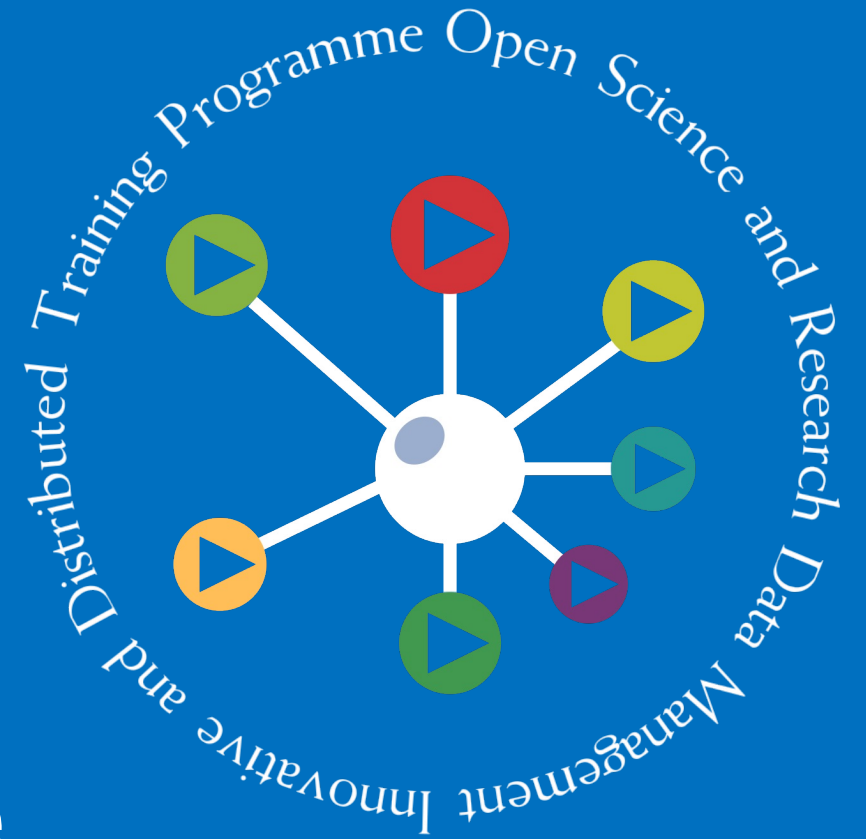
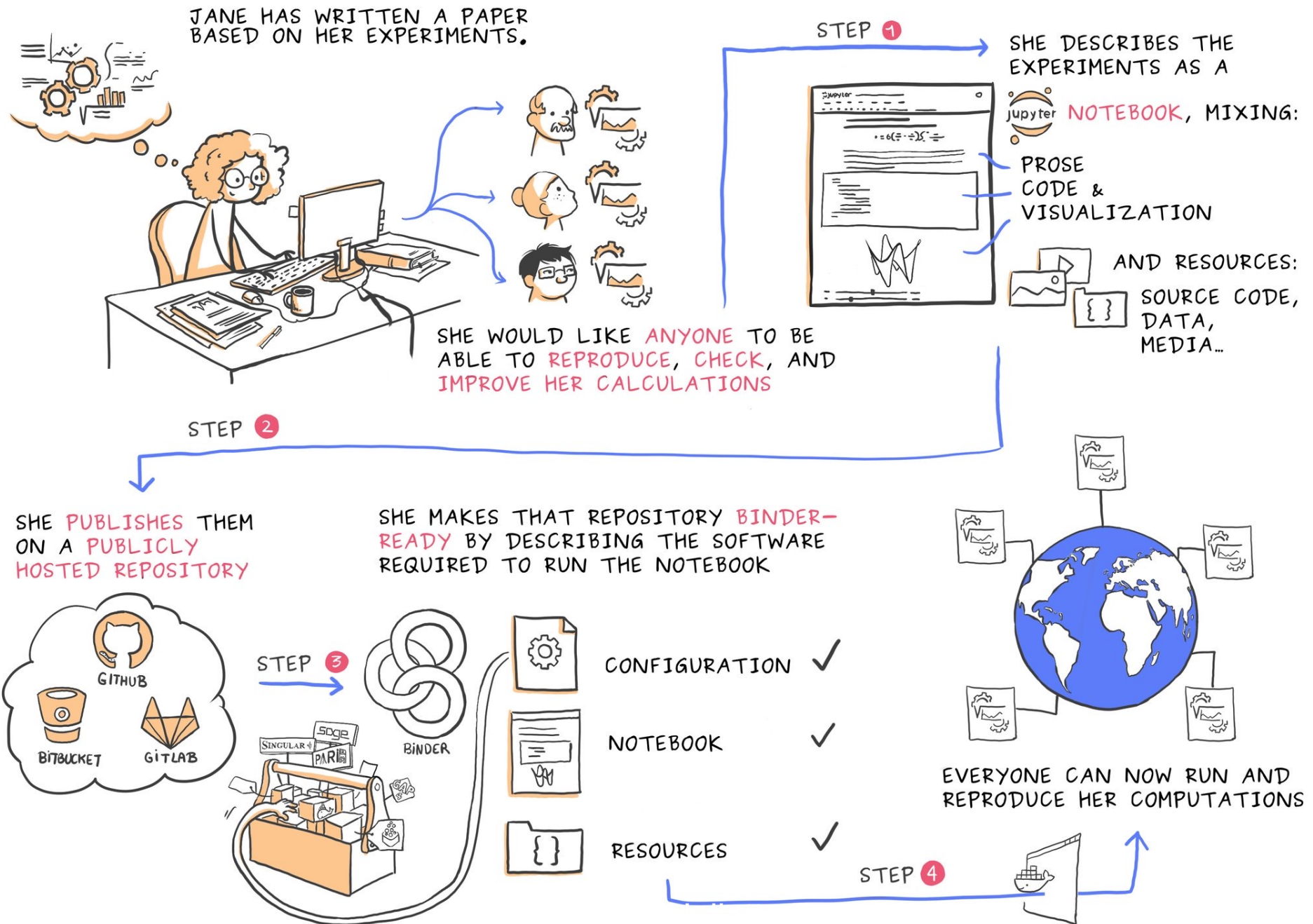


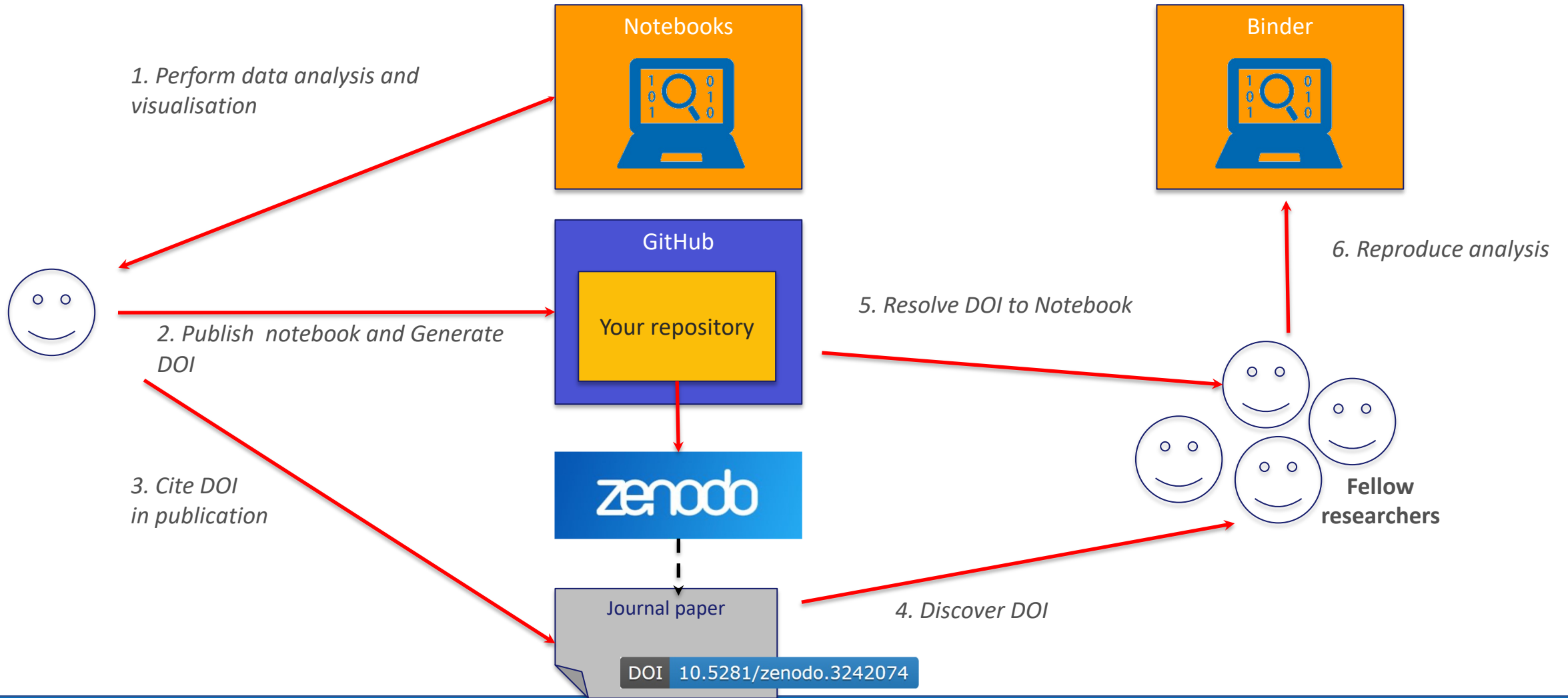
Reproducible Research and Data Analysis Exercise

University POLITEHNICA of Bucharest





Implement Open Science



1. Code storage and versioning

Git, GitHub, Sourcetree



🌐 A **free and open source** distributed **version control system** designed to handle everything from small to very large projects with speed and efficiency

🌐 Is easy to learn and has a tiny footprint with lightning fast performance

🌐 Has features like:

- 🌐 cheap local branching
- 🌐 convenient staging areas
- 🌐 multiple workflows

🌐 <https://git-scm.com>



Branching
and
Merging

Small and
Fast

Distributed

Data
Assurance

Staging
Area

Free and
Open
Source

Git development platforms



- **Social networks for software development**
 - Git repository hosting services + extra features
 - web-based interfaces on top of Git
 - collaboration features: access control, wikis, issues, projects, etc.
- Examples:
 - GitHub
 - GitLab
 - Bitbucket
 - Perforce
 - Beanstalk
 - Codebase
 - etc.

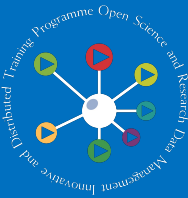
2. Reproducible environments

Binder, Jupyter Notebooks



*This tutorial is based on <https://the-turing-way.netlify.app/communication/binder/zero-to-binder.html>

Binderize your repo



- 🌐 Create a new repo on GitHub
- 🌐 Make sure the repository is public, not private!
- 🌐 Create a file called *hello.py* containing `print("Hello from Binder!")`
- 🌐 Go to <https://mybinder.org>
- 🌐 Type the URL of your repo into the “GitHub repo or URL” box
- 🌐 As you type, the webpage generates a link in the “Copy the URL below...” box
- 🌐 Open a new browser tab and visit that URL

Binderize your repo



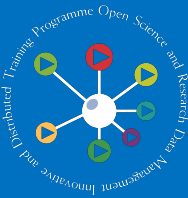
 While you wait, BinderHub is:

-  fetching your repo from GitHub
-  analysing the contents
-  building a Docker image based on your repo
-  launching that Docker image in the cloud
-  connecting you to it via your browser

 How to run your script:

-  from the launch panel, select “Terminal”
-  in the new terminal window, type `python hello.py` and press Enter

Add dependencies



🌐 Create a file called *requirements.txt* in your repo

🌐 Add a line that says `numpy==1.14.5`

🌐 Visit your Binder repo again

🌐 Check the environment:

🌐 from the launch panel, select “Python 3” from the Notebook section to open a new notebook

🌐 type the following into a new cell:

```
import numpy
print(numpy.__version__)
numpy.random.randn()
```

🌐 **!Changes made inside the Binder are not propagated in the repo!**

🌐 You can also add the Binder badge to your repo



Access external data



Small public files

-  add them directly into your GitHub repository

Medium public files

-  from a few 10s MB up to a few hundred MB
-  add a file called *postBuild* to your repo → a shell script executed as part of the image construction (only once when a new image is built)

Large public files

-  it is not practical to place large files in your GitHub repo or include them directly in the image
-  the best option is to use a library specific to the data format to stream the data as you're using it or to download it on demand as part of your code

Private files

-  there is no way to access files which are not public

Access external data



- 🌐 Go to your GitHub repo and create a file called *postBuild*
- 🌐 Add this line: `wget -q -O gapminder.csv http://bit.ly/2uh4s3g`
- 🌐 Update the *requirements.txt* file by adding `pandas` and `matplotlib`
- 🌐 Relaunch your Binder
- 🌐 Visualise the data by creating a new notebook and running the following:

```
%matplotlib inline

import pandas

data = pandas.read_csv("gapminder.csv", index_col="country")

years = data.columns.str.strip("gdpPercap_") # Extract year from last 4 characters of each column name
data.columns = years.astype(int)             # Convert year values to integers, saving results back to dataframe

data.loc["Australia"].plot()
```

Sample Binder repos



 <https://github.com/raduciobanu/trainrdm-example>

 https://mybinder.readthedocs.io/en/latest/examples/sample_repos.html

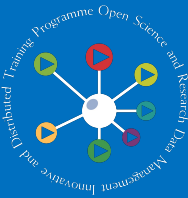
3. Archiving code and software

Zenodo



Publish GitHub repo in Zenodo

Exercise





Search

Q

Upload

Communities

 ciprian.dobre@cs.pub.ro

Home / Account / GitHub

Settings

Profile

Change password

Security

Linked accounts

Applications

Shared links

GitHub

 **GitHub Repositories**

(updated 2 minutes ago) [Sync now ...](#)

 **Get started**

1 Flip the switch

Select the repository you want to preserve, and toggle the switch below to turn on automatic preservation of your software.

ON

2 Create a release

Go to [GitHub](#) and [create a release](#). Zenodo will automatically download a .zip-ball of each new release and register a DOI.

3 Get the badge

After your first release, a DOI badge that you can include in GitHub README will appear next to your repository below.

DOI 10.5281/zenodo.8475

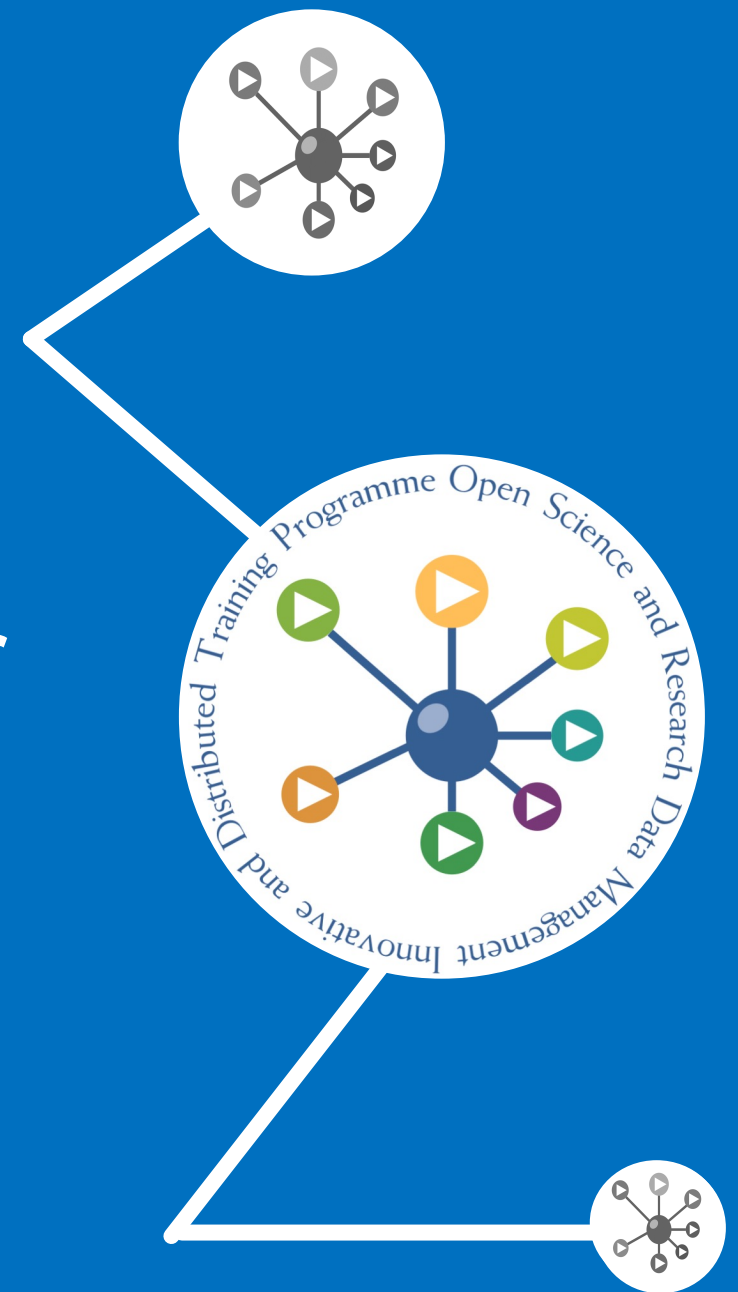
(example)



Repositories

4. Reproducible research after data analysis

Overleaf, Gnuplot



Performing reproducible research



🌐!Reminder from Monday!

🌐After data analysis

- 🌐 **generate figures and tables directly from code**
- 🌐 **automate data pre-processing, analysis and manuscript generation as “one-button” processes**
- 🌐 increase access to publications by posting preprints
- 🌐 use data and code repositories for sharing (instead of personal websites)
- 🌐 create research compendiums → archives of data, code, software and products from a research project
- 🌐 in the published manuscript, offer explicit instructions regarding where to locate data, metadata and code

Performing reproducible research



Overleaf

 collaborative cloud-based LaTeX editor used for writing, editing and publishing scientific documents

 <https://www.overleaf.com>

Gnuplot

 portable command-line driven graphing utility

 <http://www.gnuplot.info>

 Directly generate Gnuplot charts in Overleaf using the *gnuplottex* package

THANK YOU!



Follow us

