



Introduction to Computational Social Science

(Week 3: Open Science and Reproducibility)

Aliakbar Akbaritabar¹

¹Max Planck Institute for Demographic Research, Rostock, Germany;
akbaritabar@demogr.mpg.de

Intro to CSS course outline and syllabus

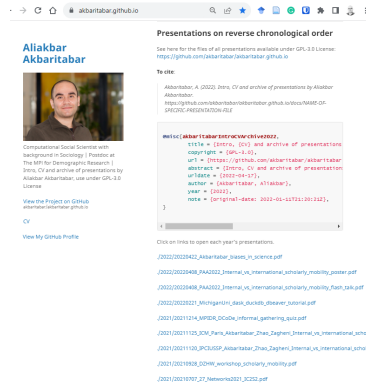


- Week 1 • Introduction ... What is "Computational Social Science"? Inductive and deductive research; Big data revolution
- Week 2 • Digital Trace Data ... Observational data; Available vs. designed data; APIs and web scrapping; Representativeness
- Week 3 • Open Science and Reproducibility ... Reproducibility crisis; Pre-registration; Version control, Git and GitHub
- Week 4 • Mobility and Migration ... Computational approaches to migration research
- Week 5 • Science of Science ... Robert K. Merton; Sociology of scientific knowledge vs Bibliometrics/ Scientometrics
- Week 6 • Network Analysis ... Tie formation mechanisms in social networks; Violence of independence of observations
- Week 7 • Ethics in Computational Social Science ... Informed consent; Personal data; GDPR
- Week 8 • Mid-semester evaluation ... Short essay and multiple choice questions
- Week 9 • Social Simulation ... Agent-based modelling for social scientists; Micro-Macro link and Coleman's boat
- Week 10 • Text as Data ... Natural Language Processing; Topic modelling; Structural Topic Models
- Week 11 • Machine Learning ... Supervised and unsupervised use of observational data; Feature learning
- Week 12 • Other CSS Skills ... Parallelization; Functional vs Object-Oriented; Graph database; DuckDB; SnakeMake Workflow
- Week 13 • Limitations of Computational Social Science ... Pitfalls of digital trace data and computational approaches; Representativeness
- Week 14 • Conclusions ... Thick vs Big data, Survey experiments; Linked data; Future of CSS
- Week 15 • Final semester evaluation + Student presentations ... Short essay and multiple choice questions OR presentation

Who am I?



- ▶ Computational Social Scientist with a background in sociology.
- ▶ Interest in science of science, scholarly migration, social and collaboration networks.
- ▶ Expertise in network analysis and modeling for large-scale and complex social media, bibliometric and textual data.
- ▶ Thanks, Sha Jiang, for generously offering to help today with the Git installation step



Practicing open science since 2011:
akbaritabar.github.io

Housekeeping rules



I will not monitor Zoom chat (today's complex setup with two in-person rooms + online + hands-on).

Instead, please use the Q&A (link on the next slide) throughout the course to post your questions.

Check if your question has already been asked by someone else, and upvote it.

Remember to mention your Operating System (Windows/Mac/etc) and the error texts.

The screenshot shows the GitHub Open Science Beg... Q&A page. The header is blue with the text "Git, GitHub Open Science Beg..." and "Q&A". Below the header is a search bar with the placeholder text "Type your question". The main content area has two tabs: "Popular" (selected) and "Recent". There are 3 questions listed. The first question is by Aliakbar Akbaritabar, posted 1 week ago, and contains links to Git downloads for Windows, Mac, and Linux. The second question is also by Aliakbar Akbaritabar, posted 1 week ago, and describes an error while installing Git on Windows OS. The third question is also by Aliakbar Akbaritabar, posted 1 week ago, and describes an error while installing Git on Mac OS. Each question has a "5" or "4" upvotes and a "1" downvote icon.

Git, GitHub Open Science Beg... Q&A

Type your question

Popular Recent 3 questions

AA Aliakbar Akbaritabar 1 week ago 5

LINKS: Windows (<https://git-scm.com/downloads/win>), Mac (<https://git-scm.com/downloads/mac>) Linux (<https://git-scm.com/downloads/linux>) (edited)

AA Aliakbar Akbaritabar 1 week ago 4

Error while installing Git on Windows OS. "ERROR-TEXT" (edited)

AA Aliakbar Akbaritabar 1 week ago 4

Error while installing Git on Mac OS. "ERROR-TEXT"

To post your questions in Q&A, share error messages, and answer polls + quiz:

Scan the QR code on the right ⇒

Or visit www.slido.com and enter the number: “1011941”.

Or visit: <https://app.sli.do/event/thfeKFo975LGRnm1pQpvHR>.



Why use plain text and version control?



What is wrong with MS Word and binary files, review tab, and tracked changes?

Improve it using name initials and dates

Or use Google Docs or something with history.

You should not be versioning files. Git can do that.

You should focus on the content!

A screenshot of a file explorer window showing a list of files. The files are named with a date, a title, and a version identifier. The file '20241024_Manuscript_mobility_of_talents_alternative_results-lb-aa.docx' is highlighted in blue.

20240220_Manuscript_mobility_of_talents_LB.docx
20240303_Manuscript_mobility_of_talents_LB_AA.docx
20240401_Manuscript_mobility_of_talents_2.docx
20240401_Manuscript_mobility_of_talents_LB_AA_rh_AA.docx
20240408_Manuscript_mobility_of_talents_CHANGES_ACCEPTED_journal_submis...
20240511_Manuscript_mobility_of_talents_CHANGES_ACCEPTED_journal_submis...
20240515_Manuscript_mobility_of_talents.docx
20240521_Manuscript_mobility_of_talents_lb_aa_rh_aa_PREVIOUS_VERSION.docx
20240811_Manuscript_mobility_of_talents_REVISIED.docx
20240811_Manuscript_mobility_of_talents_REVISIED_lb_aa.docx
20241024_Manuscript_mobility_of_talents_alternative_results.docx
20241024_Manuscript_mobility_of_talents_alternative_results-lb-aa.docx
20241031_Manuscript_mobility_of_talents_alternative_results-lb-aa-rh-aa.docx
20241031_Manuscript_mobility_of_talents_alternative_results-lb-aa-rh-aa-lb.docx
20241105_Manuscript_mobility_of_talents_alternative_results.docx

A short guide to writing and version control with R markdown, git and GitHub!

Near one year of my data analysis and writing projects with R markdown and Git

Ali

01-11-2017

This post might not include much of R code or data analysis, but believe me, behind the scenes, it can be one of the most useful things I am going to write!

As a social scientist and sociologist by near 13 years of training, I have mostly been in contact with people who think social science gang are not **Geek** or **Nerd** material! Why is that so? Why there is a common sense that as a social scientist as long as you know how to turn on a computer and use Microsoft Word and Excel along with one or two statistical analysis software packages, e.g., SPSS, Lisrel, STATA, you are good to go! And in case you happen to be able to do some more stuff with computers you can be gazed at or called with names, like geek!

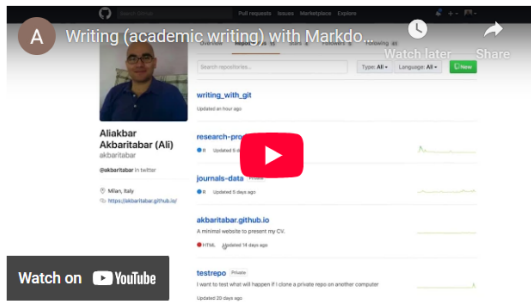
Videos and blog post:

https://akbaritabar.github.io/CV_MD/git_github_for_academic_writing.html.

Preaching this since 2017



OK! If the above description was enough, watch this **20 min** video on simple steps where to start with these tools to live an easier academic life!



Briefly, for me “Git” is like a huge “Ctrl + Z” that allows me to go back in time to whatever extent that I want and correct my mistakes, isn’t that amazing?

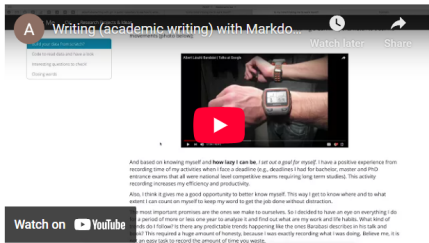
Videos and blog post:

https://akbaritabar.github.io/CV_MD/git_github_for_academic_writing.html.

Preaching this since 2017



- 1st video topics reviewed: **1:05**
- 2nd vide topics: **2:14**
 - Clone a git repository: **4:00**
 - A real example of R Markdown with R code chunks: **10:42**
 - How to go from RMD report to presentation file: **20:28**
 - Using git branches: **28:24**
 - Using git stashing: **44:47**
 - Review and links to further learning: **56:40**



Links mentioned in 2nd video (advanced topics)

- [Learn R markdown formats](#)
- [Presentation file of topics in 2nd video](#)
- [Github repository of 2nd video](#)

Videos and blog post:

https://akbaritabar.github.io/CV_MD/git_github_for_academic_writing.html.

What is Git?

DropBox, Google Drive, NextCloud, etc.

Usually, no history of files.



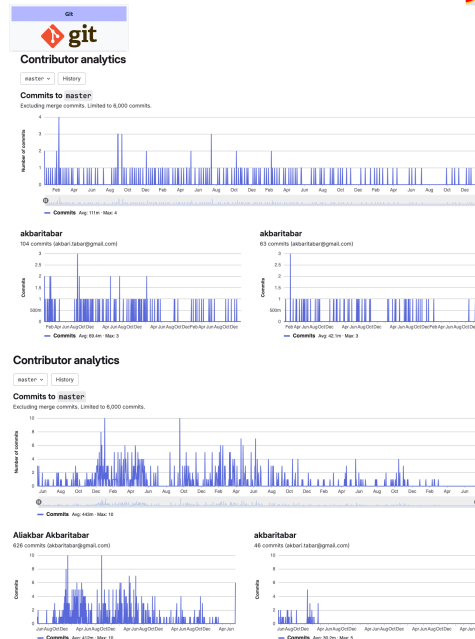
What is Git?

DropBox, Google Drive, NextCloud, etc.

Usually, no history of files.

Git is a tool for keeping a history of changes in your files and project.

Think of it as an endless "Ctrl + Z."



What is Git?

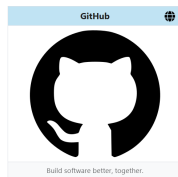
DropBox, Google Drive, NextCloud, etc.

Usually, no history of files.

Git is a tool for keeping a history of changes in your files and project.

Think of it as an endless "Ctrl + Z."

All is backed up in Git if you stage and commit it.



What is Git?

DropBox, Google Drive, NextCloud, etc.

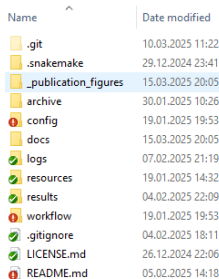
Usually, no history of files.

Git is a tool for keeping a history of changes in your files and project.

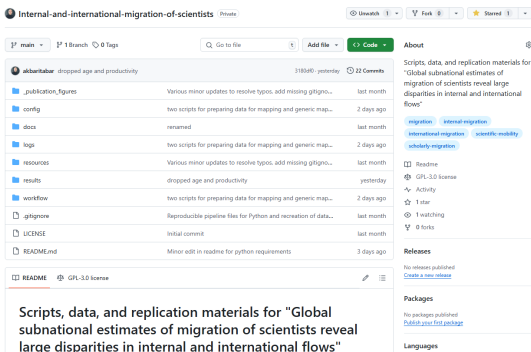
Think of it as an endless "Ctrl + Z."

All is backed up in Git if you stage and commit it.

Web hosting of Git in GitHub, GitLab, BitBucket, etc., like file sharing with history!



Name	Date modified
.git	10.03.2025 11:22
.snakemake	29.12.2024 23:41
_publication_figures	15.03.2025 20:05
archive	30.01.2025 10:26
config	19.01.2025 19:53
docs	15.03.2025 20:05
logs	07.02.2025 21:19
resources	19.01.2025 14:32
results	04.02.2025 22:09
workflow	19.01.2025 19:53
.gitignore	04.02.2025 18:11
LICENSE.md	26.12.2024 22:06
README.md	05.02.2025 14:18



Internal-and-international-migration-of-scientists Private

Unwatch 1 Fork 0 Starred 3

main 1 Branch 0 Tags

Go to file Add file <> Code About

akbaritabar · dropped age and productivity 318940 · yesterday 22 Commits

_publication_figures	Various minor updates to resolve typos, add missing gitignore...	last month
config	two scripts for preparing data for mapping and generic map...	2 days ago
docs	renamed	last month
logs	two scripts for preparing data for mapping and generic map...	2 days ago
resources	Various minor updates to resolve typos, add missing gitignore...	last month
results	dropped age and productivity	yesterday
workflow	two scripts for preparing data for mapping and generic map...	2 days ago
.gitignore	Reproducible pipeline files for Python and recreation of data...	last month
LICENSE	Initial commit	last month
README.md	Minor edit in readme for python requirements	3 days ago

README GPL-3.0 license

Scripts, data, and replication materials for "Global subnational estimates of migration of scientists reveal large disparities in internal and international flows"

No releases published
Create a new release

Packages
No packages published
Publish your first package

Languages


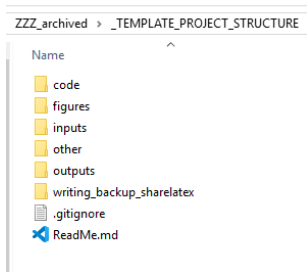


A clear and consistent folder structure

Create a clear structure and stick with it.

This will be (through time) clear to you and your collaborators.

A habit of using numbered scripts or workflow managers.



Name	Date modified
.ipynb_checkpoints	14.04.2024 18:17
.tmp	20.09.2021 16:41
dask-worker-space	23.09.2021 08:20
method_prototype	04.06.2021 14:22
.gitignore	03.06.2023 21:17
0_build_corpus.sql	02.02.2021 09:43
1_ROR_API_organization_di...	01.06.2022 10:56
2_process_orcid_2020.py	01.06.2022 10:56
3_join_SCP_authorship_wit...	31.05.2024 08:49
4_mobility_analysis_using_...	01.06.2022 10:56
5_visualize_mobility_flows.r	03.08.2022 16:27
6_mobility_analysis_using_...	01.06.2022 10:56
7_network_based.r	30.11.2021 17:50
8_functions_to_process_sc...	08.11.2021 16:10
9_figures_for_presentation.R	08.11.2021 16:10
10_scp_using_arangodb.aql	24.01.2022 12:01
11_mobility_analysis_using...	27.06.2023 22:16
12_clara_mobility_figure_fo...	15.09.2022 14:00
13_prepare_migration_data...	14.06.2023 16:27
14_replication_of_figure_1.py	18.02.2024 13:04
15_replication_of_figure_2.py	05.06.2023 17:49
16_replication_of_figure_2_...	03.06.2023 21:19
17_typologies_of_INT_IN_...	18.08.2024 13:39
18_typologies_of_INT_IN_...	24.06.2024 17:09
19_prepare_migration_data...	31.05.2024 09:43
20_replication_of_figure_1_...	18.12.2024 18:16
21_replication_of_figure_1_...	02.06.2024 22:36
22_answering_reviewer_co...	25.12.2024 21:42
23_multiple_affiliations_per...	23.12.2024 17:12
24_multiple_affiliations_per...	24.12.2024 17:28

Hands-on part

To post your questions in Q&A, share error messages, and answer polls + quiz:

Scan the QR code on the right ⇒

Or visit www.slido.com and enter the number: “1011941”.

Or visit: <https://app.sli.do/event/thfeKFo975LGRnm1pQpvHR>.



See course instructions; Links also in Q&A platform

Install git (No admin rights needed)

OR, download, unzip, and use git portable for Windows

Check by writing "git - -version" (two dashes, without space) and pressing enter.

GitHub (Beginner): Using Git and Github for Open Science & connection with OS repositories for sharing data and codes (OSF and Zenodo)

(Aliakbar Akbaritabar)

Prior knowledge required: None (just being enthusiastic about doing research suffices!)

Technical equipment required: Personal laptop with permission to install software (if you cannot install software, you can follow along on the projector)

ToDo before the workshop: Please download and install Git for Windows (<https://git-scm.com/downloads/win>), Mac (<https://git-scm.com/downloads/mac>) or Linux (<https://git-scm.com/downloads/linux>). Please note: no GUI installation is needed. Only follow the instructions in one of those links, depending on your operating system, and install the Git software. If you were asked for Windows installation to add "context menu shortcut" and "add Git to PATH" please select "yes" for both (selected by default).

Description:

This tutorial assumes no prior knowledge about version control and using Git, GitHub, and similar tools for research. At the end of the tutorial, participants will know what they will gain by using version control and how they can use different version control tools in writing scientific text. The tutorial covers academic writing and analysis scripts, how version control helps, and the benefits of switching to Latex, RMarkdown, and other plain text formats for academic writing. Participants will learn to use Git and GitHub in Terminal (CLI) and popular IDEs such as RStudio and Visual Studio Code. Participants will learn how to deposit replication materials on Open Science Framework (OSF) and Zenodo and share a view-only link for peer review.

- ▶ Command line is your friend (for me, I make less mistakes than GUIs)
- ▶ "pwd" (print working directory, only for Mac/Linux/Bash)
- ▶ "dir" (directory, for Windows/Bash)
- ▶ "ls" (list, only for Mac/Linux/Bash)
- ▶ "cd" (change directory)
- ▶ Or easier "open git bash here" in right-click context menu
- ▶ "- -argument" and "-arg"
- ▶ help COMMAND-NAME
- ▶ "git - -version"

Essential Git Commands

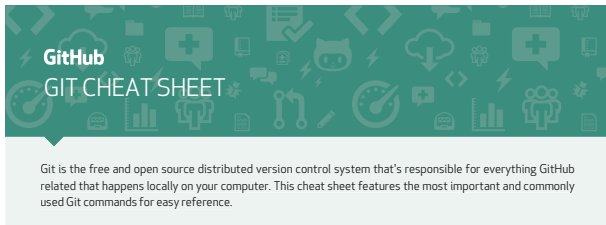
Get the git cheat-sheet



GitHub education:

[https:](https://education.github.com/git-cheat-sheet-education.pdf)

[//education.github.com/
git-cheat-sheet-education.
pdf.](https://education.github.com/git-cheat-sheet-education.pdf)



INSTALLATION & GUIs

With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

GitHub for Windows

<https://windows.github.com>

GitHub for Mac

<https://mac.github.com>

For Linux and Solaris platforms, the latest release is available on the official Git web site.

Git for All Platforms

<http://git-scm.com>

SETUP

Configuring user information used across all local repositories

```
git config --global user.name "[firstname lastname]"
```

set a name that is identifiable for credit when review version history

```
git config --global user.email "[valid-email]"
```

set an email address that will be associated with each history marker

```
git config --global color.ui auto
```

set automatic command line coloring for Git for easy reviewing

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

git status

show modified files in working directory, staged for your next commit

git add [file]

add a file as it looks now to your next commit (stage)

git reset [file]

unstage a file while retaining the changes in working directory

git diff

diff of what is changed but not staged

git diff --staged

diff of what is staged but not yet committed

git commit -m "[descriptive message]"

commit your staged content as a new commit snapshot

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

git branch

list your branches, a * will appear next to the currently active branch

git branch [branch-name]

```
git config - -list
```

```
git config - -global user.name "[firstname lastname]"
```

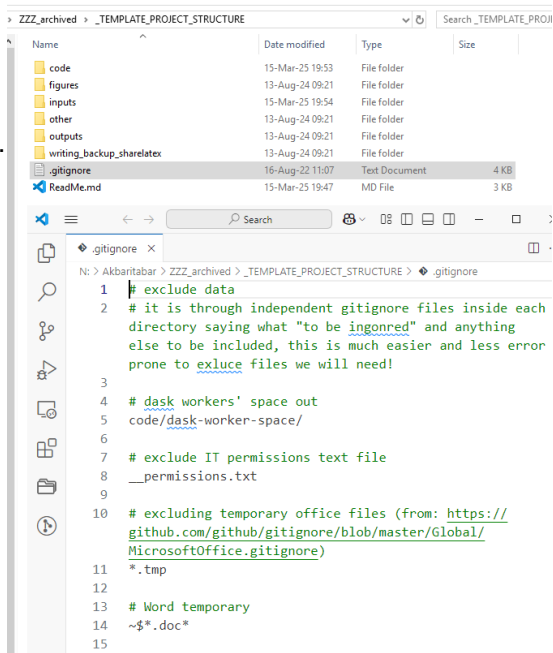
```
git config - -global user.email "[valid-email]"
```

Initiates a new repository in the local directory and tells git to start watching this directory.

Git .ignore file

Defines things to be ignored.
Can have one in the
template folder structure
and copy/paste and reuse
and include a customized
one in sub-folder

```
"logs/" ; "*.notes" ;  
"pattern*/"
```



Git .ignore file



Defines things to be ignored.

Can have one in the
template folder structure
and copy/paste and reuse
and include a customized
one in sub-folder

```
"logs/" ; "*.notes" ;  
"pattern*/"
```

Z_archived > _TEMPLATE_PROJECT_STRUCTURE > code				Search code
Name	Date modified	Type	Size	
.gitignore	15-Mar-25 19:54	Text Document	1 KB	

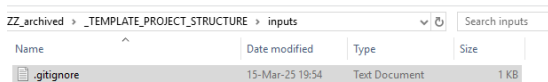
```
.gitignore  
1 # Ignore NOTHING in this directory, INCLUDE ALL  
2 !*  
3 |
```

Codes folder, include everything.

Git .ignore file

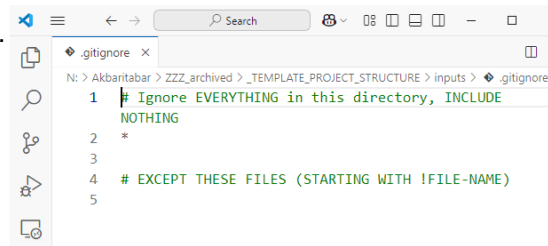
Defines things to be ignored.
Can have one in the
template folder structure
and copy/paste and reuse
and include a customized
one in sub-folder

```
"logs/" ; "*.notes" ;  
"pattern*/"
```



ZZ_archived > _TEMPLATE_PROJECT_STRUCTURE > inputs

Name	Date modified	Type	Size
.gitignore	15-Mar-25 19:54	Text Document	1 KB



```
1 # Ignore EVERYTHING in this directory, INCLUDE  
   NOTHING  
2 *  
3  
4 # EXCEPT THESE FILES (STARTING WITH !FILE-NAME)  
5
```

Inputs folder, exclude everything EXCEPT some.

Reports the status of the working directory/files.

Adds the selected file (or pattern of files) to the staging area to commit to history (git, watch these files).

```
git commit -m "[descriptive message]"
```



Writes the staged files to history.

Shows the history.

Similar to MS Word's review tab, it shows word-level changes in the staging area (not yet committed; for committed, add "- -cached").

For experiments that might be thrown away. Allows branching off of the main trunk of history (that can later be merged back into main history, more advanced for now).

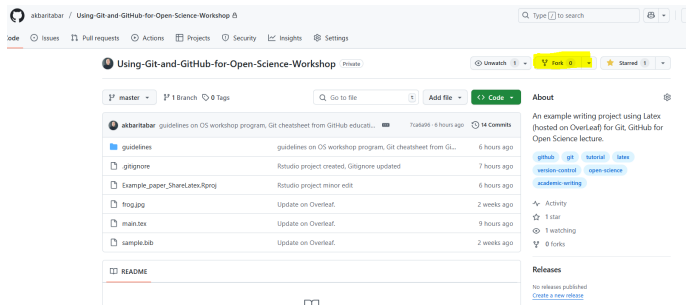
Fork a repository on GitHub



GitHub, bring a copy of this repository and its history, and store it "on my GitHub profile."

BTW, keep it linked to the original repository!

GitHub repository:
tinyurl.com/osgithub25



git clone [url]



Git, bring a copy of this repository and its history, and store it here "on my computer".

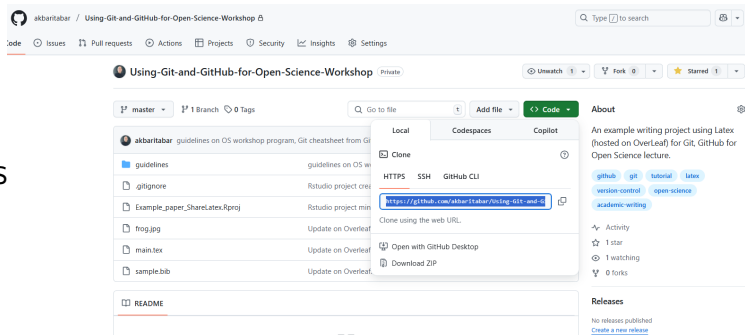
The screenshot shows a GitHub repository page. At the top, the repository name is 'Using-Git-and-GitHub-for-Open-Science-Workshop' by user 'akbaritabar'. Below the repository name, there are tabs for 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The main content area shows a list of files and folders: 'guidelines', '.gitignore', 'Example_paper_ShareLatex.Rproj', 'frog.jpg', 'main.tex', and 'sample.bib'. Each file has a description of its last update and the time since the last commit. For example, 'guidelines' was updated 6 hours ago. To the right of the file list, there is an 'About' section with a description: 'An example writing project using LaTeX (hosted on Overleaf) for Git, GitHub for Open Science lecture.' Below the 'About' section, there are links to 'github', 'git', 'tutorial', 'latex', 'version-control', 'open-science', and 'academic-writing'. At the bottom of the repository page, there is a 'README' section.

Link: tinyurl.com/osgithub25

git clone [url]



Git, bring a copy of this repository and its history, and store it here "on my computer".



Link: tinyurl.com/osgithub25

Git, tell me where I have cloned this repository from.

Git, I have made some changes in files locally, I added and committed them. My git status is clean. Send these changes to the remote, where I cloned it from (if I have write/push rights).

Git, I want to work on this local repository, but before that, I want to have the most updated version of it, in case a collaborator has made changes, "bring (pull) those changes here".

Git and plain text for writing/coding

Writing using plain text



RMarkdown (R code in the same file)

Issues with collaborators' wishes to compile

Latex, OverLeaf, and ShareLatex

Name	Date modified
.git	22-Jan-25 16:16
.Rproj.user	22-Jan-25 16:16
.vscode	07-Jun-21 08:57
code	23-Dec-24 16:52
data	17-Feb-25 14:07
libs	05-May-21 18:55
method_prototype_presentation_cache	18-Apr-21 20:15
other	03-Oct-23 15:26
reports	20-Dec-24 14:09
Scopus_author_export_analysis_cache	05-Apr-21 22:41
Scopus_author_export_analysis_files	05-Apr-21 22:41
.gitignore	22-Sep-21 11:13
.Rhistory	22-Jan-25 16:17
apa.csl	28-Jan-21 22:13
eu_mobility.Rproj	22-Jan-25 16:15
FullBibliographyList.bib	30-Mar-21 10:25
method_prototype_presentation.html	06-May-21 09:49
method_prototype_presentation.pdf	06-May-21 09:54
method_prototype_presentation.Rmd	06-May-21 09:49
README.md	20-Nov-21 11:46
Scopus_author_export_analysis.log	05-Apr-21 22:42
Scopus_author_export_analysis.pdf	05-Apr-21 22:42
Scopus_author_export_analysis.Rmd	28-Apr-21 11:36
stylereference.docx	10-Sep-20 10:25

Writing using plain text



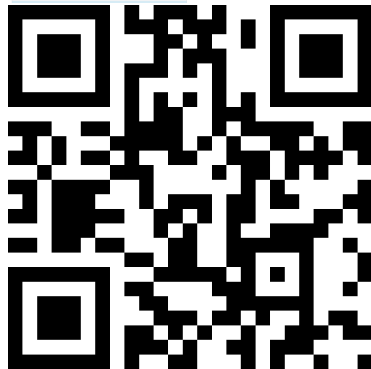
RMarkdown (R code in the same file)

Issues with collaborators' wishes to compile

Latex, OverLeaf, and ShareLatex

Our example repository + anonymous editing)

tinyurl.com/latexex25



Backup of ShareLatex/OverLeaf on private GitHub repository (credit: <https://jigarius.com/blog/multiple-git-remote-repositories>)

1. `git remote -v` (to use later)
2. `git remote set-url - --add - --push origin https://github.com/REPO-URL-HERE.git`
3. (add sharelatex push URL again) `git remote set-url - --add - --push origin https://git@sharelatex.gwdg.de/git/code-to-take`
4. `git remote -v` (to check all is well? should show 3 URLs)
5. `git pull - --all`
6. `git push - --all` (wait, why did it give 2 outputs?! ;))
7. Everything up-to-date
8. Everything up-to-date

Git in popular IDEs

Note: choose your IDE as if you are choosing a lifetime companion

My advice: Use Git and GitHub in Terminal (CLI)

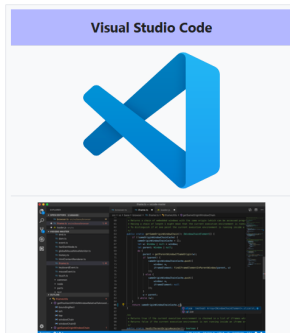
For RStudio:

1. Add git executable path in global options under version control
2. Create a project
3. Open terminal
4. Or, open git tab/pane



Highly recommended IDE with a lot of flexibility and multilingual!

1. Install VS Code
2. It should already be added to the user's PATH
3. Open Git Bash
4. "cd" to the project folder
5. Or right-click and select "open git bash here"
6. Write "code ." and press enter
7. The VS Code terminal will already inherit git
8. Check it with "git --version"
9. It works even with the portable git
10. Check out extensions such as "Git Graph"



A word on Jupyter project,
Jupyter notebooks, Jupyter
Lab, version control, and
using them with plain scripts

Reproducibility, OSF and Zenodo

Deposit replication materials on
Open Science Framework (OSF)

Share a view-only link for peer
review (Credit: `https://help.osf.io/article/201-create-a-view-only-link-for-a-project`).

How to create a release in GitHub
and publish it on Zenodo.

Check also: `https://github.com/dgraziotin/disclose-data-dbr-first-then-opendata`

More?

Where to go next?



- ▶ <https://github.blog/developer-skills/programming-languages-and-frameworks/what-is-git-our-beginners-guide-to-version-control/>
- ▶ <https://skills.github.com/>
- ▶ <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

More advanced topics

- ▶ Usable for "academic writing using plain text"
- ▶ "Coding for analysis etc"
- ▶ "Workflow management using SnakeMake" + Targets in R
- ▶ GitHub pages, shell script to automate a personal website? paper introduction?
- ▶ Example: <https://akbaritabar.github.io> that uses a shell script and markdown plain text files
- ▶ GitHub CodeSpace (push "." on a repository)
- ▶ GitHub actions to automate recurring works
- ▶ Beamer and presentations in plain text and \LaTeX (even these slides are version-controlled)
- ▶ Latexdiff Perl script for tracked changes in the Revise and Resubmit stage of publication



The SnakeMake Report window displays a workflow rule named 'snakefile_map_figures.smk'. The rule's input is a file path, and its output is a file path. The rule is defined in a shell script. The dependency graph on the right shows the workflow's structure, with nodes representing targets and arrows indicating dependencies.

```
SnakeMake Report
Rule: duckdb_multiple_affiliations_per_author_career_year

INPUT
G:\AI\wp_rp_2020\wp_rp_2020_authorshp_srcs\srcs\BGR.parquet
G:\AI\GWAResources\migration\internal_mobility_province.parquet
G:\AI\GWAResources\migration\international_mobility_province.parquet

OUTPUT
G:\AI\GWAResults\figures\author_level_mvmt_affiliations.png
G:\AI\GWAResults\tables\descriptive_table_multiple_affiliation_types\descriptive_table_multiple_affiliation_types.parquet
G:\AI\GWAResults\tables\descriptive_table_multiple_affiliation_dominant_type_per_author.txt

CODE
1 (python workflow/scripts/hv_duckdb_initialize_affiliations_per_author_career_year

workflow > rules > snakefile_map_figures.smk
You, last month | 1 author (You)
1 ## File name to use in search: snakefile_map_figures.smk ##
2
3 # =====
4 ## NMR and MEI MAPS for all time_span combinations ##
5 # =====
6
7 rule plot_map_NMR_AND_MEI_MAPS:
8     input:
9         rules.prepare_data_for_mapping.output
10    output:
11        NMR_AND_MEI_MAPS
12    log:
13        NMR_AND_MEI_MAPS_LOG
14    shell:
15        "(python workflow/scripts/generic_src_mapping_figures.
16        py --input {input} --MEASURE_MAPPED {wildcards.
17        measure_mapped} --MIGRATION_SYSTEM {wildcards.
18        migration_system} --GEO_REGION {wildcards.geo_region}
19        --TIME_SPAN {wildcards.time_span} --output {output})
20        2> {log}"
```

Quiz time!

To post your questions in Q&A, share error messages, and answer polls + quiz:

Scan the QR code on the right ⇒

Or visit www.slido.com and enter the number: “1011941”.

Or visit: <https://app.sli.do/event/thfeKFo975LGRnm1pQpvHR>.



Thanks for your attention!



Questions and comments are welcome!

Aliakbar Akbaritabar

Research Scientist

Max Planck Institute for Demographic Research
(MPIDR), Rostock, Germany.

Email: akbaritabar@demogr.mpg.de

LinkedIn/BlueSky/Twitter/Mastodon: @akbaritabar

- ▶ <https://de.m.wikipedia.org/wiki/Git>
- ▶ https://de.m.wikipedia.org/wiki/Datei:Dropbox_Logo_01.svg
- ▶ https://de.m.wikipedia.org/wiki/Google_Drive
- ▶ <https://de.m.wikipedia.org/wiki/Bitbucket>
- ▶ <https://de.m.wikipedia.org/wiki/GitLab>
- ▶ <https://de.m.wikipedia.org/wiki/GitHub>
- ▶ <https://de.m.wikipedia.org/wiki/RStudio>
- ▶ https://de.m.wikipedia.org/wiki/Visual_Studio_Code
- ▶ <https://de.m.wikipedia.org/wiki/Overleaf>