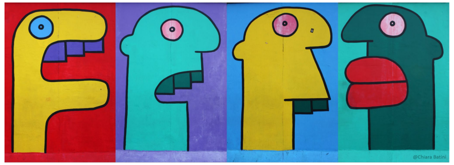
# Reproducible science in action

# Setting up git for practicals

Google scholar

Franco Marsico

Github: https://github.com/MarsicoFL



EMBO POPGEN

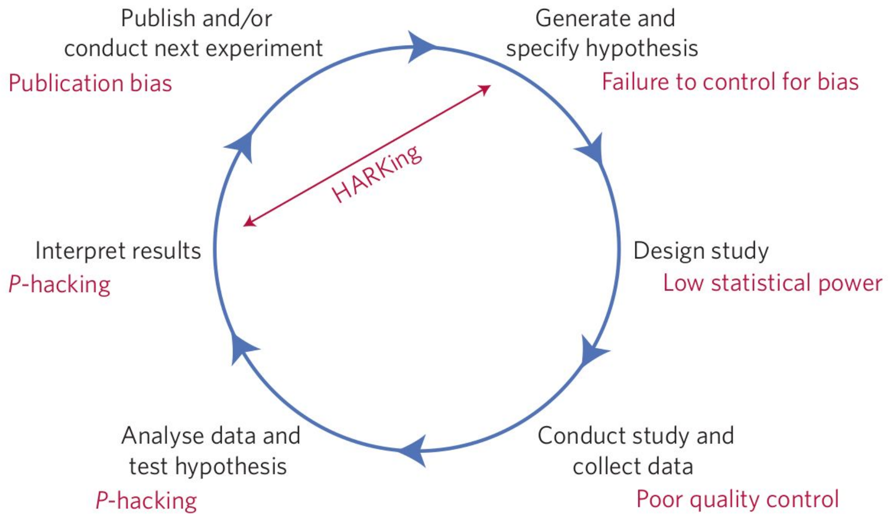
## Overview

- What is reproducible science?
- Code reproducibility
- A conceptual introduction to Git
- Using GitHub repositories
- Practical example: Setting up RStudio with GitHub
- Using git in the command line

# Reproducible Science

**Definition**: Reproducible science refers to the ability to replicate the results of a scientific study using the same methods, data, and conditions. Its application is expected to:

- Ensures the credibility and reliability of scientific findings.
- Enhances transparency and trust in research.
- Facilitates the efficient accumulation and validation of knowledge.


Publish and/or conduct next experiment
Publication bias
Generate and specify hypothesis
Failure to control for bias
HARKing
Interpret results
*P*-hacking
Design study
Low statistical power
Analyse data and test hypothesis
*P*-hacking
Conduct study and collect data
Poor quality control

Munafò, M. R., Nosek, B. A., Bishop, D. V., Button, K. S., Chambers, C. D., Percie du Sert, N., ... & Ioannidis, J. (2017). A manifesto for reproducible science. *Nature human behaviour*, *1*(1), 1-9.

# What we can do?

**Transparency and Open Science:**

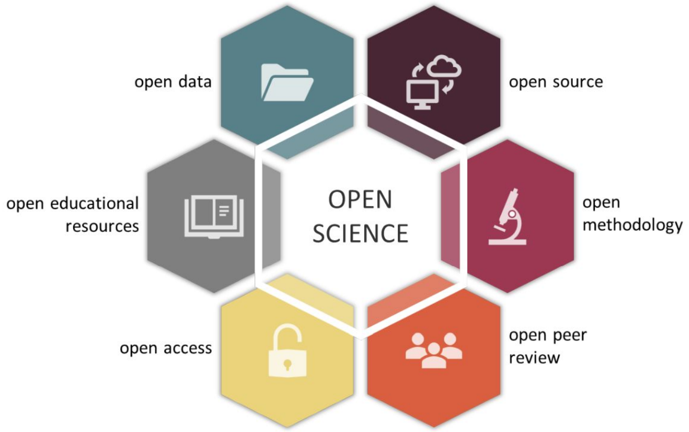
- Share data, materials, and code openly.
- Use open-access repositories and platforms like GitHub.

**Blinding and Methodological Rigor:**

- Implement blinding to reduce bias.
- Employ rigorous and standardized methods.

**Collaboration and Team Science:**

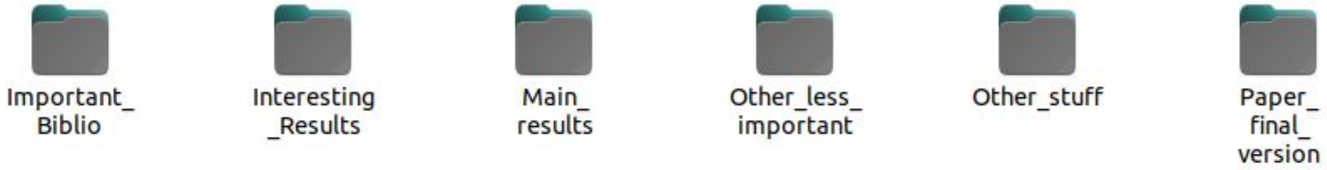
- Foster collaborative efforts and multi-site studies to enhance statistical power and generalizability.



Gallagher, R. V., Falster, D. S., Maitner, B. S., Salguero-Gómez, R., Vandvik, V., Pearse, W. D., ... & Enquist, B. J. (2020). Open Science principles for accelerating trait-based science across the Tree of Life. Nature ecology & evolution, 4(3), 294-303.

## Sharing the code

It is not enough to share the data and the code



Interpretability and traceability become central

## Good practices when sharing the code

**Documentation:**

- Provide clear and comprehensive documentation.
- Include a README file with instructions on how to use the code.

**Licensing:**

- Choose an appropriate open-source license (e.g., MIT, GPL).

**Version Control:**

- Use Git for version control to track changes and collaborate efficiently.
- Create a well-organized repository structure.

**Accessibility:**

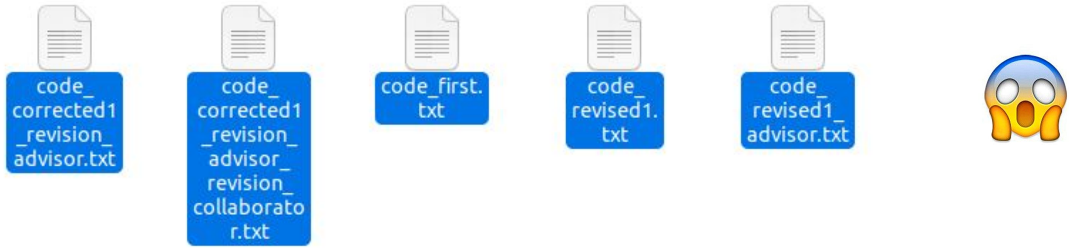
- Share your code on platforms like GitHub or GitLab.

**Examples and Tests:**

- Include example datasets and usage examples.
- Provide tests to ensure the code works as intended.

https://docs.github.com/es

# An introduction to Git

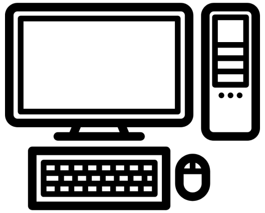

code_corrected1_revision_advisor.txt
code_corrected1_revision_advisor_revision_collaborator.txt
code_first.txt
code_revised1.txt
code_revised1_advisor.txt

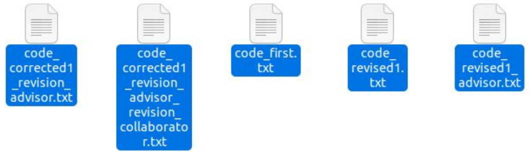
**VCS:** Version Control Systems are designed to organize collaborative software development

- They store all revisions of each file.
- They allow us to switch between versions and view differences.
- Each revision must be explicitly created with a message indicating the changes made.
- Distributed: Each user works independently on their own computer but can share their modifications with others.

# Git: Key concepts

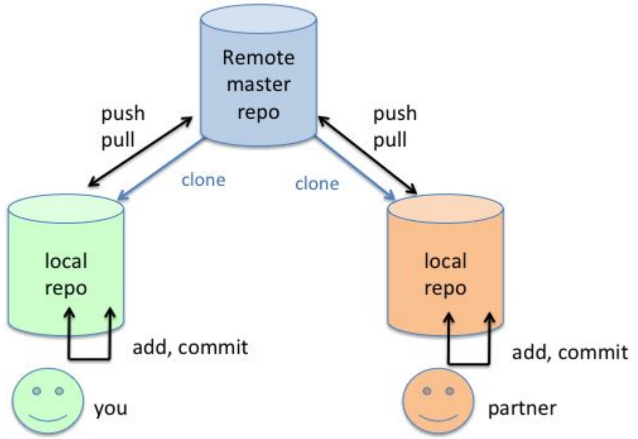
- **Local repository**: The place where the files of a project are stored, along with all the additional information needed for version control.
- **Revision**: A snapshot of the repository at a given moment.
- **History**: A set of revisions ordered chronologically.
- **Remote repository**: A server to which Git sends changes when a push is executed.



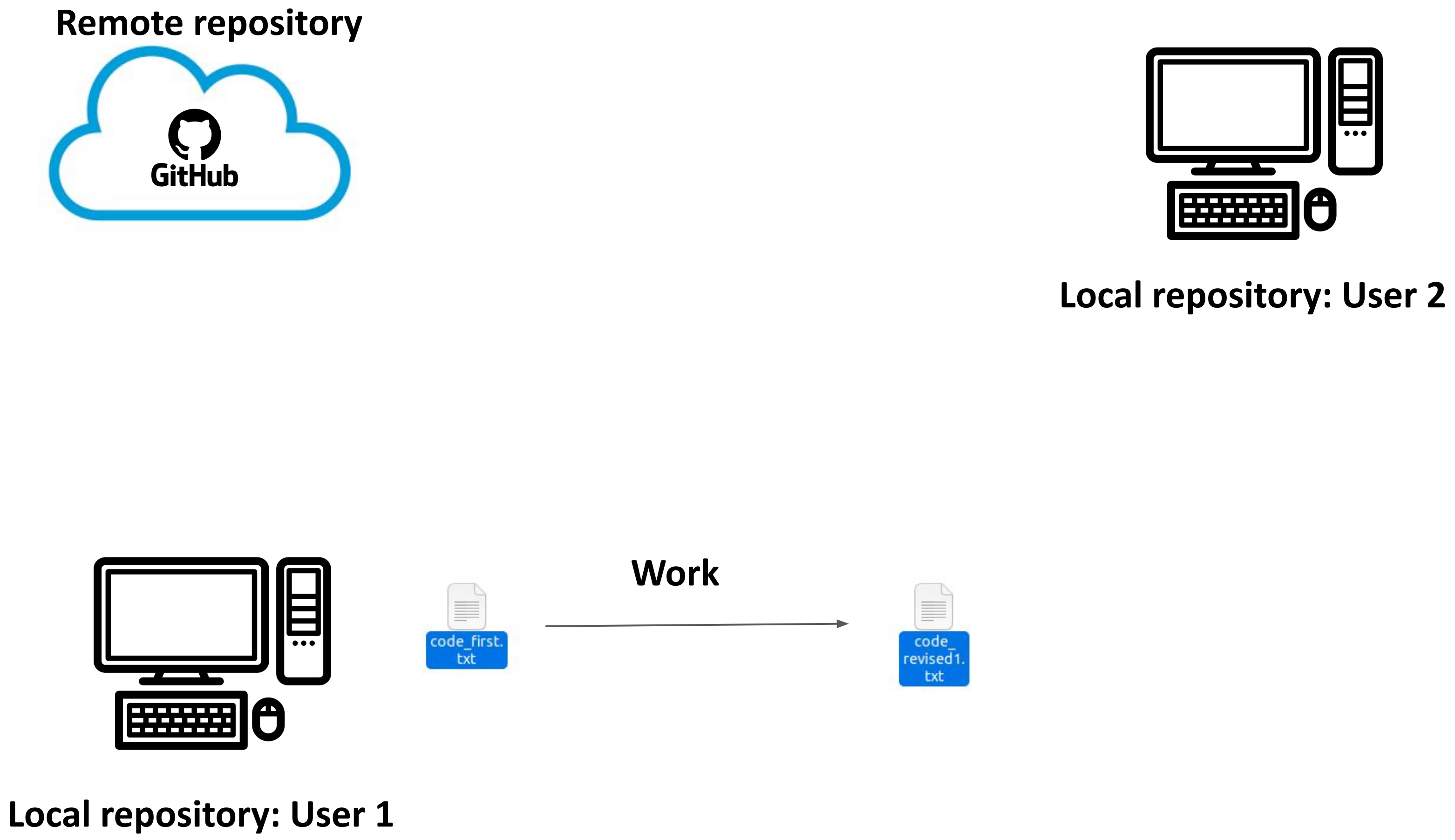

code_corrected1_revision_advisor.txt
code_corrected1_revision_advisor_revision_collaborator.txt
code_first.txt
code_revised1.txt
code_revised1_advisor.txt

# Git actions

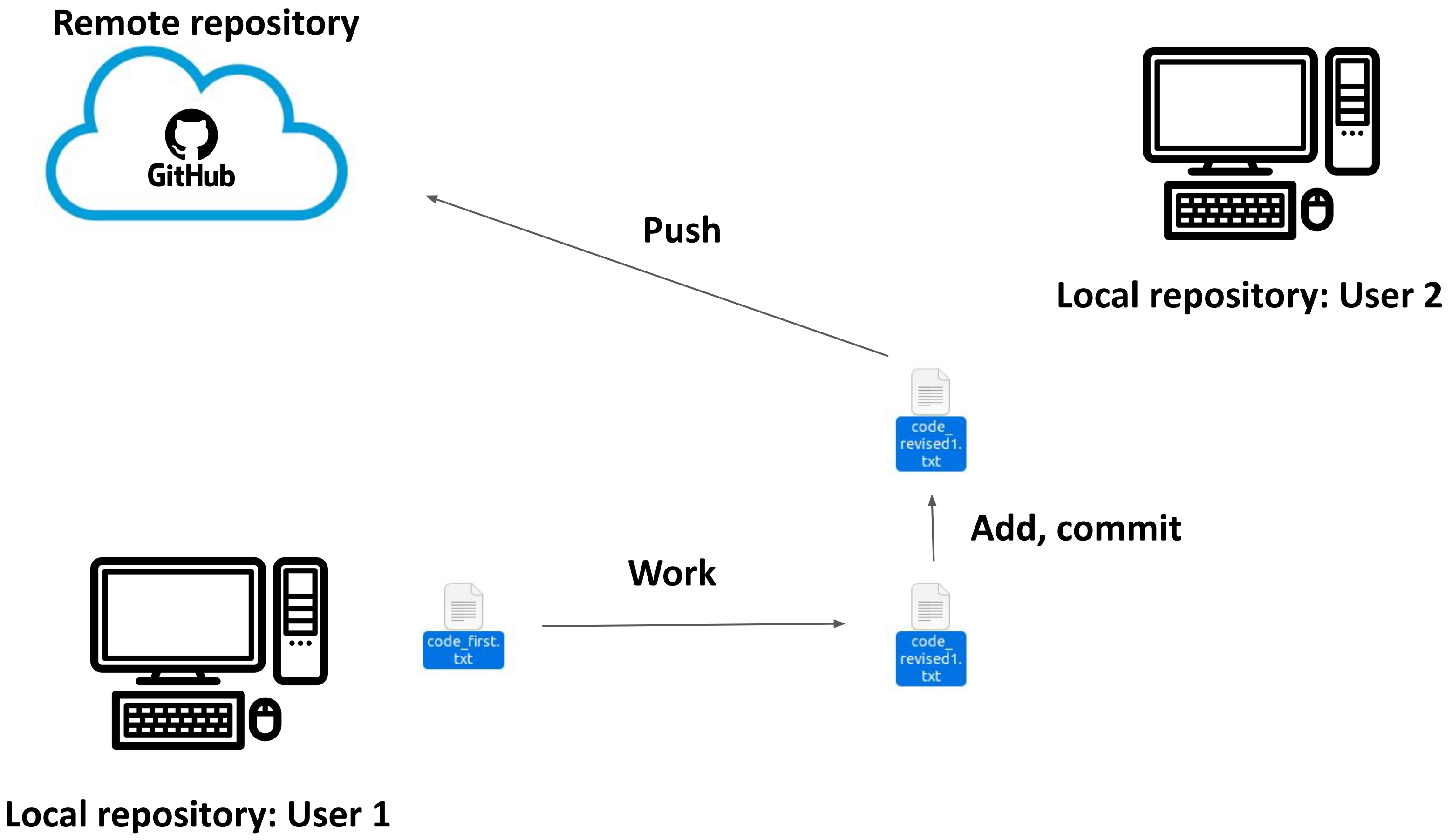
- **add**: Adds a file to the revision.
- **commit**: Creates a new revision with a message describing changes.
- **clone**: Downloads a repository and its history.
- **push**: Shares local commits with a remote server or user.
- **pull**: Updates local repository with changes from others.
- **merge**: Combines local changes with others' changes.
- **status**: Shows the state of the local repository.



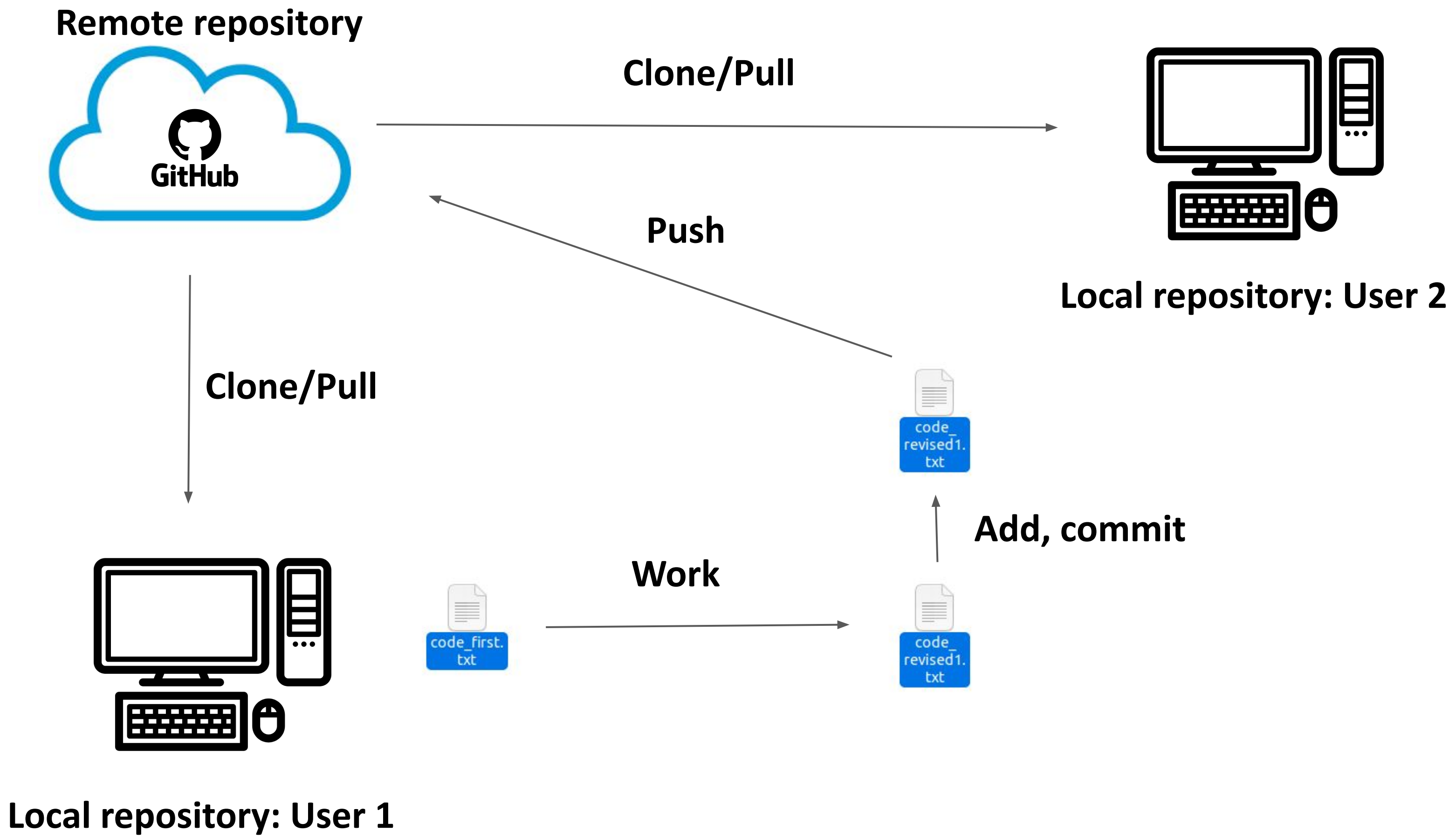
## Git actions


Remote repository
GitHub
Local repository: User 2
Work
code_first.txt
code_revised1.txt
Local repository: User 1

# Git actions


Remote repository
GitHub
Push
Local repository: User 2
code_revised1.txt
Add, commit
Work
code_first.txt
code_revised1.txt
Local repository: User 1

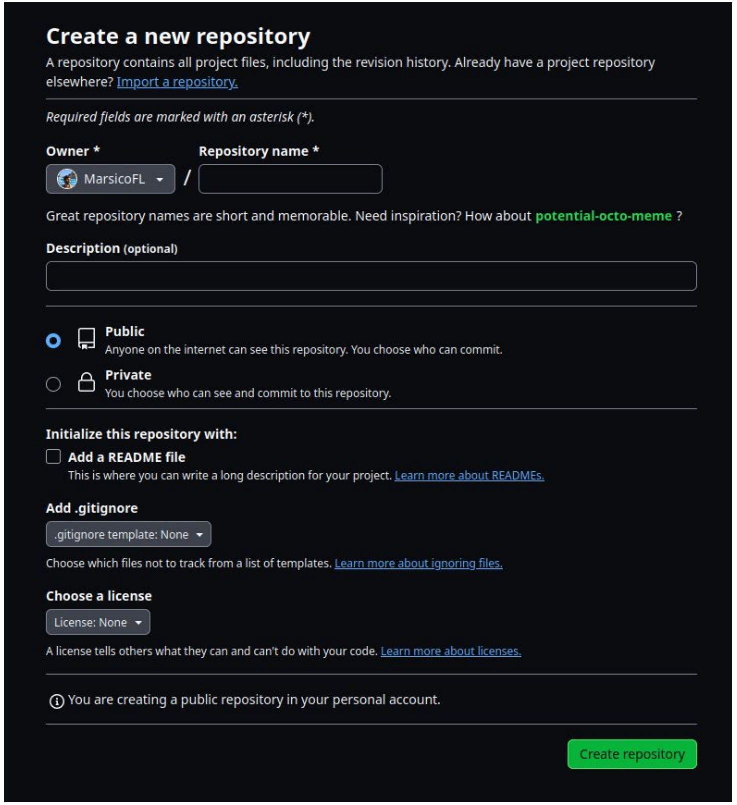
# Git actions


Remote repository
GitHub
Clone/Pull
Push
Local repository: User 2
Clone/Pull
code_revised1.txt
Add, commit
Work
code_first.txt
code_revised1.txt
Local repository: User 1

# Using GitHub repositories

- Several services offer Git repository hosting:
  - Provide a convenient web interface for:
    - Creating repositories
    - Managing user access
    - Viewing files
    - Comparing revisions
    - Viewing history
- Well-known services include:
  - GitHub
  - GitLab
  - Bitbucket

Repositories > New



**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 *: MarsicoFL / Repository name *:

Great repository names are short and memorable. Need inspiration? How about **potential-octo-meme** ?

Description (optional)

[✓ 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.

You are creating a public repository in your personal account.

## Using GitHub repositories

Code

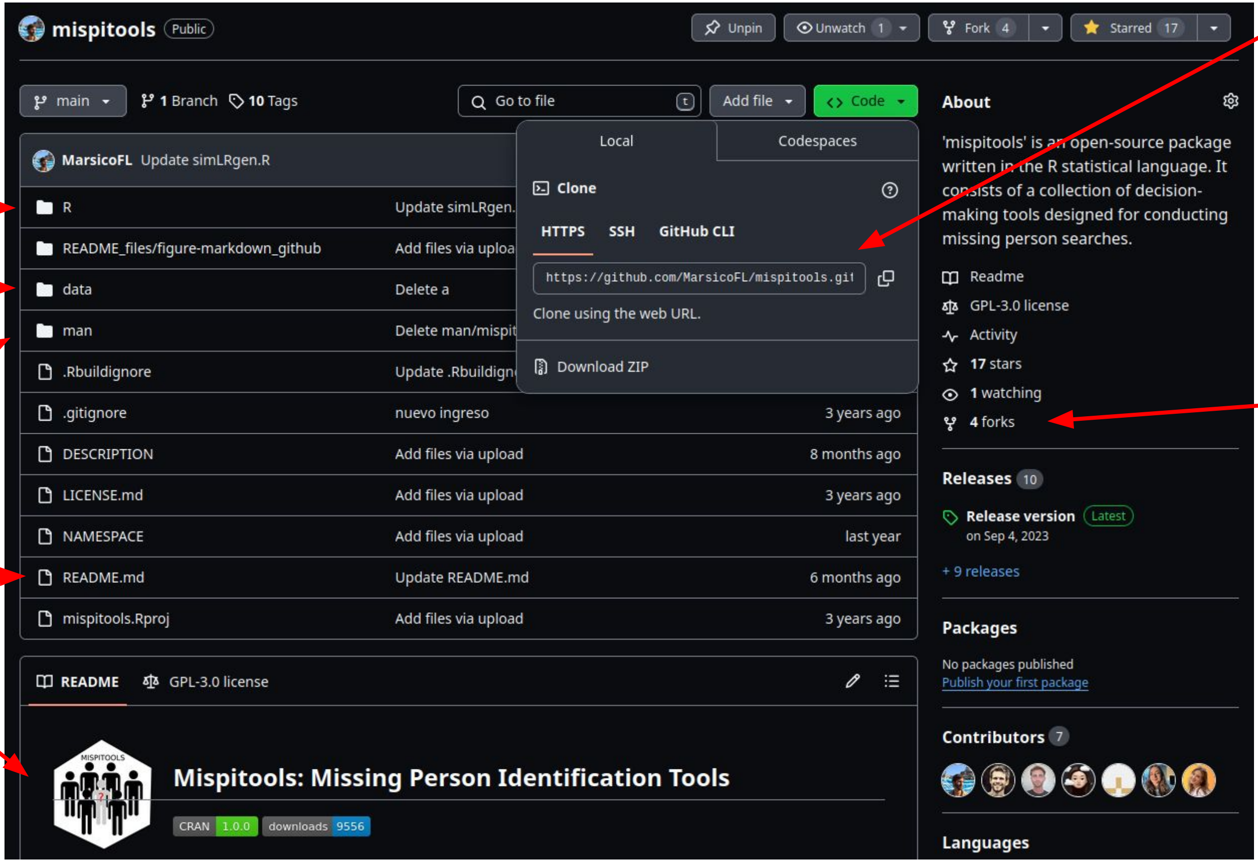
Data

Code and data documentation

General documentation

Path

Copies



mispitools Public

MarsicoFL Update simLRgen.R

| | | |
|---|---|---|
| R | Update simLRgen. | |
| README_files/figure-markdown_github | Add files via uploa | |
| data | Delete a | |
| man | Delete man/mispit | |
| .Rbuildignore | Update .Rbuildign | |
| .gitignore | nuevo ingreso | 3 years ago |
| DESCRIPTION | Add files via upload | 8 months ago |
| LICENSE.md | Add files via upload | 3 years ago |
| NAMESPACE | Add files via upload | last year |
| README.md | Update README.md | 6 months ago |
| mispitools.Rproj | Add files via upload | 3 years ago |

Clone

```
https://github.com/MarsicoFL/mispitools.git
```

Clone using the web URL.

Download ZIP

**About**

'mispitools' is an open-source package written in the R statistical language. It consists of a collection of decision-making tools designed for conducting missing person searches.

Readme
GPL-3.0 license
Activity
17 stars
1 watching
4 forks

**Releases** 10

Release version Latest
on Sep 4, 2023

+ 9 releases

**Packages**

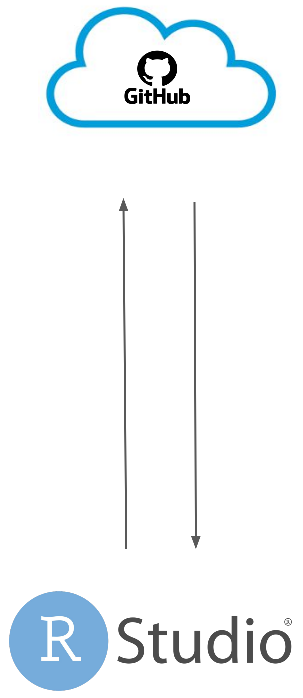
No packages published
Publish your first package

**Contributors** 7

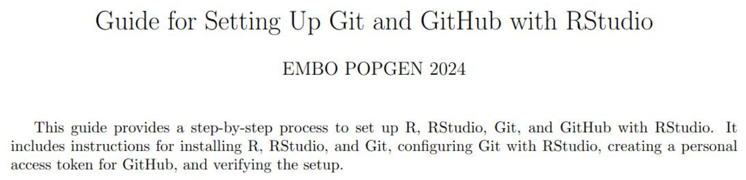
**Languages**

**Mispitools: Missing Person Identification Tools**

# Practical example: Setting up RStudio with GitHub



## 1 - Setting Up the key



Guide for Setting Up Git and GitHub with RStudio

EMBO POPGEN 2024

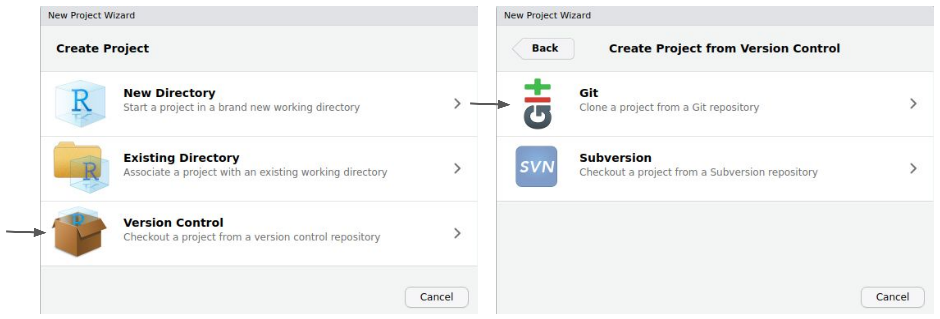
This guide provides a step-by-step process to set up R, RStudio, Git, and GitHub with RStudio. It includes instructions for installing R, RStudio, and Git, configuring Git with RStudio, creating a personal access token for GitHub, and verifying the setup.

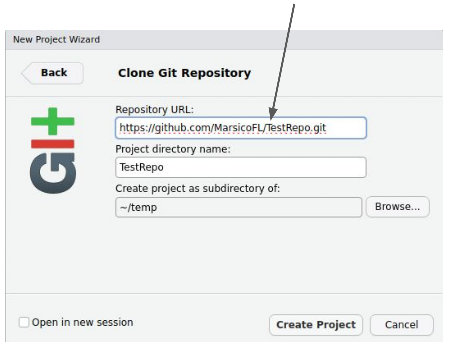
https://github.com/ColonnaLab/EMBO_popgen/blob/main/popgen2024/Franco_Marsico/Tutorial_RStudio_Github.pdf

## 2 - Open a new git project

RStudio: File > New Project

Your github repo


New Project Wizard
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel
New Project Wizard
Back
Create Project from Version Control
Git
Clone a project from a Git repository
SVN
Subversion
Checkout a project from a Subversion repository
Cancel



New Project Wizard

Back | Clone Git Repository

Repository URL: https://github.com/MarsicoFL/TestRepo.git

Project directory name: TestRepo

Create project as subdirectory of: ~/temp | Browse...

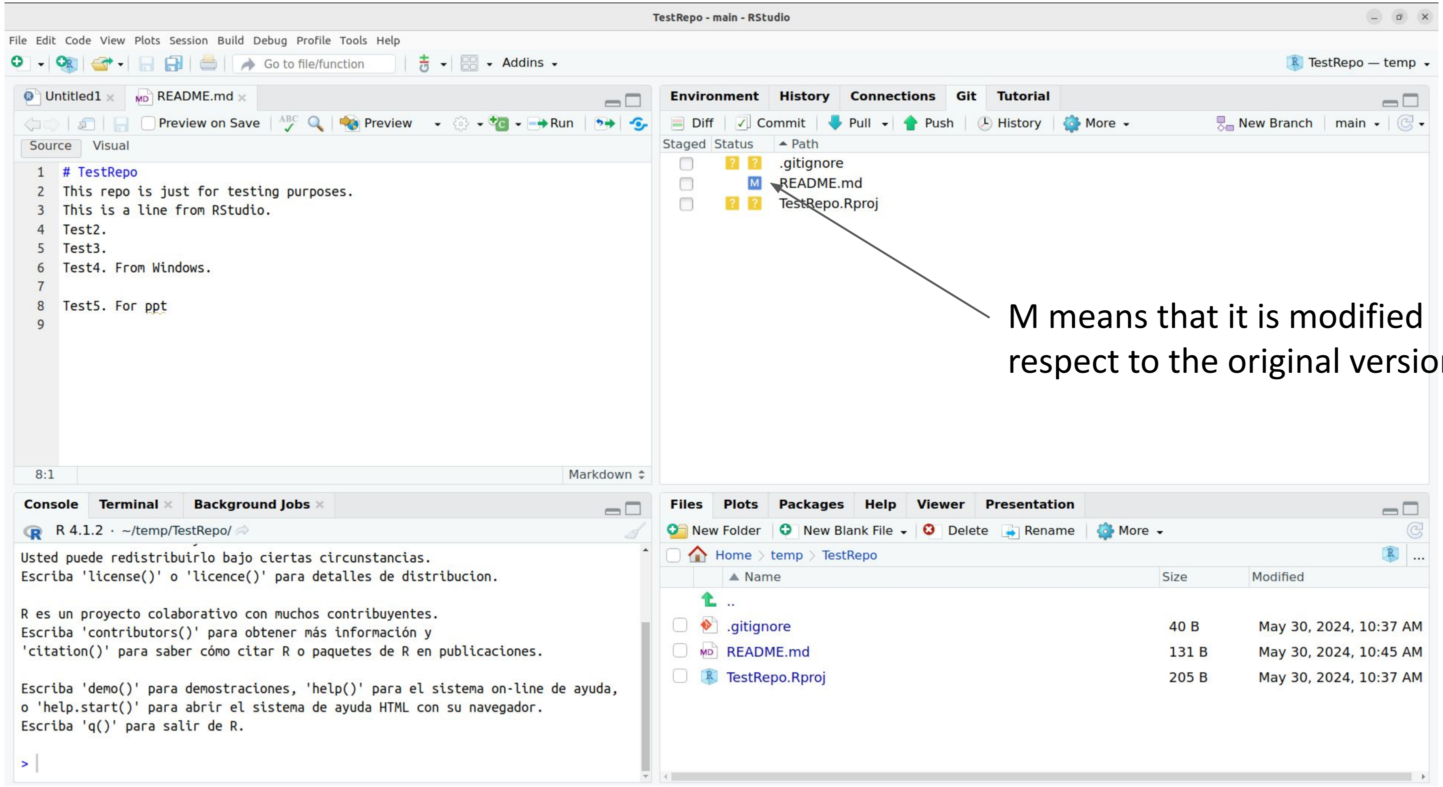
☐ Open in new session

Create Project | Cancel

# Practical example: Setting up RStudio with GitHub

## 3 - Open and modify files

Panel showing git actions


TestRepo - main - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
TestRepo — temp
Untitled1
README.md
Preview on Save
Preview
Run
Source
Visual
8:1
Markdown
Environment History Connections Git Tutorial
Diff Commit Pull Push History More
New Branch
main
Staged Status Path
Console Terminal Background Jobs
R 4.1.2 · ~/temp/TestRepo/
Files Plots Packages Help Viewer Presentation
New Folder New Blank File Delete Rename More
Home › temp › TestRepo

```
# TestRepo
This repo is just for testing purposes.
This is a line from RStudio.
Test2.
Test3.
Test4. From Windows.

Test5. For ppt
```

| Staged | Status | Path |
|---|---|---|
| ☐ | ? ? | .gitignore |
| ☐ | M | README.md |
| ☐ | ? ? | TestRepo.Rproj |

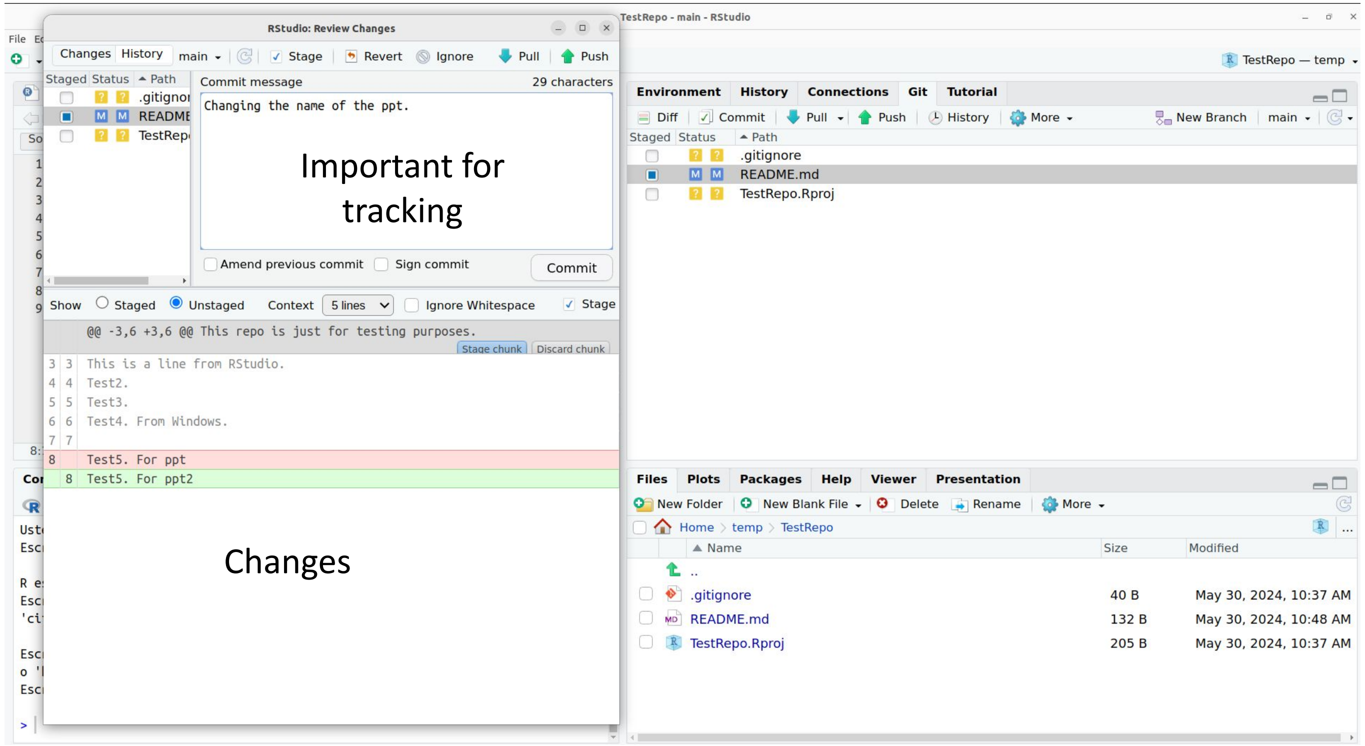
```
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribucion.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para obtener más información y
'citation()' para saber cómo citar R o paquetes de R en publicaciones.

Escriba 'demo()' para demostraciones, 'help()' para el sistema on-line de ayuda,
o 'help.start()' para abrir el sistema de ayuda HTML con su navegador.
Escriba 'q()' para salir de R.

>
```

| Name | Size | Modified |
|---|---|---|
| .. | | |
| .gitignore | 40 B | May 30, 2024, 10:37 AM |
| README.md | 131 B | May 30, 2024, 10:45 AM |
| TestRepo.Rproj | 205 B | May 30, 2024, 10:37 AM |

M means that it is modified respect to the original version

# Practical example: Setting up RStudio with GitHub

## 4 - Commit selecting modified files



Commit message:

```
Changing the name of the ppt.
```

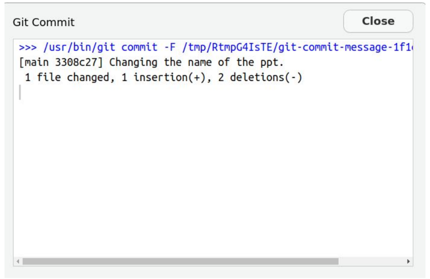
Important for tracking

```
@@ -3,6 +3,6 @@ This repo is just for testing purposes.
3 3  This is a line from RStudio.
4 4  Test2.
5 5  Test3.
6 6  Test4. From Windows.
7 7
8    Test5. For ppt
  8  Test5. For ppt2
```

Changes

| Name | Size | Modified |
|---|---|---|
| .. | | |
| .gitignore | 40 B | May 30, 2024, 10:37 AM |
| README.md | 132 B | May 30, 2024, 10:48 AM |
| TestRepo.Rproj | 205 B | May 30, 2024, 10:37 AM |

After commit this message appears

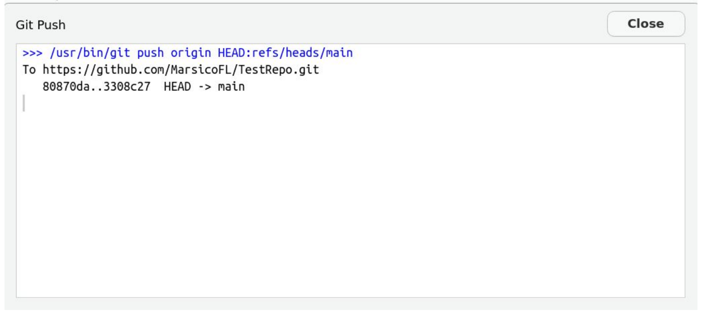


```
>>> /usr/bin/git commit -F /tmp/RtmpG4IsTE/git-commit-message-1f1
[main 3308c27] Changing the name of the ppt.
 1 file changed, 1 insertion(+), 2 deletions(-)
```

# Practical example: Setting up RStudio with GitHub

5 - Push

Git panel > Push

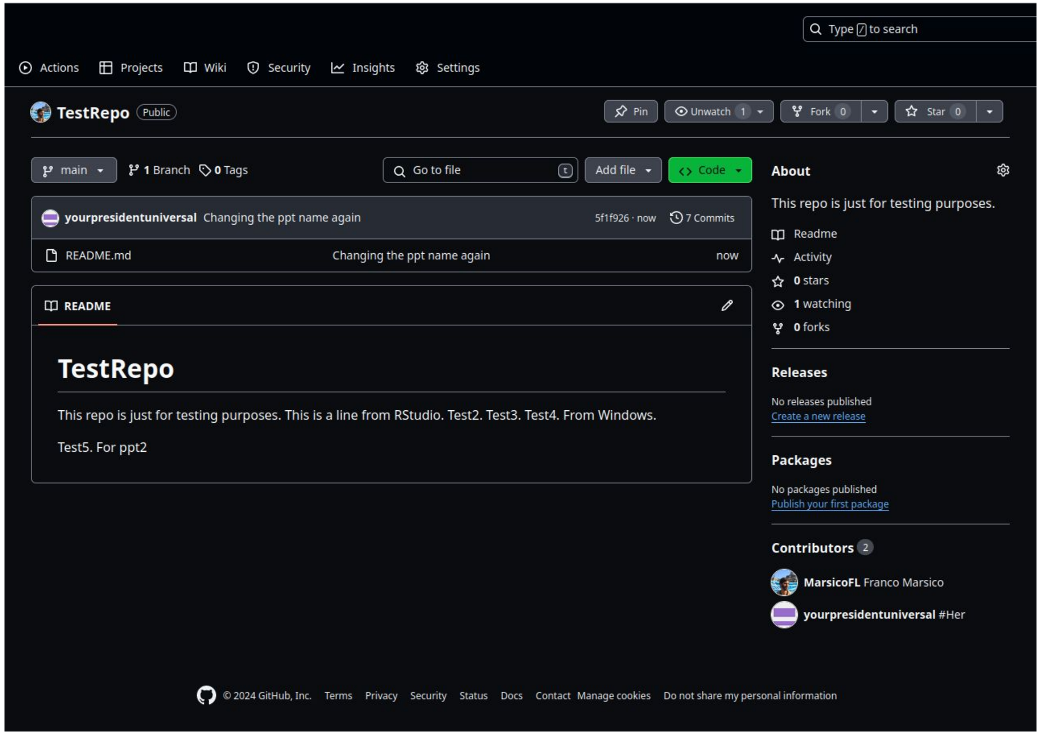


```
>>> /usr/bin/git push origin HEAD:refs/heads/main
To https://github.com/MarsicoFL/TestRepo.git
   80870da..3308c27  HEAD -> main
```

6 - Pull

If the repository is a collaborative exercise is a good practice pull (actualize with changes from other users) after push.

Changes are incorporated in the remote repository



yourpresidentuniversal Changing the ppt name again 5f1f926 · now 7 Commits

README.md Changing the ppt name again now

This repo is just for testing purposes.

TestRepo

This repo is just for testing purposes. This is a line from RStudio. Test2. Test3. Test4. From Windows.

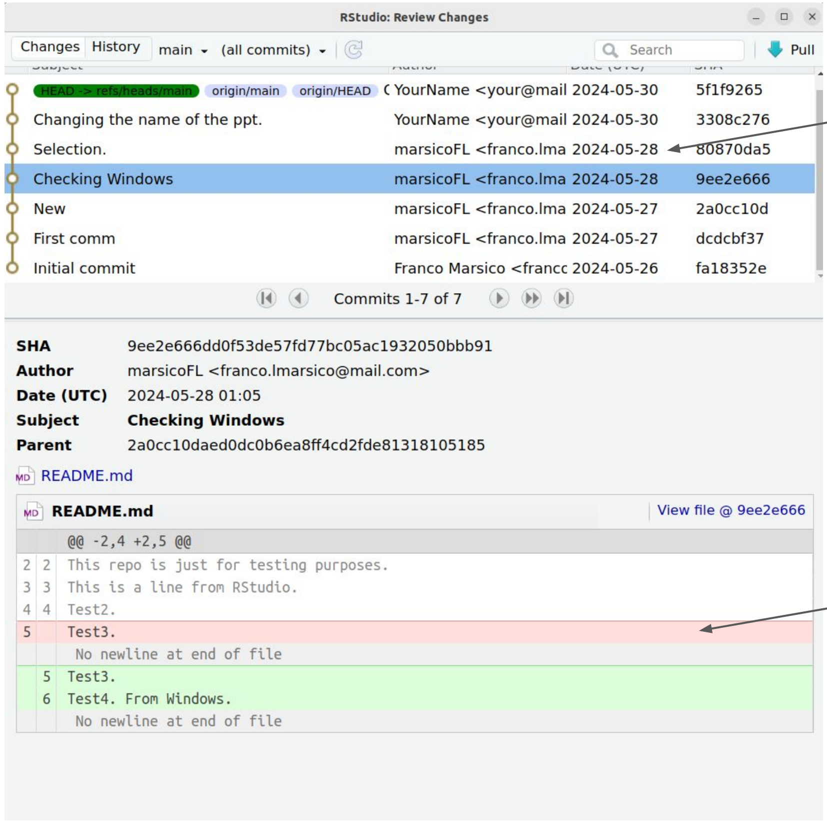
Test5. For ppt2

MarsicoFL Franco Marsico

yourpresidentuniversal #Her

# Practical example: Setting up RStudio with GitHub

## 7 - History



| | | | |
|---|---|---|---|
| HEAD -> refs/heads/main origin/main origin/HEAD ( | YourName <your@mail | 2024-05-30 | 5f1f9265 |
| Changing the name of the ppt. | YourName <your@mail | 2024-05-30 | 3308c276 |
| Selection. | marsicoFL <franco.lma | 2024-05-28 | 80870da5 |
| Checking Windows | marsicoFL <franco.lma | 2024-05-28 | 9ee2e666 |
| New | marsicoFL <franco.lma | 2024-05-27 | 2a0cc10d |
| First comm | marsicoFL <franco.lma | 2024-05-27 | dcdcbf37 |
| Initial commit | Franco Marsico <francc | 2024-05-26 | fa18352e |

| | |
|---|---|
| **SHA** | 9ee2e666dd0f53de57fd77bc05ac1932050bbb91 |
| **Author** | marsicoFL <franco.lmarsico@mail.com> |
| **Date (UTC)** | 2024-05-28 01:05 |
| **Subject** | **Checking Windows** |
| **Parent** | 2a0cc10daed0dc0b6ea8ff4cd2fde81318105185 |

README.md

**README.md**

```
        @@ -2,4 +2,5 @@
2  2    This repo is just for testing purposes.
3  3    This is a line from RStudio.
4  4    Test2.
5       Test3.
         No newline at end of file
   5    Test3.
   6    Test4. From Windows.
         No newline at end of file
```

User

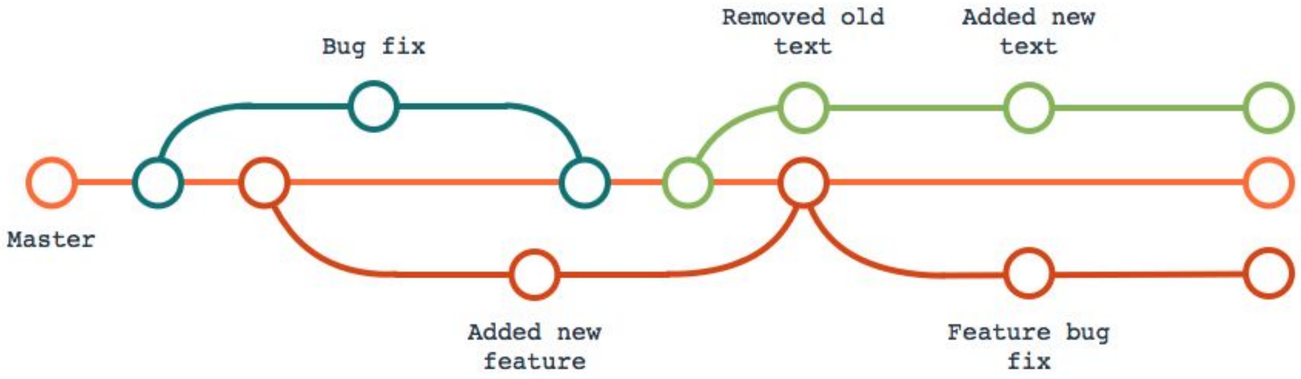
Commit message

Changes

# Practical example: Setting up RStudio with GitHub

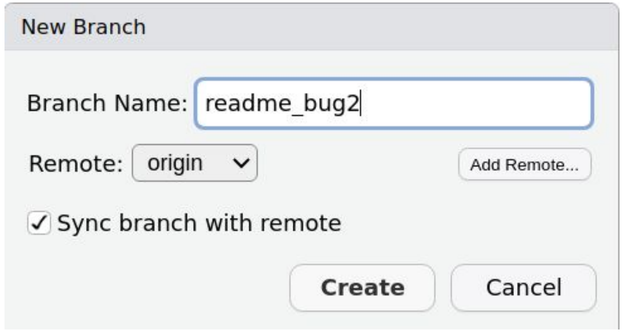
8 - Branching

- **branch**: A fork of the repository's history.
- **merge**: The action of combining the history of two different branches.
- **master**: The main branch that always exists.
- **HEAD**: The most current revision of the branch you are on.
- **checkout**: The action to switch from one branch to another (also works with commits!).



# Practical example: Setting up RStudio with GitHub

Git panel > New Branch


New Branch

Branch Name: readme_bug2

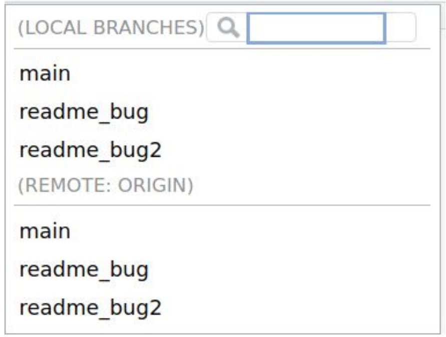
Remote: origin

Add Remote...

✓ Sync branch with remote

Create

Cancel


(LOCAL BRANCHES)

main

readme_bug

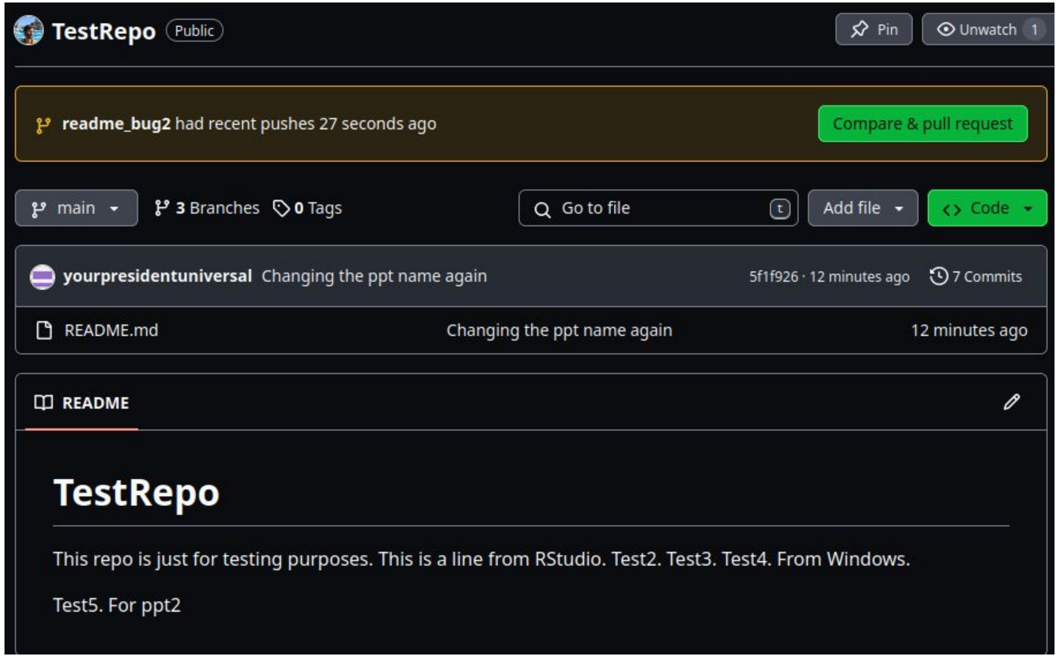
readme_bug2

(REMOTE: ORIGIN)

main

readme_bug

readme_bug2


TestRepo Public

Pin

Unwatch 1

readme_bug2 had recent pushes 27 seconds ago

Compare & pull request

main

3 Branches

0 Tags

Go to file

Add file

Code

yourpresidentuniversal Changing the ppt name again 5f1f926 · 12 minutes ago 7 Commits

README.md Changing the ppt name again 12 minutes ago

README

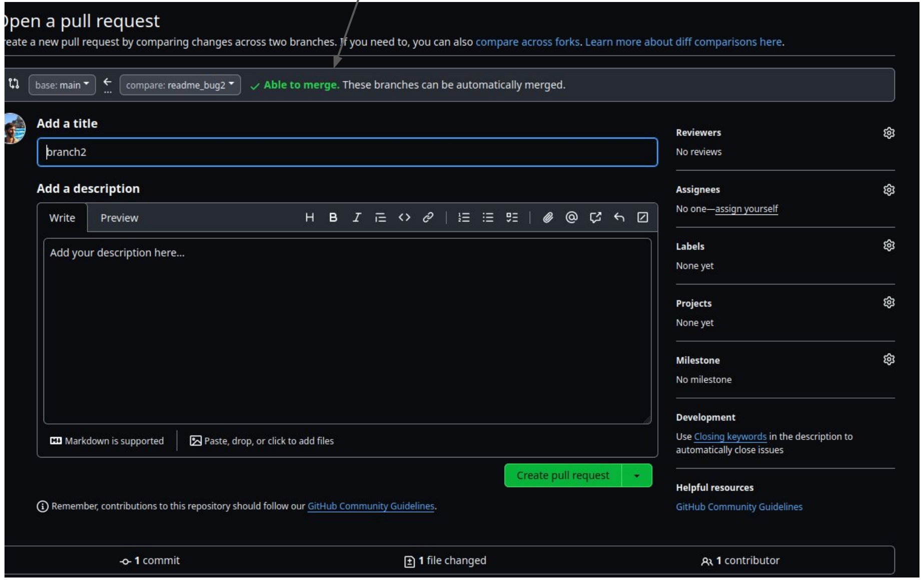
TestRepo

This repo is just for testing purposes. This is a line from RStudio. Test2. Test3. Test4. From Windows.

Test5. For ppt2

# Practical example: Setting up RStudio with GitHub

Check for conflicts



pen a pull request

eate a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff comparisons here.

base: main ← compare: readme_bug2 ✓ Able to merge. These branches can be automatically merged.

Add a title

branch2

Add a description

Add your description here...

Markdown is supported | Paste, drop, or click to add files

Create pull request

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

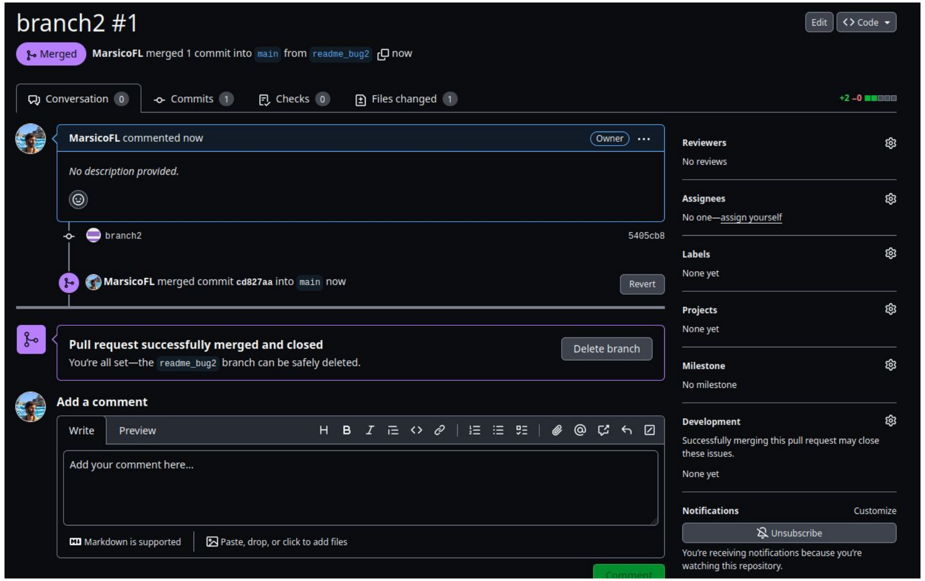
Projects: None yet

Milestone: No milestone

Development: Use Closing keywords in the description to automatically close issues

Helpful resources: GitHub Community Guidelines

1 commit | 1 file changed | 1 contributor



branch2 #1

Merged MarsicoFL merged 1 commit into `main` from `readme_bug2` now

Conversation 0 | Commits 1 | Checks 0 | Files changed 1 | +2 −0

MarsicoFL commented now (Owner)

*No description provided.*

branch2 5405cb8

MarsicoFL merged commit cd827aa into `main` now — Revert

Pull request successfully merged and closed

You're all set—the `readme_bug2` branch can be safely deleted. — Delete branch

Add a comment

Add your comment here...

Markdown is supported | Paste, drop, or click to add files

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Development: Successfully merging this pull request may close these issues. None yet

Notifications: Customize — Unsubscribe

You're receiving notifications because you're watching this repository.

## Using git in the command line

Why use the git command line?

- The RStudio GUI is very useful for GitHub, but it may fail with other clients.
- It allows working with the terminal and not in interactive mode (useful when working in HPC).
- It provides access to a full range of commands and not only those supported by the GUI.
- Avoiding the GUI is more efficient.

## Conclusions

- We have introduced key aspects of reproducible science.
- Assessed the role of sharing code in reproducibility.
- Introduced central concepts of Git and GitHub.
- Set up RStudio GUI for connecting with GitHub.
- Discussed the benefits of Git command line functions.