Spatial data, code

Workshop FAIR data and data reuse for ESG researchers – Module 4

October 18, 2022, by Maarten Storm, Cindy Quik and Luc Steinbuch

WAGENINGEN
UNIVERSITY & RESEARCH

https://doi.org/10.4121/21399975

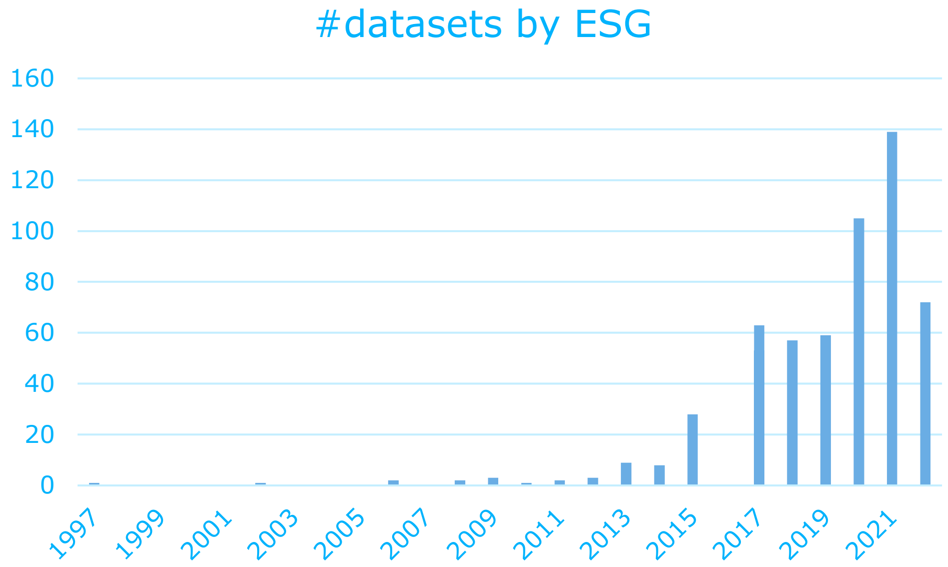
This module

- Context: ESG
- Spatial data + FAIR
- Code + FAIR

https://doi.org/10.4121/21399975

ESG - datasets and software[1]

	datasets	software
WEnR:	180	25
WU-ESG:	420	25
Total:	600	50



For comparison: 17,500 peer reviewed articles, 3,100 reports

According to research.wur.nl, accessed7 Oct 2022; numbers are approximations and rounded

ESG experiences and opinions

In 2020 a student interviewed 17 ESG researchers (WU-ESG and WEnR, bias towards GRS):

- Almost all interviewees used also, or only, existing data

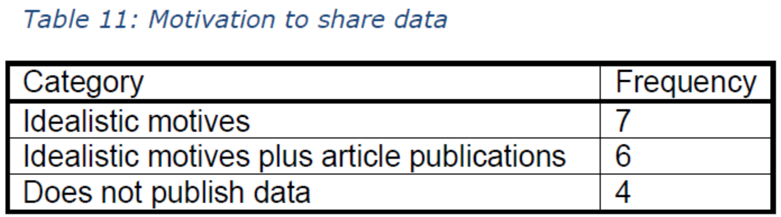
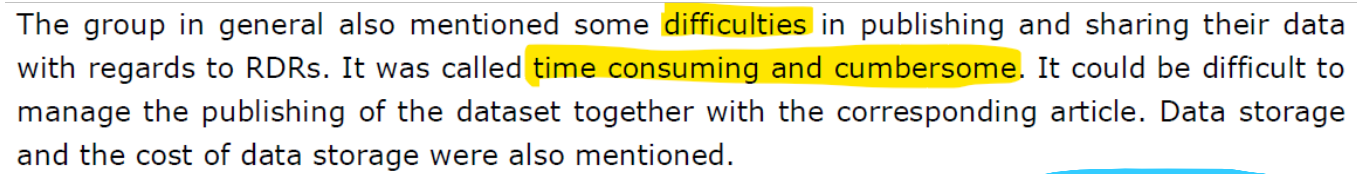
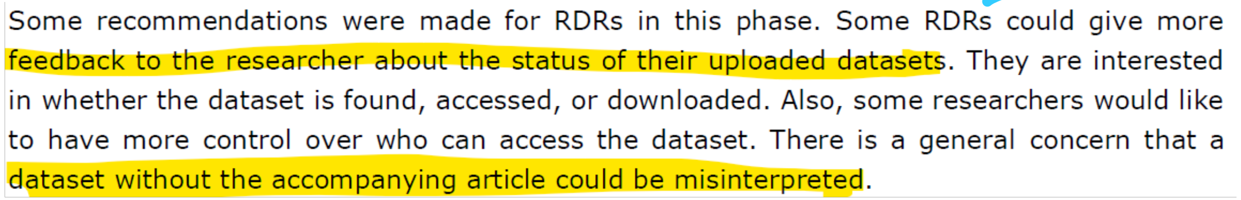


Table 11: Motivation to share data

Category	Frequency
Idealistic motives	7
Idealistic motives plus article publications	6
Does not publish data	4



The group in general also mentioned some difficulties in publishing and sharing their data with regards to RDRs. It was called time consuming and cumbersome. It could be difficult to manage the publishing of the dataset together with the corresponding article. Data storage and the cost of data storage were also mentioned.



Some recommendations were made for RDRs in this phase. Some RDRs could give more feedback to the researcher about the status of their uploaded datasets. They are interested in whether the dataset is found, accessed, or downloaded. Also, some researchers would like to have more control over who can access the dataset. There is a general concern that a dataset without the accompanying article could be misinterpreted.

RDS: Research Data Repository

Source: *The use of Research Data Repositories by researchers*, Christian klein Gebbink, Thesis Report GIRS-2020-46

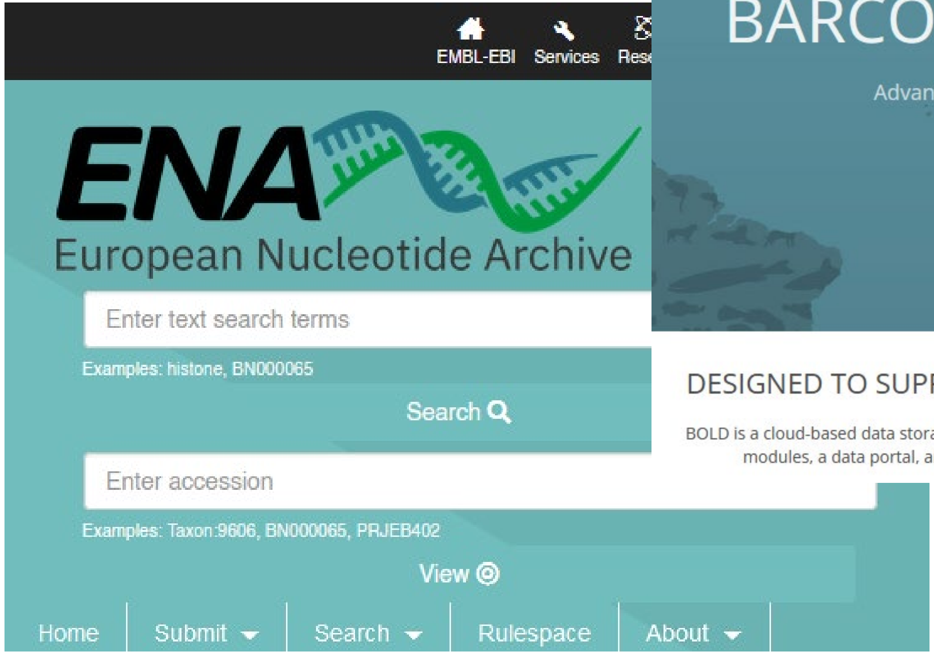
ESG Domain specific output

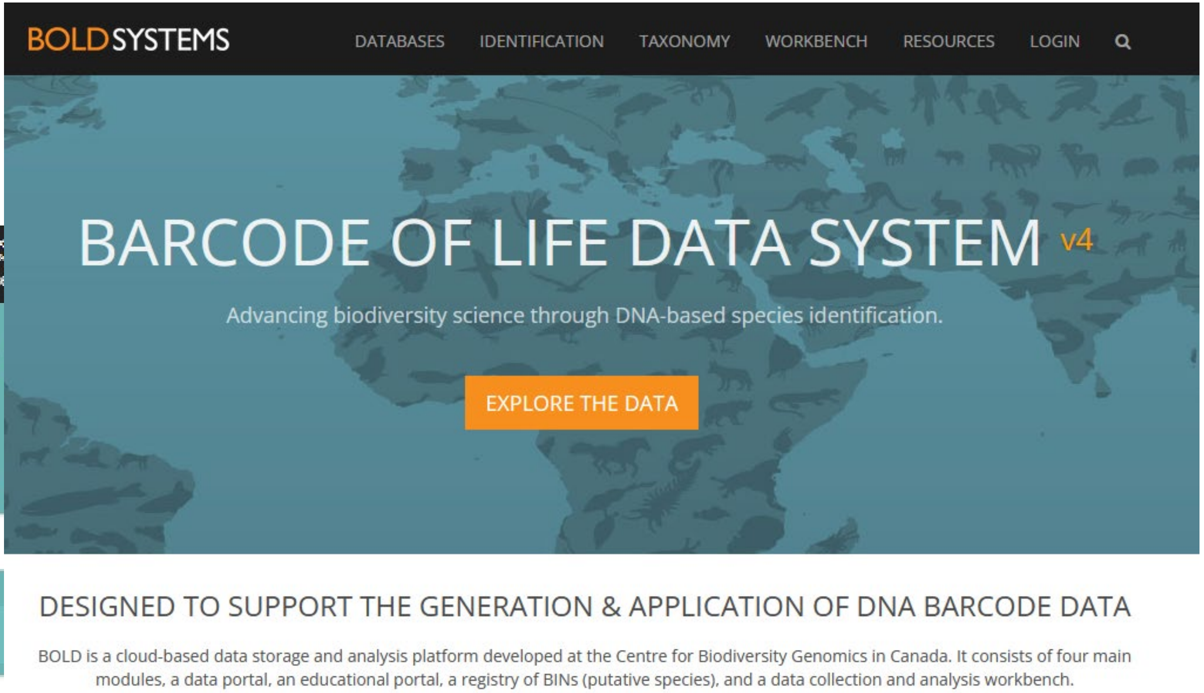
- Physical observations
  Several physical labs, for example NCL; analysis results from external labs
- “Social science” data such as graphics, text, video;
  Team Biodiversity and Policy; Regional Development and Spatial Use; Cultural Geography Group; Forest and Nature Conservation Policy Group. Scenario modelling.
- Designer data such as Autocad, Adobe Photoshop & Illustrator
  Architecture and Spatial Planning group
- Genetic data
  * Ecology * groups and teams, Soil biology group
- Geographic data - space, eventual time and direction;
  Many teams and groups related to soil, water, ecology, weather & climate, landscape, socio-economic factors etc.
- Code such as Python, R, Matlab etc
  Also many teams and groups

WAGENINGEN
UNIVERSITY & RESEARCH

https://doi.org/10.4121/21399975

Genetic data




BOLD SYSTEMS
DATABASES
IDENTIFICATION
TAXONOMY
WORKBENCH
RESOURCES
LOGIN
BARCODE OF LIFE DATA SYSTEM v4
Advancing biodiversity science through DNA-based species identification.
EXPLORE THE DATA

DESIGNED TO SUPPORT THE GENERATION & APPLICATION OF DNA BARCODE DATA

BOLD is a cloud-based data storage and analysis platform developed at the Centre for Biodiversity Genomics in Canada. It consists of four main modules, a data portal, an educational portal, a registry of BINs (putative species), and a data collection and analysis workbench.

https://www.boldsystems.org; https://www.ebi.ac.uk/ena/browser/home

WAGENINGEN UNIVERSITY & RESEARCH

Geographic data + FAIR challenges

Vector data (point, line, area); Raster data (pixels); 2D or 3D; time

Can be related to soil, water, ecology, landscape, socio-economic factors, pollution, weather & climate, etc.

Often BIG (especially raster data) -> Storage during and after research

Often calculation intensive -> Reproducible workflow

Lots of different sources, many of which open access -> Keep track of licenses

Different file formats -> How to store “Interoperable”?

Different projections -> Decide on a 'standard' projection

Metadata -> Make your data Findable

WAGENINGEN UNIVERSITY & RESEARCH

Geographic data and storage

Large volumes -> Storage during and after research

Where to store your data? WUR Data Storage Finder: https://library.wur.nl/storagefinder/

What to store within WUR storage or in repository?

- Original data: will it be available in the further future somewhere else?
- Scripts: to be able to reproduce the workflow.
- End result: can it be regenerated using the scripts?

Which versions to keep? Only the latest or also previous?

Geographic data: practical use

- When working from home, using .arge datasets from W:/Yoda/Sharepoint can take a lot of time because of network speed
    - First download the data you need to your computer
    - Process the data on your computer
    - Upload end results (and scripts and intermediate results) to W:/Yoda/Sharepoint.

Geographic data and reproducable workflows

Often calculation intensive -> Reproducible workflow

- Most GIS software provide solutions for storage of workflow

- Status A requirements for most important datasets and models of WENR:

https://intranet.wur.nl/Project/WRModellingToolbox/Pages/YGim_IMb5keB0iMxf2Lj5A

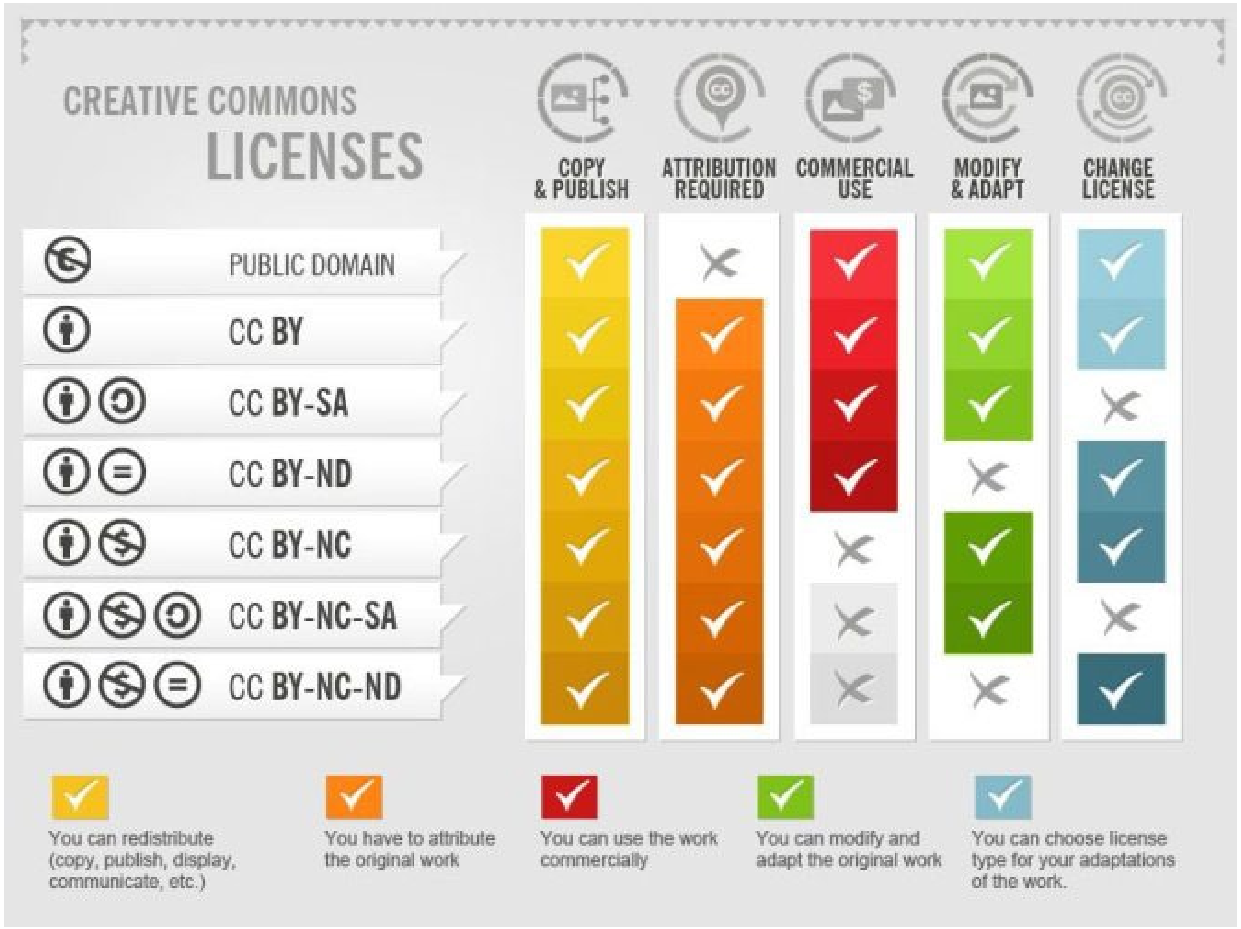
Geographic data and licenses

Lots of different sources, not always open access -> Keep track of licenses

License of source data can be a restriction: is it allowed to publish derived data?

- Example CC-ND

Add a license to all your published materials!



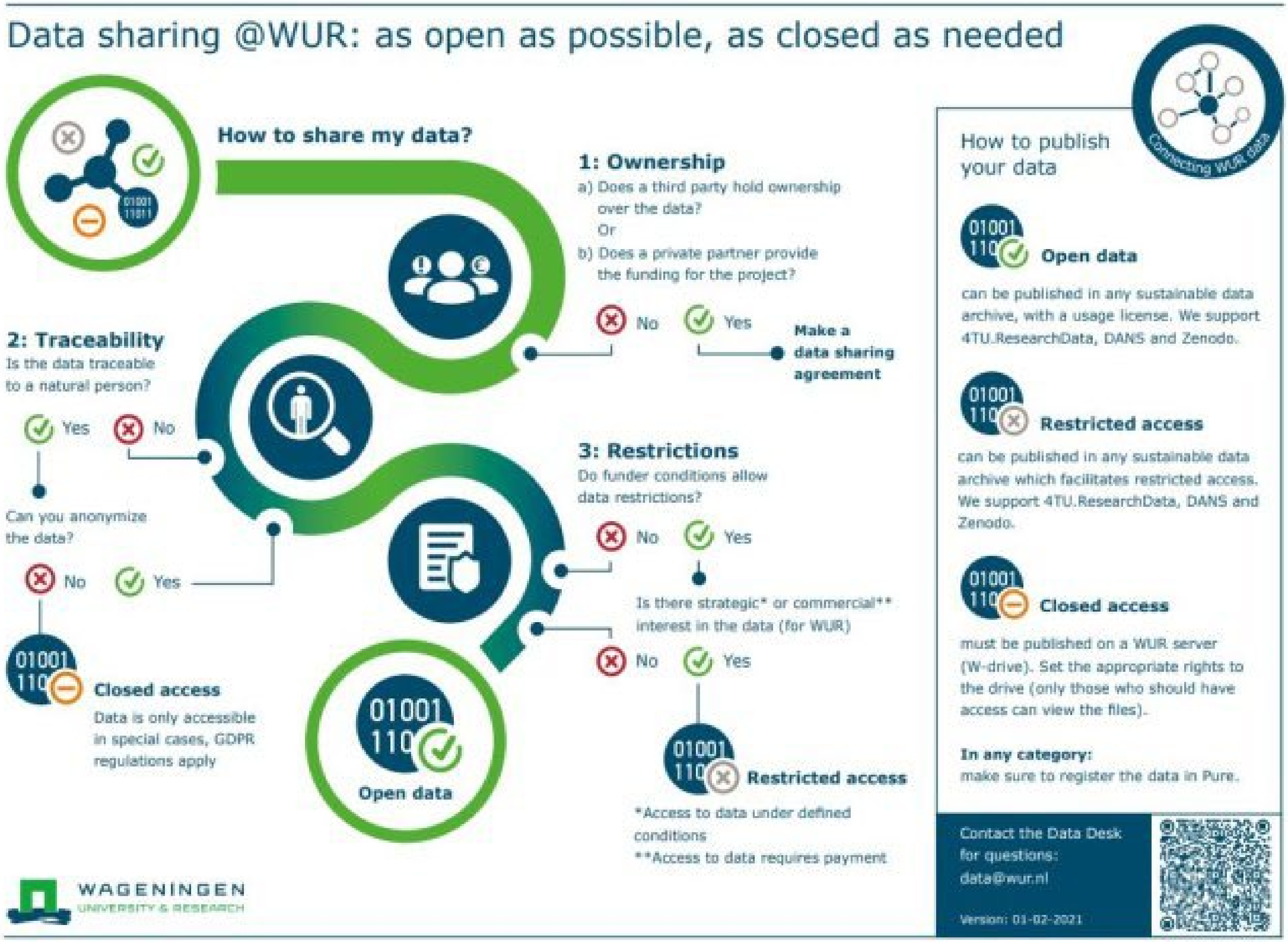
CREATIVE COMMONS LICENSES

	COPY & PUBLISH	ATTRIBUTION REQUIRED	COMMERCIAL USE	MODIFY & ADAPT	CHANGE LICENSE
PUBLIC DOMAIN	✓	✗	✓	✓	✓
CC BY	✓	✓	✓	✓	✓
CC BY-SA	✓	✓	✓	✓	✗
CC BY-ND	✓	✓	✓	✗	✓
CC BY-NC	✓	✓	✗	✓	✓
CC BY-NC-SA	✓	✓	✗	✓	✗
CC BY-NC-ND	✓	✓	✗	✗	✓

✓	You can redistribute (copy, publish, display, communicate, etc.)
✓	You have to attribute the original work
✓	You can use the work commercially
✓	You can modify and adapt the original work
✓	You can choose license type for your adaptations of the work.

Creative Commons licenses by Foter (CC-BY-SA)

WAGENINGEN UNIVERSITY & RESEARCH



Data sharing @WUR: as open as possible, as closed as needed

How to share my data?

1: Ownership

a) Does a third party hold ownership over the data?

Or

b) Does a private partner provide the funding for the project?

No — Yes → **Make a data sharing agreement**

2: Traceability

Is the data traceable to a natural person?

Yes — No

Can you anonymize the data?

No — Yes

Closed access

Data is only accessible in special cases, GDPR regulations apply

3: Restrictions

Do funder conditions allow data restrictions?

No — Yes

Is there strategic* or commercial** interest in the data (for WUR)

No — Yes

Restricted access

*Access to data under defined conditions

**Access to data requires payment

Open data

How to publish your data

Open data

can be published in any sustainable data archive, with a usage license. We support 4TU.ResearchData, DANS and Zenodo.

Restricted access

can be published in any sustainable data archive which facilitates restricted access. We support 4TU.ResearchData, DANS and Zenodo.

Closed access

must be published on a WUR server (W-drive). Set the appropriate rights to the drive (only those who should have access can view the files).

In any category:

make sure to register the data in Pure.

Contact the Data Desk for questions:

data@wur.nl

Version: 01-02-2021

https://www.wur.nl/en/value-creation-cooperation/collaborating-with-wur-1/wdcc/research-data-management-wdcc/finishing/research-data-sharing-and-guidelines.htm

Geographic data and file formats

Different fileformats -> How to store "Interoperable"?

Within geo-domain:

- Geopackage (.gpkg) for vector data
- GeoTiff (.tiff) for raster data

Both can be read by all kinds of GIS-software.

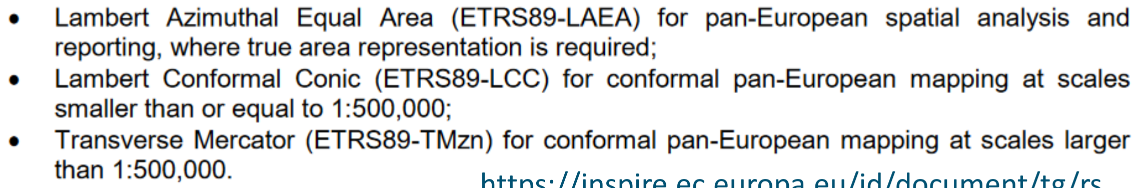
Shapefile: sometimes adviced by repository, but is not a real open format -> reverse engineered. Downsides: only 8 characters for a fieldname. Multiple files have to be kept together (.shp, .prj, .dbf, etc).

GeoDesk experience: a gpkg cannot be used by multiple people with write access at the same time. We use Esri filegeodatabase for vector data. Not really open, but usable with less limitations than shapefiles.

Geographic data and coordinate systems

Different projections -> How to store “Interoperable”?

- Doing analysis of multiple datasets require that they have the same projection.
- Define 'standard' coordinate reference system within project.
    - Netherlands: RD_New
    - Europe: ETRS89



- Lambert Azimuthal Equal Area (ETRS89-LAEA) for pan-European spatial analysis and reporting, where true area representation is required;
- Lambert Conformal Conic (ETRS89-LCC) for conformal pan-European mapping at scales smaller than or equal to 1:500,000;
- Transverse Mercator (ETRS89-TMzn) for conformal pan-European mapping at scales larger than 1:500,000.

https://inspire.ec.europa.eu/id/document/tg/rs

- Be sure to use the right transformation!

Geographic data and metadata

Metadata = data about the data (to be able to find the right data)

What metadata (standard) is required by the repository?

ISO19115: standard for metadata of geographic information

- Used for European INSPIRE datasets
- Used by the Dutch Government (Dutch profile on ISO19115: https://www.geonovum.nl/geo-standaarden/metadata)
- Used in metadata portals like
    - https://inspire-geoportal.ec.europa.eu/ (National governments in EU)
    - www.nationaalgeoregister.nl (mainly from Dutch government)
    - WUR Metadata portal (soon to be published)

WAGENINGEN
UNIVERSITY & RESEARCH

Geographic data & privacy & data classification

GDPR regulations: rules regarding collection and publication of personal information

Data collection: remove personal data if provided and not needed

Analysis results + exact location/address (house/farm) = personal information

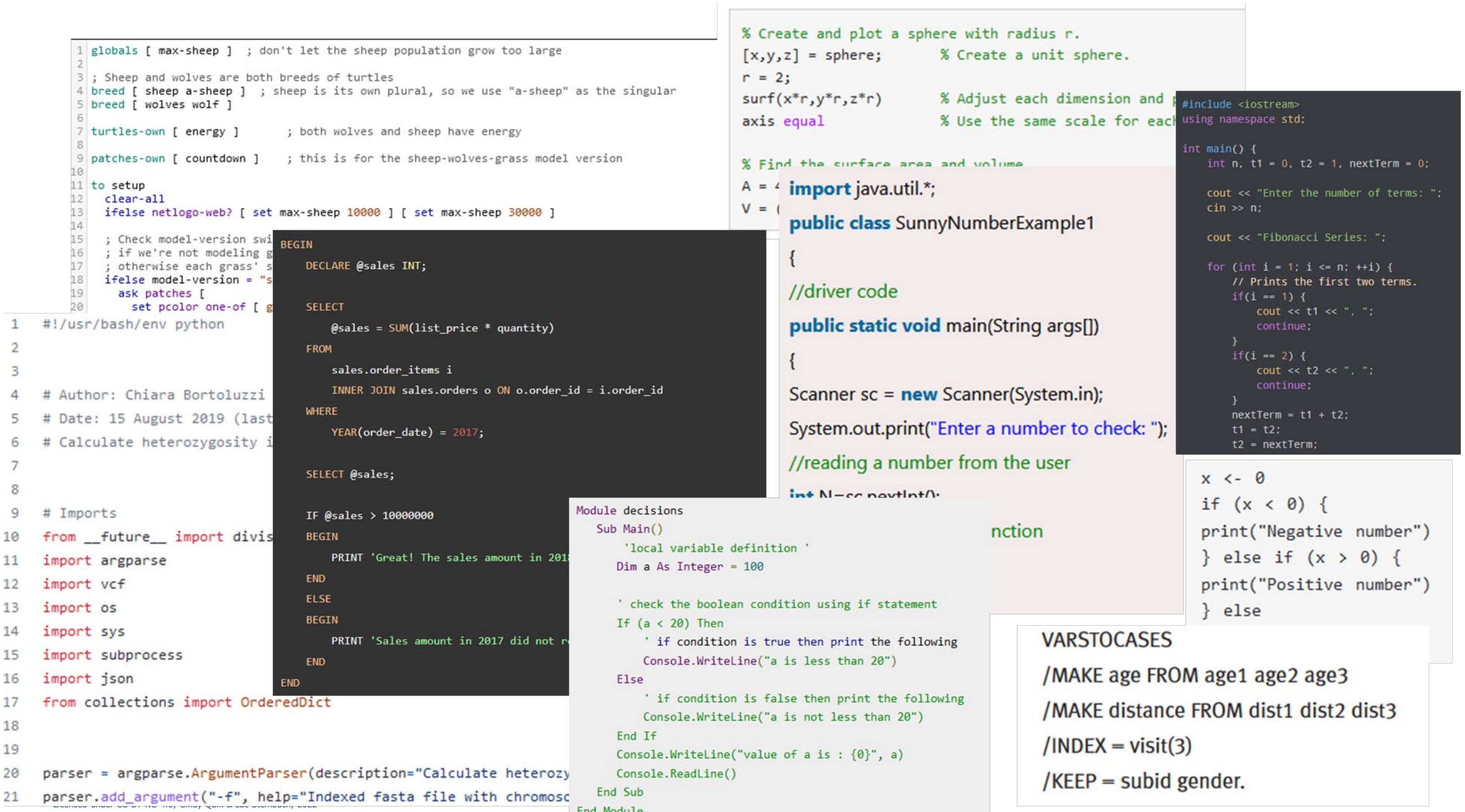
Limits sharability and thus reusability

Example CBS: resolution of statistical raster data depends on input data. Published data may not be traceable to person/company.

Data classification and whitelist: https://intranet.wur.nl/umbraco/en/practical-information/information-security/data-classification/

WAGENINGEN
UNIVERSITY & RESEARCH

FAIRizing code



```
globals [ max-sheep ]  ; don't let the sheep population grow too large

; Sheep and wolves are both breeds of turtles
breed [ sheep a-sheep ]  ; sheep is its own plural, so we use "a-sheep" as the singular
breed [ wolves wolf ]

turtles-own [ energy ]       ; both wolves and sheep have energy

patches-own [ countdown ]    ; this is for the sheep-wolves-grass model version

to setup
  clear-all
  ifelse netlogo-web? [ set max-sheep 10000 ] [ set max-sheep 30000 ]

  ; Check model-version swi
  ; if we're not modeling g
  ; otherwise each grass' s
  ifelse model-version = "s
    ask patches [
      set pcolor one-of [ g
```

```
% Create and plot a sphere with radius r.
[x,y,z] = sphere;       % Create a unit sphere.
r = 2;
surf(x*r,y*r,z*r)       % Adjust each dimension and
axis equal              % Use the same scale for each

% Find the surface area and volume
A = 
V = 
```

```
#include <iostream>
using namespace std;

int main() {
    int n, t1 = 0, t2 = 1, nextTerm = 0;

    cout << "Enter the number of terms: ";
    cin >> n;

    cout << "Fibonacci Series: ";

    for (int i = 1; i <= n; ++i) {
        // Prints the first two terms.
        if(i == 1) {
            cout << t1 << ", ";
            continue;
        }
        if(i == 2) {
            cout << t2 << ", ";
            continue;
        }
        nextTerm = t1 + t2;
        t1 = t2;
        t2 = nextTerm;
```

```
#!/usr/bash/env python


# Author: Chiara Bortoluzzi
# Date: 15 August 2019 (last
# Calculate heterozygosity i


# Imports
from __future__ import divis
import argparse
import vcf
import os
import sys
import subprocess
import json
from collections import OrderedDict


parser = argparse.ArgumentParser(description="Calculate heterozy
parser.add_argument("-f", help="Indexed fasta file with chromosc
```

```
BEGIN
    DECLARE @sales INT;

    SELECT
        @sales = SUM(list_price * quantity)
    FROM
        sales.order_items i
        INNER JOIN sales.orders o ON o.order_id = i.order_id
    WHERE
        YEAR(order_date) = 2017;

    SELECT @sales;

    IF @sales > 10000000
    BEGIN
        PRINT 'Great! The sales amount in 201
    END
    ELSE
    BEGIN
        PRINT 'Sales amount in 2017 did not r
    END
END
```

```
import java.util.*;
public class SunnyNumberExample1
{
//driver code
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
System.out.print("Enter a number to check: ");
//reading a number from the user
int N=sc.nextInt();
nction
```

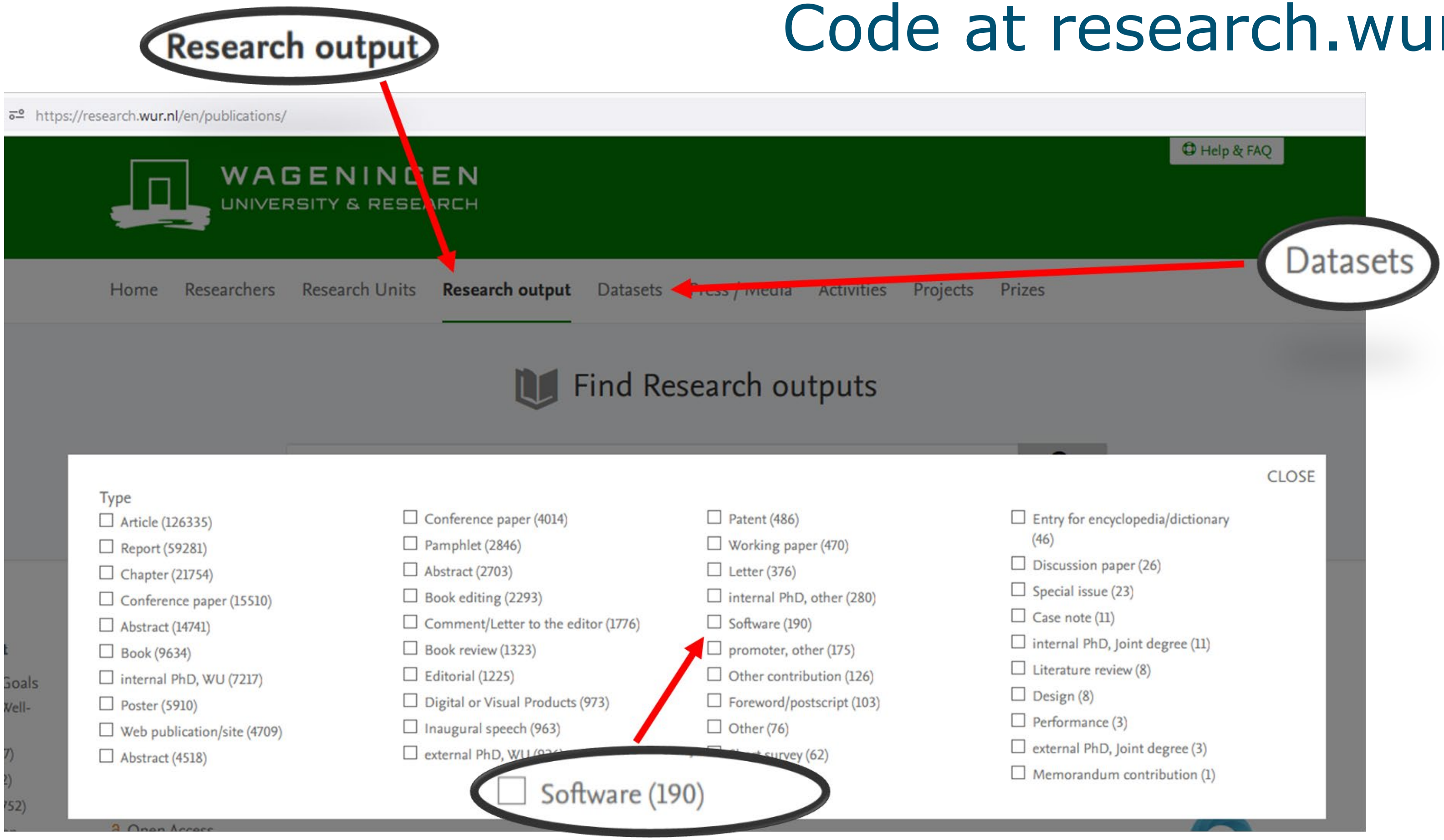
```
x <- 0
if (x < 0) {
print("Negative number")
} else if (x > 0) {
print("Positive number")
} else
```

```
Module decisions
   Sub Main()
      'local variable definition '
      Dim a As Integer = 100

      ' check the boolean condition using if statement
      If (a < 20) Then
         ' if condition is true then print the following
         Console.WriteLine("a is less than 20")
      Else
         ' if condition is false then print the following
         Console.WriteLine("a is not less than 20")
      End If
      Console.WriteLine("value of a is : {0}", a)
      Console.ReadLine()
   End Sub
End Module
```

```
VARSTOCASES
/MAKE age FROM age1 age2 age3
/MAKE distance FROM dist1 dist2 dist3
/INDEX = visit(3)
/KEEP = subid gender.
```

Code at research.wur.nl



https://research.wur.nl/en/publications/

WAGENINGEN UNIVERSITY & RESEARCH

Help & FAQ

Home | Researchers | Research Units | **Research output** | Datasets | Press / Media | Activities | Projects | Prizes

Find Research outputs

CLOSE

Type

Article (126335)	Conference paper (4014)	Patent (486)	Entry for encyclopedia/dictionary (46)
Report (59281)	Pamphlet (2846)	Working paper (470)	Discussion paper (26)
Chapter (21754)	Abstract (2703)	Letter (376)	Special issue (23)
Conference paper (15510)	Book editing (2293)	internal PhD, other (280)	Case note (11)
Abstract (14741)	Comment/Letter to the editor (1776)	Software (190)	internal PhD, Joint degree (11)
Book (9634)	Book review (1323)	promoter, other (175)	Literature review (8)
internal PhD, WU (7217)	Editorial (1225)	Other contribution (126)	Design (8)
Poster (5910)	Digital or Visual Products (973)	Foreword/postscript (103)	Performance (3)
Web publication/site (4709)	Inaugural speech (963)	Other (76)	external PhD, Joint degree (3)
Abstract (4518)	external PhD, WU	survey (62)	Memorandum contribution (1)

Differences data (in the strict sense) <-> code

- **Data** are **observations providing evidence**

- Code is a **result of a creative process**, provides a tool
- Is **executable**
- Often largely **dependent** on other software
- Shorter lifetime, need for maintenance, **fluid** rather than solid; versioning needed
- More expectation to be “**open**”
- Might explicitly be changed and extended by **others**

And:
FAIR Digital Research Objects
 <- FAIR data + FAIR code + FAIR other research resources (+Publications?)

Lamprecht, A.-L., Garcia, L., Kuzak, M., Martinez, C., Arcila, R., Martin Del Pico, E., . . . Harrow, J. (2020). Towards FAIR principles for research software. Data Science, 37-59. doi:10.3233/DS-190026

WAGENINGEN
UNIVERSITY & RESEARCH

Attention points for code

- Archived for **reproducibility** and often maintained for **reusability**
- Software development is **not recognized well** within academia, and is also outside the peer review / quality control system.
- Not so much **software engineering practices** from companies etc. are adopted by the programming scientific community.
- **Sustainability of software** should be supported.

According to certain research, scientists:
- spend 30% of their time on developing software;
- 90% is self-taught, have no formal education in creating software

Hasselbring, W., Carr, L., Hettrick, S., Packer, H., & Tiropanis, T. (2020). From FAIR research data toward FAIR and open research software. Information Technology, 39-47. doi:10.1515/itit-2019-0040

WAGENINGEN UNIVERSITY & RESEARCH

FAIR extentions for code

- **Findable**: Indexed in search engines; having a DOI + metadata
  Each version should have a DOI
- **Accessible**: Storage on a server with an Internet connection (open or restricted access). Metadata always accessible
- **Interoperable**: Machine readable, also in the future
  Software dependencies documented;
  Software should support to produce FAIR data
- **Reusable**: Human understandable (for example README.txt), copyright
  Solid documentation, also inside the code files
  Use licenses compatible with software dependencies (≠ data licenses)

WAGENINGEN UNIVERSITY & RESEARCH

Lamprecht, et al., 2020

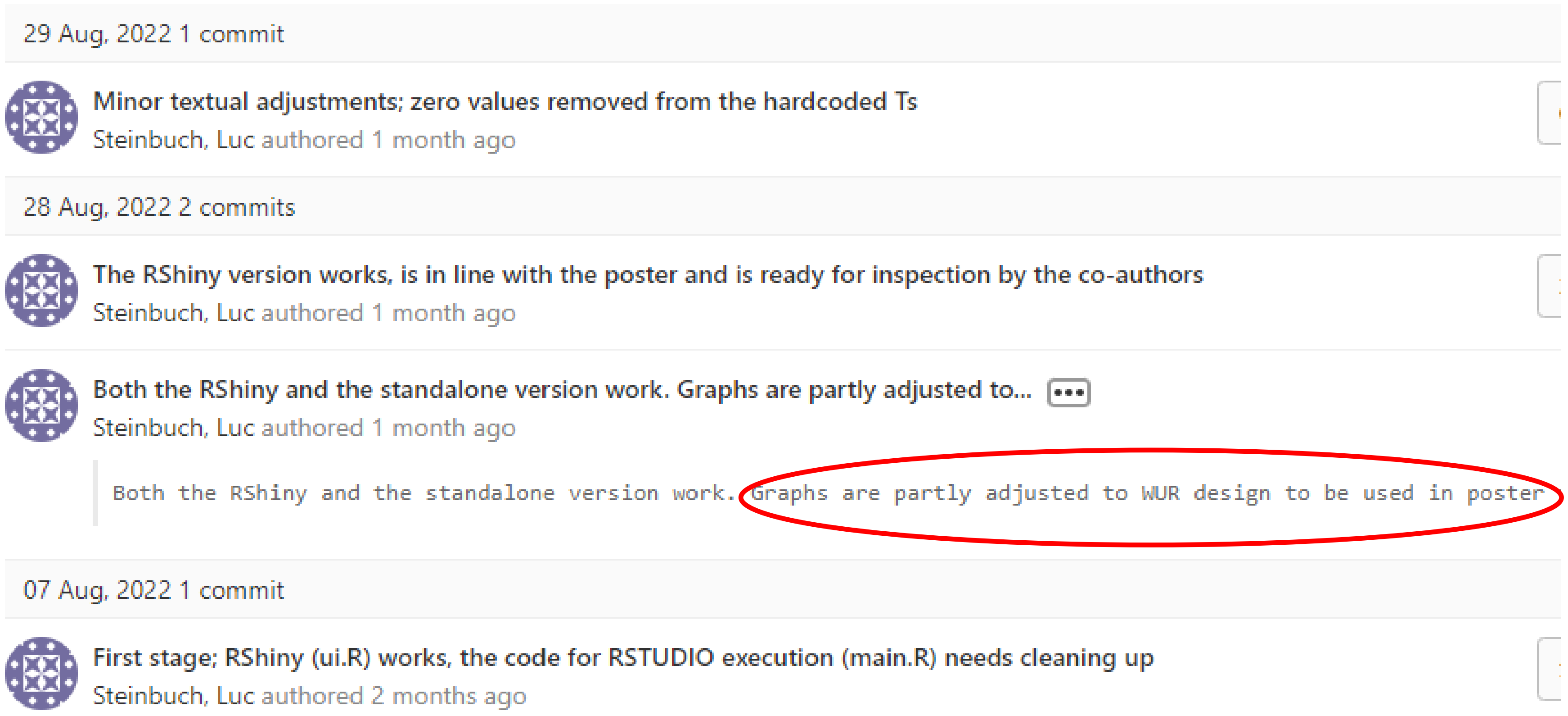
Version control

- **Findable**: Indexed in search engines; having a DOI + metadata
  Each version should have a DOI

Source code management / version control:

- Keeps track of changes in code and the reasons behind
- Both a website and locally installed software

Version control: Commits



29 Aug, 2022 1 commit

Minor textual adjustments; zero values removed from the hardcoded Ts
Steinbuch, Luc authored 1 month ago

28 Aug, 2022 2 commits

The RShiny version works, is in line with the poster and is ready for inspection by the co-authors
Steinbuch, Luc authored 1 month ago

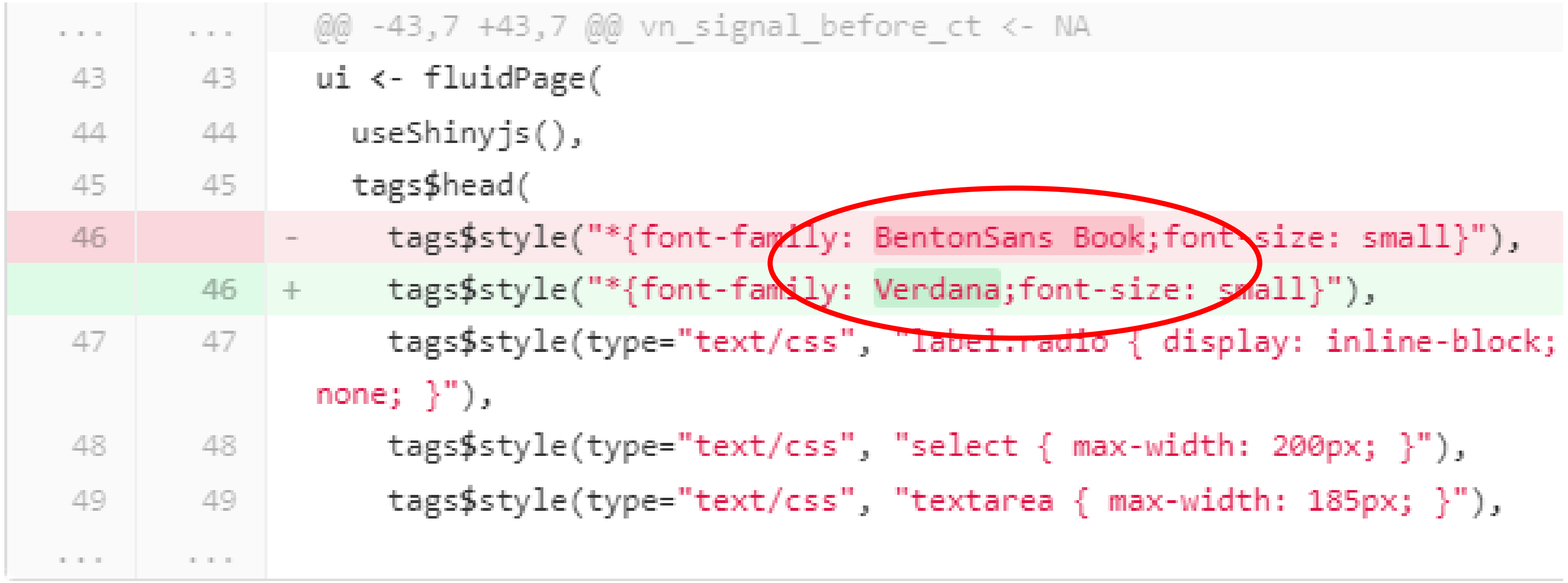
Both the RShiny and the standalone version work. Graphs are partly adjusted to...
Steinbuch, Luc authored 1 month ago

```
Both the RShiny and the standalone version work. Graphs are partly adjusted to WUR design to be used in poster
```

07 Aug, 2022 1 commit

First stage; RShiny (ui.R) works, the code for RSTUDIO execution (main.R) needs cleaning up
Steinbuch, Luc authored 2 months ago

Version control



```
...   ...   @@ -43,7 +43,7 @@ vn_signal_before_ct <- NA
            ui <- fluidPage(
              useShinyjs(),
              tags$head(
          -       tags$style("*{font-family: BentonSans Book;font-size: small}"),
          +       tags$style("*{font-family: Verdana;font-size: small}"),
                  tags$style(type="text/css", "label.radio { display: inline-block;
            none; }"),
                  tags$style(type="text/css", "select { max-width: 200px; }"),
                  tags$style(type="text/css", "textarea { max-width: 185px; }"),
...   ...
```

Version control

Source code management / version control

- **Keeps track** of changes in code and the reasons behind.
- Great for **collaboration and/or sharing**.
- De facto standard: **git***; preferably git.wur.nl.
- Not frozen, no DOI (-> later)
- Steep learning curve, not so easy installation.
- Can help to make your code FAIR

Some more info: https://geoscripting-wur.github.io/RProjectManagement/

FAIR extentions for code: How about the DOI?

- GIT* do have an URL
- Cross reference *data repository <-> GIT**
- 4TU offers related DOI's to track the development of software (actually also of data)[1]

1: see https://data.4tu.nl/info/en/use/publish-cite/after-uploading -> "Modification and versioning"

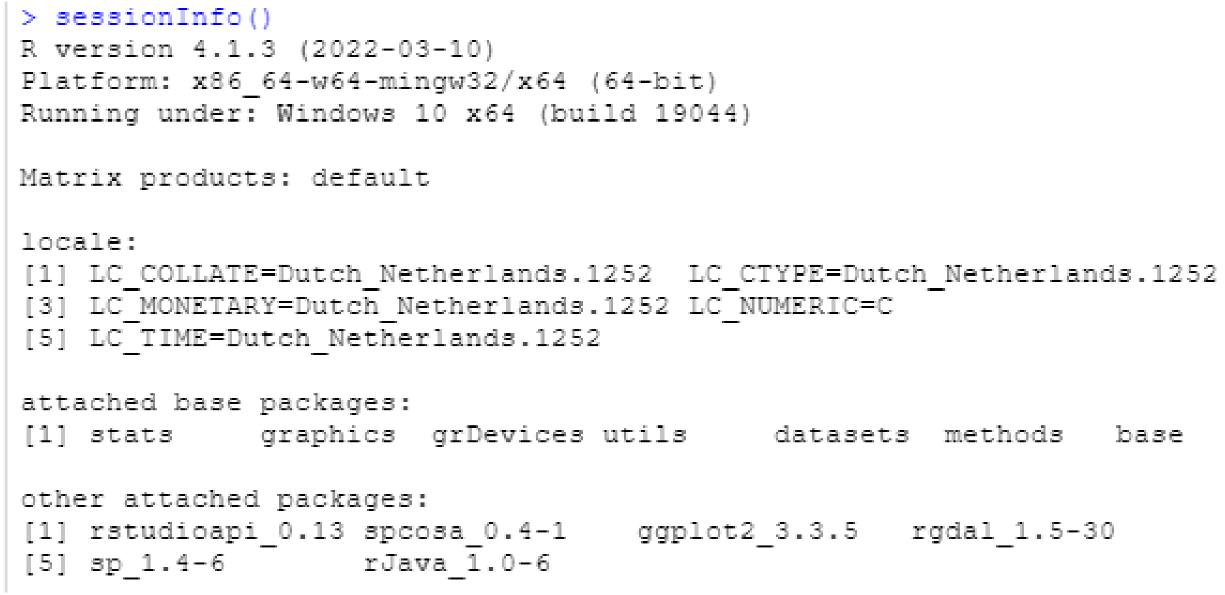
Repeated: FAIR extentions for code

- **Findable**: Indexed in search engines; having a DOI + metadata
  Each version should have a DOI
- **Accessible**: Storage on a server with an Internet connection (open or restricted access). Metadata always accessible
- **Interoperable**: Machine readable, also in the future
  Software dependencies documented;
  Software should enable to produce FAIR data
- **Reusable**: Human understandable (for example README.txt), copyright
  Solid documentation, also inside the code files
  Usage licenses compatible with software dependencies (≠ data licenses)

WAGENINGEN UNIVERSITY & RESEARCH

Interoperable

- **Interoperable**: Machine readable, also in the future
  Software dependencies documented;
  Software should enable to produce FAIR data



```
> sessionInfo()
R version 4.1.3 (2022-03-10)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19044)

Matrix products: default

locale:
[1] LC_COLLATE=Dutch_Netherlands.1252  LC_CTYPE=Dutch_Netherlands.1252
[3] LC_MONETARY=Dutch_Netherlands.1252 LC_NUMERIC=C
[5] LC_TIME=Dutch_Netherlands.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] rstudioapi_0.13 spcosa_0.4-1    ggplot2_3.3.5   rgdal_1.5-30
[5] sp_1.4-6        rJava_1.0-6
```

Interoperable

Current developments:

Offer complete workflows (Jupyter Notebooks on SURF, Google engine), or in other words: software-as-service rather than software-as-code

Recreating exact environment, machine independent (Kubernetes, Docker)

WAGENINGEN
UNIVERSITY & RESEARCH

Licenses for code

- **Reusable**: Human understandable (for example README.txt), copyright
  Solid documentation, also inside the code files
  Usage licenses compatible with software dependencies (≠ data licenses)

No license: in theory others cannot use it

Software licenses vs. data licenses:

- Explicit distribution of source code
- High dependency
- Often many contributors (simultaneously, consecutively)

Ideological discussion: *Open* vs. *Free* software

WAGENINGEN
UNIVERSITY & RESEARCH

Licenses for code

Non-free, for example non-commercial use only (for example: *JRL*)

Copyleft: almost everything allowed but license remains, credits to creator (*GNU General Public License = GPL*)

Permissive: almost everything including relicencing, credits to original author (*MIT, Apache, MPL*)

Dedicate to public domain: creator not held responsible (*CC0*)

CC-BY-SA 4.0 is one way compatible with GPLv3

Good documentation & good code I

- **Reusable**: Human understandable (for example README.txt), copyright
  Solid documentation, also inside the code files
  ~~*Usage licenses compatible with software dependencies (≠ data licenses)*~~

Write programs for people, not computers:

"The true test of good code is how easy it is to change it"

- **Do not require your readers to keep more than a handful of facts in memory** at once. Unlike computers, humans have a limited working memory, short attention span, but good pattern matching capabilities
- Make names consistent, distintive, and meaningful
- Choose style guide and stick to it. Be consistent. See also lintr for R or pylint for Phyton

Hunter-Zinck, H., de Siqueira, A. F., Vásquez, V. N., Barnes, R., & Martinez, C. C. (2021). Ten simple rules on writing clean and reliable open-source scientific software. *PLOS Computational Biology*, 1-9. doi:10.1371/journal.pcbi.1009481

Good documentation & good code II

Document interfaces and reasons, not the implementation

If your code needs a lot of comments to be understood, perhaps you need **better variable names**, better function names, or a better organisation (-> refactor*)

Documentation in the relevant locations (README.md per project, explanation of function in that function).

Documentation can be extracted from code; and the other way around, sometimes code is part of the overarching documentation (Markdown, Jupyter notebooks).

Let your code be **reviewed**!

* Refactor: inprove readabiliy and eventual speed, while keeping functionality

WAGENINGEN UNIVERSITY & RESEARCH

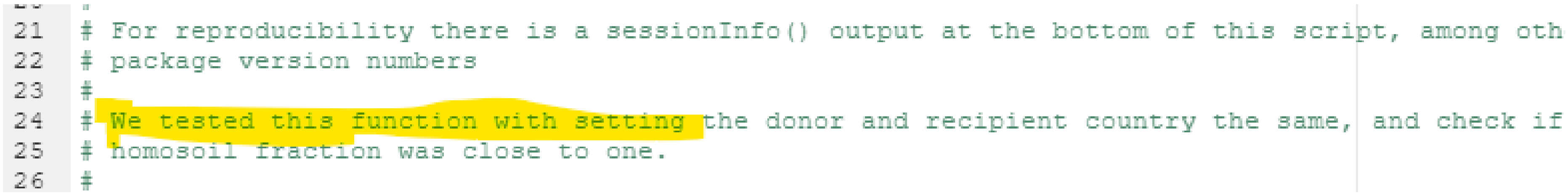
Good code: testing

Defensive programming (for example input-check with functions, explicit variable type definition); acts also as extra layer of documentation

Apply **unit tests** (gives an input the expected output) and eventual **property-based testing** (with many random inputs, does all outputs make sense?). On function- and larger level.

Use simplified cases, artificial data

Calculate test coverage. Rule from practice: aim at > 60% (R: testCoverage, Ph: coverage).

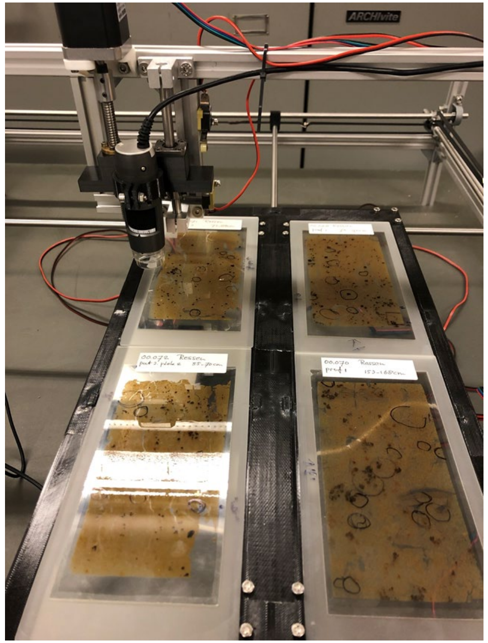


```
# For reproducibility there is a sessionInfo() output at the bottom of this script, among oth
# package version numbers
#
# We tested this function with setting the donor and recipient country the same, and check if
# homosoil fraction was close to one.
#
```

WAGENINGEN UNIVERSITY & RESEARCH

Extra: share information

ISRIC – world soil information:
Maps, monoliths, soil samples; thin section microscope



Pictures provided by ISRIC – world soil information

https://doi.org/10.4121/21399975

Time for a break!

Please be back at 13:55

WAGENINGEN
UNIVERSITY & RESEARCH