README

Replication files for “Rally ’round the barrack: Far-right support and the military”, by Francisco Villamil, Stuart J. Turnbull-Dugarte, and José Rama.

This folder includes all the necessary materials to replicate the results from both the main Article and the Online Appendix. Most of them have been run using R 4.3 on macOS 12.3. We include the output from `sessionInfo()` at the end of this readme file. **NOTE:** The spatial analyses are computationally expensive and need a lot of RAM. We run it on a Google Cloud VM instance (e2-highmem-16, 128 GB RAM), and took around 72h. Without the spatial analyses, the code just takes a few minutes to build.

The code is organized in tasks that depend on the results of the previous ones. All the code is self-contained. Each folder contains the code and the output of one task. The taskflow is as follows:

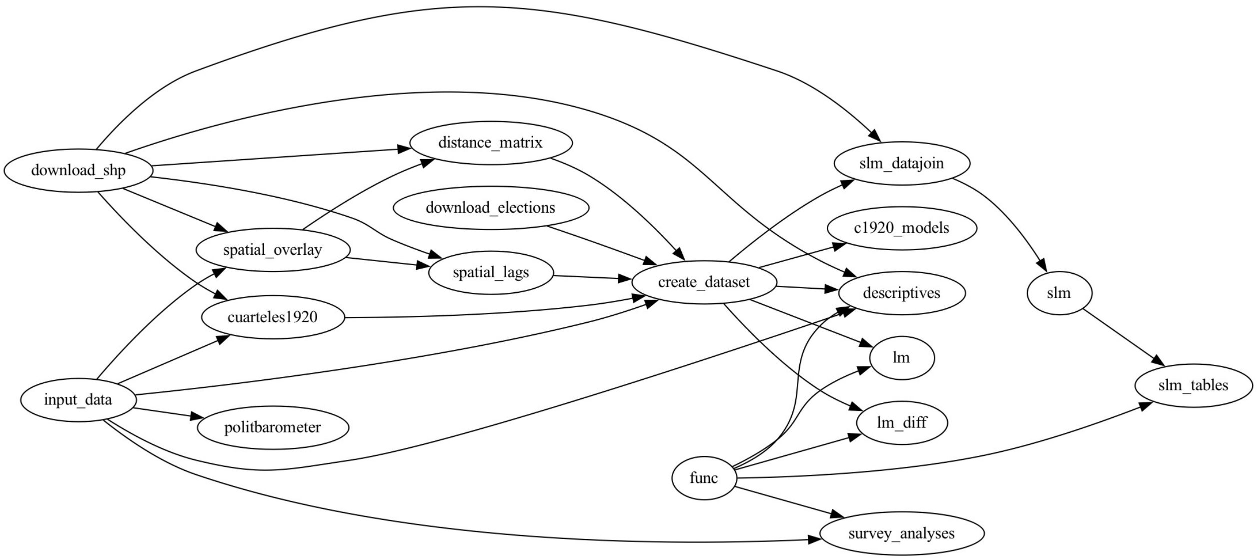

Figure 1: taskflow

In addition, there is an additional task (`cses`) that is self-contained.

Files correspoding to tables and figures in main text

- Table 1: `survey_analyses/output/tab_survey_freq.tex`
- Table 2: `survey_analyses/output/tab_survey_models.tex`
- Table 3: `lm/output/tab_base.tex`
- Table 4: `lm_diff/output/tab_lm_diff.tex`
- Table 5: `lm_diff/output/tab_lm_diff_int.tex`
- Table 6: `slm_tables/output/tab_sem.tex`
- Table 7: `slm_tables/output/tab_sdem.tex`

- Table 8: cses/output/tab_cses.tex
- Figure 1: survey_analyses/output/sim_mil_gap_ideo.pdf
- Figure 2: descriptives/output/cuarteles_map.pdf
- Figure 3: descriptives/output/rainplot_army_a19.pdf & rainplot_army_n19.pdf
- Figure 4: descriptives/output/madrid_map.pdf
- Figure 5: lm_diff/output/diff_int_pp.pdf & diff_int_ild_pp.pdf

---

Raw data files

There are two types of raw files that are needed to run the analyses. The download_shp folder contains shapefiles for Spanish provinces and census sections, while the input_data folder contains a set of files that were previously downloaded or created:

- barometers_full.csv: individual-level dataset that contains all surveys (monthly barometers) merged, which is used in the individual-level analyses. We define its variables in the separate codebook.
- cuarteles.csv: location of military facilities in Spain, coded as explained in the main paper
- cuarteles1920.csv: location of military facilities in Spain in 1920, coded from historical archives as explained in the main paper and in the Online Appendix
- INE_census.csv: census data for all Spanish municipalities, see this repository
- results1936.csv: electoral results from 1936 elections, from replication data from Villamil (2021)
- secc_censal_indic_demograficos.csv: demographic data at the census section-level
- secc_censal_renta.csv: income data at the census section-level
- ZA2391_v13-0-0.dta.zip: Politbarometer data, obtained from the GESIS website
- cses_imd_csv.zip: CSES data, downloaded from cses.org

In addition, the func folder contains a set of R functions predefined that are used throghout the code.

Running the code and replicating results

When running the scripts, any order is fine as long as it respects the taskflow above. Also, it is possible to run separate scripts or to use make to run the whole project (notes on this below). We present here an order that **creates the datasets**, runs the **main analyses** and the **extra analyses**. Every task corresponds to a folder that includes only one **.R file** and an **output** folder (if it's not there or is deleted, you might have to create it before running the R code). The R files read from the output folders of previous tasks.

Note: the setwd command is commented out in every file. Remember to change it to the current directory, unless you're using make.

Data creation:

- download_elections: downloads the electoral data from Spain, using the infoelectoral package

- spatial_overlay: performs the spatial overlay between the military facilities dataset and the census sections shapefile
- cuarteles1920: does the same spatial overlay as above but using the dataset on military facilities existing in 1920
- distance_matrix: calculates distance between military facilities and census sections
- spatial_lags: creates the neigboring matrices between different census sections
- create_dataset: creates the final dataset used in the local-level analyses
- slm_datajoin: merges the data frame with the spatial features and creates the object needed for the spatial models

Main analyses:

- survey_analyses: runs the individual-level analyses
- lm: runs the main local-level models
- lm_diff: runs the local-level linear models on diffusion
- slm: runs the spatial models. Contains three R scripts (`slm_invd.R`](.) and [`slm_nb.R`), which correspond to the models that use inverse distance and neighboring (contiguous and within 2km) matrices.
- slm_tables: creates tables out of the output from the spatial models.
- cses: runs the analyses on cross-country differences using the data from CSES

Extra:

- descriptives: runs the descriptives statistics and graphs, including maps and summary statistics
- c1920_models: runs the models using the 1920 data on military facilities, included in Appendix I
- politbarometer: runs the analyses using survey data from the Politbarometer in Germany, included in Appendix K

Using `make`

From the command line (Unix/macOS), the following script will download the repository, clean up all output files, and run all scripts again. By default, `make` (or [`make all`) will run everything but the memory-intensity spatial analyses. `make really_all` runs the whole project. (Check [`make help`.)

```
git clone https://github.com/franvillamil/vox_military
cd vox_military
make clean
make really_all
```

Note, however, that these analyses contain a lot of spatial data manipulation and analyses, and it is not uncommon that there are problems with spatial libraries when running these scripts, particularly when using [`Rscript`].

OS and software

Most of the analyses can be run on a normal computer. We have used R 4.3.1 on a 2015 MacBook Pro with 16GB of memory (on macOS Monterey).

One task (the spatial analyses in `slm`), however, requires much more memory and might not be suitable in a normal computer. This applies to two R scripts, `slm/slm_invd.R` and `slm/slm_nb.R`. In our case, we run them on a Google Cloud VM instance (e2-highmem-16, 128 GB RAM), and took around 72h.

This is the `sessionInfo()` output in thel local computer after running `distance.R`:

```
R version 4.3.1 (2023-06-16)
Platform: x86_64-apple-darwin20 (64-bit)
Running under: macOS Monterey 12.3.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: Europe/Madrid
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] rgeos_0.6-4 rgdal_1.6-7 sp_2.0-0    dplyr_1.1.2 tidyr_1.3.0

loaded via a namespace (and not attached):
 [1] vctrs_0.6.3        cli_3.6.1          rlang_1.1.1        KernSmooth_2.23-21 DBI_1.1.3
[12] grid_4.3.1         classInt_0.4-9     tibble_3.2.1       lifecycle_1.0.3    compiler_4
[23] utf8_1.2.3         pillar_1.9.0       magrittr_2.0.3     tools_4.3.1        proxy_0.4-
```