SOP

Martín Alberdi, Dirck De Kleer, Paula Dümpelmann and Ludwig Schulze

2023-09-21

Contents

Summary

a. Purpose: This SOP outlines the guidelines and procedures for the Team 1 of the EUI's 2023/2024 "Practicum in Reproducible Research Methods" Course. The main goal of this SOP is to provide all team members information about how to organize project materials and workflow.

b. Reproducibility: Your work contributes to scientific knowledge. For scientific knowledge to cumulate, it must be reproducible by others.

c. Transparency: You are working as part of a team. Everything you do affects all other team members. For your team to be able to understand your work, all steps must be transparent so that any outsider can comprehend and catch up with your work.

d. Work Load: A fully reproducible workflow means higher up-front time investment for reduced errors and reduced later work load. The team should expect to spend as much as 30 percent of their time documenting and annotating their work. Transparency and reproducibility are prioritized over implementing quick and dirty solutions.

e. Scope: The following sections apply to all team members regardless of their location.

Communication and Reporting

a. Monthly Check-Ins: Team members will have monthly check-ins with each other during the extended break of the course in order to discuss project progress, clarify objectives, and address any questions or concerns. Team members are expected to be available at a standard time every week while time differences are taken into account. If any team member will be unavailable, notify each other in advance.

b. Communication Channels: The primary mode of written communication will be through a dedicated Whatsapp channel.

c. Response Time: Team members should respond to communication from other team members promptly and maintain open lines of communication. In general, responses are expected within 48 hours, or sooner if a matter is highly urgent.

d. Absences: A team member who will be unavailable for a period of time due to personal or medical reasons should notify the team, giving as much advance notice as possible.

Project Management

a. Project Plan: The team will create a clear project plan outlining objectives, deliverables, and timelines. You are encouraged to provide feedback on aspects of the plan that you consider would benefit from modifying.

b. Task Assignment: The team will decide together on the allocation of different tasks.

c. Task Reassignment: If you are assigned a task and discover that your skills are not fully adequate to it, or that it is taking you far longer than expected, you should make your other team members aware of this. The team will consequently decide on a task reassignment.

d. Task Prioritization: The team decides together on the priorities of the project.

e. Timelines and Deadlines: Team members are responsible for meeting the agreed-upon deadlines for each task and informing each other in advance if they anticipate delays. They are encouraged to provide feedback if specific timelines or deadlines seem unrealistic.

f. Task Documentation: Team members should maintain detailed documentation of their work, including methods, results, and any challenges encountered. The standard place to do this is in a project logbook. Each team member usually maintains a single logbook.

g. Project Documentation: Projects typically reside in a dedicated Github repository, to which team members will be granted access. Team members should pull from the repo before starting work and should push frequently.

h. Public Posting: Once a project is complete, it will be put in the public domain. As a result, you should keep in mind that eventually there will need to be a data codebook and that all code underlying everything reported in a publication will have to be reproducible. Even if you are not working directly on writing a codebook, all data will have to be documented by someone when that person writes the codebook. Try to anticipate this and to make sure your work easily permits it. Assume that the person writing the codebook cannot contact you directly for information.

i. Interoperability: All the work you do on a project must be designed to run on other computers. This requires you build in from the start code that is platform-independent.

Directory and File Structure and Naming

a. Directory Structure: Each project will be organized in a hierarchical file structure, with the topmost level typically named for the overall project. Every team member will have an identical copy of the project directories and files.

b. Directory and File Names: Directory and file names should not contain any blank spaces, and should only use capital letters if required for readability. The underscore character should be used to divide words for readability as necessary.

c. Dates: Dates should be formatted according to the International Organization for Standardization (ISO) guidelines, i.e. yyyymmdd.

d. Directory and File Ordering: To the extent possible, directory and file names should be organized sequentially, i.e. using a two-digit leading number system so that they hang in order of operations (e.g. 01_admin, 02_lit_bib, 03_funding, 04_design, 05_pap, 06_irb, 07_analysis, 08_presentations, 09_papers, 10_scripts, etc.).

e. Master Version: There should only be a single canonical version of any file; e.g. we should never encounter files such as "data_v1" and "data_v2" or "paper_v1" and "paper_v2_myinitials." If you need to temporarily generate different versions of the same file for a specific purpose, you may wish to branch in Git or to work locally and only push when you have fully resolved an issue.

f. File Names: Files should be named for their contents and not for their authors. We should find files such as "results_section.tex" and "female_voters1950.png." We should not encounter files such as "results_joe.tex" or "female_voters1950_Lucas_version.png."

g. READMEs: Every directory and subdirectory should contain a plain text "readme" file that describes what is contained in the directory, who assembled the material, when it was assembled, where it was sourced, when the readme was written or updated, and any other essential information.

h. Tables and Figures: Tables and figures should be output into appropriately labelled directories and should be pulled into documents from those locations. Tables and figures should be manipulated using code, not manually. File names for tables and figures should be comprehensible, meaning a longer, more precise name is preferred to a shorter, more ambiguous name.

i. Confidential Information: If you have confidential project information that should not be shared with all team members, please tag relevant files and directories using .gitignore so you retain the information but it does not pass into the repo.

Data Management

a. Data Security: Team members should adhere to all institutional and ethical guidelines regarding data security and privacy. No data or project documents should be placed on devices that are not owned personally by the team members and to which the team members do not have exclusive access. For legal and ethical reasons, data that contains identifying information about research subjects must be handled with particular care. Team members should exercise vigilance in protecting their hardware from theft.

b. Data Organization: Team members should organize data in a structured manner, labeling files and folders appropriately, as described under Directory and File Structure and Naming.

c. Raw Data: Raw data should be exclusively stored in separate directories, appropriately labelled. Generally, raw data is organized by source and/or type (e.g. electoral_data, census_data, VDEM_data, etc.) although occasionally, it will be organized along other lines. Each raw data directory should include a readme that documents sourcing details and other relevant information that will be included in a codebook. Once assembled, raw data should be locked; i.e. it must remain exactly as input, scraped, or downloaded. It should not be mixed with cleaned or transformed data.

d. Download and Access Documentation and Dates: Data that is downloaded from the internet should include the full url location and the download date in an accompanying readme.

e. Clean Data: Cleaned data should be stored separately from raw data. Cleaned data should (usually) be stored in directories whose structure parallel those of the raw data.

f. Variable Names: Variables should be named with comprehensible English-language labels (e.g. "party") and not with incomprehensible labels (e.g. "var1", "var2", etc.). Binary variables should be named according to the meaning assigned to 1; e.g. instead of "gender" (which introduces doubt about the meaning of 0 and 1), name the variable "male" if men are coded 1 and women 0.

g. Variable Transformations: In projects that incorporate variable transformations, some useful conventions are:

h. In case of language translation, a variable might end with *_eng* or *_urdu*.

i. In case of inverting the order, a variable might end with *_rev*.

j. Value Labels: Variables that take different values should be assigned value labels within the dataset; e.g. "sex" = (0,1,2,3), labelled 0 = "does not disclose," 1 = "male," 2 = "female," 3 = "transgender."

k. Missing Values: Missing values are generally coded "NA" or some other standard identifier, such as a period or -999. Different kinds of missingness should be precisely indicated; e.g. missing because unavailable from source material is different from missing because the unit does not exist. These differences must be documented to the extent possible.

l. Data Manipulation: The standard order of operations for data manipulation is to bring in raw data, clean and store the data, and create new variables. These operations should be performed in distinct and clearly labelled files, operating over distinct and clearly labelled directories. Directories and subdirectories will commonly take the form: 01_data <– 01_raw_data; 02_clean_data; 03_datasets, where 03_datasets are the final processed datasets to be analyzed. The various files performing these operations will be named according to the following type of conventions: 02_cleaning_code <– 00_master.R, 01_build_rawdatasource1.R, 02_build_rawdatasource2.R, 03_build_rawdatasource3.R, 04_create_voteshares.R, 05_create_partyid.R, 06_add_covariates.R, and so forth. Often many variable transformations are required and these should be done in well-labelled files that operate on the dataset before it is analyzed in order to reduce repetition. The 00_master.R file should perform a full clean run of all processes.

m. Codebook: It is usually most efficient to draft a codebook as soon as a dataset is built rather than waiting until we are preparing to post the dataset, when many details will have been forgotten. This also allows team members to quickly ascertain what variables have been created and how.

n. Clean Runs: A clean run of the dataset assembly process should begin by removing all objects in memory and by removing all previously generated versions of temporary files and output datasets. The code to do this should be at the top of the file. This may be relaxed in very large simulation or assembly processes, where removing all objects would then require too lengthy a process.

o. File Headers: Every file that includes code should begin with a standard header that includes: purpose, author, initial date programmed, and any other essential information that permits other team members to operate on the file (e.g. source files, output files, where the code falls in the pipeline, etc.).

p. Data Backup: Team members should push all materials to Github frequently so that they are backed up on the server. In general, this means that work should be pushed every time the team member gets up from the computer, or at a minimum at the end of the day.

q. Git Conflicts: It is common to experience conflicts when pushing to a Git repo. Simple conflicts can usually be easily resolved by going into the offending file, reviewing the conflicts (demarcated by HEAD), and removing whichever version seems obsolete or wrong. Before doing this, you may want to communicate with other users to investigate the origin of a conflict. This may be especially helpful if you know that someone else is simultaneously editing the same document as you.

r. Data and Project Sharing: All project materials should be considered confidential and team members should discuss specifics of the project only among each other.

Literature and Bibliography

a. Literature: The team will decide throughout the process who should be responsible for writing the literature review.

b. Bibliography: References are stored in a bibtex file, which is located in the "lit_bib" directory under the project.

c. Reference Keys: In written work, citation keys take the format "authorlastnameYY." In cases where a single author has multiple publications in the same year, the format changes to "authorlastnameYYa," "authorlastnameYYb," etc. For clarity, articles should be stored under names that are identical to the reference key that will be used.

Collaboration and Teamwork

a. Collaboration Tools: Team members should utilize collaboration tools such as shared document platforms or version control systems. As a rul of thumb, this means working on Github.

b. Team Meetings: Team members will normally be required to attend regularly held virtual team meetings to discuss progress, share insights, and collaborate with other team members.

c. Peer Support: Team members should actively support and collaborate with their fellow team members, sharing knowledge and expertise to enhance the overall project outcomes.

Research Ethics

a. Research Ethics: Team members should adhere to ethical guidelines in all aspects of their work, including data collection, analysis, and reporting. Any ethical concerns should be communicated with each other.

b. Human Subjects Training: If you will be interacting with human subjects for data collection, you will need to have received a certificate from CITI showing that you have been appropriately trained in the protection of human subjects. This requires you take the CITI course on Social-Bahavioral-Educational (SBE) Foundations (https://about.citiprogram.org/course/social-behavioral-educational-sbe-foundations/). The course costs $129 and is a reimbursable research expense.

c. Regulatory Compliance: In addition to approval by an Institutional Review Board at the team members' home institution, the team is committed to complying with regulations in the country where the research is situated. As part of these processes, your name and other identifying details will be provided to the university where the research team is based and/or the university of record for the IRB, the funding agency, the contracting institution, and possibly government and/or university authorities in your own country.

d. Plagiarism: If you contribute to written work, you should make sure to cite your sources appropriately. Do not copy material off the web or from ChatGPT without first consulting other team members about whether that is appropriate for the specific task.

e. Fraud: In order to protect the project from potential accusations of data fraud, it is essential to retain meticulous, highly detailed documentation about data collection procedures.

Professional Development

a. Training Opportunities: Team members may be provided with access to training materials, webinars, or workshops to enhance their skills and knowledge in relevant areas.

b. Skill Development: Team members should actively seek opportunities to develop their research skills and stay updated with the latest developments in their field.

c. Professional Conduct: Team members should maintain professionalism in all interactions and conduct themselves with integrity and respect towards fellow team members, with the public, and with stakeholders. Team members should report any instances of harassment by the public, stakeholders, or other team members to Professor Golden and the EUI. All reports will be handled confidentially.

Contracts

a. Public Funding: Team members are drawing their income from scarce research funds created by public taxes and should be mindful of that.

b. Payments: This work does not yield any financial gains other than through the improved job marking opportunities the course is likely to bring about.

c. Contract Length: The contract ends with the conclusion of the EUI course.

Performance Evaluation

a. Performance Assessment: Professor Golden will evaluate the performance of the team members based on their progress, quality of work, adherence to deadlines, and overall contribution to the project.

b. Feedback: Professor Golden will provide regular feedback to the team members, highlighting areas of improvement and acknowledging their strengths.

c. Performance Improvement: Team members should proactively address any performance-related concerns raised by the Professor Golden and take necessary steps to improve their performance.

Conclusions

This SOP serves as a guideline for the Team 1 of the EUI's 2023/2024 "Practicum in Reproducible Research Methods" Course. Adhering to these procedures will ensure effective communication, efficient project management, and high-quality research outcomes. Adhering to these procedures will ensure effective communication, efficient project management, and high-quality research outcomes.