

READ ME

November 7, 2023

1 Intro

Note: if you're reading this off the homepage of my Github, this is just the LaTeX version of the README. Go to the "**instructions**" folder to see the PDF version.

This template is designed to be a comprehensive template for research projects; from do-files that provide macros to create output formats in a consistent fashion, but also the file organization that harmonizes the analysis phase with the writing phase. The two biggest issues I've dealt with in applying results to paper is that the table and other output formats are cumbersome to design and to be consistent throughout the paper, and updating draft with latest versions of outputs. To address these issues, the DO-file template comes with a suite of preset formats so that outputs are consistent with each other, but also the path of where the outputs are stored allows for replacing and adding new figures is less problematic. This process can be further expedited when using online editor, Overleaf, with repository syncing with GitHub. Once set up, changes made in Stata can seamlessly pushed into your paper in Overleaf with minimal effort and editing. This is a quick breakdown of structure of this template, and how the do-file functions. In future update, I'll provide a more in-depth tutorial on how to use the do-file. But for now, look over this document, download the whole project (click the green "**Code**" button on the project homepage https://github.com/gxstorer/project_template and then "**Download ZIP**"). To see the whole process, read the "**tutorial.pdf**" in the "**instructions**" folder.

2 Template Design

Project folders are categorized into two subfolders: "**data**" and "**project**". Data stores folders pertaining to the inputs used in the project, meanwhile the project folder stores various outputs that come from the data.

- Data:
 - **original_data**: A folder where any source data that the project is based on is located. Keeping this separate helps ensure the preservation of the original data for reproducibility.
 - **working_data**: A folder where any produced data within the project is located. Merging datasets, converting .CSV files into .DTA files will all be located here and separate from original data.
 - **code**: DO-files used in project.
- Project:
 - **sections**: Each section of a paper falls into 8 distinct sections, and each may require minor formatting changes. This folder stores each of the .tex files and a set of starter templates to allow ease in the process of producing an initial document.
 - **tables**: Tables produced in project will be stored as .tex files that all can be found in this one folder. DO-file will use the local macros to direct all tables produced to be stored in this folder.
 - **figures**: Graphs and images produced in project will be stored in this folder.
 - **equations**: Equations will be stored in this folder.

3 DO-file

Within the "code" folder, there is a DO-file labeled as "do_file_template.dta". This template is designed to automate outputs as efficiently as possible through the use of macros. Many outputs are intended to follow a consistent design to enhance readability, but implementing a custom design and maintaining consistency is quite cumbersome.

- Each line of code generally falls into 4 columns:
 (1) command (2) command code (3) command syntax (4) comments*
 * Lengthy command codes and syntax are spaced into multiple columns for better visibility.
- Comments will explain what the code does, but also will highlight if there is a manual entry that is needed that are categorized as:
 - (Required) : User must make some kind of action before running do-files
 - (Manual) : During editing phase, editor will have to make a manual change from template before running do-files.
 - (Optional) : User may manually change code syntax if applicable.
 - (NOTE) : Additional information regarding manual changes of given code.
- DO-file automates at two levels: global and locals.
 - **globals**: Many outputs use the same types of commands, and when going through the editing process, it's cumbersome to get all outputs using a consistent format throughout; some tables may have different decimal placements, or using different LaTeX commands. During the editing phase, modify which commands that are desired to be consistent throughout.
 - **locals**: Each output is going to have output-specific commands (i.e. title). The local macros section provides a template set of local macros that are to be placed with each exported output. The '**locals**' macro groups all locals together so that the export command is short and to make troubleshooting more efficient.
- Checklist for running this specific do-file:
 - Ensure project folder holds the main folders.
 - * Internal users: ensure the local macro "project" matches folder name and that the path is set.
 - * External users: Set path to project as noted in the "**Set Current Directory**" section of DO-file.

- Review all "(Optional)" codes and confirm those options.
- Packages required: **ESTOUT** & **REGHDFE**
- Optional package: **SCHEMEPACK**
 - * If not currently downloaded, then manually enter these commands:
 - **ssc install estout**
 - **ssc install reghdfc**
 - **ssc install schemepack**

4 Process

1. Create Overleaf Project.
2. Clone GitHub repository onto local drive.
3. Replace Overleaf file with template project.
4. Push template to Overleaf via GitHub.
5. Use template DO-file to work on project. **template_demo.do** is a simplified example of how the template looks when being used.
6. Push outputs to Overleaf.
7. Compile changes in Overleaf.
8. Edits can be made either in Overleaf or locally, and GitHub will sync all changes.
9. Expedite bibliography with Zotero syncing.

Link to Overleaf Template: **Project Template**