

# **BERD Course: Make Your Research Reproducible**

Heidi Seibold

# Table of contents

<b>Overview</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>1 Reproducible research and how to get there</b>	<b>5</b>
1.1 What is reproducible research? . . . . .	5
1.2 Hurdles: reproducible research is hard . . . . .	5
1.3 FAIR principles for reproducible research . . . . .	5
1.4 Teams work that fosters reproducibility . . . . .	6
<b>2 Get your research project organized</b>	<b>7</b>
2.1 Naming things . . . . .	7
2.2 File and folder organization . . . . .	7
2.3 Documentation . . . . .	7
2.3.1 Documenting data . . . . .	7
2.3.2 Documenting code . . . . .	7
2.3.3 Documenting research projects . . . . .	7
<b>3 Computational workflows</b>	<b>8</b>
<b>4 Publishing research outputs</b>	<b>9</b>
4.1 Licensing . . . . .	9
4.2 Where to publish . . . . .	9
4.3 FAIR revisited . . . . .	9
<b>5 Summary</b>	<b>10</b>
<b>References</b>	<b>11</b>

# Overview

This booklet helps you implement impactful steps in **making your research reproducible** (and open).

Making your work reproducible may seem daunting right now, but we'll take it **step by step** and you can choose what you want to implement now and what you want to keep for later. This is a process and there is no need to take all the steps at once. Let's just try to move towards reproducible on the reproducibility scale.

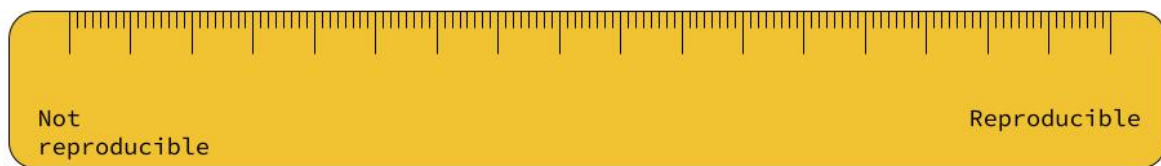


Figure 1: Reproducibility scale

# Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

# 1 Reproducible research and how to get there

## **i** Learning targets

- You will know what reproducibility is, what the hurdles are, and why it is important for research.
- You will know about the FAIR data principles and be able to discuss the issues in implementing them.
- You will be able to discuss the complexity of collaboration in research and communicating and working more effectively is important for good research.

## 1.1 What is reproducible research?

A popular definition of reproducibility comes from [The Turing Way Book](#):

At *The Turing Way*, we define **reproducible research** as work that can be independently recreated from the same data and the same code that the original team used. Reproducible is distinct from replicable, robust and generalisable as described in the figure below.

## 1.2 Hurdles: reproducible research is hard

From the definition above, reproducible research sounds really easy. It is a minimum standard. But still it is hard!

TODO: more on hurdles

## 1.3 FAIR principles for reproducible research

TODO: see also <https://data.research.cornell.edu/content/preparing-fair-data-reuse-and-reproducibility>

		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

Figure 1.1: Matrix defining reproducible research (from The Turing Way)

#### 1.4 Teams work that fosters reproducibility

TODO: ...

## 2 Get your research project organized

### Learning targets

- You will be able to create a sensible folder structure for your research projects.
- You will be able to use good names for files, folders, functions, ...
- You will have a good understanding of documentation requirements for your research project.

### 2.1 Naming things

### 2.2 File and folder organization

### 2.3 Documentation

#### 2.3.1 Documenting data

#### 2.3.2 Documenting code

#### 2.3.3 Documenting research projects

## 3 Computational workflows

### Learning targets

- You will have the version control system git installed on your computer and be able to run the important commands.
- You will be able to keep your computing environment stable.
- You will have a roadmap on how to automate your code.



## 4 Publishing research outputs

### Learning targets

- You will know about licenses.
- You will be able to pick the right platform to share your data.
- You will know how to implement the FAIR principles in practice.

### 4.1 Licensing

### 4.2 Where to publish

### 4.3 FAIR revisited

## 5 Summary

In summary, this book has no content whatsoever.

$1 + 1$

[1] 2

## References

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.