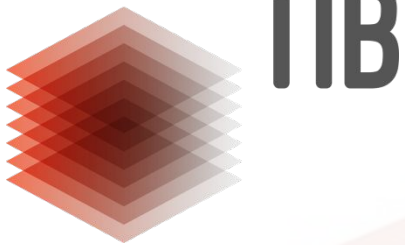
LEIBNIZ INFORMATION CENTRE
FOR SCIENCE AND TECHNOLOGY
UNIVERSITY LIBRARY



TIB

Reusable

Luke Johnston, Mateusz Kuzak, Katrin Leinweber

TIB, 12. July 2018 Recording: doi.org/10.5446/37827
FAIR Data & Software (Carpentries-based workshop) **#TIBFDS**

Leibniz Association

to be Reusable

R1. (meta)data have a **plurality of accurate and relevant attributes**

R1.1 (meta)data are released with a clear and accessible data **usage licence**

R1.2 (meta)data are associated with their **provenance**

R1.3 (meta)data meet domain-relevant **community standards**

Your institution’s / repository’s role

- provide metadata schema in human- & machine-readable format
- request relevant general and / or subject-specific metadata from researchers
- offer licence file upload or references
- implement discipline-specific (meta)data standards if necessary
  - check relevance regularly

Your role as a scientist

Reusable

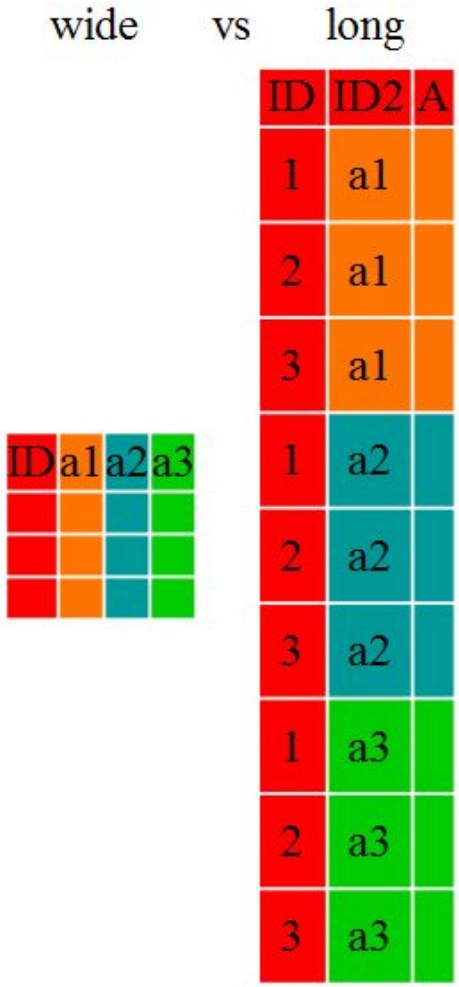
TIB

- be as detailed as possible when adding (meta)data to provide useful context
  - Purpose of data creation / collection, date, conditions, parameter settings, etc.
  - raw or processed data or both?
  - explain variable / column / parameter names, if not self-explanatory already or vocabulary-defined
  - document & cite datasets & software (+ version) that you used
- set a licence, preferably CC-BY & “provide a link to the license”
  - if applicable, provide information on additional legal conditions
- specify provenance (your role in collecting / generating the data), citation wish
- use community standards for data archiving & publication, or explain other choices
- request that repositories in your field of study collect these details

Reusability Agenda

1. Tidy(ing) data
2. Citing data & software
3. Packaging functions & data in R

Tidy data



Happy families are all alike; every unhappy family is unhappy in its own way.

-- Leo Tolstoy

Tidy datasets are all alike but every messy dataset is messy in its own way.

-- Hadley Wickham

- 1 table per type & 1 type per table
- 1 variable per column & 1 column per variable
- 1 observation per row & 1 row per observation
- 1 value per cell
- column headers are IDs
- MS Excel, macOS Numbers, LibreOffice Calc etc. nudge towards “wide”

Wright & Naupaka: Software Carpentry: R for Reproducible Scientific Analysis (v2016.06) doi:10.5281/zenodo.57520

Tidy data (Wickham, 2014, doi:10.18637/jss.v059.i10)

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

Table 1: Typical presentation dataset.

	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

Table 2: The same data as in Table 1 but structured differently.

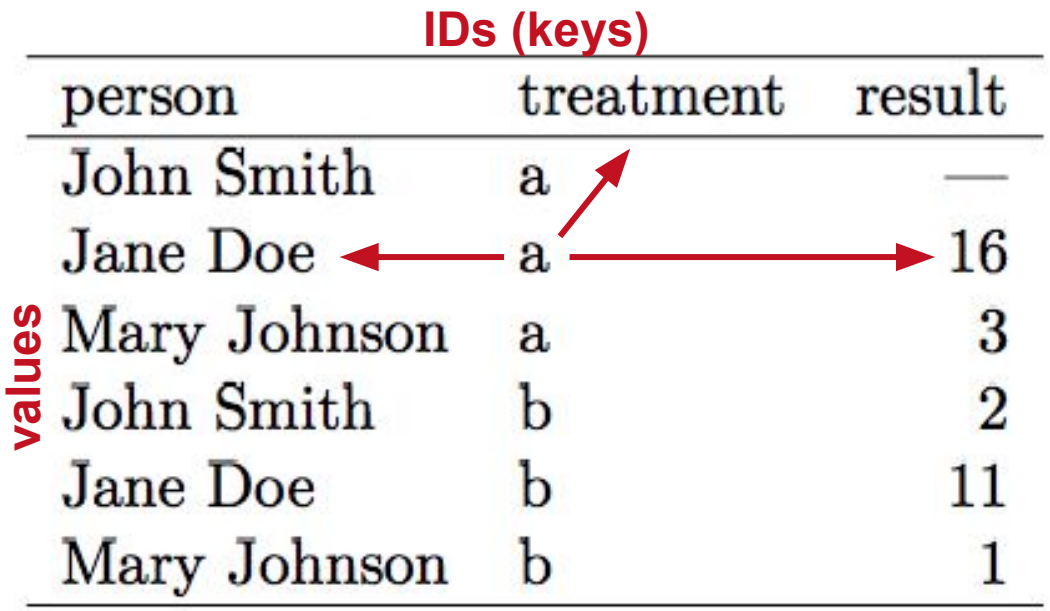
ID: “persons / patients” values: names

Both are variables!

observations (ID: “results”)

ID: “treatment” values: a & b

Tidy data (Wickham, 2014, doi:10.18637/jss.v059.i10)



person	treatment	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

- **each value belongs to exactly 1 var & 1 obs**
- structure consistent with semantic meaning
- allows conclusion about missing data
- processable in `tidyverse` & `pandas`

Reusability Agenda

1. Tidy(ing) data
2. **Citing data & software**
3. Packaging functions & data in R

Citing data & software

- Your experiences with reference managers (BibTeX, Citavi, EndNote, Mendeley, Zotero etc.)?
- quality control factors for citation metadata:
  - Which do authors (have to) provide (to a repository)?
  - Which are included in a citation style? Can you modify that style?
- play “developer options” section from doi.org/10.5446/35351#t=02:06,08:30

Software citations (GitHub.com/FORCE11/FORCE11-sciwg)

`@software{}` in Bib(La)TeX / biber (source)

Citation File Format (source)

`codemeta.json` (source)

Bib(La)TeX style updates needed

developer provides citation metadata

`DESCRIPTION`, `CITATION.bib` / `.cff`, etc.

or: has them built `codemeta.json`

user finds metadata
on landing page or in source code

import into reference manager / build pipeline
via artifact, web scraper, DOI lookup, by copy-pasting, etc.

Rendered in document according to a citation style

Citing data & software

- demo: import Zenodo.org/record/1308061 into Zotero
- demo: RStudio > Packages > Update, run PANGAEA example, then install updates

Reusability Agenda

1. Tidy(ing) data
2. Citing data & software
3. **Packaging functions & data in R**

Basic rules for interoperable scripts

- load modules / packages / etc. explicitly atop the file: `import … as …` & `library('…')`
- hard-coding absolute folder paths results in errors for anyone else
  - instead: relative paths within the organised project folder (see above)

❌ `numpy.loadtxt(fname='/Users/YOU/project-X/data/inflammation-01.csv')`

☺ `numpy.loadtxt(fname='../data/inflammation-01.csv')` **or** `__file__`

❌ `setwd("C:\Users\YOU\path\that\nobody\else\has")`

☺ `.rproj` files

from: Jenny Bryan (2018) Project-oriented workflow & more tips on Twitter.com/HadleyWickham/status/940021008764846080

Another solution: build a module / package! => FAIR-R/04

LEIBNIZ INFORMATION CENTRE
FOR SCIENCE AND TECHNOLOGY
UNIVERSITY LIBRARY

TIB

Which questions do you have for us?

Contact information:
Katrin.Leinweber@TIB.eu & Angelina.Kraft@TIB.eu
T +49 511 762-14693 & -14238

Leibniz Association