GitHub Introduction Workshop

Luis J. Chueca
RL3. Terrestrial Ecosystems

March 2024

BC3 BASQUE CENTRE FOR CLIMATE CHANGE
Klima Aldaketa Ikergai

luisjavier.chueca@bc3research.org

https://github.com/ljchueca

@LuisjaChueca

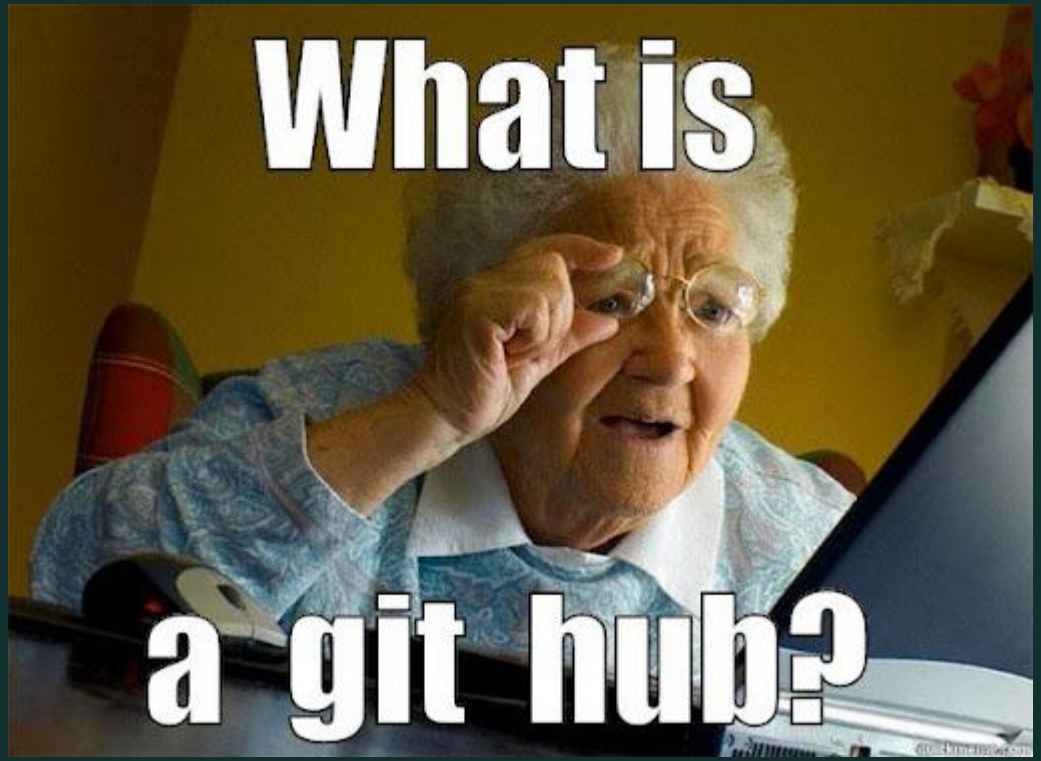
Index

1.

What is GitHub?

1.

What is GitHub?



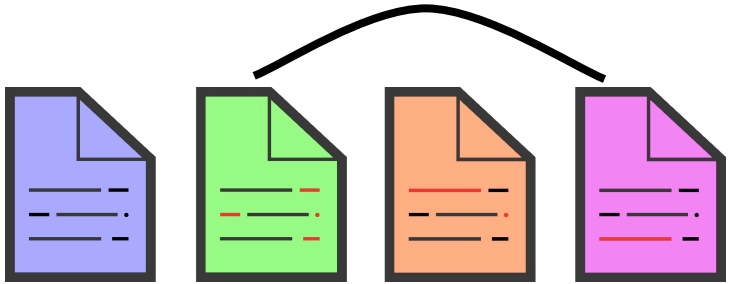
Let’s start from the beginning

***Git** is a **version control system** that facilitates project management by acting as a repository for code modifications made.*

*A **collaboration hub** is a type of workspace designed to promote interaction, community, and teamwork*

It allows:

to check changes made
to consult and restore versions



GitHub

GitHub is a cloud-based application development platform based on the Git code repository, which facilitates collaborative work.

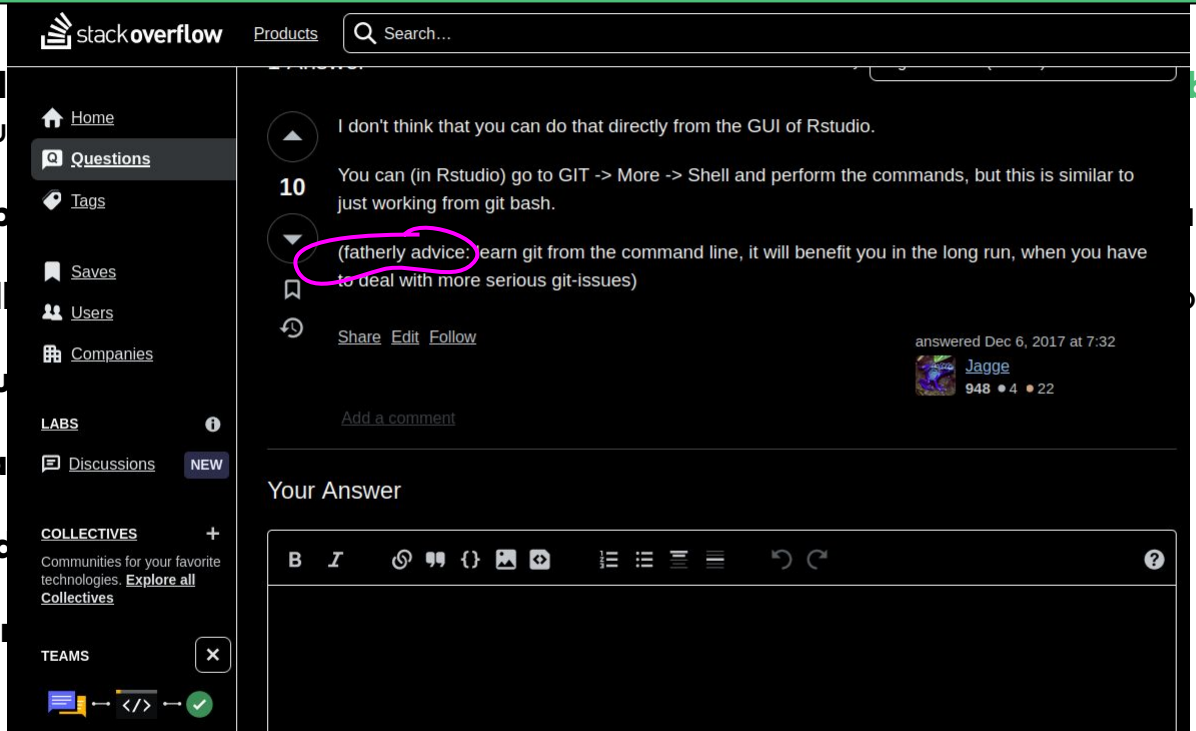
- **Online platform:** It's an online version of Git. To correctly understand **GitHub**, you need first to understand **Git**.
- **Repository hosting:** Allows to store and manage files and its versions from any device
- **Collaborative:** Teammates (or external) can suggest modifications of you code
- **Issues tracking**
- **Branching and merging between collaborators**
- **Code review**
- **more**

BC3

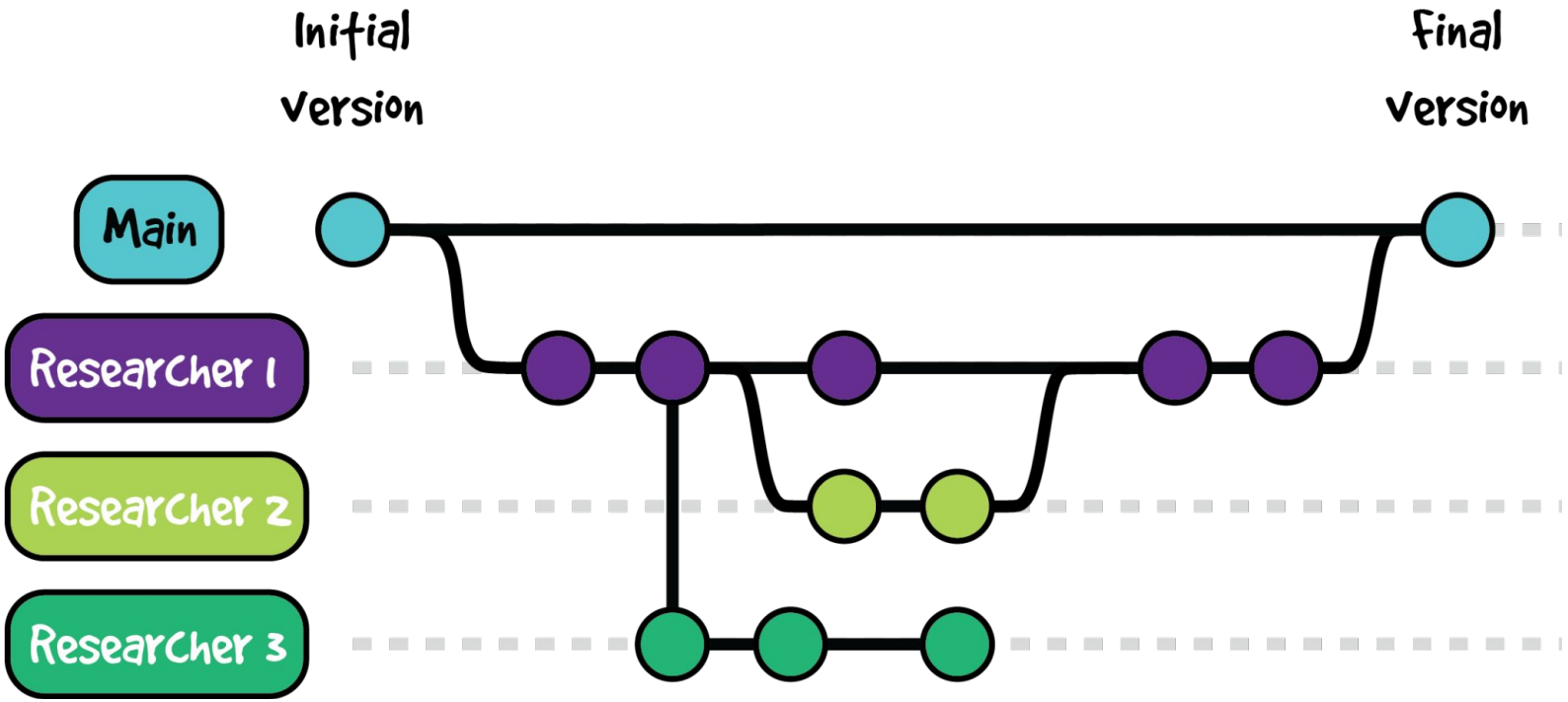
GitHub

GitHub is a cloud-based application development platform based on the Git code repository, which facilitates collaborative work.

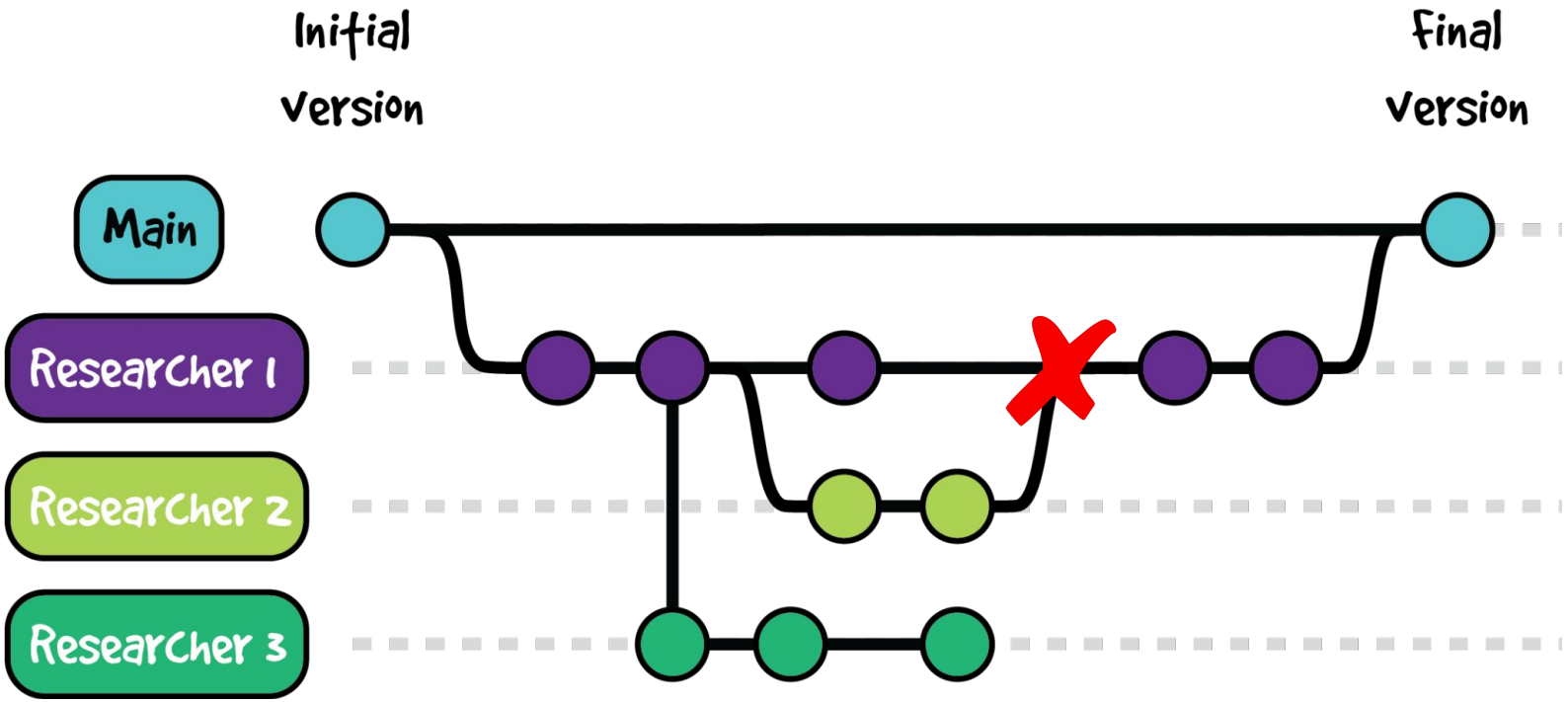
- **Onl** … **b**, you need first to u…
- **Rep** … …ny device
- **Col** … …ode
- **Issu**…
- **Bra**…
- **Coo**…
- **mor**…



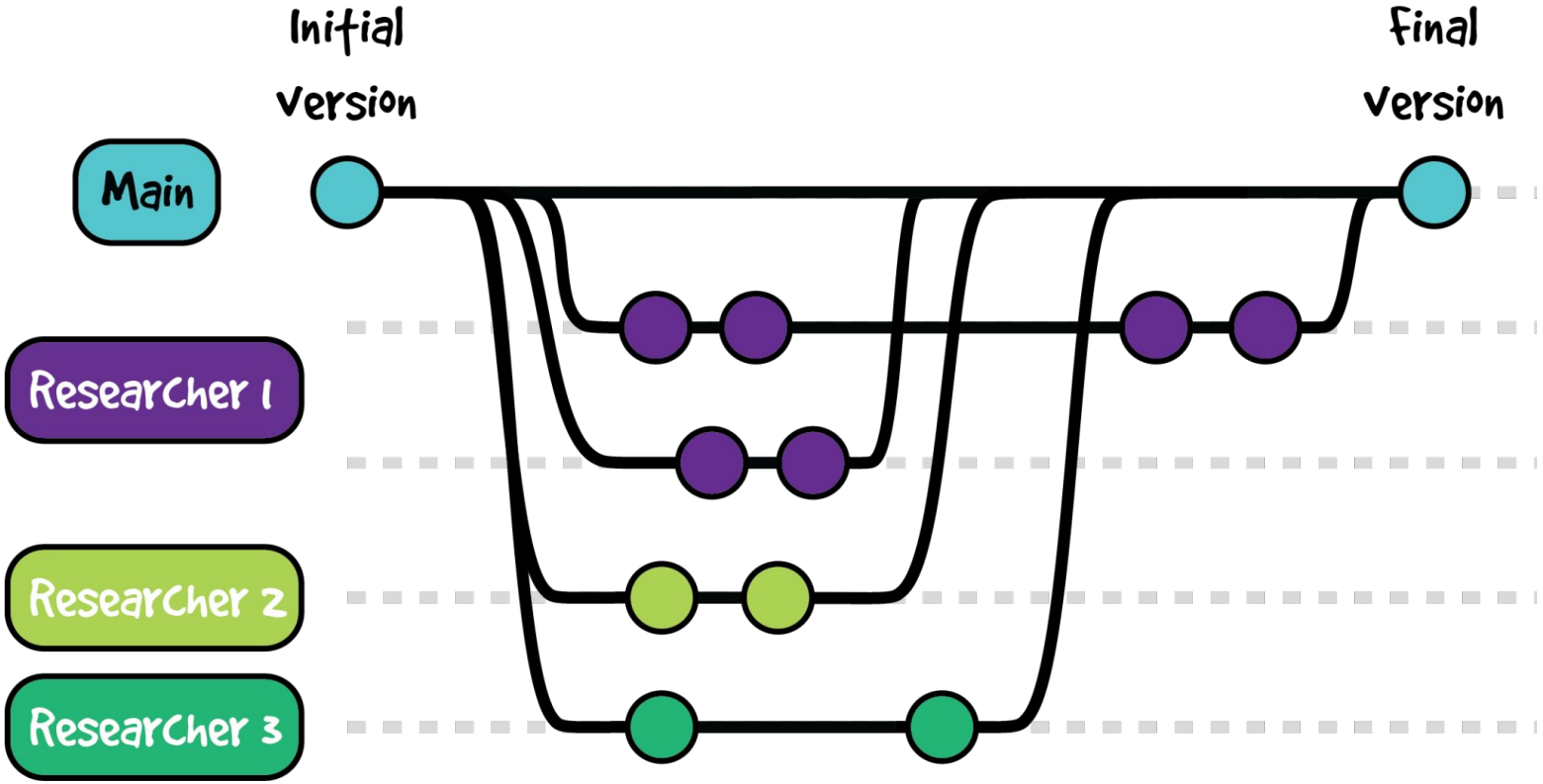
Git flow


Initial
Version
Final
Version
Main
Researcher 1
Researcher 2
Researcher 3

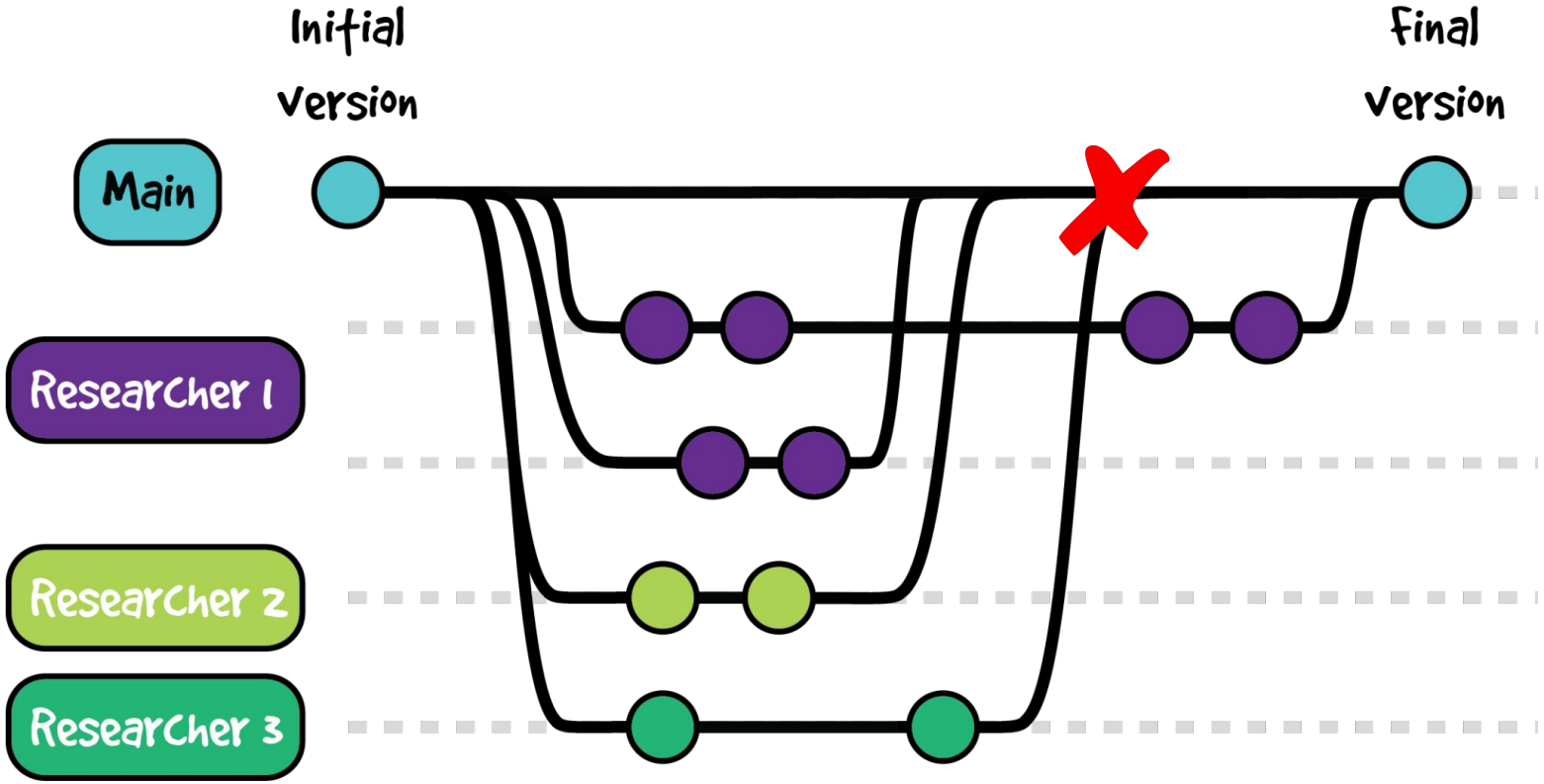
Git flow


Initial
Version
Final
Version
Main
Researcher 1
Researcher 2
Researcher 3

Git flow

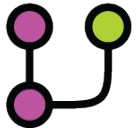

Initial
Version
Final
Version
Main
Researcher 1
Researcher 2
Researcher 3

Git flow

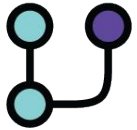

Initial
Version
Final
Version
Main
Researcher 1
Researcher 2
Researcher 3

Organizations

Data Repository 

Repository A  

Workflow 1
Publication 1

Repository B  

Workflows 2,3
Publication 2

Launched in 2008
Purchased/Bought by Microsoft in 2018

Alternatives?

GitHub vs. GitLab: What's the difference?

https://www.getclockwise.com/blog/github-vs-gitlab

Disclaimer!

I have basic user GitHub skills

I work exclusively on GNU/Linux and Mac OS. I'm not Windows user

I don't code in RStudio very often

but I'll try to do my best!

Go to:
https://github.com/ljchueca/BC3-git-workshop

What we need?

1. **GitHub** personal account

1. **Git** installed in our computer

1. **R** and **RStudio** updated

2.

Creating a repository

Creating a repository

https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-new-repository

1) Go to **GitHub** webpage
2) Click on a new repository button

    a) Short and concise name
    b) Avoid special characters and white spaces
    c) Select between public or private
    d) Add **.gitignore** and select **R**
    e) Edit README file

3.

Synchronize GitHub with our Code Editor - RStudio

Connect GitHub with RStudio

Personal Access Token (PAT)

1) Sign in for **GitHub**
2) Create a **PAT** for GitHub. In RStudio

```
library(usethis)
create_github_token()
```

1) In the opened browser:
   a) New personal access token -> create
   b) Copy and save it

Connect GitHub with RStudio

Personal Access Token (PAT)

4) Store **PAT** to connect RStudio and GitHub. In RStudio

```
library(gitcreds)
gitcreds::gitcreds_set()
```

Clone a repository locally

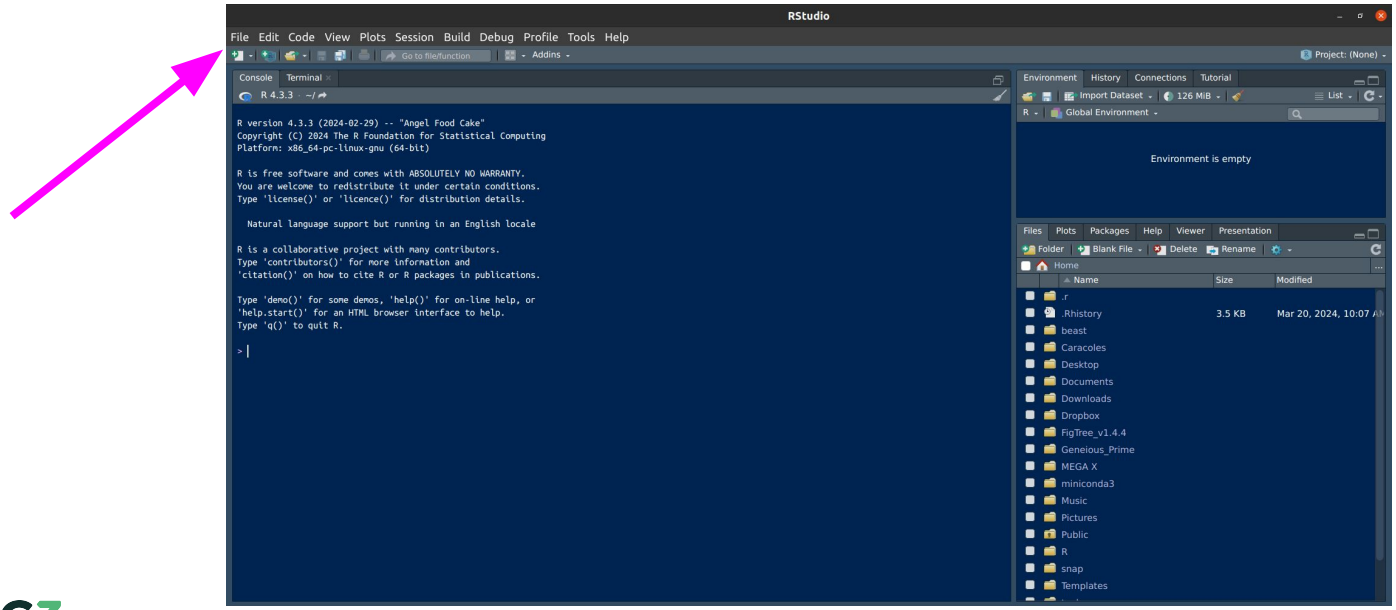
Go to the repository url
Code -> Copy url to clipboard

a) In RStudio: File > New Project > Version Control > Git

Clone a repository locally

Go to the repository url
Code -> Copy url to clipboard

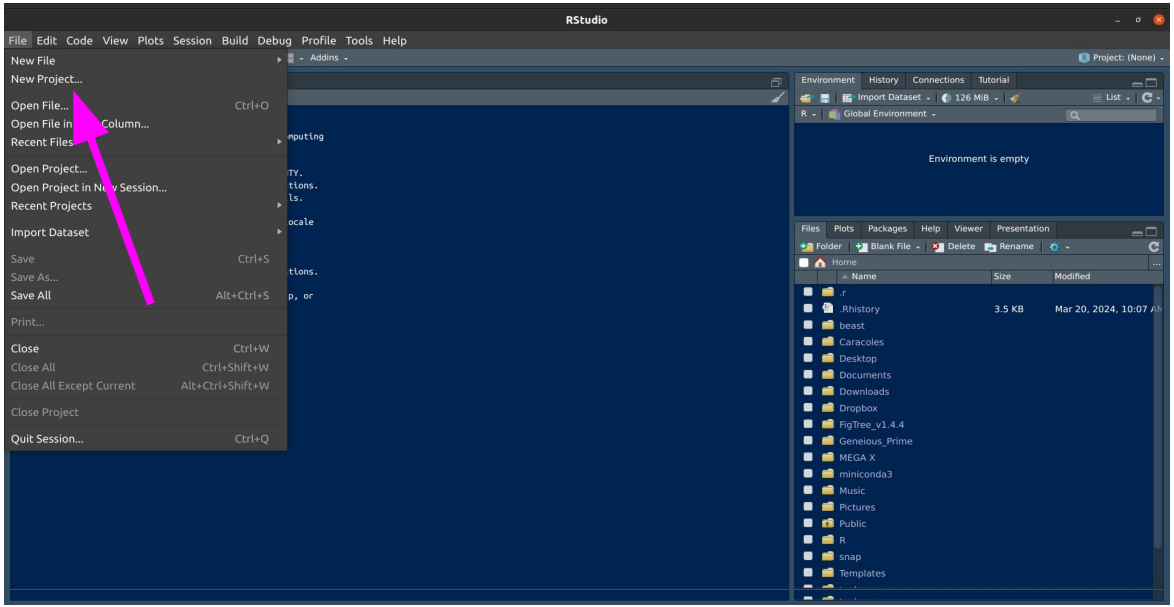
a) In RStudio: File > New Project > Version Control > Git



Clone a repository locally

Go to the repository url
Code -> Copy url to clipboard

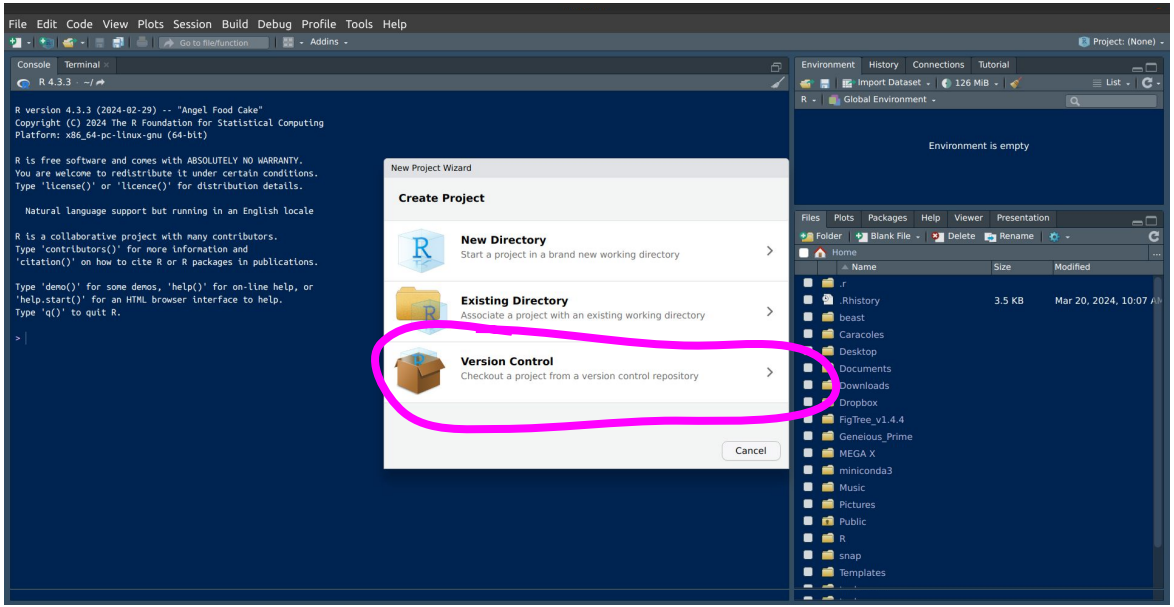
a) In RStudio: File > New Project > Version Control > Git



Clone a repository locally

Go to the repository url
Code -> Copy url to clipboard

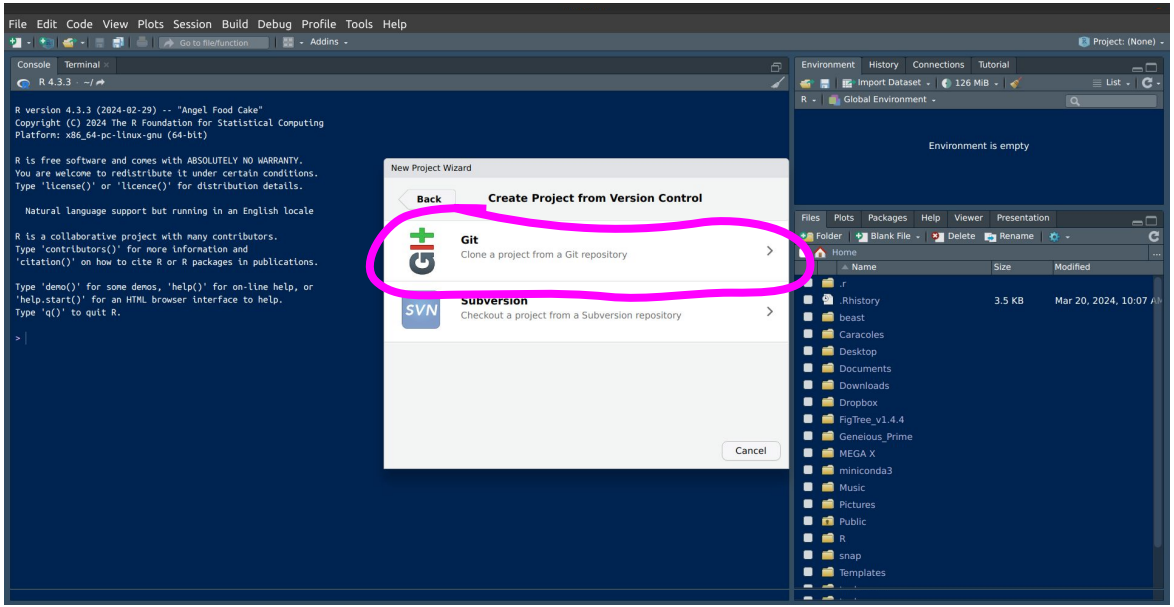
a) In RStudio: File > New Project > Version Control > Git



Clone a repository locally

Go to the repository url
Code -> Copy url to clipboard

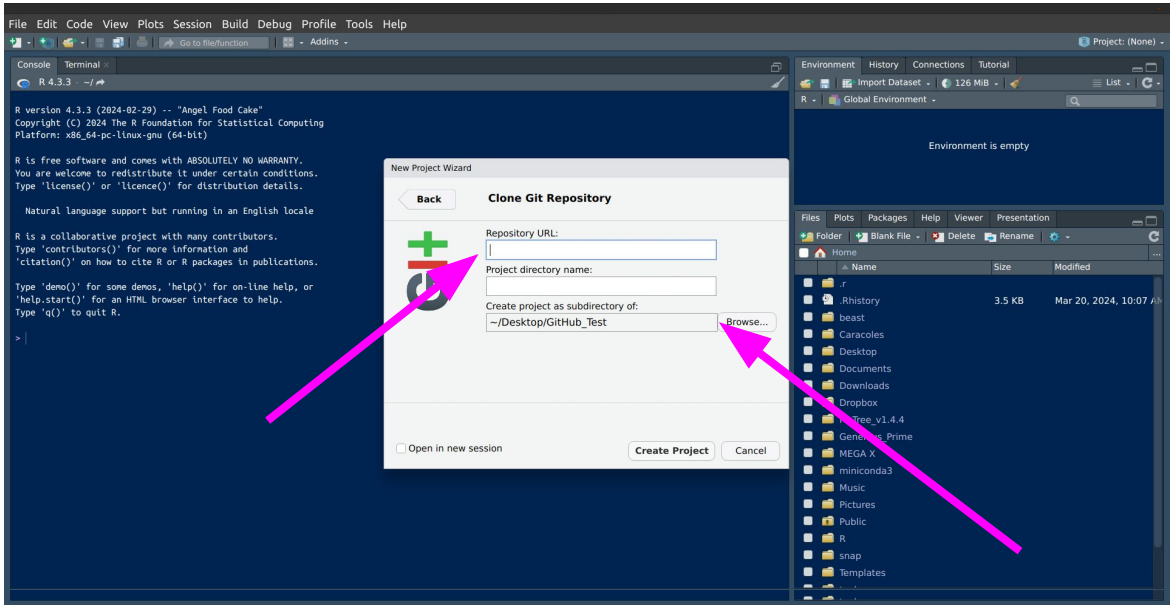
a) In RStudio: File > New Project > Version Control > Git



Clone a repository locally

Go to the repository url
Code -> Copy url to clipboard

a) In RStudio: File > New Project > Version Control > Git



Clone a repository locally

Go to the repository url
Code -> Copy url to clipboard

b) In **Terminal**

```
# Clone a GitHub repository

git clone "url"
```

Create folders/directories

In **GitHub** webpage:

- Add file
  - + Create new file
    - write the folder/directory name and file name
    - e.g. ‘subfolder_a/.keep’

In the **Terminal**:

- Create a directory (subfolder_a) and empty file (.keep)
- Commit and push

```
# Create a directory
mkdir 'NEW_DIRECTORY'
cd 'NEW_DIRECTORY'
touch .keep
```

4.

Commit changes

```
# Check branches available
git branch -a
# Switch to any branch
git checkout 'BRANCH NAME'
```

```
git status
```

Will tell you the *branch* you are on, the *deleted*, *modified*, *created* files, if the local commits are pushed, which changes are staged...

Stage the changes

```
git add 'FILE_TO_BE_STAGED'
git add . # Everything is added to be committed
```

Commit

```
# Commit Changes

git commit -m "WRITE YOUR MESSAGE HERE"
git pull
git push
```

```
# Check branches available
git branch -a
# Switch to any branch
git checkout 'BRANCH NAME'

# Commit Changes
git status
git pull
git add 'FILE_TO_BE_STAGED'
git add . # Everything is added to be committed

git commit -m "WRITE YOUR MESSAGE HERE"
git pull
git push
```

Check your commits and changes

```
# List Commits
git log             # or
git log --oneline   # or
git reflog
```

You will see the **SHA** (Simple Hashing Algorithm = unique id of your commit), the author, date/time, commit message

Check your commits and changes

```
# List Commits
git log             # or
git log --oneline   # or
git reflog

# See the changes of a commit
git show <SHA>
```

You will see the **SHA** (Simple Hashing Algorithm = unique id of your commit), the author, date/time, commit message

Undo commits

Different approaches

```
git reset # dangerous: you go back to the desired commit and
"erase" all the commit created since.

git reset - - hard <SHA>
```

You can bypass

```
git reset --hard <previous-SHA>
git reset --soft <last-SHA>
```

5.

Compare and merge branches

```
# Switch to main branch
git checkout main

# Merge branches to main
git merge 'BRANCH NAME'
git status

# Merge branches to main. If they are very different
git merge --allow-unrelated-histories 'BRANCH NAME'

git commit -m "WRITE YOUR MESSAGE HERE"
git pull
git push
```

6.

Issues

Issues

New issue

- Add a title
- Add a description
  - Many different options
  - Preview

Submit new issue

7.

Discussion

a) Questions?
b) What did you expect?
c) Suggestions?

Thank you!

Connect Git with RStudio

Create a SSH password in our computers: Mac, Linux or Windows

Add the SSH password to our GitHub account

https://anderfernandez.com/blog/como-vincular-y-usar-git-con-rstudio/