

Problem Set 1: Introduction to Git and GitHub

Applied Quantitative Methods for the Social Sciences II

Carlos III–Juan March Institute, Spring 2026

Instructions:

- **Deadline:** February 12, before class
- This problem set introduces you to Git and GitHub, which you will use throughout the course to submit your work
- For this course, you will submit all assignments as plain .R files in a public GitHub repository
- Complete all the tasks below and send me the link to your GitHub repository

1 What is Version Control?

Version control is a system that records changes to files over time. It allows you to:

- Track the history of your project
- Revert to previous versions if something goes wrong
- Collaborate with others without overwriting each other's work
- Keep a backup of your work in the cloud

Git is the most widely used version control system. It tracks changes locally on your computer.

GitHub is a web platform that hosts Git repositories online, making it easy to share code, collaborate, and back up your work.

2 Why Use Git and GitHub?

In research and data analysis, version control is essential for:

- **Reproducibility:** Others (and your future self) can see exactly what you did
- **Collaboration:** Multiple people can work on the same project
- **Backup:** Your work is safely stored online
- **Transparency:** Open science practices require sharing code and data

Many journals and research groups now require or encourage sharing code via GitHub.

3 The Basic Git Workflow

Git works with three main concepts:

1. **Working directory:** The files on your computer
2. **Staging area:** Files you've marked to be included in the next snapshot
3. **Repository:** The history of all snapshots (commits)

The basic workflow is:

1. Make changes to your files
2. **Stage** the changes you want to save: `git add filename`
3. **Commit** the staged changes with a message: `git commit -m "description"`
4. **Push** your commits to GitHub: `git push`

4 Ways to Use Git

There are several ways to interact with Git and GitHub. All of them do the same thing—choose whichever you find most comfortable, or mix and match.

4.1 Option 1: GitHub Web Interface

The simplest way to get started. You can create repositories, upload files, and make commits directly on <https://github.com>. Good for beginners, but limited for complex workflows.

4.2 Option 2: Command Line (Terminal)

The most powerful and flexible option. You use commands like `git add`, `git commit`, and `git push` in your terminal (Terminal on Mac, Git Bash on Windows). This is what most experienced users prefer.

4.3 Option 3: RStudio Integration

If you use RStudio, it has built-in Git support. You can stage, commit, and push using buttons in the Git pane (top-right by default). This is convenient if you're already working in RStudio. To enable it, you need to have Git installed and configure RStudio to recognize it (Tools → Global Options → Git/SVN).

5 Tasks

Complete the following tasks. You can use any of the methods described above (web, command line, or RStudio).

5.1 Task 1: Create a GitHub Account

If you don't already have one, create a free account at <https://github.com>.

- Choose a professional username (you may use this for years)
- Add a profile picture and brief bio if you like

5.2 Task 2: Create a Repository

Create a new **public** repository for this course. Call it something like aqmss2 or quant-methods-2026.

Via the web interface:

1. Click the "+" icon in the top right, then "New repository"
2. Enter a name and description
3. Make sure "Public" is selected
4. Check "Add a README file"
5. Click "Create repository"

Via the command line:

```
mkdir aqmss2
cd aqmss2
git init
echo "# AQMSS II" > README.md
git add README.md
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/YOUR-USERNAME/aqmss2.git
git push -u origin main
```

5.3 Task 3: Edit the README

The README file is the “front page” of your repository. Edit it to include:

- Your name
- A brief description of the repository (e.g., “Problem sets for AQMSS II, Spring 2026”)
- Optionally, a table of contents or list of files

Via the web: Click on README.md, then the pencil icon to edit. When done, scroll down and click “Commit changes.”

Via command line: Edit the file locally, then:

```
git add README.md
git commit -m "Update README with course info"
git push
```

5.4 Task 4: Create a Folder for Problem Sets

Create a folder called `problem_sets` (or similar) in your repository. Inside it, create a file called `ps1.R` with a comment header:

```
# Problem Set 1
# AQMSS II, Spring 2026
# [Your Name]

# This file will contain my solutions for PS1
```

Commit this file to your repository.

5.5 Task 5: Explore the History

Look at the commit history of your repository:

Via the web: Click on “Commits” (or the clock icon) to see all your commits.

Via command line:

```
git log --oneline
```

Take a screenshot of your commit history showing at least 2–3 commits.

6 Submission

Send me an email with:

1. The URL of your GitHub repository (e.g., <https://github.com/username/aqmss2>)

2. The screenshot of your commit history

I will check that your repository is public and contains a README and at least one .R file.

7 Optional: Setting Up Git Locally

If you want to use Git from the command line or RStudio, you'll need to set it up on your computer:

1. **Install Git:** Download from <https://git-scm.com/downloads>
2. **Configure your identity:**

```
git config --global user.name "Your Name"
git config --global user.email "your@email.com"
```

3. **Clone your repository** (download it to your computer):

```
git clone https://github.com/YOUR-USERNAME/aqmss2.git
```

For RStudio users: Go to File → New Project → Version Control → Git, and paste your repository URL. RStudio will clone the repository and set up a project for you.

8 Resources

- GitHub's official guides: <https://docs.github.com/en/get-started>
- Happy Git with R (excellent for R users): <https://happygitwithr.com>
- Git cheat sheet: <https://education.github.com/git-cheat-sheet-education.pdf>
- Pro Git book (free online): <https://git-scm.com/book/en/v2>

We will cover more advanced Git workflows and reproducible computing practices later in the course (Session 5). For now, just get comfortable with the basics!