



Environmental Analysis Program

Pomona College
Biogeochemistry Laboratory

SOP-L801

RStudio Projects and GitHub

Standard Operating Procedure

SOP Number	SOP-L801
Revision	1.07
Author(s)	Marc Los Huertos, Isaac Medina
Date	2026-06-26
Organization	EA Program, Pomona College

FOR INTERNAL USE ONLY

This document is the property of the Environmental Analysis Program at Pomona College.
Unauthorized distribution is not permitted.

Table of contents

1 Purpose	2
2 Scope	2
3 Acknowledgements	2
4 Definitions	2
5 Safety and Hazards	3
6 Accessing RStudio	3
6.1 Pomona College RStudio Server	3
6.2 Local Installation	3
7 Setting Up GitHub	3
7.1 Create a GitHub Account	3
7.2 Configure SSH Keys	3
8 Working with Repositories	4
8.1 Clone an Existing Repository	4
8.2 The Pull-Edit-Commit-Push Workflow	4
8.3 Branching Workflow	4
9 Writing Quarto Documents	5
9.1 Rmd vs. Qmd	5
10 Recommended Resources	5
11 Advanced Topics	5
11.1 Protecting the Main Branch	5
11.2 GitHub Issues for Task Tracking	5
11.3 .gitignore	6
12 References	6
13 Training Record	6

1. Purpose

This SOP provides instructions for setting up RStudio, connecting it to GitHub, and managing collaborative research projects using version control. R (R Core Team, 2024), RStudio (Posit PBC, 2024), and GitHub (GitHub Inc., 2024) together enable collaborative, transparent, reproducible, and accessible data analysis. For a comprehensive practical guide, see (Bryan, 2023).

2. Scope

- **Applies to:** All EA Program researchers using R for data analysis and collaborative projects
- **Does not apply to:** Local-only R scripts with no version control; Python or other language workflows
- **Location:** RStudio Server at Pomona College; local installations

3. Acknowledgements

Substantial contributions: Aparna Chintapalli and Isaac Medina. We acknowledge all students who have followed these instructions and suggested improvements.

4. Definitions

- **R:** A free, open-source programming environment for statistical computing, maintained by a global community of statisticians and programmers (R Core Team, 2024).
- **RStudio:** A user interface for R that provides tools to write code, view results, manage files, and integrate with version control. Available as a desktop application or via the Pomona College server (Posit PBC, 2024).
- **GitHub:** A web-based hosting service for Git repositories that enables collaborative code and document management (GitHub Inc., 2024).
- **Repository (repo):** A GitHub project folder that contains all files, history, and collaboration settings.
- **Version control:** A system that tracks changes to files over time, allowing multiple people to collaborate while maintaining a complete history of who changed what and when.
- **SSH key:** A cryptographic key pair used to establish a secure connection between RStudio and GitHub without entering a password each time.
- **Clone:** Creating a local copy of a GitHub repository on your computer or server.
- **Branch:** A parallel line of development that lets you work on a feature or fix without affecting the main codebase until you are ready to merge.
- **Pull:** Downloading the latest changes from GitHub to your local copy.
- **Commit:** Saving a snapshot of your changes with a descriptive message.
- **Push:** Uploading committed changes from your local copy to GitHub.
- **Pull request (PR):** A request to merge changes from one branch into another; the standard review mechanism for collaborative repositories.

5. Safety and Hazards

This is a computational procedure with no physical hazards.

6. Accessing RStudio

6.1. Pomona College RStudio Server

The recommended method for EA Program work:

1. Open a browser and go to `rstudio.pomona.edu`
2. Log in with your Pomona College credentials
3. Your files are stored on the server (no local installation needed)

6.2. Local Installation

Install in this order if working offline:

1. R: cran.r-project.org
2. RStudio Desktop: posit.co/download/rstudio-desktop
3. Git: git-scm.com

7. Setting Up GitHub

7.1. Create a GitHub Account

1. Go to github.com and create a free account
2. Use your Pomona email address
3. Choose a professional username (e.g., `firstlast` or `lastname`)

7.2. Configure SSH Keys

SSH keys let RStudio communicate with GitHub securely. Set them up once per machine (or server account).

1. In RStudio, open the Terminal tab
2. Generate a key pair:

```
ssh-keygen -t ed25519 -C "your_pomona_email@pomona.edu"
```

3. Accept the default file location; leave the passphrase blank for server use
4. Print the public key:

```
cat ~/.ssh/id_ed25519.pub
```

5. Copy the output
6. On GitHub: Settings > SSH and GPG keys > New SSH key > paste and save
7. Test the connection:

```
ssh -T git@github.com
```

A message beginning “Hi username!” confirms success.

8. Working with Repositories

8.1. Clone an Existing Repository

1. On GitHub, navigate to the repository and click **Code** > **SSH**
2. Copy the SSH URL (`git@github.com:org/repo.git`)
3. In RStudio: File > New Project > Version Control > Git
4. Paste the URL; choose a local directory; click **Create Project**

8.2. The Pull-Edit-Commit-Push Workflow

Always pull before editing to avoid merge conflicts:

1. **Pull:** Click the blue down-arrow in the Git pane (or `git pull`)
2. **Edit:** Make your changes in R scripts or `.qmd` files
3. **Stage:** Check boxes next to changed files in the Git pane
4. **Commit:** Click Commit; write a clear message (e.g., “Add soil texture analysis for sites 4-6”)
5. **Pull again:** Pull once more before pushing to catch any concurrent changes
6. **Push:** Click the green up-arrow to push to GitHub

8.3. Branching Workflow

For experimental analyses or features that are not yet ready for the main branch:

```
git checkout -b feature/my-analysis # create and switch to new branch
# ... make changes and commits ...
git push -u origin feature/my-analysis
# Open a Pull Request on GitHub when ready to merge
```

9. Writing Quarto Documents

9.1. Rmd vs. Qmd

Feature	Rmd	Qmd
Maintained by	RStudio/Posit	Quarto project (open source)
Requires R	Yes	No (also supports Python, Julia)
Chunk options	In header: <code>{r, echo=TRUE}</code>	On lines: <code># echo: true</code>
SOP system uses	Legacy	Preferred for new documents

For new EA Program documents, use `.qmd`. Existing `.Rmd` files will continue to work.



Common Chunk Options

- `#| echo: true` shows the code in the output (use for teaching)
- `#| echo: false` hides the code (use for reports)
- `#| eval: false` shows the code but does not run it
- `#| fig-cap: "Caption text"` adds a figure caption
- `#| warning: false` suppresses warning messages

10. Recommended Resources

- [RStudio YouTube Introduction](#)
- [GitHub YouTube Introduction](#)
- [Understanding the GitHub Flow](#)
- Happy Git and GitHub for the user: (Bryan, 2023) – the most comprehensive practical guide
- [Quarto Documentation](#)

11. Advanced Topics

11.1. Protecting the Main Branch

For repositories with multiple active contributors, enable branch protection rules on GitHub:

1. Go to Settings > Branches > Add rule
2. Enter `main` as the branch name pattern
3. Check **Require a pull request before merging**
4. Optionally check **Require approvals** (set to 1)

11.2. GitHub Issues for Task Tracking

Create an issue for each task, assign it to a team member, and close it when the work is merged. This creates a record of who did what and when.

11.3. .gitignore

Add file patterns that should never be committed:

```
.Rhistory  
.RData  
.Rproj.user/  
docs/  
_freeze/  
*.pdf
```

12. References

Bryan, J. (2023). *Happy Git and GitHub for the useR*. <https://happygitwithr.com/>
GitHub Inc. (2024). *GitHub Docs*. <https://docs.github.com/>
Posit PBC. (2024). *RStudio IDE*. <https://posit.co/products/open-source/rstudio/>
R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. <https://www.R-project.org/>

13. Training Record

Before using R/RStudio/GitHub independently for EA Program research, personnel must:

1. Read this SOP in full
2. Create a GitHub account and configure SSH keys (Sections 11.1–11.3)
3. Successfully clone, pull, edit, commit, and push to the **beginnersluck** practice repository (Sections 12–13)
4. Create and render at least one **.qmd** file (Section 17)
5. Demonstrate the Pull-Edit-Commit-Push workflow under supervision
6. Be signed off by the PI or project lead