

Research data management and publishing at your fingertips!

fairly Toolset

National Open Science Festival, 31/08/2023
Erasmus University Rotterdam



Serkan
Girgin



Manuel
Garcia Alvarez



Jose
Urra Llanusa



Magno
Barreto de Araujo



We usually publish research data at **the last minute**

- Research data are produced during the whole research lifecycle.
- Data publication and sharing happens mostly at the end.
- Data published in a hurry lack important supplementary information and metadata limiting reusability.
- On the other hand, periodic high-quality data publication takes time and effort.



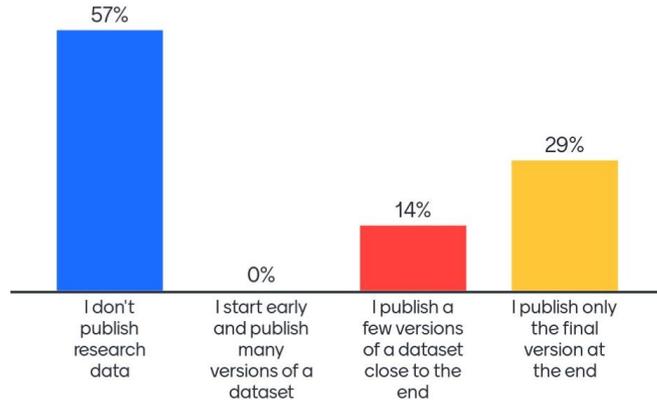
We cannot publish research data directly from our **digital research environments**

- Digital research environments facilitate research data production by providing (interactive) analysis tools.
- They are well connected to some research infrastructure, e.g. code repositories.
- However, their interoperability with research data repositories is weak.

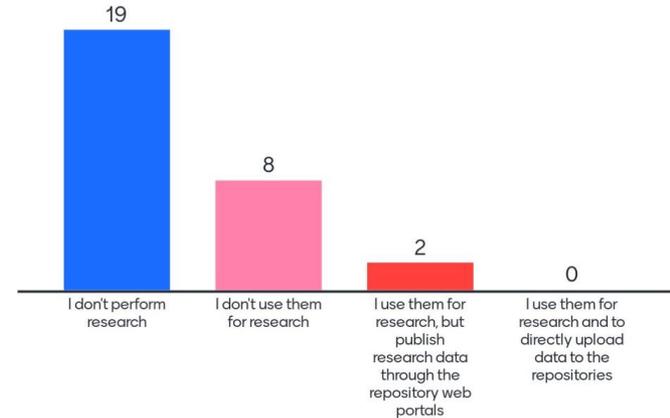


What do **you** think?

When do you publish your research data?



How do you use digital research environments?



Seamless integration of research environments and data repositories may facilitate data management practices

How to integrate?

- Local datasets with data and metadata
- Direct and simple data transfer
- Less data input through forms
- Onsite quality checks

What are the benefits?

- Less time and effort for research data publishing
- More frequent data sharing during research lifecycle
- Improved quality of shared research data



Our project aimed at enabling **local management** and **easy publishing** of research data

- Design of a methodology to integrate research environments to research data repositories
- Development of a modular open-source software tool implementing the methodology
- Demonstration at [4TU.ResearchData](#) and [ITC Geospatial Computing Platform](#)
- Provision of technical documentation and end-user training

Funded by the [NWO Open Science Fund](#), File No. 203.001.114

For more information, please check the project proposal:

DOI [10.5281/zenodo.6026285](https://doi.org/10.5281/zenodo.6026285)



We designed a **three-tier architecture** to serve different needs

1. Python package: **fairly**

- Provides an API to create and manage research data by using Python
- Enables further development by interested parties

2. Command line interface: **fairly CLI**

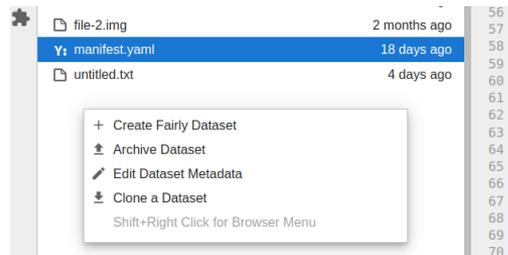
- Provides commands to create and manage research data
- Enables RDM without programming

3. JupyterLab extension: **jupyter-fairly**

- Enables RDM inside a virtual research environment

```
1 import fairly
2
3 # Create a local dataset
4 dataset = fairly.create_dataset('/path/dataset')
5
6 # Set metadata
7 dataset.set_metadata({
8     "title": "My wonderful dataset",
9     "license": "CC BY 4.0",
10    "keywords": ["FAIR", "data"],
11    "authors": [
12
```

```
Administrator: Command Prompt
D:\>fairly clone https://doi.org/10.4121/21588096.v1
Cloning 'Earthquake Precursors detected by convolutional neural
4TU.ResearchData
6 files, 2.16B
Downloading 'results_session1.csv' (524Mb)...
```



The screenshot shows a file browser interface with a context menu open over the file 'manifest.yaml'. The menu items are:

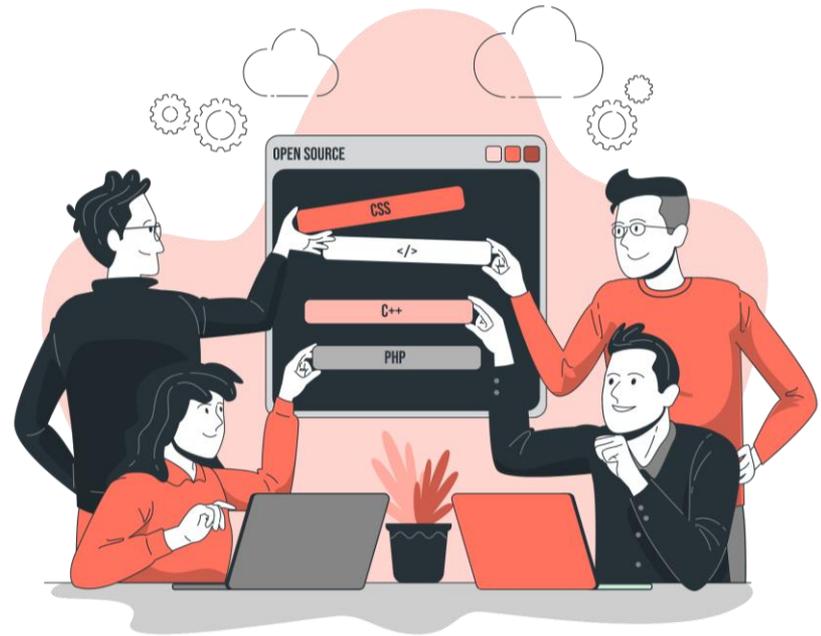
- + Create Fairly Dataset
- 📁 Archive Dataset
- ✎ Edit Dataset Metadata
- 📂 Clone a Dataset

Below the menu items, it says "Shift+Right Click for Browser Menu". The file list shows 'file-2.img' (2 months ago), 'manifest.yaml' (18 days ago), and 'untitled.txt' (4 days ago).

```
56 #
57 # Each array ele
58 #
59 # - name: Name o
60 # - affiliation:
61 # - orcid: ORCID
62 # - gnd: GND ide
63 #
64 # Example:
65 #
66 # [{"name": 'Doe,
67 # 'affiliation':
68 # 'Kowalski, Jaci
69
70
```

Open-source software developed by following best practices

- Open-source Python code
 - Continuous integration (Github) and unit testing (pytest)
 - Documented source code (Google style guidelines)
- Object-oriented modules
 - Task-oriented classes for different components of RDM
 - Easily extendable by implementing abstract classes
- Minimum dependency on 3rd party packages
 - Direct use of repository platform REST APIs



We implemented support for **multiple repository platforms**



The screenshot shows the Zenodo website interface. At the top, there is a blue header with the Zenodo logo, a search bar, and navigation links for "Upload" and "Communities". Below the header, there is a section for "Featured communities" with a "Transform to Open Science" card. The card features the TOPS NASA logo and text describing the mission. Below this, there is a "Recent uploads" section with a card for a "A large-scale COVID-19 Twitter chatter dataset for open scientific research - an international collaboration". The card includes a list of authors and a "View" button. On the right side, there is a "Need help?" section with a "Contact us" button and a list of services provided.

The screenshot shows the figshare website interface. At the top, there is a white header with the figshare logo, a "Browse" button, and a search bar. Below the header, there is a large banner with a colorful molecular structure background. The banner contains the text "store, share, discover research" and "get more citations for all of the outputs of your academic research over 80,000 citations of figshare content to date". Below this, there is a section for "ALSO FOR INSTITUTIONS & PUBLISHERS" and a quote: "figshare wants to open scientific data to the world" with the WIRE logo. At the bottom, there is a caption for the background figure: "The background figure: Comparative model of novel coronavirus 2019-nCoV by Christian Gruber in Virology."

(more is coming soon!)



A rich set of features is available for **efficient** data management

- Quick research dataset cloning
 - One-command retrieval of metadata and all data files by using URL address, DOI, or record identifier
 - Automatic extraction of archived data files (e.g. .zip, .tar.gz)
- Local metadata management
 - Creation and editing of metadata locally by using your favorite text editor or API methods
- Quick dataset publication
 - One-command creation of research data records at online data repositories in a unified way



A rich set of features is available for **smart** data management

- Unattended large dataset uploading
 - Easy uploading of a high number of data files and folders, including large files
 - Automatic creation of archive files (e.g. .zip, .tar.gz) if folders are not supported by the data repository
- Smart dataset synchronization
 - Automatic identification of added, removed, or modified files and folders
 - Upload / download of files and folders only if necessary
 - Easy versioning of datasets in a unified way considering the repository rules **COMING SOON**



How to access a **remote dataset** and **store** it locally?

```
1 import fairly
2
3 # Open a remote dataset
4 dataset = fairly.dataset("doi:10.4121/21588096.v1")
5
6 # Get dataset information
7 dataset.id
8 {'id': '21588096', 'version': '1'}
9
10 dataset.url
11 'https://data.4tu.nl/articles/dataset/.../21588096/1'
12
13 dataset.size
14 33339
15
16 len(dataset.files)
17 6
18
19 dataset.metadata
20 Metadata({'keywords': ['Earthquakes', 'precursor'], ...
21 'online_date': '2022-11-24T07:50:39'})
22
23 # Update metadata
24 dataset.metadata["keywords"] = ["Landslides", "precursor"]
25 dataset.save()
26
27 # Store dataset to a local folder (i.e. clone dataset)
28 local_dataset = dataset.store("/path/dataset")
```

Import **fairly** package

Open dataset by using URL, DOI, or id
(creates a lazy remote dataset object)

Access dataset information
(retrieves information and caches it if possible)

Modify metadata attribute(s)

Save changes

(updates dataset record on the repository)

Store dataset as a local dataset, i.e. clone locally
(downloads and stores metadata + data files)

How to create a **local dataset** and **deposit** it to a repository?

```
1 import fairly
2
3 # Initialize a local dataset
4 dataset = fairly.init_dataset('/path/dataset')
5
6 # Set metadata
7 dataset.set_metadata(
8     title="My dataset",
9     keywords=["FAIR", "data"],
10    authors=[
11        "0000-0002-0156-185X",
12        {"name": "John", "surname": "Doe"}
13    ]
14 )
15 dataset.metadata["license"] = "CC-BY-4.0"
16
17 # Add data files and folders
18 dataset.includes.extend([
19     "README.txt",
20     "*.csv",
21     "train/*.jpg",
22 ])
23
24 # Save dataset
25 dataset.save()
26
27 # Upload to the data repository
28 remote_dataset = dataset.upload("4tu")
```

Import **fairly** package

Initialize a local dataset at the specified folder
(creates a manifest file with the default metadata template)

Set metadata attributes
(attributes can be set together or individually)

Set data files
(files to be included or excluded can be indicated by name or pattern)

Set dataset manifest
(manifest file is updated)

Update dataset to a data repository, i.e. deposit
(creates dataset record, sets metadata, and uploads files and folders)

How to access a **remote dataset** and **store** it locally *easily*?

```
Administrator: Command Prompt
D:\fairly>fairly dataset clone --help

Usage: fairly dataset clone [OPTIONS] [PATH]

Clones a dataset by using its URL address, DOI or ID among other arguments
Examples:
>>> fairly dataset clone <url|doi>
>>> fairly dataset clone https://zenodo.org/record/6026285
>>> fairly dataset clone --url <url> --token <token>
>>> fairly dataset clone <repository> <id>
>>> fairly dataset clone --repo zenodo --id 6026285

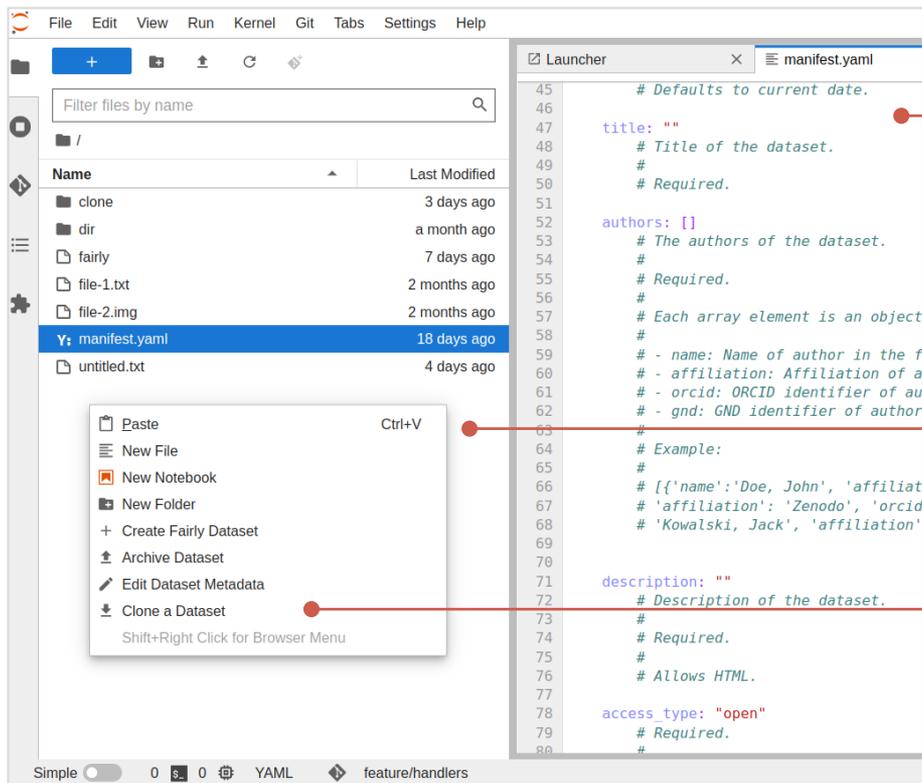
Arguments
  path      [PATH] Path where the dataset will be downloaded [default: ./]

Options
  --url      TEXT  URL option argument
  --token    TEXT  Token option argument
  --repo     TEXT  Repository option argument
  --id       TEXT  ID option argument
  --help     TEXT  Show this message and exit.

D:\fairly>
```

Run **fairly** command line interface

How to access a **remote dataset** and **store** it locally *more easily*?



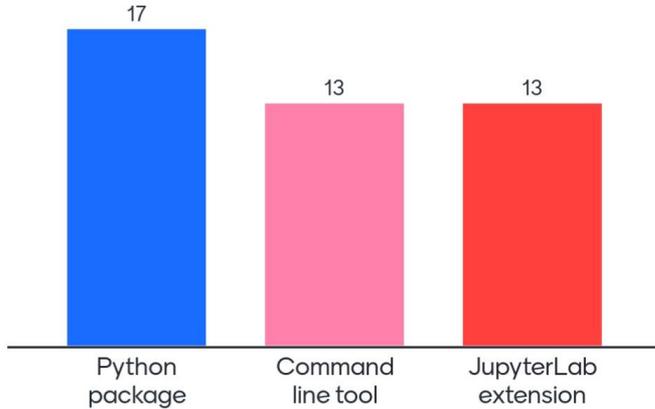
Enable **jupyter-fairly** extension

Open context menu
(right click)

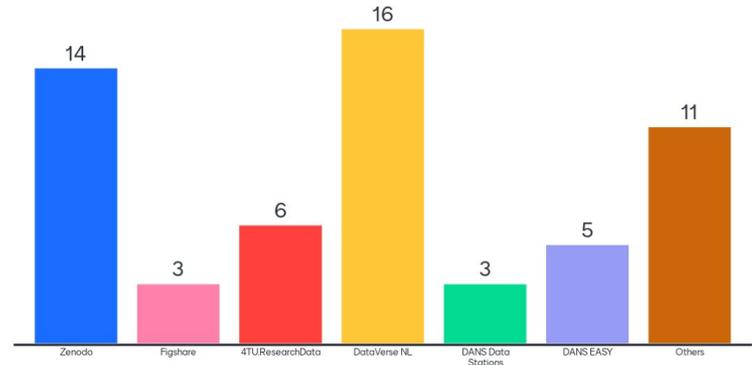
Select 'Clone a Dataset'
Enter URL address or DOI, click 'Ok'

Do they sound **interesting**?

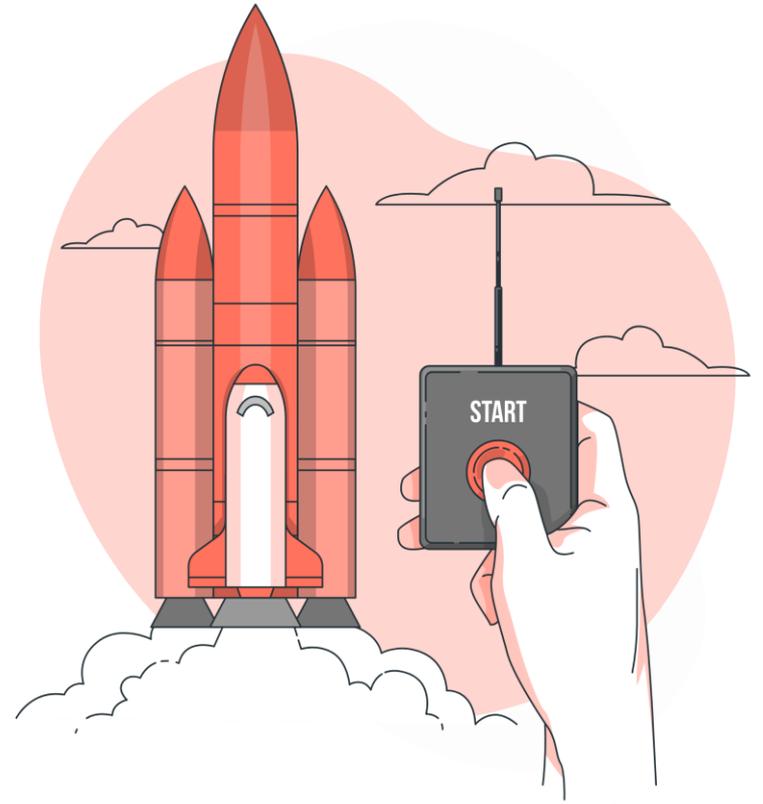
Which fairly tools might be useful for you?



Which research data repositories do you use?



**Let's try
together!**



**Let's discuss
together!**



<https://storyset.com>

Can you think of some potential **use cases**?

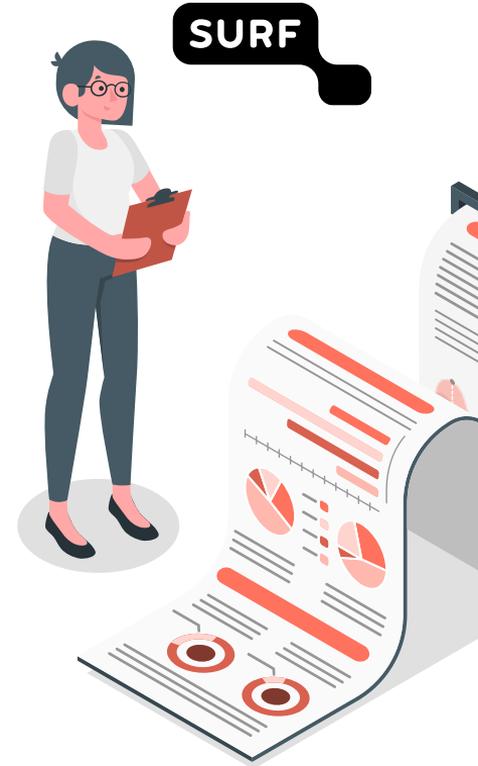
- To have a **standard** research data development workflow, like the use of `git` for research code?
Can help changing research data management culture.
- To deposit **large** datasets?
Can help uploading complex and big datasets.
- To publish updated versions of datasets **periodically**?
Can help automatizing update processes.
- To **embed** research data management into workflows?
Can help developing improved research workflows.



We are developing a platform to provide **analysis-ready exploratory research environment** with data

- Development of an **open-source software** to create and manage interactive computing environments with analysis-ready data
- Development of **template interactive notebooks** to facilitate rapid exploratory data analysis
- Operationalizing of a **prototype platform** – opendataexplorer.org
- **Feasibility and benchmarking study** to use the SURF infrastructure
- Development of the user **documentation and training material**
- Organizing a **training workshop**

Funded by the [SURF DCC Investment for Digital Infrastructure Call](#)



Join us to develop the community in a **sustainable** way

- **Co-design**

Voice your ideas to improve the methodology according to the needs of different research disciplines and communities.

- **Testing**

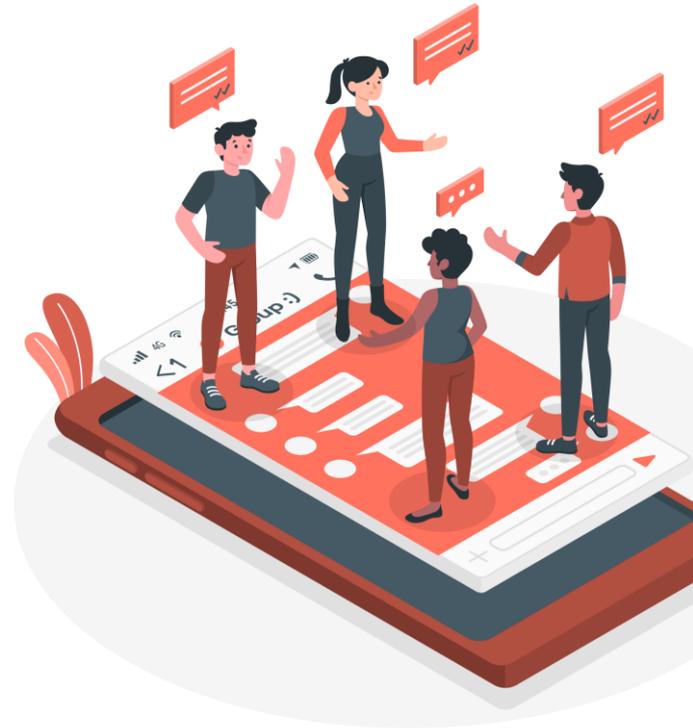
Test the tools and provide feedback to correct issues and improve features.

- **Co-development**

Take part in the co-development effort with your programming and writing skills to improve code and documentation.

- **Visibility**

Promote the tools if you find them useful.



You can start to **contribute now** by filling a short 5-min survey on research data publishing practices

How to integrate research environments to data repositories to facilitate FAIR practices?



Computing environ-
repositories. **Unfor-**
time and effort, es

JupyterFAIR proje
publish it in a data
4TU.ResearchData
(<https://zenodo.org>)

We would like to h
answering the ques

Thanks for your co

*Disclaimer: The survey
graphs of the data coll
research articles and p
collect IP addresses. If y
s.girgin@utwente.nl. Ju*

	Not Important	Slightly Important	Important	Very Important	Essential
Storing metadata in the working environment so that it can be edited directly.	<input type="radio"/>				
Editing metadata with a text editor so that it can be created and updated easily.	<input type="radio"/>				
Importing some metadata available in the documentation (e.g. README file) so that it doesn't need to be entered manually.	<input type="radio"/>				



<https://forms.office.com/r/Xg7RqwsTiS>

Check our online resources to **learn more** about the tools

Code Repositories



<https://github.com/ITC-CRIB/fairly>

<https://github.com/ITC-CRIB/jupyter-fairly>

Watch the repositories for new features and fixes!

User Documentation



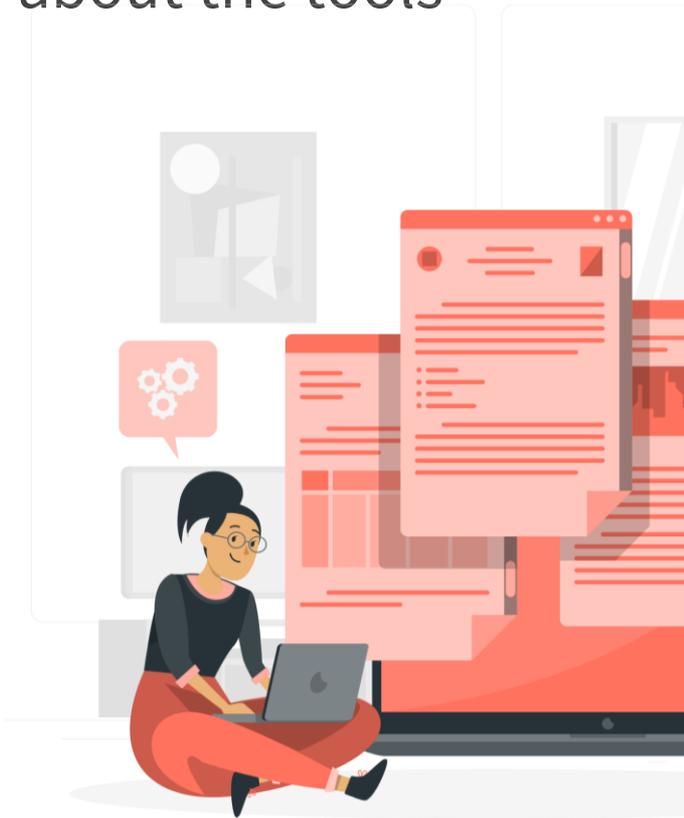
<https://fairly.readthedocs.io/en/latest>

Package Distributions



<https://pypi.org/project/fairly>

<https://pypi.org/project/jupyter-fairly>



Thanks for your time!

Please contact us for further **questions** or **training requests**:



Dr. Ing. Serkan Girgin MSc
s.girgin@utwente.nl



Manuel Garcia Alvarez
m.g.garciaalvarez@tudelft.nl



José Carlos Urra Llanusa
j.c.urrallanusa@tudelft.nl



<https://twitter.com/JupyterFAIR>*

Follow us to get informed on new features and events!