gesis Leibniz Institute for the Social Sciences

Open Methodology in practice: Reproduzierbare Forschung mit R

Dr. Tobias Heycke
28.06.2019

Part 3: git

Schedule

- R Markdown & Code coding
- Git
- GitHub

Repetition

Table and Figures

Any questions concerning tables and figures?

More R Markdown comments

- if something does not work, try again
- if something does not work, close and restart
- if somethign does not work, update R, R Studio and packages

Task

- Take another look at your style guide
- Work on your Rmd from yesterday for 10 more minutes.
- Then swap your Rd file from yesterday with your neighbor!
- Run the file of your neighbor (they are not allowed to give feedback)
- Give feedback!

Bash (Unix shell)

Bash

- Command processor
- Bash is a Unix Shell
- BASH = Bourne Again SHell
- Part of GNU project

How to use it?

- On Windows: installed with git (search for Git Bash)
- R Studio: Terminal (Shift+Alt+T, Tools > Global Options > Terminal > Shell > Git Bash)

Commands: cd

Move the directories

- `cd absolute pathname` (move to this path)
- `cd relative pathname` (move to this path)
- `cd ../` (back one folder)

Use tab and double tab for autocomplete/show all files

Commands: ls

Lists all files in current working directory

- `ls -l` (long format)
- `ls -a` (also display hidden files)
- can be combined: `ls -l -a` or `ls -la`

Commands: more

The shell has many functionalities (e.g., moving, copying files and counting words in documents). What I also use often:

- `rm filname` (remove file!)
- `head filename` (show head of file)
- more info: https://librarycarpentry.org/lc-shell/

Nano I

- Try to create a text file without a name (just an ending)
- Does not work using Windows Explorer
- Nano is a (very) simple editor
- Does not work as well using the ‘R Studio Terminal’

Nano II

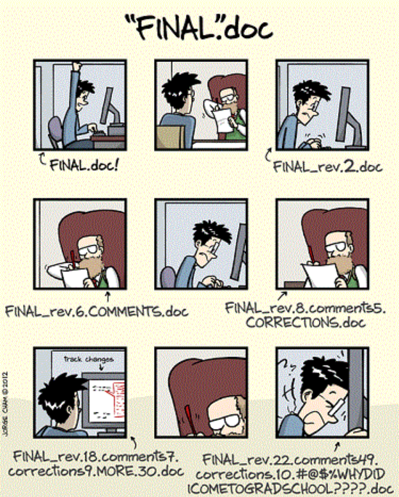
- create/open file: `nano name.ending`
- Save File: `Ctrl + O` then `Enter` (does not work in R Studio, use Ctrl + X instead)
- Close editor: `Strg + x`

Task

- go to a folder of your choosing
- display all files in the folder
- create a file called `.gitignore`
- display all files in the folder
- remove the file `.gitignore`

git

Version controle



We need the following software

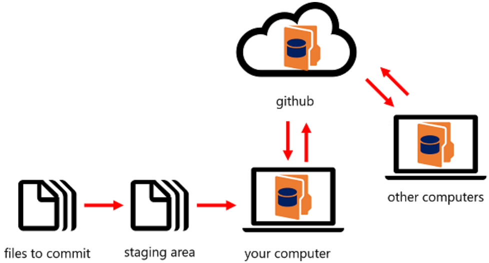
- R (r-project.org)
- RStudio (rstudio.com)
- git (git-scm.com)
- account at github.com

What is git?

- Original author: Linus Torvalds
- Version Control
- History of all changes (who, what, when?)
- Merging changes of multiple collaborators in one file
- Processes tracking offline
- Free and open source

Features of git

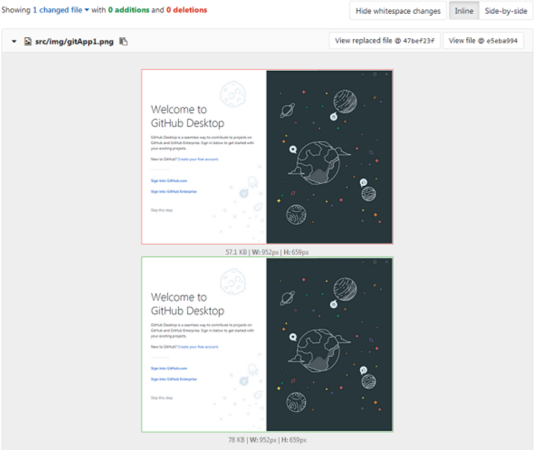
- Reverting changes
- Compare versions
- Use a git server to collaborate and back-up files



Text files

- git can only integrate and show changes in text files
- binary files (images, etc.) can be tracked and uploaded but changes cannot be shown in GUI or online
- Track changes for MS Word is improving
- max file size: 100 MB
- max repo size: ~1GB

Track changes binary files



Showing 1 changed file with 0 additions and 0 deletions

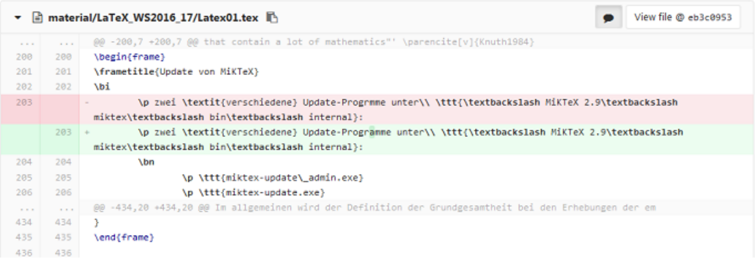
src/img/gitApp1.png

57.1 KB | W: 952px | H: 659px

78 KB | W: 952px | H: 659px

Hide whitespace changes
Inline
Side-by-side
View replaced file @ 47bef23f
View file @ e5eba994
Welcome to GitHub Desktop
Welcome to GitHub Desktop

Track changes text file

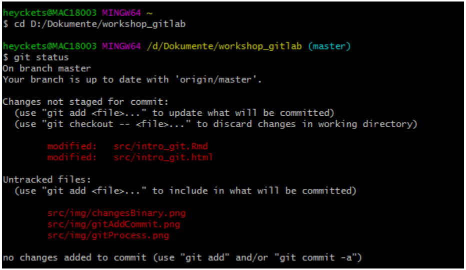


```
@@ -200,7 +200,7 @@ that contain a lot of mathematics"' \parencite[v]{Knuth1984}
 \begin{frame}
 \frametitle{Update von MiKTeX}
 \bi
-        \p zwei \textit{verschiedene} Update-Progrmme unter\\ \ttt{\textbackslash MiKTeX 2.9\textbackslash miktex\textbackslash bin\textbackslash internal}:
+        \p zwei \textit{verschiedene} Update-Programme unter\\ \ttt{\textbackslash MiKTeX 2.9\textbackslash miktex\textbackslash bin\textbackslash internal}:
         \bn
                 \p \ttt{miktex-update\_admin.exe}
                 \p \ttt{miktex-update.exe}
@@ -434,20 +434,20 @@ Im allgemeinen wird der Definition der Grundgesamtheit bei den Erhebungen der em
 }
 \end{frame}
```

Git Bash

Git can be used with the git bash or a GUI

We will use RStudio as a GUI (other options: GitHub & GitKraken)



```
heyckets@MAC18003 MINGW64 ~
$ cd D:/Dokumente/workshop_gitlab

heyckets@MAC18003 MINGW64 /d/Dokumente/workshop_gitlab (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   src/intro_git.Rmd
        modified:   src/intro_git.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        src/img/changesBinary.png
        src/img/gitAddCommit.png
        src/img/gitProcess.png

no changes added to commit (use "git add" and/or "git commit -a")
```

Setting up git on your computer

Install git (git-scm.com)

Set your commit email address

- Open the Git Bash
- Set an e-mail address in Git

```
git config --global user.email "tobias.heycke@gesis.org"
```

https://help.github.com/articles/setting-your-commit-email-address-in-git/

Set your commit Name

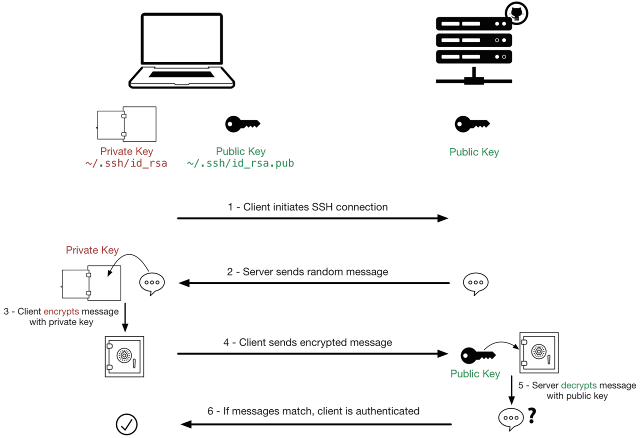
- Open the Git Bash
- Set your name in Git

```
git config --global user.name "Tobias Heycke"
```

https://help.github.com/articles/setting-your-username-in-git/

Authentification using an ssh key

Authentification using an ssh key


Private Key
~/.ssh/id_rsa
Public Key
~/.ssh/id_rsa.pub
Public Key
1 - Client initiates SSH connection
Private Key
2 - Server sends random message
3 - Client encrypts message with private key
4 - Client sends encrypted message
Public Key
5 - Server decrypts message with public key
6 - If messages match, client is authenticated
?

Source: https://sebastien.saunier.me/blog/2015/05/10/github-public-key-authentication.html

Check for exsisting ssh key

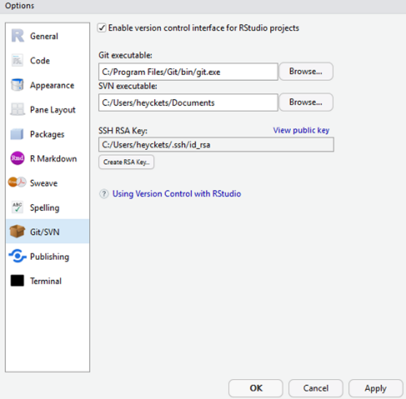
- Type in the following statement in the Git Bash

```
ls -al ~/.ssh
```

If something like “id_dsa.pub” is listed, you already have an SSH Key

Creating an ssh key using R Studio

- In RStudio > Tools > Global Options
- You don't need to enter a password



☑ Enable version control interface for RStudio projects

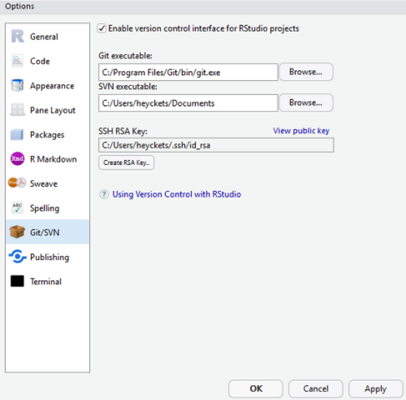
Git executable: C:/Program Files/Git/bin/git.exe

SVN executable: C:/Users/heyckets/Documents

SSH RSA Key: C:/Users/heyckets/.ssh/id_rsa

R Studio and git

Make sure the first box is ticked and the “git.exe” is included in the first box



Options

☑ Enable version control interface for RStudio projects

Git executable: C:/Program Files/Git/bin/git.exe

SVN executable: C:/Users/heyckets/Documents

SSH RSA Key: C:/Users/heyckets/.ssh/id_rsa

If ssh key exists

- Type in the following statement in the Git Bash

```
clip < ~/.ssh/id_rsa.pub [win]
pbcopy < ~/.ssh/id_rsa.pub [mac]
```

Your public key is now in your clipboard (i.e., you can now paste it using Ctrl + V)

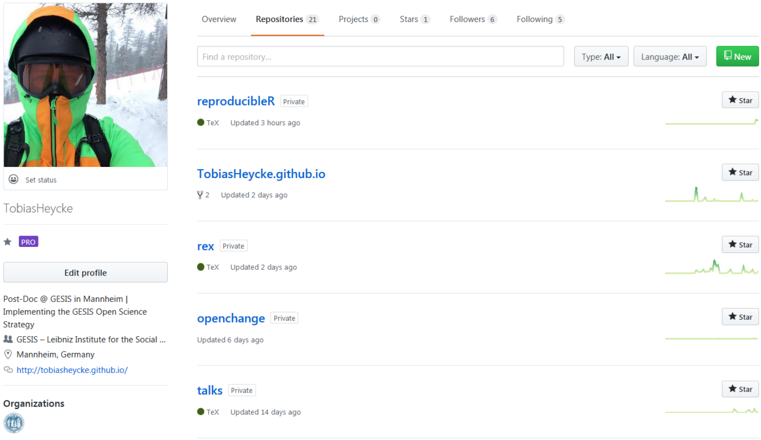
You can open the id_rsa.pub file with a text editor and paste the key

Add ssh key to GitHub

- Go to github.com
- Log in
- Go to ProfilePic > setting > SSH and GPG keys
- New SSH key [upper right corner]
- Paste public (!) key into the key box
- Give the key a name (e.g., ‘GESIS PC’)

We’re ready to use git now!

GitHub Repositories



Overview | Repositories 21 | Projects 0 | Stars 1 | Followers 6 | Following 5

Find a repository… | Type: All ▾ | Language: All ▾ | New

reproducibleR Private ★ Star
● TeX Updated 3 hours ago

TobiasHeycke.github.io ★ Star
2 Updated 2 days ago

rex Private ★ Star
● TeX Updated 2 days ago

openchange Private ★ Star
Updated 6 days ago

talks Private ★ Star
● TeX Updated 14 days ago

Set status

TobiasHeycke

★ PRO

Edit profile

Post-Doc @ GESIS in Mannheim | Implementing the GESIS Open Science Strategy

GESIS – Leibniz Institute for the Social ...

Mannheim, Germany

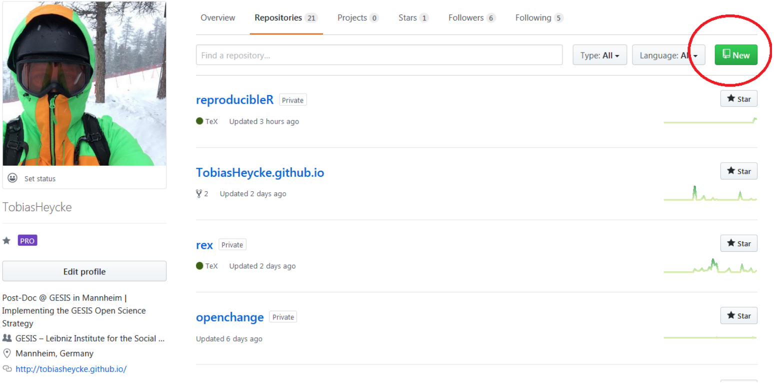
http://tobiasheycke.github.io/

Organizations

GitHub New Repository

Give it a good name - you will stick with it!

Choose if you want your repository to be public or private.



Set status

TobiasHeycke

PRO

Edit profile

Post-Doc @ GESIS in Mannheim | Implementing the GESIS Open Science Strategy

GESIS – Leibniz Institute for the Social ...

Mannheim, Germany

http://tobiasheycke.github.io/

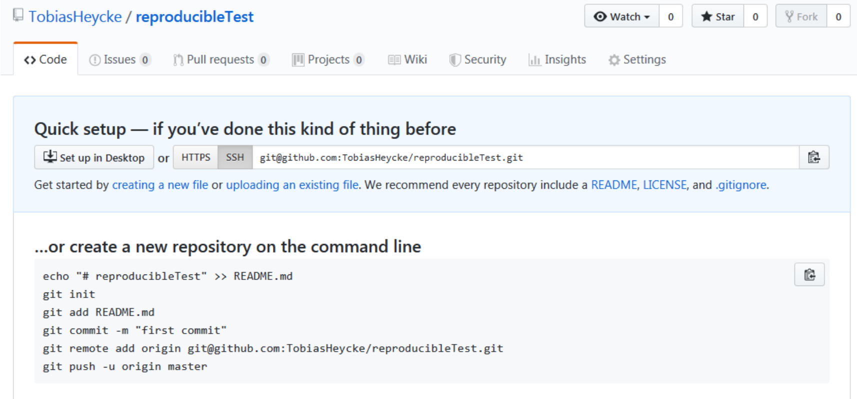
reproducibleR Private
● TeX Updated 3 hours ago

TobiasHeycke.github.io
2 Updated 2 days ago

rex Private
● TeX Updated 2 days ago

openchange Private
Updated 6 days ago

Finish New Repository



Quick setup — if you've done this kind of thing before

git@github.com:TobiasHeycke/reproducibleTest.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# reproducibleTest" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:TobiasHeycke/reproducibleTest.git
git push -u origin master
```

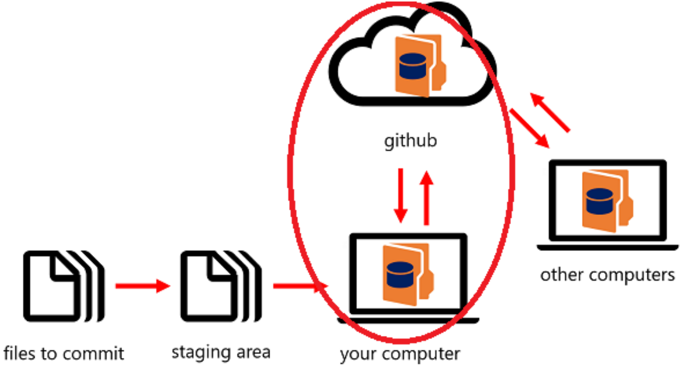
Adding new members

Go to your repository

- Settings
- Collaborators
- Add collaborators by username or e-mail address

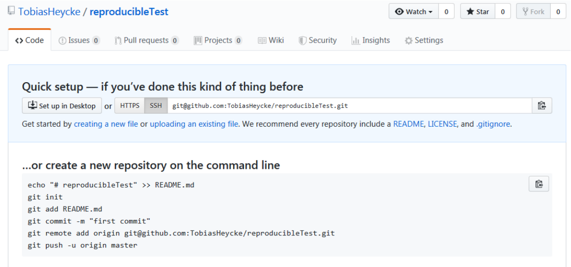
Clone a repository

Reminder


github
other computers
files to commit
staging area
your computer

Get the ssh address

Go to the projects main page and copy the ssh address



Quick setup — if you've done this kind of thing before

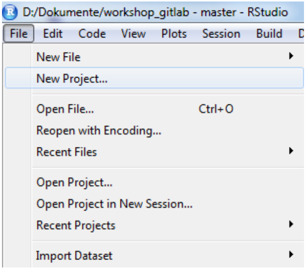
```
git@github.com:TobiasHeycke/reproducibleTest.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

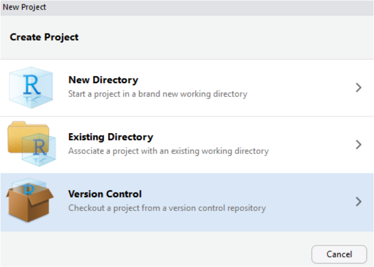
...or create a new repository on the command line

```
echo "# reproducibleTest" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:TobiasHeycke/reproducibleTest.git
git push -u origin master
```

Add a new git project with RStudio I

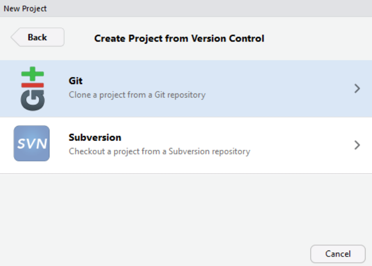


Add a new git project with RStudio II



New Project
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel

Add a new git project with RStudio III



New Project

Back

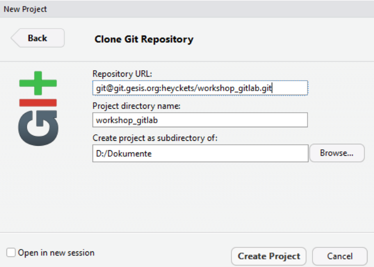
Create Project from Version Control

Git
Clone a project from a Git repository

Subversion
Checkout a project from a Subversion repository

Cancel

Add a new git project with RStudio IV



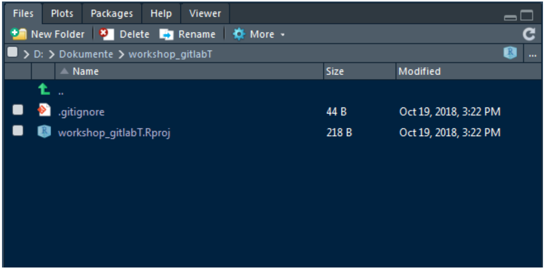
Repository URL: git@git.gesis.org:heyckets/workshop_gitlab.git

Project directory name: workshop_gitlab

Create project as subdirectory of: D:/Dokumente

Add a new git project with RStudio V

- RStudio automatically adds a Rproj file
- RStudio also creates a .gitignore file



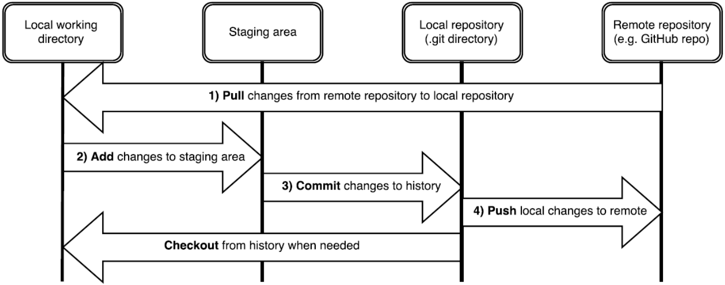
Name	Size	Modified
..		
.gitignore	44 B	Oct 19, 2018, 3:22 PM
workshop_gitlabT.Rproj	218 B	Oct 19, 2018, 3:22 PM

.gitignore

- As the name says, all files (or files with a specific ending) are ignored by git
- You can open the file and add *.Rproj to the list
- You can add any file type, folder or file to the list

Stage, Commit, Upload

Git workflow


Local working directory
Staging area
Local repository (.git directory)
Remote repository (e.g. GitHub repo)
1) **Pull** changes from remote repository to local repository
2) **Add** changes to staging area
3) **Commit** changes to history
4) **Push** local changes to remote
Checkout from history when needed

Source: Vuorre, M., & Curley, J. P. (2018). Curating Research Assets: A Tutorial on the Git Version Control System. Advances in Methods and Practices in Psychological Science, 1(2), 219–236.

Git workflow with the git bash

Stage I

Select files which you want to include in your ‘snapshot’

```
git status
```

```
On branch master

No commits yet

Untracked files: (use “git add . . . ” to include in what will be committed)

    .gitignore

nothing added to commit but untracked files present (use “git add” to track)
```

Stage II

Add the file(s)

```
git add -A [add all changed/new files]
git add .gitignore
```

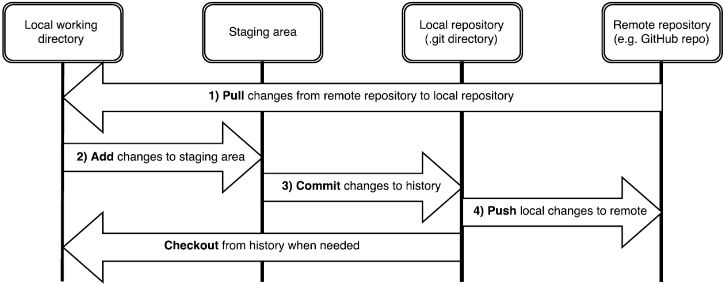
[When nothing really happens: good!]

Stage III

```
git status
```

```
On branch master

No commits yet

Changes to be committed: (use “git rm –cached . . . ” to unstage)

    new file:   .gitignore
```

Git workflow


Local working directory
Staging area
Local repository (.git directory)
Remote repository (e.g. GitHub repo)
1) **Pull** changes from remote repository to local repository
2) **Add** changes to staging area
3) **Commit** changes to history
4) **Push** local changes to remote
Checkout from history when needed

Source: Vuorre, M., & Curley, J. P. (2018). Curating Research Assets: A Tutorial on the Git Version Control System. Advances in Methods and Practices in Psychological Science, 1(2), 219–236.

Commit I

To only submit a title of the commit

```
git commit -m"first commit"
```

```
[master (root-commit) d7c25ce] first commit
1 file changed, 5 insertions(+)
create mode 100644 .gitignore
```

Commit II

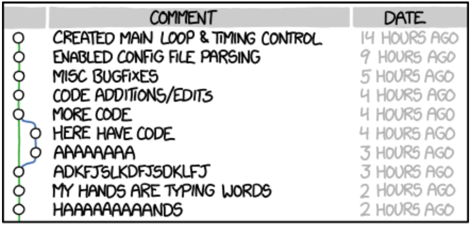
To only submit a title of the commit

```
git commit
```

Will open an editor:

- First row type title of commit
- Third row (plus) write details
- Exit with: `Esc` then type `:x!` then `enter` [win]
- Exit with: `Ctrl` + `x` [mac]

Commit message I



COMMENT	DATE
CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
ENABLED CONFIG FILE PARSING	9 HOURS AGO
MISC BUGFIXES	5 HOURS AGO
CODE ADDITIONS/EDITS	4 HOURS AGO
MORE CODE	4 HOURS AGO
HERE HAVE CODE	4 HOURS AGO
AAAAAAAA	3 HOURS AGO
ADKFJSLKDFJSDKLFJ	3 HOURS AGO
MY HANDS ARE TYPING WORDS	2 HOURS AGO
HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

Commit message II

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
- Do not end the subject line with a period
- Use the imperative mood in the subject line
- Use the body to explain what and why (not how)
- Capitalize the subject line
- Wrap the body at 72 characters

see https://chris.beams.io/posts/git-commit

Commit III

```
git status
```

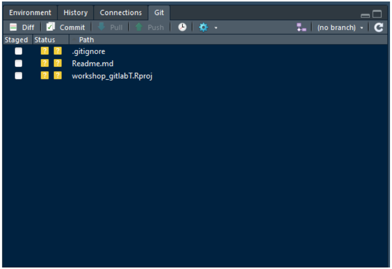
```
On branch master
Your branch is ahead of ‘origin/master’ by 1 commit.
(use “git push” to publish your local commits)
nothing to commit, working tree clean
```

Push

```
git push origin master
```

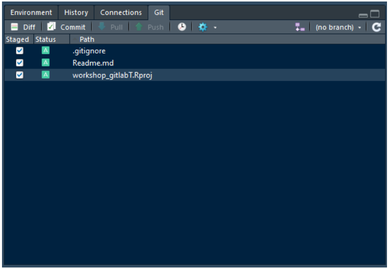
```
Enumerating objects: 3, done.

Counting objects: 100% (3/3), done.

Writing objects: 100% (3/3), 258 bytes | 258.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To github.com:TobiasHeycke/reproducibleTest.git

[new branch] master -> master
```

Git in RStudio



Staged	Status	Path
□	? ?	.gitignore
□	? ?	Readme.md
□	? ?	workshop_gitlabT.Rproj

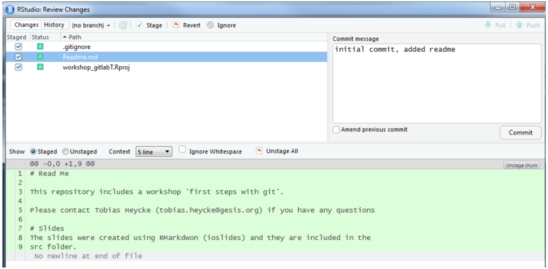
Stage



Staged	Status	Path
☑	A	.gitignore
☑	A	Readme.md
☑	A	workshop_gitlabT.Rproj

see also: http://gitolite.com/uses-of-index.html

Commit I



Staged	Status	Path
☑	A	.gitignore
☑	A	Readme.md
☑	A	workshop_gitlabT.Rproj

Commit message

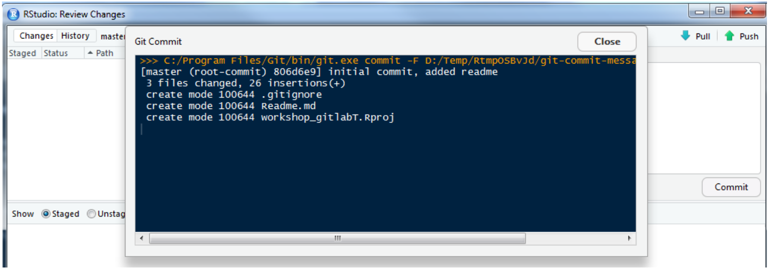
```
initial commit, added readme
```

□ Amend previous commit

Show ● Staged ○ Unstaged Context 5 line □ Ignore Whitespace

```
@@ -0,0 +1,9 @@
# Read Me

This repository includes a workshop 'first steps with git'.

Please contact Tobias Heycke (tobias.heycke@gesis.org) if you have any questions

# Slides
The slides were created using RMarkdwon (ioslides) and they are included in the
src folder.
 No newline at end of file
```

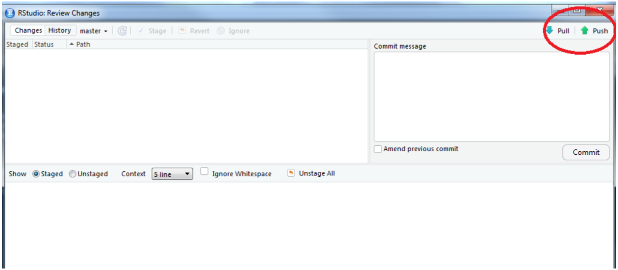
Commit II



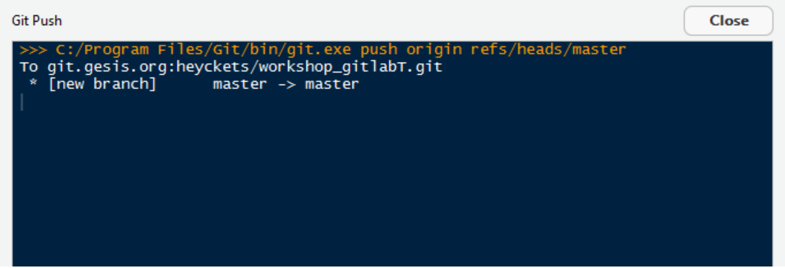
```
>>> C:/Program Files/Git/bin/git.exe commit -F D:/Temp/RtmpOSBvJd/git-commit-messa
[master (root-commit) 806d6e9] initial commit, added readme
 3 files changed, 26 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 Readme.md
 create mode 100644 workshop_gitlabT.Rproj
```

Push

Everything happened on your local machine so far. Push to upload all changes to GitHub



Push II



```
>>> C:/Program Files/Git/bin/git.exe push origin refs/heads/master
To git.gesis.org:heyckets/workshop_gitlabT.git
 * [new branch]      master -> master
```

Pull

You will download the newest state from GitHub

Always pull before starting to work!

```
git pull origin master
```

Task

- Create repo online
- Clone it with R Studio
- Add *.RProj to .gitgnore
- Stage, Commit, Push first commit

add a README file

- Usually the first step is to create a Read me file and add it to the main folder:
- Should be called for example README.md
- Include some information in the file
- One README for each (sub-)project (e.g., paper)
- The next step would be to add a licence file.

Task

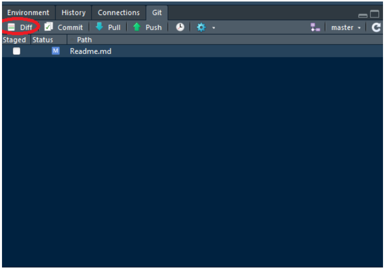
- Create a readme file in the main git folder
- Add some markdown to git
- Stage, Commit, Push first commit

Make changes to documents

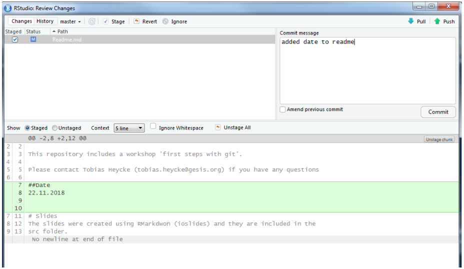
Workflow:

- Pull
- Adjust file
- Stage/Commit
- Push

See diff in R Studio I



See diff in R Studio II



Staged	Status	Path
☑	M	Readme.md

Commit message: added date to readme

□ Amend previous commit

Show ● Staged ○ Unstaged Context 5 line □ Ignore Whitespace Unstage All

```
@@ -2,8 +2,12 @@

This repository includes a workshop 'first steps with git'.

Please contact Tobias Heycke (tobias.heycke@gesis.org) if you have any questions

##Date
22.11.2018


# Slides
The slides were created using RMarkdwon (ioslides) and they are included in the
src folder.
 No newline at end of file
```

See changes online

- At github.com you can see all changes that were made
- Go to the repository
- Click on the commit massage of the README.md

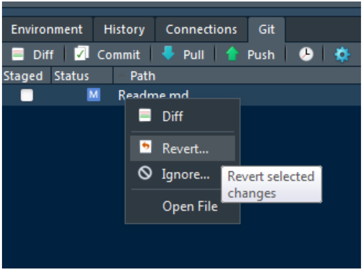
Task

- Adjust the readme file
- Stage, Commit, Push first commit
- Check out the repository on GitHub

Revert

You can revert local changes to the last committed version

Very handy if you want to try out something quickly



Collaboration

Collaboration

- Git is very useful when collaborating (especially when writing manuscripts in Rmd)
- Git can integrate changes by multiple parties
- You can see who changed what when (and why)
- You cannot accidentally delete anything

Task

- Add you neighbor as a contributor
- Adjust the read me file of your neighbor
- Add, commit, push
- Pull and check out changes made by your neighbor (what was exactly changed?)

Merge Conflicts

- When changing the same line of code, a human needs to decide which line to keep
- Remember: always `pull` before starting to work on a file!
- Git will indicate where the conflicts are by adding the following:

```
<<<<<<< HEAD

[user A's porposed version of the text]

======

[user B's porposed version of the text]

>>>>>>> 21ff8de568768eae68768bf564ae546878e
```

Merge conflicts II

- User A then can manually remove everything that is not necessary
- Then add, commit, push

See also: https://www.git-tower.com/learn/git/ebook/en/command-line/advanced-topics/merge-conflicts

Task

- Pull the repository of your neighbor
- Change something in the first row of the read me file
- add, commit, push
- Change something in the first row of your read me (do not pull the changes)
- handle the conflict

Branching

- Create copy of everything and work in it
- When changes work, merge with master branch
- Create branch by clicking on this button
- make changes
- (on branch development)$ git merge master
- (resolve any merge conflicts if there are any)
- git checkout master
- git merge development (there won’t be any conflicts now)

Issues

- You can report issues on GitHub
- Helpfull tool when collaborating

Pull request

Check out pull request for a public repository

Retrieving older versions from git

see Supplementary Material of: Vuorre, M., & Curley, J. P. (2018). Curating Research Assets: A Tutorial on the Git Version Control System. Advances in Methods and Practices in Psychological Science, 1(2), 219–236.

Links

Enjoy git!

- https://guides.github.com/introduction/flow/
- https://www.youtube.com/githubguides
- http://kbroman.org/github_tutorial
- http://happygitwithr.com
- http://meldmerge.org/

Further reproducibility steps

Further steps:

- package version (and OS)
- checkpoint and packrat
- Docker Container (see for example https://codeocean.com/2018/10/15/no-evaluative-conditioning-effects-with-briefly-presented-stimuli/)
- redoc

Shameless self promotion

More info on open science:

@TobiasHeycke 🐦

Workshop Preregistration:

https://training.gesis.org/?site=pDetails&pID=0xE7495ACCA0084D7390F3E1351C3621BF

Evaluation

End of Workshop

Thank you