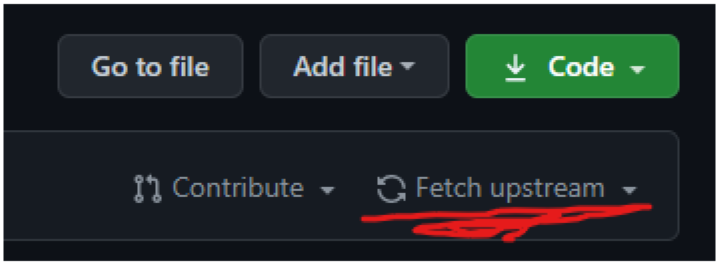
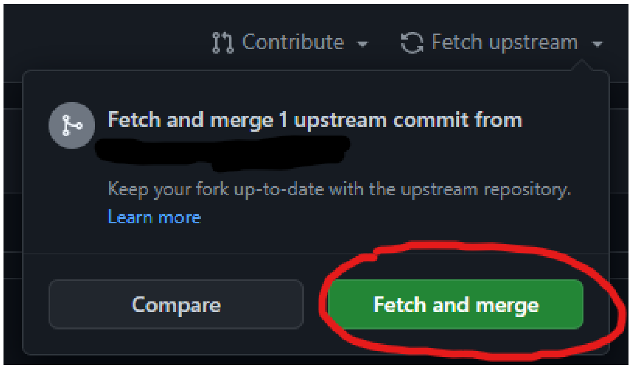


Figure 5: Fetch and merge



Fetch and merge 1 upstream commit from [redacted]

Keep your fork up-to-date with the upstream repository. Learn more

3 The R Workflow

3.1 Syntax and Coding style

Although you should feel free to use your preferred one, it would be much easier for all of us to follow a similar syntax style. As you will see browsing the scripts, I usually follow a rather standardized syntax,

marginally inspired by Hadley Wickham's style guide.
In any case, the most important thing is to *maintain the code tidy and readable*.

3.2 Packages

In order to build a stacked data matrix in R you will need a set of R packages in order to import, reshape, and mutate/recode the data. This is the list of packages that we will use in the following weeks:

- `tidyverse` is an opinionated collection of R packages designed for data science. All packages that are part of it share an underlying design philosophy, grammar, and data structure;
- `magrittr` provides a set of operators, such as the 'pipe'-like operator `%>%`, which make your code more readable. Although being part of the `tidyverse`, you will have to load it separately in order to have at hand the full set of operators that the package provides;
- `haven` package enables R to read and write various data formats used by other statistical softwares. As in the case of `magrittr`, `haven` is part of the `tidyverse`, nonetheless you will have to load it separately;
- `data.table` is another package for data manipulation operations (such as subset, group, update, join,...), sometimes considered as an alternative to the `tidyverse`[1];
- `labelled` provides functions to manipulate metadata as variable labels. This package is particularly useful when working with Stata or SPSS dataset files;
- `here` enables easy file referencing by using the top-level directory of a file project to easily build file paths;
- `miceadds` contains functions which complements existing functionality in R and in particular the `mice` package for multiple imputation procedures. Mostly we will use it for a very convenient function (`source.all`) that allows to source multiple scripts contained in a single folder;
- `stringr` consists in a package for string manipulation.

3.3 Scripts Workflow

The stacking procedure is achieved with a single R script ('EES2019_stack.R') sourcing collateral scripts stored in the 'Scripts' subdirectories (see Sect. 2). About the latter ones, the scripts of interest are those stored in the 'country_spec_scripts' folder. These scripts are country-specific and their main purpose is (or, better, will be) creating a set of SDMs that then will be the backbone of the EES 2019 SDM.

The workflow so far is at if follows:

1. With the main script ('EES2019_stack.R') we (a) install and load the main R packages, and then (b) we load the EES data and collateral data frames;
2. Then we source the country-specific data frames, that will create a set of SDMs containing just four variables: the variable identifying the EES study (named `countrycode`), the EES survey respondents ID code variable (named `respid`), the stacked party variable (named `party`), and the variable identifying each single combination of the first two (named `stack`);
3. Finally these SDMs are stacked on each other and ordered.

[1] If you are interested, you can find a rather comprehensive comparison here.

This workflow represents most of the passages for creating an SDM including all the original variables of the EES 2019 voter study.

4 To-do and Deadlines

Our main task in the following two weeks will be creating the country-specific scripts mentioned in the previous passages. For facilitating your job, I created an exemplary country-specific script ('EES2019_it_stack.R', located in the 'Scripts/country_spec_scripts/' subdirectory) which refers to the Italian voter study of the EES 2019. In the following lines you can find a step by step guide of such script that you shall use as a reference one for creating yours.

Before testing this script and working on yours, launch the main one ('EES2019_stack.R' located in the 'Scripts' folder). By doing so you will install/load the R packages and the main data frames.

4.1 Country-specific Procedure: An exemplary R Script

The script dedicated to the 2019 EES Italian voter study starts creating a set of **country-specific data frames** by filtering the main ones.

```
EES2019_it <-
  EES2019 %>%
  filter(countrycode==1380)

EP2019_it <-
  EP2019 %>%
  filter(countryshort=='IT')


EES2019_cdbk_it <-
  EES2019_cdbk %>%
  filter(countryshort=='IT')
```

In the second step the Italian voter study **respondents' ID codes** are selected and stored as a numeric vector. These codes represent the first element of our SDM.

```
respid <-
  EES2019_it$respid %>%
  as.numeric()
```

The third step represents the key one, namely the passage in which **the relevant parties are selected**. As already mentioned during our first call, there are several criteria that can be used to make such selection. In creating our SDMs two criteria will be used:

1. The parties selected are only those for which the EES 2019 provides a propensity to vote (PTV) variable (`Q10_PTV`; See the EES2019 Codebook);
2. The parties selected with the first criterion (1) have obtained at least one seat in the 2019 EP elections.

In the exemplary script, first the auxiliary data frames are inspected.

```
ptv_crit <-
  EES2019_cdbk_it %>%
  dplyr::select(partyname, Q10_PTV)

votes_crit <-
  EP2019_it %>%
  filter(party_name!='Other parties')
```

According to the codebook data frame, the 2019 EES voter study provides a PTV variable only for seven Italian parties. According to the 2019 EP election results, these seven parties obtained at least one seat in the European Parliament (otherwise they would be categorized as 'Other parties' in the EP results data frame). Hence, said parties are those that are going to be selected.

In order to uniquely identify and thus select such parties we rely on the identification codes of the original vote choice variable of the 2019 EES voter study (`Q7`; See the EES2019 Codebook).

```
party <-
  EES2019_cdbk_it %>%
  dplyr::select(partyname, Q10_PTV, Q7) %>%
  na.omit() %>%
  .$Q7
```

At this point we have the essential elements for creating an SDM, namely the respondent ID codes (`respid`) and the list of the relevant parties (`party`), and we can proceed to stack the observations.
This is achieved by first creating a data frame with one row for each combination of said two variables. Then we add to the data frame the EES study identification variable (`countrycode`), and we create a variable identifying each voter-party relationship (`stack`).

```
EES2019_it_stack <-
  expand_grid(respid, party) %>%
  mutate(countrycode = EES2019_it$countrycode %>% unique,
         stack = paste0(respid, '-', party)) %>%
  dplyr::select(countrycode, respid, party, stack)
```

At this point the SDM of the Italian EES 2019 voter study is ready. The scripts concludes by cleaning the environment and keeping only the SDM and the datasets loaded in the main script.

```
rm(list=ls(pattern='_it$|_crit$'))
rm(list = c('respid', 'party'))
```

4.2 Who Does What

In the following two weeks we will work on the country-specific data frames. This is the list of scripts that each of you should complete before the deadline specified in the following section:

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

I will take care of the remaining ones (namely: Belgium, Bulgaria, Croatia, Cyprus, Greece, Netherlands)

4.3 Deadlines

- **17.08.2021**: Country-specific scripts and pull requests completed;
- **21.08.2021**: Zoom call for discussing the work done and eventual issues;
- **30.08.2021**: The EES2019 SDM with the original EES 2019 voter studies variables is completed.

NB: Between the 9th and the 16th of august I will be on vacation :) Thus, please fork and clone the repository before the end of this week so to solve any issue before my departure.