Integrating Overleaf, Dropbox, GitHub, and STATA for Automated Workflows

Domenico Rossignoli
DISEIS, Università Cattolica del Sacro Cuore, Milano

April 23, 2025

Outline

Introduction

- Overview of our integrated workflow.
- How Overleaf, Dropbox. GitHub, and STATA work together.
- Benefits of automating the process.

Requirements

- An Overleaf **Premium** account: GitHub syncronization is not available on the free plan
- A Dropbox account
- A GitHub account
- A STATA license

Note: This integration works with any statistical software/language. You may easily apply the same instructions to **R** or **Python**.

Creating an Overleaf Project

- Start by creating a new project in **Overleaf**.
- Open the Menu in your Overleaf Project.
- Under “Sync” click on **GitHub**.
- Select your owner name and provide a name for the repository.
- Overleaf will automatically set up the repository and sync your project.

Note: Overleaf does not sync to an existing GitHub repository: it creates a new one.

Note2: The name of your GitHub repo will be the folder name in your Dropbox!

Cloning the Repository Locally

- Once Overleaf has created and synced the GitHub repository, go to GitHub to copy the repository URL.
- Open your terminal (**Command prompt**) or Git Bash.
- Navigate to your desired folder using the `cd` command.
- If you want to clone the GitHub folder in **Dropbox**, navigate to your Dropbox folder.
- Run the clone command:

```
git clone https://github.com/your-username/your-new-repository.git
```

Note: paste in the above command the repository URL that you copied in GitHub

STATA Workflow

- Change directory into the repository:

```
cd your-new-repository
```

- Work in STATA to generate tables and output files.
- Save these outputs in the local folder of your Git repository.
- You can make use of `global` macros to store the path of your folder in Dropbox

Automating Git Operations in STATA (Part 1)

- At the end of your STATA do-file, change the working directory to the Git folder.
- Execute Git commands via the shell.

Example Code:

```
* Define the Git repository folder using a macro
local git_folder "C:\Path\To\GitRepo"

* Retrieve current date from STATA
local oggi = c(current_date)

* Create commit message including the date
local commit_msg "Automatic update from STATA - `oggi'"

* Change to the Git repository directory
cd "`git_folder'"
```

Automating Git Operations in STATA (Part 2)

```
* Pull remote changes (e.g., from Overleaf updates)
!git pull origin main

* Add all modified files
!git add .

* Commit the changes with the commit message
!git commit -m "`commit_msg'"

* Push the changes to GitHub
!git push origin main
```

Note: I found sometimes tricky to make the local for date work correctly. You can remove it, and just replace the message with a standard invariant message.

Syncing with Overleaf

- Your Overleaf project remains linked to the newly created GitHub repository.
- Changes pushed from your local machine (via STATA) will be visible in Overleaf after a sync.
- If needed, pull updates in Overleaf to ensure your project is current.

Final Considerations

- Always perform a `git pull` before adding, committing, and pushing to integrate remote changes.
- Automating Git operations in STATA helps save time and reduces the risk of human error.
- Use clear commit messages (including the current date) for better version control.