REPRODUCIBILITY IN SCIENCE

WHAT IT IS AND WHY TO CARE, WITH EXAMPLES FROM THE DATALAD WORLD

Adina Wagner
mas.to/@adswa
Cognitive and Affective Biopsychology,
Institute of Neuroscience and Medicine, Brain & Behavior (INM-7)
Research Center Jülich

Slides: DOI 10.5281/zenodo.19692938 (Scan the QR code)
files.inm7.de/adina/talks/html/hida2026.html

LOGISTICS

- **QR Code -** Crowdsourced notes, networking, & anonymous questions at hedgedoc.psychoinformatics.de/7X6uaPPAR2-wkskcP0W0-A#
- JupyterHub: jupyter.edu.datalad.org.
- Collaboration Hub: hub.edu.datalad.org.
- Didn't get a user name by email? Speak up!

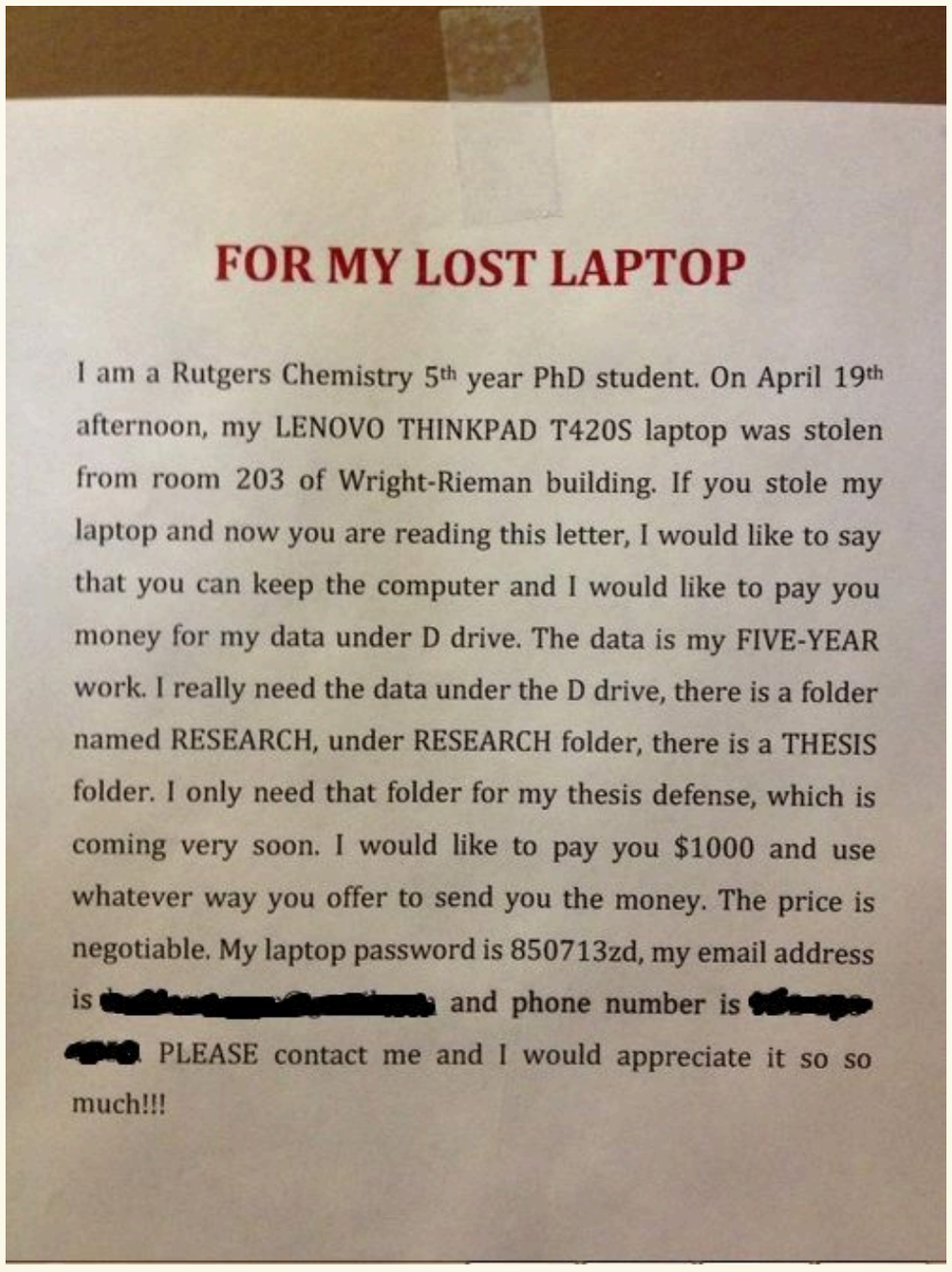
COMMON PROBLEMS IN SCIENCE

You write a paper about an algorithm, stay up late to generate good-looking figures, but you have to tweak parameters and display options to make it work AND look good. The next morning, you have no idea which parameters produced which figures, and which of the figures fits to what you report in the paper.

Image credit: Illustration adapted from Scriberia and The Turing Way

COMMON PROBLEMS IN SCIENCE

Your research project produces phenomenal results, but your laptop, the only place that stores the source code for the results, is stolen/breaks



FOR MY LOST LAPTOP

I am a Rutgers Chemistry 5th year PhD student. On April 19th afternoon, my LENOVO THINKPAD T420S laptop was stolen from room 203 of Wright-Rieman building. If you stole my laptop and now you are reading this letter, I would like to say that you can keep the computer and I would like to pay you money for my data under D drive. The data is my FIVE-YEAR work. I really need the data under the D drive, there is a folder named RESEARCH, under RESEARCH folder, there is a THESIS folder. I only need that folder for my thesis defense, which is coming very soon. I would like to pay you $1000 and use whatever way you offer to send you the money. The price is negotiable. My laptop password is 850713zd, my email address is [redacted] and phone number is [redacted]. PLEASE contact me and I would appreciate it so so much!!!

COMMON PROBLEMS IN SCIENCE

A graduate student approaches their supervisor, complaining that the supervisors research idea does not work. After weeks of discussion, it becomes apparent that oral communication doesn't suffice - the student can't sufficiently explain the environment (data, algorithms, ...) they constructed, and if the supervisor can't enter and use the students project there's no way to find a fix.

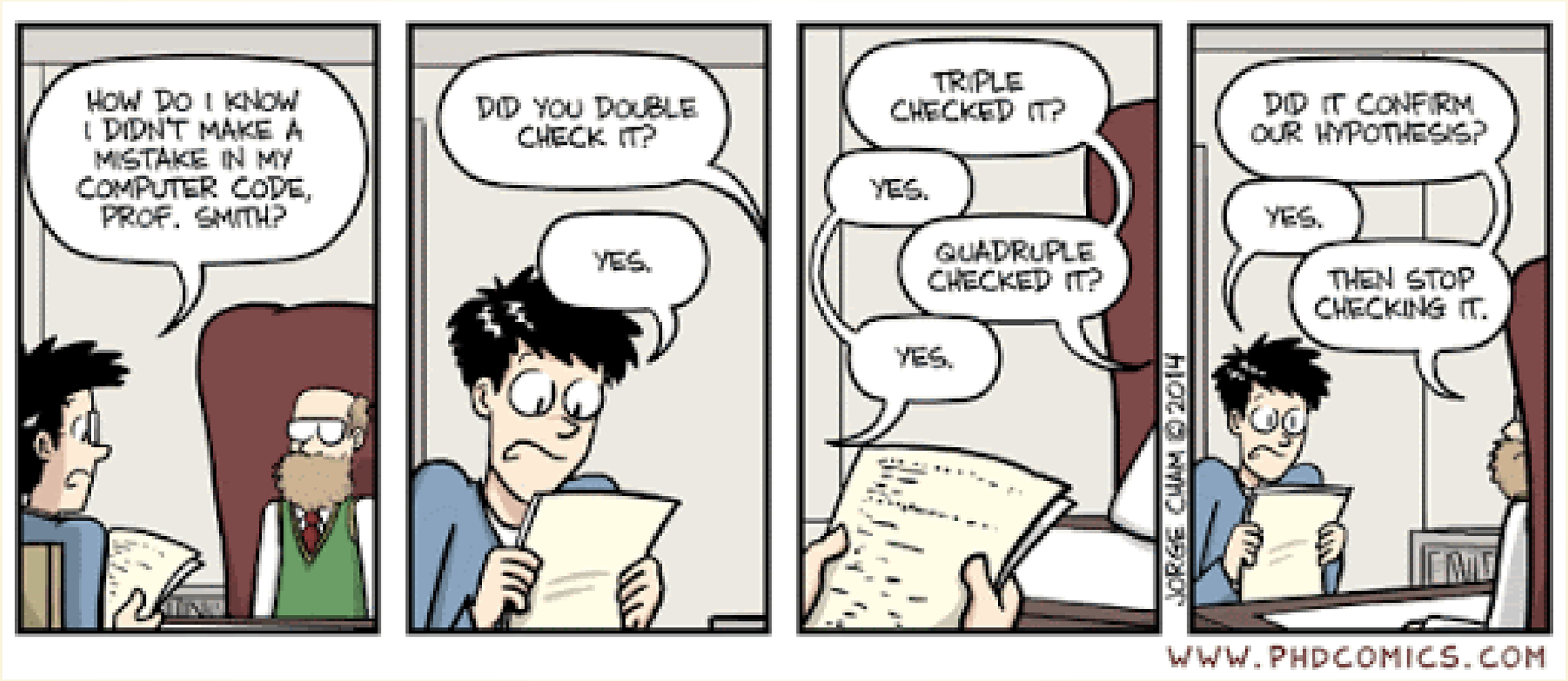


Image credit: http://phdcomics.com/comics.php?f=1693

COMMON PROBLEMS IN SCIENCE

A Post-doc wrote a script during the PhD that applied a specific method to a dataset. Now, with new data and a new project, they try to reuse the script, but forgot how it worked.

COMMON PROBLEMS IN SCIENCE

You try to recreate results from another lab's published paper. You base your re-implementation on everything reported in their paper, but the results you obtain look nowhere like the original.

SOUNDS FAMILIAR?

Did you encounter any of those in your work so far?

1. Forgot how own results were generated
2. Lost single source of data
3. Miscommunication about analysis with supervisor
4. Can't get previous code to run
5. Failure to reproduce other's work
6. Something else related to reproducibility

https://etc.ch/uhA

Direct Poll

~~COMMON~~ OLD PROBLEMS IN SCIENCE

All these problems were paraphrased from Buckheit & Donoho, **1995**

Data from many fields suggests reproducibility is lower than is desirable[8–14]; one analysis estimates that 85% of biomedical research efforts are wasted[14], while 90% of respondents to a recent survey in *Nature* agreed that there is a 'reproducibility crisis'[15]. Whether 'crisis' is the appropriate term to describe the current state or trajectory of science is debatable, but accumulated evidence indicates that there is substantial room for improvement with regard to research practices to maximize the efficiency of the research community's use of the public's financial investment in research.

"A manifesto for reproducible science" by Munafò et al., 2017, Nature Human Behavior

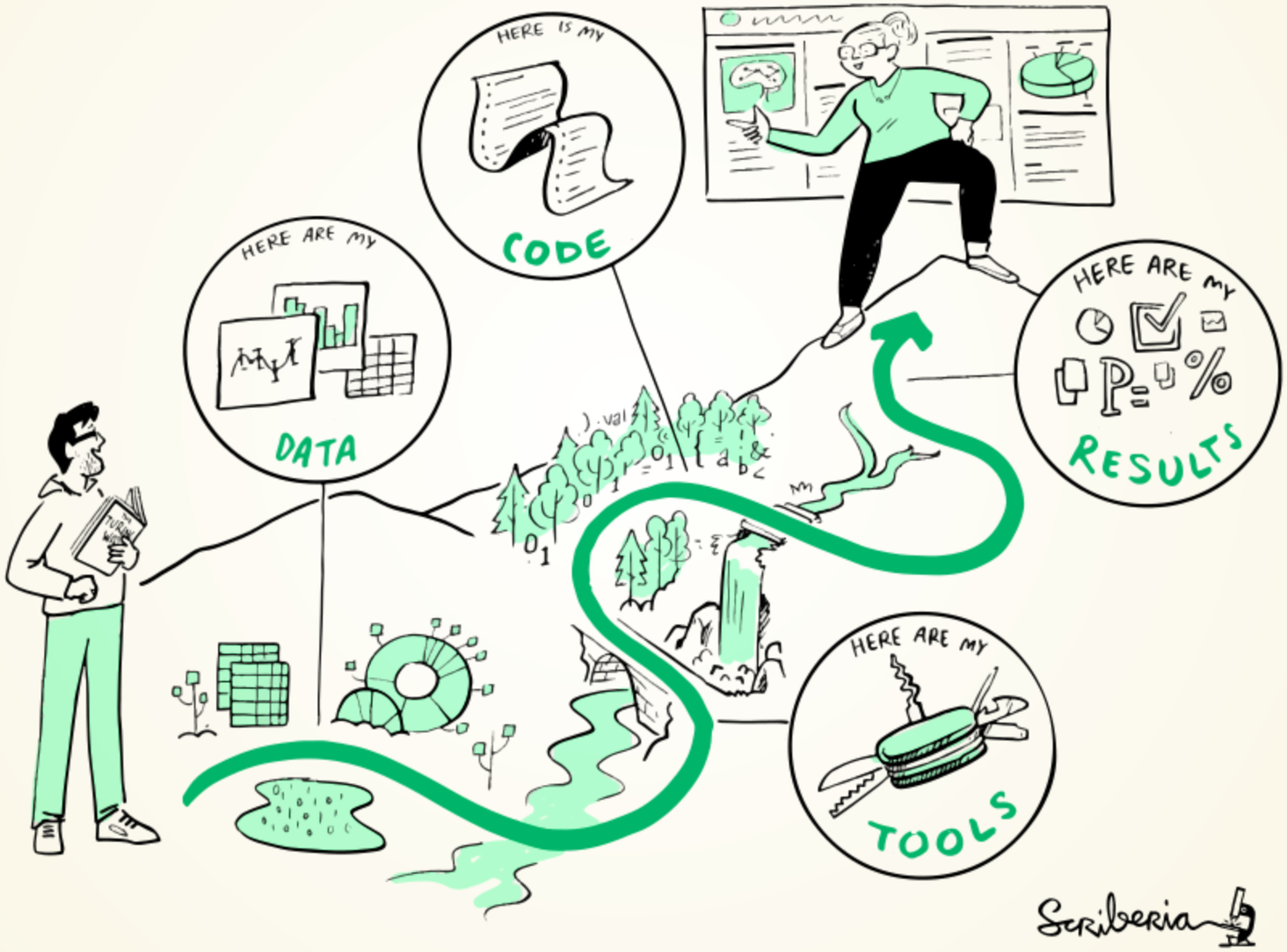
DEFINITIONS

	Same data	New data
Same methods	Reproducibility	Replication
New methods	Robustness	Generalization

see e.g., Freese & Peterson, 2017

"Authors provide all the necessary data and the computer codes to run the analysis again, re-creating the results." - Claerbout & Karrenbach, **1992**

THE ROAD TO REPRODUCIBILITY



The building blocks of a scientific result are rarely static

Analysis code evolves

(Fix bugs, add functions, refactor, ...)

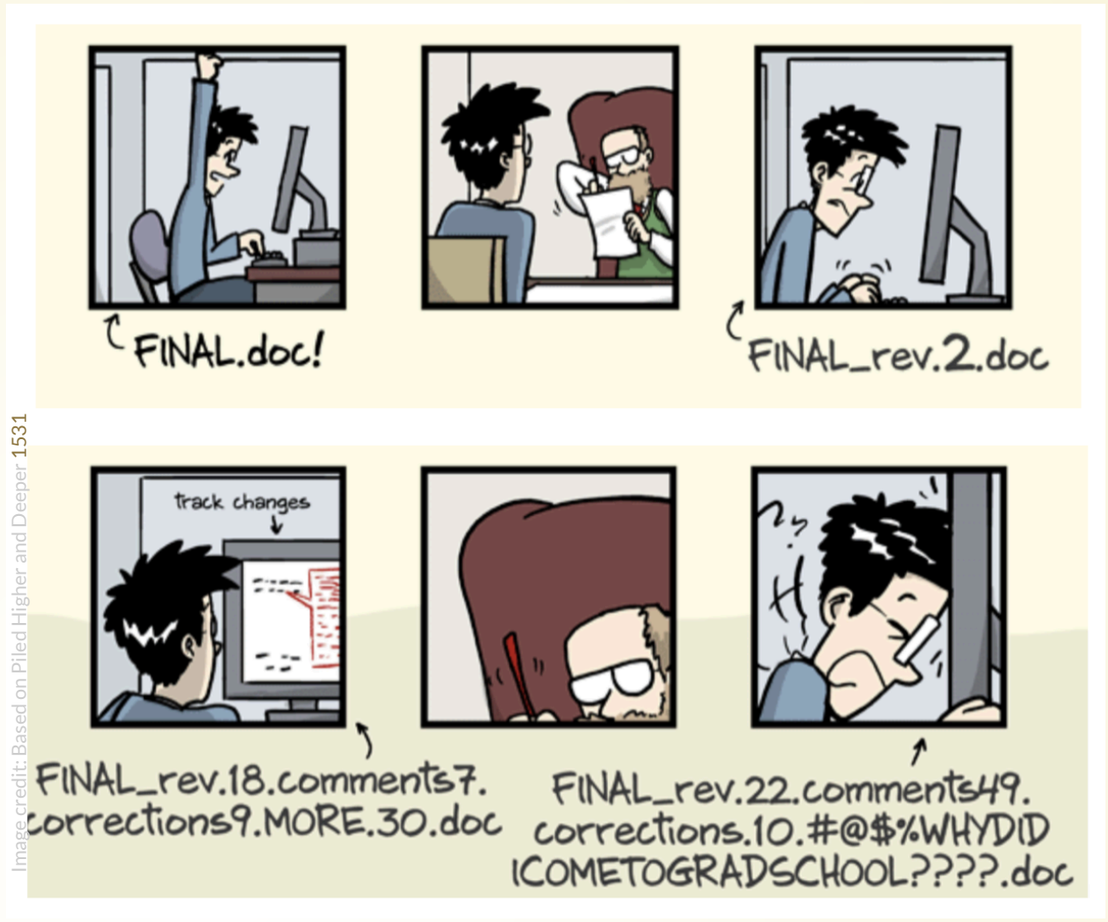


Image credit: Based on Piled Higher and Deeper 1531

Image credit: Scriberia and The Turing Way (CC-BY)

VERSION CONTROL

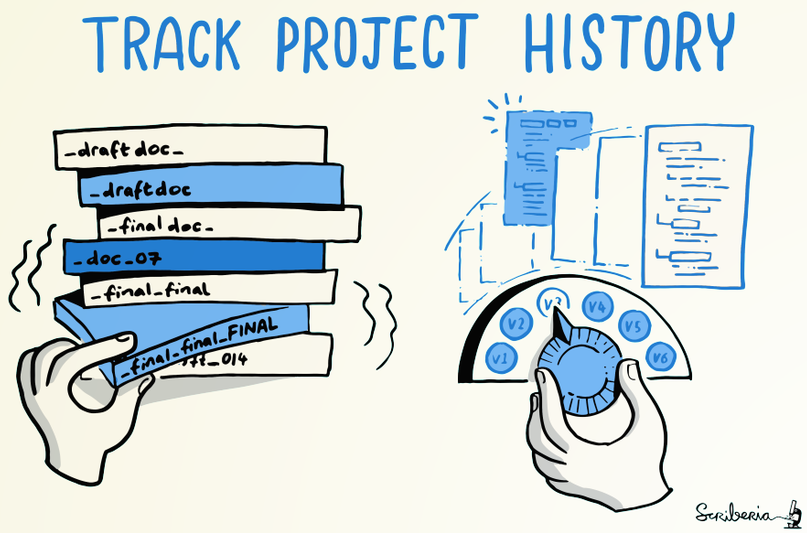


Image credit: CC-BY Scriberia & The Turing Way

Version control

- keep things organized
- keep track of changes
- revert changes or go back to previous states
- collect and share digital provenance
- industry standard: Git


```
2022-01-30 15:47 +0100 Michael Hanke    o Be explicit re FAIRification
2022-01-30 15:27 +0100 Michael Hanke    o Add statement on numerical precision
2022-01-30 11:36 +0100 Michael Hanke    o (Re)define RIA
2022-01-30 11:04 +0100 Małgorzata Wierzba  o Add MW's funding
2022-01-28 17:05 +0100 Felix Hoffstaedter  o reword bitidentity comment on reproducebility
2022-01-28 16:33 +0100 Adina Wagner     o Remove 'powerful' from snakemake's description as it is unspecific
2022-01-28 16:07 +0100 Adina Wagner     o R1: Finish the sentences on Dask and Spark
2022-01-28 15:10 +0100 Adina Wagner     o Revert "Move reference to {fig:imageqc} to results as well"
2022-01-28 14:35 +0100 Adina Wagner     o Add the compiled bibliography file into the repo, needed in resubmission
2022-01-28 14:28 +0100 Adina Wagner     o Apply @loj's suggestion on Parsl
2022-01-28 12:12 +0100 Małgorzata Wierzba  o Minor tweak
2022-01-28 11:40 +0100 Małgorzata Wierzba  o Fix typo
2022-01-28 11:36 +0100 Małgorzata Wierzba  o Move reference to {fig:imageqc} to results as well
2022-01-28 10:11 +0100 Małgorzata Wierzba  o Minor tweak
```

The building blocks of a scientific result are rarely static

Data changes

(errors are fixed, data is extended, naming standards change, an analysis requires only a subset of your data...)

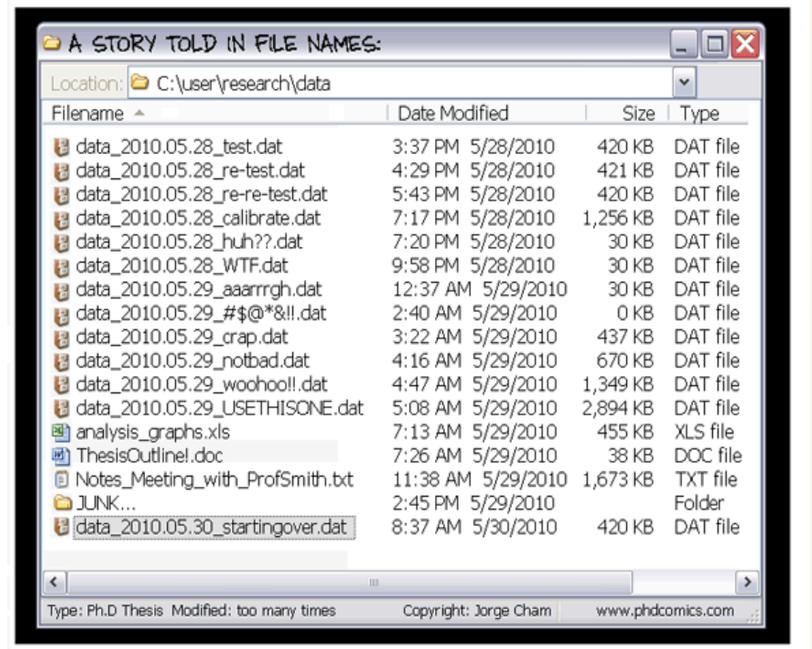


Image credit: Piled Higher and Deeper 1323

Large data version control (e.g., git-annex, DataLad)

```
2020-03-13 10:46 +0100 Adina Wagner     o [DATALAD RUNCMD] add non-defaced
2020-03-13 10:29 +0100 Adina Wagner     o [DATALAD RUNCMD] reconvert DICOM
2018-05-11 09:23 +0200 Michael Hanke    o [master] {origin/HEAD} {origin/m
2018-05-11 09:19 +0200 Michael Hanke    o Enable DataLad metadata extracto
2018-05-11 09:17 +0200 Michael Hanke    o [DATALAD] new dataset
2018-05-11 09:17 +0200 Michael Hanke    o [DATALAD] Set default backend fo
2018-01-19 14:19 +0100 Michael Hanke    o <v1.5> Update changelog for 1.5
2018-01-19 14:09 +0100 Michael Hanke    o BF: Re-import respiratory trace
2018-01-14 18:59 +0100 Michael Hanke    o Fix type in physio log converter
2017-01-10 10:10 +0100 Michael Hanke    o ENH: Report per-stimulus events
2016-12-10 20:18 +0100 Michael Hanke    o Add BIDS-compatible stimuli/ dir
2016-11-15 07:04 +0100 Michael Hanke    o Minor tweaks to gaze overlay scr
2016-10-30 11:03 +0100 Michael Hanke    o Add "TaskName" meta data field f
2016-09-21 08:33 +0200 Michael Hanke    o Add task-*_physio.json files
2016-09-21 08:23 +0200 Michael Hanke    o BF: Fix task label in file names
2016-08-04 13:14 +0200 Michael Hanke    o Update changelog
2016-08-03 22:22 +0200 Michael Hanke    o Add cut position information to
2016-05-27 17:35 +0200 Michael Hanke    o {origin/_} Mention openfmri as d
2016-04-04 09:31 +0200 Michael Hanke    o Update publication links
2016-03-31 11:26 +0200 Michael Hanke    o Disable invalid test
[main] 6da25fb6fee2c698d35f52066698b6f94850f4d2 - commit 10 of 79  27%
```

```
commit 6da25fb6fee2c698d35f52066698b6f94850f4d2
Refs: v1.0-19-g6da25fb6
Author:     Michael Hanke <michael.hanke@gmail.com>
AuthorDate: Fri Jan 19 14:09:53 2018 +0100
Commit:     Michael Hanke <michael.hanke@gmail.com>
CommitDate: Fri Jan 19 14:11:23 2018 +0100

    BF: Re-import respiratory trace after bug fix in converter (fixes gh-
---
 ...er_task-movielocalizer_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ..._task-objectcategories_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ..._task-objectcategories_run-2_recording-cardresp_physio.tsv.gz | 2 +-
 ..._task-objectcategories_run-3_recording-cardresp_physio.tsv.gz | 2 +-
 ..._task-objectcategories_run-4_recording-cardresp_physio.tsv.gz | 2 +-
 ...calizer_task-retmapccw_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...calizer_task-retmapclw_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...calizer_task-retmapcon_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...calizer_task-retmapexp_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...2_ses-movie_task-movie_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...2_ses-movie_task-movie_run-2_recording-cardresp_physio.tsv.gz | 2 +-
[diff] 6da25fb6fee2c698d35f52066698b6f94850f4d2 - line 1 of 2391  0%
```

LEAVING A TRACE

"Shit, which version of which script produced these outputs from which version of what data?"

"Shit, why buttons did I click and in which order did I use all those tools?"

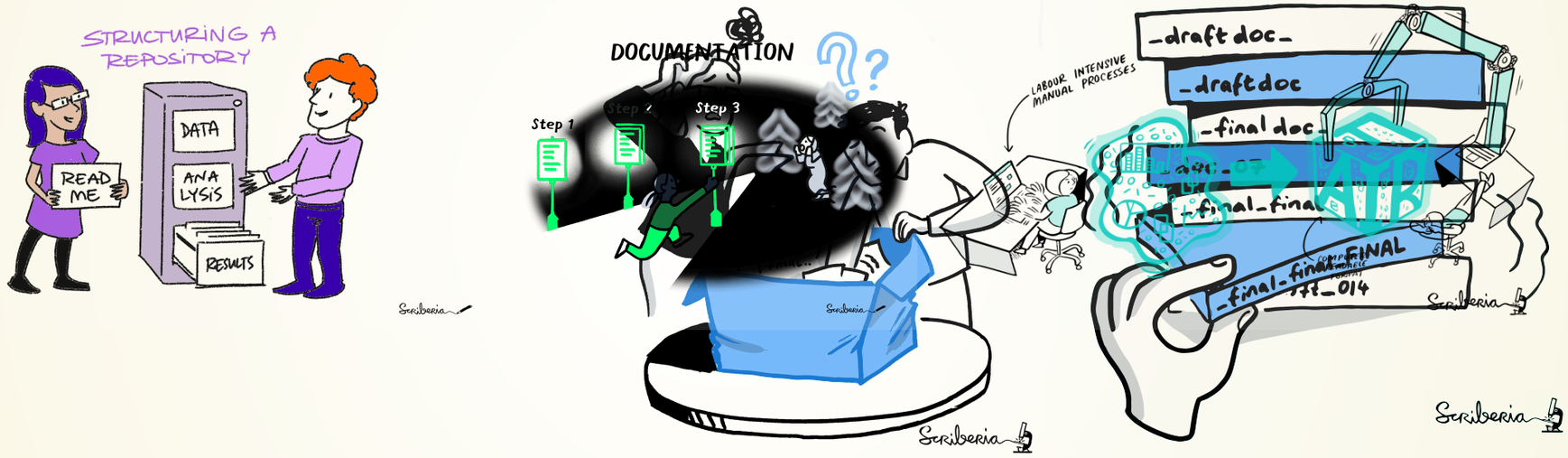


Image credit: CC-BY Scriberia and The Turing Way

1) Create an intuitive structure, and

2) write (plenty! of) documentation as you go, and

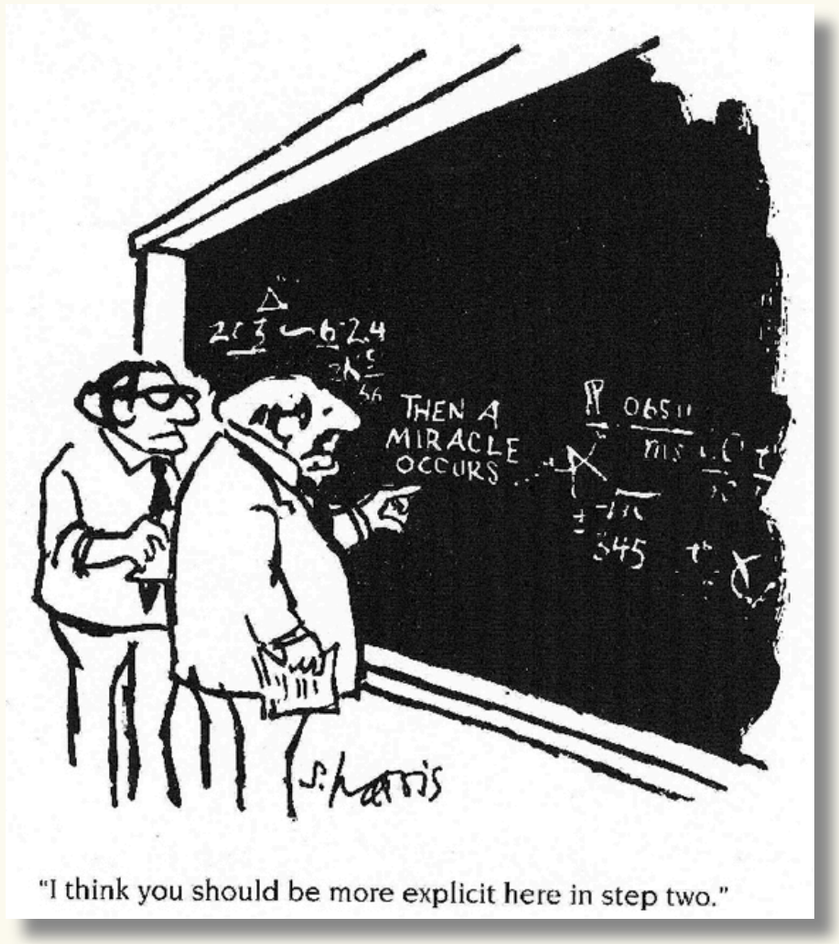
3) make your processes machine-readable

Tools and tricks: Perkel, 2020, checklist for computational reproducibility

METHODS DOCUMENTATION AND PROVENANCE

Analytic flexibility leads to sizeable variations in results

(see e.g., Carp. 2012 and Botvinik-Nezer, 2020 for examples from neuroimaging)



"I think you should be more explicit here in step two."

- provide information on how data came into existence
- change data through documented code, not manually
- relate changes in data to changes in code

REPRODUCIBILITY IS HIGHLY TECHNICAL

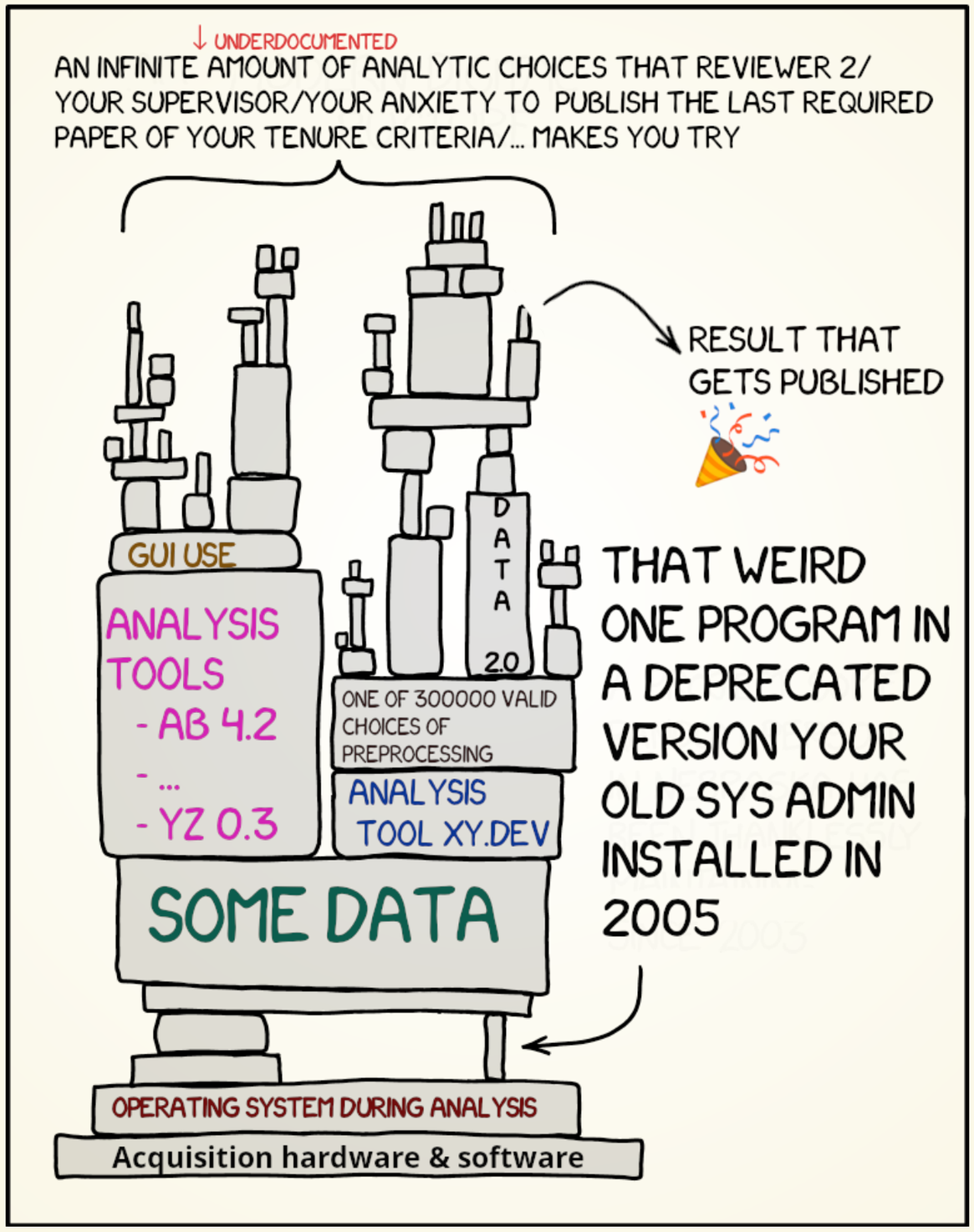


Image credit: Based on xkcd.com/2347/ (CC-BY)

Reproducibility Management in Neuroscience - Specific Issues and Solutions (DOI 10.5281/zenodo.4285927)

YOUR OWN REPRODUCIBILITY MANAGEMENT

What tools do you use to aid with reproducible science?

LET'S TRY DATALAD

jupyter.edu.datalad.org

username:

You got it per email (your first name)

password:

Set at first login, at least 8 characters

GIT IDENTITY SETUP

Check Git identity:

```
git config --get user.name
git config --get user.email
```

copy

Configure Git identity:

```
git config --global user.name "Adina Wagner"
git config --global user.email "adina.wagner@t-online.de"
```

copy

Configure DataLad to use latest features:

```
git config --global --add datalad.extensions.load next
```

copy

USING DATALAD IN A TERMINAL

Check the installed version:

```
datalad --version
```

copy

For help on using DataLad from the command line:

```
datalad --help
```

copy

```
The help may be displayed in a pager - exit it by pressing "q"
```

For extensive info about the installed package, its dependencies, and extensions, use `datalad wtf`. Let's find out what kind of system we're on:

```
datalad wtf -S system
```

copy

USING DATALAD VIA ITS PYTHON API

Open a Python environment:

```
ipython
```

copy

Import and start using:

```
import datalad.api as dl
dl.create(path='mydataset')
```

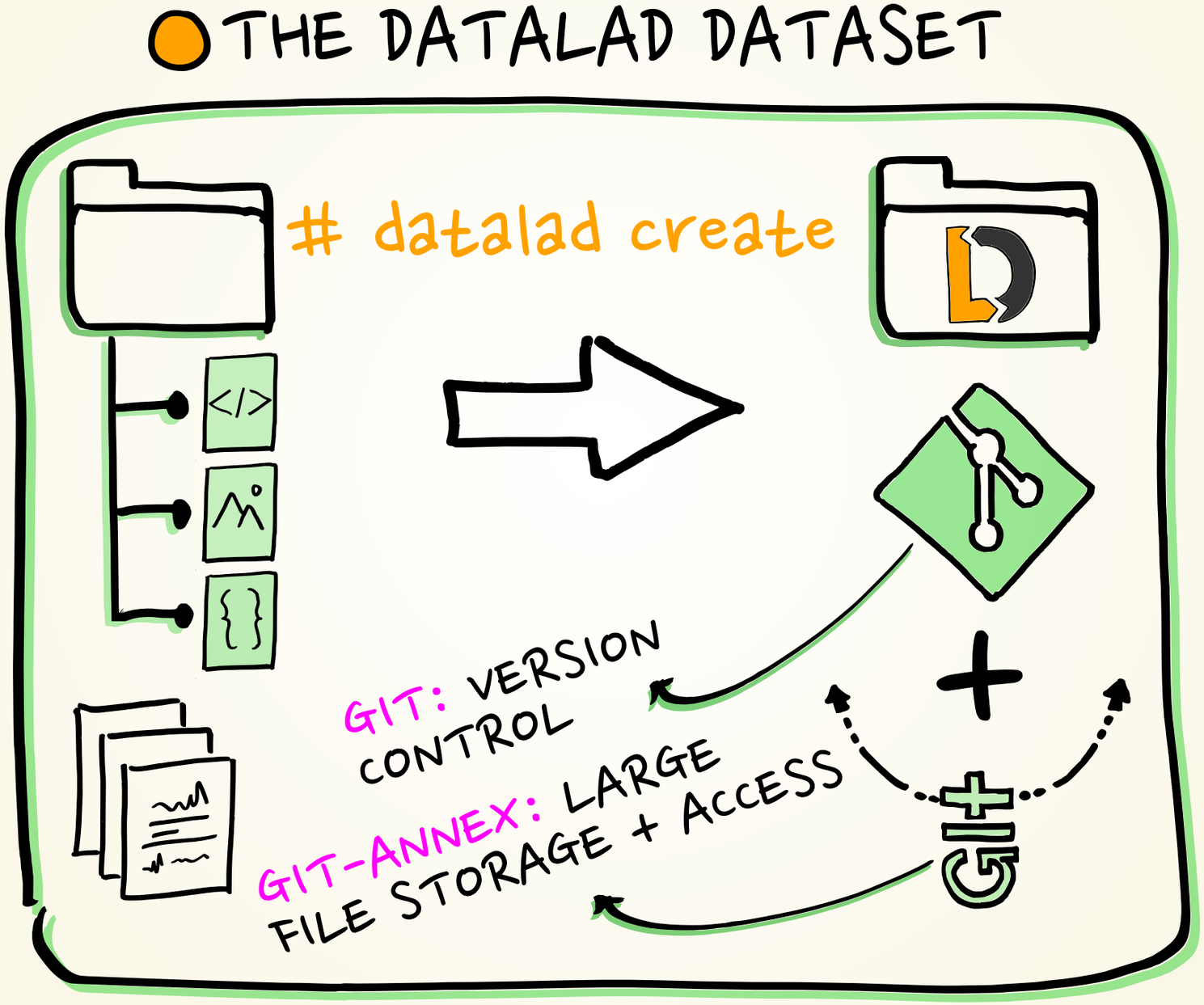
copy

Exit the Python environment:

```
exit
```

copy

DATALAD DATASETS...



...DATALAD DATASETS

Create a dataset (here, with the `yoda` configuration, which adds a helpful structure and configuration for data analyses):

```
datalad create -c yoda my-analysis
```

copy

Let's have a look inside. Navigate using `cd` (change directory):

```
cd my-analysis
```

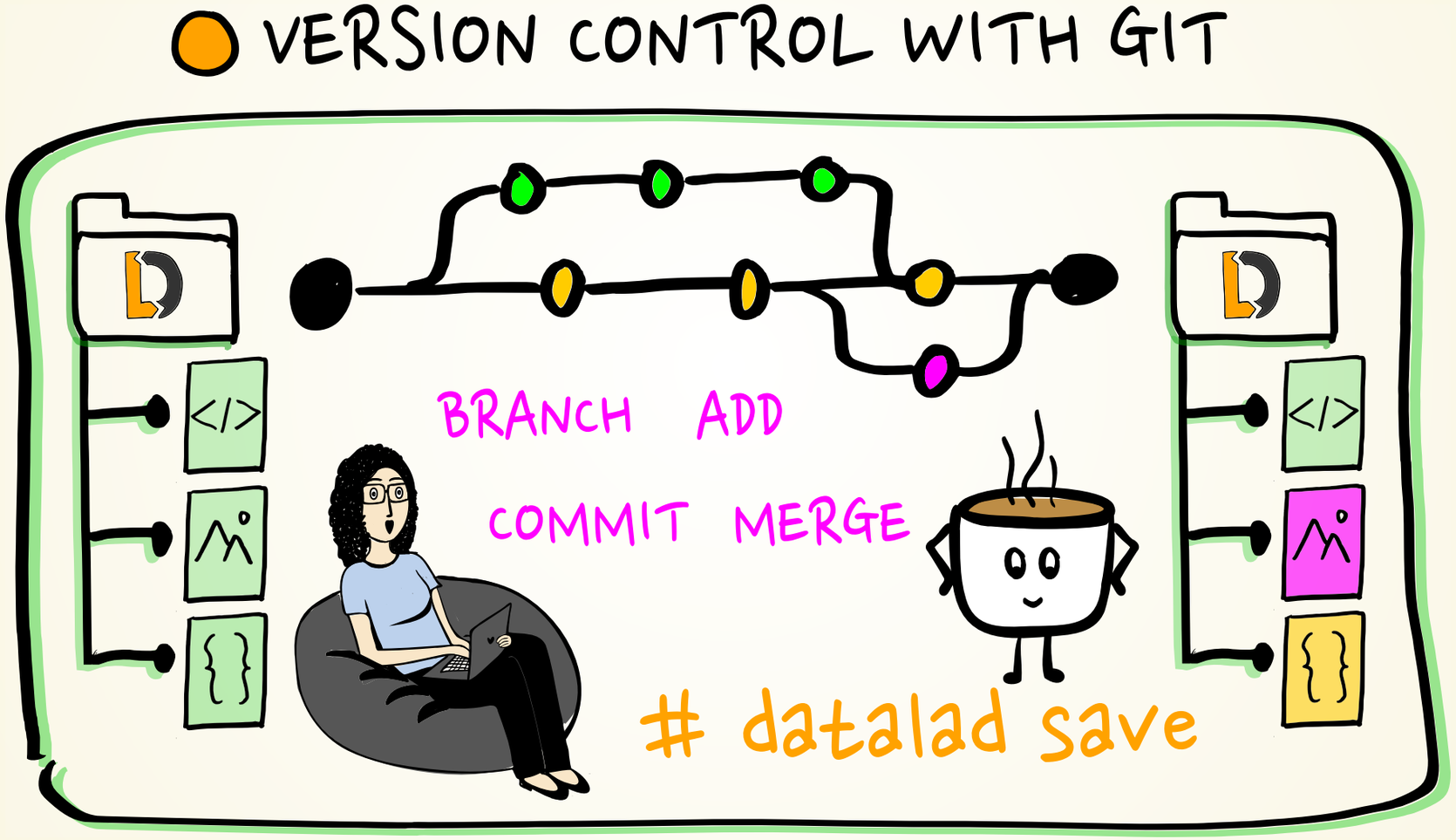
copy

List the directory content, including hidden files, with `ls`:

```
ls -la .
```

copy

VERSION CONTROL...



...VERSION CONTROL

The yoda-configuration added a README placeholder in the dataset. Let's add Markdown text (a project title) to it:

```
echo "# My example DataLad dataset\n\nContains a small data analysis for my project" >| README.md
```
copy

Now we can check the `status` of the dataset:

```
datalad status
```
copy

We can save the state with `save`

```
datalad save -m "Adjust boilerplate README to project"
```
copy

Let's add code for a data analysis from an external source:

```
wget https://hub.datalad.org/edu/scripts/raw/branch/main/iris/classification_analysis.py -O code/
```
copy

Save again:

```
datalad save -m "Add analysis script"
```
copy

...VERSION CONTROL

Now, let's check the dataset history:

```
git log
```
copy

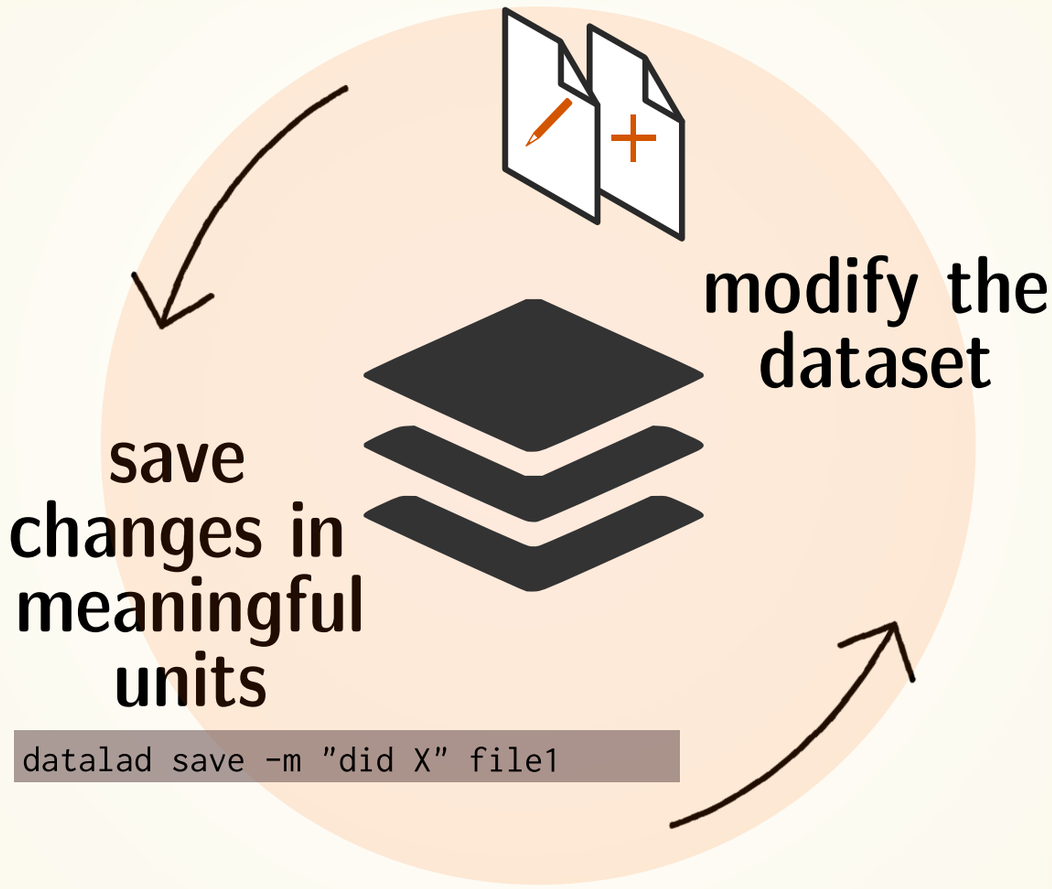
We can also make the history prettier:

```
tig
```
copy

```
(navigate with arrow keys and enter, press "q" to go back and exit the program)
```

LOCAL VERSION CONTROL

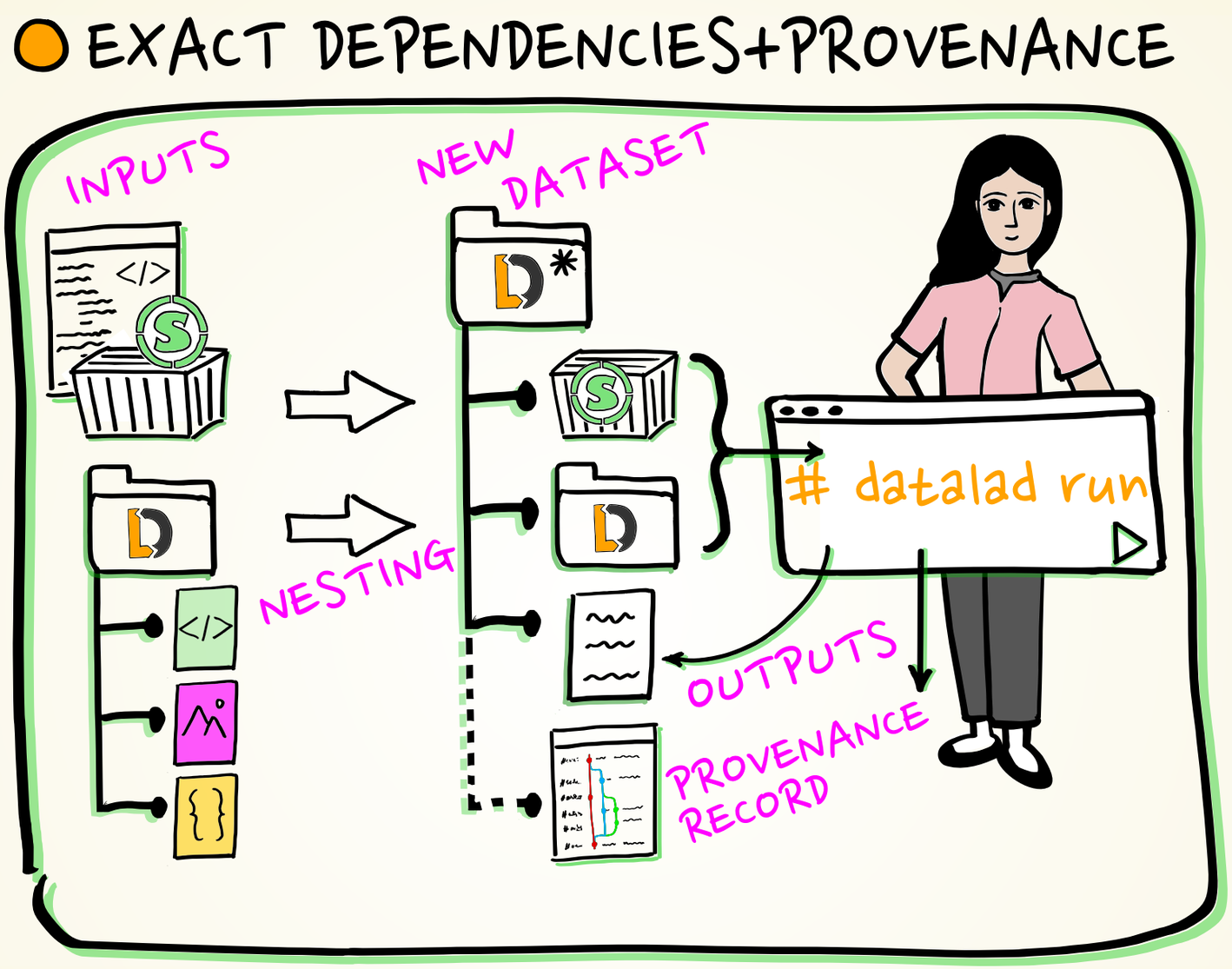
Procedurally, version control is easy with DataLad!



Advice:

- Save meaningful units of change
- Attach helpful commit messages

COMPUTATIONALLY REPRODUCIBLE EXECUTION I...



- which script/pipeline version
- was run on which version of the data
- to produce which version of the results?

... COMPUTATIONALLY REPRODUCIBLE EXECUTION I

A variety of processes can modify files. A simple example: Code formatting

```
black code/classification_analysis.py
```

Version control makes changes transparent:

```
git diff
```

But its useful to keep track beyond that. Let's discard the latest changes...

```
git restore code/classification_analysis.py
```

... and record precisely what we did

```
datalad run -m "Reformat code with black" \
 "black code/classification_analysis.py"
```

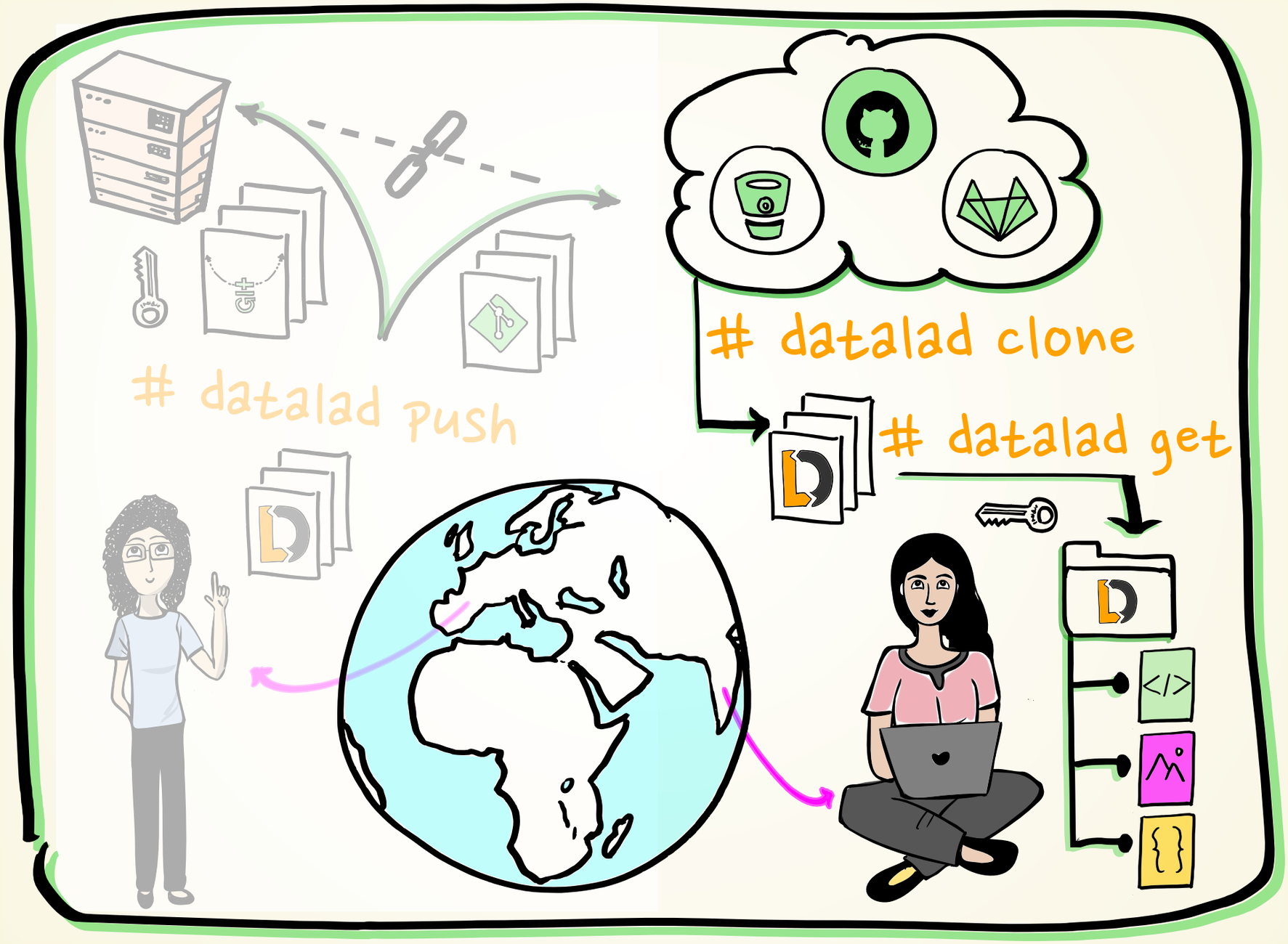
let's take a look:

```
git show
```

... and repeat!

```
datalad rerun
```

DATA CONSUMPTION & TRANSPORT...



...DATA CONSUMPTION & TRANSPORT...

You can install a dataset from remote URL (or local path) using `clone`. Either as a stand-alone entity:

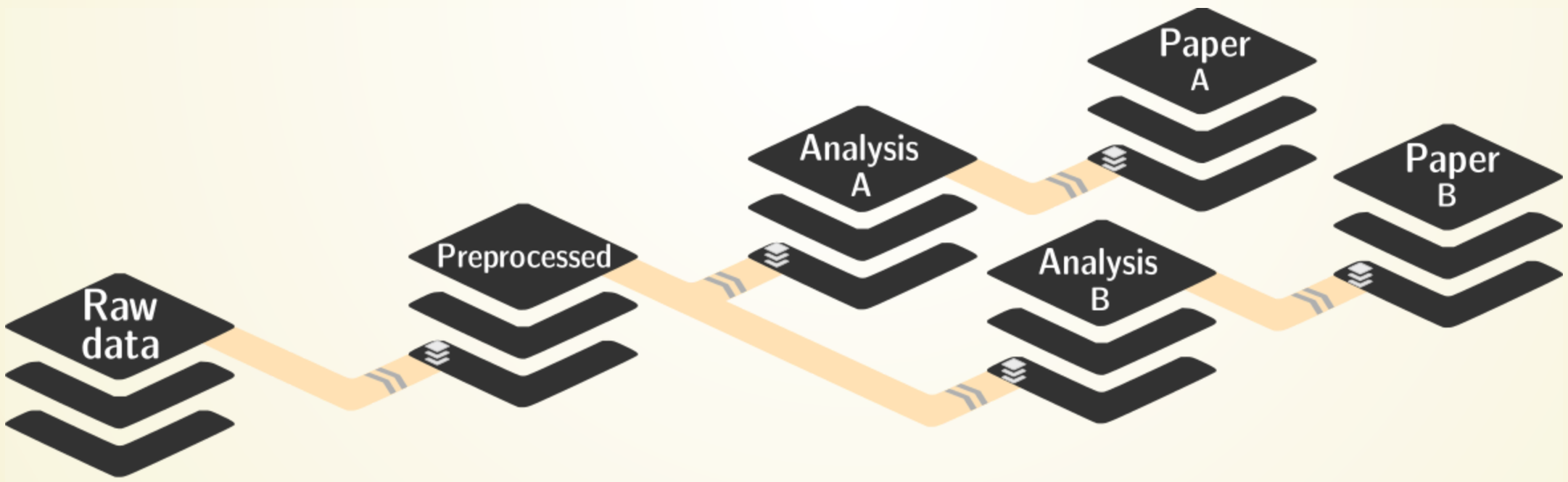
```
# just an example:
datalad clone \
https://github.com/psychoinformatics-de/studyforrest-data-phase2.git
```

copy

Or as linked dataset, nested in another dataset in a superdataset-subdataset hierarchy:

```
# just an example:
datalad clone -d . \
https://github.com/psychoinformatics-de/studyforrest-data-phase2.git
```

copy



...DATASET NESTING

Let's make a nest!
Clone a dataset with analysis data into a specific location ("input/") in the existing dataset, making it a *sub*dataset:

```
datalad clone --dataset . \
   https://hub.datalad.org/edu/iris_data.git \
   input/
```

copy

Let's see what changed in the dataset, using the `subdatasets` command:

```
datalad subdatasets
```

copy

... and also `git show`:

```
git show
```

copy

We can now view the cloned dataset's file tree:

```
cd input
ls
```

copy

...and also its history

```
tig
```

copy

Let's check the dataset size (with the du disk-usage command):

```
du -sh
```

copy

Let's check the *actual* dataset size:

```
datalad status --annex
```

copy

Let's check try to print the file contents into the terminal (`cat`):

```
cat iris.csv
```

copy

...DATA CONSUMPTION & TRANSPORT

We can retrieve actual file content with `get`:

```
datalad get iris.csv
```

If we don't need a file locally anymore, we can `drop` its content:

```
datalad drop iris.csv
```

No need to store all files locally, or archive results with Giga/Terra-Bytes of source data:

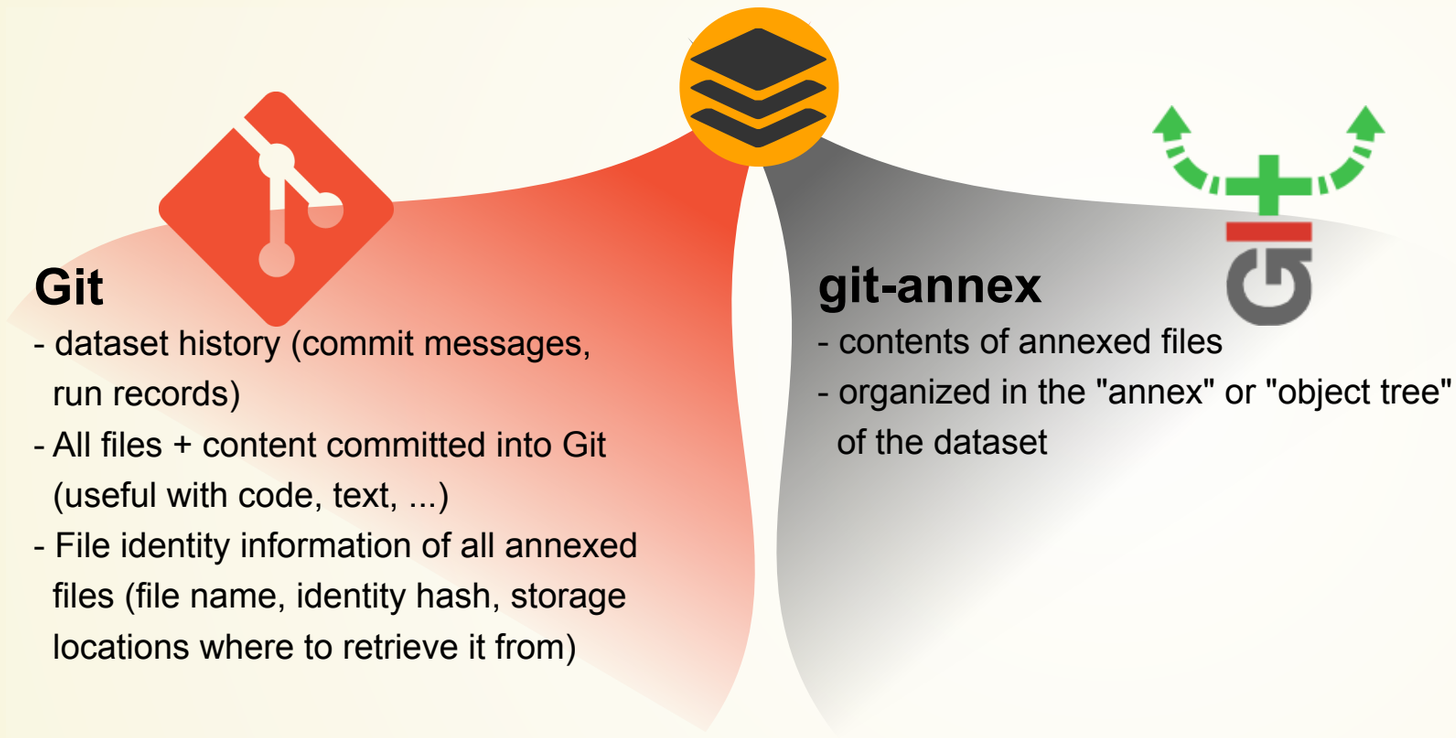
```
dl.get('input/sub-01')
[really complex analysis]
dl.drop('input/sub-01')
```

If data is published anywhere, your data analysis can carry an actionable link to it, with barely any space requirements.

GIT VERSUS GIT-ANNEX

Data in datasets is either stored in Git or git-annex

By default, everything is annexed, i.e., stored in a dataset annex by git-annex



- With annexed data, only content identity (hash) and location information is put into Git, rather than file content. The annex, and transport to and from it is managed with **git-annex**

GIT VERSUS GIT-ANNEX

...COMPUTATIONALLY REPRODUCIBLE EXECUTION...

Try to execute the downloaded analysis script. Does it work?

```
cd ..
python code/classification_analysis.py
```

copy

- Software can be difficult or impossible to install (e.g. conflicts with existing software, or on HPC) for you or your collaborators
- Different software versions/operating systems can produce different results: Glatard et al., doi.org/10.3389/fninf.2015.00012
- **Software containers** encapsulate a software environment and isolate it from a surrounding operating system. Two common solutions: Docker, Singularity

...COMPUTATIONALLY REPRODUCIBLE EXECUTION

With the `datalad-container` extension, we can add software containers to datasets and work with them. Let's add a software container with Python software to run the script

```
datalad containers-add python-env --url shub://adswa/resources:2
```

copy

inspect the list of registered containers:

```
datalad containers-list
```

copy

Now, let's try out the `containers-run` command:

```
datalad containers-run -m "run classification analysis in python environment" \
  --container-name python-env \
  --input "input/iris.csv" \
  --output "pairwise_relationships.png" \
  --output "prediction_report.csv" \
  "python3 code/classification_analysis.py {inputs} {outputs}"
```

copy

What changed after the `containers-run` command has completed?
We can use `datalad diff` (based on `git diff`):

```
datalad diff -f HEAD~1
```

copy

We see that some files were added to the dataset!
And we have a complete provenance record as part of the git history:

```
git log -n 1
```

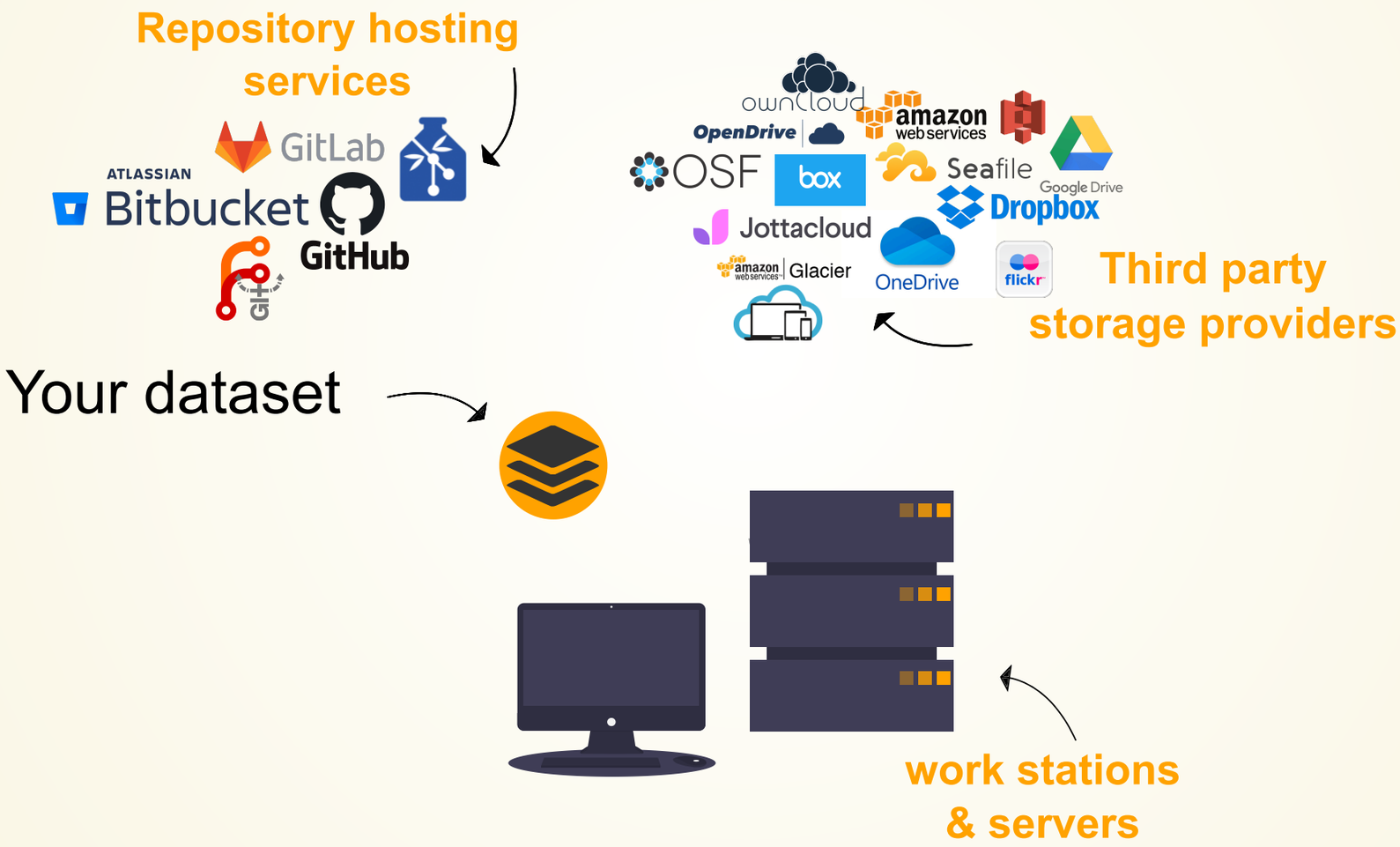
copy

...COMPUTATIONALLY REPRODUCIBLE EXECUTION...

- The `datalad run` can run any command in a way that links the command or script to the results it produces and the data it was computed from
- The `datalad rerun` can take this recorded provenance and recompute the command
- The `datalad containers-run` (from the extension "datalad-container") can capture software provenance in the form of software containers in addition to the provenance that datalad run captures

"SHARE DATA LIKE SOURCECODE"

Datasets can be cloned, pushed, and updated from and to **local** and **remote** paths, **remote hosting services**, external **special remotes**



We will use Forgejo-aneksajo: hub.edu.datalad.org:

OBJECTIVE: PUBLISH THE DATASET TO FORGEJO

Preparation: Obtain a token Go to hub.edu.datalad.org/user/settings

OBJECTIVE: PUBLISH THE DATASET TO FORGEJO

- Create a new repository `my-analysis` in the webinterface: https://hub.edu.datalad.org/repo/create
- Register a sibling / remote URL in the `my-analysis` dataset, using the URL https://hub.edu.datalad.org/USER-NAME/my-analysis.git (replace USER-NAME with your forgejo account name):

```
git remote add origin https://hub.edu.datalad.org/USER-NAME/my-analysis.git
```
copy

- Push the dataset and its file contents. What gets reported in your terminal?

```
datalad push --to origin
```
copy

(Supply your account name and the token as password when prompted in the terminal!)

IN THE FORGEJO WEBINTERFACE, EXPLORE YOUR NEWLY CREATED REPOSITORY.

OBJECTIVE: CLONE YOUR NEIGHBOURS DATASET

- Clone your right neighbours dataset (replace USER-NAME with *their* forgejo account name). Make sure you're not inside your own dataset.

```
datalad clone https://hub.edu.datalad.org/USER-NAME/my-analysis.git other-analysis
```
copy

  - Find the commit hash of their run commit. Rerun their analyses

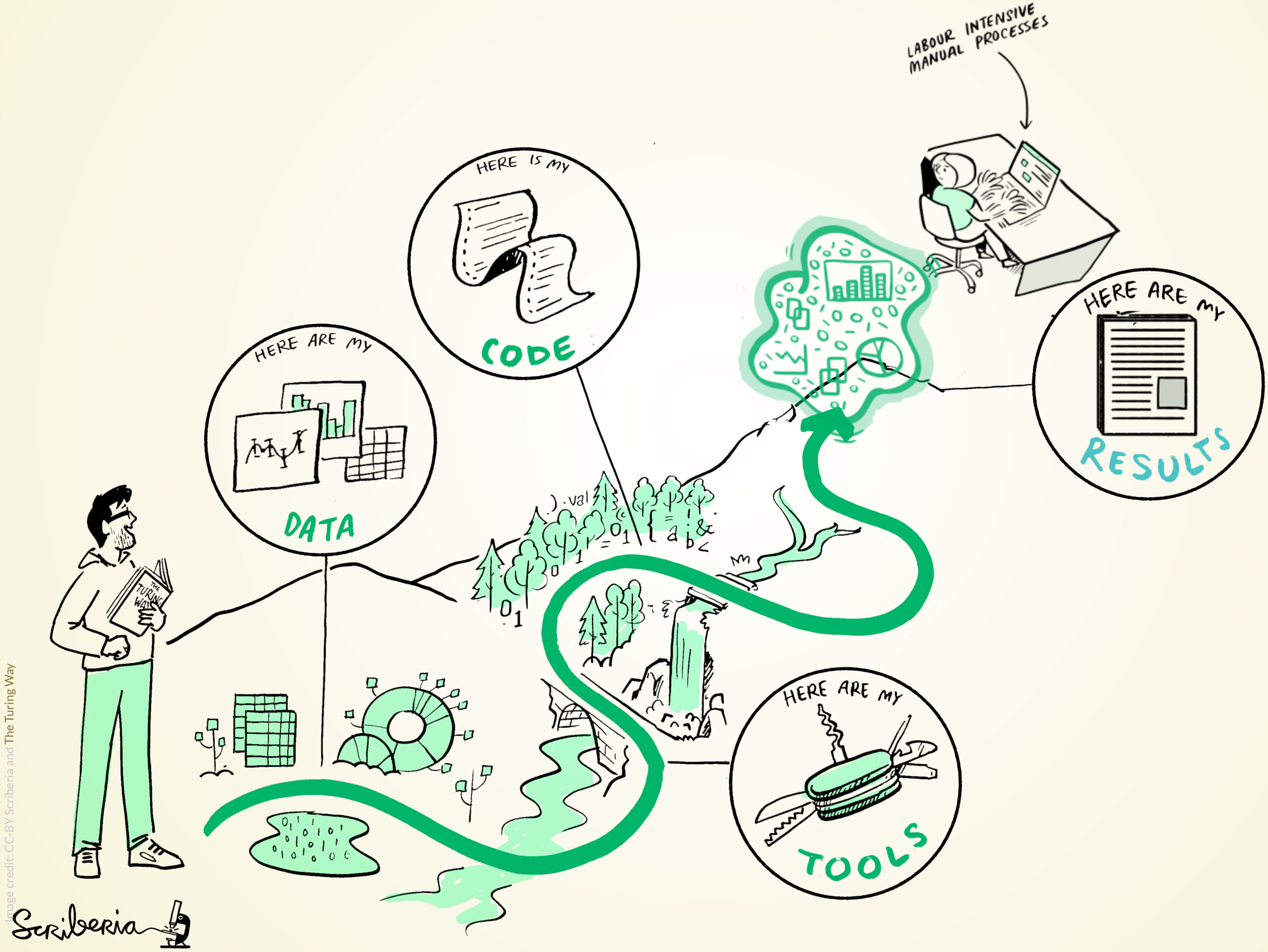
```
datalad rerun HASH
```
copy

Objective: Stay up to date

- While "push" publishes new developments, "datalad update" fetches or pulls them.
- "datalad update" *fetches*, "datalad update --how merge" *pulls* updates.
- "-s" declares the sibling to update from.
- "-r" performs a recursive update.
- Try pushing and pulling an update yourself.

```
datalad update --how merge -s origin
```

SHARE THE INGREDIENTS, BUT ALSO THE RECIPE!



BUT WHAT'S IN IT FOR ME? "SELFISH" REASONS FOR REPRODUCIBILITY

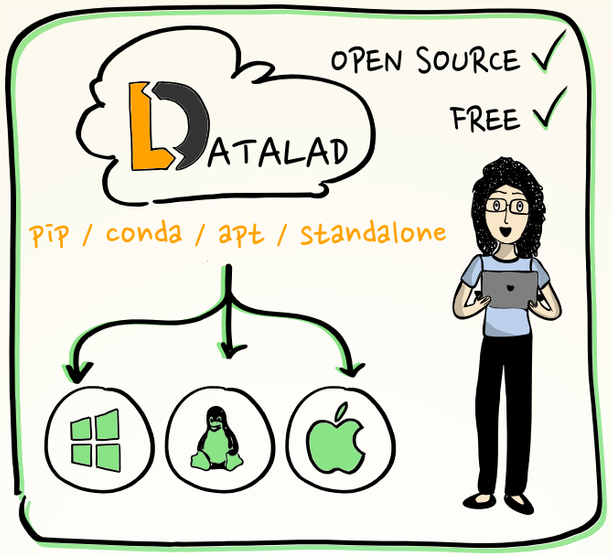
"[...] science is all about more publications, more impact factor, more money and more career. More, more, more ...
So how does working reproducibly help me achieve more as a scientist?" - Markowetz, 2015

- You want to avoid the disaster of publishing "a miracle"
- You will be faster (in the long run)
  - Finding and fixing errors will be faster
  - Progress on new projects will happen faster
- Researchers (reviewers!) will have more trust in your findings
- Data sharing can foster collaboration (with your past self, inside and outside your institution) and lead to new projects and publications
- You acquire (technical) skills that will likely become increasingly important for your career, either in academia or industry

It's just useful for your everyday work and makes your life easier!

see e.g., Markowetz, 2015, Genome Biology; Poldrack, 2019, Neuron

DATALAD



- Domain-agnostic **command-line tool** (+ **graphical user interface**), built on top of Git & Git-annex
- Major features:

  Version-controlling arbitrarily large content

  Version control data & software alongside to code!

  Transport mechanisms for sharing & obtaining data

  Consume & collaborate on data (analyses) like software

  (Computationally) reproducible data analysis

  Track and share provenance of all digital objects

  (... and much more)

FURTHER RESOURCES AND STAY IN TOUCH

If you have questions after the workshop...

Reach out to the DataLad team via

- Matrix (free, decentralized communication app, no app needed). We run a weekly Zoom office hour (Monday, 2pm Berlin time) from this room as well.
- The development repository on GitHub

Reach out to the (Neuro-) user community with

- A question on neurostars.org with a `datalad` tag

Find more user tutorials or workshop recordings

- On DataLad's YouTube channel
- In the DataLad Handbook
- In the DataLad RDM course
- In the Official API documentation
- In an overview of most tutorials, talks, videos at github.com/datalad/tutorials

ACKNOWLEDGEMENTS

DataLad software & ecosystem

- Psychoinformatics Lab, Research center Jülich
- Center for Open Neuroscience, Dartmouth College
- Joey Hess (git-annex)
- *>100 additional contributors*

Funders

NSF 1429999

germany-usa

SPONSORED BY THE Federal Ministry of Education and Research

BMBF 01GQ1411

JÜLICH Forschungszentrum

DFG

EUROPEAN UNION
European Regional Development Fund

cbbs center for behavioral brain sciences

SACHSEN-ANHALT
Ministerium für Wissenschaft und Wirtschaft

Collaborators

Human Brain Project

CONP PCNO

OpenNEURO

eBRAIN Health

MOTOR SFB 1451

brainlife.io

cbrain

VirtualBrainCloud

THANK YOU FOR YOUR ATTENTION!

Slides: DOI 10.5281/zenodo.19692938 (Scan the QR code)

Women in Neuroscience

Women neuroscientists are underrepresented in neuroscience. You can use the Repository for Women in Neuroscience to find and recommend neuroscientists for conferences, symposia or collaborations, and help making neuroscience more open & divers.

HOW DOES THIS RELATE TO REPRODUCIBILITY?

EXHAUSTIVE TRACKING

The building blocks of a scientific result are rarely static

Data changes

(errors are fixed, data is extended, naming standards change, an analysis requires only a subset of your data...)

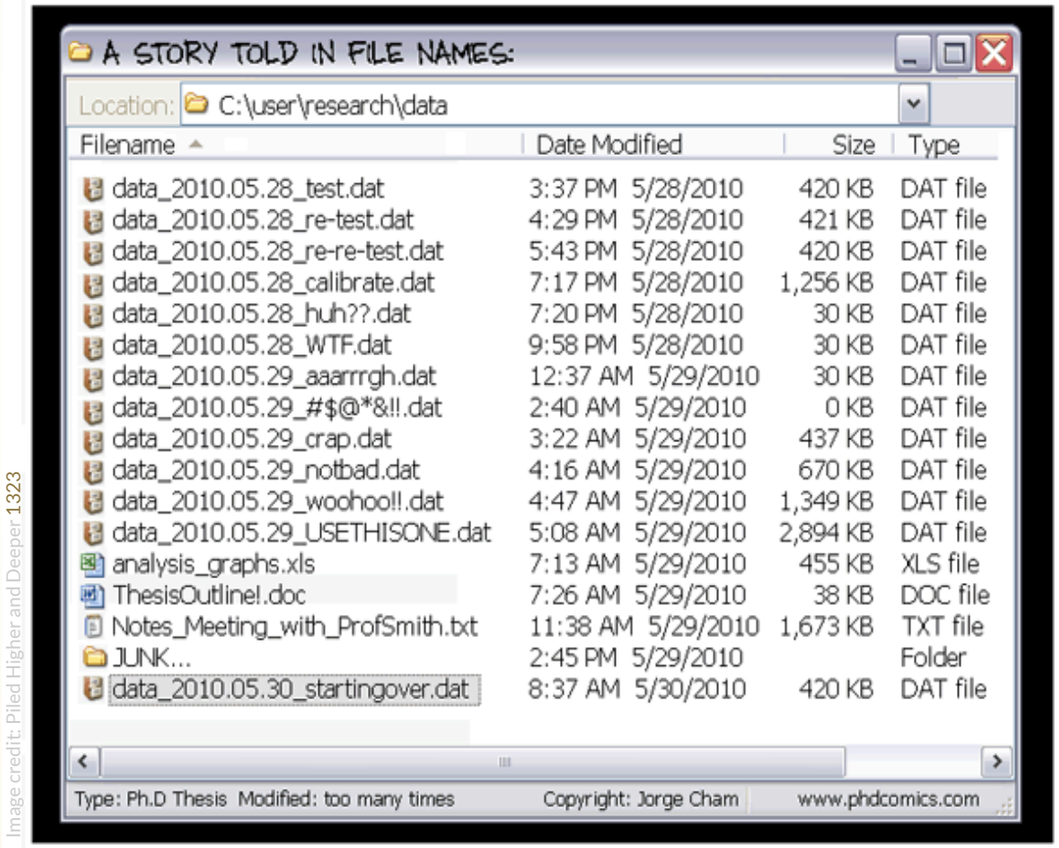


Image credit: Piled Higher and Deeper 1323

EXHAUSTIVE TRACKING

Once you track changes to data with version control tools, you can find out *why* it changed, *what* has changed, *when* it changed, and *which version* of your data was used at which point in time.

```
2020-03-13 10:46 +0100 Adina Wagner     o [DATALAD RUNCMD] add non-defaced
2020-03-13 10:29 +0100 Adina Wagner     o [DATALAD RUNCMD] reconvert DICOM
2018-05-11 09:23 +0200 Michael Hanke    o [master] {origin/HEAD} {origin/m
2018-05-11 09:19 +0200 Michael Hanke    o Enable DataLad metadata extracto
2018-05-11 09:17 +0200 Michael Hanke    o [DATALAD] new dataset
2018-05-11 09:17 +0200 Michael Hanke    o [DATALAD] Set default backend fo
2018-01-19 14:19 +0100 Michael Hanke    o <v1.5> Update changelog for 1.5
2018-01-19 14:09 +0100 Michael Hanke    o BF: Re-import respiratory trace
2018-01-14 18:59 +0100 Michael Hanke    o Fix type in physio log converter
2017-01-10 10:10 +0100 Michael Hanke    o ENH: Report per-stimulus events
2016-12-10 20:18 +0100 Michael Hanke    o Add BIDS-compatible stimuli/ dir
2016-11-15 07:04 +0100 Michael Hanke    o Minor tweaks to gaze overlay scr
2016-10-30 11:03 +0100 Michael Hanke    o Add "TaskName" meta data field f
2016-09-21 08:33 +0200 Michael Hanke    o Add task-*_physio.json files
2016-09-21 08:23 +0200 Michael Hanke    o BF: Fix task label in file names
2016-08-04 13:14 +0200 Michael Hanke    o Update changelog
2016-08-03 22:22 +0200 Michael Hanke    o Add cut position information to
2016-05-27 17:35 +0200 Michael Hanke    o {origin/_} Mention openfmri as d
2016-04-04 09:31 +0200 Michael Hanke    o Update publication links
2016-03-31 11:26 +0200 Michael Hanke    o Disable invalid test
[main] 6da25fb6fee2c698d35f52066698b6f94850f4d2 - commit 10 of 79      27%
```

```
commit 6da25fb6fee2c698d35f52066698b6f94850f4d2
Refs: v1.0-19-g6da25fb6
Author:     Michael Hanke <michael.hanke@gmail.com>
AuthorDate: Fri Jan 19 14:09:53 2018 +0100
Commit:     Michael Hanke <michael.hanke@gmail.com>
CommitDate: Fri Jan 19 14:11:23 2018 +0100

    BF: Re-import respiratory trace after bug fix in converter (fixes gh-
---
 ...er_task-movielocalizer_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ..._task-objectcategories_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ..._task-objectcategories_run-2_recording-cardresp_physio.tsv.gz | 2 +-
 ..._task-objectcategories_run-3_recording-cardresp_physio.tsv.gz | 2 +-
 ..._task-objectcategories_run-4_recording-cardresp_physio.tsv.gz | 2 +-
 ...calizer_task-retmapccw_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...calizer_task-retmapclw_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...calizer_task-retmapcon_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...calizer_task-retmapexp_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...2_ses-movie_task-movie_run-1_recording-cardresp_physio.tsv.gz | 2 +-
 ...2_ses-movie_task-movie_run-2_recording-cardresp_physio.tsv.gz | 2 +-
[diff] 6da25fb6fee2c698d35f52066698b6f94850f4d2 - line 1 of 2391      0%
```

DIGITAL PROVENANCE

DATA TRANSPORT: SECURITY AND RELIABILITY - FOR DATA

Decentral version control for data integrates with a variety of services to let you store data in different places - creating a resilient network for data

ULTIMATE GOAL: REUSABILITY

Teamscience on more than code:

Q: Adjustment to tiny labeling mistake (ref Zemblys, 2018)? #2

Closed **adswa** opened this issue on Mar 8, 2019 · 1 comment

adswa commented on Mar 8, 2019 — Contributor

Zemblys et al., 2018, report a minor labeling mistake in one image file here (or a pay-wall free version here) in Appendix 2:

> We found an obvious labeling mistake in the one of the validation trials (file UH29_img_Europe_labelled_MN. We fixed this error by reassigning 75 samples, [3197,3272) (zero-based index), from the saccade to the fixation class.

I checked the data file in question and it appears to still contain the erroneous saccade labels. Just to reconfirm: this labeling error has not been fixed in the data file in this repository, correct?

If you wish, I can PR a fixed file, the issue at hand is intended to just reconfirm my assumption.

Thanks in advance!

adswa added a commit to adswa/remodnav that referenced this issue on Mar 8, 2019

`ENH/FIX: use file with fixed labels.` ··· `1b2b162`

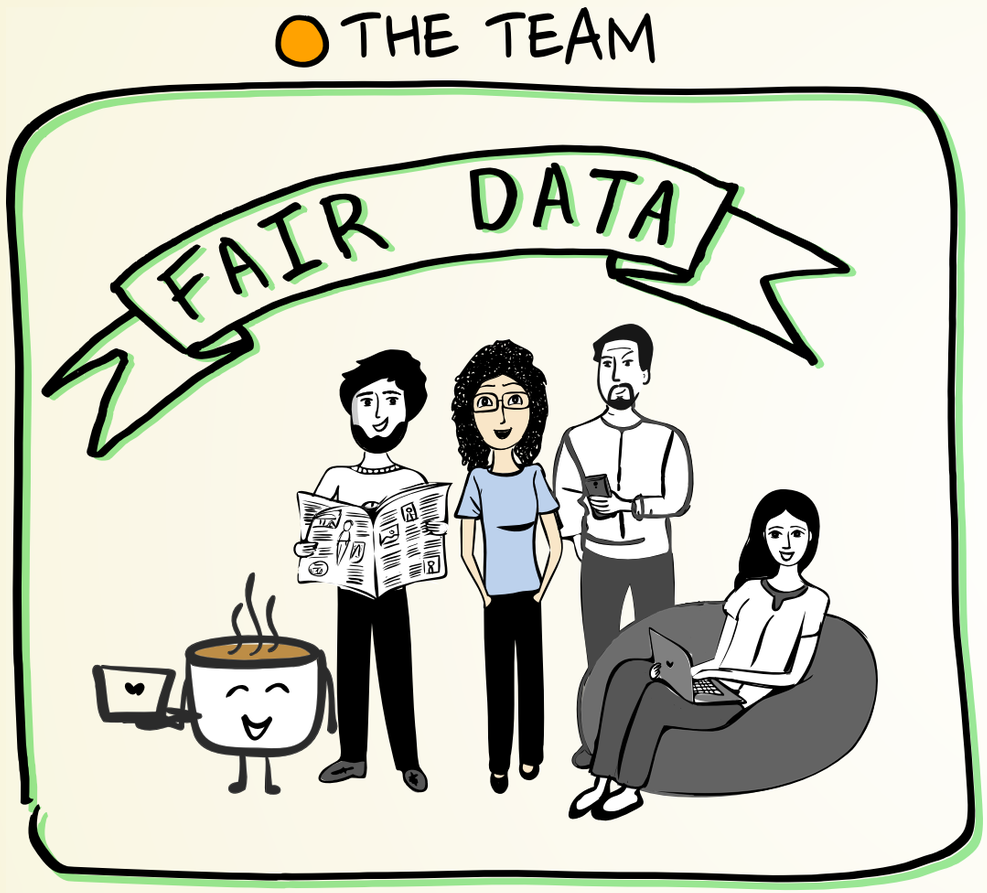
```
2019-03-08 12:38 +0100 Richard Andersson M─┐  [master] {origin/master} {origin/HEAD} Merge pull request #3 from AdinaWagner/datafix
2019-03-08 11:35 +0100 Adina Wagner        │ o ENH/FIX: relabel erroneous saccades to fixations, closes #2.
2018-12-05 15:27 +0100 Richard Andersson o─┘ Uploaded a folder consting only of the data used in the original article
2017-08-22 19:40 +0200 Richard Andersson o Code added
2016-12-14 10:35 +0100 richardandersson  o Added currently shared data and stimuli.
2016-12-14 10:29 +0100 richardandersson  o Deleted -- to be reuploaded
2016-12-14 10:08 +0100 Richard Andersson o Add files via upload
2016-12-14 10:07 +0100 Richard Andersson o Add files via upload
2016-12-14 10:04 +0100 Richard Andersson I Initial commit
```

Fixed data

Original data

DATALAD USECASES

A COMMON USECASE



- Alice is a PhD student in a research team.
- She works on a fairly typical research project: Data collection & processing.
- First sample → final result = complex process

HOW DOES ALICE GO ABOUT HER DAILY JOB?

A COMMON USECASE

- In her project, Alice likes to have an automated record of:
  - when a given file was last changed
  - where it came from
  - what input files were used to generate a given output
  - why some things were done.

- Even if she doesn't share her work, this is essential for her future self
- Her project is exploratory: Frequent changes to her analysis scripts
- She enjoys the comfort of being able to return to a previously recorded state

THIS IS: *LOCAL VERSION CONTROL*

A COMMON USECASE

- Alice's work is not confined to a single computer:
  - Laptop / desktop / remote server / dedicated back-up
  - Alice wants to automatically & efficiently synchronize

- Parts of the data are collected or analyzed by colleagues. This requires:
  - distributed synchronization with centralized storage
  - preservation of origin & authorship of changes
  - effective combination of simultaneous contributions

THIS IS: *DISTRIBUTED VERSION CONTROL*

A COMMON USECASE

- Alice applies local version control for her own work, and reproducibly records it
- She also applies distributed version control when working with colleagues and collaborators
- She often needs to work on a subset of data at any given time:
  - all files are kept on a server
  - a few files are rotated into and out of her laptop
- Alice wants to publish the data at project's end:
  - raw data / outputs / both
  - completely or selectively

THIS IS: *DATA MANAGEMENT (WITH DATALAD 😀)*