

Manuscript

1. Introduction

2. Setting Up a Collaborative Workspace

- How to create a repository in Git Hub
- How to link it to Rstudio
- How to create a project in your local computer and export it to GitHub
- Basic commands for Git and GitHub from the RStudio terminal

A collaborative workflow requires a collaborative work space that enables everyone participating to share and contribute to a project.

There are multiple options for such a workspace, like Teams, Discord, or even over e-mail. When working with code however, online repositories on GitHub can be a good alternative.

In this tutorial we assume that you will be working with RStudio and have already downloaded R and Rstudio. In addition you are going to need to have the version control software Git installed and have an account in GitHub. If you need guidance for this you can find a helpful tutorial [here](#).

What is a repository?

A repository is basically like a project box where you collect all the files, data, graphs and code scripts from your project.

Online repositories can be accessed from the internet and from any computer, while a local repository is only stored in a specific computer and cannot be accessed elsewhere. When setting up a collaborative work space its advantageous to have an online repository so that multiple people can contribute from their own computer to a shared repository, without having to send files by mail etc. In addition we can connect the online repository what a local repository which allows us to work and make changes using our own computer and then we can upload it to the online repository.

What is Git?

Git is a version control software that allows you to track the different version of your files. It basically allows you to keep a detailed history of changes you have done in your document and also what other people have added or removed in your collaborative documents. Having a version controls software set up for your workflow is very handy as it prevents major losses of documents and changes, and if any error is introduced in a document or code, you can track it back to see what and who submitted it. This fosters reproducibility, transparency, collaboration and robustness for your project.

What is GitHub?

GitHub is a collaborative online platform that allows you to host and join online repositories. Its kinda like facebook for coding. GitHub allows us to share and collaborate with the people on the same code at the same time. It can also be used to host webpages and other stuff.

In this tutorial we will only work with the RStudio interface and the online GitHub interface. However, if you want an expanded commandline and interface for GitHub you can use GitHub CLI and/or GitHub Desktop. See tutorials here: [GitHub CLI](#) and [GitHub Desktop](#).

How to create a project that is connected between RStudio and GitHub?

When creating a new project and you want to link your local project with an online repository, you can go about it two ways basically.

- a) You can create the online repository and then clone it down to your computer
- b) You can create a local repository and then push it online to GitHub

We'll go through both options here, starting with the online repository.

1. Create a new online repository in GitHub ([Video tutorial](#))

Image file path:

resources/images/

Once you've logged into GitHub, navigate to the top right corner of your page and find the + tab. Drop it down to reveal the "New repository" option. Click on it.

[Creating a new online repository](#)

This will take you to the repository creation page.

Here you give your repository a name, a description of what it will entail and wherever it is public or not.

You also have the options of adding a README file and a .gitignore file upon creation, but it is possible to create these after the repository is made as well.

[!Repository setup page](#)

README

A README file is a descriptive file that should explain what the project/repository is about, how it is organized and what the data in it means etc. Any additional information you want people to know when using your repository should go into the README.

.gitignore

The .gitignore file is an information file that tells Git what types of files it should track, or specifically not track. This is useful when you for example dont want to track the generated images or graphs from your code, but just your code.

when doing a commit on a file that has been staged, that version of the file goes into the version history. It is also tracked.

- we make snapshots of the files using commit

git push - sed things to remote directory on git hub

git pull - sends files from remote to your working directory

git fetch - sends remote file to your local repository

git merge - moving and merging from local repository to working directory

concsol - communicate with R

terminal - communicate with Git and Git hub and quarto (the command promt)

command = git status

(in terminal)

- shows what is in the repository and what is committed and
- adding a *and then the file type (feks *.Rproj) allows you to igor all files of that type

read me file is added to the list of files and the file is opened in the editor. when writing in the README file git hub will convert it to a htlm file.

.md filending stands for markdown

3. Conducting Simulations Before Data Acquisition

4. Including Data Packages for Distribution

5. Creating Visualizations from Data Packages