An Introduction to Git and Github

Center for Development Research (ZEF)

Jonas Guthoff[1]

April 12, 2023

[1] Center for Development Research (ZEF), University of Bonn

Before the session

1.) Do you have a GitHub.com account? If not, please go to https://github.com, join and sign up.

2.) Please share your GitHub username with Jonas (via slack or Email)

3.) Have you installed GitHub Desktop? If not, please go to https://desktop.github.com and download it.

4.) Have you logged in at least once on GitHub Desktop? If not open GitHub Desktop and log in using your GitHub account.

5.) Have you been invited to the `github.com/jonasguthoff/ZEF_test` repository? And have you accepted the invitation?

Introduction to the session

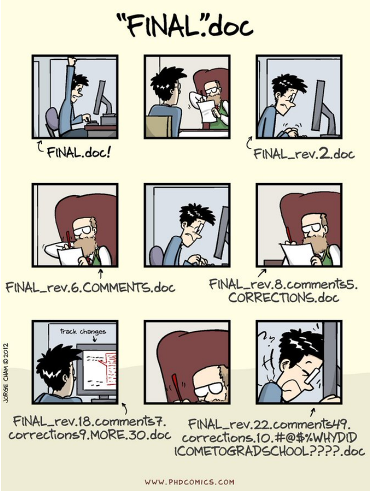
Goals of the session:

1.) Today’s session will not make you an expert in Git and GitHub, but it will teach you how to use the fundamental building blocks of Git. This may or may not be enough depending on which role you will have.

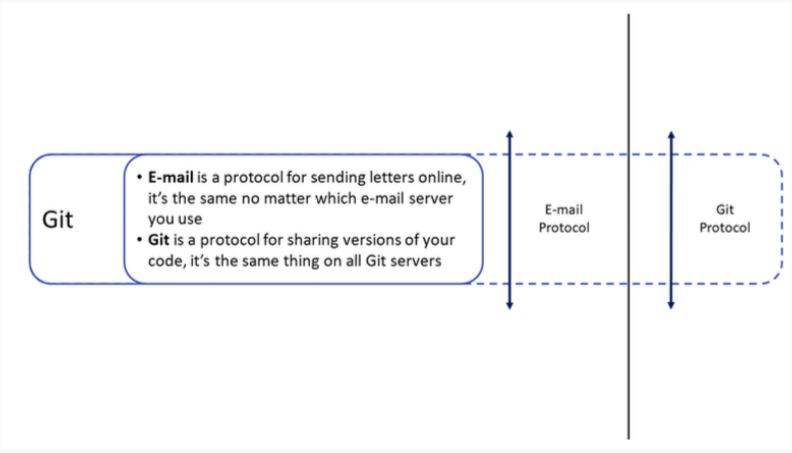
2.) Learn the fundamentals of the `iefolder` structure and how to work with it.

What is Git used for?

- Git solves the Final.doc problem
- Common solution to the Final.doc problem. Name all your docs like *YYMMDD_docname_INITIALS.doc*
- Git tracks YYMMDD and INITIALS for all edits without the user having to remember it.
- That's far from everything, Git also solves:
  - Conflicting copy problem (DropBox etc.)
  - I can't re-produce my Baseline report problem
  - Who wrote this code 4 years ago and why?
  - And much much more...

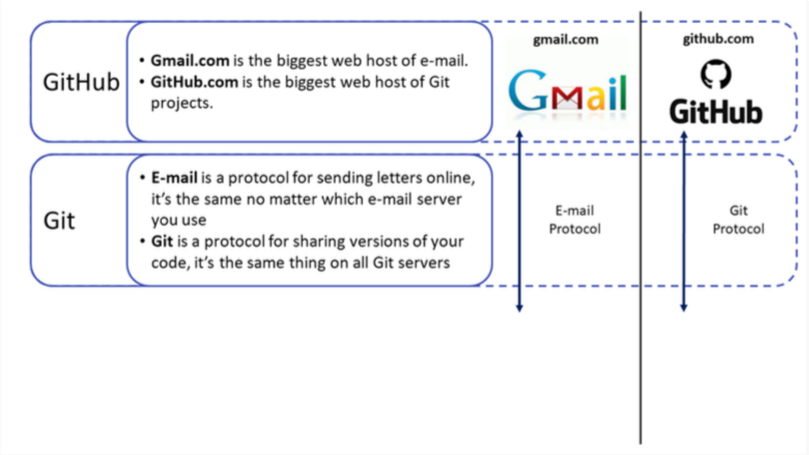

"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

What is Git, GitHub and GitHub Desktop?



- **E-mail** is a protocol for sending letters online, it's the same no matter which e-mail server you use
- **Git** is a protocol for sharing versions of your code, it's the same thing on all Git servers

What is Git, GitHub and GitHub Desktop?



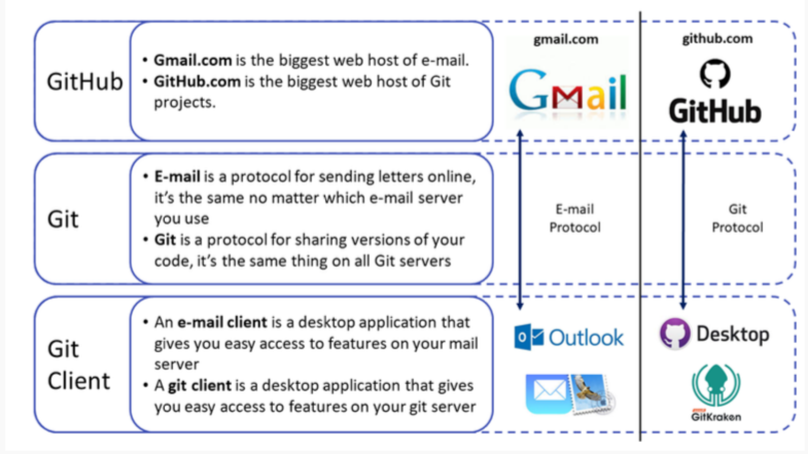
GitHub

- **Gmail.com** is the biggest web host of e-mail.
- **GitHub.com** is the biggest web host of Git projects.

Git

- **E-mail** is a protocol for sending letters online, it's the same no matter which e-mail server you use
- **Git** is a protocol for sharing versions of your code, it's the same thing on all Git servers

What is Git, GitHub and GitHub Desktop?



GitHub

- **Gmail.com** is the biggest web host of e-mail.
- **GitHub.com** is the biggest web host of Git projects.

Git

- **E-mail** is a protocol for sending letters online, it's the same no matter which e-mail server you use
- **Git** is a protocol for sharing versions of your code, it's the same thing on all Git servers

Git Client

- An **e-mail client** is a desktop application that gives you easy access to features on your mail server
- A **git client** is a desktop application that gives you easy access to features on your git server

What will we learn during this session?

In this introduction to Git and GitHub we will learn to:

- Explore the history of a project folder in GitHub and see what different team members are currently working on
- Download a project folder from GitHub so you can work on it
- Create a space in the project folder where you can make your edits
- Make edits and share those versions with your team.

Key Concepts

There are three Git concepts needed to do this:

- Clone
- Commit
- Branch

Clone

What is cloning?

Cloning is similar to downloading a **repository** to your computer.

The difference between cloning and downloading is that **when Git clones a repository it remembers where you downloaded it from.** This is necessary so that Git knows where to share the edits you make to the files in the repository.

How do I clone a repository from GitHub?

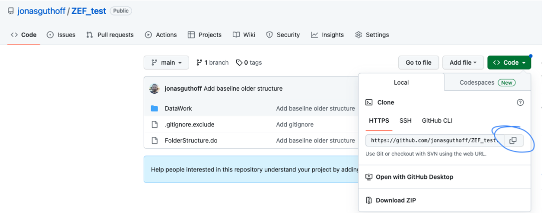
How to clone a repository:

1.) Create a folder on your computer, under **Home** named `Github`

2.) Go to the **main page** of `github.com/jonasguthoff/ZEF_test`.

3.) Click on the green Code button and Clone or download, then click on the 2 squares next to the url (see image) to save to clipboard

5.) Open Github desktop, click on `File -> Clone repository..` and paste the url from the clipboard



jonasguthoff Add baseline older structure

DataWork	Add baseline older structure
.gitignore.exclude	Add gitignore
FolderStructure.do	Add baseline older structure

Help people interested in this repository understand your project by adding

`https://github.com/jonasguthoff/ZEF_test.`

Use Git or checkout with SVN using the web URL.

Explore the clone

Explore the clone!

- The structure is created using the `iefolder` command, which is part of hte `ietoolkit`. You can install the package typing `ssc install ietoolkit` in your Stata console.

Collaboration on a Repository

Collaboration on a repository

In order to collaborate on a repository, we need to introduce two topics:

Commits

Branches

Commit

Collaboration on a repository

Instead of having a list of each saved version of a file, in Git we use commits to indicate what is each meaningful difference between two versions of our project folder.

Each commit is a snap shot of all files in the project folder, and lists how that snap shot differ from the previous snap shot (the previous commit).

Each commit has a time stamp and tracks who did the commit. This is very similar to the *YYMMDD_docname_INITIALS.doc* solution to the Final.doc problem.

How to make a commit

We need to introduce branches before we can all commit to the same repository, so I for now, let me show you how to make a commit:

1. I add a new ***do file*** file in the clone
2. I use GitHub desktop to commit the new file to the repository
3. Can you see the new file on your computer?
4. Can you see it if you sync in GitHub Desktop?

Exploring commits

Now when we know what a commit is, we can start exploring how the `github.com/jonasguthoff/ZEF_test` repository was created.

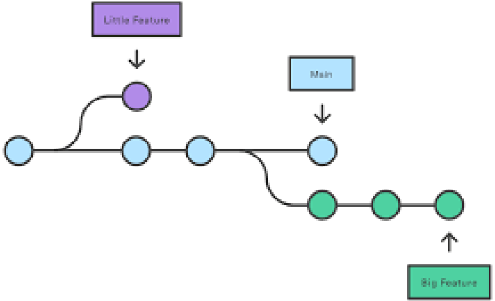
We will see a list of commits, that at first sight is similar to the the version history for example on DropBox, but in **Git the version list is more meaningful, as it is a list of only meaningful differences**.

Branch

Introducing branches

Branches are the ”killer feature” of Git. This is where Git becomes really powerful as a collaboration tool and as version control.

Branches allow you to create a copy of the code where you can experiment, if you like the result, **you can very easily merge your experiment with the main version of your code.**



Looking at branches

One more way to explore the repository

- https://github.com/jonasguthoff/ZEF_test/commits
- https://github.com/jonasguthoff/ZEF_test/network

Exploring branches

- Which version is in the clone on your computer? They are all actually in your clone, but only one is shown - checked out - at the time
- What happens to the content of the folder on your computer when you check out another branch in GitHub Desktop?

Working with branches

A typical Git work flow involves multiple branches and there are tools in GitHub to makes that work flow easy, but that is not within today's scope. Although, what you should know after this training is only how to create your own branch and how to commit to it.

Create a branch:

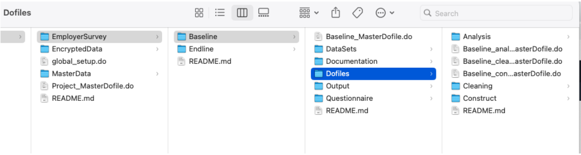
- Go to `https://github.com/jonasguthoff/ZEF_test` and click the button where it says Branch: master.
- Write your name in the field and click Create branch: your name. Make sure it says from 'master'.
- See how the button now says Branch: your name
- Go to Github Desktop and check out your branch.

Combining Commit & Branch

Now it is time to collaborate

Now it is time for you to collaborate:

1. Make sure your branch is checked out in GitHub Desktop.
2. Open Stata, create a new do file and add some content.
3. Save the do file in one of the baseline or endline dofile folders (**Cleaning, Construct, Analysis**) and close it.



Dofiles

EmployerSurvey
EncryptedData
global_setup.do
MasterData
Project_MasterDofile.do
README.md

Baseline
Endline
README.md

Baseline_MasterDofile.do
DataSets
Documentation
Dofiles
Output
Questionnaire
README.md

Analysis
Baseline_anal...asterDofile.do
Baseline_clea...asterDofile.do
Baseline_con...asterDofile.do
Cleaning
Construct
README.md

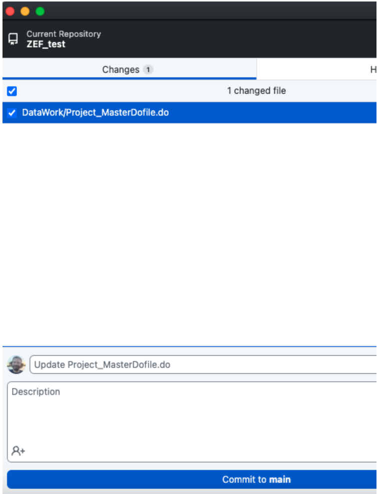
First commit

1. Open the changes tab in GitHub Desktop
2. GitHub Desktop tracks your clone and has noticed that you changed something in it
3. Then you need to do the three steps required to commit a file to the repository:

   3.1 Make sure the file you want to add is checked

   3.2 Write a commit message and click Commit to master

   3.3 Click the sync button



Current Repository
ZEF_test

Changes 1

☑ 1 changed file

✓ DataWork/Project_MasterDofile.do

Update Project_MasterDofile.do

Description

Commit to **main**

Check your contribution

Check your commit on GitHub:

- Go to https://github.com/jonasguthoff/ZEF_test/network
    - Can you find your commit?
- Go to https://github.com/jonasguthoff/ZEF_test/commits
    - Can you find your commit?

Next steps for the research team

Next steps for us as a research team

Before adopting Git, we may want to discuss the following items:

1. Agree on work flow (branches)
2. Where do we store data and where code?