

Live Demo – Connect Rstudio and Git (necessary only once)

The screenshot illustrates the process of connecting RStudio to Git. The RStudio interface is shown with the **Tools** menu open, highlighting **Global Options...**. The **Options** dialog box is then displayed, showing the **Git/SVN** section. The **Enable version control interface for RStudio projects** checkbox is checked. The **Git executable** path is set to `C:/Program Files/Git/bin/git.exe. The SSH RSA key section shows a key named C:/Users/Caro/.ssh/id_rsa with a Create RSA Key... button. The Apply button is highlighted.`

Annotations and instructions:

- 1: Tools menu
- 2: Global Options...
- 3: Git/SVN section
- 4: Enable version control interface for RStudio projects
- 5: Git executable: C:/Program Files/Git/bin/git.exe
- 6: Create RSA Key...
- 7: Apply

! You have to install git first !
On mac: /usr/local/git/bin/git

We will need the SSH Key later...

This is like a password! Keep it safe!

Live Demo – Initialize git (necessary only once)

The image shows a sequence of five steps to initialize git in RStudio:

1. Click on the **Tools** menu in the top menu bar.
2. Click on **Terminal** in the Tools menu, which opens a submenu.
3. Click on **New Terminal** in the Terminal submenu.
4. In the new terminal window, enter the command: `$ git config --global user.name "CaroZygar"`
5. In the terminal window, enter the command: `$ git config --global user.email "caro_zyg@gmx.de"`

The terminal output shows the successful configuration of git user name and email.

Live Demo – start a new project which includes git

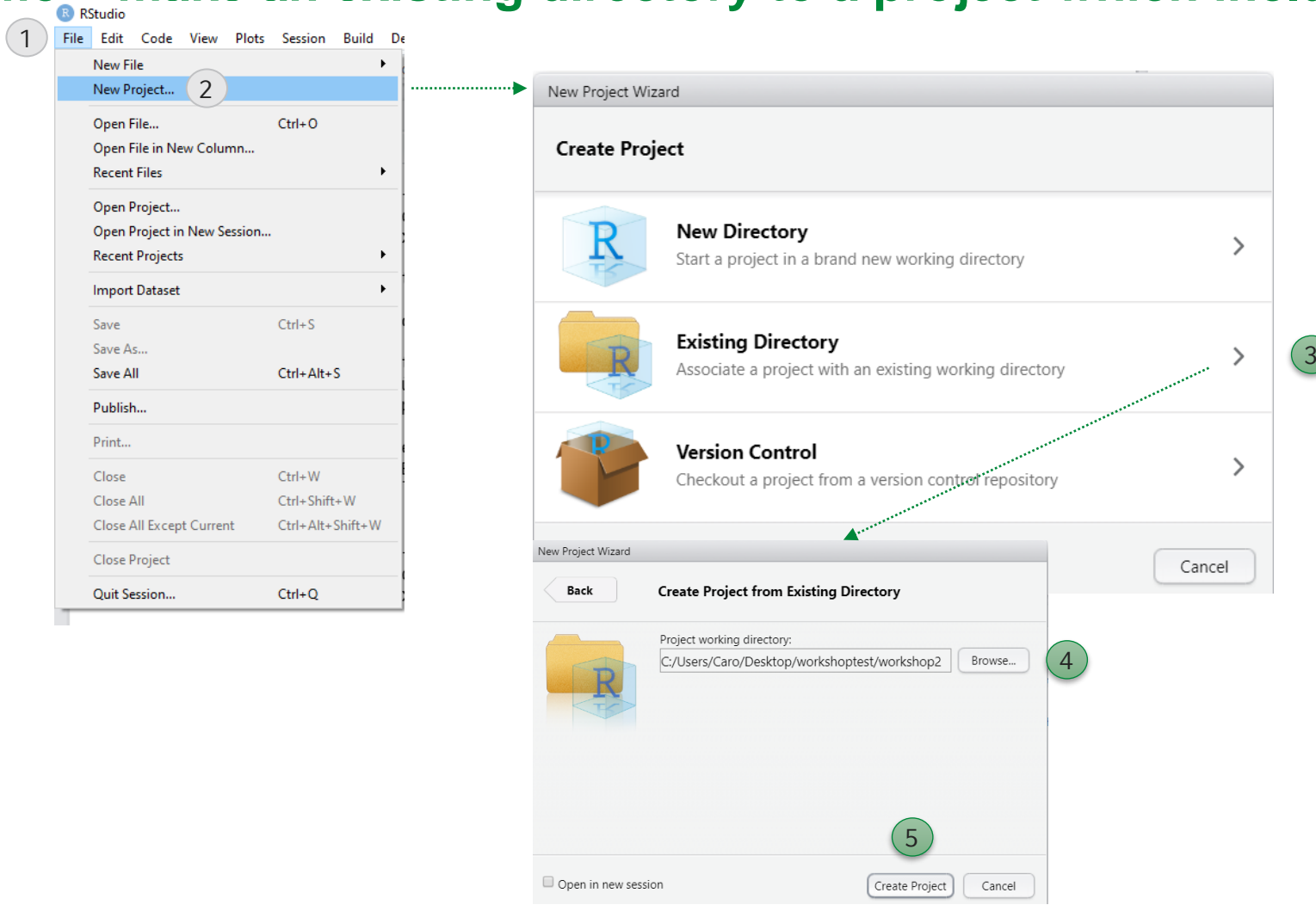
The screenshot illustrates the process of creating a new R project in RStudio, specifically one that includes a git repository. The steps are numbered 1 through 8:

- Click on the **File** menu.
- Click on **New Project...**
- In the **New Project Wizard**, select **New Directory**.
- Click on **New Project** in the **Project Type** list.
- Enter the **Directory name** (e.g., `workshop`).
- Click **Browse...** to select the parent directory (e.g., `C:/Users/Caro/Desktop/workshoptest`).
- Check the **Create a git repository** checkbox.
- Click **Create Project**.

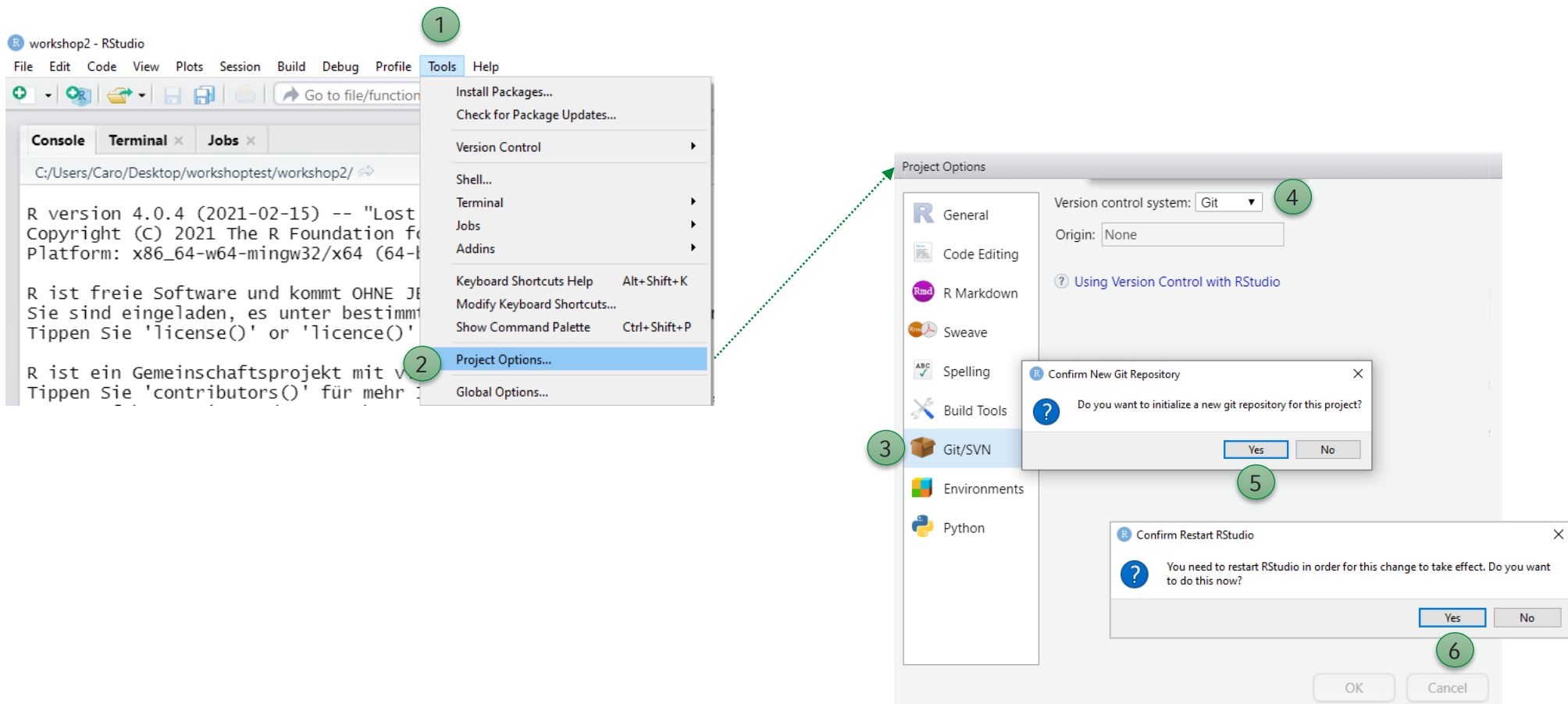
Additional annotations include:

- A callout box stating: "Irrespective of git: Using projects in R is useful, because your paths are relative to the project directory and you can re-open all associated R-scripts!"
- A note: "This will be relevant later..." pointing to the **Version Control** section of the wizard.
- A dashed arrow points from the **New Project...** menu item to the **New Project Wizard** dialog.
- A dashed arrow points from the **New Directory** option to the **Create New Project** sub-dialog.

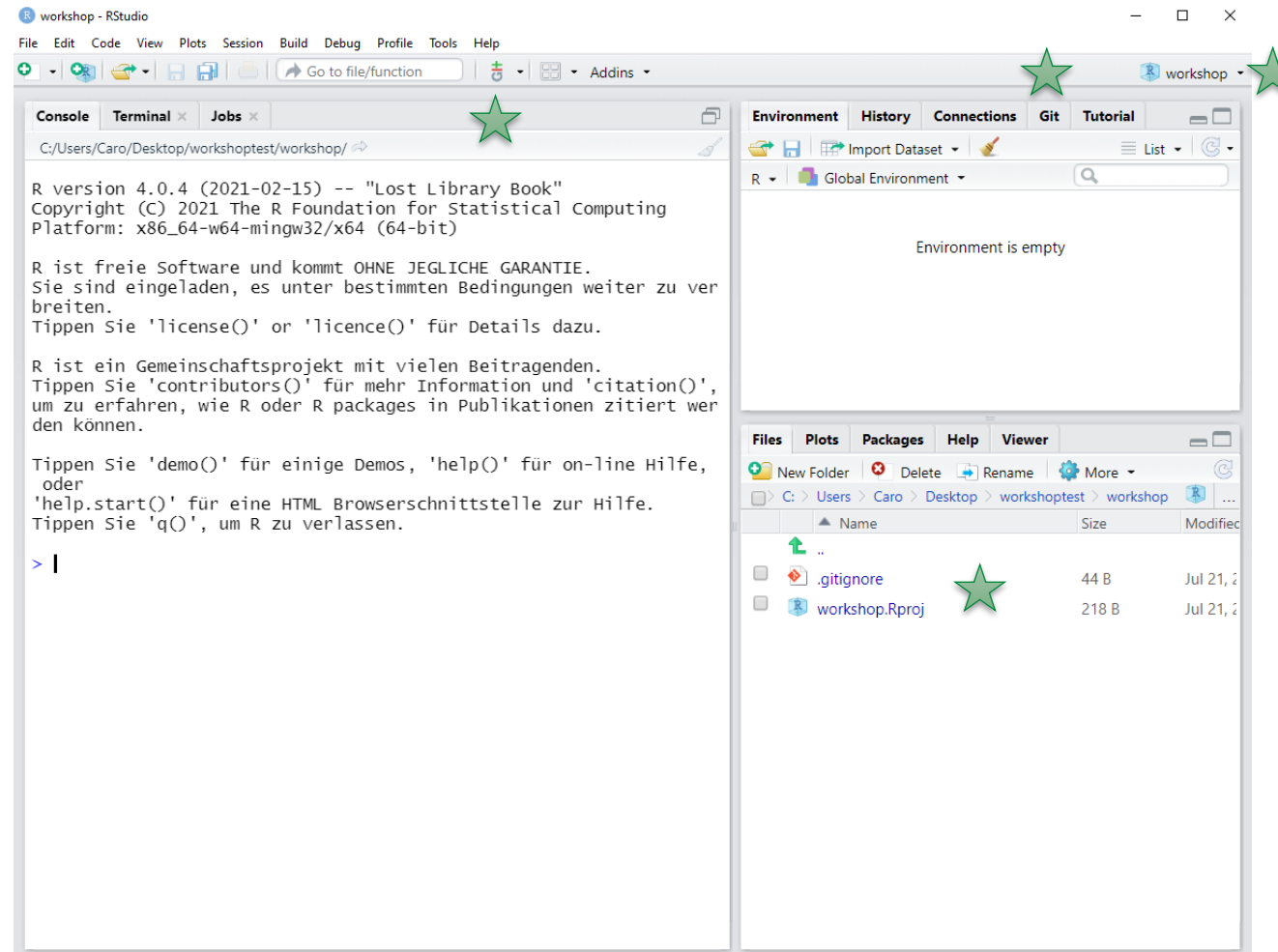
Live Demo – make an existing directory to a project which includes git (1)



Live Demo – make an existing directory to a project which includes git (2)



Live Demo – yay, we have a new project which includes git! :-)



Live Demo – committing

The screenshot illustrates the RStudio interface during a Git commit process. The main window shows the Source editor with a file named '0-preprocessing.R' (1). The Environment pane (2) displays the project files, including '.gitignore', '0-preprocessing.R', and 'workshop2.Rproj'. The Staged files section (3) shows the files to be committed. The Commit button (4) is visible in the Environment pane. The Review Changes window (5) shows the commit message 'yay, my first commit! :-)' and the 'Commit' button. The Git Commit dialog box (6) is open, showing the commit message and the files to be committed. The Close button (7) is visible in the dialog box.

workshop2 - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

0-preprocessing.R

Source on Save Run Source

1

Environment Try Connections Git Tutorial

Diff Commit

Staged Status Path

.gitignore

0-preprocessing.R

workshop2.Rproj

RStudio: Review Changes

Changes History (no branch) Stage Revert Ignore Pull Push

Staged Status Path

.gitignore

0-preprocessing.R

workshop2.Rproj

Commit message

yay, my first commit! :-)

Amend previous commit Commit

Show Staged Unstaged Context 5 lines Ignore Whitespace Unstage All

@@ -0,0 +1,13 @@

```
1 Version: 1.0
2
3 RestoreWorkspace: Default
4 SaveWorkspace: Default
5 AlwaysSaveHistory: Default
6
7 EnableCodeIndexing: Yes
8 UseSpacesForTab: Yes
9 NumSpacesForTab: 2
10 Encoding: UTF-8
11
12 RnwWeave: Sweave
13 LaTeX: pdfLaTeX
```

Git Commit

>>> C:/Program Files/Git/bin/git.exe commit -F C:/Users/Caro/AppData/Local/Temp/Rt [master (root-commit) d39794b] yay, my first commit! :-)

3 files changed, 17 insertions(+)

create mode 100644 .gitignore

create mode 100644 0-preprocessing.R

create mode 100644 workshop2.Rproj

Close

Live Demo – working in R with git

The screenshot illustrates the RStudio interface for working with Git. The main editor shows the file `0-preprocessing.R` with the following R code:

```
1 ## =====  
2 ## generate data  
3 ## =====  
4  
5 a <- c(1:5)
```

The Git pane on the right shows the file `0-preprocessing.R` staged for commit. The `Review Changes` dialog is open, showing the commit message `I did something new!`. The `Staged` pane at the bottom shows the file's diff, highlighting the changes made.

Numbered callouts indicate the following steps:

- 1: The R code in the editor.
- 2: The file `0-preprocessing.R` in the Git pane.
- 3: The diff view in the `Staged` pane.
- 4: The `Review Changes` dialog.
- 5: The commit message input field.
- 6: The `Commit` button.

Live Demo – restore earlier versions (of files that can be opened in R)

The screenshot illustrates the process of restoring an earlier version of a file in RStudio using Git. The main editor shows a file named `0-preprocessing.R` with the following content:

```
1 ## =====  
2 ## generate data  
3 ## =====  
4 |  
5 a <- c(1:5)
```

The RStudio interface includes a menu bar (File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help) and a toolbar with icons for file operations and running code. The bottom-left pane displays the 'RStudio: Review Changes' window, showing a list of commits:

Changes	History	master	(all commits)	Search	Pull
Subject	Author	Date	SHA		
I did something new!	CaroZygar <caro_zyg@gmx.de>	2021-07-21	2641cf93		
yay, my first commit! :-)	CaroZygar <caro_zyg@gmx.de>	2021-07-21	d39794b1		

The bottom-right pane shows the 'Environment' tab with a 'Diff' view. A green arrow points from the 'Diff' tab to the 'Commit' tab, which is labeled with a green circle '1'. The 'Commit' tab shows a list of commits, with the first commit (SHA: d39794b1) highlighted. A green arrow points from this commit to the '0-preprocessing.R' file in the file explorer, which is labeled with a green circle '2'. The file explorer shows the file's path and a 'Save As' button. A green arrow points from the 'Save As' button to the file's content, which is labeled with a green circle '4'. The file's content is shown in the main editor, with the first line being `1 |`. A green circle '3' is located near the bottom of the file explorer.

Live Demo – restore earlier versions (of other files)

The screenshot shows the RStudio interface with the 'Git' tab selected. The 'History' pane on the right shows a list of commits. A green arrow points from the first commit (SHA: d39794b1) to the '0-preprocessing.R' file in the editor. Below the editor, the 'RStudio: Review Changes' pane shows the commit details for the selected commit.

Git History:

Staged	Status	Path
		1

RStudio: Review Changes

Changes	History	master	(all commits)
Subject	Author	Date	SHA
I did something new!	CaroZygar <caro_zyg@gmx.de>	2021-07-21	2641cf93
yay, my first commit! :-)	CaroZygar <caro_zyg@gmx.de>	2021-07-21	d39794b1

Commits 1-2 of 2

SHA	d39794b1b5a8075569ae2a7c0eed3725cc651aee
Author	CaroZygar <caro_zyg@gmx.de>
Date	2021-07-21 15:05
Subject	yay, my first commit! :-)

0-preprocessing.R

```
1 ## =====  
2 ## generate data  
3 ## =====  
4 |  
5 a <- c(1:5)
```

Im R Terminal:

4 `git checkout [SHA-Key]`

Der Stand des commits wird wiederhergestellt und kann kopiert (und wieder committet) werden

Live Demo – how I use git for papers

RStudio: Review Changes

Changes History master (all commits) Search Pull

Subject	Author	Date (UTC)	SHA
HEAD -> refs/heads/master submitted revision	CaroZygar <caro_zyg@gr>	2022-08-23	917d8844
Collabra submitted files	CaroZygar <caro_zyg@gr>	2022-04-14	7aa835a1
JESP submitted files	CaroZygar <caro_zyg@gr>	2022-04-09	c48ce42a
JEP Applied submitted files	CaroZygar <caro_zyg@gr>	2022-03-30	60432010
JEP General submitted files	CaroZygar <caro_zyg@gr>	2022-03-21	b6dd3413
moved new plot to supplemental, reinserted original	CaroZygar <caro_zyg@gr>	2022-03-14	e6a4ebb1
after first discussion with Felix, with new plots, first version	CaroZygar <caro_zyg@gr>	2022-03-11	b8cae4ea
after Felix comments, but before discussion	CaroZygar <caro_zyg@gr>	2022-03-07	780ba462
first complete version, before felix comments	CaroZygar <caro_zyg@gr>	2022-03-02	0c400ec4
Power Analysis + Results + Started Discussion	CaroZygar <caro_zyg@gr>	2022-01-26	d38869fa
added measures and sample description	CaroZygar <caro_zyg@gr>	2022-01-12	4684ff37
added procedure	CaroZygar <caro_zyg@gr>	2021-12-08	5d7400fc
after trying out trackdown and incorporating felix comments	CaroZygar <caro_zyg@gr>	2021-12-08	23a32e91

manuscript/ClosenessInt.Rmd

@@ -8,56 +8,56 @@ author:

8	8	corresponding : yes # Define only one corresponding author
9	9	address : "Leopoldstr. 13, 80802 München, Germany"
10	10	email : "caroline.zygar@psy.lmu.de"
11		role: # Contributorship roles (e.g., CRediT, https://creativecommons.org/licenses/by/4.0/)
12		- Conceptualization
13		- Data curation
14		- Formal analysis
15		- Funding acquisition
16		- Investigation
17		- Methodology
18		- Project administration
19		- Resources
20		- Supervision
21		- Visualization
22		- Writing - Original draft preparation
23		- Writing - Review & editing
	11	# role: # Contributorship roles (e.g., CRediT, https://creativecommons.org/licenses/by/4.0/)
	12	# - Conceptualization
	13	# - Data curation
	14	# - Formal analysis
	15	# - Funding acquisition
	16	# - Investigation
	17	# - Methodology

Hands-On

Tasks:

1. Get Git going in RStudio (incl. SSH and setting up name and email address)
2. Start a local project in R, give it a meaningful name, and activate git
3. Create two folders “raw_data” and “preprocessing” and make your first commit
4. Add the folder „raw_data“ to the .gitignore
5. Start a new r-script in the „preprocessing“-folder, load the packages „dplyr“ and „psych“ and save the script.
6. Make a new commit and describe it meaningfully
7. Change the r-script by replacing „dplyr“ with the „tidyverse“, by no longer loading „psych“ and by adding a line which says „a <- 4“
8. Make a new commit, and have a look at the diff
9. Revert the r-script back to the second commit

Git Ressources / Tipps & Tricks

- * <https://docs.google.com/document/d/1WvApy4ayQcZaLRpD6bvAqhWncUaPmmRimT016-PrLBk/edit>
- * <https://swcarpentry.github.io/git-novice/>
- * <https://happygitwithr.com/>
- * <https://guides.github.com/activities/hello-world/>
- * <https://ohshitgit.com/>
- * <https://twitter.com/aosmith16/status/1382424048857014272?s=09>

Rmarkdown + Git:

- * <https://resulimit.com/blog/rmd-workshop/>
- * https://twitter.com/ed_hagen/status/1381746314019414020?s=09