You are free to

- share, adapt or re-mix
- photograph, video or broadcast
- blog, live-blog or post-video

this presentation

Provided that

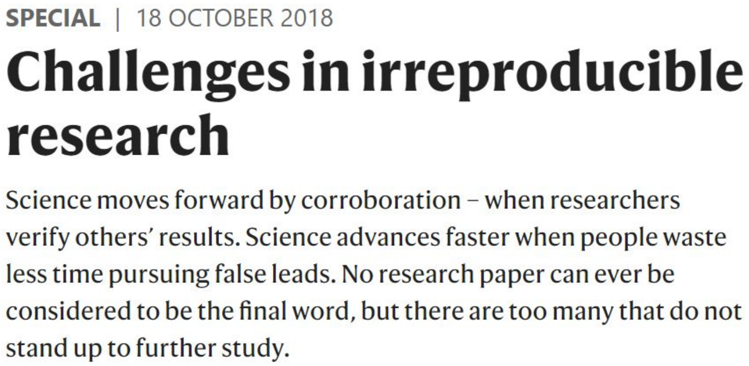
you attribute the work to its author and respect the rights and licenses associated with its components

# Best practices for research reproducibility

part I: Introduction to the FAIR principles for software and data management https://tinyurl.com/reproducible1

# why reproducible research?

because it’s good for science



SPECIAL | 18 OCTOBER 2018

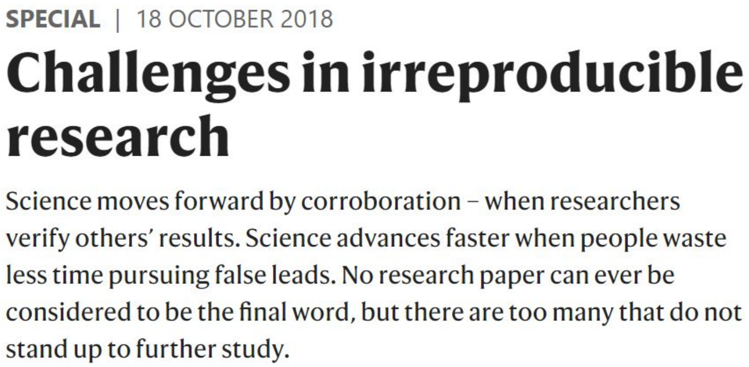
## Challenges in irreproducible research

Science moves forward by corroboration – when researchers verify others’ results. Science advances faster when people waste less time pursuing false leads. No research paper can ever be considered to be the final word, but there are too many that do not stand up to further study.

Five selfish reasons to work reproducibly - Challenges in irreproducible research

# why reproducible research?

because it’s good for science

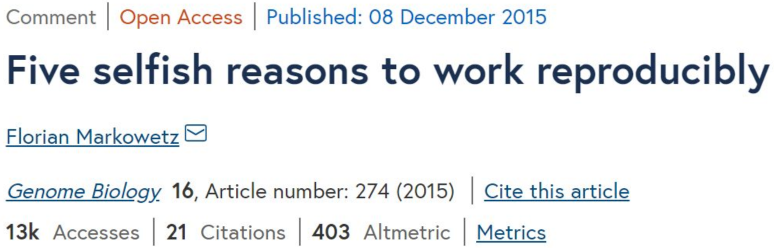


SPECIAL | 18 OCTOBER 2018

**Challenges in irreproducible research**

Science moves forward by corroboration – when researchers verify others’ results. Science advances faster when people waste less time pursuing false leads. No research paper can ever be considered to be the final word, but there are too many that do not stand up to further study.

because it’s good for you



Comment | Open Access | Published: 08 December 2015

**Five selfish reasons to work reproducibly**

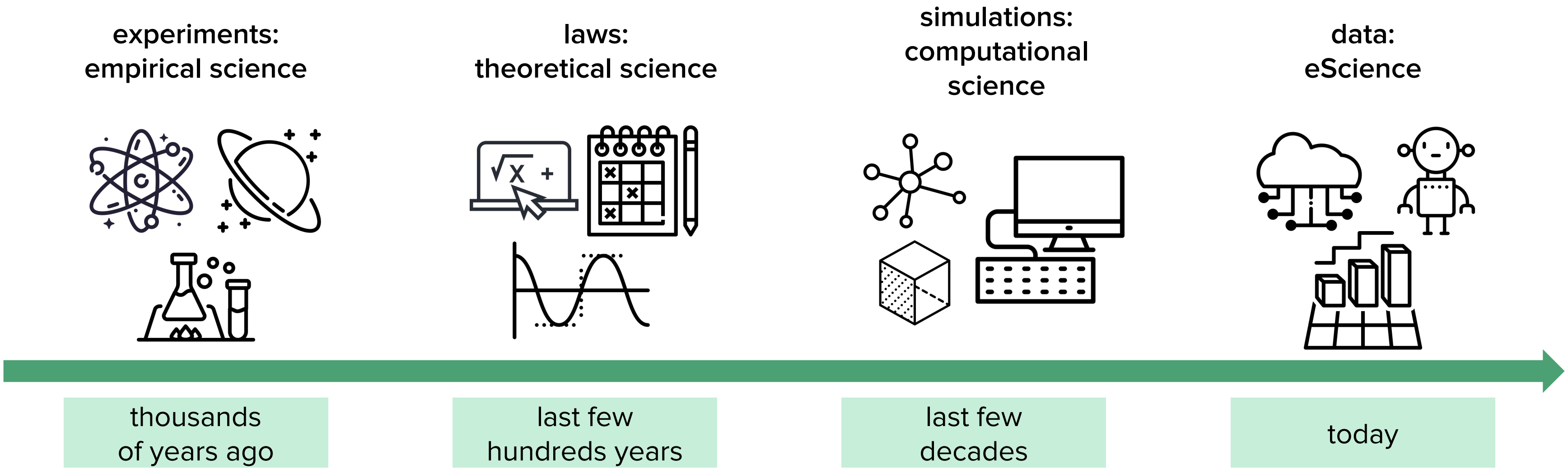
Florian Markowetz

*Genome Biology* **16**, Article number: 274 (2015) | Cite this article

**13k** Accesses | **21** Citations | **403** Altmetric | Metrics

Five selfish reasons to work reproducibly - Challenges in irreproducible research

# the 4th paradigm: data-driven science



**experiments: empirical science**

**laws: theoretical science**

**simulations: computational science**

**data: eScience**

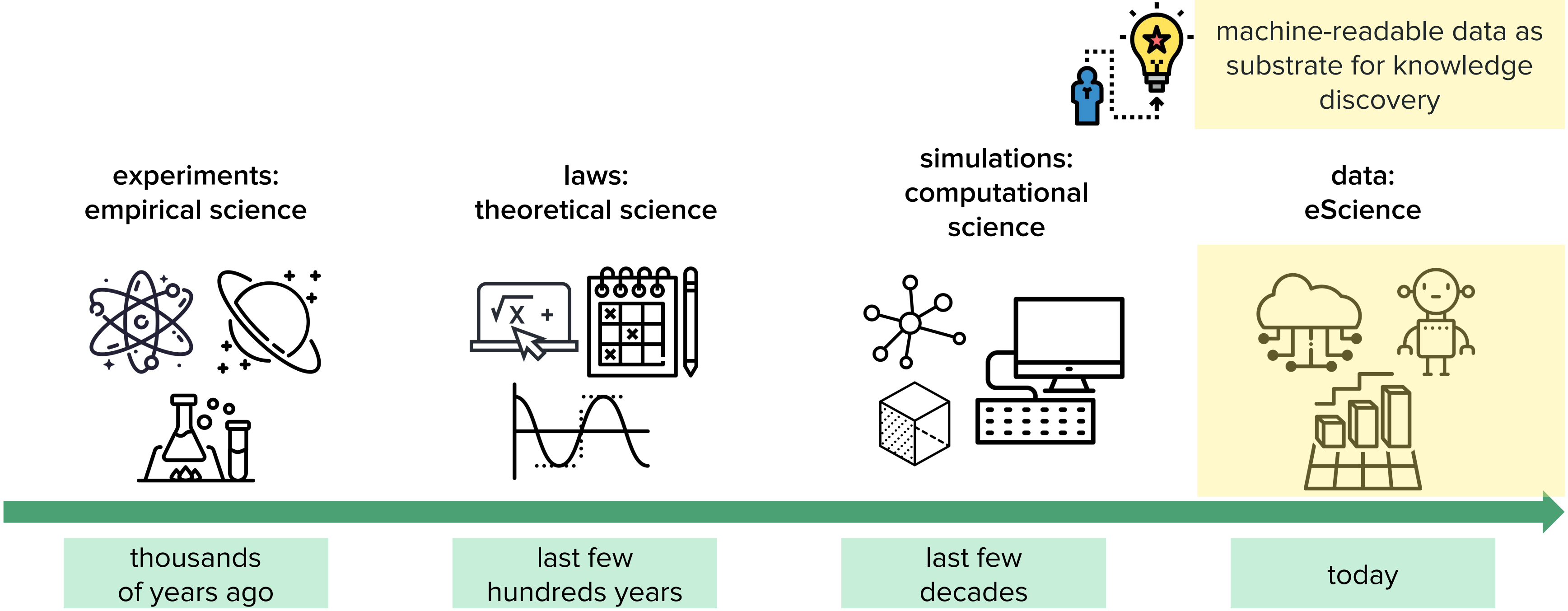
thousands of years ago

last few hundreds years

last few decades

today

# the 4th paradigm: data-driven science



machine-readable data as substrate for knowledge discovery

**experiments:**
**empirical science**

thousands of years ago

**laws:**
**theoretical science**

last few hundreds years

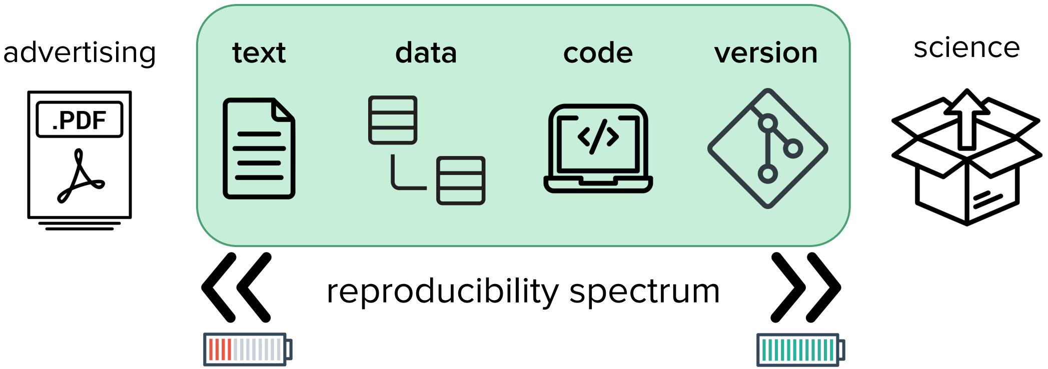
**simulations:**
**computational science**

last few decades

**data:**
**eScience**

today

# discovering and using research objects on the web



|  |  | Data |  |
|---|---|---|---|
|  |  | Same | Different |
| Analysis | Same | Reproducible | Replicable |
|  | Different | Robust | Generalisable |

**reproducibility** is the minimum standard for research **validity**

Image by Scriberia for The Turing Way community; The Turing Way

# the FAIR principles: guidance for data stewardship

PERSISTANT
FINDABLE
MEANINGFUL INTERACTION
ACCESSIBLE
FAIR
PRINCIPLES
INTEROPERABLE
FULL DISCLOSURE
REUSABLE
Scriberia

assist discovery and reuse through the web

**Findable**: sufficiently rich metadata + unique and persistent identifiers

**Accessible**: metadata and data are understandable to humans and machines and deposited in a trusted repository

**Interoperable**: metadata use a formal language for knowledge representation

**Reusable**: data have clear usage licenses and provide accurate information on provenance

FAIR != OPEN

FAIR comes in degrees

FAIR = agnostic of technical implementations

Image by Scriberia for The Turing Way community and used under a CC-BY licence; The Turing Way

# F for findable: persistent identifiers

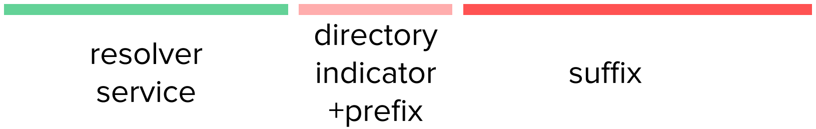
- a PID is a globally unique and long-lasting reference to something, *e.g.*, documents, files, books, people
- a PID is separated from location: if a web document is moved, the PID points to the same object in the new location (~~URLs~~)

# F for findable: persistent identifiers

- a PID is a globally unique and long-lasting reference to something, *e.g.*, documents, files, books, people
- a PID is separated from location: if a web document is moved, the PID points to the same object in the new location (~~URLs~~)

## Digital Object Identifier: DOI

- the DOI is a common identifier used for academic, professional, and governmental information such as articles, datasets, reports, and other supplemental information
- https://doi.org/10.5281/zenodo.3679141



resolver service

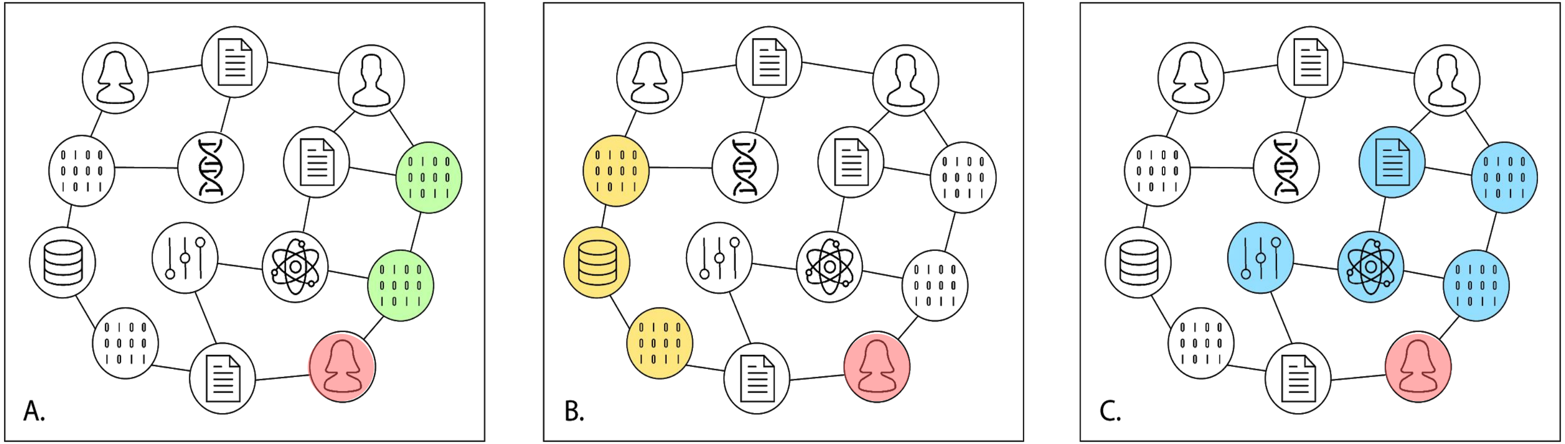
directory indicator +prefix

suffix

# author identifier: ORCID

Connecting Research
and Researchers

- PIDs need to be connected
- get an ORCID and tell your ORCID about your other identifiers



Different versions of software code

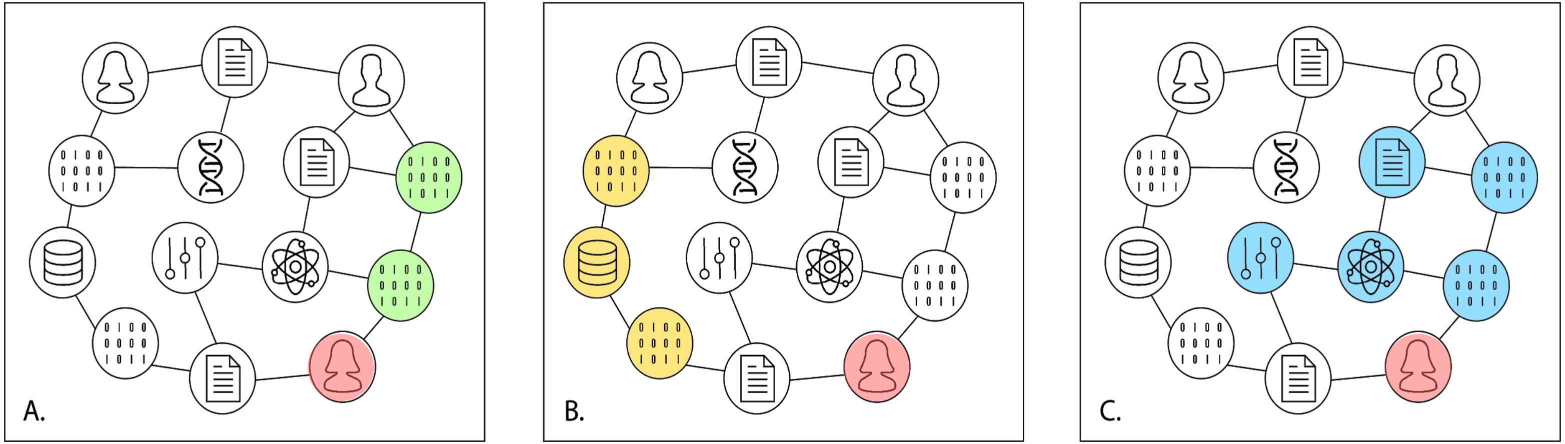
Datasets hosted by a particular repository

All digital objects connected to a research object

# author identifier: ORCID

Connecting Research and Researchers

- PIDs need to be connected
- get an ORCID and tell your ORCID about your other identifiers



Different versions of software code

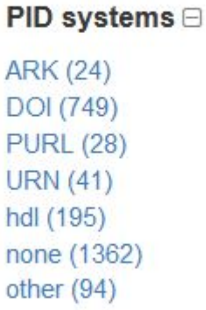
Datasets hosted by a particular repository

All digital objects connected to a research object

https://orcid.org/

# data repositories and data journals

- choose the right repository
  - run a query on https://www.re3data.org/search?query= and filter by the PID service: what do you find? (spoiler alert: very few repositories use a PID service)
- list of databases at the https://fairsharing.org/databases/
  - the Open Source Brain - the OpenNEURO...

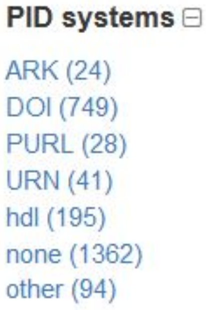


**PID systems** ⊟

ARK (24)
DOI (749)
PURL (28)
URN (41)
hdl (195)
none (1362)
other (94)

# data repositories and data journals

- choose the right repository
  - run a query on https://www.re3data.org/search?query= and filter by the PID service: what do you find? (spoiler alert: very few repositories use a PID service)
- list of databases at the https://fairsharing.org/databases/
  - the Open Source Brain - the OpenNEURO...
- data citation is necessary to give credit, incentive FAIR behaviors, track impact of datasets, measure return of investment and easily locate and access data
- when using data of others ➔ cite both the related peer-reviewed literature and the actual datasets



**PID systems** ⊟

ARK (24)
DOI (749)
PURL (28)
URN (41)
hdl (195)
none (1362)
other (94)

[dataset] 34. Frazier, JA, Hodge, SM, Breeze, JL, Giuliano, AJ, Terry, JE, Moore, CM, Makris, N. CANDI Share Schizophrenia Bulletin 2008 data; 2008. Child and Adolescent NeuroDevelopment Initiative. https://dx.doi.org/10.18116/C6159Z

# A for accessible: protocols

- accessible does not imply open
- data and metadata need to be retrievable by their identifier using a **standardized communications protocol**
- research repositories often use the OAI-PMH or REST API protocols to interface with data in the repository

# A for accessible: protocols

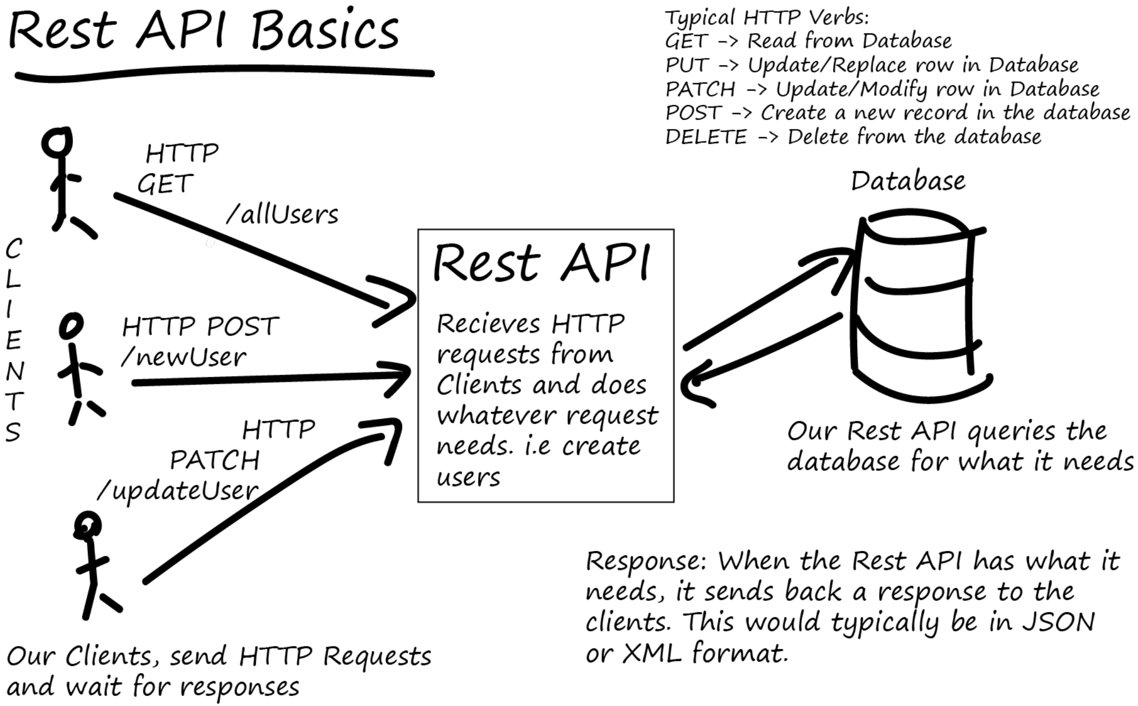
- accessible does not imply open
- data and metadata need to be retrievable by their identifier using a standardized communications protocol
- research repositories often use the OAI-PMH or REST API protocols to interface with data in the repository
- example: Zenodo has a REST API which you can also use to programmatically upload and publish your research outputs

*have you deposited any research output on Zenodo?*

zenodo

[https://zenodo.org/](https://zenodo.org/)

# REST API



Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database

Recieves HTTP requests from Clients and does whatever request needs. i.e create users

Our Rest API queries the database for what it needs

Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

Our Clients, send HTTP Requests and wait for responses

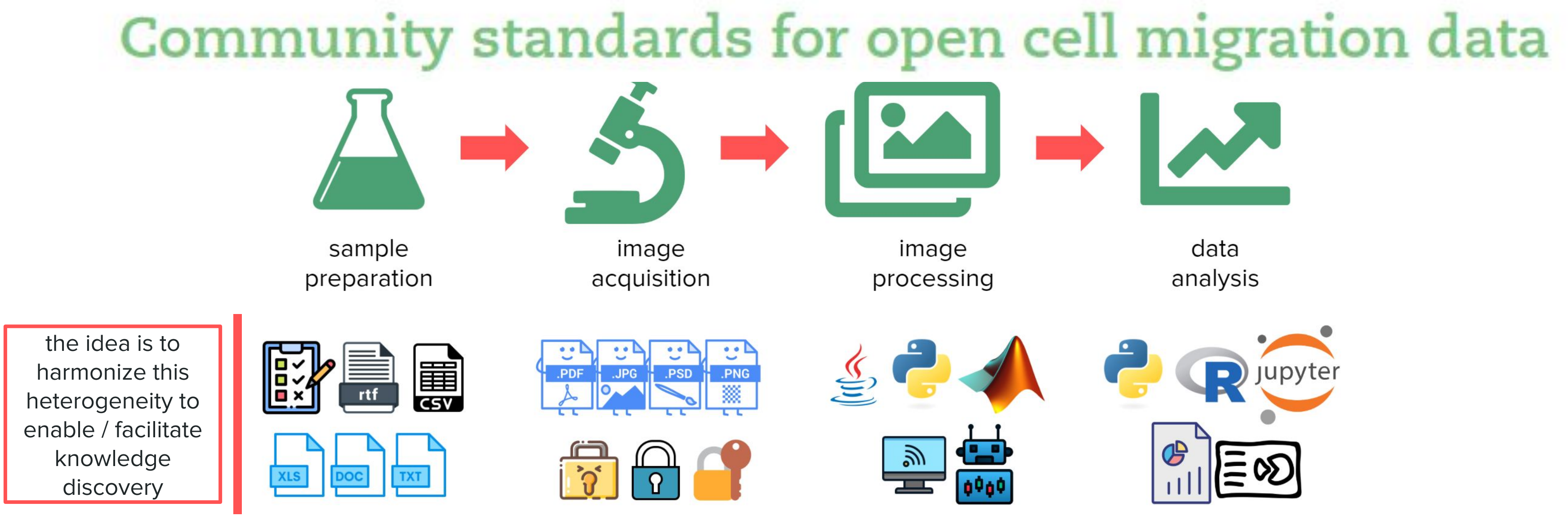
Image from What is a RESTful API?

# I for interoperable: standards

- data and metadata use a formal, accessible, shared, and broadly applicable language for knowledge representation

# I for interoperable: standards

- data and metadata use a formal, accessible, shared, and broadly applicable language for knowledge representation



the idea is to harmonize this heterogeneity to enable / facilitate knowledge discovery

Community Standards for Open Cell Migration Data, Gonzales, Masuzzo *et al.*, 2019

# standards for the Life Sciences

- the ISA framework: 'Investigation' (the project context), 'Study' (a unit of research) and 'Assay' (analytical measurement)

# standards for the Life Sciences

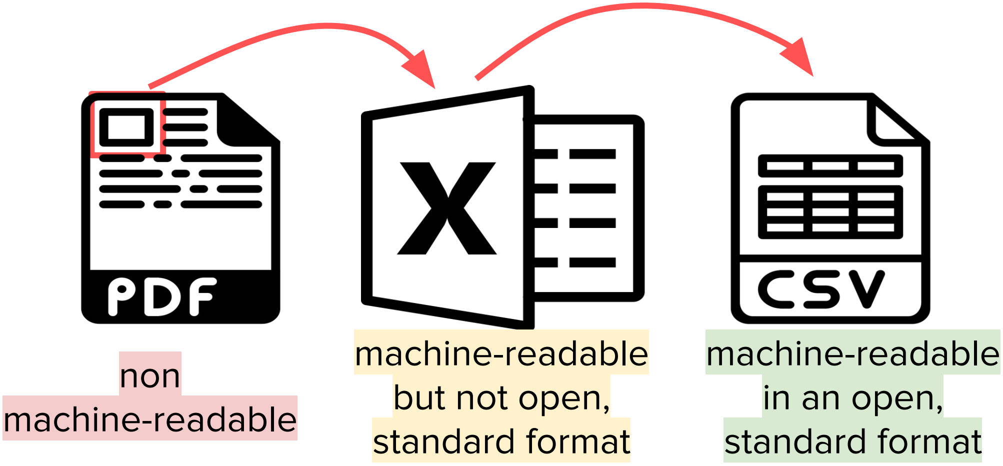
- the ISA framework: 'Investigation' (the project context), 'Study' (a unit of research) and 'Assay' (analytical measurement)
- the FAIRSharing catalogue for standards: a comprehensive list of terminology artifacts (ontologies, controlled vocabularies), data formats, models and reporting guidelines
    - some random ‘Neuroscience’ terms ➔ Brain Imaging Data Structure model / NeuroML model / Minimum Information about a Neuroscience Investigation / Neuro Behavior Ontology / NeuroInformatics Exchange Format

# standards for the Life Sciences

- the ISA framework: 'Investigation' (the project context), 'Study' (a unit of research) and 'Assay' (analytical measurement)
- the FAIRSharing catalogue for standards: a comprehensive list of terminology artifacts (ontologies, controlled vocabularies), data formats, models and reporting guidelines
  - some random ‘Neuroscience’ terms → Brain Imaging Data Structure model / NeuroML model / Minimum Information about a Neuroscience Investigation / Neuro Behavior Ontology / NeuroInformatics Exchange Format
- the BioSchemas: extends Schema.org (semantic markup for the web) to allow specifically for the description of life science resources
  - *e.g.* specification for the type BioSample: https://bioschemas.org/types/BioSample/0.1-RELEASE-2019_06_19/

# R for reusable: human- and machine-readable

- data and metadata come with clear license usage
- human-readable (PDF OK) is not the same as machine-readable (PDF not OK, CSV, JSON, XML OK)
- both need to be ensured



non machine-readable

machine-readable but not open, standard format

machine-readable in an open, standard format

# best practices: file formats

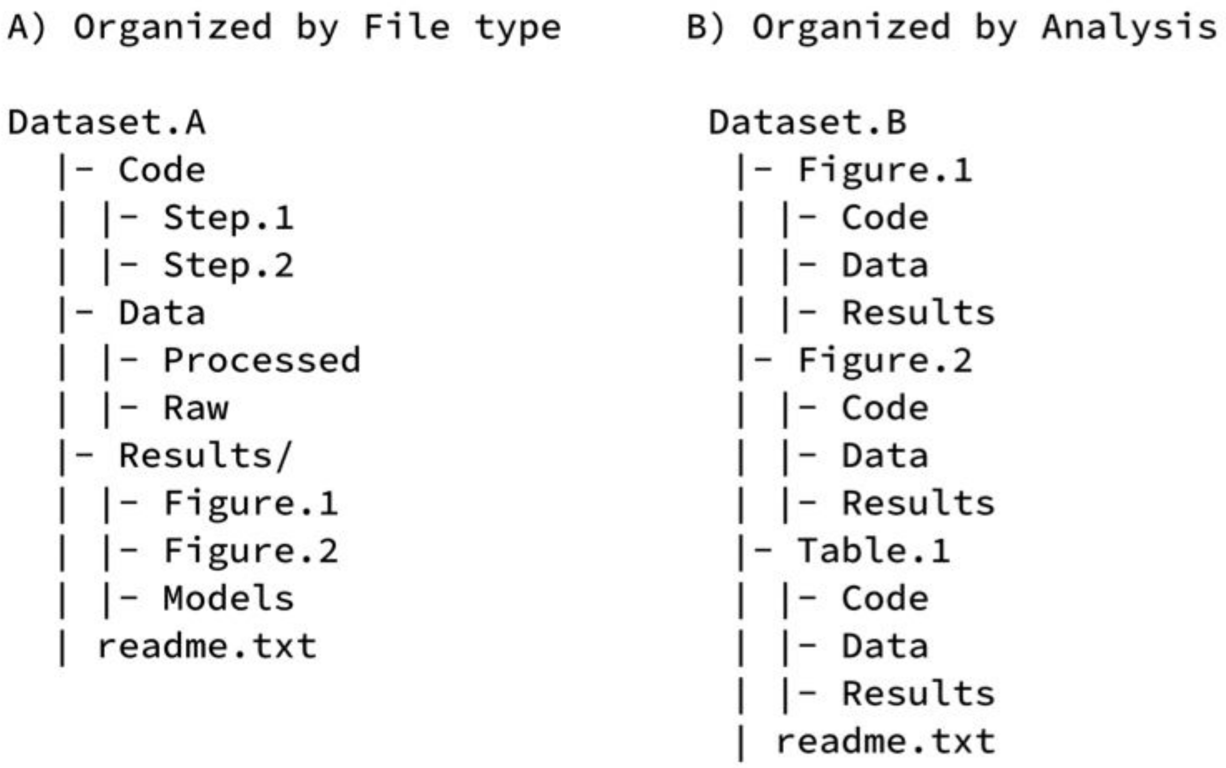
- text
  - **README files** should be in plain text format (ASCII, UTF-8)
    - (see here for a ref on how to best write a README file)
  - Comma-separated values (CSV) for tabular data
  - Semi-structured plain text formats for non-tabular data (e.g., protein sequences)
  - Structured plain text (XML, JSON)
- images: PDF, JPEG, PNG, TIFF, SVG
- audio: FLAC, AIFF, WAV, MP3, OG
- video: AVI, MPEG, MP4
- compressed file archive: TAR.GZ, 7Z, ZIP

List of open formats

# best practices: file naming

- the Stanford Libraries guidance on file naming is a great place to start
- general rule: know what a file is before you double-click on it
- avoid spaces and special characters
- use common letter case patterns such as *kebab-case*, *camelCase*, or *snake_case*
- include author name, project name, type of data, type of analysis, date, and file extension
  - some examples from Dryad:
    - 1900-2000_sasquatch_migration_coordinates.csv
    - Smith-fMRI-neural-response-to-cupcakes-vs-vegetables.nii.gz
    - 2015-SimulationOfTropicalFrogEvolution.R

# best practices: data directories



```
A) Organized by File type

Dataset.A
  |- Code
  |  |- Step.1
  |  |- Step.2
  |- Data
  |  |- Processed
  |  |- Raw
  |- Results/
  |  |- Figure.1
  |  |- Figure.2
  |  |- Models
  | readme.txt
```

```
B) Organized by Analysis

Dataset.B
  |- Figure.1
  |  |- Code
  |  |- Data
  |  |- Results
  |- Figure.2
  |  |- Code
  |  |- Data
  |  |- Results
  |- Table.1
  |  |- Code
  |  |- Data
  |  |- Results
  | readme.txt
```

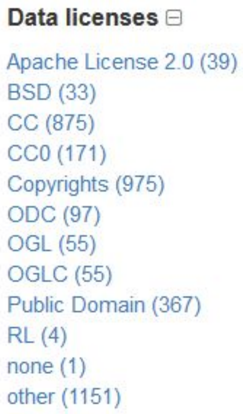
from Dryad FAIR Data

# best practices: licenses

- governamental data belong to the public domain
- the new H2020 credo for research data is ‘**as open as possible, as closed as necessary**’

# best practices: licenses

- governamental data belong to the public domain
- the new H2020 credo for research data is ‘as open as possible, as closed as necessary’
- for data accompanying scientific publications, the **Creative Commons** licences are recommended
    - ○ run a query on https://www.re3data.org/search?query= filtering by data licenses: what do you find?



**Data licenses** ⊟

Apache License 2.0 (39)
BSD (33)
CC (875)
CC0 (171)
Copyrights (975)
ODC (97)
OGL (55)
OGLC (55)
Public Domain (367)
RL (4)
none (1)
other (1151)

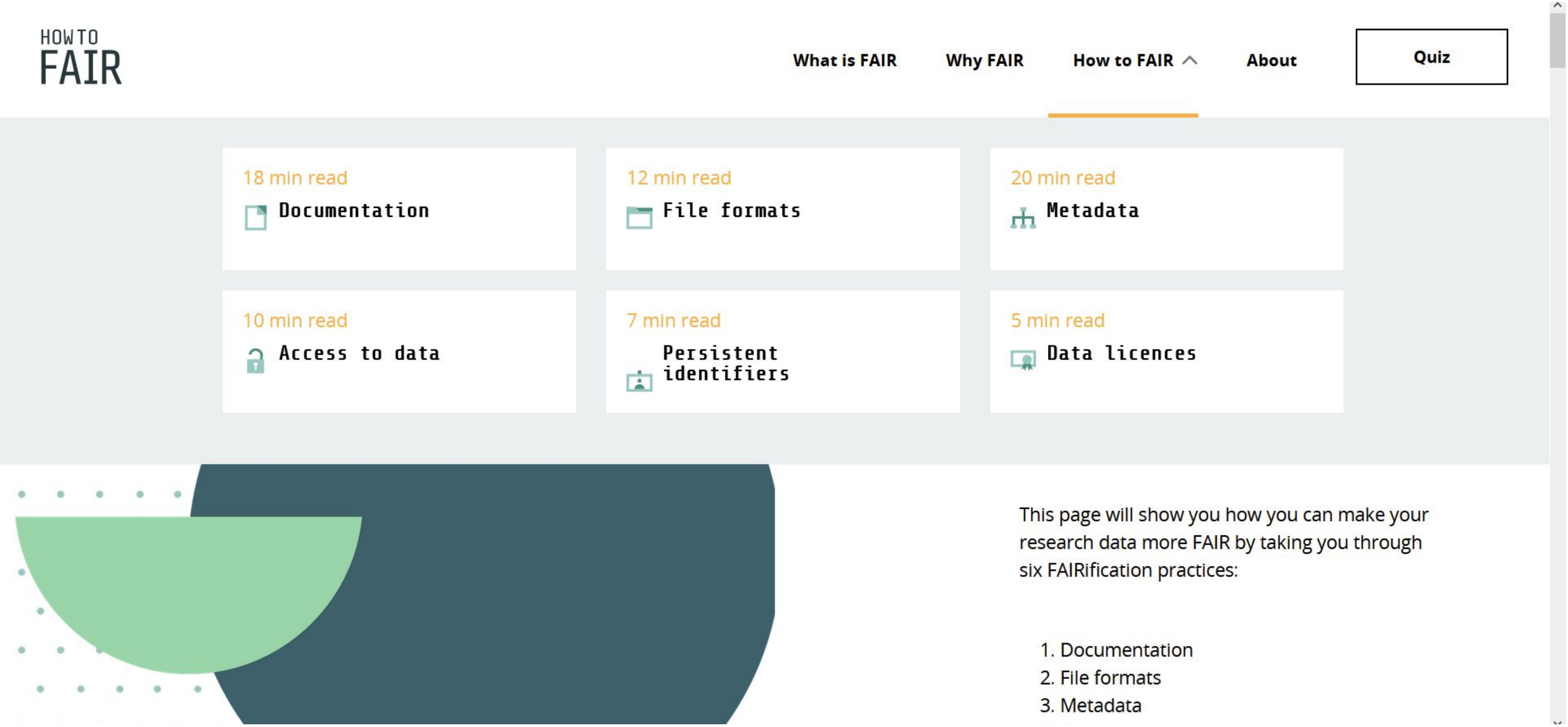
# best practices: licenses

- governamental data belong to the public domain
- the new H2020 credo for research data is ‘as open as possible, as closed as necessary’
- for data accompanying scientific publications, the Creative Commons licences are recommended
    - run a query on https://www.re3data.org/search?query= filtering by data licenses: what do you find?
- open data and content can be freely used, modified, and shared by anyone for any purpose (from the Open Definition)
- look for conformant licenses here (and next slide)

# best practices: licenses

| License | Domain | By | SA | Comments |
|---|---|---|---|---|
| Creative Commons CCZero (CC0) | Content, Data | N | N | Dedicate to the Public Domain (all rights waived) |
| Open Data Commons Public Domain Dedication and Licence (PDDL) | Data | N | N | Dedicate to the Public Domain (all rights waived) |
| Creative Commons Attribution 4.0 (CC-BY-4.0) | Content, Data | Y | N | |
| Open Data Commons Attribution License (ODC-BY) | Data | Y | N | Attribution for data(bases) |
| Creative Commons Attribution Share-Alike 4.0 (CC-BY-SA-4.0) | Content, Data | Y | Y | |
| Open Data Commons Open Database License (ODbL) | Data | Y | Y | Attribution-ShareAlike for data(bases) |

# how to FAIR



18 min read
Documentation

12 min read
File formats

20 min read
Metadata

10 min read
Access to data

7 min read
Persistent identifiers

5 min read
Data licences

This page will show you how you can make your research data more FAIR by taking you through six FAIRification practices:

1. Documentation
2. File formats
3. Metadata

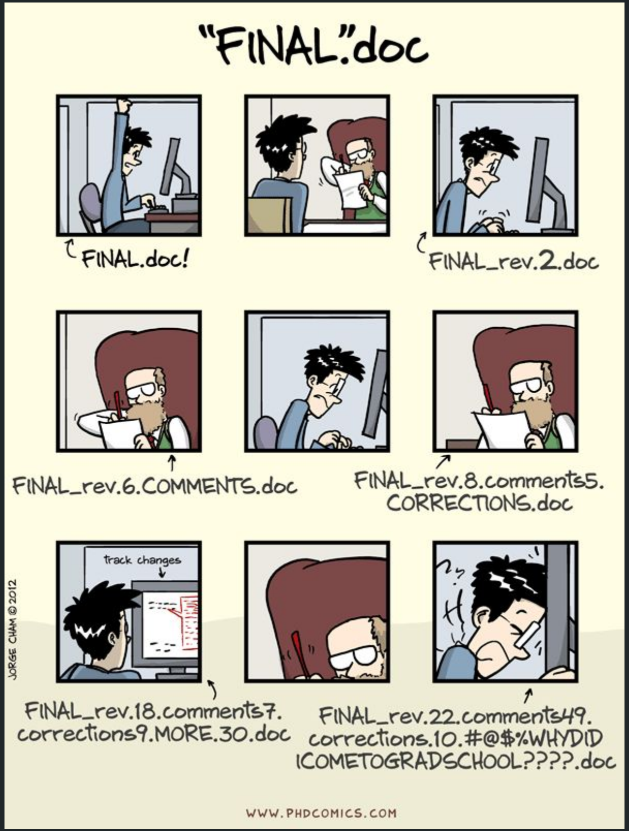
https://howtofair.dk/

# Best practices for research reproducibility

part II: Working with data in a robust, reliable and replicable manner https://tinyurl.com/reproducible2

# version control

use Git to enhance your
research reproducibility



# version control

- version control helps you **record changes** you make to the files in a directory on your computer; version control software & tools are increasingly being used to **collaborate** in research and academic environments

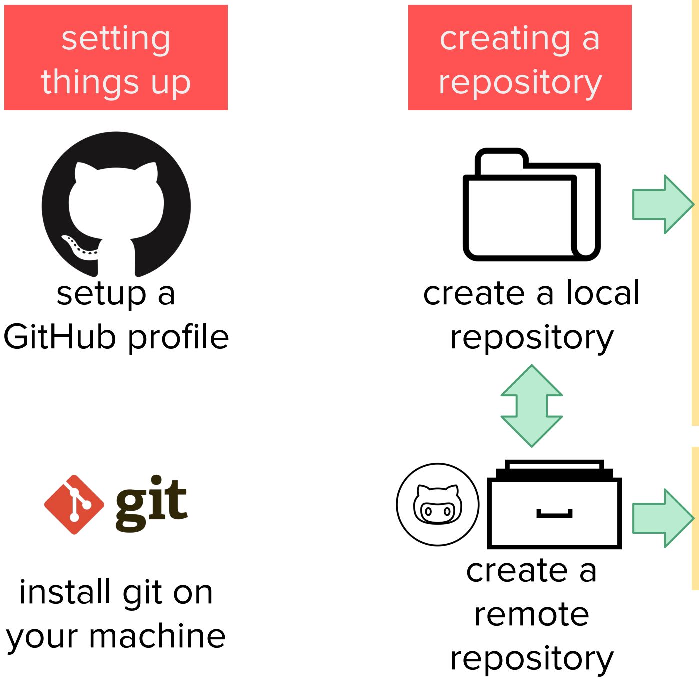
# version control

- version control helps you **record changes** you make to the files in a directory on your computer; version control software & tools are increasingly being used to **collaborate** in research and academic environments

## Benefits of using version control

- **collaboration** ➔ define formalized ways to work together and share writing and code
- **versioning** ➔ robust and rigorous log of changes to a file, without renaming files (v1, v2, final_copy)
- **rolling back** ➔ quickly undo a set of changes; this can be useful when new writing or new additions to code introduce problems
- **understanding** ➔ understand how the code or writing came to be, who wrote or contributed particular parts, and who you might ask to help understand it better
- **backup** ➔ while not meant to be a backup solution, using version control systems mean that your code and writing can be stored on multiple other computers

# version control: what we are going to do



setting things up

setup a GitHub profile

install git on your machine

creating a repository

create a local repository

create a remote repository

```
cd ~/Desktop
mkdir repo
cd repo
git init
git status
touch index.md
git status
git add index.md
git status
vi index.me
git commit -m 'add index.md'
```

```
git remote add origin https://github.com/youruser/repo.git
git remote -v
git push -u origin master
```

# git and GitHub

- git is the most used version control system
- GitHub is a popular website for hosting and sharing Git repositories remotely

**setting up git + GitHub**

if you do not have it on your machine yet, download git from https://git-scm.com/downloads

if you do not have one yet, create a GitHub account at https://github.com/

# git: create a local repository + adding/committing

`cd ~/Desktop` → move to Desktop

`mkdir repo` → create a directory called “repo” (use a name of your choice!)

`cd repo` → move to the directory

`git init` → initialize an empty repository

`ls -a` → show content of the directory, including hidden files

`git status` → show the status of the repository

# git: create a local repository + adding/committing

`cd ~/Desktop` → move to Desktop

`mkdir repo` → create a directory called “repo” (use a name of your choice!)

`cd repo` → move to the directory

`git init` → initialize an empty repository

`ls -a` → show content of the directory, including hidden files

`git status` → show the status of the repository

`touch index.md` → create an empty markdown file called index

`git status`

`git add index.md` → add the index file to the tracked files

`git status`

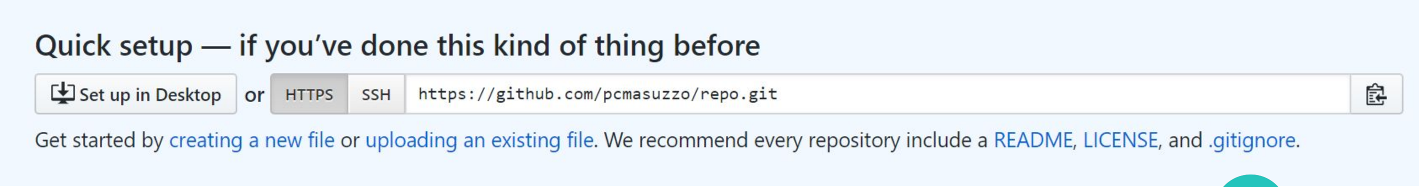
`vi index.me` → write some text to the index file (press i, type # Hello, world!, press esc, type :wq)

`git commit -m 'add index.md'`→ commit our changes to git

this is still only on your computer!

# git: create a remote repository on GitHub

- create a new repository on GitHub
  - use this resource to choose an open source license: https://choosealicense.com/
- connect your local repository with the remote repository



Quick setup — if you've done this kind of thing before

https://github.com/pcmasuzzo/repo.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

- push an existing repository from the command line

```
git remote add origin https://github.com/pcmasuzzo/repo.git
git remote -v
```

`git push -u origin master` → pushes our local changes to the remote repository

```
git status
```

modify the index file and then run:

```
git diff
```

# git: pull changes from the remote repository

- when working with others, or when we are making our own changes from different machines, we need a way of pulling those remote changes back into our local copy
- let’s go to our repository in GitHub and add there a README file
- we then have to pull the remote changes we made to keep our local repository in sync

`git pull` → pulls changes from the remote repository to the local one

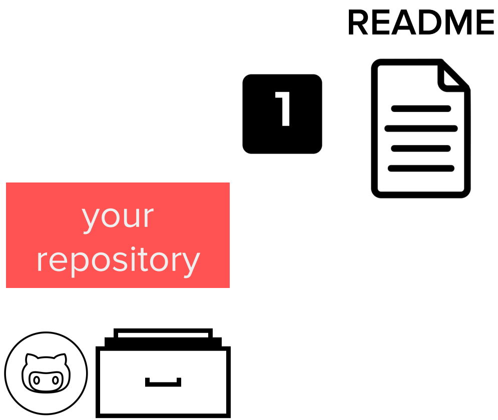
-----------------------------------------------------------------------------------------------

additional resources

[Git Cheatsheets for Quick Reference by The Carpentries]()

[GitHub Help Documentation]()

# best practices for a (code) repository



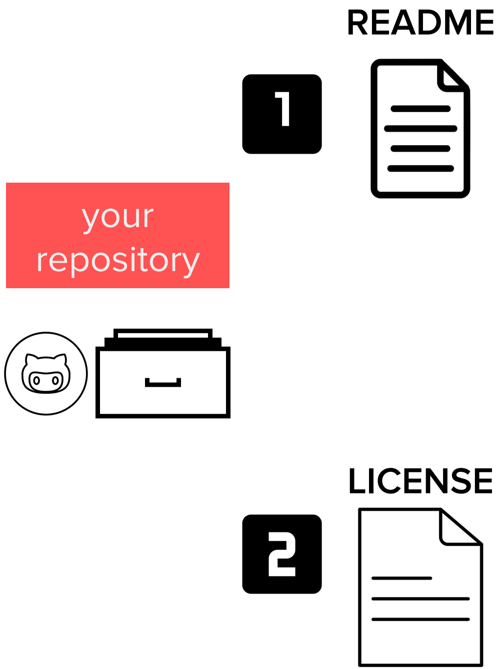
README

1

your repository

- a clear README is super important if you want people to understand the scope of your project, how to use it, the current status of it, how to get involved (more later…)
- when the README file is included in the root directory, GitHub will automatically display this on the homepage of your repository: this means it is the first thing people will often see!
  resource ➔ [A curated list of awesome READMEs]()

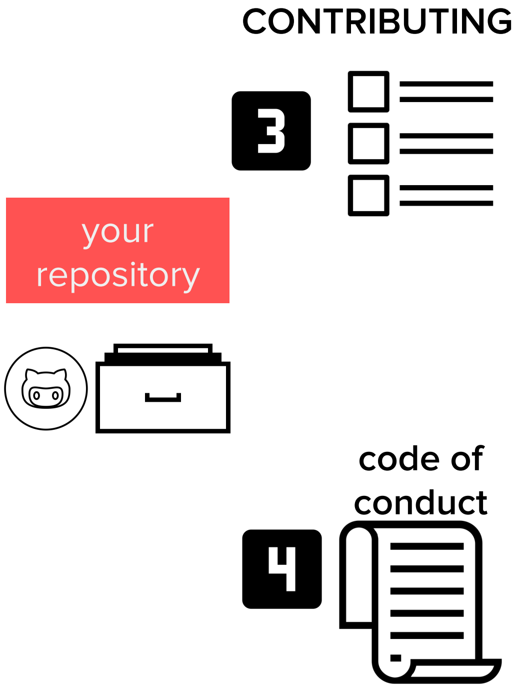
# best practices for a (code) repository



- a clear README is super important if you want people to understand the scope of your project, how to use it, the current status of it, how to get involved (more later...)
- when the README file is included in the root directory, GitHub will automatically display this on the homepage of your repository: this means it is the first thing people will often see!
  resource → [A curated list of awesome READMEs]

- MIT License: permissive license that lets people do whatever they want with your code as long as they provide appropriate attribution to you, and do not hold you liable
- Apache License 2.0: similar permissions to the MIT License, but also provides an express grant of patent rights from contributors to users
- GNU General Public License (GPL) v3: a copyleft license that requires anyone who redistributes your code, or a derivative work, to make the source available under the same terms as the original license
  resource → [Choose a License]

# best practices for a (code) repository



your repository

CONTRIBUTING

3

code of conduct

4

- a CONTRIBUTING file serves as a guide to potential contributors on how to engage with your project (and community)
- a CONTRIBUTING file can include: which protocol to use to suggest features/raise issues (*e.g.* via pull requests), which type of contributions are welcome/prioritized, link to additional external documentations, how to contact you…
  resource ➜ How to write CONTRIBUTING files

- a code of conduct is important for setting the ground rules for expected behaviour and participation for project contributors
- it is a critical element for creating and maintaining a healthy community that engages in a constructive and productive manner within a positive social atmosphere
- it is important not only to have a CoC, but, should violation of it occur, to also enforce this (never an easy task…)
  resource ➜ The Contributor Covenant

# make your code citable

- we have seen how DOIs are the backbone of the academic reference and metrics system
- if you write software for your research, you might want to make it citable
- we are going to do this by archiving one of your GitHub repositories and assigning a DOI with the data archiving tool Zenodo

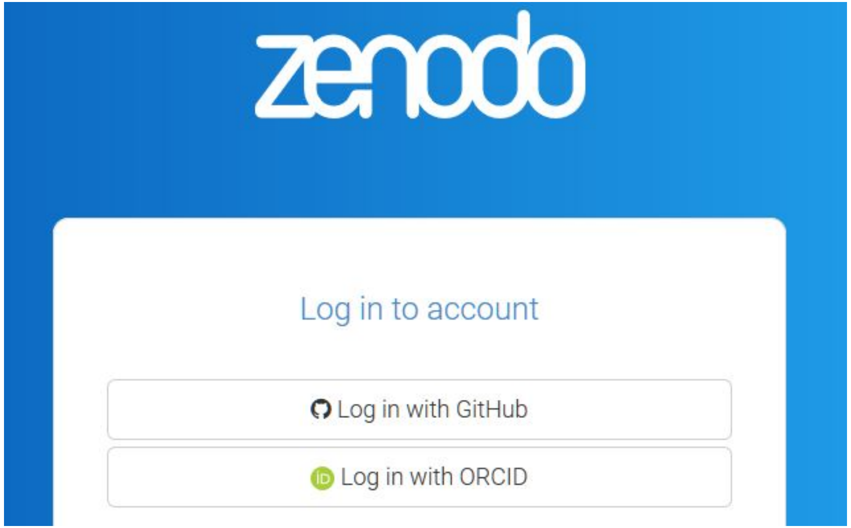
# make your code citable

- we have seen how DOIs are the backbone of the academic reference and metrics system
- if you write software for your research, you might want to make it citable
- we are going to do this by archiving one of your GitHub repositories and assigning a DOI with the data archiving tool Zenodo

1. choose a repository
   important ➔ at this stage, you should tell people how they can reuse your work by including a **license** in your repository!

# make your code citable

2. login into Zenodo



Log in to account

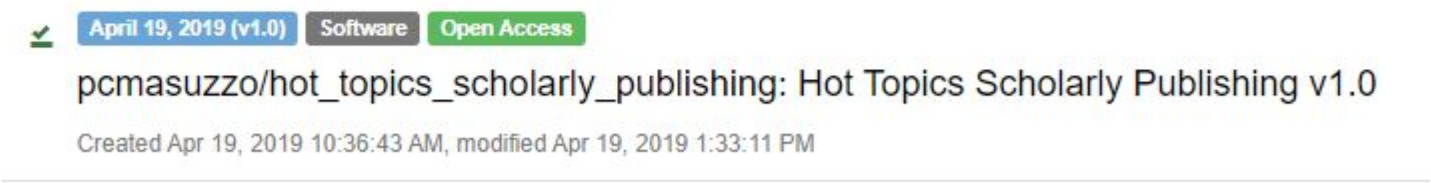
Log in with GitHub

Log in with ORCID

3. link GitHub and Zenodo (authorize Zenodo to connect with GitHub)

# make your code citable

4. on the Zenodo website navigate to the GitHub repository listing page and simply click the 'ON' button next to the repository you want to archive
5. make a new release of your code
    a. if this is your first formal release, call it v1.0.0 or v1.0
6. get a DOI: go to the Upload tab in Zenodo, select your uploaded repository, fill in some additional info and get a DOI

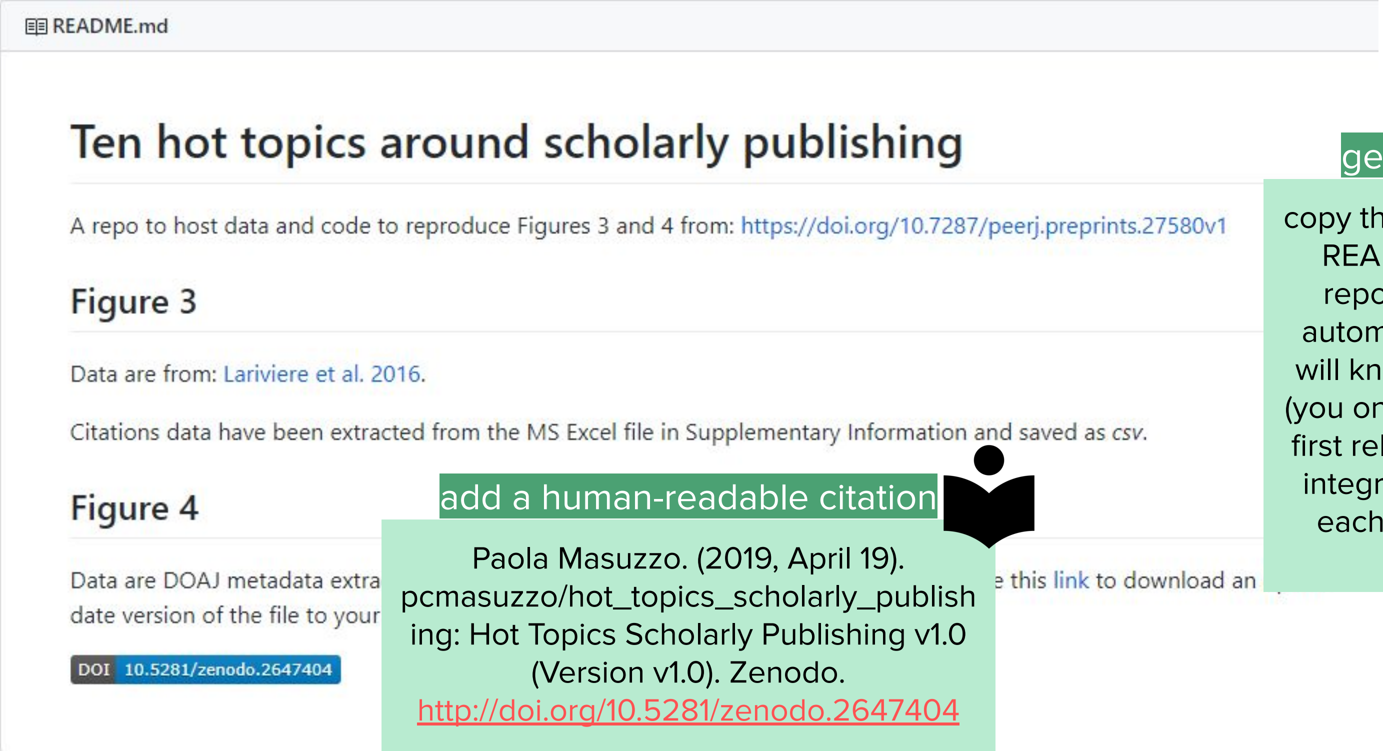


April 19, 2019 (v1.0) Software Open Access

pcmasuzzo/hot_topics_scholarly_publishing: Hot Topics Scholarly Publishing v1.0

Created Apr 19, 2019 10:36:43 AM, modified Apr 19, 2019 1:33:11 PM

# best practices to make your code citable



README.md

## Ten hot topics around scholarly publishing

A repo to host data and code to reproduce Figures 3 and 4 from: https://doi.org/10.7287/peerj.preprints.27580v1

### Figure 3

Data are from: Lariviere et al. 2016.

Citations data have been extracted from the MS Excel file in Supplementary Information and saved as *csv*.

### Figure 4

Data are DOAJ metadata extra e this link to download an
date version of the file to your

DOI 10.5281/zenodo.2647404

add a human-readable citation

Paola Masuzzo. (2019, April 19). pcmasuzzo/hot_topics_scholarly_publishing: Hot Topics Scholarly Publishing v1.0 (Version v1.0). Zenodo. http://doi.org/10.5281/zenodo.2647404

get a nice DOI badge

copy the URL for the DOI into the README file of your GitHub repo: the DOI badge will be automatically added and users will know how to cite your code (you only need to do this for your first release, the GitHub/Zenodo integration will assign a DOI to each version/release of your project repository)

# OpenRefine

working with (messy) data in a reproducible way

# OpenRefine: working with messy data

OpenRefine

A free, open source, powerful tool for working with messy data

https://openrefine.org/ ➜ download & install OpenRefine

(this might take a couple of minutes)

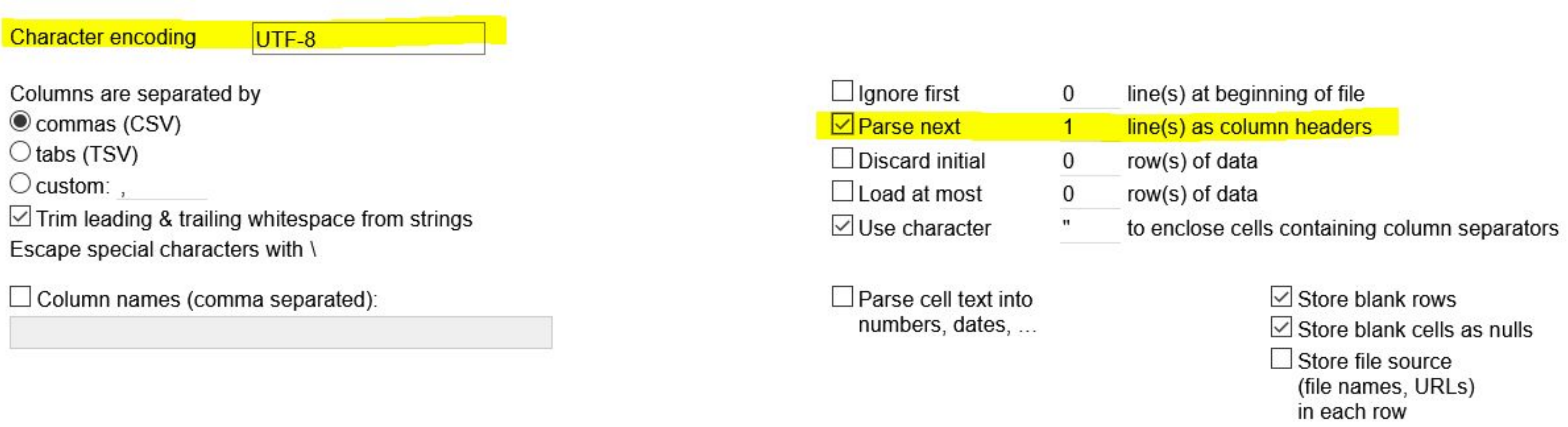
note ➜ a Java Runtime Environment is needed, if you do not have one installed, get it here: https://www.java.com/en/

- OpenRefine is ‘a tool for working with messy data’
- it uses your web browser as a graphical interface
- it works best with data in a simple tabular format and is ideal to fix inconsistencies in a data set (*e.g.*, standardizing date formatting)
- it can help you accomplish several tasks, such as split data up into more granular parts, match local data up to other data sets, enhance a data set with data from other sources

# OpenRefine: create a project

DOAJ DIRECTORY OF OPEN ACCESS JOURNALS

- OpenRefine interface runs at: http://127.0.0.1:3333/
- create a project by importing a file (for our exercises, download the DOAJ journals metadata csv file → https://doaj.org/csv)



Character encoding UTF-8

Columns are separated by
● commas (CSV)
○ tabs (TSV)
○ custom: ,
☑ Trim leading & trailing whitespace from strings
Escape special characters with \

□ Column names (comma separated):

□ Ignore first 0 line(s) at beginning of file
☑ Parse next 1 line(s) as column headers
□ Discard initial 0 row(s) of data
□ Load at most 0 row(s) of data
☑ Use character " to enclose cells containing column separators

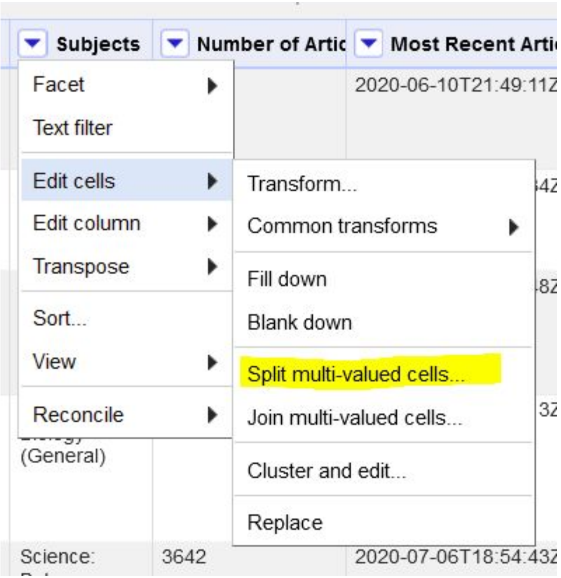
□ Parse cell text into numbers, dates, ...

☑ Store blank rows
☑ Store blank cells as nulls
□ Store file source (file names, URLs) in each row

# OpenRefine: split & merge

- rows & records views
  - in rows mode, each row represents a single record in the dataset
  - in records mode, OpenRefine links together multiple rows as belonging to the same record
- try out split cells & merge cells (➔ edit cells / split multi-valued cells)

15198 rows



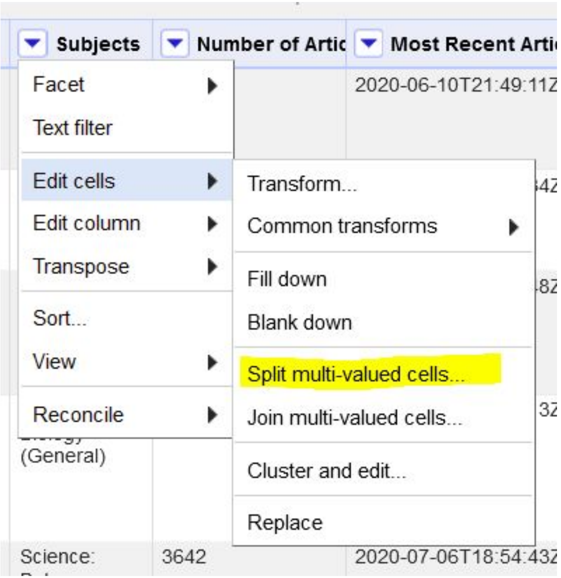
15921 rows

15198 records

# OpenRefine: split & merge

- rows & records views
  - in rows mode, each row represents a single record in the dataset
  - in records mode, OpenRefine links together multiple rows as belonging to the same record
- try out split cells & merge cells (→ edit cells / split multi-valued cells)

**15198 rows**



**15921 rows**

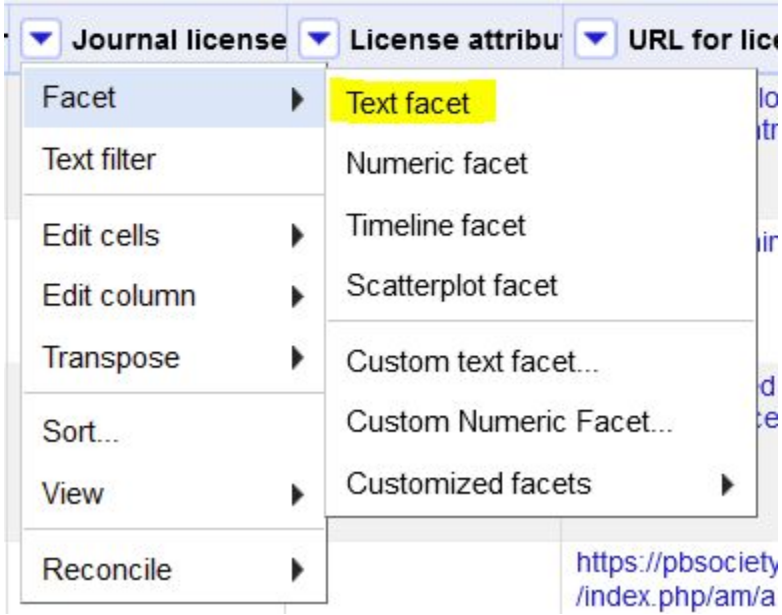
**15198 records**

**best practice** → when creating a spreadsheet with multi-valued cells, it is important to choose a separator that will never appear in the cell values themselves; this is why the pipe character ( | ) is often a good choice, while commas, colons and semi-colons should be avoided as separators

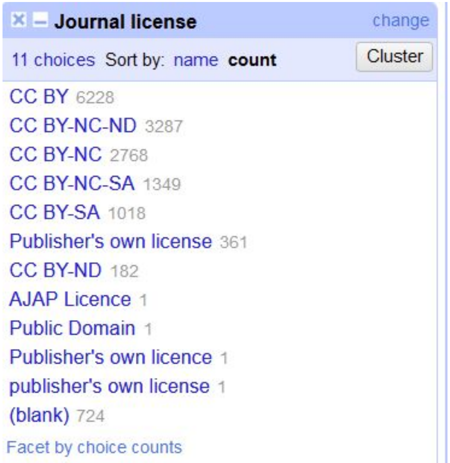
# OpenRefine: facets

- facets are one of the most useful features of OpenRefine and can help in both getting an overview of the data and to improve the consistency of the data
- a facet groups all the values that appear in a column, and then allows you to filter the data by these values and edit values across many records at the same time
- the simplest type of facet is a text facet: it groups all the text values in a column and lists each value with the number of records it appears in
- the facet information always appears in the left hand panel in the OpenRefine interface

# OpenRefine: facets



what is the most mentioned license in the DOAJ?



**Journal license** change

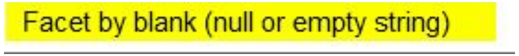
11 choices Sort by: name **count** Cluster

| | |
|---|---|
| CC BY | 6228 |
| CC BY-NC-ND | 3287 |
| CC BY-NC | 2768 |
| CC BY-NC-SA | 1349 |
| CC BY-SA | 1018 |
| Publisher's own license | 361 |
| CC BY-ND | 182 |
| AJAP Licence | 1 |
| Public Domain | 1 |
| Publisher's own licence | 1 |
| publisher's own license | 1 |
| (blank) | 724 |

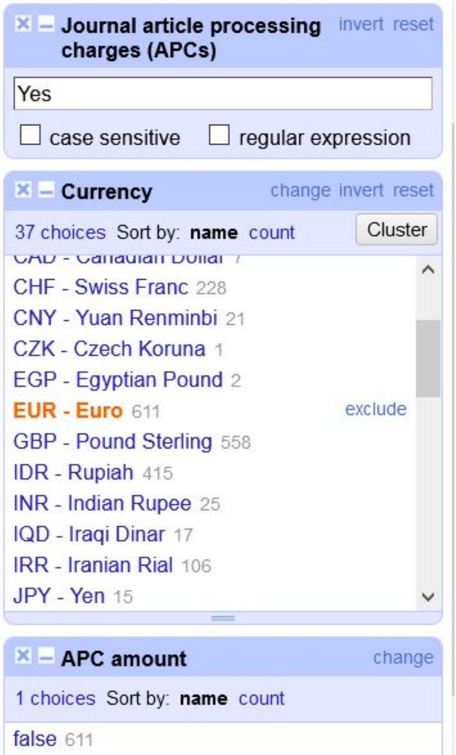
Facet by choice counts

# OpenRefine: filters & transformations

- filter journals which charge an APC (text filter = Yes)
- use a text facet on APC currency and include only EUR
- use a customized facet - facet by blank - to see how many of these journals have/have not reported the APC amount (this is very useful to look for *missing data*)



Facet by blank (null or empty string)



Yes

□ case sensitive □ regular expression

37 choices Sort by: name count

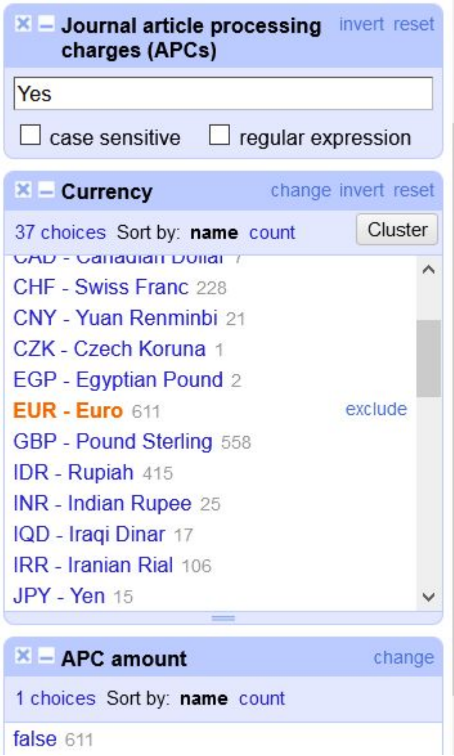
| | |
|---|---|
| CAD - Canadian Dollar | 7 |
| CHF - Swiss Franc | 228 |
| CNY - Yuan Renminbi | 21 |
| CZK - Czech Koruna | 1 |
| EGP - Egyptian Pound | 2 |
| EUR - Euro | 611 |
| GBP - Pound Sterling | 558 |
| IDR - Rupiah | 415 |
| INR - Indian Rupee | 25 |
| IQD - Iraqi Dinar | 17 |
| IRR - Iranian Rial | 106 |
| JPY - Yen | 15 |

1 choices Sort by: name count

false 611

# OpenRefine: filters & transformations

- filter journals which charge an APC (text filter = Yes)
- use a text facet on APC currency and include only EUR
- use a customized facet - facet by blank - to see how many of these journals have/have not reported the APC amount (this is very useful to look for *missing data*)



Journal article processing charges (APCs)

invert reset

Yes

□ case sensitive □ regular expression

Currency

change invert reset

37 choices Sort by: **name** count Cluster

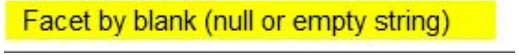
| | |
|---|---|
| CAD - Canadian Dollar | 7 |
| CHF - Swiss Franc | 228 |
| CNY - Yuan Renminbi | 21 |
| CZK - Czech Koruna | 1 |
| EGP - Egyptian Pound | 2 |
| **EUR - Euro** | 611 exclude |
| GBP - Pound Sterling | 558 |
| IDR - Rupiah | 415 |
| INR - Indian Rupee | 25 |
| IQD - Iraqi Dinar | 17 |
| IRR - Iranian Rial | 106 |
| JPY - Yen | 15 |

APC amount

change

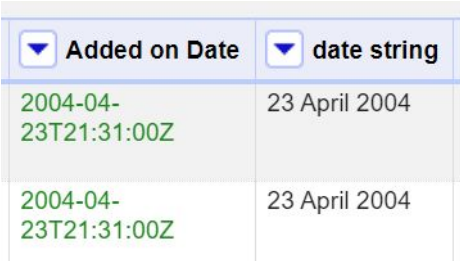
1 choices Sort by: **name** count

| | |
|---|---|
| false | 611 |



Facet by blank (null or empty string)

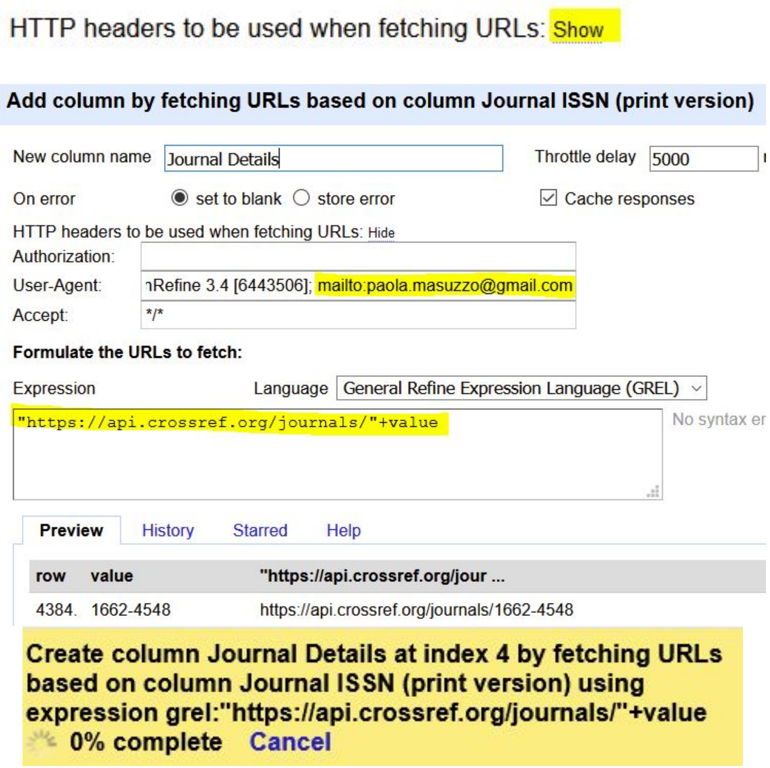
- common transformations
  - ○ transform column to date → value.toDate("dd/MM/yyyy")
  - ○ edit column → add column based on this column → value.toString("dd MMMM yyyy")



| Added on Date | date string |
|---|---|
| 2004-04-23T21:31:00Z | 23 April 2004 |
| 2004-04-23T21:31:00Z | 23 April 2004 |

# OpenRefine: advanced functions

- text filter Journal Title on *Frontiers in Neuroscience* (might need to reset other filters)
  - on the ISSN column → edit column → add column by fetching URLs
  - we're going to use the CrossRef REST API https://github.com/CrossRef/rest-api-doc
- expression → “https://api.crossref.org/journals/"+value
- this will generate a new column named Journal Details → this column will be a JSON object
- we will then use this object to get the Publisher Name (amongst other details)



HTTP headers to be used when fetching URLs: Show

**Add column by fetching URLs based on column Journal ISSN (print version)**

New column name: Journal Details

Throttle delay: 5000

On error: ● set to blank ○ store error

☑ Cache responses

HTTP headers to be used when fetching URLs: Hide

Authorization:

User-Agent: ıRefine 3.4 [6443506]; mailto:paola.masuzzo@gmail.com

Accept: */*

**Formulate the URLs to fetch:**

Expression

Language: General Refine Expression Language (GREL)

```
"https://api.crossref.org/journals/"+value
```

No syntax er

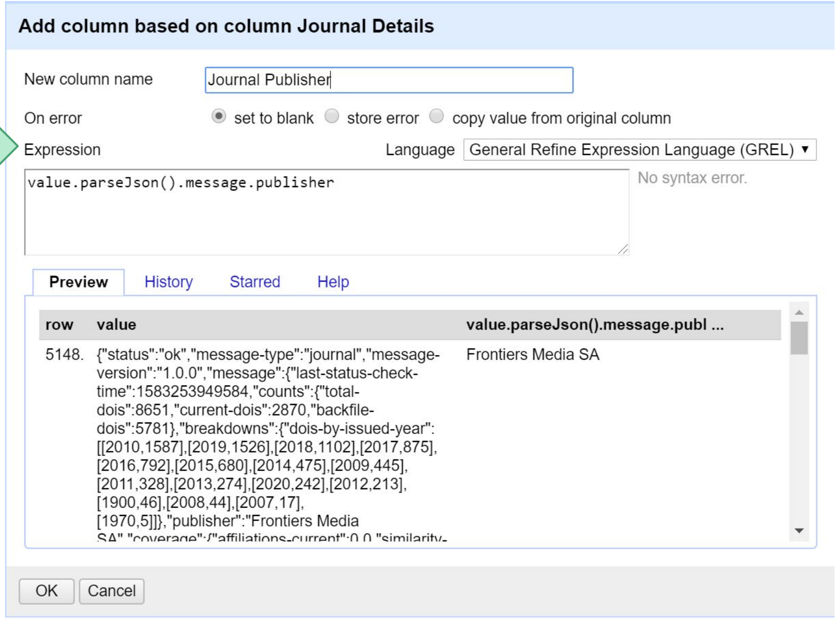
| row | value | "https://api.crossref.org/jour ... |
|---|---|---|
| 4384. | 1662-4548 | https://api.crossref.org/journals/1662-4548 |

**Create column Journal Details at index 4 by fetching URLs based on column Journal ISSN (print version) using expression grel:"https://api.crossref.org/journals/"+value**

**0% complete** **Cancel**

# OpenRefine: advanced functions

- column Journal Details ➔ edit column ➔ add column based on this column ➔ value.parseJson().message.publisher



**Add column based on column Journal Details**

New column name: Journal Publisher

On error: ● set to blank ○ store error ○ copy value from original column

Expression — Language: General Refine Expression Language (GREL) ▾

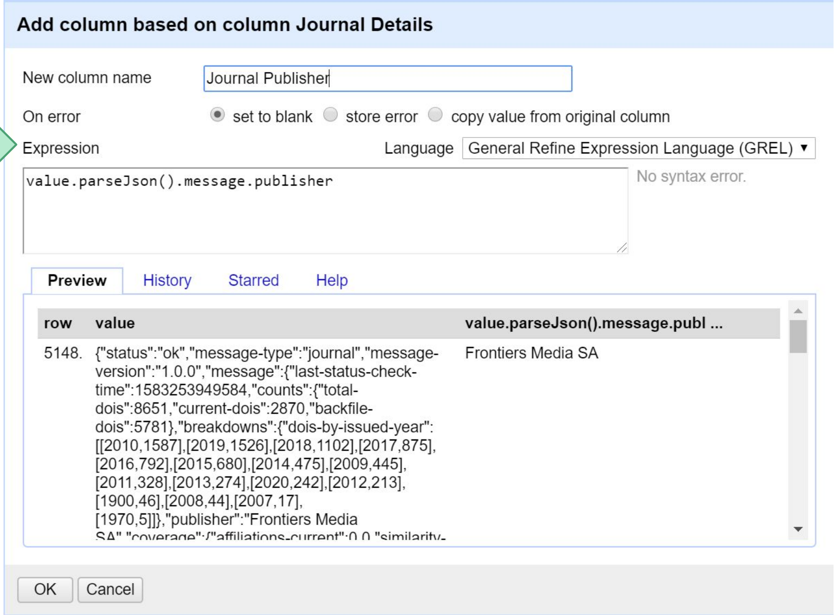
```
value.parseJson().message.publisher
```

No syntax error.

| row | value | value.parseJson().message.publ ... |
|---|---|---|
| 5148. | {"status":"ok","message-type":"journal","message-version":"1.0.0","message":{"last-status-check-time":1583253949584,"counts":{"total-dois":8651,"current-dois":2870,"backfile-dois":5781},"breakdowns":{"dois-by-issued-year":[[2010,1587],[2019,1526],[2018,1102],[2017,875],[2016,792],[2015,680],[2014,475],[2009,445],[2011,328],[2013,274],[2020,242],[2012,213],[1900,46],[2008,44],[2007,17],[1970,5]]},"publisher":"Frontiers Media SA","coverage":{"affiliations-current":0.0,"similarity- | Frontiers Media SA |

# OpenRefine: advanced functions

- column Journal Details ➔ edit column ➔ add column based on this column ➔ value.parseJson().message.publisher



**Add column based on column Journal Details**

New column name: Journal Publisher

On error: ● set to blank ○ store error ○ copy value from original column

Expression — Language: General Refine Expression Language (GREL) ▾

```
value.parseJson().message.publisher
```

No syntax error.

| row | value | value.parseJson().message.publ ... |
|---|---|---|
| 5148. | {"status":"ok","message-type":"journal","message-version":"1.0.0","message":{"last-status-check-time":1583253949584,"counts":{"total-dois":8651,"current-dois":2870,"backfile-dois":5781},"breakdowns":{"dois-by-issued-year":[[2010,1587],[2019,1526],[2018,1102],[2017,875],[2016,792],[2015,680],[2014,475],[2009,445],[2011,328],[2013,274],[2020,242],[2012,213],[1900,46],[2008,44],[2007,17],[1970,5]]},"publisher":"Frontiers Media SA","coverage":{"affiliations-current":0.0,"similarity- | Frontiers Media SA |

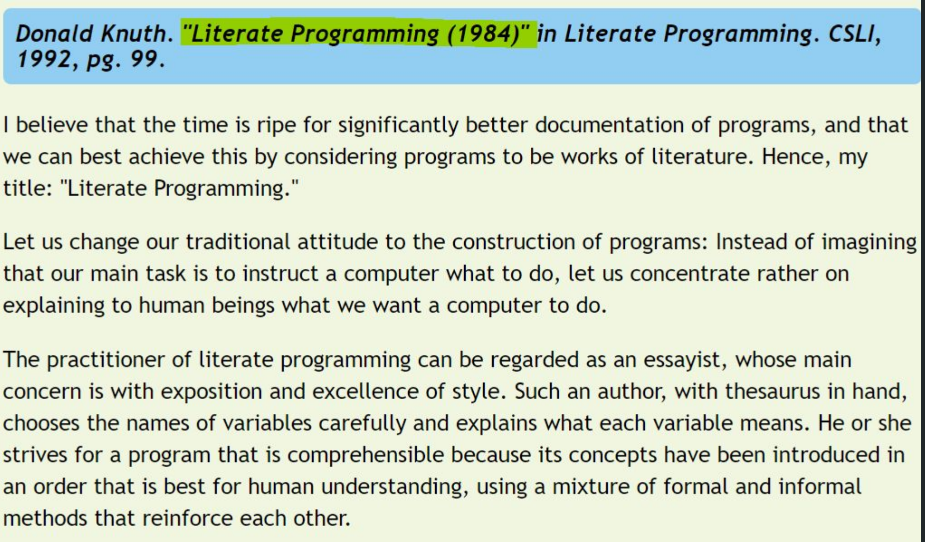
* plugins available for OpenRefine ➔ http://openrefine.org/download.html (e.g. FAIR metadata)

* VIB plugins ➔ https://www.bits.vib.be/index.php/software-overview/openrefine (history tools, pivot tool, cross function gui, and scatterplot tool using D3)

* more resources/tutorial ➔ https://libjohn.github.io/openrefine/preamble.html

# literate programming

## and reproducible research

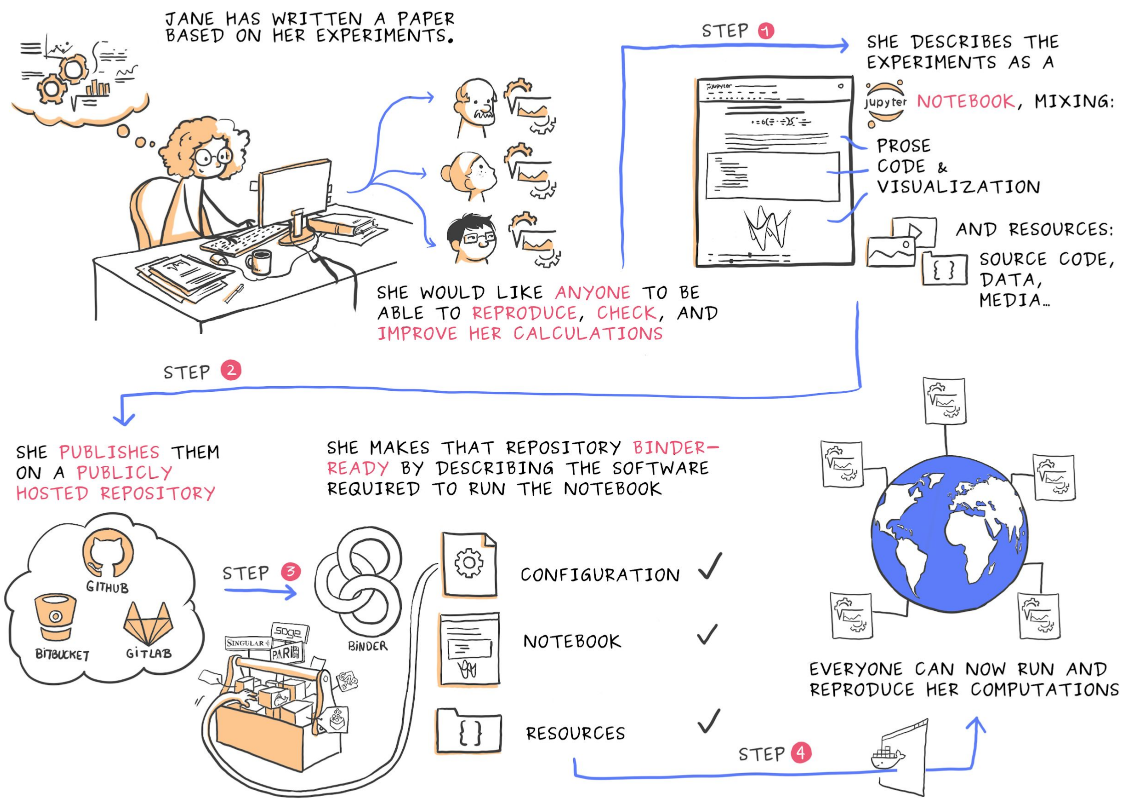


*Donald Knuth. "Literate Programming (1984)" in Literate Programming. CSLI, 1992, pg. 99.*

I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. Hence, my title: "Literate Programming."

Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

The practitioner of literate programming can be regarded as an essayist, whose main concern is with exposition and excellence of style. Such an author, with thesaurus in hand, chooses the names of variables carefully and explains what each variable means. He or she strives for a program that is comprehensible because its concepts have been introduced in an order that is best for human understanding, using a mixture of formal and informal methods that reinforce each other.


JANE HAS WRITTEN A PAPER BASED ON HER EXPERIMENTS.
SHE WOULD LIKE ANYONE TO BE ABLE TO REPRODUCE, CHECK, AND IMPROVE HER CALCULATIONS
STEP 1
SHE DESCRIBES THE EXPERIMENTS AS A jupyter NOTEBOOK, MIXING:
PROSE
CODE &
VISUALIZATION
AND RESOURCES:
SOURCE CODE,
DATA,
MEDIA...
STEP 2
SHE PUBLISHES THEM ON A PUBLICLY HOSTED REPOSITORY
GITHUB
BITBUCKET
GITLAB
STEP 3
SHE MAKES THAT REPOSITORY BINDER-READY BY DESCRIBING THE SOFTWARE REQUIRED TO RUN THE NOTEBOOK
BINDER
SINGULAR
sage
PARI
CONFIGURATION ✓
NOTEBOOK ✓
RESOURCES ✓
STEP 4
EVERYONE CAN NOW RUN AND REPRODUCE HER COMPUTATIONS

literate programming = **human-readable** (text) + **machine-readable** (code)

Publishing reproducible logbooks

# literate programming and Binder

- literate programming is  a paradigm in which a computer program is given an explanation of its logic in a natural language (*e.g.* English), together with snippets of source code. The approach is used in scientific computing and in data science routinely for **reproducible research** and **open access** purposes

- **Binder** is an open-source, free service that hosts fully functional Jupyter Notebooks, which you can open in an executable environment, making the code immediately reproducible by anyone, anywhere

Reproducible, sharable, interactive computing environments

Project Jupyter | The Binder Project

# why do we need all of this?

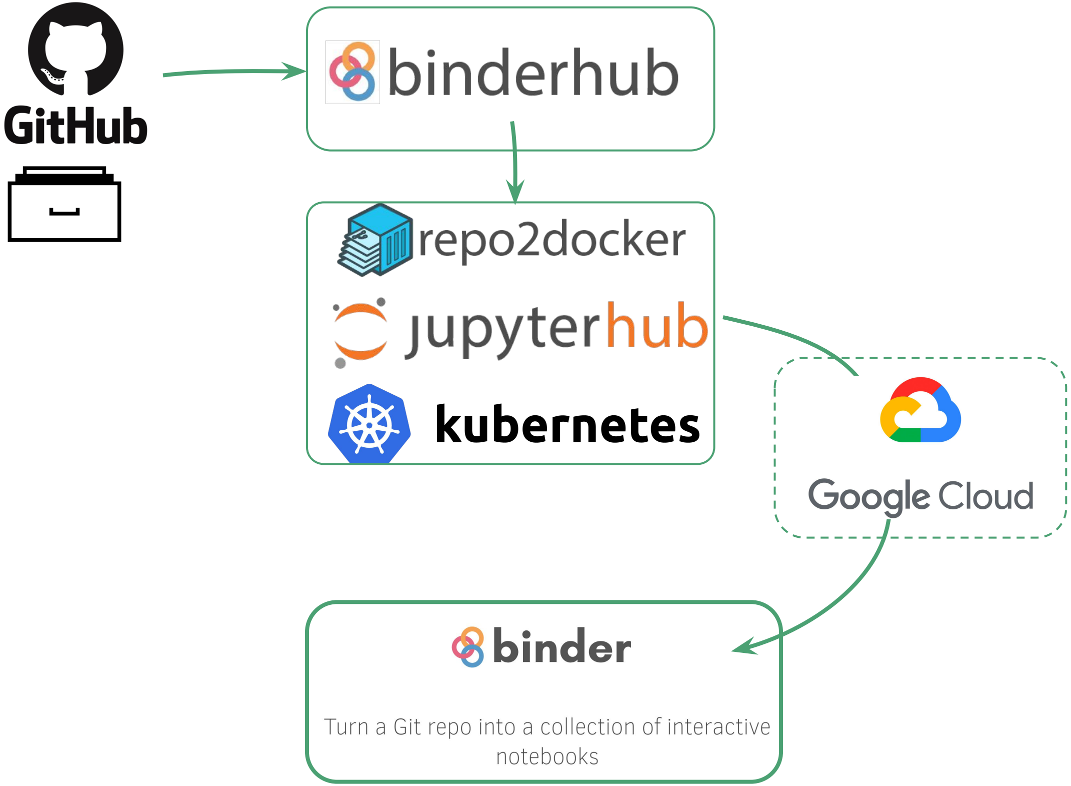
to be able to run code, we need

- hardware to run the code on
- software, including:
  - the code itself
  - the programming language (e.g. Python, R, Julia, and so on)
  - relevant packages (e.g. pandas, matplotlib, tidyverse, ggplot)

# why do we need all of this?

to be able to run code, we need

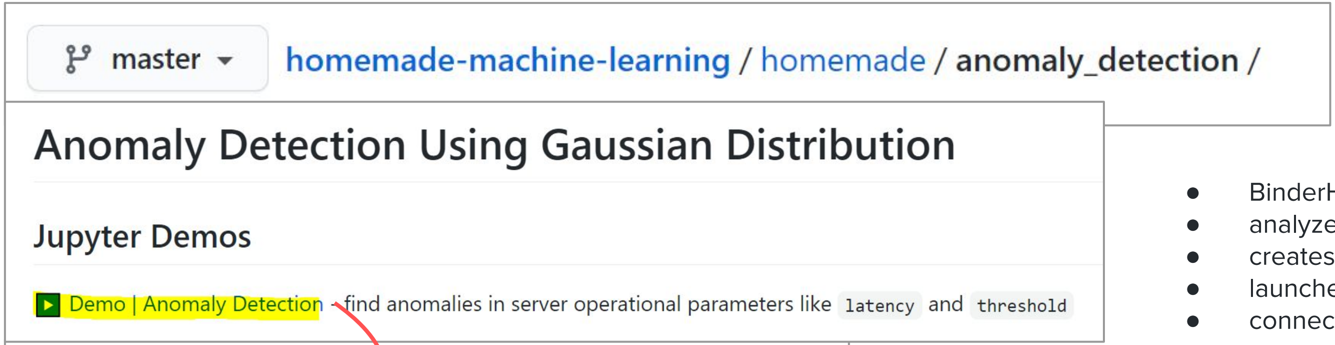
- hardware to run the code on
- software, including:
    - the code itself
    - the programming language (e.g. Python, R, Julia, and so on)
    - relevant packages (e.g. pandas, matplotlib, tidyverse, ggplot)



Turn a Git repo into a collection of interactive notebooks

# an example

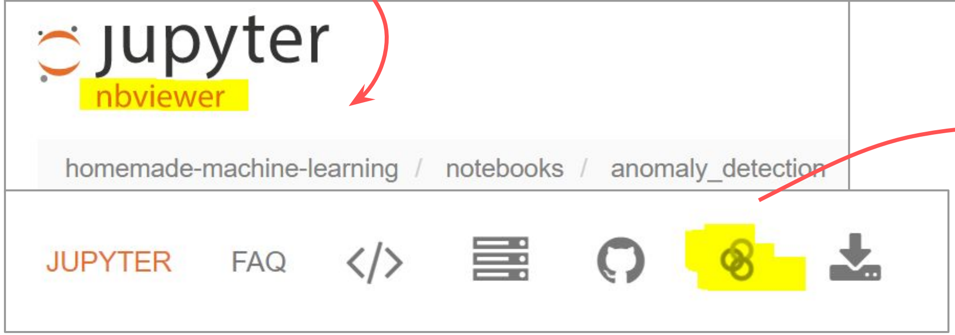
GitHub repo: https://github.com/trekhleb/homemade-machine-learning


master homemade-machine-learning / homemade / anomaly_detection /

Anomaly Detection Using Gaussian Distribution

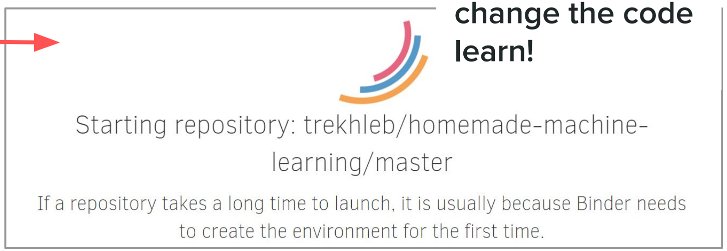
Jupyter Demos

▶ Demo | Anomaly Detection - find anomalies in server operational parameters like `latency` and `threshold`



- BinderHub fetches the repo from GitHub
- analyzes the content
- creates a Docker file based on the content
- launches the Docker image in the Cloud
- connects you to this mage via the browser

**reproduce experiment change the code learn!**


Starting repository: trekhleb/homemade-machine-learning/master

If a repository takes a long time to launch, it is usually because Binder needs to create the environment for the first time.

# let’s binderize a repository

- go to https://mybinder.org and type the URL of your repo into the "GitHub repo or URL" box
  it should look like this:
  https://github.com/your-username/my-first-binder (I’ll use this repo of mine:
  https://github.com/pcmasuzzo/modelli_epidemiologici)

- as you type, the webpage generates a link in the "Copy the URL below..." box;
  it should look like this: https://mybinder.org/v2/gh/your-username/my-first-binder/master
  (for me it’s: https://mybinder.org/v2/gh/pcmasuzzo/modelli_epidemiologici.git/master)

- copy it, open a new browser tab and visit that URL;
  you will see a "spinner" as Binder launches the repo;
  if everything ran smoothly, you'll see a Jupyter Notebook interface

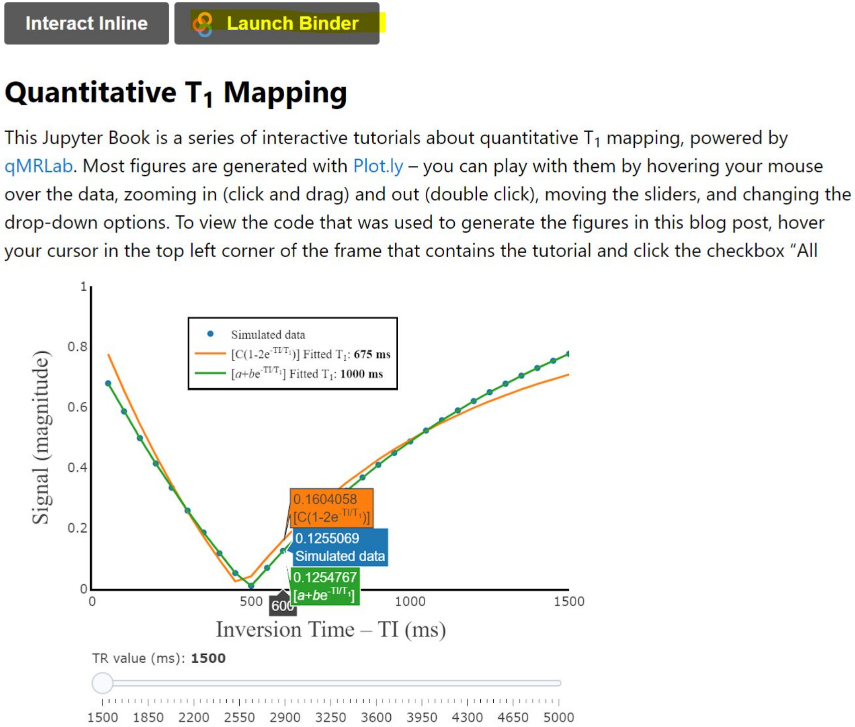
Starting repository:
pcmasuzzo/modelli_epidemiologici.git/master

note: pushing changes back to the GitHub repo through the container is not possible with Binder

- show a binder badge via the README file

http://bit.ly/zero-to-binder

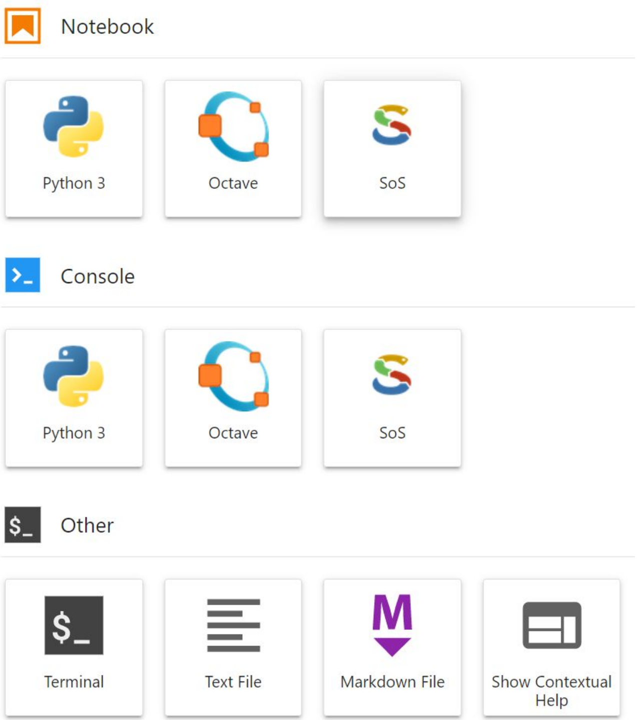
# Neurolibre: interactive & reproducible neuroscience



### Quantitative $T_1$ Mapping

This Jupyter Book is a series of interactive tutorials about quantitative $T_1$ mapping, powered by qMRLab. Most figures are generated with Plot.ly – you can play with them by hovering your mouse over the data, zooming in (click and drag) and out (double click), moving the sliders, and changing the drop-down options. To view the code that was used to generate the figures in this blog post, hover your cursor in the top left corner of the frame that contains the tutorial and click the checkbox "All

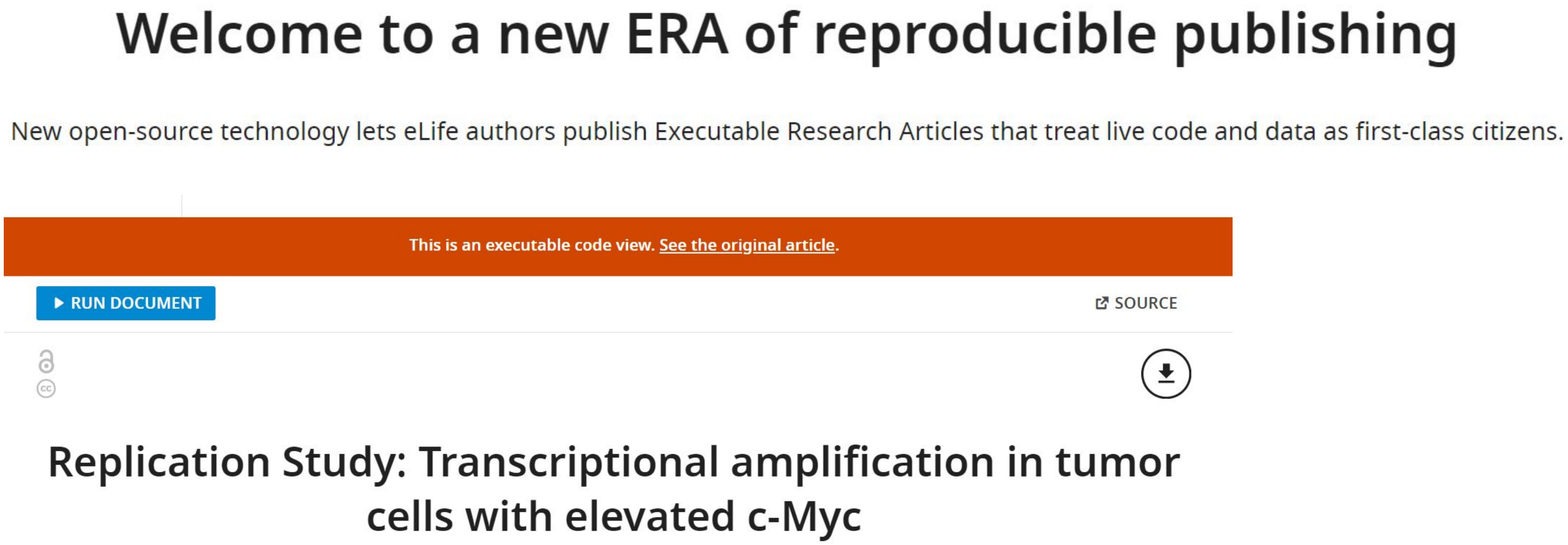
TR value (ms): **1500**



NeuroLibre - SoS - Script of Script

# towards reproducible publishing

eLIFE



## Welcome to a new ERA of reproducible publishing

New open-source technology lets eLife authors publish Executable Research Articles that treat live code and data as first-class citizens.

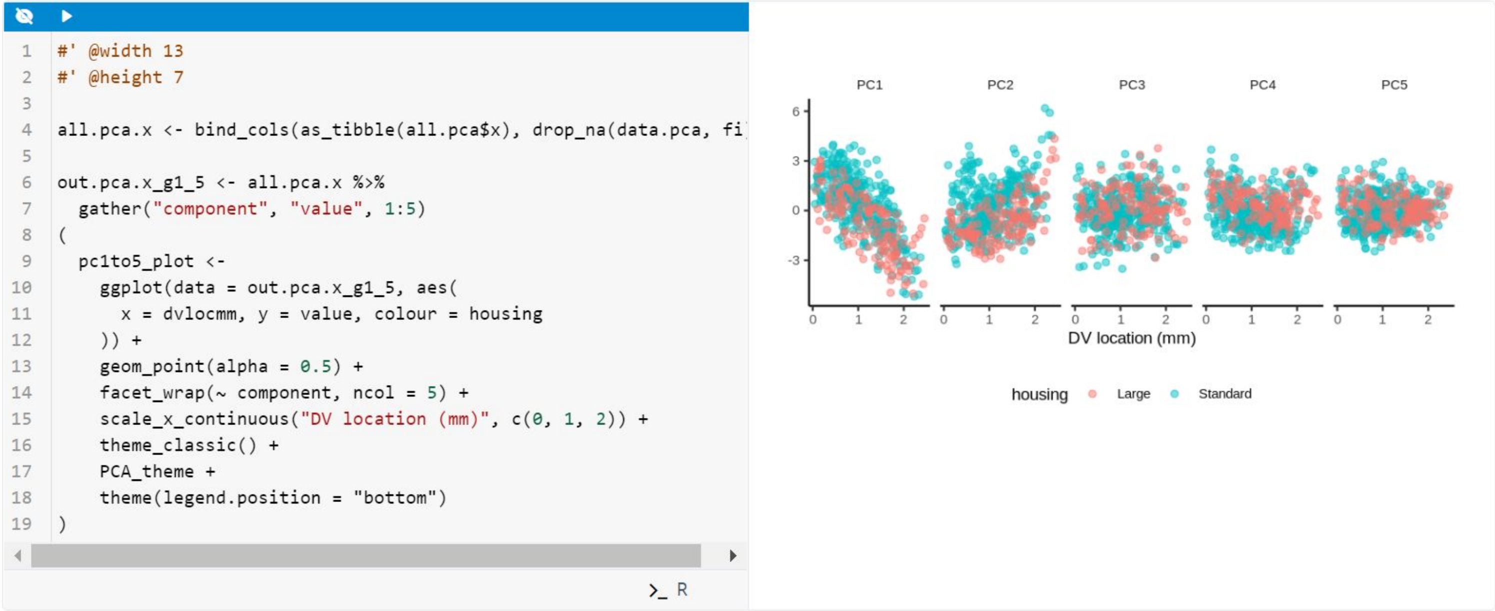
This is an executable code view. See the original article.

### Replication Study: Transcriptional amplification in tumor cells with elevated c-Myc

Welcome to a new ERA of reproducible publishing - https://elifesciences.org/articles/30274/executable

# re-execute, validate, reproduce… learn!

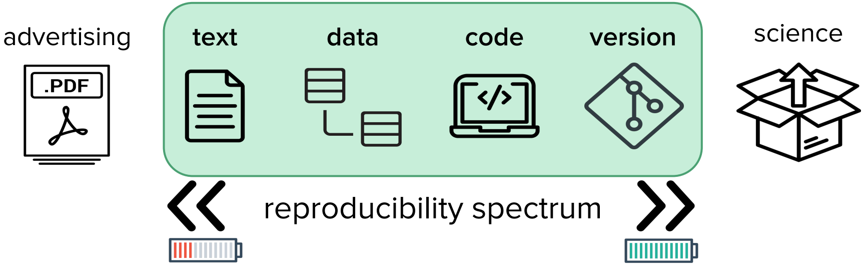
**Figure 7E**

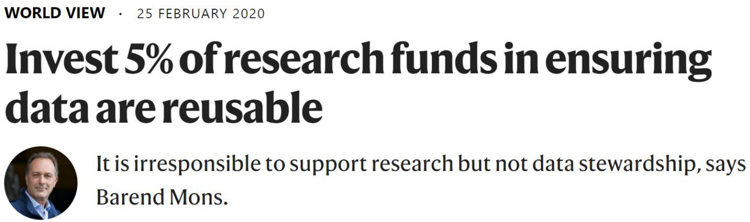


```r
#' @width 13
#' @height 7

all.pca.x <- bind_cols(as_tibble(all.pca$x), drop_na(data.pca, fi

out.pca.x_g1_5 <- all.pca.x %>%
  gather("component", "value", 1:5)
(
  pc1to5_plot <-
    ggplot(data = out.pca.x_g1_5, aes(
      x = dvlocmm, y = value, colour = housing
    )) +
    geom_point(alpha = 0.5) +
    facet_wrap(~ component, ncol = 5) +
    scale_x_continuous("DV location (mm)", c(0, 1, 2)) +
    theme_classic() +
    PCA_theme +
    theme(legend.position = "bottom")
)
```

[Welcome to a new ERA of reproducible publishing](https://elifesciences.org/articles/30274/executable) - https://elifesciences.org/articles/30274/executable

# reproducible? not without people





WORLD VIEW · 25 FEBRUARY 2020

**Invest 5% of research funds in ensuring data are reusable**

It is irresponsible to support research but not data stewardship, says Barend Mons.

data stewards and managers

software developers

(open) champions

people who believe in people

[Cultural obstacles to research data management and sharing at TU Delft](), Plomp et al., 2019

# Thank you

@pcmasuzzo

paola.masuzzo@gmail.com