# Source control with git and Github

Eric Marcon

11 October 2024

# Principles

git is a source manager:

- Track changes: much more than a backup!
- Ability to go back in time;
- Several competing versions: branches.

GitHub is a collaboration platform based on Git.

- Multiple developers: the end of attached files!
- Presentation of results: GitHub pages;
- Continuous integration: GitHub Actions.

# Installation

## RStudio must detect Git
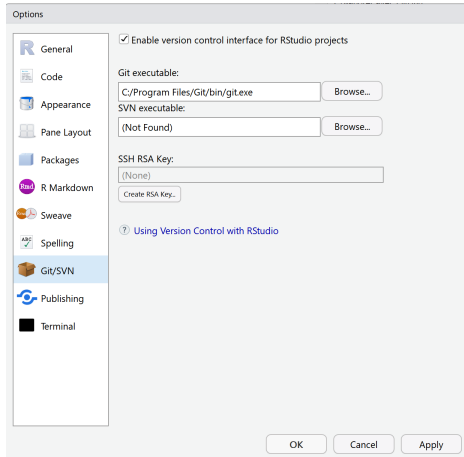


Otherwise, install it.

Source control with git and Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull Request

Write Documents

## Open an account on GitHub

# GitHub security 1

Source control
with git and
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull
Request

Write
Documents

Your computer must be able to write to git.

In RStudio, Tools / Global Options/ Git-SVN, create a new SSH key.

Copy the private key to GitHub, your accounts, Settings, SSH and GPG keys.

Source control
with git and
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull
Request

Write
Documents

Create a GitHub token for CI:

- User account settings;
- Developer Settings > Personal Access Tokens;
- Generate a token, describe it as "git-RStudio" and give it "repo" authorization.

Save the token!

Practical

# Create a project

Source control
with git and
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull
Request

Write
Documents

From scratch:

- In GitHub:
  - *New Repository*
  - Choose name (no special characters)
  - Do not add anything else

- Copy the URL from *Clone or Download*.

- In RStudio: New project / Version Control / Git, paste the URL.

From an existing RStudio project:

- Put the project under version control:

  - *Tools /Version Control /Project Setup…*
  - Select *Git*.

- Create a repository on GitHub and retrieve its URL:
  https://github.com/MyAccount/MyRepo.git

- In RStudio Terminal, run :

```
git remote add origin https://github.com/MyAccount/MyRepo.git
git branch -M master
git push -u origin master
```

Modified files are displayed in the RStudio Git window.

Use .gitignore to hide non-tracked files.

Source control
with git and
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull
Request

Write
Documents

- Create an R file with one line of code and save it.

After each work session, *Commit* the result.

Select files to commit.

Enter a clear message: summary on the first line.

*Push* modifications to GitHub to make them public.

*Pull* to retrieve changes from GitHub.

Source control
with git and
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull
Request

Write
Documents

Declare a collaborator.

Work with two or more people on the same file.

Content of a work session:

- Pull,
- Modify,
- Commit,
- Push.

The elementary data is the line.

Conflicting changes imply a conflict.

Minimize conflicts: in a text, one sentence = one line.

In case of conflict, decide which code or text to keep.

Source control
with git and
Github

Eric Marcon

## Clock icon in Git window
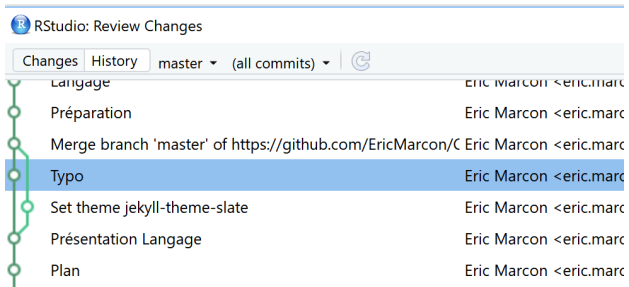
# Practical

Edit the same line of README.md :

- online on GitHub,
- locally.

Commit, Pull, observe the conflict, solve it.

The neighbor on the left invites the neighbor on the right to GitHub.

Both modify the project.

- Remember to pull before modifying.
- Push quickly to limit conflicts.

# Branches

Modify the project without disturbing its stable state.

Application :

- develop new functionality,
- test it, correct bugs,
- make it visible when finished.

Click on New Branch

Work in the new branch (pull, commit, push).

Move to master branch.

Run:

```
git merge branch_to_merge
```

Source control
with git and
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull
Request

Write
Documents

# Fork and Pull Request

Goal: modify someone else's repository

Fork: create a copy of the repository

Start a new branch, modify it.

Branch integration request: *Pull Request*.

On GitHub.

Dialogue possible.

If accepted, merge branch and delete.

# Write Documents

# Use memoiR

Install the **memoiR** package.

Create a new document with memoiR: Stylish Document.

Knit it to check everything is OK.

Use git with your document.

Create an appropriate `gitignore`:

```
memoiR::build_gitignore()
```

Store the knitted files into /docs

```
memoiR::build_githubpages()
```

Make your first commit.

Create your repo on GitHub.

Copy the commands to upload an existing git repo:

```
git remote add origin https://github.com/MyAccount/MyRepo.git
git branch -M master
git push -u origin master
```

In the header of your Rmd file, update the repository name and the URL.

Commit.

Add a README:

```
memoiR::build_readme()
```

Commit.

On GitHub, your repo, Settings / Secrets and variables /
Action :

- Add two secrets:
    - EMAIL: your email address
    - GH_PAT: your personal access token

In RStudio, run

```
memoiR::build_ghworkflow()
```

Commit, Push and look at Actions.