

R Version Control

Vincent Arumadri

2025-07-08

R Version Control with Git/GitHub

Git/GitHub enables continuous software/code development by tracking code changes to the **initial commit**, tracking who made the changes and enabling code collaboration. Here I will go through the basics on how to step-up Git/GitHub for version control with R/RStudio.

Required programs

To start off, ensure you have the following programs installed:

1. **Git** – the base program for tracking code changes.
 - **Windows:** You can download Git [here](#). Double-click the downloaded .exe file to start the installation process.
 - **macOS:** First you need to install **Homebrew**. Copy and paste the Homebrew download link from (<https://brew.sh/>) to the Terminal and run.

If you already have Homebrew installed, open Terminal and type:

```
brew install git
```

2. **R** – the base R program
 - **Windows:** You can download R [here](#)
 - **macOS:** You can download R [here](#)
3. **RStudio** – IDE (integrated development environment) for R
 - **Windows and macOS:** You can download RStudio [here](#)
4. **GitHub Desktop** – for version control and collaboration
 - **Windows and macOS:** You can download GitHub Desktop [here](#)

Getting started

1. **Sign up for GitHub** – Got to GitHub and create a free account:

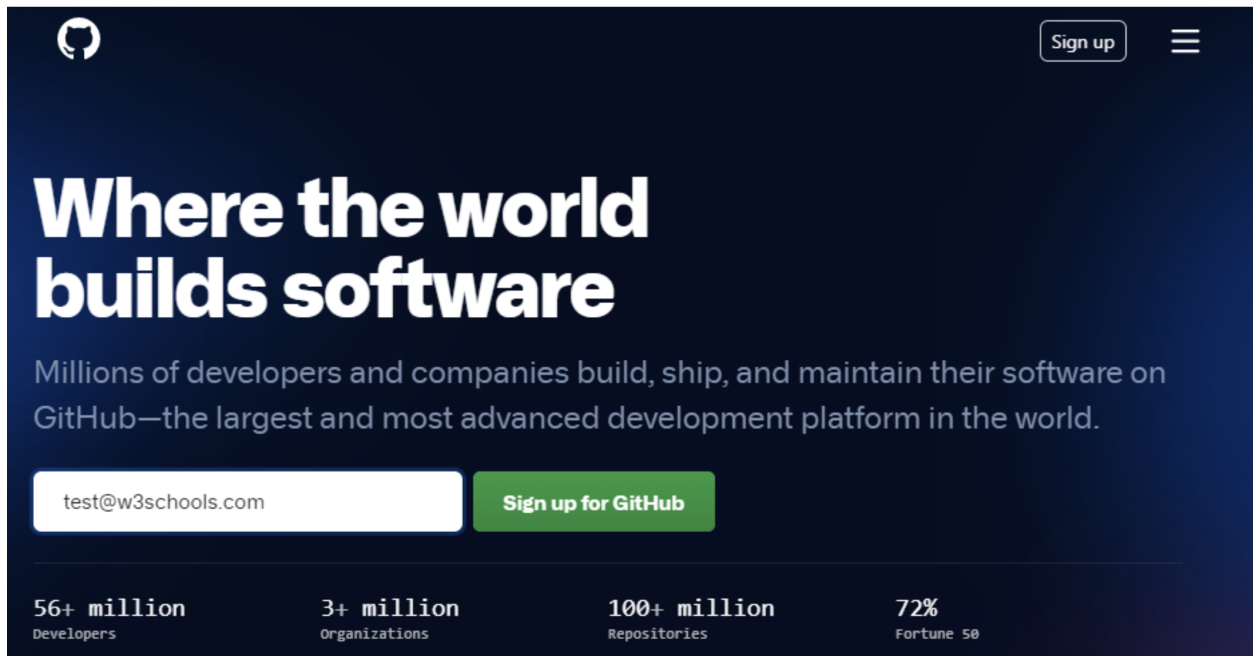
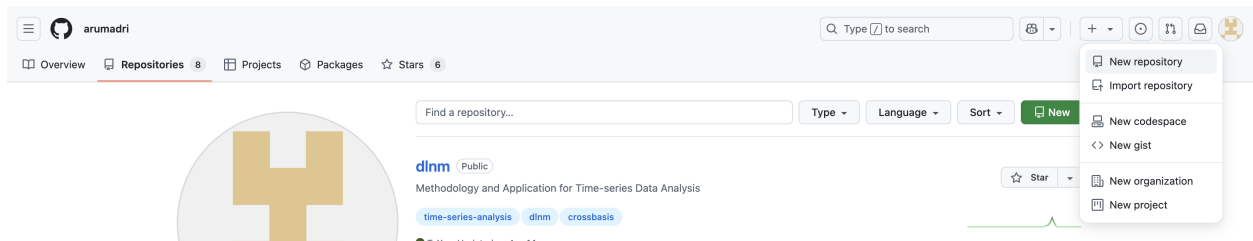


Figure 1: GitHub account setup

2. **Create a Repository** After signing in, click the + button to create a new repository:



3. **Fill in the repository details** (name, description, public/private, etc.) and click Create repository:

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner * **Repository name ***

arumadri / jrc_template

jrc_template is available.

Great repository names are short and memorable. Need inspiration? How about ubiquitous-system ?

Description (optional)

R Version control using GitHub

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

This will set `main` as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

Create repository

Figure 2: GitHub new repository details

Now you have set up your **remote repository**!!

4. Clone remote repository to local machine (laptop/desktop)

Inside the repository click the <>Code button and select **Open with GitHub Desktop** in the drop down menu.

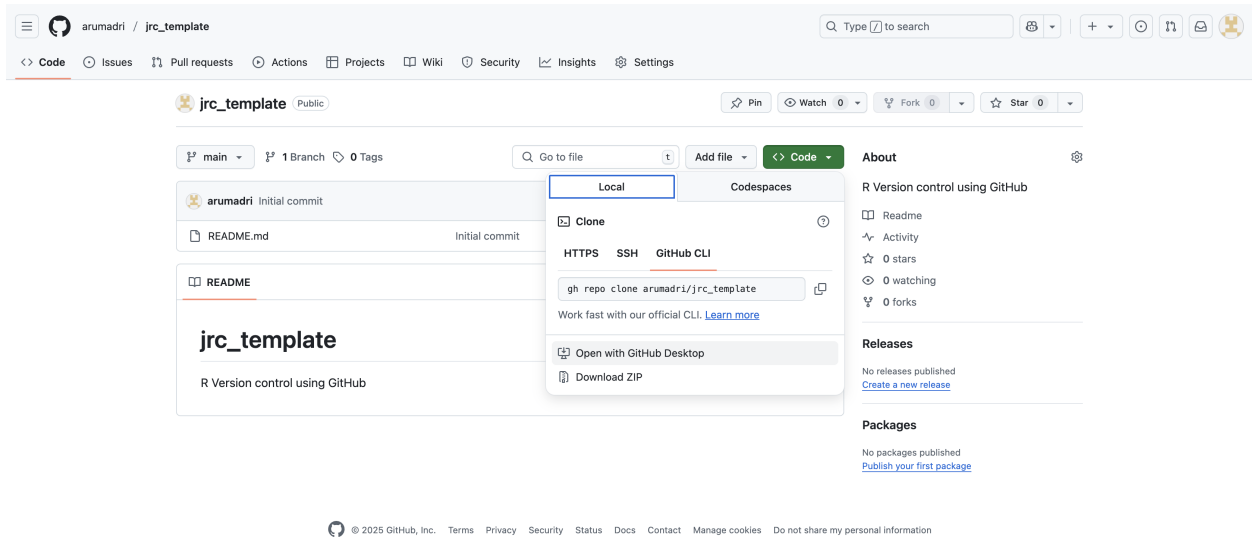


Figure 3: Clone remote to local

5. Select path for local repository

Clone a Repository

GitHub.com GitHub Enterprise **URL**

Repository URL or GitHub username and repository
(hubot/cool-repo)

Local Path

< > jrc_template

Name	Date Modified	Size	Kind
README.md	Today at 13:47	47 bytes	Markdown File

6. Add or (save your new r scripts/files) to this folder

This will initialize the **commit** and **push** prompt in GitHub Desktop to commit and push the changes respectively to the **remote repository**.

These prompts enable changes made to the folder (adding new files) or files (new edits to code) on the **local** machine to be **committed** and **pushed** to the **remote** for tracking and hence **version control**.

**

The screenshot displays the GitHub Desktop application interface at the top, showing the local repository 'jrc_template' with a list of files including 'clone_to_local', 'github_account.png', 'new_repository', 'on_local', 'path_on_local', 'r_version_control.pdf', 'r_version_control.Rmd', 'README.md', and 'repository_details'. Below this, the GitHub web interface is shown, displaying the repository page for 'jrc_template' by 'arumadri'. The page includes a 'Clone' button, a 'Code' button, and a 'README' section. A 'Clone' modal is open, showing options to clone the repository using HTTPS, SSH, or GitHub CLI. The modal also includes a 'Local' section with a 'Clone' button and a 'Codespaces' section with a 'Clone' button. The 'README' section shows the repository's purpose: 'R Version control using GitHub'.

Name	Date Modified	Size	Kind
clone_to_local	Today at 13:22	370 KB	PNG image
github_account.png	Today at 12:38	736 KB	PNG image
new_repository	Today at 12:53	195 KB	PNG image
on_local	Today at 13:47	50 KB	PNG image
path_on_local	Today at 13:45	145 KB	PNG image
r_version_control.pdf	Today at 13:55	1,6 MB	PDF Document
r_version_control.Rmd	Today at 13:54	3 KB	R Markdown File
README.md	Today at 13:47	47 bytes	Markdown File
repository_details	Today at 13:32	620 KB	PNG image

Current Repository: jrc_template
Current Branch: main
Fetch origin: Last fetched 1 minute ago

Changes: 9 changed files
History: clone_to_local.png

Added

arumadri / jrc_template

main 1 Branch 0 Tags

Go to file

Add file

Code

About

R Version control using GitHub

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Clone

Local

Codespaces

HTTPS SSH GitHub CLI

gh repo clone arumadri/jrc_template

Work fast with our official CLI. [Learn more](#)

Open with GitHub Desktop

Download ZIP

jrc_template

R Version control using GitHub

© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

7. Push and pull changes to or from GitHub

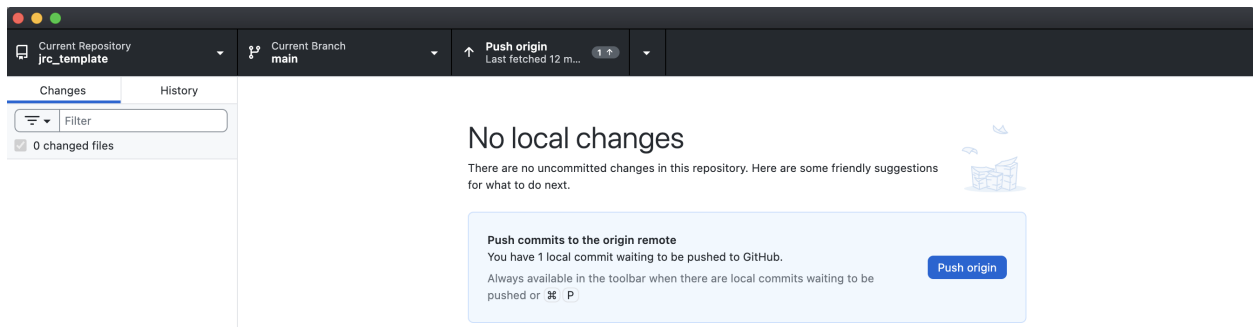
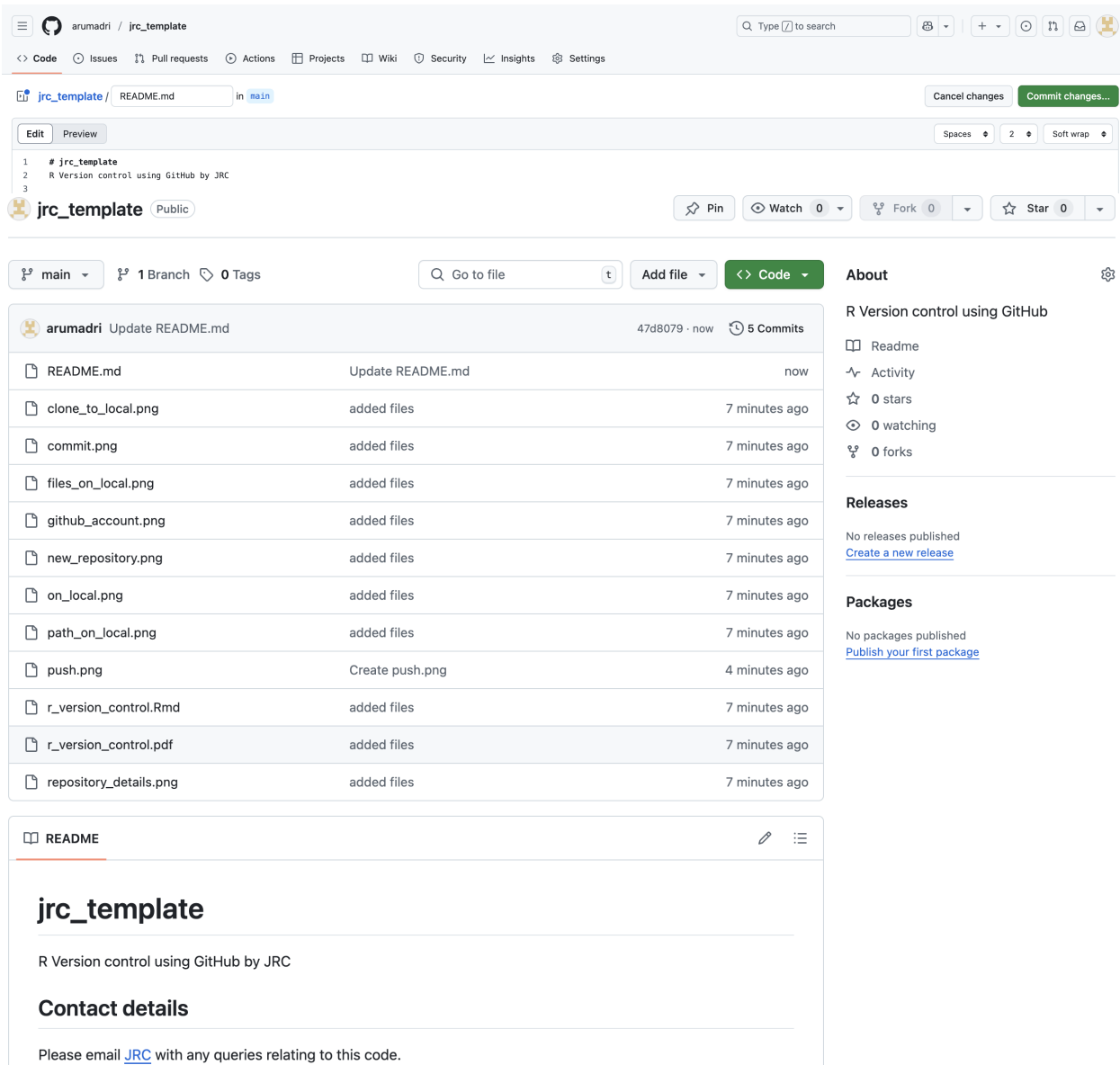


Figure 4: Push



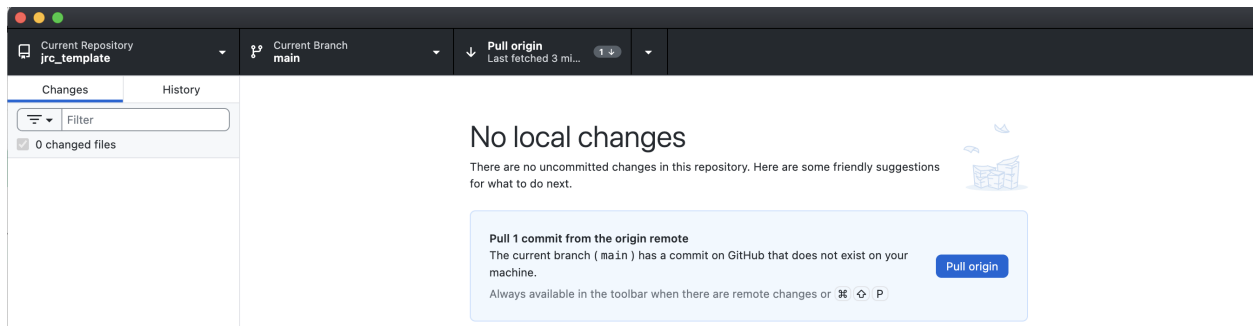


Figure 5: Pull to local

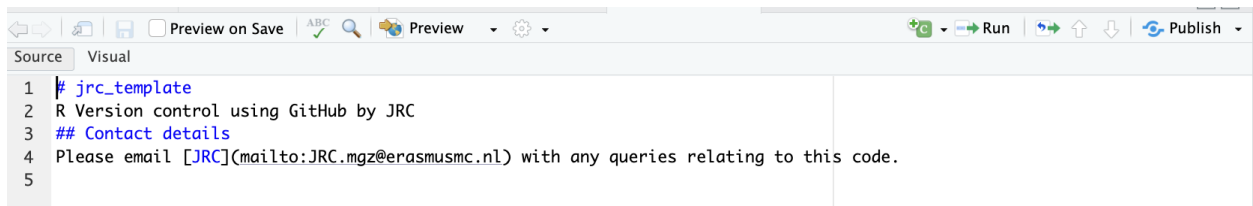


Figure 6: Commit changes on local