DATA MANAGEMENT WITH DATALAD

AN INTRODUCTION

Adina Wagner
@AdinaKrik

JÜLICH
Forschungszentrum

Psychoinformatics lab,
Institute of Neuroscience and Medicine, Brain & Behavior (INM-7)
Research Center Jülich

Slides: https://github.com/datalad-handbook/course/

WHAT IS (RESEARCH) DATA MANAGEMENT?

Image credit: JISC; CC-BY-SA-ND

WHAT IS (RESEARCH) DATA MANAGEMENT?

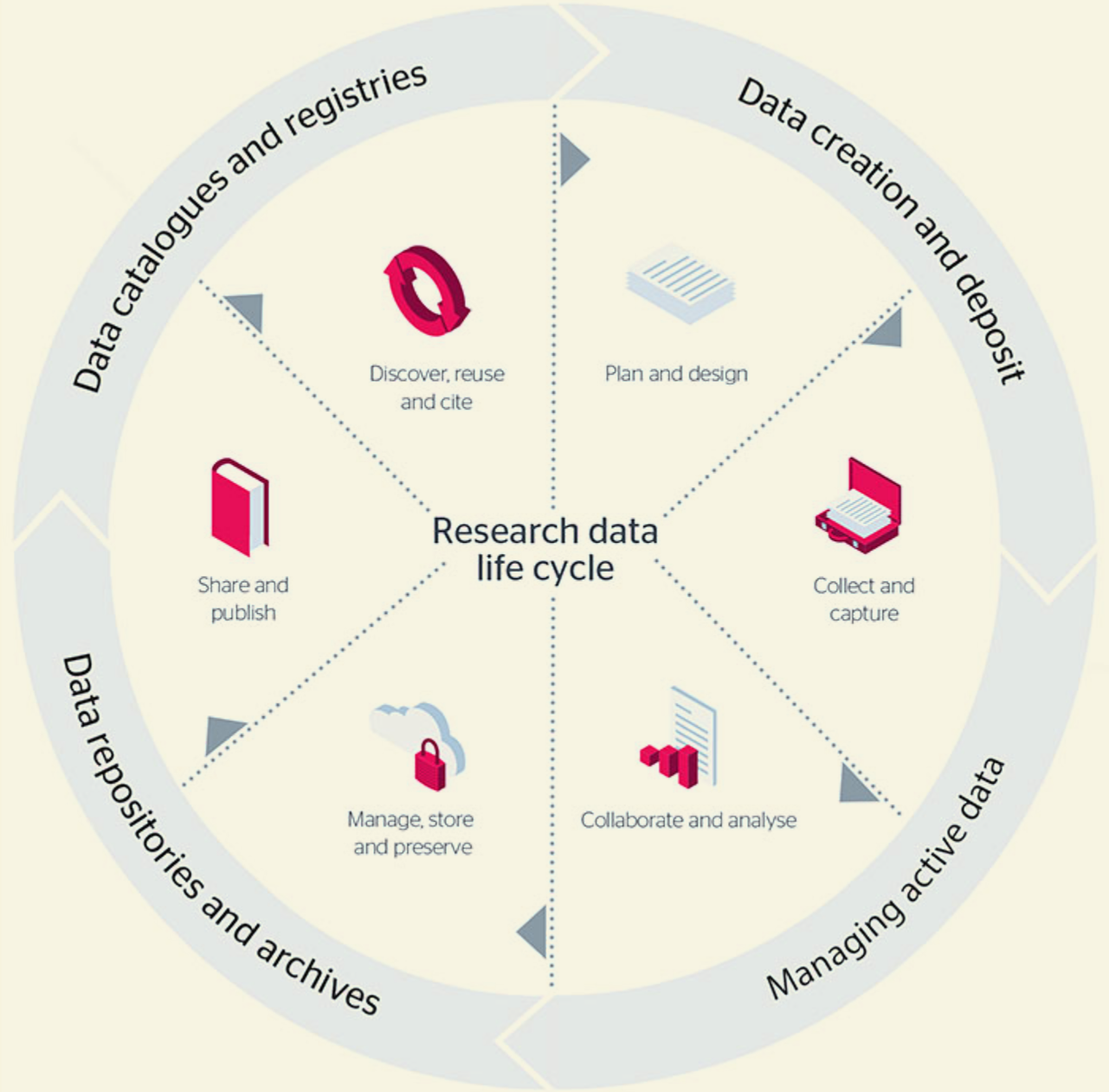
- (Research) Data = every digital object involved in your project: code, software/tools, raw data, processed data, results, manuscripts ...

WHAT IS (RESEARCH) DATA MANAGEMENT?

- (Research) Data = every digital object involved in your project: code, software/tools, raw data, processed data, results, manuscripts ...
- ... needs to be properly managed - from its creation to its use, publication, sharing, archiving, re-use, or destruction... (keyword: FAIR data)

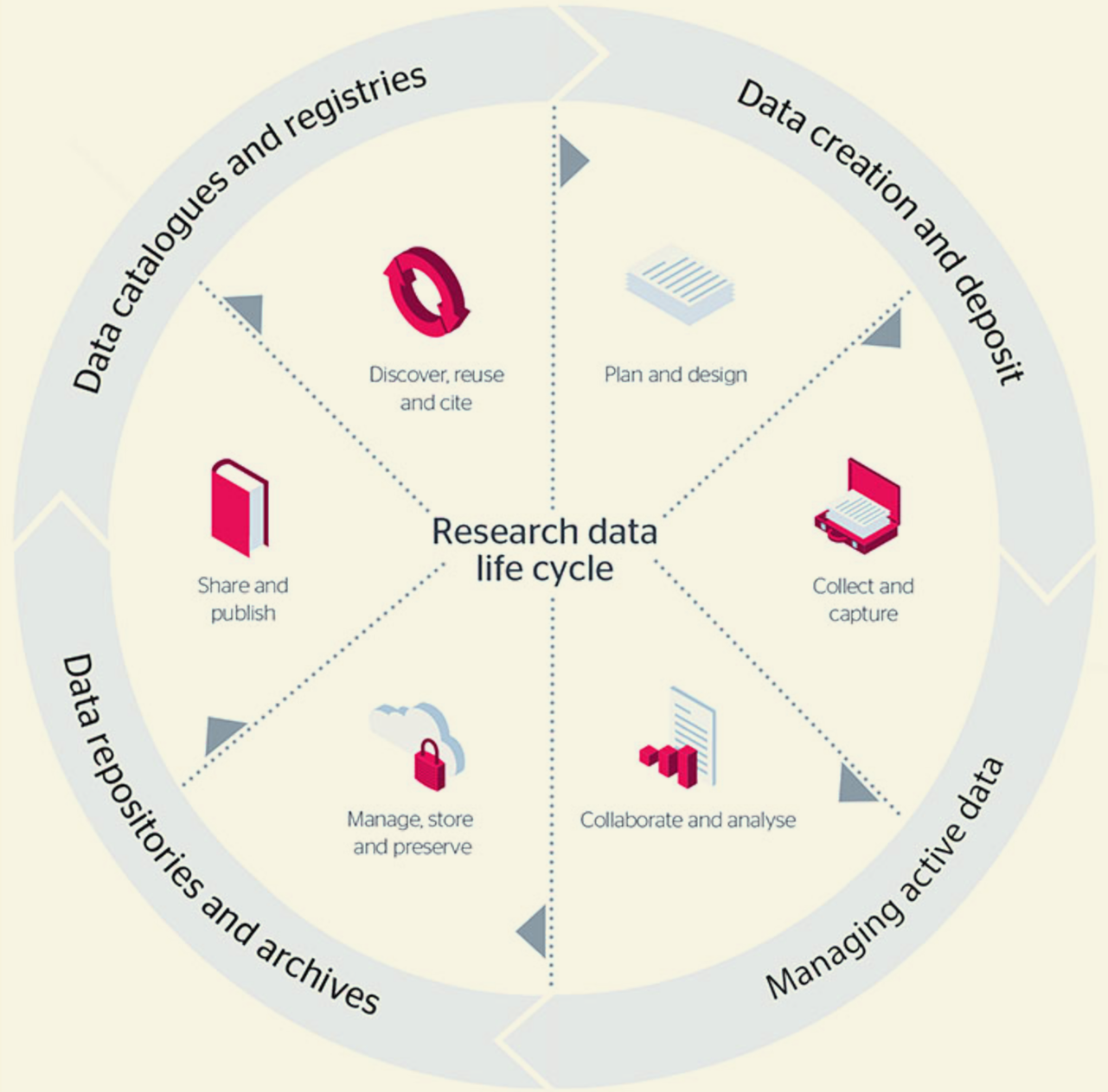
WHAT IS (RESEARCH) DATA MANAGEMENT?

- (Research) Data = every digital object involved in your project: code, software/tools, raw data, processed data, results, manuscripts ...
- ... needs to be properly managed - from its creation to its use, publication, sharing, archiving, re-use, or destruction... (keyword: FAIR data)



WHAT IS (RESEARCH) DATA MANAGEMENT?

- (Research) Data = every digital object involved in your project: code, software/tools, raw data, processed data, results, manuscripts ...
- ... needs to be properly managed - from its creation to its use, publication, sharing, archiving, re-use, or destruction... (keyword: FAIR data)



- Research data management is a key component for reproducibility, efficiency, and impact/reach of data analysis projects

WHY DATA MANAGEMENT?

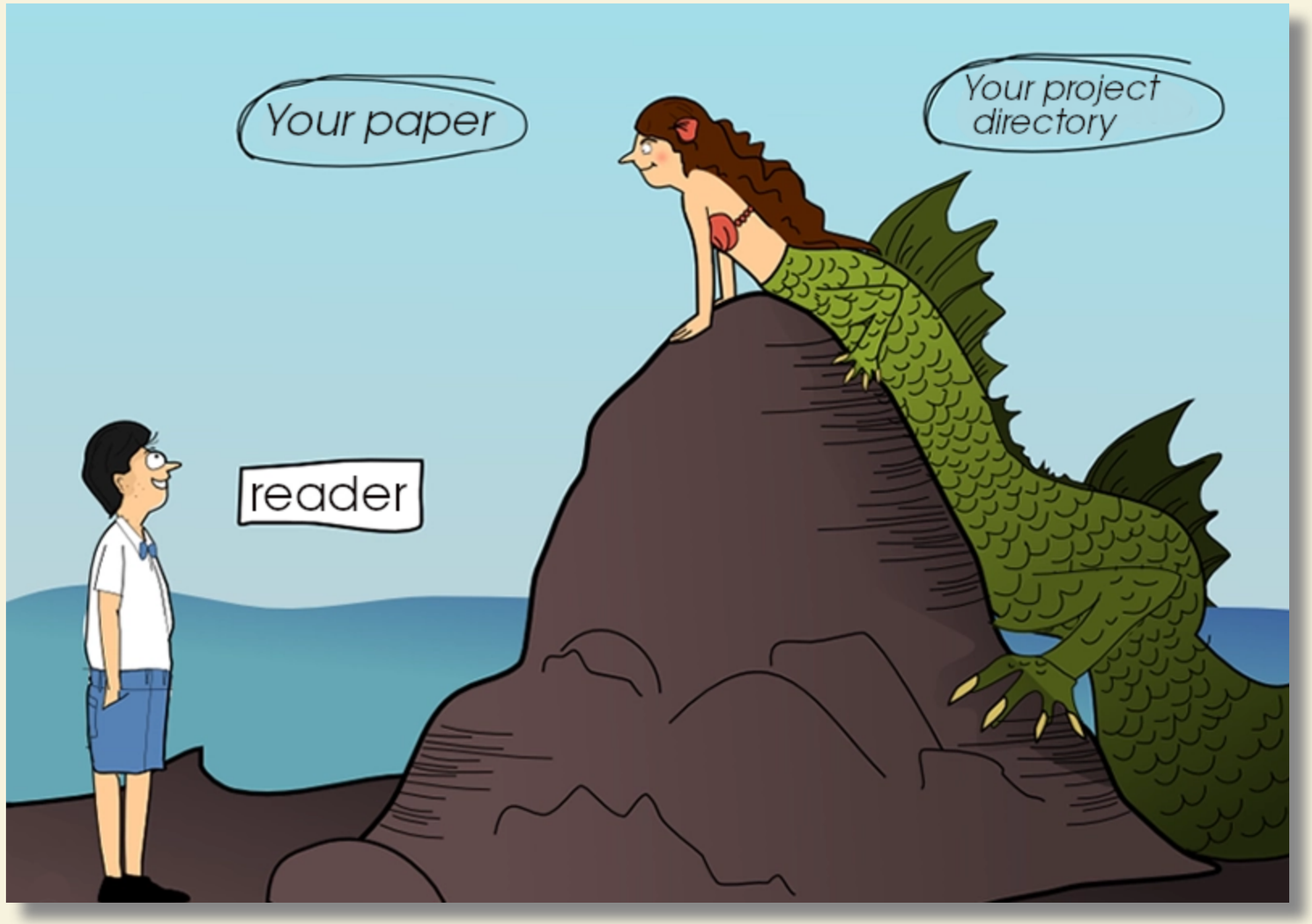


Image credit: adapted from https://dribbble.com/shots/3090048-Front-end-vs-Back-end

WHY DATA MANAGEMENT?

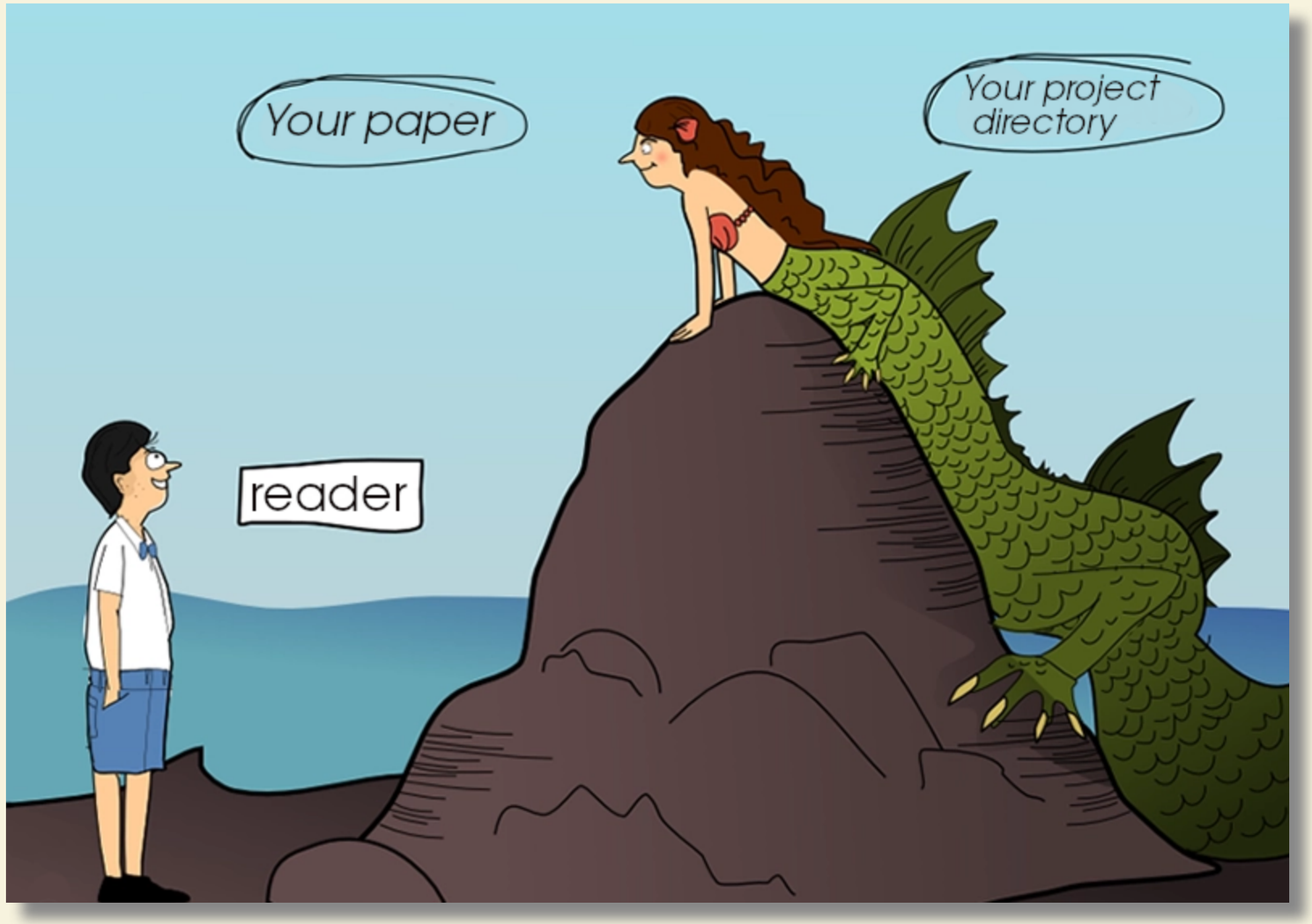


Image credit: adapted from https://dribbble.com/shots/3090048-Front-end-vs-Back-end

- Funders & publishers require it

WHY DATA MANAGEMENT?

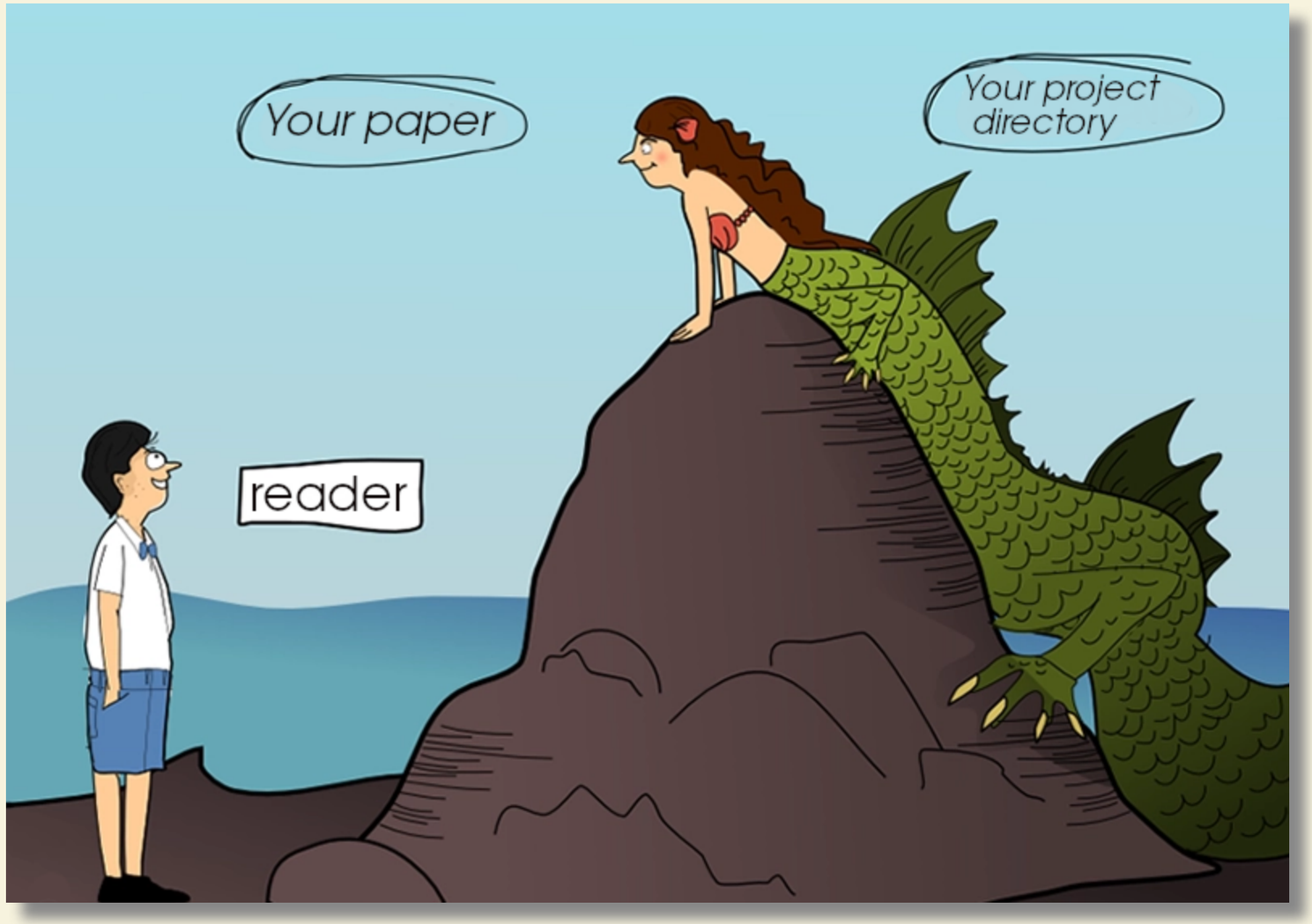


Image credit: adapted from https://dribbble.com/shots/3090048-Front-end-vs-Back-end

- Funders & publishers require it
- Scientific peers increasingly expect it

WHY DATA MANAGEMENT?

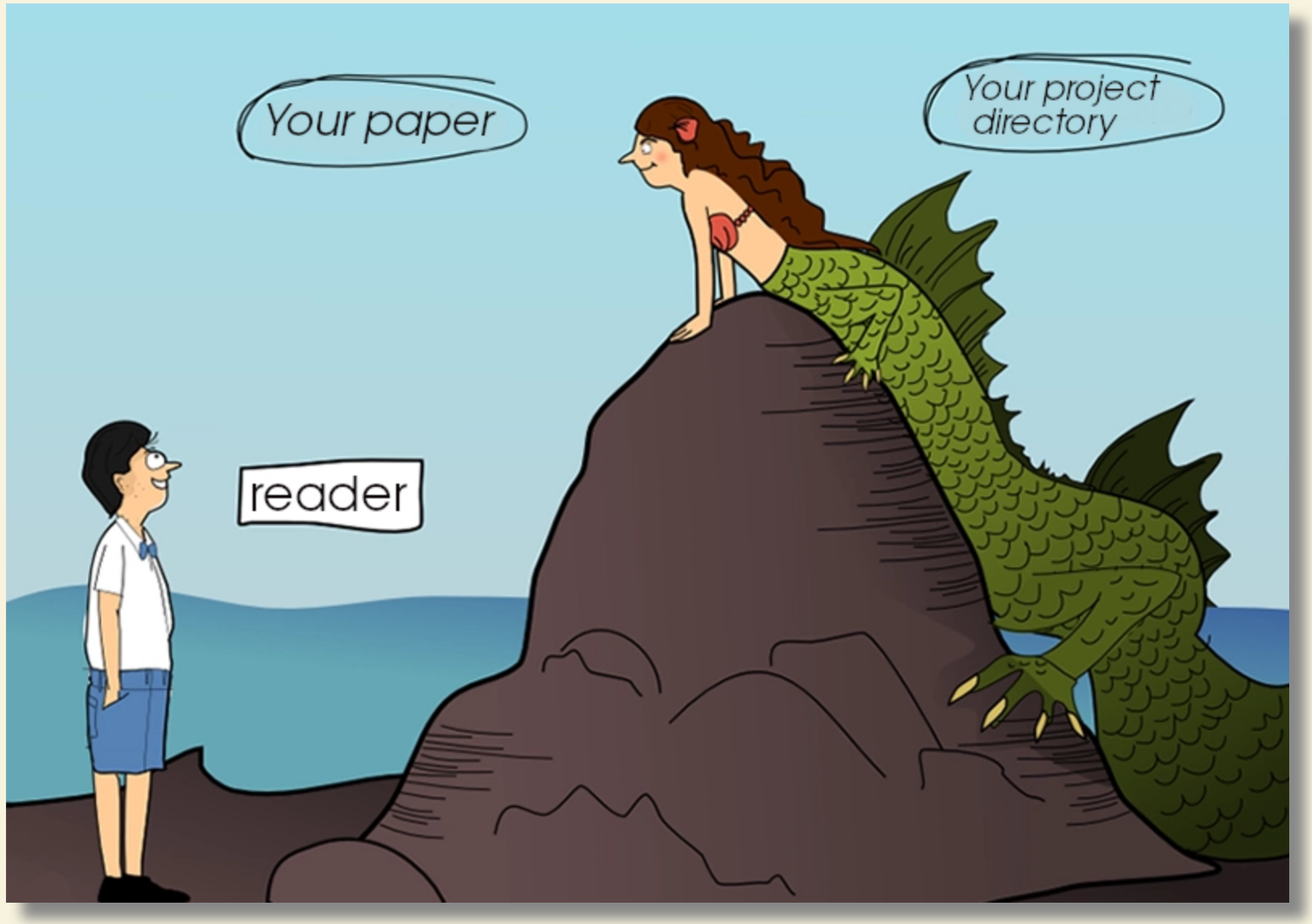


Image credit: adapted from https://dribbble.com/shots/3090048-Front-end-vs-Back-end

- Funders & publishers require it
- Scientific peers increasingly expect it
- The quality and efficiency of your work improves

THE MOST INTERESTING DATASETS OF OUR FIELD REQUIRE IT!

Image credit: https://infostory.files.wordpress.com/2013/03/big_data_cartoon.jpeg

- Exciting datasets (UKBiobank, HCP, ...) are orders of magnitudes larger then previous public datasets, and neither the computational infrastructure nor analysis workflows scale to these dataset sizes.

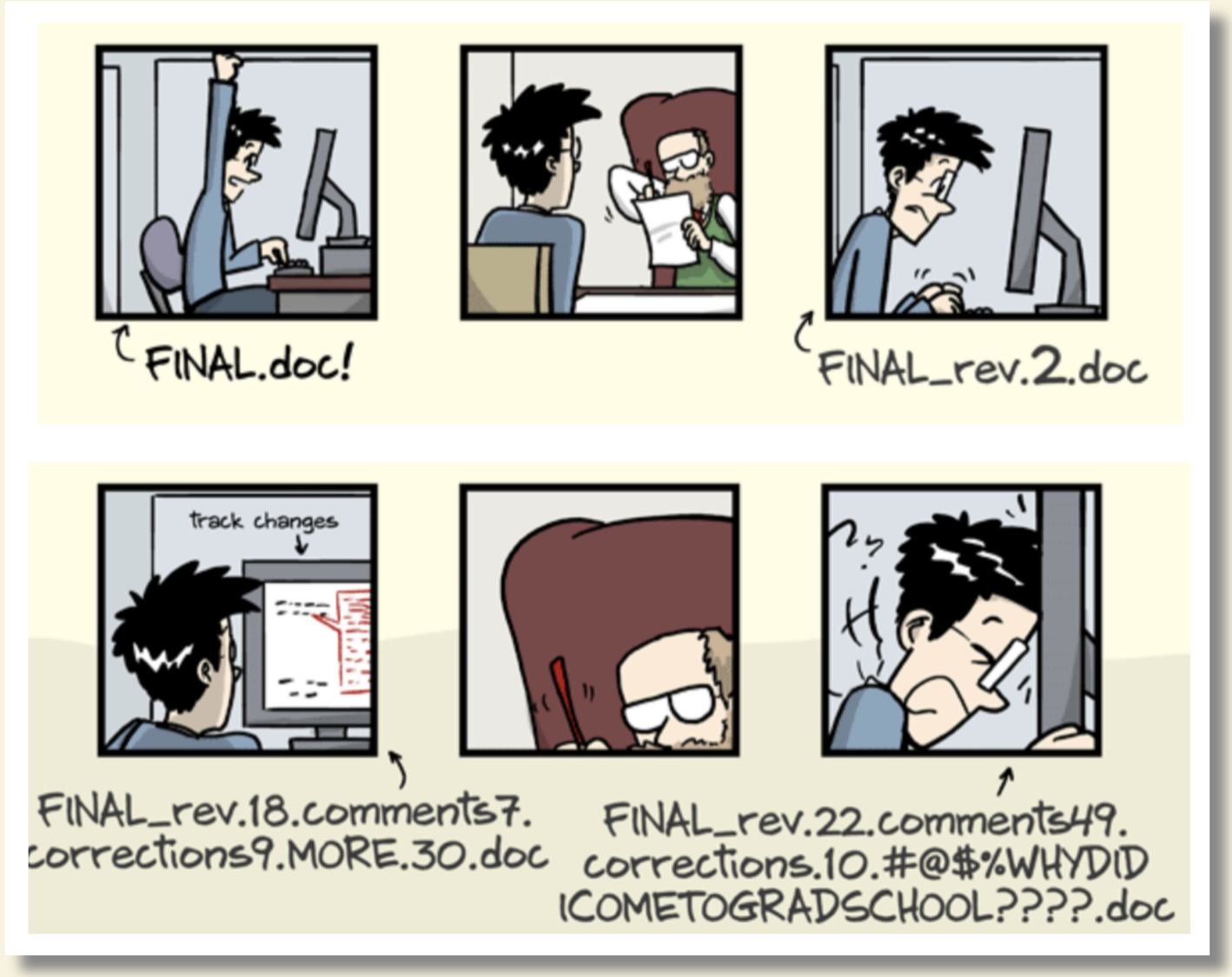
HOW IS (RESEARCH) DATA MANAGEMENT POSSIBLE?

There are tools and concepts that can help:

- **Version control** your data
- **Document** *everything*, ideally automatically (Provenance capture)

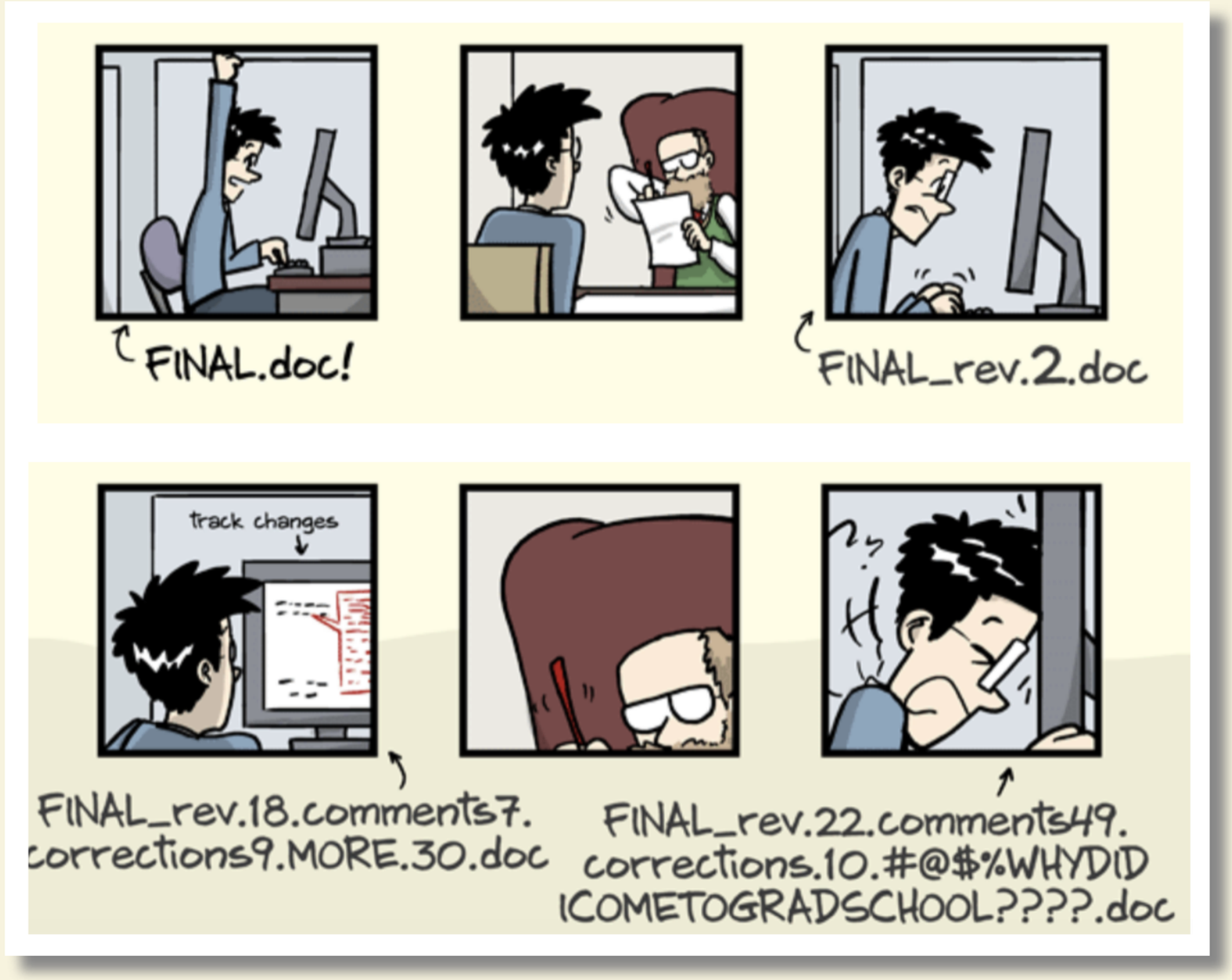
WHY VERSION CONTROL?

WHY VERSION CONTROL?



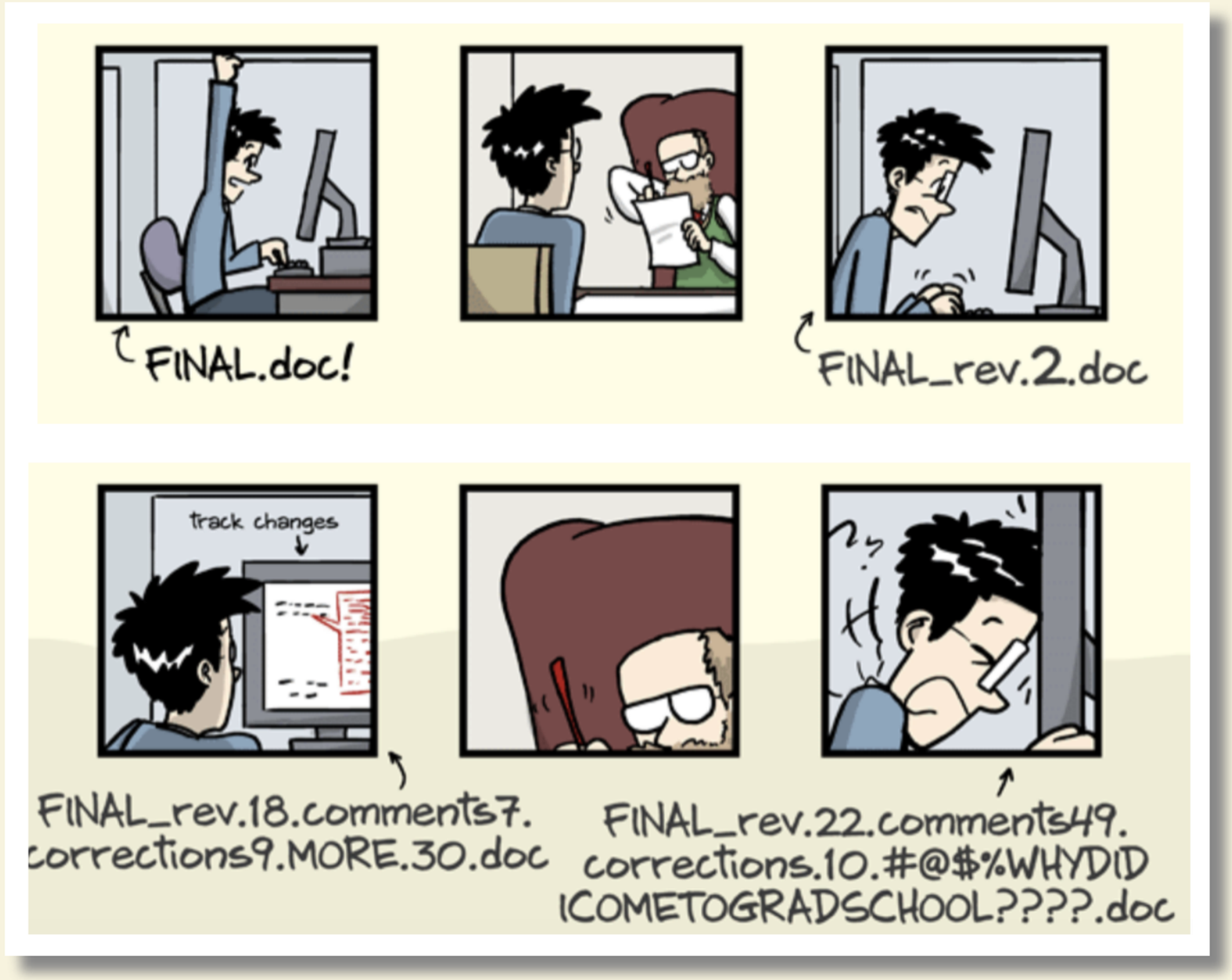
- keep things organized

WHY VERSION CONTROL?



- keep things organized
- keep track of changes

WHY VERSION CONTROL?



- keep things organized
- keep track of changes
- Mostly done for code, but **data** changes as well!

WHY PROVENANCE CAPTURE?

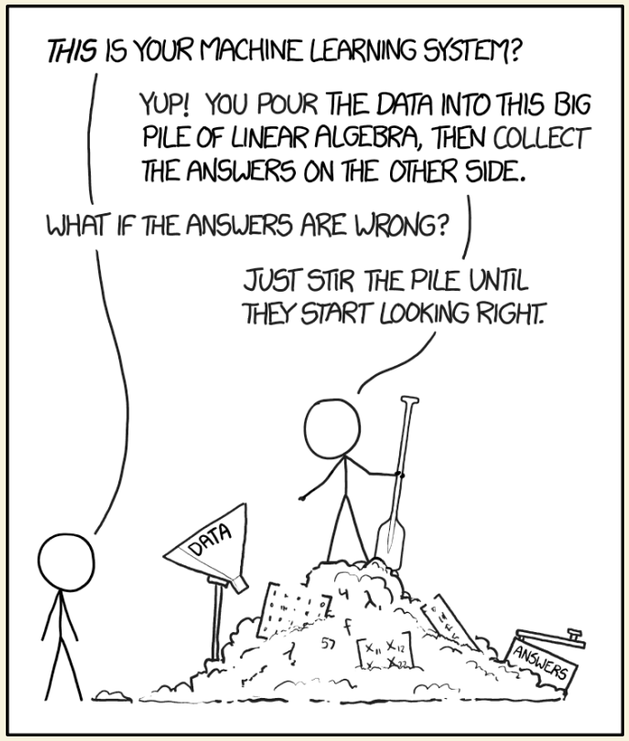


Image credit: https://xkcd.com/1838/

- Data Provenance: How did a digital object come to be in its present form?
- Many uses, but among others, it aids the reproducibility of science

WHY PROVENANCE CAPTURE?

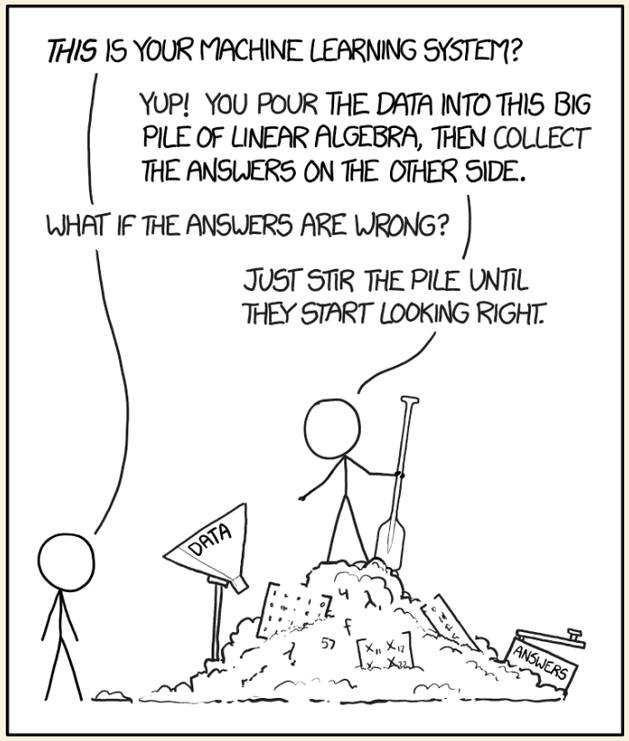


Image credit: https://xkcd.com/1838/

- Data Provenance: How did a digital object come to be in its present form?
- Many uses, but among others, it aids the reproducibility of science
    - Who generated an output and when?

WHY PROVENANCE CAPTURE?

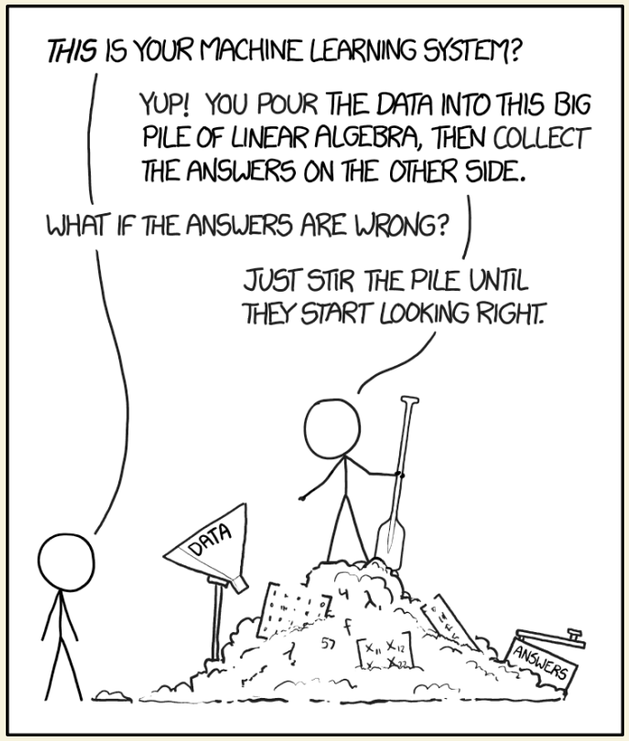


Image credit: https://xkcd.com/1838/

- Data Provenance: How did a digital object come to be in its present form?
- Many uses, but among others, it aids the reproducibility of science
    - Who generated an output and when?
    - Which (version of which) data was used, and where does this input data come from?

WHY PROVENANCE CAPTURE?

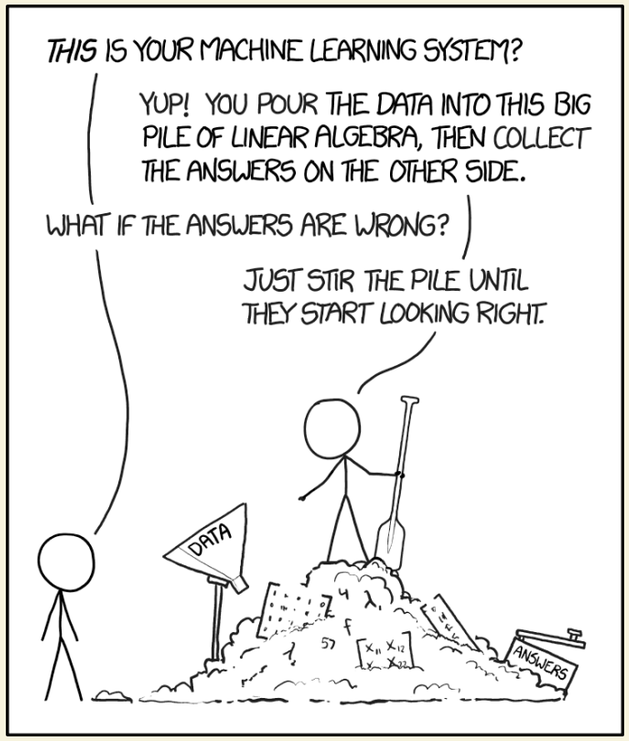


Image credit: https://xkcd.com/1838/

- Data Provenance: How did a digital object come to be in its present form?
- Many uses, but among others, it aids the reproducibility of science
    - Who generated an output and when?
    - Which (version of which) data was used, and where does this input data come from?
    - Which scripts with which parameters created which results, and how?

WHY PROVENANCE CAPTURE?

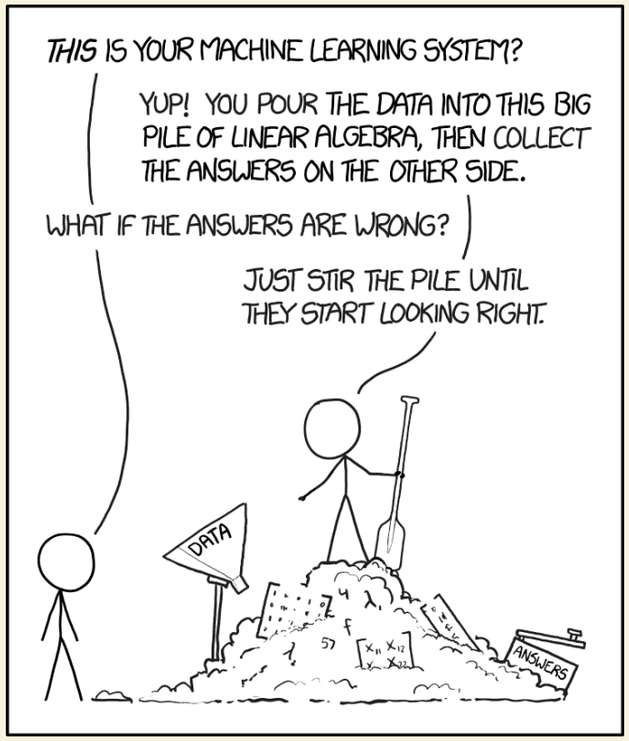


Image credit: https://xkcd.com/1838/

- Data Provenance: How did a digital object come to be in its present form?
- Many uses, but among others, it aids the reproducibility of science
    - Who generated an output and when?
    - Which (version of which) data was used, and where does this input data come from?
    - Which scripts with which parameters created which results, and how?
    - What is the software (libraries, environment) used for a computation and its dependencies to data or code?

DATALAD CAN HELP WITH DATA MANAGEMENT

DATALAD CAN HELP WITH DATA MANAGEMENT

What is it?

DATALAD CAN HELP WITH DATA MANAGEMENT

What is it?
Why should I use it?

DATALAD CAN HELP WITH DATA MANAGEMENT

What is it?
Why should I use it?
How can I use it?

DataLad

DataLad is a data management multitool.

Let's see it in action

REPRODUCIBLE PAPER - A MAGIC TRICK?

If curious, you can read up all the details and a step-by-step instruction here.

DataLad IN BRIEF

- A command-line tool, available for all major operating systems (Linux, macOS/OSX, Windows)
- Build on top of Git and Git-annex
- **Allows...**
    ... version-controlling arbitrarily large content,
    ... easily sharing and obtaining data (note: no data hosting!),
    ... (computationally) reproducible data analysis,
    ... and *much* more
- Completely domain-agnostic

DataLad IN BRIEF

- A command-line tool, available for all major operating systems (Linux, macOS/OSX, Windows)
- Build on top of Git and Git-annex
- **Allows...**
  - ... version-controlling arbitrarily large content,
  - ... easily sharing and obtaining data (note: no data hosting!),
  - ... (computationally) reproducible data analysis,
  - ... and *much* more
- Completely domain-agnostic

  Today: Basic concepts and commands.

DataLad IN BRIEF

- A command-line tool, available for all major operating systems (Linux, macOS/OSX, Windows)
- Build on top of Git and Git-annex
- **Allows...**
  - ... version-controlling arbitrarily large content,
  - ... easily sharing and obtaining data (note: no data hosting!),
  - ... (computationally) reproducible data analysis,
  - ... and *much* more
- Completely domain-agnostic

Today: Basic concepts and commands.

➲ **For more:** Read the DataLad Handbook

DATALAD DATASETS

- DataLad's core data structure

DATALAD DATASETS

- DataLad's core data structure
    - Dataset = A directory managed by DataLad

DATALAD DATASETS

- DataLad's core data structure
    - Dataset = A directory managed by DataLad
    - Any directory of your computer can be managed by DataLad.

DATALAD DATASETS

- DataLad's core data structure
    - Dataset = A directory managed by DataLad
    - Any directory of your computer can be managed by DataLad.
    - Datasets can be *created* (from scratch) or *installed*

DATALAD DATASETS

- DataLad's core data structure
  - Dataset = A directory managed by DataLad
  - Any directory of your computer can be managed by DataLad.
  - Datasets can be *created* (from scratch) or *installed*
  - Datasets can be nested: *linked subdirectories*

EXPERIENCE A DATALAD DATASET

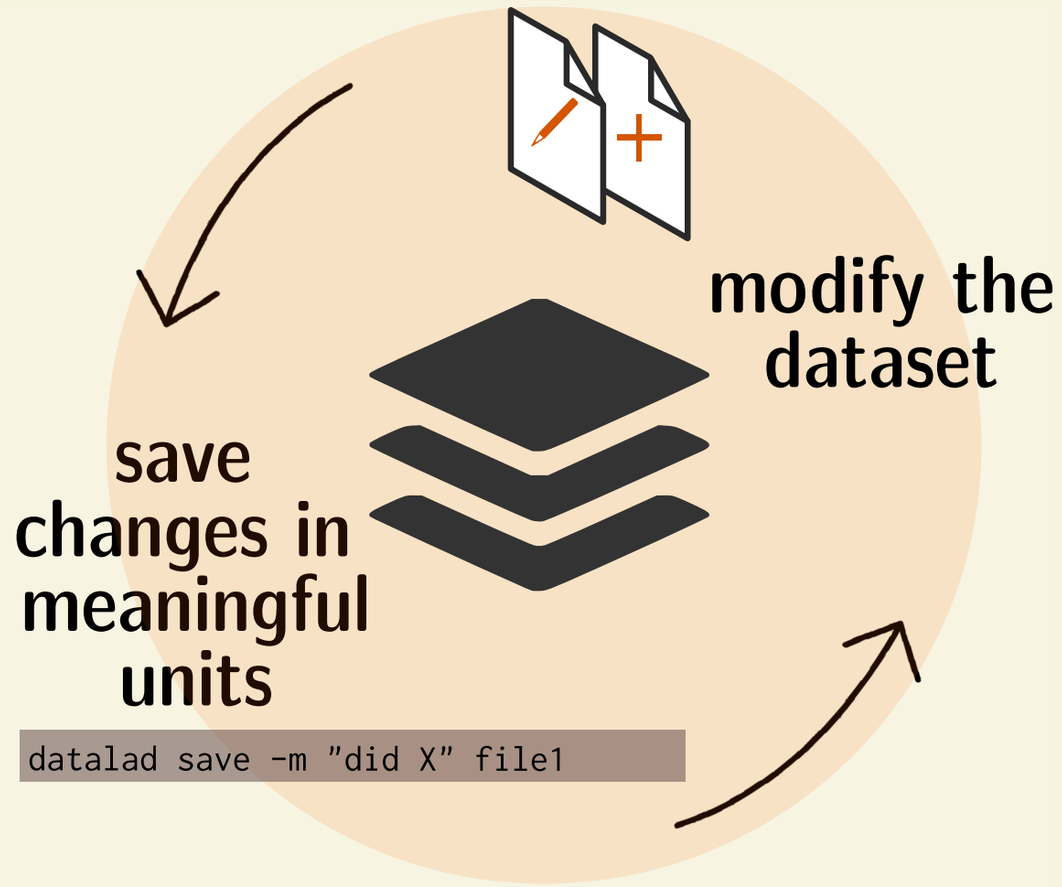
Code to follow along:
http://handbook.datalad.org/en/latest/code_from_chapters/01_dataset_basics_code.html

LOCAL VERSION CONTROL

Procedurally, version control is easy with DataLad!

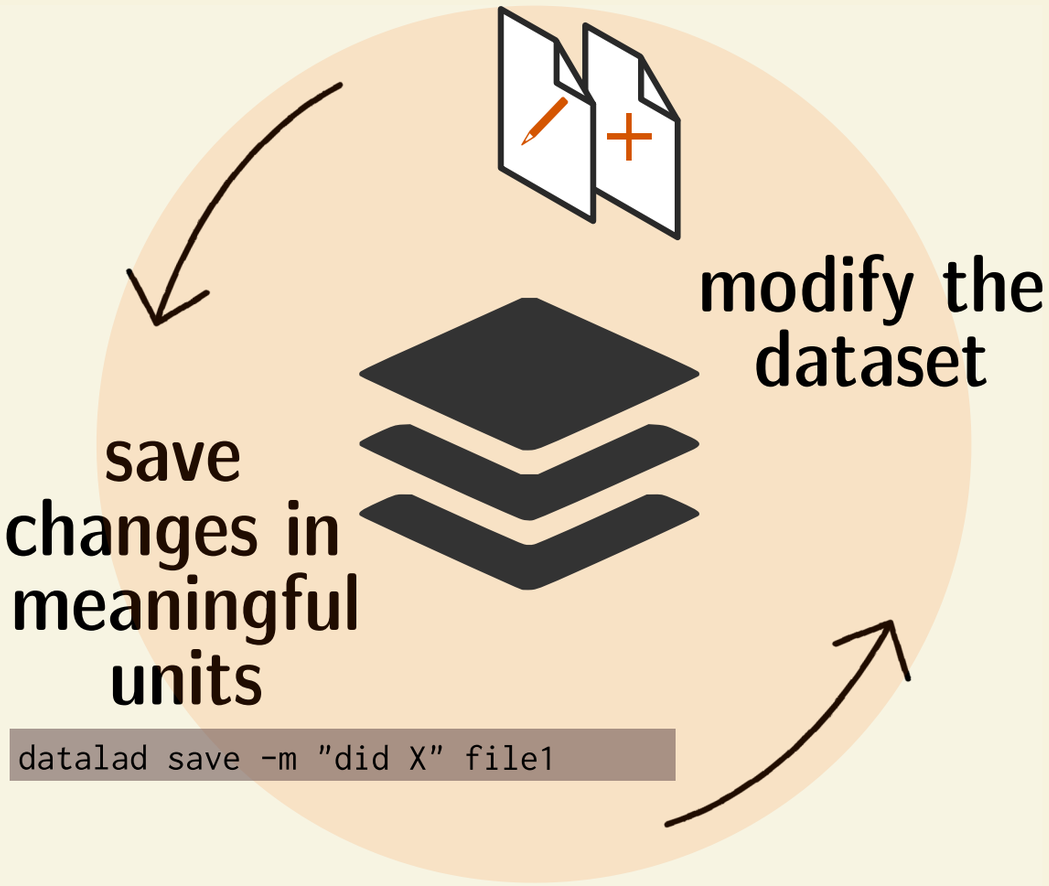
LOCAL VERSION CONTROL

Procedurally, version control is easy with DataLad!



LOCAL VERSION CONTROL

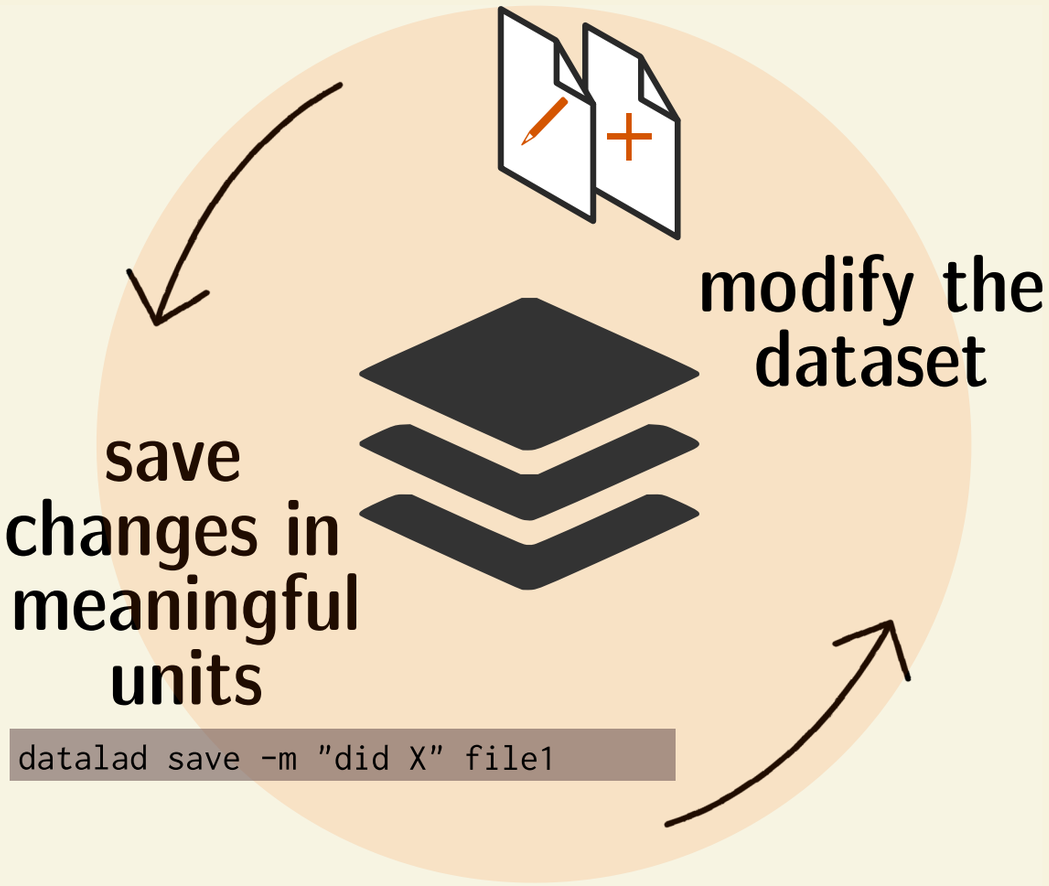
Procedurally, version control is easy with DataLad!



Advice:

LOCAL VERSION CONTROL

Procedurally, version control is easy with DataLad!

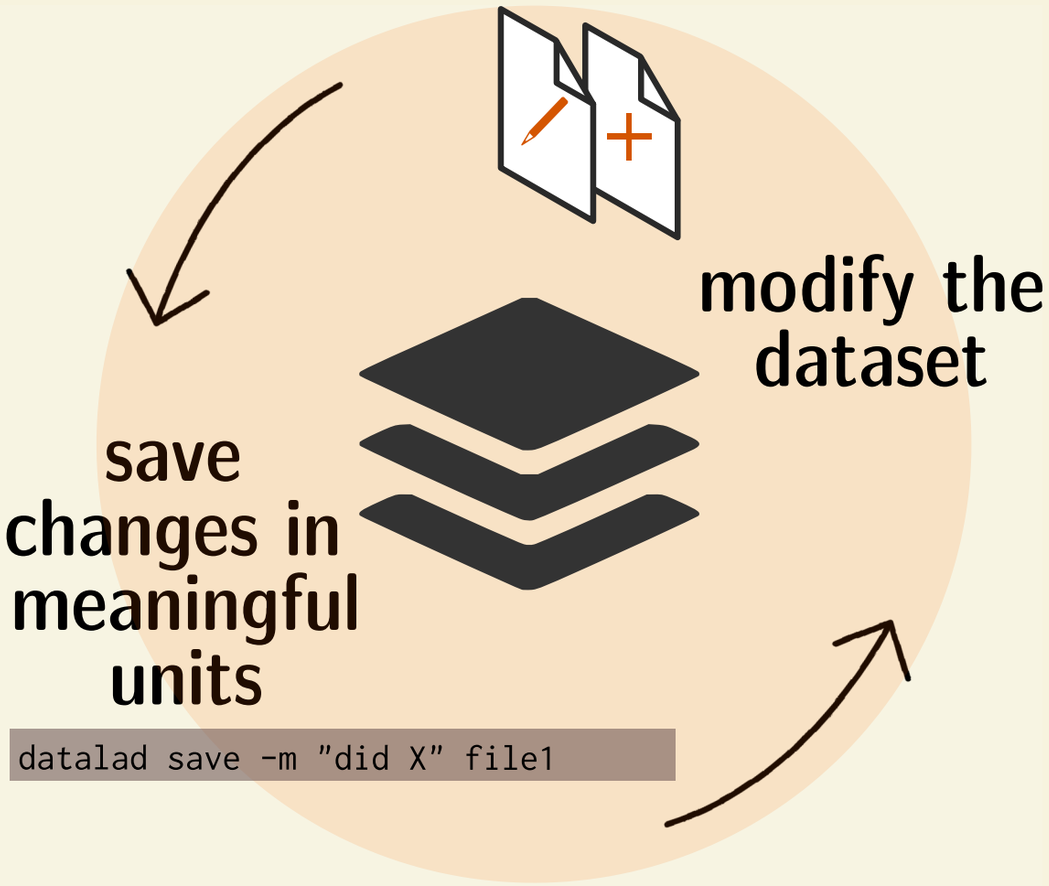


Advice:

- Save *meaningful* units of change

LOCAL VERSION CONTROL

Procedurally, version control is easy with DataLad!



Advice:

- Save *meaningful* units of change
- Attach helpful commit messages

SUMMARY - LOCAL VERSION CONTROL

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

A dataset has a *history* to track files and their modifications.

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

A dataset has a *history* to track files and their modifications.

Explore it with Git (**git log**) or external tools (e.g., **tig**).

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

A dataset has a *history* to track files and their modifications.

Explore it with Git (**git log**) or external tools (e.g., **tig**).

`datalad save` records the dataset or file state to the history.

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

A dataset has a *history* to track files and their modifications.

Explore it with Git (**git log**) or external tools (e.g., **tig**).

`datalad save` records the dataset or file state to the history.

Concise **commit messages** should summarize the change for future you and others.

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

A dataset has a *history* to track files and their modifications.

Explore it with Git (**git log**) or external tools (e.g., **tig**).

`datalad save` records the dataset or file state to the history.

Concise **commit messages** should summarize the change for future you and others.

`datalad download-url` obtains web content and records its origin.

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

A dataset has a *history* to track files and their modifications.

Explore it with Git (**git log**) or external tools (e.g., **tig**).

`datalad save` records the dataset or file state to the history.

Concise **commit messages** should summarize the change for future you and others.

`datalad download-url` obtains web content and records its origin.

It even takes care of saving the change.

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

A dataset has a *history* to track files and their modifications.

Explore it with Git (**git log**) or external tools (e.g., **tig**).

`datalad save` records the dataset or file state to the history.

Concise **commit messages** should summarize the change for future you and others.

`datalad download-url` obtains web content and records its origin.

It even takes care of saving the change.

`datalad status` reports the current state of the dataset.

SUMMARY - LOCAL VERSION CONTROL

`datalad create` creates an empty dataset.

Configurations (**-c yoda**, **-c text2git**) are useful.

A dataset has a *history* to track files and their modifications.

Explore it with Git (**git log**) or external tools (e.g., **tig**).

`datalad save` records the dataset or file state to the history.

Concise **commit messages** should summarize the change for future you and others.

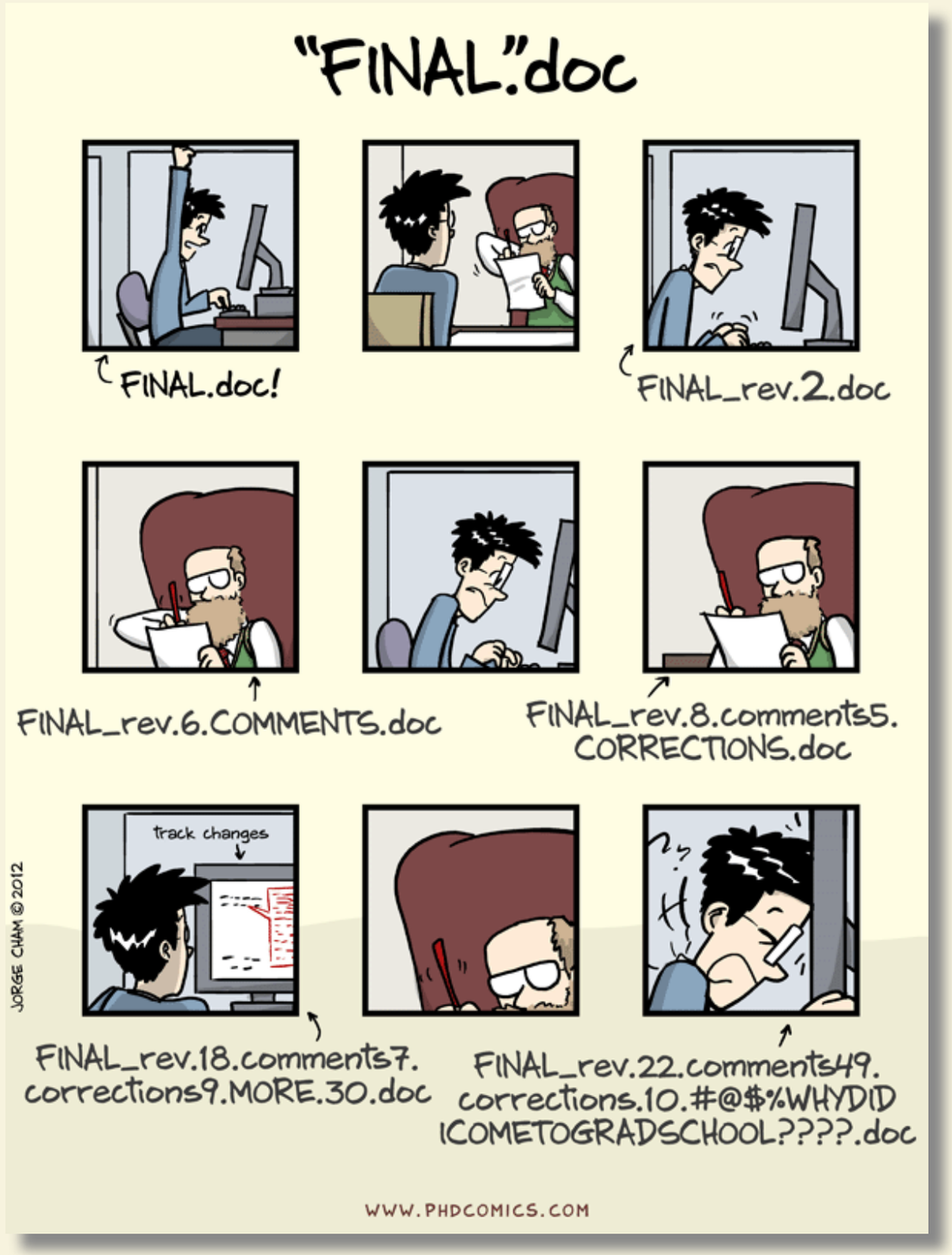
`datalad download-url` obtains web content and records its origin.

It even takes care of saving the change.

`datalad status` reports the current state of the dataset.

A clean dataset status is good practice.

FROM HERE



FROM HERE

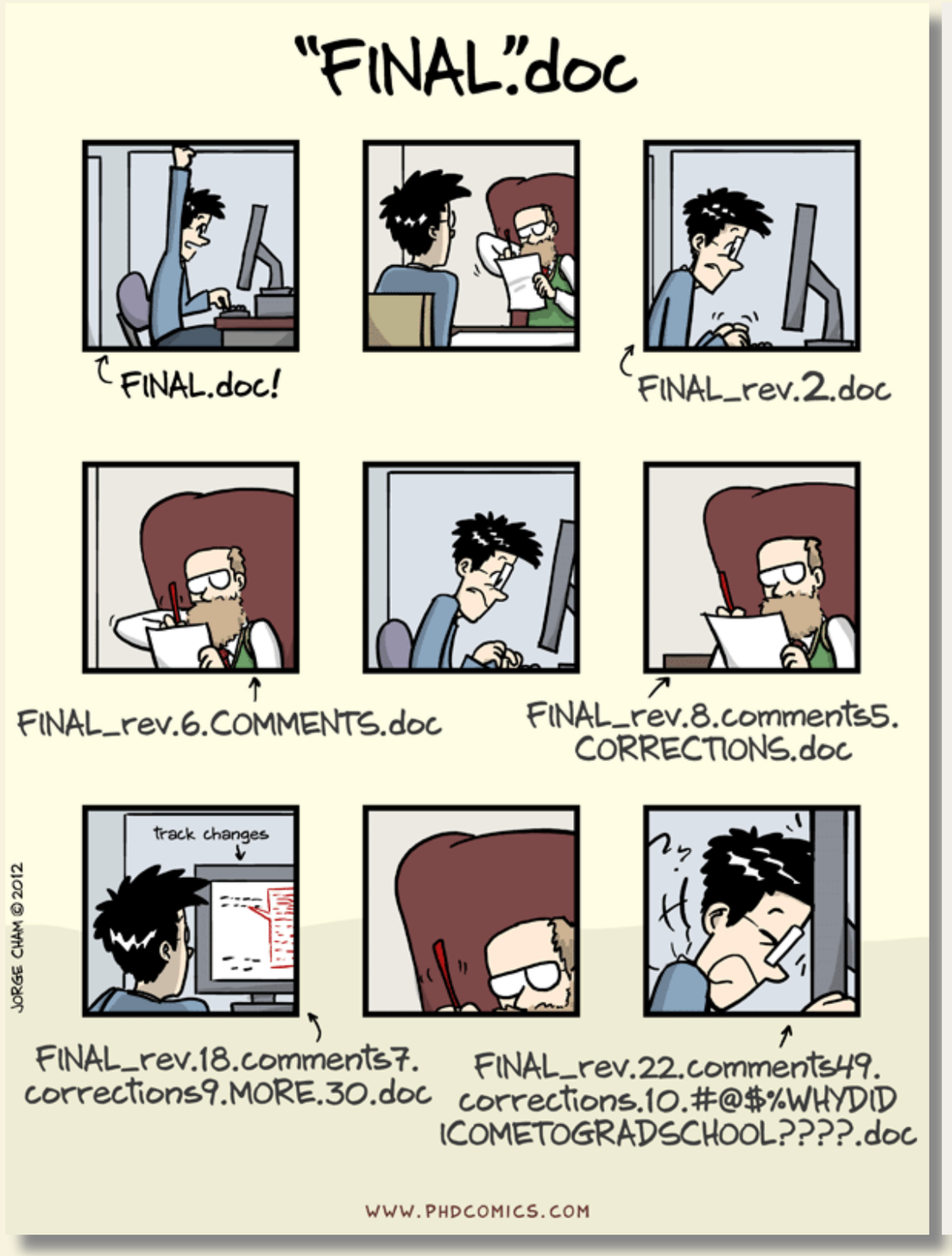


Image credit: www.phdcomics.com www.linode.com

TO THIS:

FROM HERE

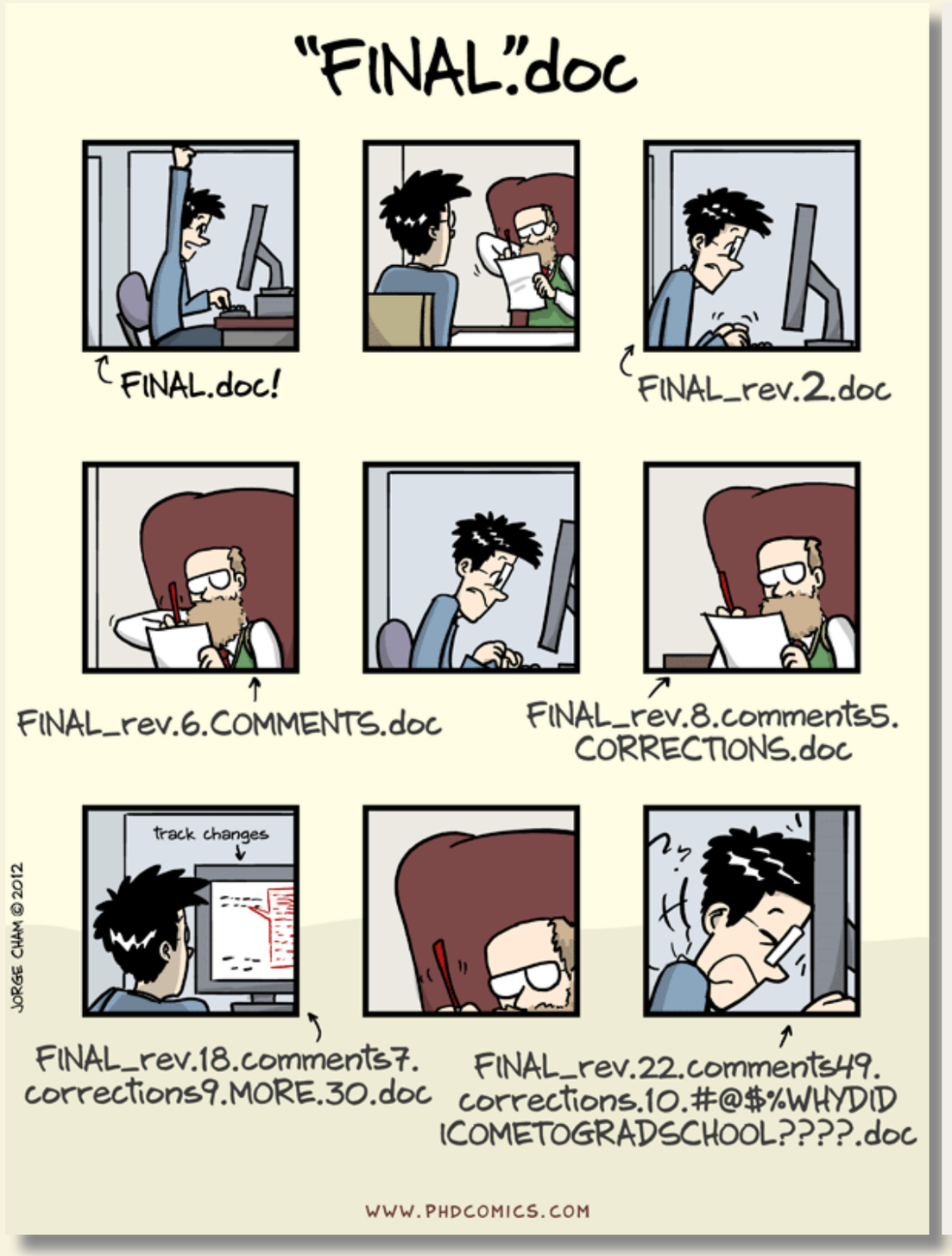


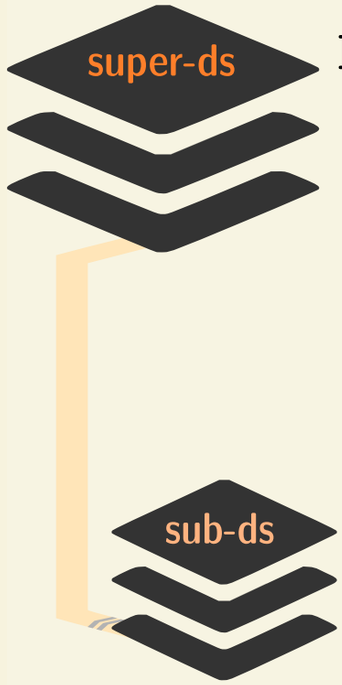
Image credit: www.phdcomics.com www.linode.com

TO THIS:

BUT: Version control is only one aspect of data management

CONSUMING DATASETS

CONSUMING DATASETS



DataLad-101/
  books/
    byte-of-python.pdf
    progit.pdf
    TLCL.pdf
  recordings/
    longnow/
      Long_Now___Conv[...]/
        ...
      Long_Now___Seminars[...]/
        2003_12_13[...]
        2003_11_15[...]
        ...
  notes.txt

Dataset structure is fully flexible to be able to accommodate domain standards or personal preferences.

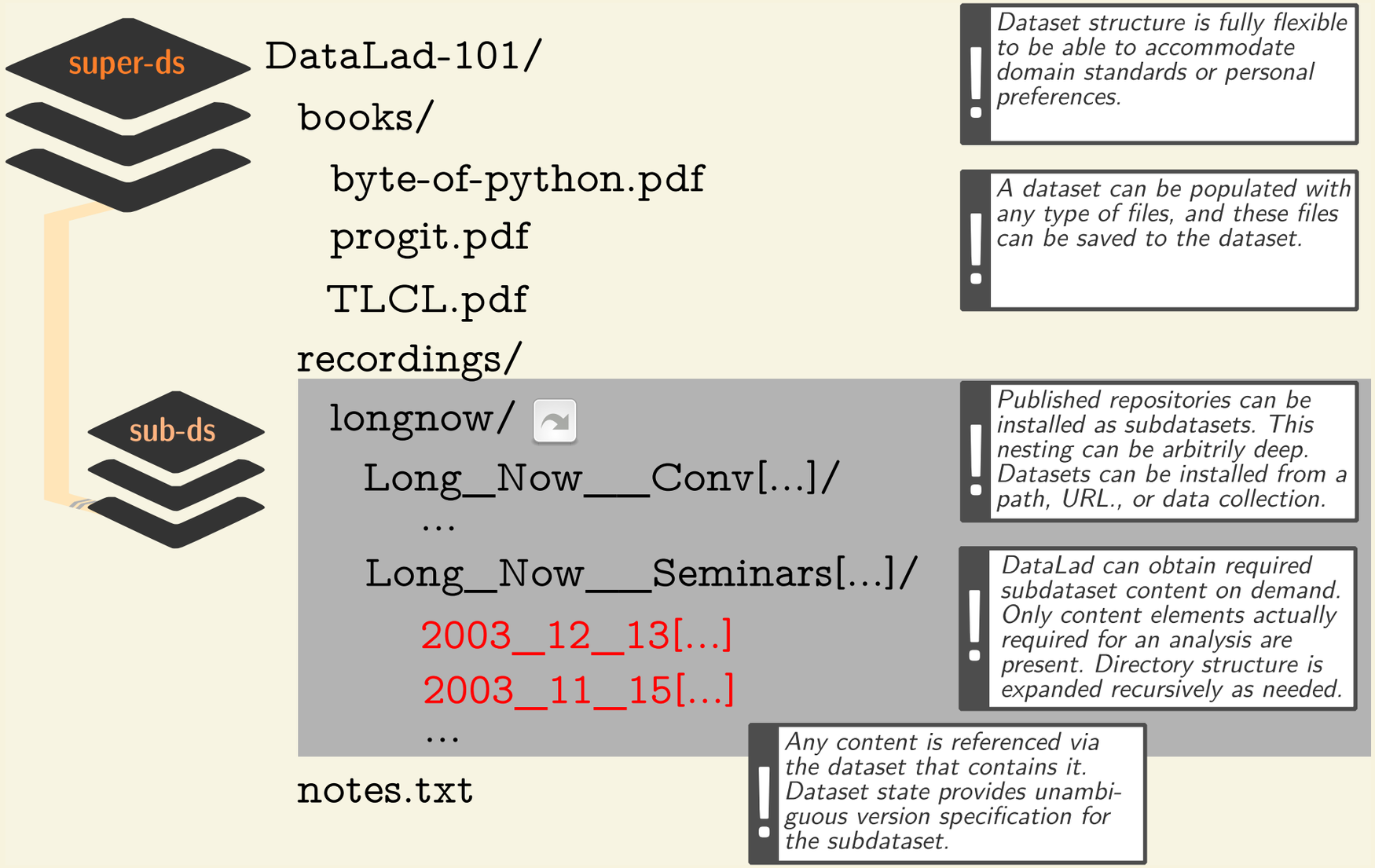
A dataset can be populated with any type of files, and these files can be saved to the dataset.

Published repositories can be installed as subdatasets. This nesting can be arbitrily deep. Datasets can be installed from a path, URL., or data collection.

DataLad can obtain required subdataset content on demand. Only content elements actually required for an analysis are present. Directory structure is expanded recursively as needed.

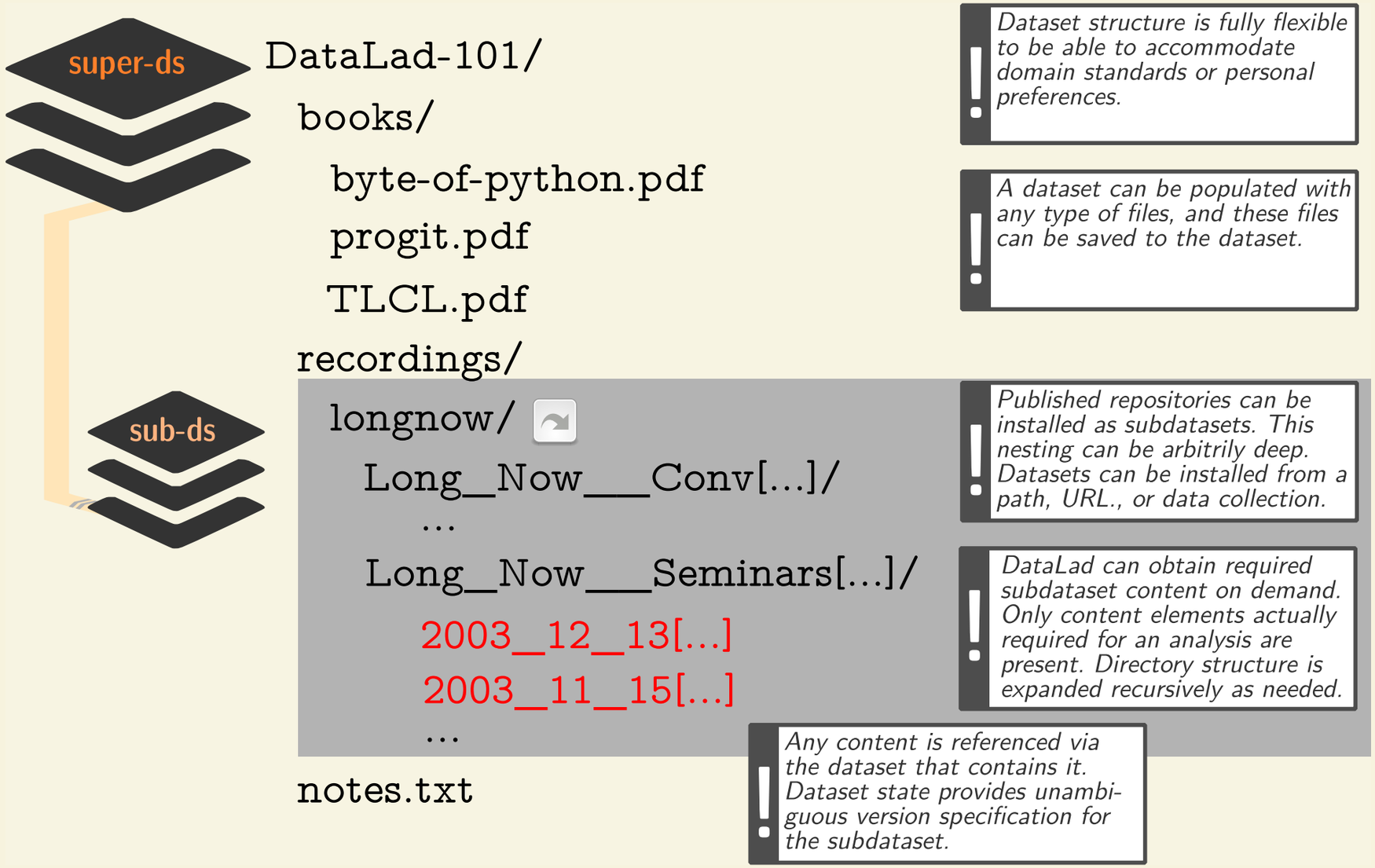
Any content is referenced via the dataset that contains it. Dataset state provides unambiguous version specification for the subdataset.

CONSUMING DATASETS



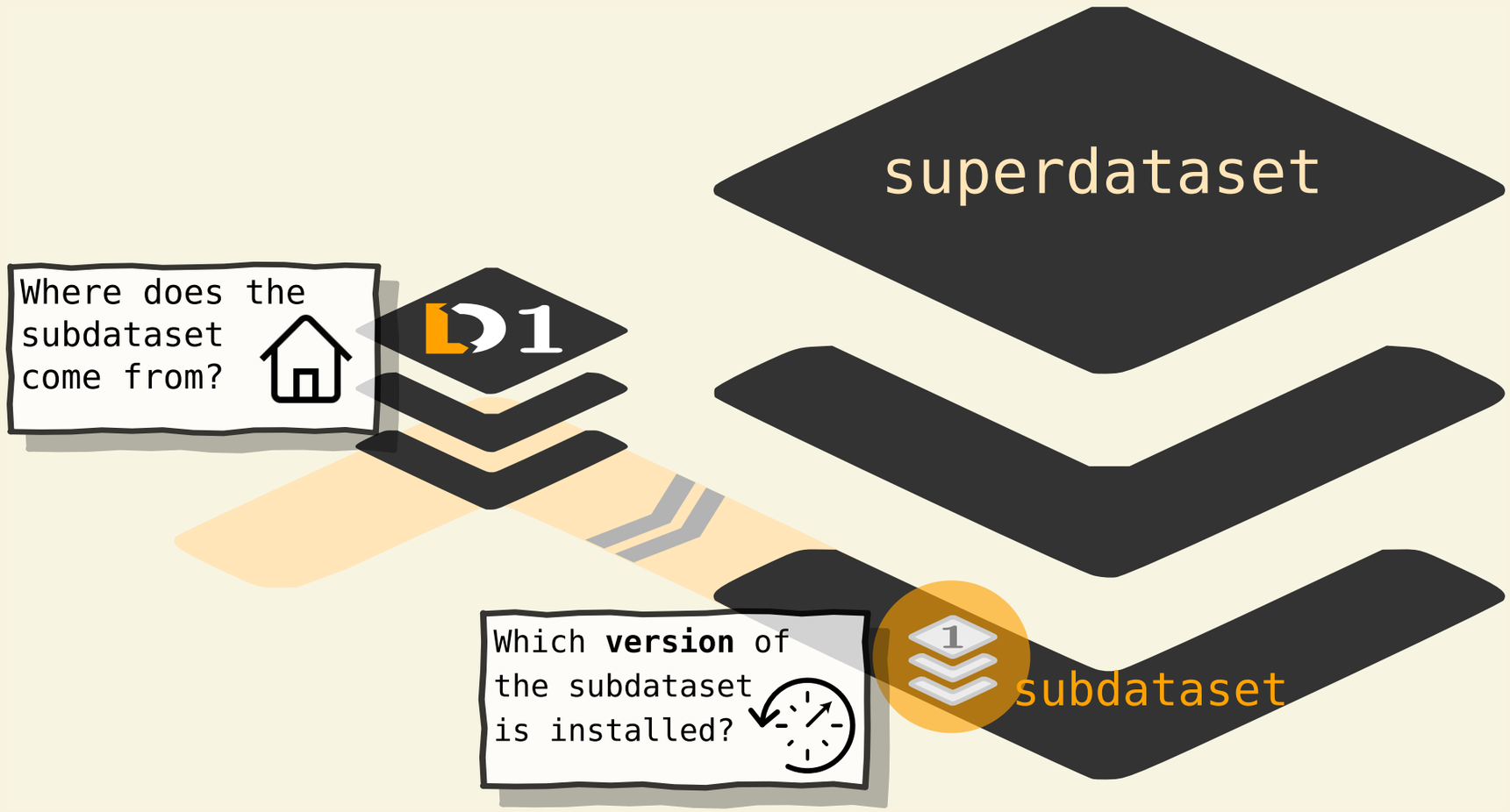
- Datasets are light-weight: Upon installation, only small files and meta data about file availability are retrieved.

CONSUMING DATASETS



- Datasets are light-weight: Upon installation, only small files and meta data about file availability are retrieved.
- Content can be obtained on demand via `datalad get`.

DATASET NESTING



SUMMARY - DATASET CONSUMPTION & NESTING

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed “on its own”: Specify the source (url, path, …) of the dataset, and an optional **path** for it to be installed to.

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed “on its own”: Specify the source (url, path, ...) of the dataset, and an optional **path** for it to be installed to.

Datasets can be installed as subdatasets within an existing dataset.

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed “on its own”: Specify the source (url, path, ...) of the dataset, and an optional **path** for it to be installed to.

Datasets can be installed as subdatasets within an existing dataset.

The **--dataset/-d** option needs a path to the root of the superdataset.

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed “on its own”: Specify the source (url, path, ...) of the dataset, and an optional **path** for it to be installed to.

Datasets can be installed as subdatasets within an existing dataset.

The **--dataset/-d** option needs a path to the root of the superdataset.

Only small files and metadata about file availability are present locally after an install.

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed "on its own": Specify the source (url, path, ...) of the dataset, and an optional **path** for it to be installed to.

Datasets can be installed as subdatasets within an existing dataset.

The **--dataset/-d** option needs a path to the root of the superdataset.

Only small files and metadata about file availability are present locally after an install.

To retrieve actual file content of larger files, `datalad get` downloads large file content on demand.

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed "on its own": Specify the source (url, path, ...) of the dataset, and an optional **path** for it to be installed to.

Datasets can be installed as subdatasets within an existing dataset.

The **--dataset/-d** option needs a path to the root of the superdataset.

Only small files and metadata about file availability are present locally after an install.

To retrieve actual file content of larger files, `datalad get` downloads large file content on demand.

`datalad status` can report on total and retrieved repository size

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed "on its own": Specify the source (url, path, ...) of the dataset, and an optional **path** for it to be installed to.

Datasets can be installed as subdatasets within an existing dataset.

The **--dataset/-d** option needs a path to the root of the superdataset.

Only small files and metadata about file availability are present locally after an install.

To retrieve actual file content of larger files, `datalad get` downloads large file content on demand.

`datalad status` can report on total and retrieved repository size

using `--annex` and `--annex all` options.

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed "on its own": Specify the source (url, path, ...) of the dataset, and an optional **path** for it to be installed to.

Datasets can be installed as subdatasets within an existing dataset.

The **--dataset/-d** option needs a path to the root of the superdataset.

Only small files and metadata about file availability are present locally after an install.

To retrieve actual file content of larger files, `datalad get` downloads large file content on demand.

`datalad status` can report on total and retrieved repository size

using `--annex` and `--annex all` options.

Datasets preserve their history.

SUMMARY - DATASET CONSUMPTION & NESTING

`datalad clone` installs a dataset.

It can be installed "on its own": Specify the source (url, path, ...) of the dataset, and an optional **path** for it to be installed to.

Datasets can be installed as subdatasets within an existing dataset.

The **--dataset/-d** option needs a path to the root of the superdataset.

Only small files and metadata about file availability are present locally after an install.

To retrieve actual file content of larger files, `datalad get` downloads large file content on demand.

`datalad status` can report on total and retrieved repository size

using `--annex` and `--annex all` options.

Datasets preserve their history.

The superdataset records only the *version state* of the subdataset.

REPRODUCIBLE DATA ANALYSIS

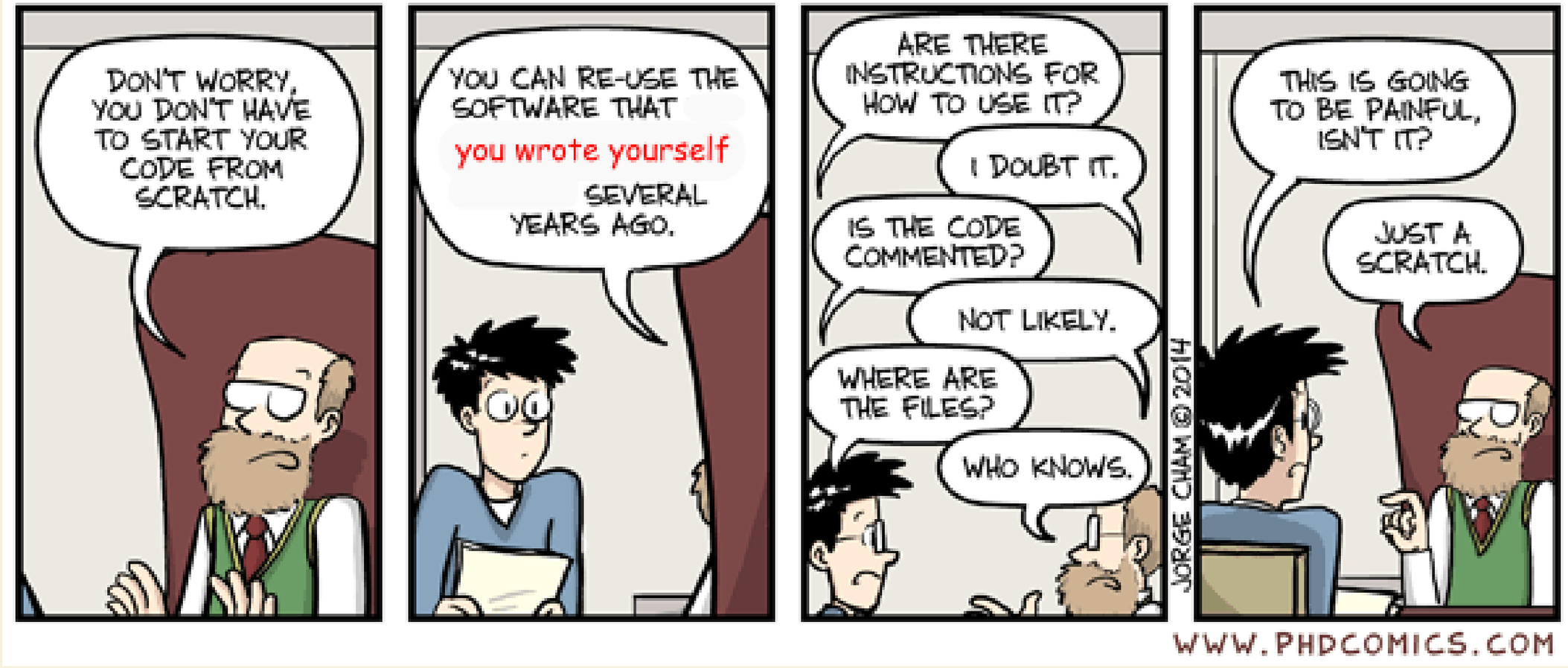


Image credit: Full comic at http://phdcomics.com/comics.php?f=1979

REPRODUCIBLE DATA ANALYSIS

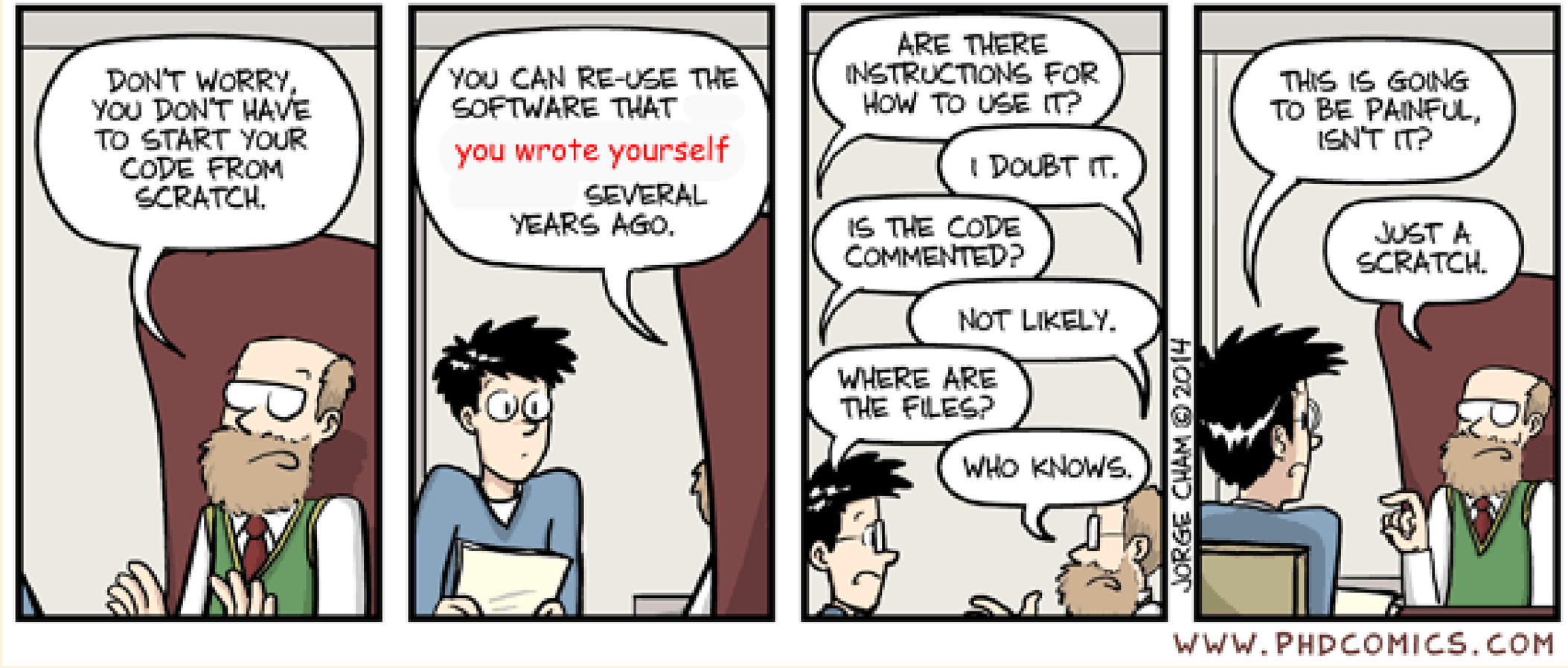


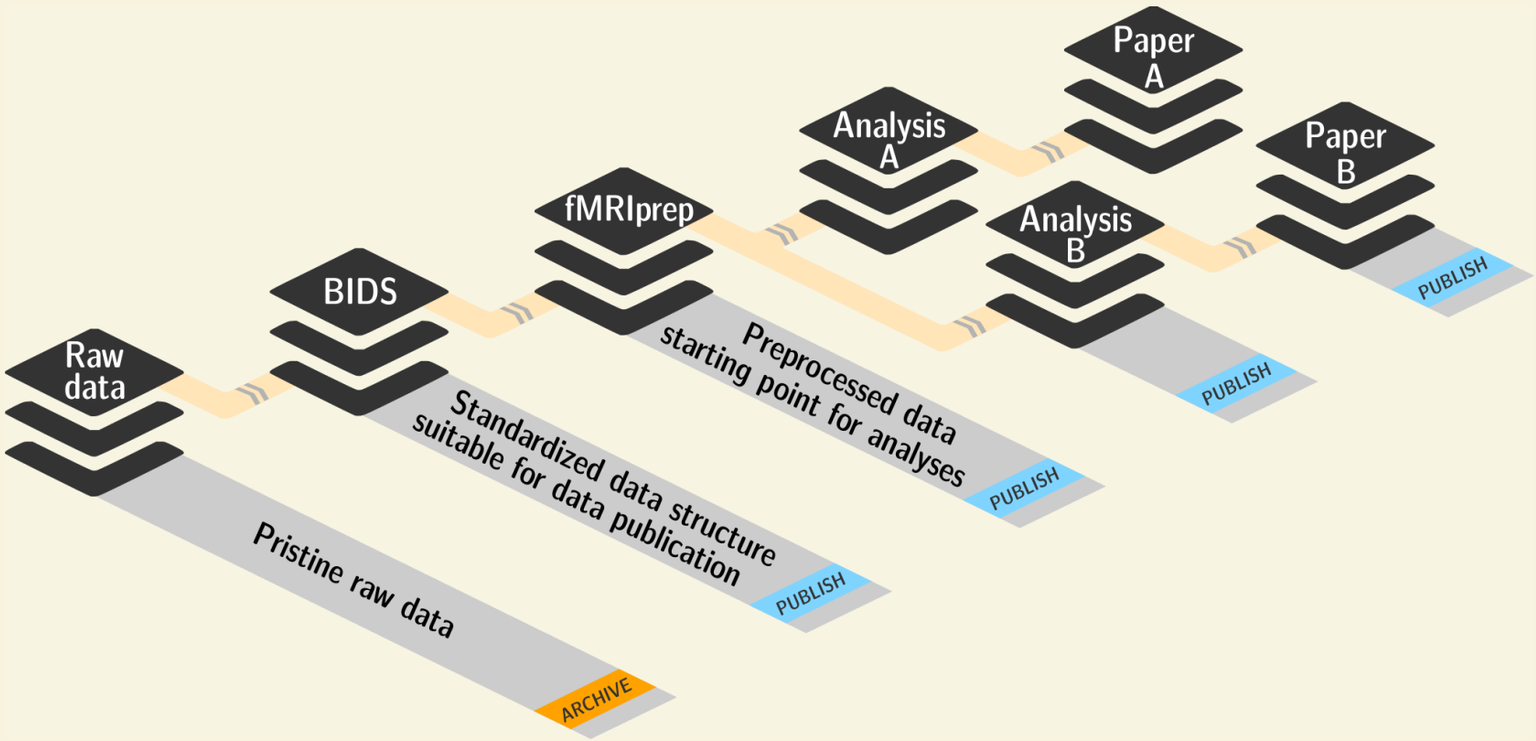
Image credit: Full comic at http://phdcomics.com/comics.php?f=1979

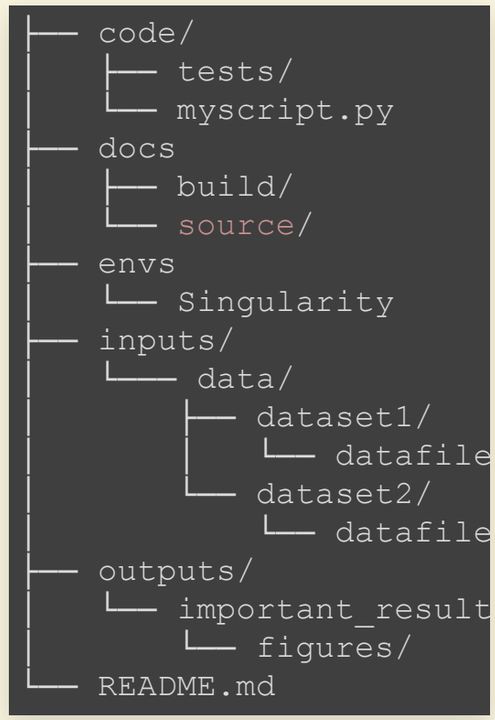
Code to follow along:
http://handbook.datalad.org/en/latest/code_from_chapters/10_yoda_code.html

BASIC ORGANIZATIONAL PRINCIPLES FOR DATASETS

Keep everything clean and modular

- An analysis is a superdataset, its components are subdatasets, and its structure modular



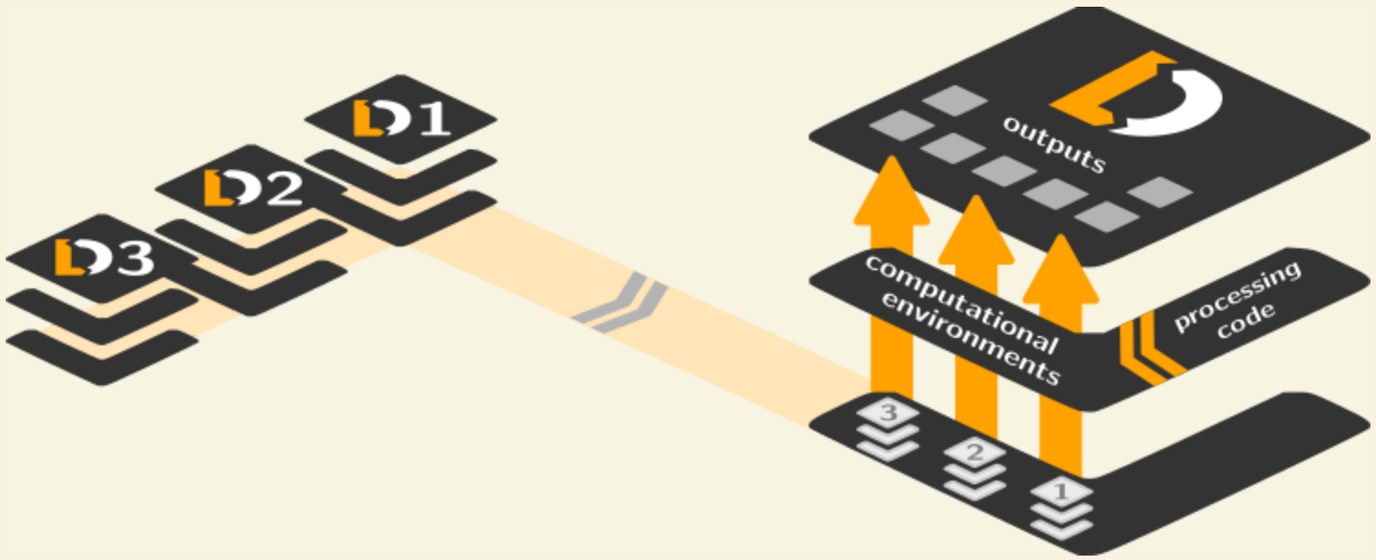


- do not touch/modify raw data: save any results/computations *outside* of input datasets
- Keep a superdataset self-contained: Scripts reference subdatasets or files with *relative paths*

BASIC ORGANIZATIONAL PRINCIPLES FOR DATASETS

Record where you got it from, where it is now, and what you do to it

- Link datasets (as subdatasets), record data origin
- Collect and store provenance of all contents of a dataset that you create

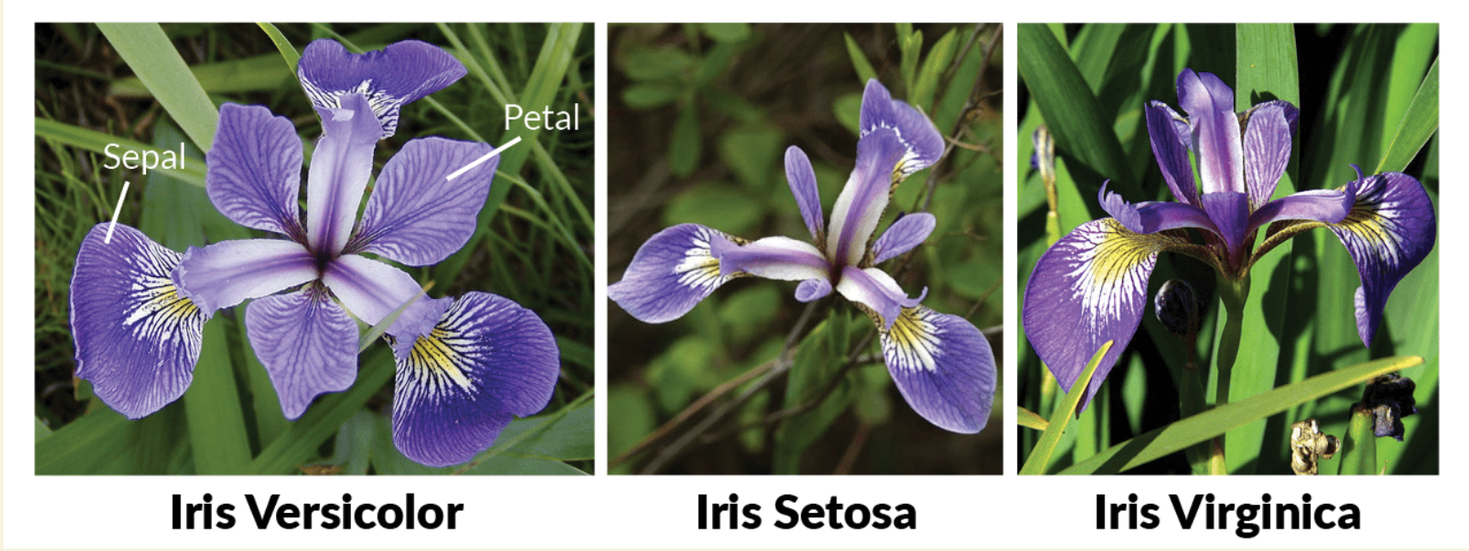


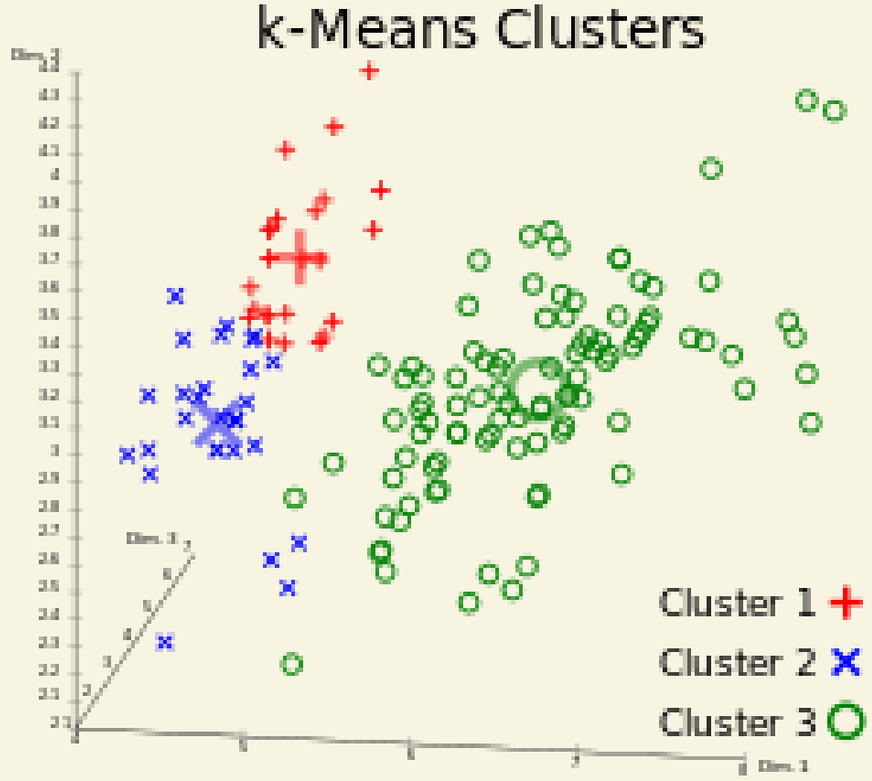
Document everything:

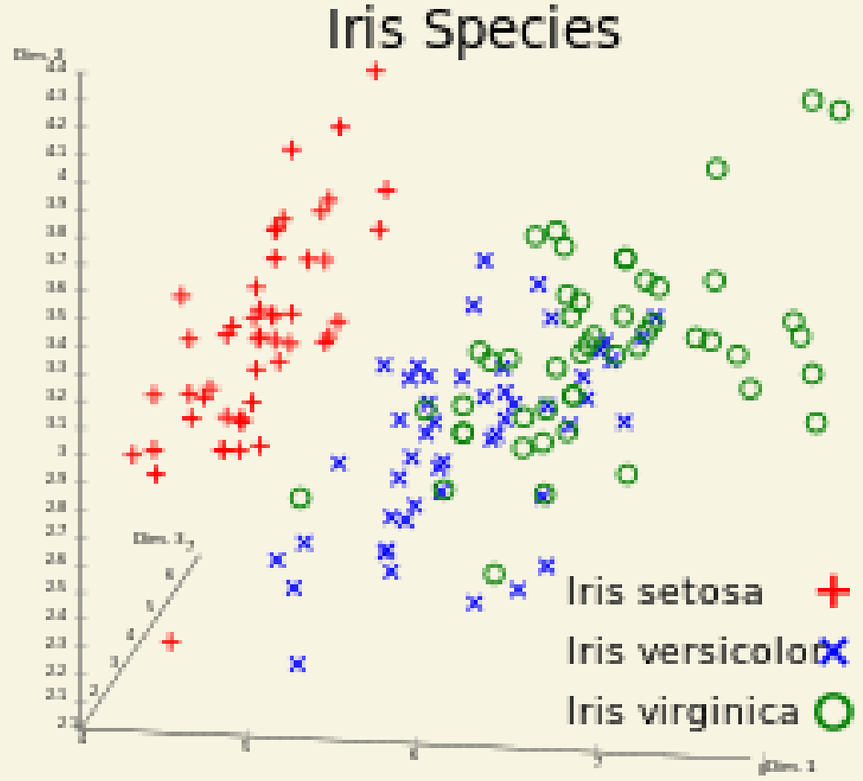
- Which script produced which output? From which data? In which software environment? …

Find out more about organizational principles in the YODA principles!

A CLASSIFICATION ANALYSIS ON THE IRIS FLOWER DATASET





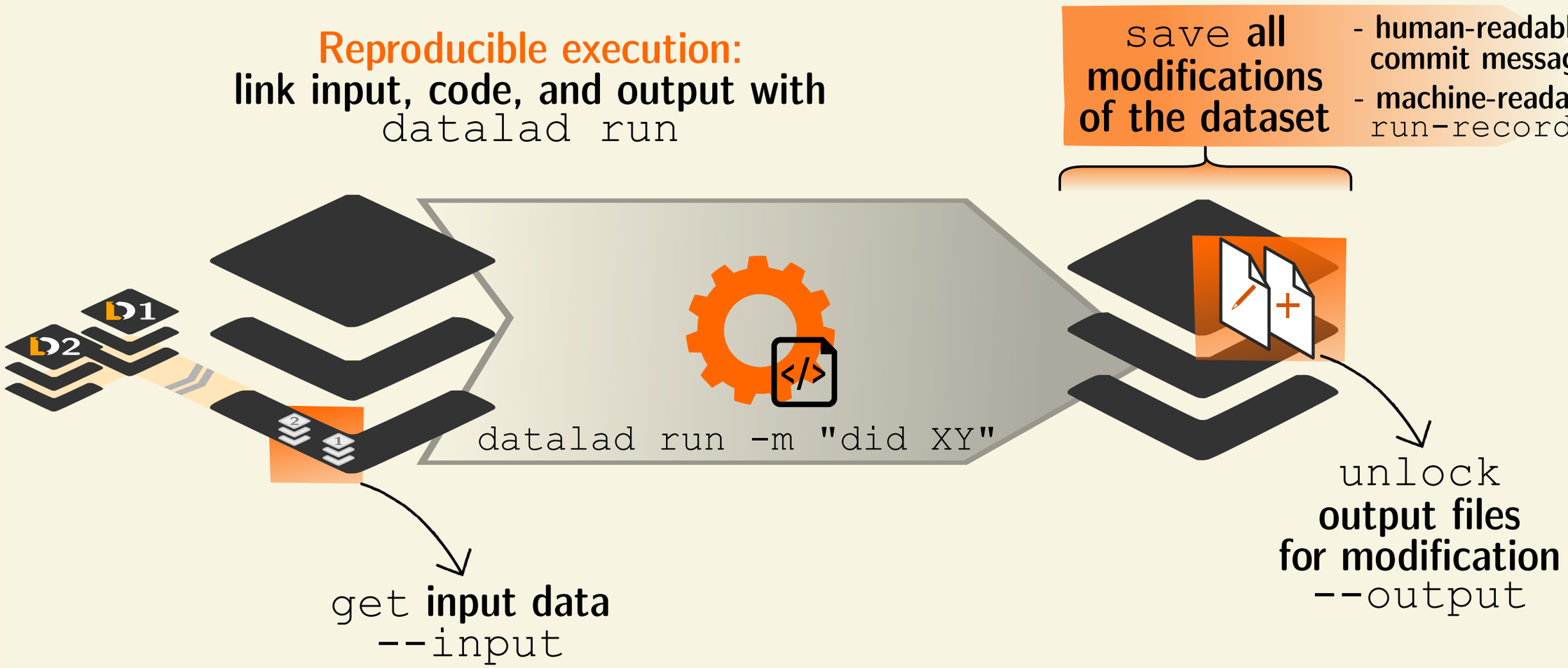


REPRODUCIBLE EXECUTION & PROVENANCE CAPTURE

datalad run

REPRODUCIBLE EXECUTION & PROVENANCE CAPTURE

datalad run



COMPUTATIONAL REPRODUCIBILITY

- Code may produce different results or fail if run in a different software environment
- DataLad datasets can store (and share) software environments (Docker or Singularity containers) and reproducibly execute code inside of the software container, capturing software as additional provenance
- DataLad extension: `datalad-container`

datalad-containers run

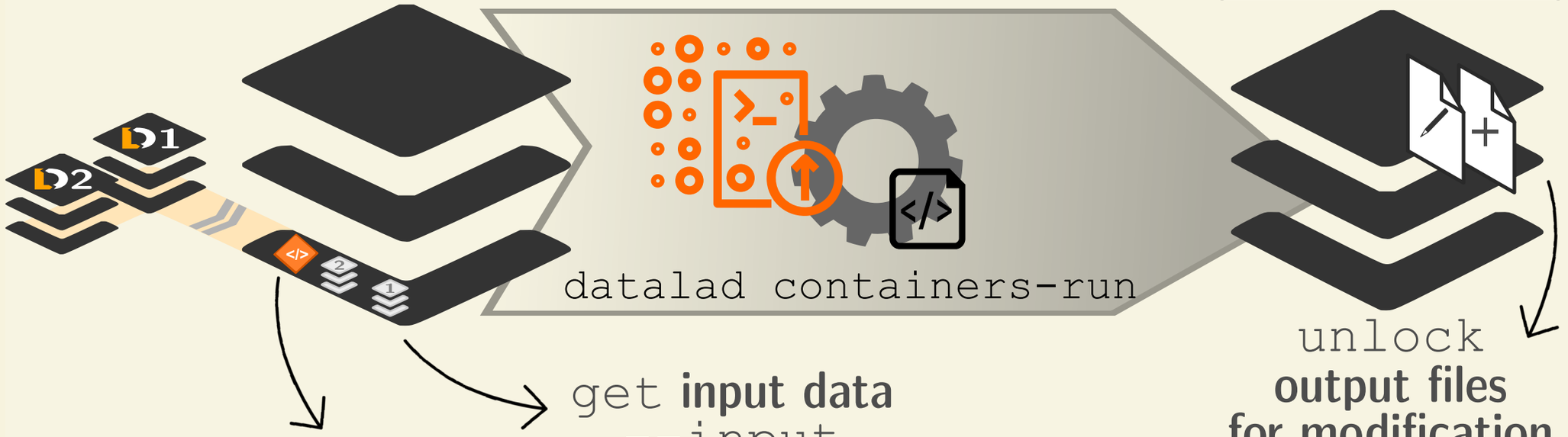
COMPUTATIONAL REPRODUCIBILITY

- Code may produce different results or fail if run in a different software environment
- DataLad datasets can store (and share) software environments (Docker or Singularity containers) and reproducibly execute code inside of the software container, capturing software as additional provenance
- DataLad extension: `datalad-container`

datalad-containers run

link input, code, output, and software with
datalad containers-run

save all
modifications
of the dataset



get input data
--input

unlock
output files
for modification
--output

get and use registered software
container for computation
--container-name

HOW TO GET STARTED WITH DATALAD

HOW TO GET STARTED WITH DATALAD

Read the DataLad handbook

HOW TO GET STARTED WITH DATALAD

Read **the DataLad handbook**

An interactive, hands-on crash-course (free and open source)

HOW TO GET STARTED WITH DATALAD

Read the DataLad handbook

An interactive, hands-on crash-course (free and open source)

Check out or used public DataLad datasets, e.g., from OpenNeuro

HOW TO GET STARTED WITH DATALAD

Read the DataLad handbook

An interactive, hands-on crash-course (free and open source)

Check out or used public DataLad datasets, e.g., from OpenNeuro

```
$ datalad clone ///openneuro/ds000001
[INFO   ] Cloning http://datasets.datalad.org/openneuro/ds000001 [1 other candidates] into '/tmp
[INFO   ] access to 1 dataset sibling s3-PRIVATE not auto-enabled, enable with:
|              datalad siblings -d "/tmp/ds000001" enable -s s3-PRIVATE
install(ok): /tmp/ds000001 (dataset)

$ cd ds000001
$ ls sub-01/*
sub-01/anat:
sub-01_inplaneT2.nii.gz  sub-01_T1w.nii.gz

sub-01/func:
sub-01_task-balloonanalogrisktask_run-01_bold.nii.gz
sub-01_task-balloonanalogrisktask_run-01_events.tsv
sub-01_task-balloonanalogrisktask_run-02_bold.nii.gz
sub-01_task-balloonanalogrisktask_run-02_events.tsv
sub-01_task-balloonanalogrisktask_run-03_bold.nii.gz
sub-01_task-balloonanalogrisktask_run-03_events.tsv
```

ACKNOWLEDGEMENTS

The DataLad Handbook

- Laura Waite
- Michael Hanke
- 17+ additional contributors

- Yaroslav Halchenko
- Joey Hess (git-annex)
- Benjamin Poldrack
- Kyle Meyer
- 22+ additional contributors

THANK YOU!

QUESTIONS?