



Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

# Source control with git and Github

Eric Marcon

13 October 2024



Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

# Principles



# git

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

git is a source manager:

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



# GitHub

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

GitHub is a collaboration platform based on Git.

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



Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

# Installation



# git: install

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

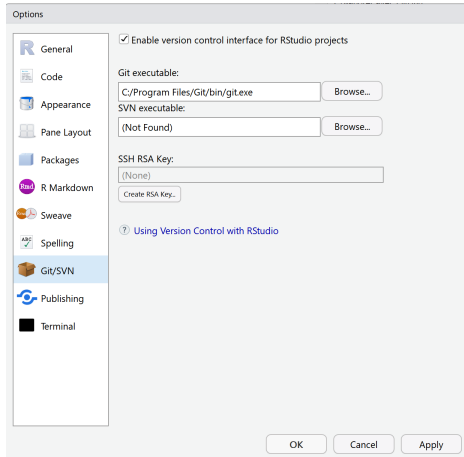
Practical

Branches

Fork and Pull  
Request

Write  
Documents

## RStudio must detect Git



Otherwise, install it.



# GitHub

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

## Open an account on GitHub

The screenshot shows the GitHub profile of Eric Marcon. The browser address bar displays 'https://github.com/EricMarcon'. The profile header includes a profile picture of a man with grey hair and a beard, the name 'Eric Marcon', and the username 'EricMarcon'. Below the name is a link to 'Add a bio'. The 'About' section lists 'AgroParisTech' as the employer, 'Kourou, French Guiana' as the location, and an email address 'eric.marcon@ecofog.gf' with a link to a website 'http://www.ecofog.gf/isip.php?...'.

The 'Popular repositories' section displays four repositories:

- endropart**: Entropy Partitioning to Measure Diversity (1 star, 1 fork)
- dismas**: Distance Based Measures of Spatial Structures (1 star)
- SpatDiv**: Spatially Explicit Measures of Diversity (1 star)
- ericmarcon.github.io**: Home page

The 'Synchro' repository is also listed: 'Synchronize folders for Windows' (1 star, 1 fork).

The 'WMI' repository is listed: 'WMI wrapper for Visual Studio apps' (1 star, 1 fork).

The 'Contributions' section shows '429 contributions in the last year' with a heatmap visualization of contributions from March to February. The heatmap shows a high density of contributions in the latter half of the year, particularly in November and December. A link 'Learn how we count contributions.' is provided at the bottom of the heatmap.



# GitHub security: SSH

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. We choose **SSH authentication** for simplicity.

- In RStudio, Tools / Global Options/ Git-SVN, create a new SSH key (no passphrase).
- Click on *View public key*. Copy the key.
- On GitHub, your accounts, Settings, SSH and GPG keys, Add New SSH Key, paste the key. Name it after your computer.





# GitHub security: PAT

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

Create a GitHub token for Continuous Integration:

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

Save the token!



Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

# 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.



# Create a project

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

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:  
[git@github.com](https://github.com):MyAccount/MyRepo.git
- In RStudio Terminal, run :

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



# Basic operations: filter

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

Modified files are displayed in the RStudio *git* window.

Use `.gitignore` to hide non-tracked files.



# Basic operations: commit

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.



# Basic operations: synchronize

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

*Push* modifications to GitHub to make them public.

*Pull* to retrieve changes from GitHub.



# Basic operations: collaborate

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.





# Conflicts

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

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.



# Project tracking

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

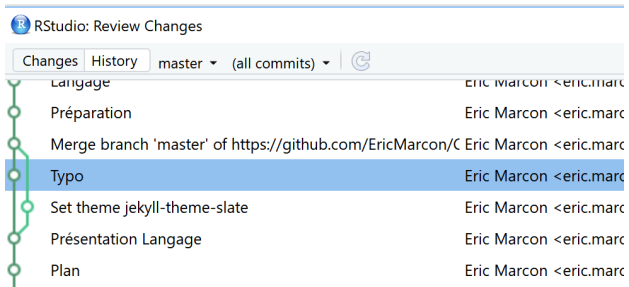
Practical

Branches

Fork and Pull  
Request

Write  
Documents

## Clock icon in Git window





Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

**Practical**

Branches

Fork and Pull  
Request

Write  
Documents

# Practical



# Generate a conflict

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

**Practical**

Branches

Fork and Pull  
Request

Write  
Documents

Edit the same line of README.md :

- online on GitHub,
- locally.

Commit, Pull, observe the conflict, solve it.



# Collaborate with your neighbor

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

**Practical**

Branches

Fork and Pull  
Request

Write  
Documents

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.



Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

**Branches**

Fork and Pull  
Request

Write  
Documents

# Branches



# Purpose

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

Modify the project without disturbing its stable state.

Application :

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



# Create a branch

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

Click on the New Branch button.

Type its name.

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





# Merge

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

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



# Fork

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

Goal: modify someone else's repository

**Fork**: create a copy of the repository

Start a new branch, modify it.



# Pull Request

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

Branch integration request: *Pull Request*.

On GitHub.

Dialogue possible.

If accepted, merge branch and delete.



Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

## Write Documents



# Use memoirR

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

Install the **memoiR** package.

Create a new document with memoirR: Stylish Document.

Knit it to check everything is OK.



# Source Control

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

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.



# Upload to GitHub

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

Create your repo on GitHub.

Copy the commands to upload an existing git repo:

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





# Declare repo in header

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

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

Commit.

Add a README:

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

Commit.



# Continuous integration

Source control  
with git and  
Github

Eric Marcon

Principles

Installation

Practical

Practical

Branches

Fork and Pull  
Request

Write  
Documents

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.