



Intro to GitHub

Club stats iEES-Paris, 27/10/2021

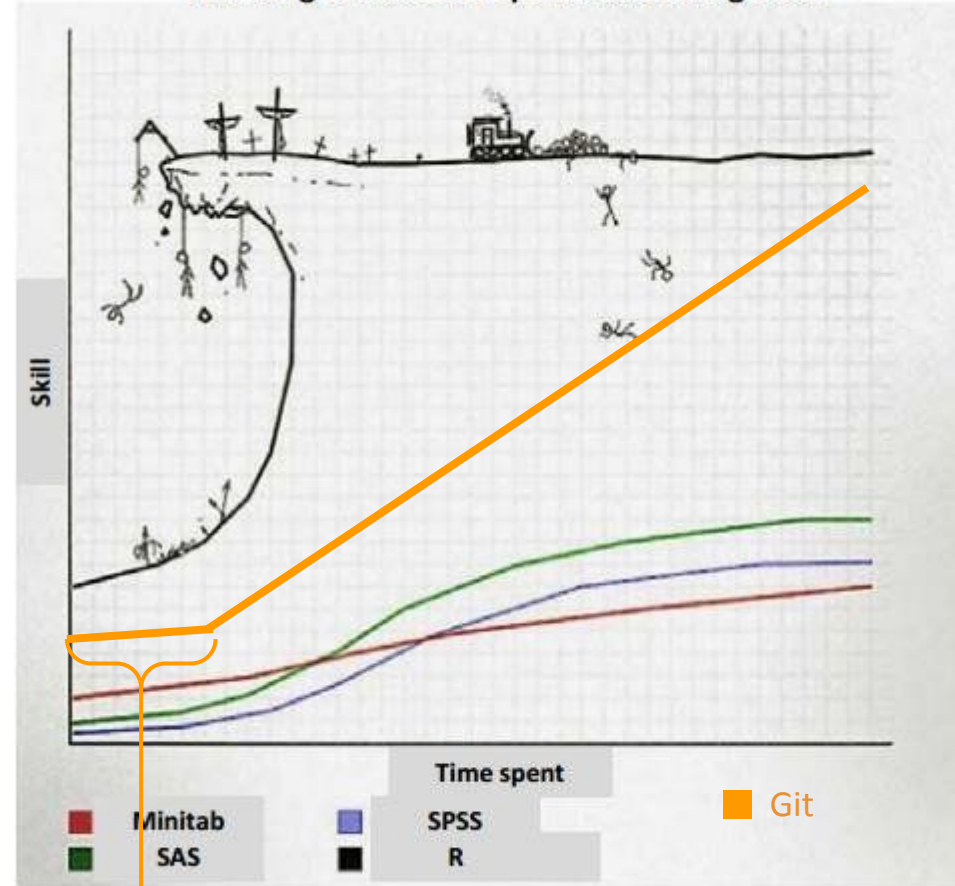
Rosalie Bruel

✉ rosaliebruel@gmail.com

🐦 @RosalieBruel

🐙 rosalieb

Learning Curves of Popular Stats Programs

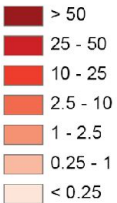


mental barrier over learning a new tool

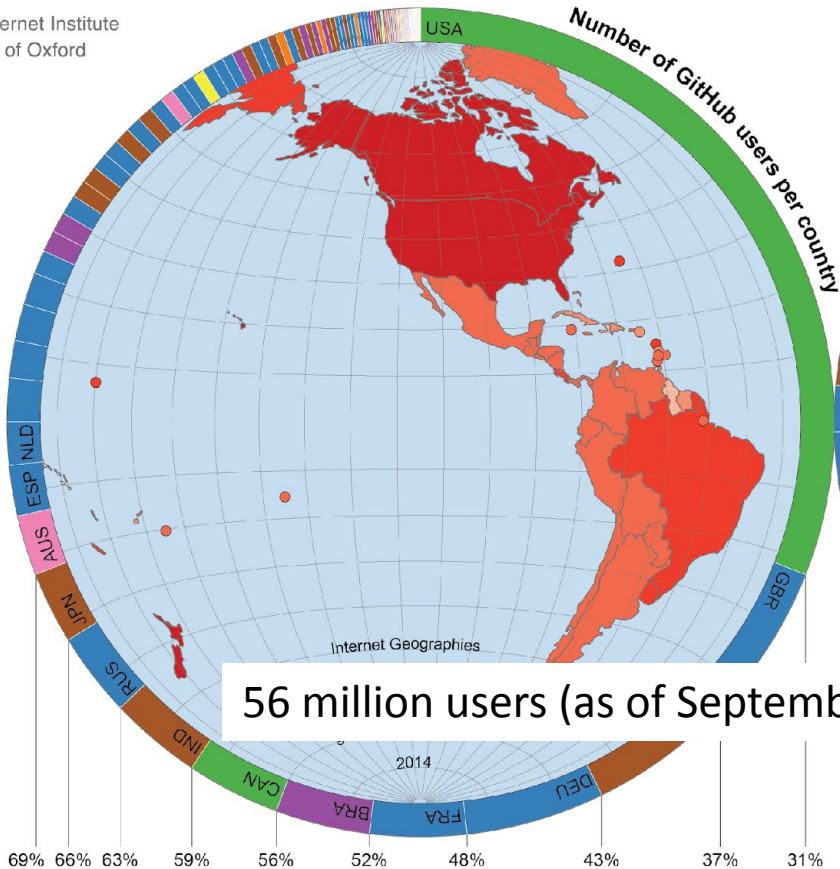


Oxford Internet Institute
University of Oxford

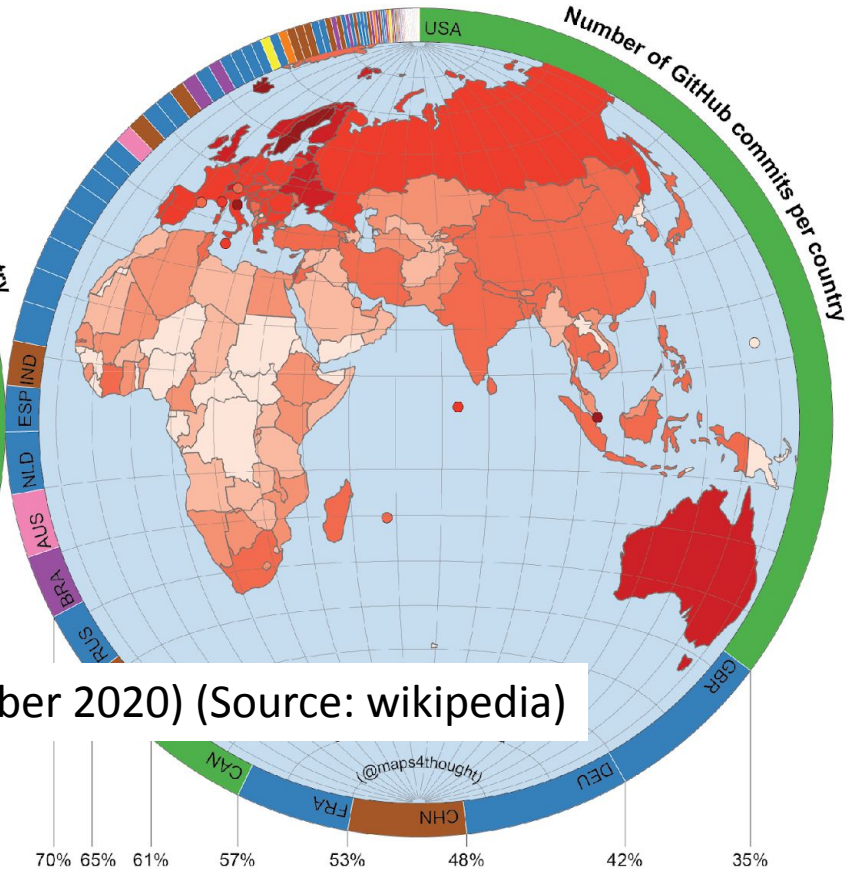
GitHub users
per 100,000
Internet users



data sources:
GitHub.com
World Bank
data.worldbank.org



56 million users (as of September 2020) (Source: wikipedia)



GitHub | Mapping collaborative software

What are Git and GitHub?

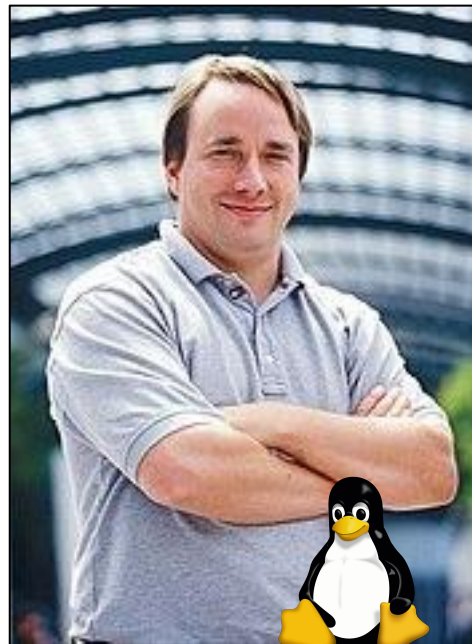
Git

Version control software created by Linus Torvalds, father of Linux (2005)

GitHub

A website service that uses Git (so basically a website where you can do version control) (2008) - popular alternative: **GitLab** (2014)

[GitHub/GitLab are to Git
what Rstudio is to R]



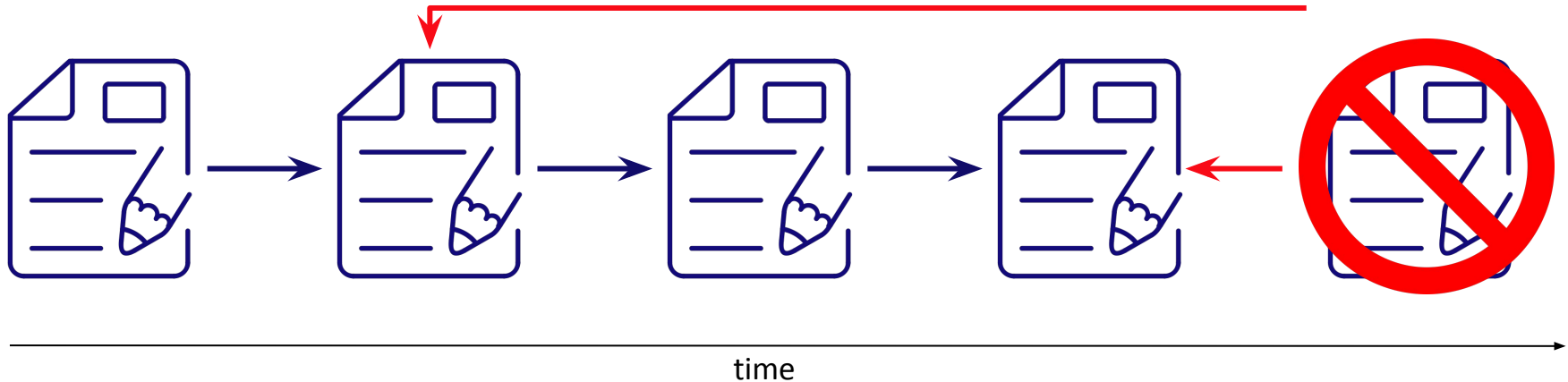
Linus
Torvalds

What is version control?

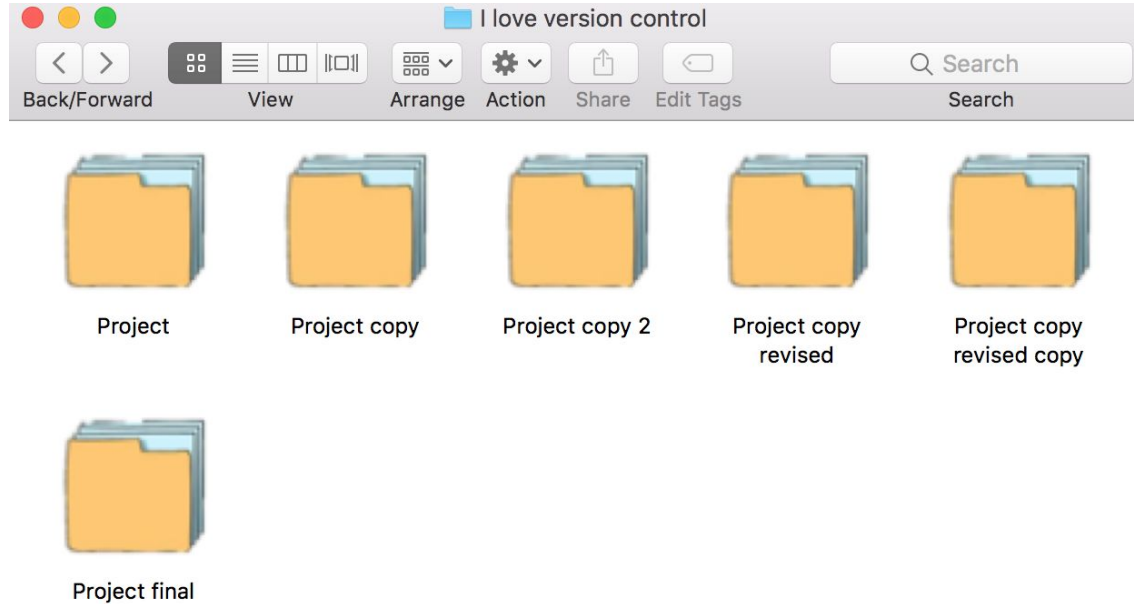
[List of content](#)

Version control systems keep track of project updates, providing a detailed history.

You can easily review or revert to any previous version if something breaks.



Why would you even need a version control system?



FYI!

Just so you know... There are many version control systems, each with their strengths and weaknesses. Choose the one that works best for you!
Git is good for codes.

SUBVERSION®



(Better?) alternatives to GitHub exist too.

[Read comparative reviews](#)



GitLab

- Does not share stats on number of users, but growing user base
- Open source! It may help overcome the current interface issues in the upcoming years.
- Best storage offer in a free version



GitHub

- 40+ millions users: advantages of a widely used platform when community matters (true in research)
- used to limit the number of collaborators in the free version: no longer true (?)
- Easy interface



Bitbucket

- 10 millions users in 2019 (no recent stats, but probably the least popular right now)
- Better for developers using Jira

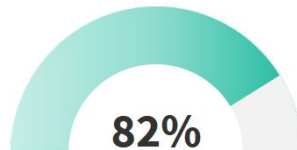
GitLab vs GitHub

Feature maturity comparison against set of industry standards

[Learn more about our scoring methodology →](#)



GitLab

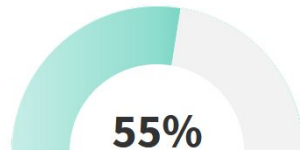


105 / 128

Features & Capabilities



GitHub



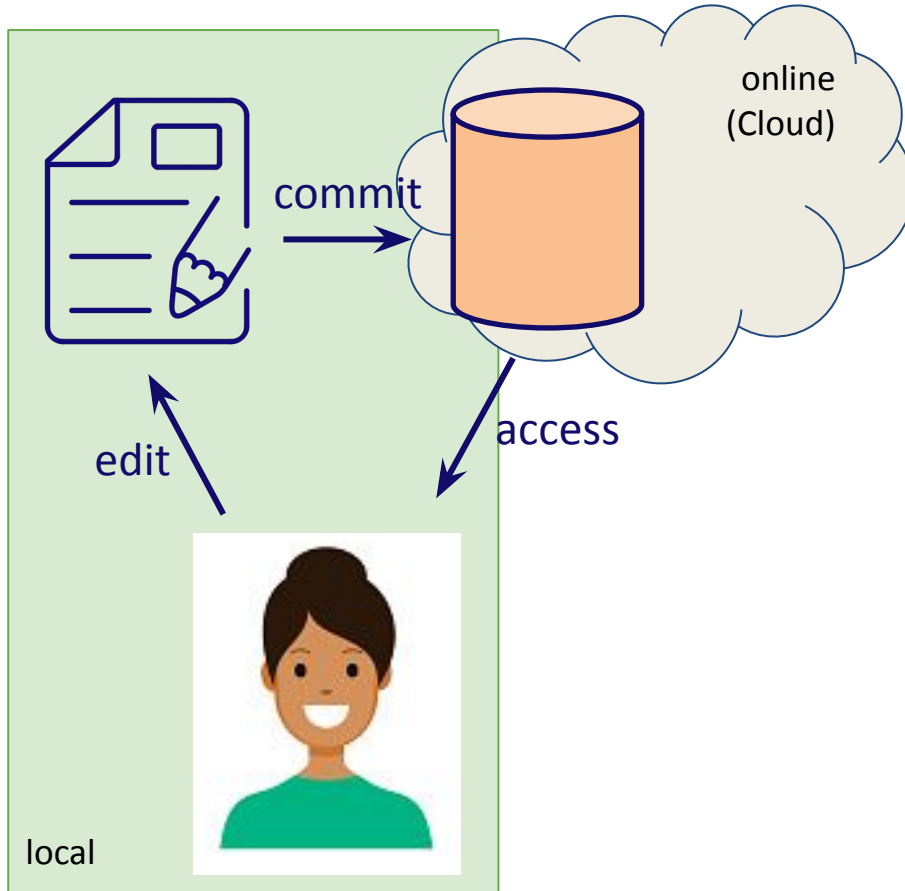
70.5 / 128

Features & Capabilities




(source: [GitLab](#))

[List of content](#)

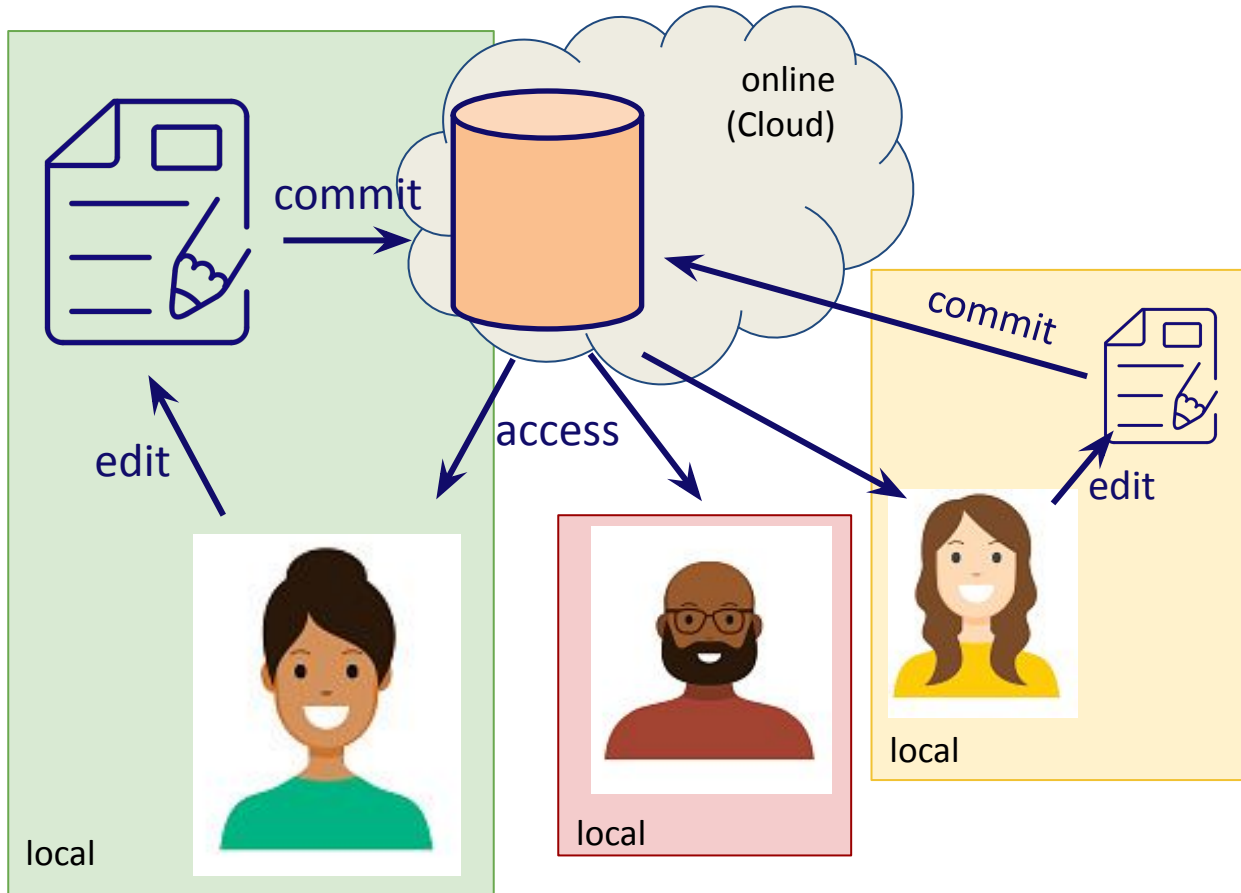
How does Git work?



Terminology:

- Repository 
- Files 
- User 
- Commit

How does Git work?



Terminology:

- Repository 
- Files 
- Users 
- Commit



I will now replace Git by GitHub on the slides, assuming you chose to use a **service** (e.g., GitHub, GitLab) to give to your repositories an **address on the Cloud** (web address).

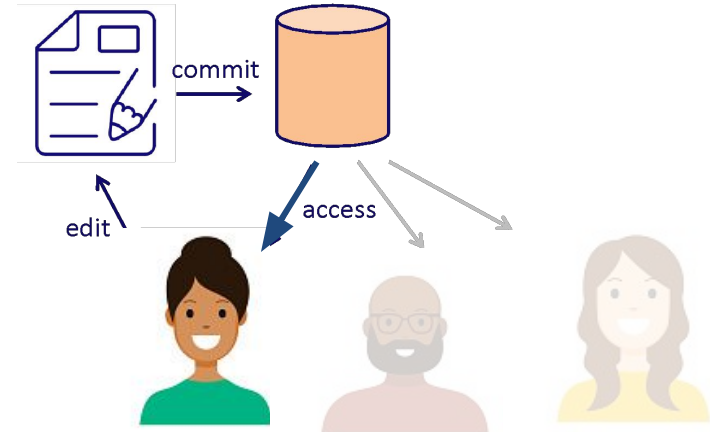
Option 1 – stick to the basics

It's possible to use GitHub solely for:

- Backups
- Make your codes available after paper publication
- Access codes from publications
- Access codes not implemented in packages

Terminology:


- Repository 
- Files 
- Users 
- Commit

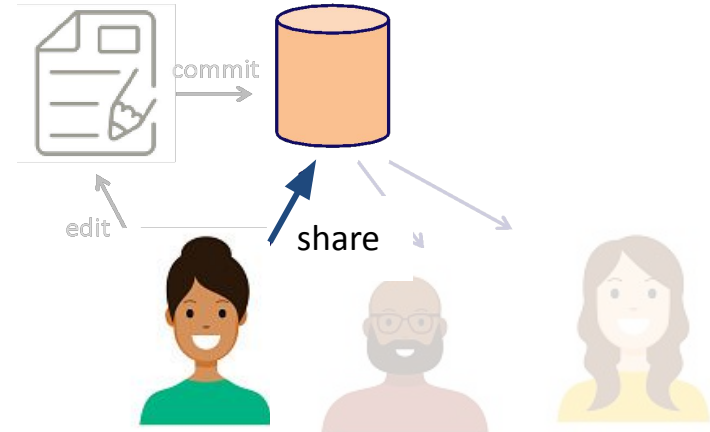


It's possible to use GitHub solely for:

- Backups
- Make your codes available after paper publication
- Access codes from publications
- Access codes not implemented in packages

Terminology:




- Repository 
- Files 
- Users 
- Commit

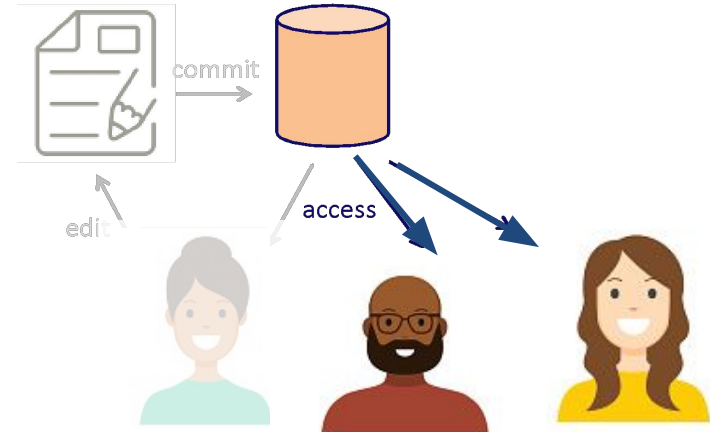


It's possible to use GitHub solely for:

- Backups
- Make your codes available after paper publication
- Access codes from publications
- Access codes not implemented in packages

Terminology:

- Repository 
- Files 
- Users 
- Commit



Backups/make your codes available: create a repository


If you use the minimum features, such repository would look exactly like if you had a Google Drive folder.

See [what makes a “good” repository](#)

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 rosalieb ▾

Repository name

myrepository ✓

Great repository names are short and memorable. Need inspiration? How about **literate-lamp**.

Description (optional)

Test



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Access codes from publications / blogs / other: visit author's repository

Unless stated otherwise, all analyses were conducted using R (version 3.1.3, R Core Team, 2015) with several additional packages: *vegan* (version 2.2-1, Oksanen et al., 2014) was used to fit the ordinations, *analog* (version 0.16-0, Simpson, 2007; Simpson and Oksanen, 2014) was used to fit the principal curves and time tracks, and additive models were fitted using the *mgcv* package (version 1.8.4, Wood, 2004, 2006, 2011). Additional R functions written by GLS (based on suggestions by Simon Wood, pers. comm.) were used to evaluate the first derivative of the trend splines and form the pointwise confidence interval. R scripts implementing the analyses and reproducing the figures are available online from <https://github.com/gavinsimpson/bennion-frontiers-2015>.

Bennion *et al.*, 2015

We solved the boundary-value problem (5) with the `bvpSolve()` package in R using function `bvptwp()` (Mazzia et al. 2014). An R script to illustrate the method using a simple ecological model is found at https://github.com/SRCarpen/Exit_Time_R

Carpenter *et al.* (2021)

fitting these models, and demonstrate how to fit HGAMs on example data. All code and data used to generate this paper are available at: github.com/eric-pedersen/mixed-effect-gams.

Pedersen *et al.* (2019)


[List of content](#)

Access codes from publications / blogs / other: visit author's repository

repository address shared in the paper: [github.com/user/repo](https://github.com/eric-pedersen/mixed-effect-gams)

 [eric-pedersen / mixed-effect-gams](https://github.com/eric-pedersen/mixed-effect-gams)

Public

 Notifications

 Star

85

 Fork

18

 Code

 Issues

 Pull requests

 Actions


 Projects

 Wiki

 Security

 Insights

 master ▾

 24 branches

 6 tags

Go to file

Code ▾

must be public for anyone
to access



eric-pedersen updated code to explicitly set stringsAsFactors=TRUE to acc



code

Added fix to deal with change in r



compiled_paper

updated code to explicitly set strin



data

Compiling final_final_last_final.pdf



figures

changed letters on figures to upp



paper_sections

updated code to explicitly set strin



.gitignore

added list of intermediate files to be ignored



.zenodo.json

In-process edits



LICENSE

LICENSE

 Clone

HTTPS GitHub CLI

<https://github.com/eric-pedersen/mixed-efl>

Use Git or checkout with SVN using the web URL.

 Open with GitHub Desktop

 Download ZIP

Two ways to clone
(create a local copy of
the repository)

About

Repo for tutorial/paper on mixed-effect GAMs

 Readme

 MIT License

Releases 4

 Final version submitted to P...
on 24 Mar 2019

Latest

+ 3 releases

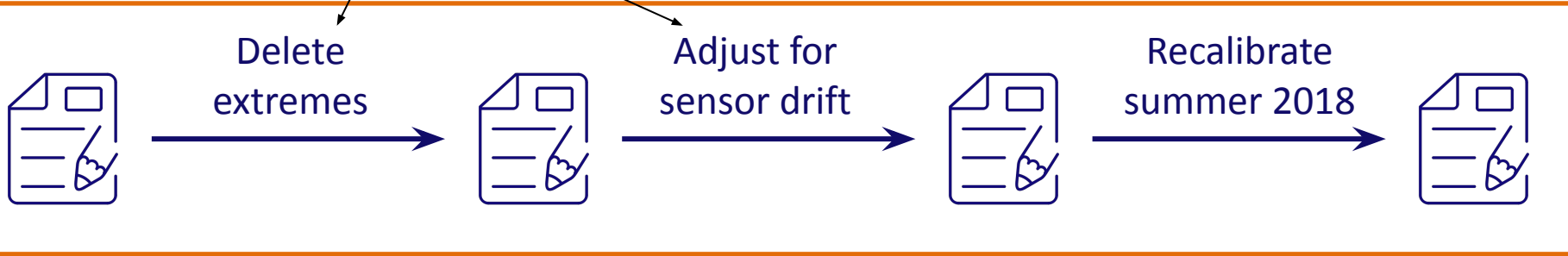
[List of content](#)

Option 2 – use the version control
capacities




Git for version control

The possibility to add a comment detailing what you worked on helps you review your past code.
(But this feature is only as helpful as the message you write is clear)

Master branch







Terminology:

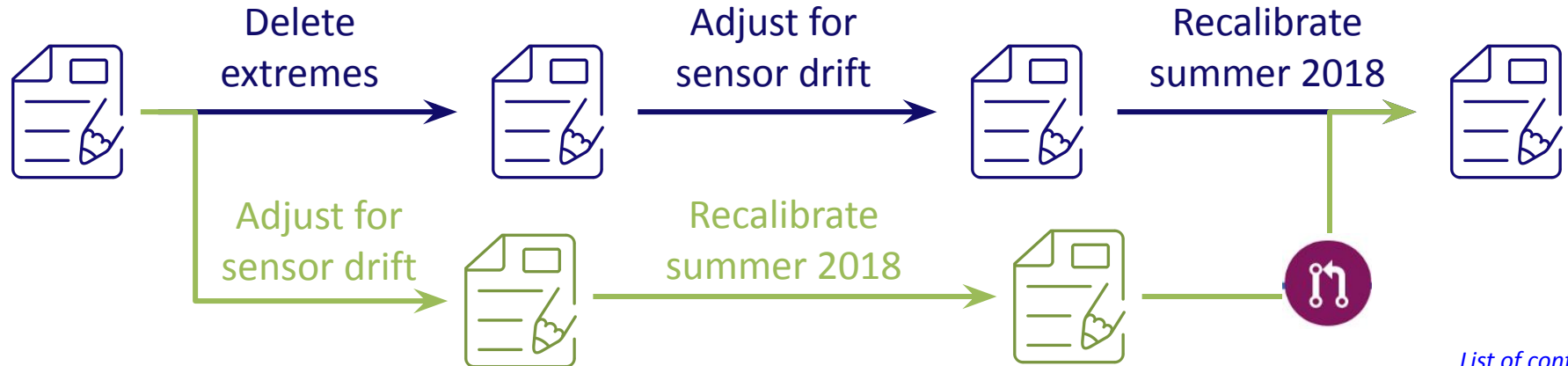
- Repository 
- Files 
- Users 
- Commit
- Master branch

Git for version control

Featured branches are 'alternative realities' that allow you to test scripts




Terminology:

- Repository 
- Files 
- Users 
- Commit
- Master branch
- Feature branch
- Pull request 



Git for version control

Terminology:

- Repository 
- Files 
- Users 

fixed nls() error, added transformations

[Browse files](#)

 master

 **taylorstewart** authored and **taylorstewart** committed on Feb 2, 2015

1 parent [b47351d](#)

commit [a826adaf3f4d0ce33ff9d1ab6921b7d0be09cfea](#)

 Showing **3 changed files** with **32 additions** and **28 deletions**.

Unified

Split

41  server.R

View



@@ -196,7 +196,7 @@ shinyServer(function(input, output, session) {

196 196 m <- m[m\$season == input\$season,]

197 197 }

198 198 # Optional: filter by serial

199 - if (!is.null(input\$serial) && input\$serial != "All") {

199 + if (!is.null(input\$serial) && input\$serial != "") {

200 200 m <- m[m\$serial == input\$serial,]

201 201 }

Change at line 199: "All" was deleted

Autumn 2016 Merge #3



taylorstewart merged 3 commits into taylorstewart:master from lake-erie-biological-station:master

Conversation 0

Commits 3

Checks 0

Files changed 38



taylorstewart commented on Dec 4, 2016

Owner



No description provided.



lake-erie-biological-station added some commits on Jul 19, 2016



Merge pull request #4 from taylorstewart/master ...

significant "house cleaning" of repository



data quality review ...

This corrects an issue with the wingspread measurements and derived values per swept area; also two large lake sturgeon weights were added based up on published length weight regressions.



UPDATED with Autumn 2016 data ...

still needs some small revisions to fix MathJax and review the water quality table for functionality

Unique ID per action

69c2649

bd126dd

bbbe03e

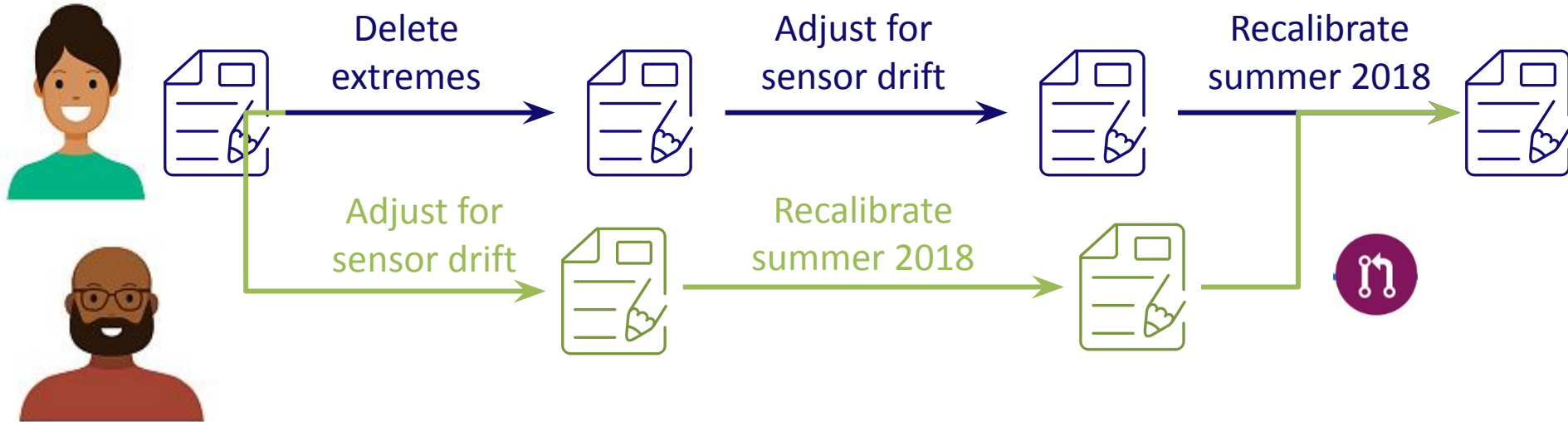


taylorstewart merged commit 8e14865 into taylorstewart:master on Dec 4, 2016

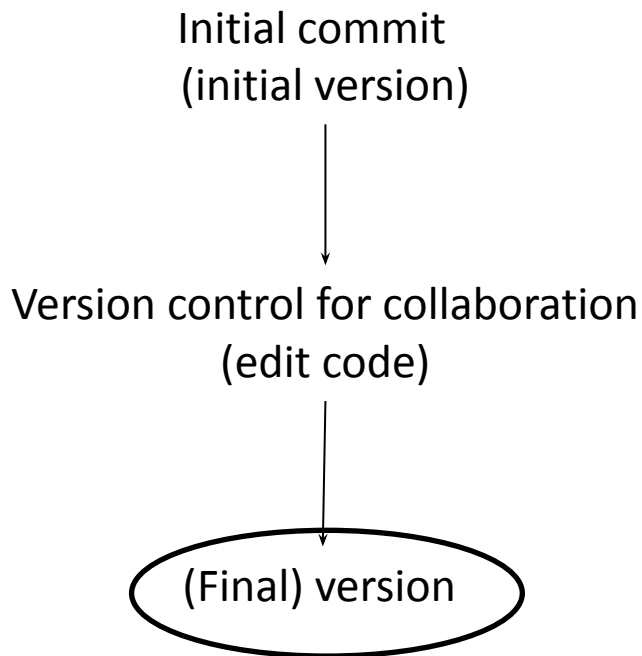
Revert

[List of content](#)

Git track changes between local and remote repositories, and allows clones (if different users are working on the same file)



Version control summary



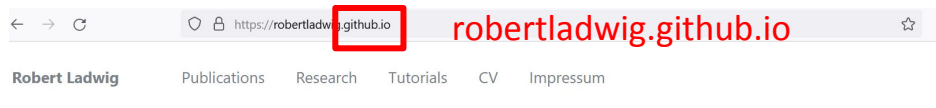
You may want to access only this,
and using Github would still be
relevant

Option 3 – Types of output

Types of output

- Local R folder (back-up, code-sharing, etc.)
- R-package (developing version of a CRAN R package, or R package existing only on GitHub). You will need the library devtools in R to access a repository.

```
install.packages("devtools")
devtools::install_github("rosalieb/serac", build_vignettes = TRUE)
library(serac)
```



- Website



Dr. Robert Ladwig

Aquatic Ecosystem Modeling,
Physical Limnology and
Theoretical Ecology

📍 Location

✉ Email

🐦 Twitter

🌐 Github

🔍 Google Scholar

About me

New paper is out: R. Ladwig, L. Rock, and H. Dugan, Impact of salinization on lake stratification and spring mixing, Limnology & Oceanography Letters (2021). Check it out [here](#).



Welcome, my name is Robert. I am a limnologist with a keen interest in aquatic ecosystem modeling, aquatic ecology and theory-guided data science.

Other resources

Desktop versions

mac.github.com

dekstop.github.com

Linux version still in development?

FYI – Possible to write from Terminal on your dekstop, and then a few command:

cd = change directory

cd .. = up in directory

cd Dekstop/work

ls = tells you which document are in the repository

git clone (and paste the repository from github online).

git status= gives you which file are present on your computer

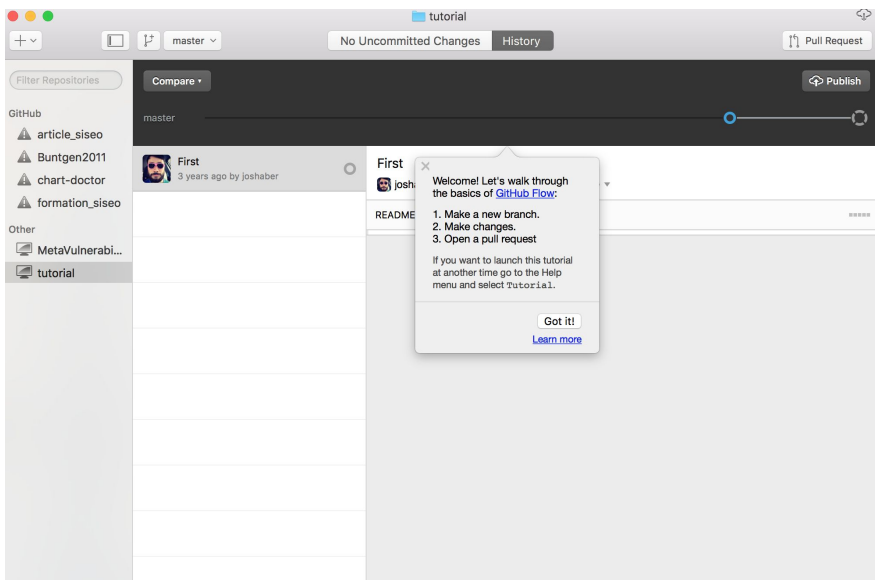
git add= if you want to commit your script.

git commit

git push

git pull

esc:wq



Why ECR may be interested in this tool?

(Why you should encourage your students to use Git.)

- **Alternative to academia: data scientist jobs.** Companies will expect you to know how to use version control.

[NATUREJOBS](#) | [NATUREJOBS BLOG](#)

From Doctorate to Data Science: A very short guide

21 Jul 2017 | 13:00 GMT | Posted by [Jack Leeming](#) | Category: [Academia](#), [Blog](#), [Career paths](#), [Collaboration](#), [Communication](#), [Data](#), [Industry](#), [Mobility](#), [PhD](#), [Postgraduate](#), [Research](#), [Social media](#), [Technology](#)

- **Open science:** sharing codes
- **Ageless reason:** auditability and reproducibility!

Collaborations

Advanced

Version control

Collaborate on the same code

Multiple branches (fork/merge)

“Stay in your lane” approach

Intermediate

Version control of the same script

Beginner

Single user

Share/Access final version of a code

You can use Git for one or several reasons and have various levels of expertise depending on the topic

R project

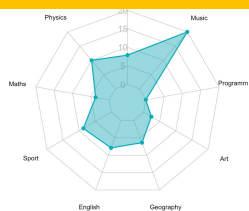
Site web

R package

Apps?

etc.

Output



Summary:

- Version control is your friend.
- Sharing is caring.
- Depending on your use, you may be better off with systems like 'track changes' in Word, or Dropbox...
- ...but for codes, specific systems exist.
- The ball is in your court if you ever need to collaborate on codes!

Questions?

Links to specific sections of the presentation

[Generalities on Git/GitHub and what is version control](#)

[Basics of Git: backup, share/access codes and packages](#)

[Git- intermediate: version control](#)

[Git-intermediate: types of tools](#)

[Summary and concluding remarks](#)

Additional topics:

[What makes a “good” repository](#)

[Why can't anyone write in the GitHub repository of the iEES-club?](#)

[Issues and what they are?](#)

[Links to diverse resources](#)

Sources up until now:

- Benjamin Wilcox
<https://www.slideshare.net/BenjaminWilcox/version-control-is-your-friend>
- YouTube channel GitHub, the coding train
- Check out:
<https://guides.github.com/activities/hello-world/>
- Wilson *et al.* (2017) Good enough practices in scientific computing, *Plos Computational Biology*

Rosalie Bruel

✉ rosaliebruel@gmail.com

🐦 @RosalieBruel

🐙 rosalieb

[List of content](#)

Component of a “good” repository

- readme file (.md) *(see following slide for example)*
 - Documentation: what is inside the repository, how to cite, preferred ways of reporting an issue, etc.
- License
 - If your repository is public, anyone using your code is free to do so. You can choose among licenses templates to limit the terms of use.
- gitignore file (.txt) *(see following slide for example)*
 - By definition, Git will keep in memory ALL of what you uploaded and whatnot. For example, if every day you upload your progress and your current .Rdata file, it will use a big amount of storage that you don't really need to be using (if you save the code, you can recreate the objects later).
 - The gitignore list extensions or folders to not commit (e.g., png, jpeg, pdf, csv, xlsx, Rdata, etc.)

README.md

.md language

what we see

<> Edit file

👁 Preview

<> Edit file

👁 Preview

```
1
2 <!-- README.md is generated from README.Rmd. Please edit that file -->
3 <!-- badges: start -->
4 <!-- badges: end -->
5
6 # rhobo <a href="https://github.com/rosalieb/rhobo" target="_blank"></a>
7
8 The goal of rhobo is to read in the raw HOBO files downloaded through
9 the HOBO software and compute the dissolved oxygen correction factors.
10 Functions are also included to help append new data to previous data
11 files.
12
13 The steps match the routine for data from the artificial lakes at the
14 [CEREEP-Ecotron](https://www.cereep.bio.ens.psl.eu/spip.php?article45),
15 and may need to be adapted to other datasets format to be widely usable.
16
17 **Authors**: Rosalie Bruel (iEES-Paris, CNRS, France) and Sophie Guillon
18 (Mines ParisTech, France)
19
20 ## Installation
21
22 You can install the development version from
23 [GitHub](https://github.com/) with:
24
25 # install.packages("devtools")
26 devtools::install_github("rosalieb/rhobo")
27 library(rhobo)
```

rhobo

The goal of rhobo is to read in the raw HOBO files downloaded through the HOBO software and compute the dissolved oxygen correction factors. Functions are also included to help append new data to previous data files.

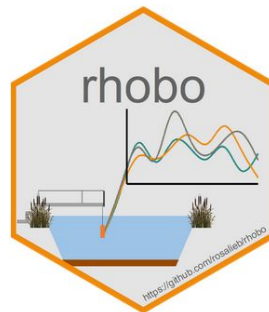
The steps match the routine for data from the artificial lakes at the [CEREEP-Ecotron](https://www.cereep.bio.ens.psl.eu/spip.php?article45), and may need to be adapted to other datasets format to be widely usable.

Authors: Rosalie Bruel (iEES-Paris, CNRS, France) and Sophie Guillon (Mines ParisTech, France)

Installation

You can install the development version from [GitHub](https://github.com/) with:

```
# install.packages("devtools")
devtools::install_github("rosalieb/rhobo")
library(rhobo)
```



[List of content](#)

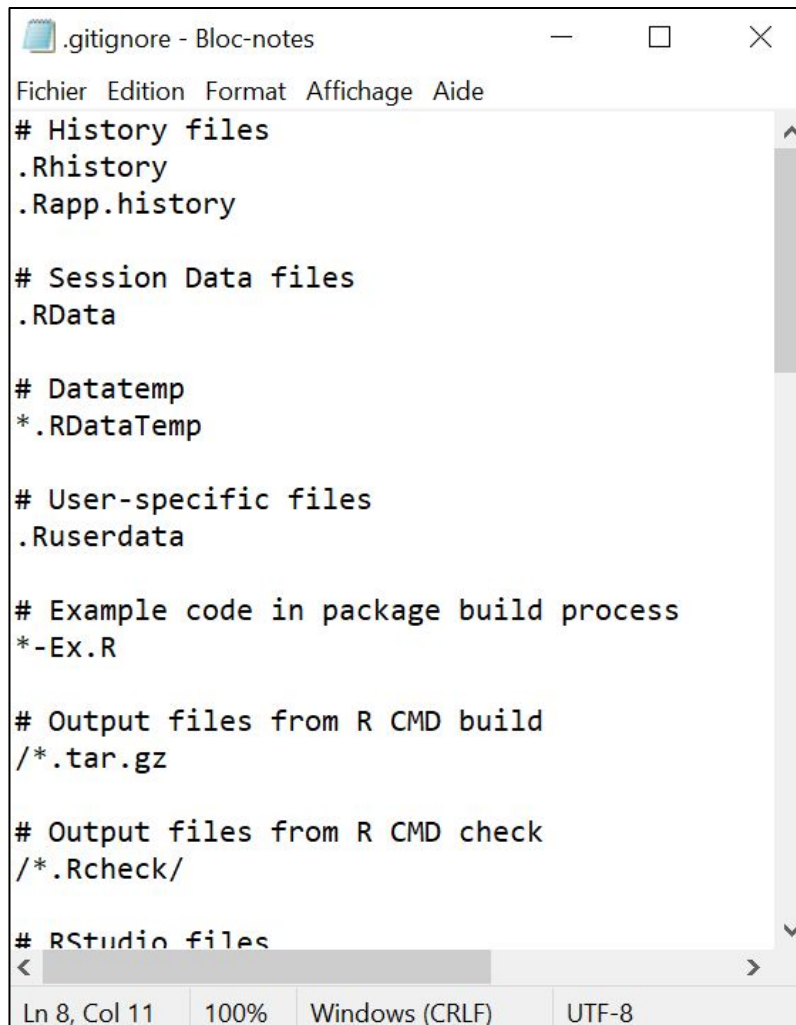
gitignore.txt

A text file, with the list of folders and/or extensions that Git need to omit (i.e., not push).

Can support complex syntax:

```
/Output/Figures/*  
!/Output/Figures/toshare/  
!Output/Figures/toshare/*
```

This tell us that all the Files in Output/Figures should NOT be pushed, except if they are in Output/Figures/toshare

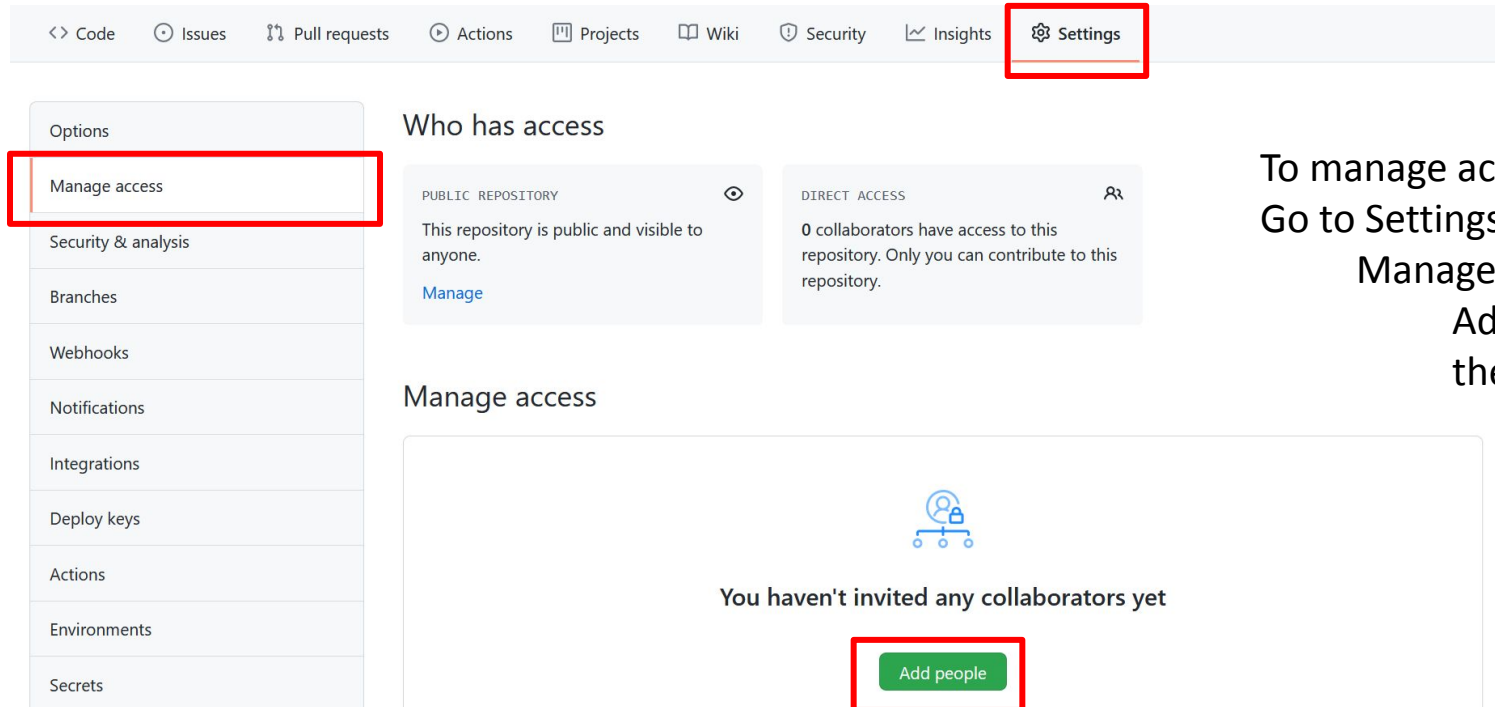
A screenshot of a text editor window titled ".gitignore - Bloc-notes". The window has a menu bar with "Fichier", "Edition", "Format", "Affichage", and "Aide". The content of the file is as follows:

```
# History files  
.Rhistory  
.Rapp.history  
  
# Session Data files  
.RData  
  
# Datatemp  
*.RDataTemp  
  
# User-specific files  
.Ruserdata  
  
# Example code in package build process  
*-Ex.R  
  
# Output files from R CMD build  
/*.tar.gz  
  
# Output files from R CMD check  
/*.Rcheck/  
  
# RStudio files
```

The status bar at the bottom shows "Ln 8, Col 11", "100%", "Windows (CRLF)", and "UTF-8".

Access to a repository

- A **public repository** can be cloned and forked by anyone.
- Only collaborators can commit onto a public repository.
- You have to be a collaborator to clone, fork, commit, etc. a **private repository**



The screenshot displays the GitHub repository settings interface. At the top, a navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Settings link is highlighted with a red box. On the left sidebar, the 'Manage access' option is also highlighted with a red box. The main content area is titled 'Who has access' and contains two panels: 'PUBLIC REPOSITORY' and 'DIRECT ACCESS'. The 'PUBLIC REPOSITORY' panel states that the repository is public and visible to anyone, with a 'Manage' link. The 'DIRECT ACCESS' panel shows that 0 collaborators have access. Below these panels is a 'Manage access' section with a message: 'You haven't invited any collaborators yet' and a green 'Add people' button, which is highlighted with a red box. The left sidebar lists various settings categories: Options, Manage access, Security & analysis, Branches, Webhooks, Notifications, Integrations, Deploy keys, Actions, Environments, and Secrets.

To manage access:

Go to Settings >

Manage access >

Add people (by
their username)

Limiting who can contribute to the master branch avoids breaking the code (otherwise you and I could just go and break up code). Someone must review your pull-request (merge-request).

desktop / desktopPublic

Watch1.5kStar13.3kFork7k

<> CodeIssues620Pull requests14ActionsProjects2SecurityInsights

development136 branches399 tags

Go to fileAdd fileCode

niik Merge pull request #13177 from desktop/linkbutton-tooltips ...8565b125 hours ago30,739 commits

.githubRead-only GITHUB_TOKEN for CI jobslast month

.vscoderemove unused vscode setting4 months ago

appUse custom tooltips for link buttons7 hours ago

docsfeat: add syntax mapping for HAML - feature #13000last month

About

Simple collaboration from your desktop

[desktop.github.com](#)

electrongithubgittypescriptgithub-desktop

ReadmeMIT License

Releases

399 tags

Contributors250

+ 239 contributors

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

The repository network shows the 100 most recently pushed forks. Do you need to see more forks? Please [give us feedback](#) on your usage of this feature.

Owners

2425

Forks (alternative branches)

250 contributors

[List of content](#)

Issues (1/2)

- bring to the attention of the owner(s) of the repository somme issues with the development. Can have assignees, tags, images can be uploaded, etc.
- Can be useful to assign tasks

The screenshot displays the GitHub interface for the repository **sfigary / GLEON_ZIG**. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The repository name is followed by a 'Private' label and statistics: 6 Unwatch, 0 Stars, and 1 Fork. Below this, a tabbed interface shows 'Issues' with 11 items, alongside Pull requests, Actions, Projects, Wiki, Security, and Insights. A search bar contains the query 'is:issue is:open'. To the right of the search bar are filters for Labels (9) and Milestones (0), along with a 'New issue' button. The main content area shows a list of issues with columns for checkboxes, status (11 Open, 1 Closed), and sorting options (Author, Label, Projects, Milestones, Assignee, Sort). The first three issues are:

- ☐ **Litt/Degasperi Lake Washington - data cleaning**
#12 opened 16 days ago by sfigary
- ☐ **Scordo Castle - Data Cleaning**
#11 opened on 1 Sep by sfigary
- ☐ **General issues**
#9 opened on 25 Aug by sfigary

Below these, the fourth issue is partially visible:

- ☐ **Tartarotti Lake Piburg - Data Cleaning**
#8 opened on 17 Aug by sfigary

Issues (2/2) - example of an issue in github.com/desktop/desktop

Create Tag with Message #12995

New issue



facundo91 opened this issue on 20 Sep · 1 comment



facundo91 commented on 20 Sep



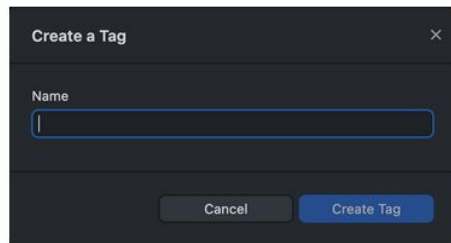
Describe the feature or problem you'd like to solve

It's not possible to create a new Tag with a Message on it. We use the message as part of our CI/CD pipeline, therefore, we can't use Github Desktop for our daily work.

Proposed solution

Add a Message Input Text Field in the Create Tag dialog.

Additional context



Assignees

No one assigned

Labels

enhancement

Projects

None yet

Milestone

No milestone

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

Notifications

Customize



You're not receiving notifications from this thread.

3 participants



sergiou87 added the enhancement label on 21 Sep

3 participants



ADustyOldMuffin commented

Contributor



29 days ago



I'd be interested in taking this one on if it's available for community pickup.



1



Write

Preview

H B I

@

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Comment

Remember, contributions to this repository should follow its contributing guidelines and code of conduct.

Other resources:

[Jenny Bryan's Happy Git with R:](#) Chapter 1 introduces Git and its benefits

[Git in R:](#) Lecture notes from Dr. Nina Overgaard Therkildsen's Collaborative and Reproducible Data Science in R course at Cornell University. Lessons 2-5 provide step-by-step instructions for setting up Github and pairing it with Rstudio. These links are shared with permission from Dr. Nina Overgaard Therkildsen.

[GLEON's GitHub workshop](#) by Joe Stachelek

Build a R package and share it on GitHub <https://sahirbhatnagar.com/rpkg/#building-an-r-package>

Create your website in Rmarkdown and publish it via GitHub:

https://www.emilyzabor.com/tutorials/rmarkdown_websites_tutorial.html

Github project management:

<https://openscapes.github.io/series/github-issues.html>

GitHub vs. GitLab vs. Bitbucket <https://jelvix.com/blog/bitbucket-vs-github-vs-gitlab>