Introduction to Workflow and Git(Hub)

Best practice: Folder and file structure

1. separate folders for scripts, data, output, and reports
2. raw data files separate from processed data
3. clear and consistent names for script and output files
4. numbering, lowercase, and separate words with underscores or hyphens
5. if date necessary, usually in the end, sort by YYYYMMDD
6. multiple script files for different (sub) tasks (max 100 lines)

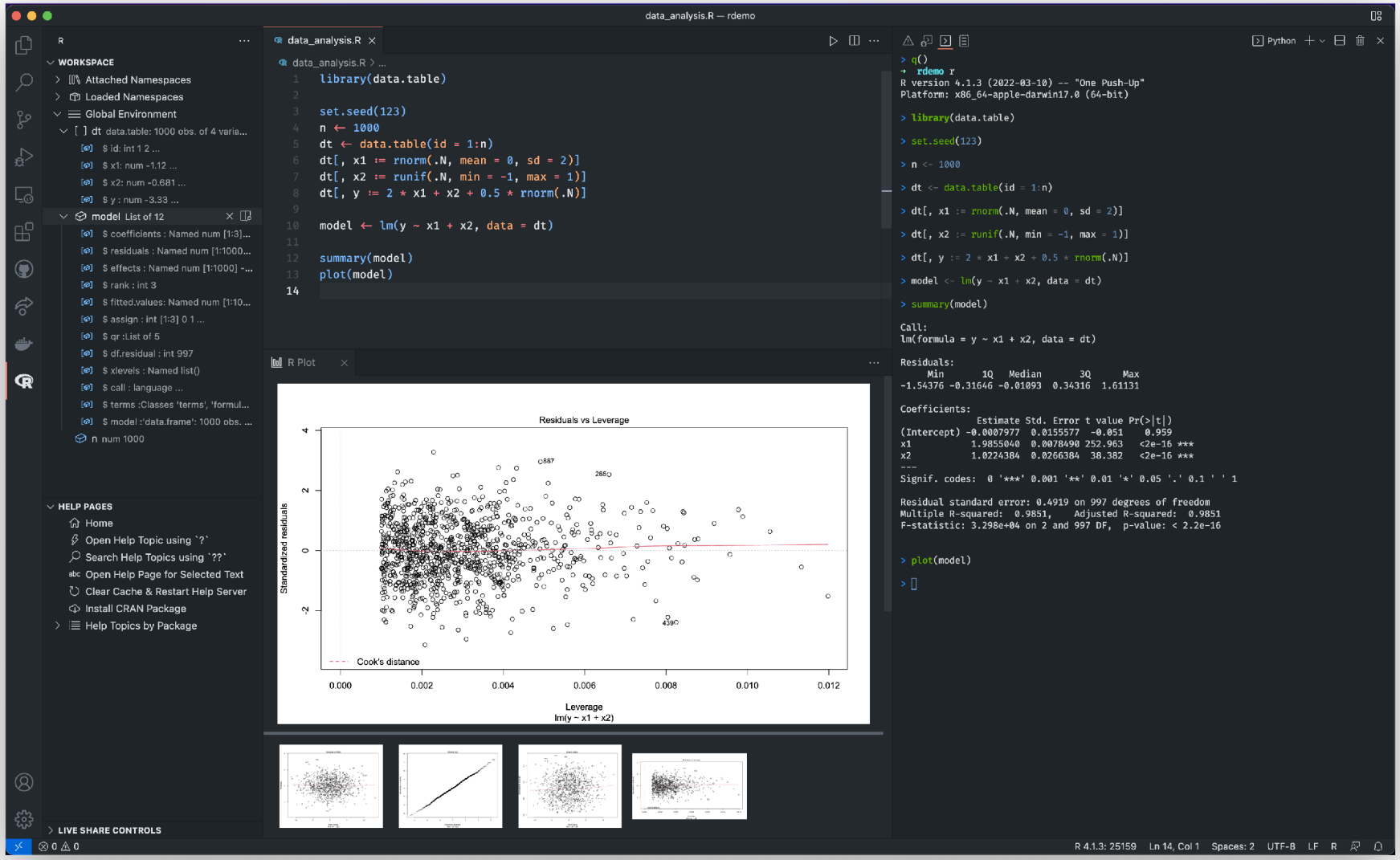
Best practice: Efficient R scripts

1. define libraries, default variables, source code at top of script
2. comment and structure sections (# ---- headline ----)
3. use pipe operator |> (magritter: %>%) for combining functions
4. use indentations and spaces for readability
5. DRY - use lists, lapply, vectorization, and functions
6. use relative paths for data and output
7. avoid hard coded subsetting and indexing

Best practice: Resources

- Best Coding Practices for R
- Structuring R projects
- R Best Practices
- Tips for organising your R code
- Nice R Code
- Repeating things: looping and the apply family

Code Editor: VSCode

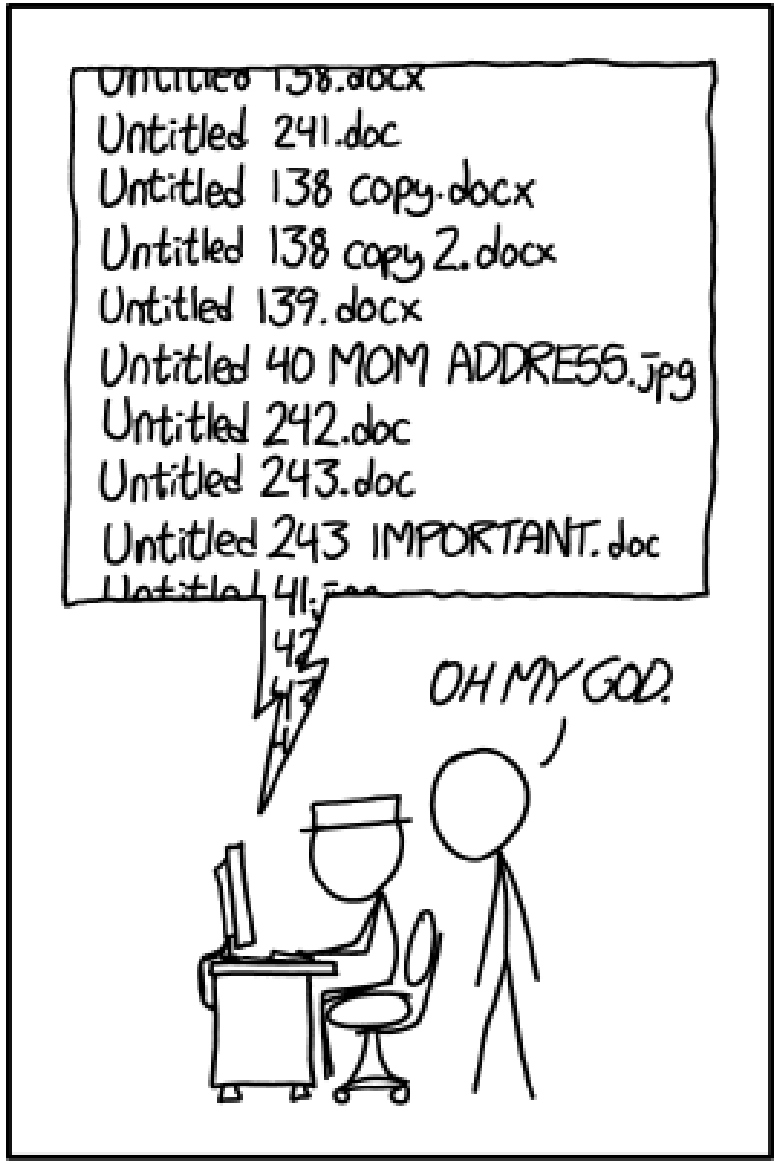


Code Editor: Benefits

- Swiss army knife of coding and file management
  - search (and replace) in whole project folder
  - side-by-side editor windows
  - better file and folder management
  - customizable (with extensions)
- multiple languages supported (e.g. R, Python, LaTeX, Markdown)
- Git(Hub): easy integration for better workflow
- with R:
  - run multiple R Sessions in parallel
  - scripts still editable if process busy

Code Editor: Resources

- https://code.visualstudio.com/docs/languages/r
- https://renkun.me/2019/12/11/writing-r-in-vscode-a-fresh-start/
- https://schiff.co.nz/blog/r-and-vscode/
- https://rolkra.github.io/R-VSCode/

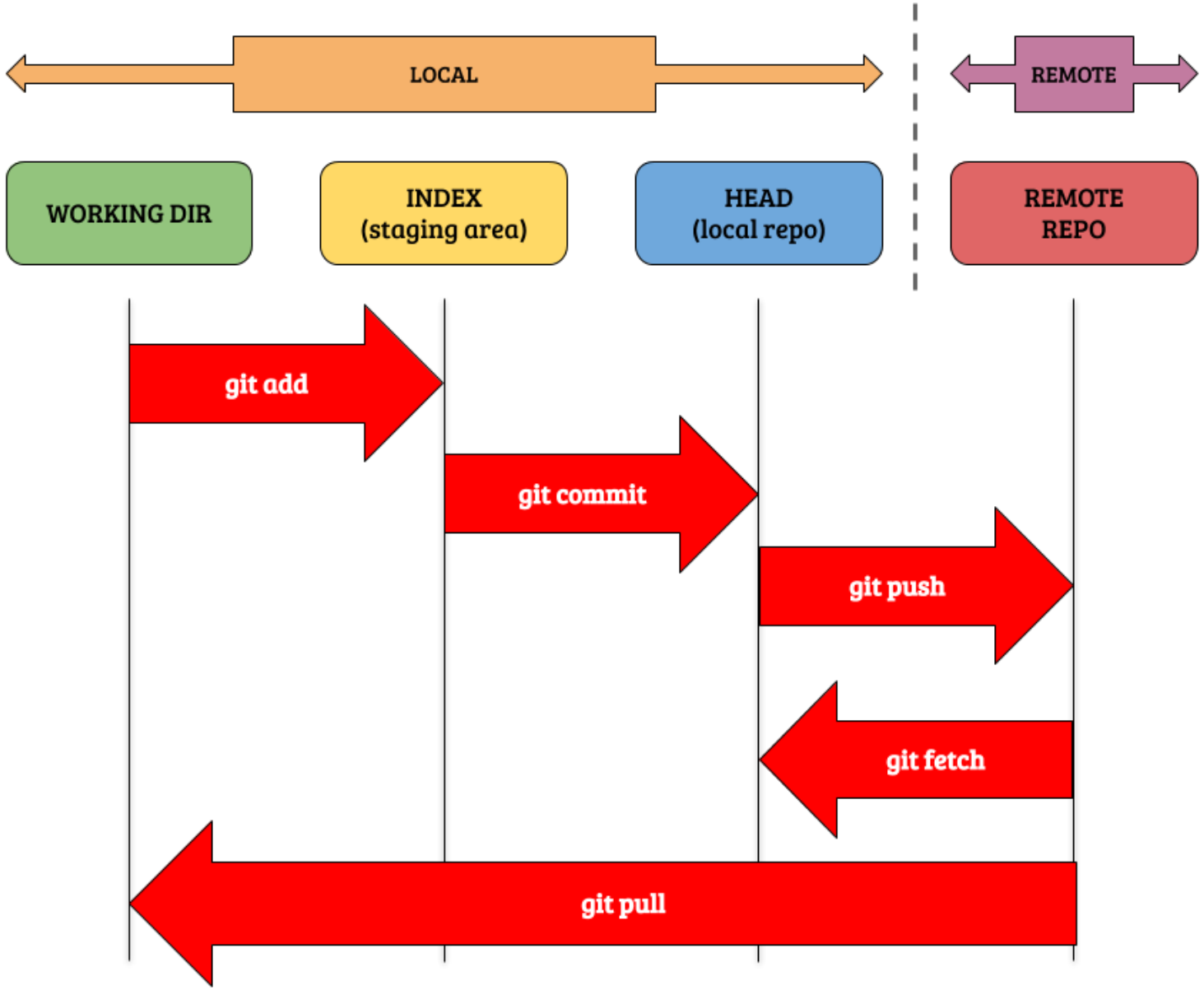


Git: Idea and concept

- distributed version control system
- track and document changes in code
- compare and find differences
- "time travel maschine"
- helps to imagen changes as smaller tasks
- share, publish, and collaborate on projects

Git: How to use it

	Ease of use	Efficiency	Nerdiness
CLI	● ○ ○ ○ ○	● ● ○ ○ ○	● ● ● ● ●
GUI	● ● ● ○ ○	● ● ● ○ ○	● ● ○ ○ ○
RStudio	● ● ● ● ○	● ● ● ○ ○	● ● ● ○ ○
VSCode	● ● ● ● ●	● ● ● ● ○	● ● ● ● ○


LOCAL
REMOTE
WORKING DIR
INDEX
(staging area)
HEAD
(local repo)
REMOTE
REPO
git add
git commit
git push
git fetch
git pull

Git: First and basic steps

```
$ git config --global user.name <your name>
$ git init <your repository name>
$ git status
```

```
$ git add <file-name-1> <file-name-2> OR --all
$ git commit -m “<commit-message>”
OR BOTH IN ONE
$ git commit -am “<commit-message>”
```

GitHub: Remote and cooperative workflow

```
$ git clone <git-repo-url>
```

```
$ git branch < branch-name>
$ git checkout <name-of-your-branch>
OR BOTH IN ONE
$ git checkout -b <name-of-your-branch>
```

```
$ git push -u (<remote> OR origin) <branch-name>
```

```
& git fetch
& git merge <branch-name>
OR BOTH IN ONE
$ git pull <remote> (<branch-name>)
```

GitHub: Credentials

- for GitHub remote commands: username and personal access token needed
- Settings → Developer Setting → Personal Access Tokens → Generate new token
- enter username and PAT everytime, or:
    i. `git config credential.helper store`
    ii. use VSCode

GitHub: .gitignore

Why: text file with folders and files (patterns) not to track

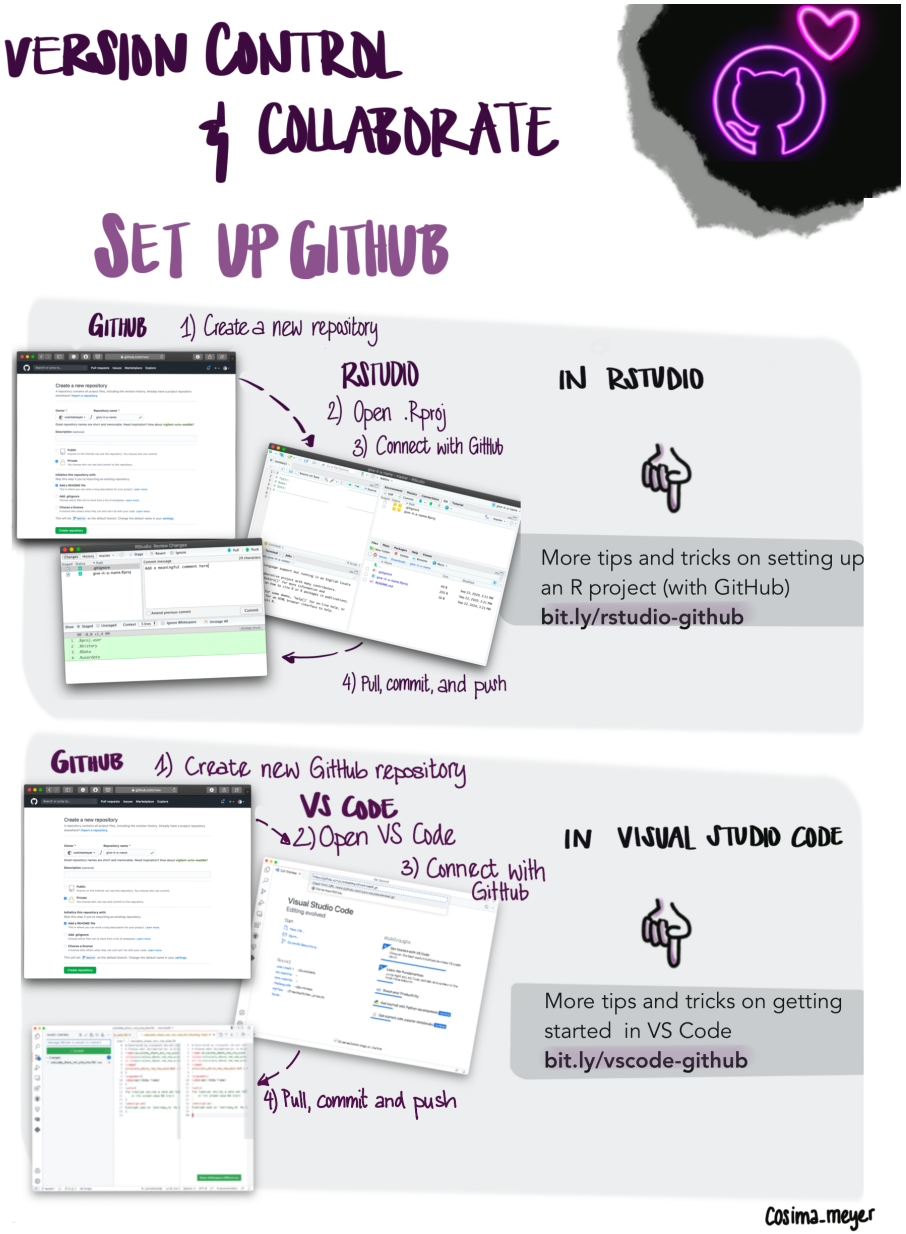
What: sensitive data; temp and old files; big data files; outputs
→ usually track just plain text files (e.g. R scripts, TeX source, etc.)

How: 2 approaches (helpful online tool creates content automatically)

```
/data/old
passworts.txt
*.doc
------OR------
/*
!.gitignore
!/scripts
```

Git: Ressources

- Intro to Git, for the Social Scientist
- Git for Social Scientists
- Git for Students in the Social Sciences
- GitHub - The Perks of Collaboration
- AI tool suggesting git command



GitHub: Course material

1. install git
2. create a GitHub account
3. tell user name to become a collaborator
4. clone our course material repository
5. add a personal folder and test file
6. push this changes to the remote repository
7. pull changes of the other participants