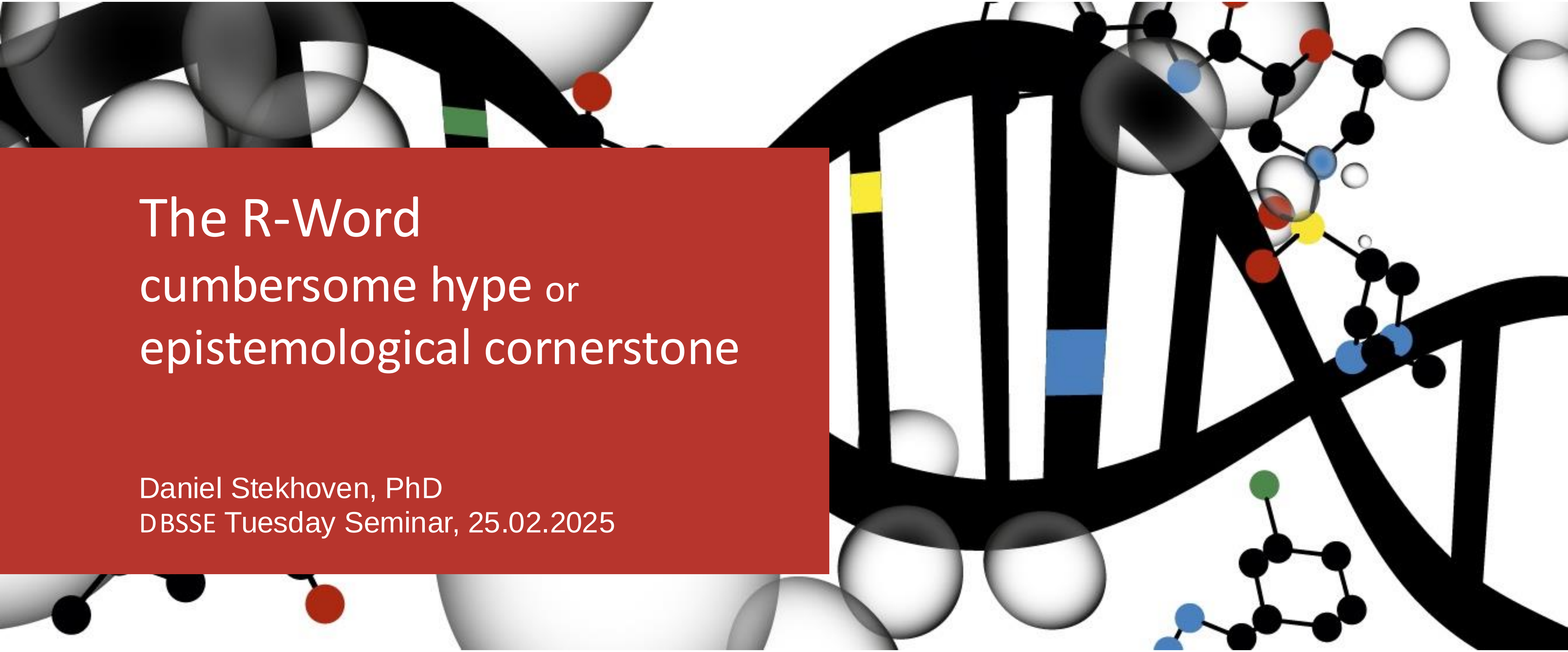
ETH zürich

nexus
Personalized Health Technologies



The R-Word
cumbersome hype or epistemological cornerstone

Daniel Stekhoven, PhD
DBSSE Tuesday Seminar, 25.02.2025

Mentimeter Join at menti.com | use code 6641 0175

The three Rs

Reading

Writing

Arithmetic

The four Rs

Reading

Writing

Arithmetic

Rationality

1. Rationality: What It Is, Why It Seems Scarce, Why It Matters, Steven Pinker, 2021, ISBN 978-0525561996

The five Rs

Reading

Writing

Arithmetic

Rationality

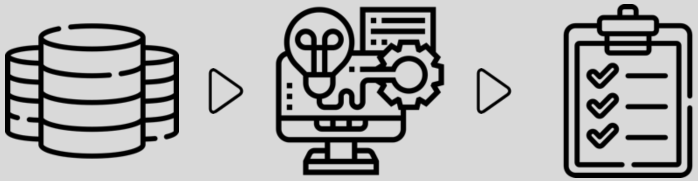
Reproducibility

1. Rationality: What It Is, Why It Seems Scarce, Why It Matters, Steven Pinker, 2021, ISBN 978-0525561996
2. Rougier, N. P. (2016). R-words. Available online at: https://github.com/ReScience/ReScience-article/issues/5

Definitions

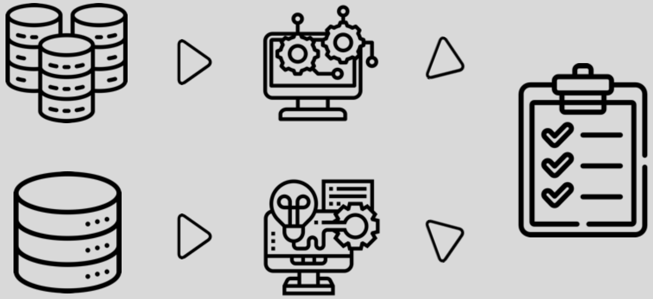
Reproducibility

... obtaining consistent computational results using the **same input data**, computational steps, methods, code, and conditions of analysis.



Replicability

... obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its **own data**.

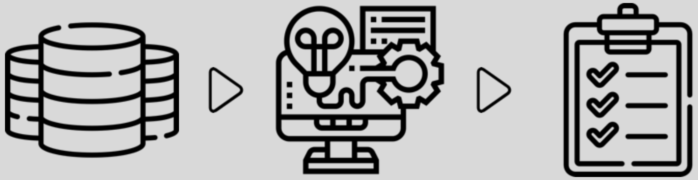


1. N. A. of S., Engineering, and Medicine. (2020) Harvard Data Science Review, 2(4). https://hdsr.mitpress.mit.edu/pub/nas-report-highlights
2. Icons from https://www.flaticon.com/

Definitions

Methods Reproducibility

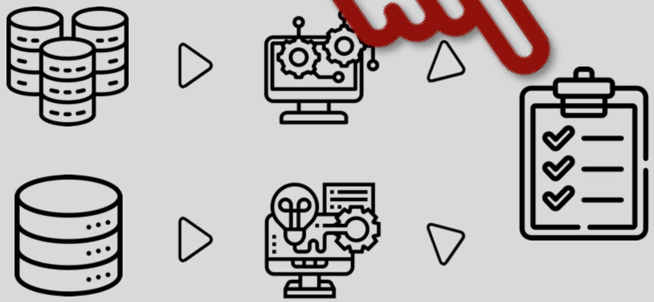
... obtaining consistent computational results using the **same input data**, computational steps, methods, code, and conditions of analysis.



Results Reproducibility

~~Replicability~~

... obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its **own data**.



+ Inferential Reproducibility

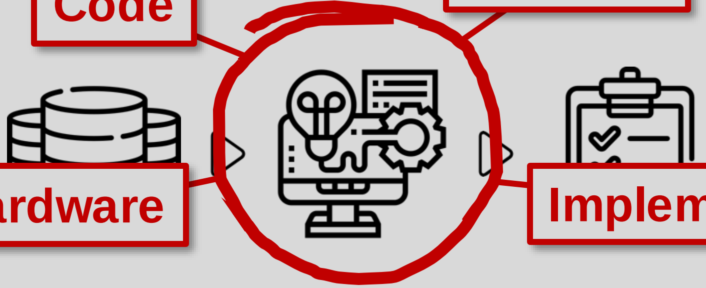
... drawing qualitatively similar conclusions.

1. N. A. of S., Engineering, and Medicine. (2020) Harvard Data Science Review, 2(4). https://hdsr.mitpress.mit.edu/pub/nas-report-highlights
2. Icons from https://www.flaticon.com/
3. Goodman S.N. et al.Sci. Transl. Med.8,341ps12-341ps12(2016). DOI:10.1126/scitranslmed.aaf5027

Definitions

Methods Reproducibility

... obtaining consistent computational results using the **same input data**, computational steps, methods, co and conditions of anal

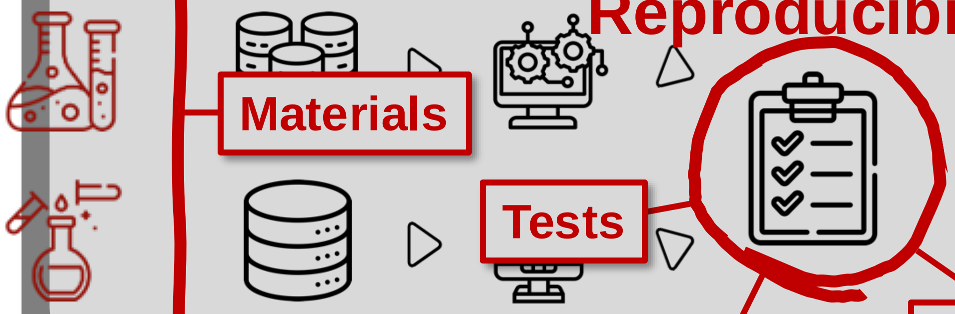


Computational Reproducibility

Results Reproducibility

~~Replicability~~

... obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its own data.



Statistical Reproducibility

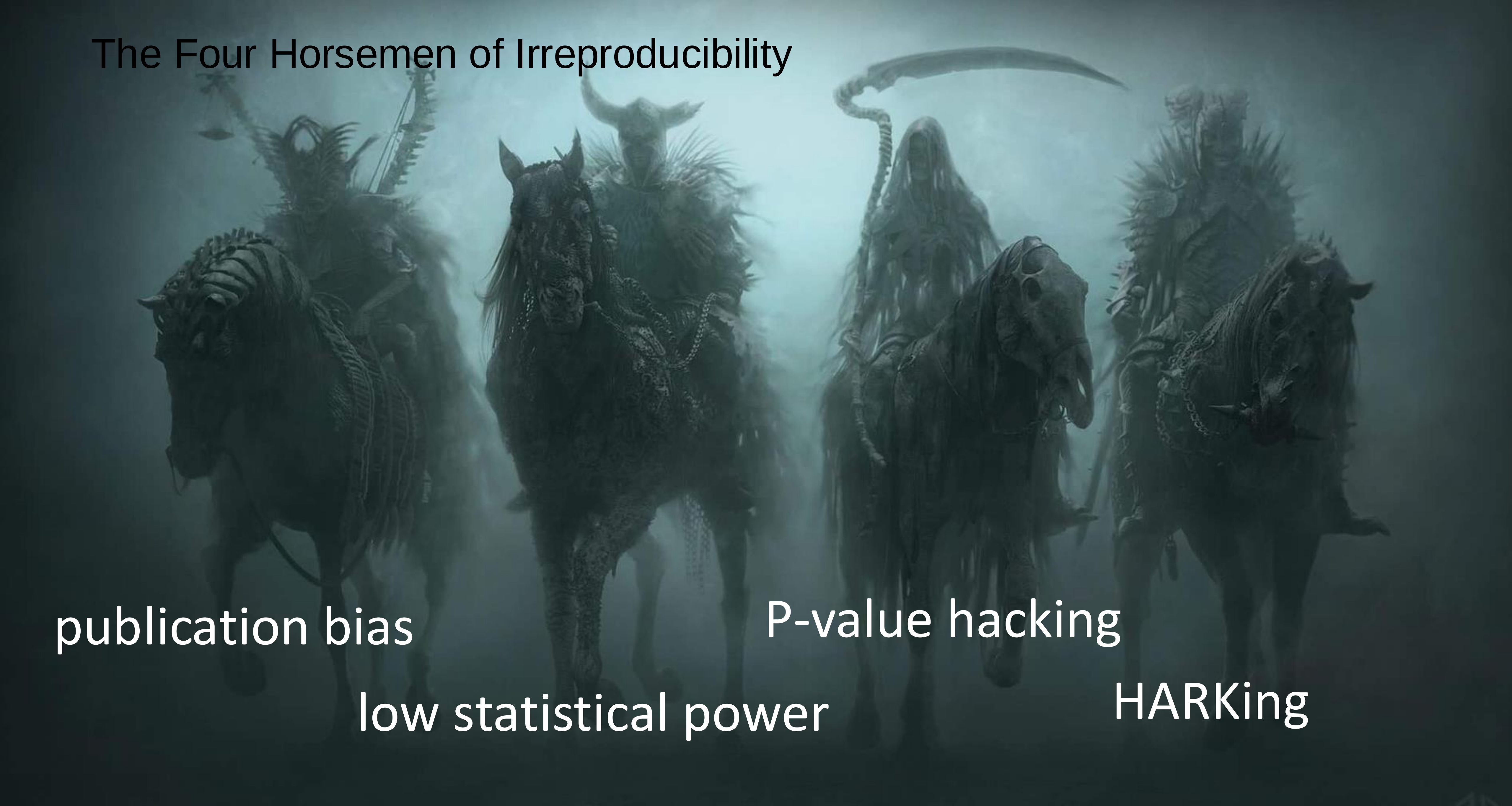
Empirical Reproducibility

+ Inferential Reproducibility

... drawing qualitatively similar conclusions.

1. N. A. of S., Engineering, and Medicine. (2020) Harvard Data Science Review, 2(4). https://hdsr.mitpress.mit.edu/pub/nas-report-highlights
2. Icons from https://www.flaticon.com/
3. Goodman S.N. et al.Sci. Transl. Med.8,341ps12-341ps12(2016). DOI:10.1126/scitranslmed.aaf5027
4. Stodden, V. (2014). Edge.org. https://www.edge.org/response-detail/25340

The Four Horsemen of Irreproducibility



publication bias

low statistical power

P-value hacking

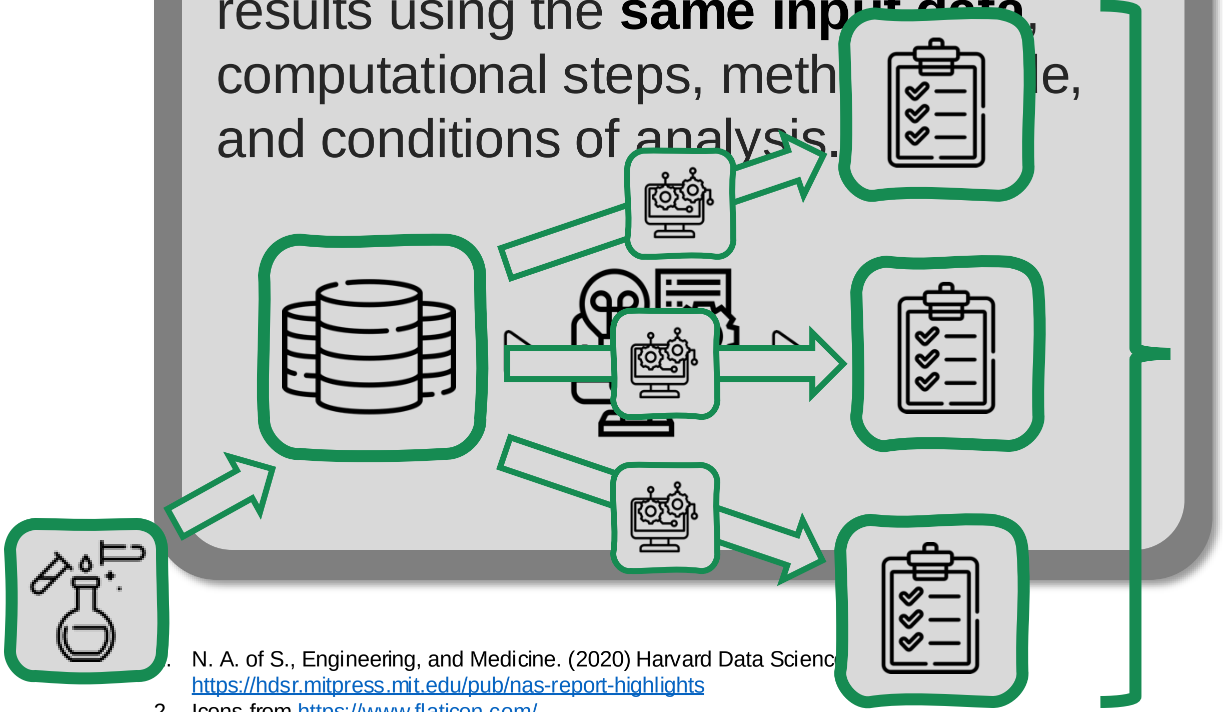
HARKing

Inspired by Dorothy Bishop, Artwork by David Masson san Gabriel & Jared Krichevsky

Definitions

Reproducibility

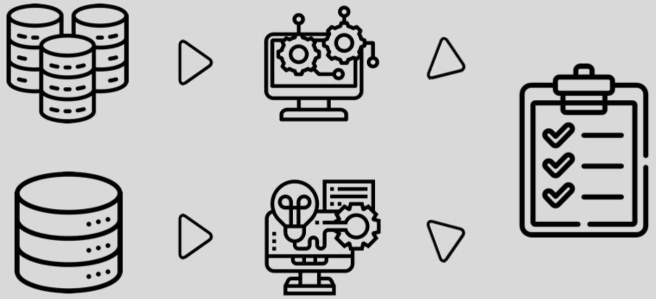
... obtaining consistent computational results using the same input data, computational steps, meth le, and conditions of analysis.



Genomic Reproducibility

Replicability

... obtaining consistent results across studies aimed at answering the same scientific question, each of which has obtained its **own data**.



1. N. A. of S., Engineering, and Medicine. (2020) Harvard Data Scienc
https://hdsr.mitpress.mit.edu/pub/nas-report-highlights
2. Icons from https://www.flaticon.com/
3. Baykal, P.I., Łabaj, P.P., Markowetz, F. *et al.* Genomic reproducibility in the bioinformatics era. *Genome Biol* **25**, 213 (2024). https://doi.org/10.1186/s13059-024-03343-2

Definitions

PROJECT DESIGN
COLLABORATION
REPRODUCIBILITY
the TURING WAY
COMMUNICATION & OUT REACH
ETHICS
TOO EASY NOT TO DO
Scriberia

1. The Turing Way Community, & Scriberia. (2024).
Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. https://doi.org/10.5281/ZENODO.3332807

Definitions

		Data / Material	
		same	different
Analysis / Protocol	same	**Reproducibility**	**Replicability**
	different	**Robustness**	**Generalisation**

1. The Turing Way Community, & Scriberia. (2024).
   Illustrations from The Turing Way: Shared under CC-BY 4.0 for reuse. https://doi.org/10.5281/ZENODO.3332807

Reproducibility

“Promise”

Raw Data

Protocol

		Data / Material	
		same	different
Analysis / Protocol	same	Reproducibility	Replicability
	different	Robustness	Generalisation

Result

Replicability

“Original Idea”

Protocol

		Data / Material	
		same	different
Analysis / Protocol	same	Reproducibility	Replicability
	different	Robustness	Generalisation

Different Raw Data

Similar Result

Robustness

		Data / Material	
		same	different
Analysis / Protocol	same	Reproducibility	Replicability
	different	**Robustness**	Generalisation

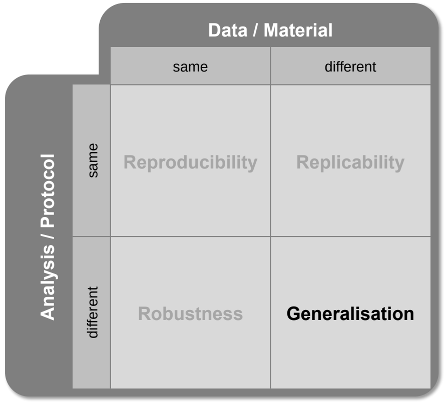
“Plan”

Different Protocol

Raw Data

Similar Result

Generalisation



		Data / Material	
		same	different
Analysis / Protocol	same	Reproducibility	Replicability
	different	Robustness	**Generalisation**

“Hypothesis”

Different Protocol

Different Raw Data

Similar Result

How would you rate your own level of reproducibility?

Move the scale on your phones or laptops!

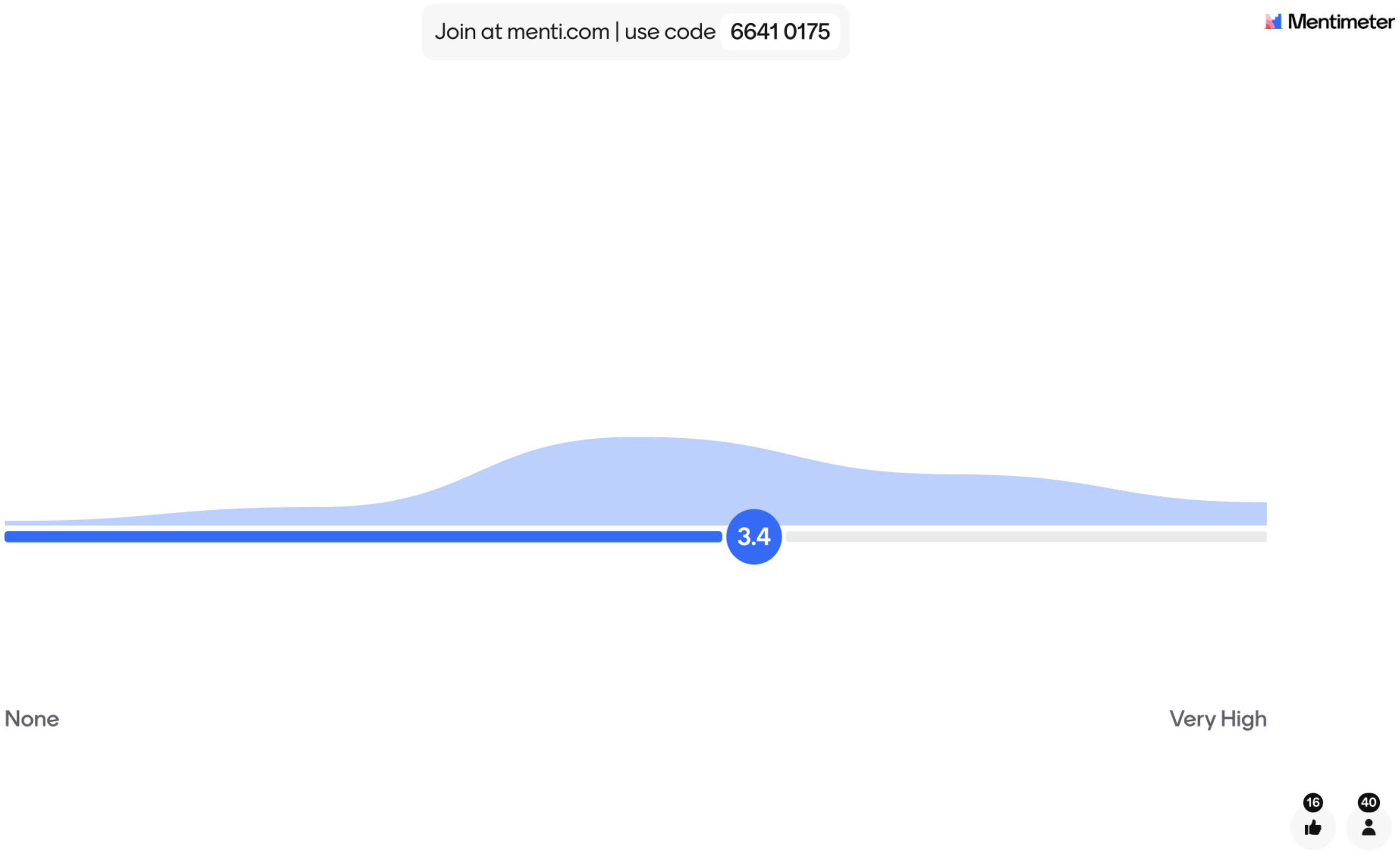
The levels are:

1. None
2. Low
3. Medium
4. High
5. Very High

Mentimeter

Join at menti.com | use code 6641 0175

How would you rate your own level of reproducibility?



Join at menti.com | use code 6641 0175

The Holy Trinity of (Computational) Reproducibility

Version Control

... track the evolution of your code.

GitHub GitLab

Workflow Management

... organise the steps of your analysis (across tools).

snakemake nextflow

Containers

... ensure consistency of your environments.

APPTAINER docker

The Lesser Trinity of (Computational) Reproducibility

Data Management

... especially for larger datasets and reuse.

Documentation

... help others (and the future you) understand.

Tests & Continuous Integration

... sustainability for your methods and applications.

Where are now? Where could we possibly go to? What is needed?

	Level 1 **Ephemeral Work**	Level 2 **Carpe Diem**	Level 3 **Sweet Spot**	Level 4 **Assembly Line**	Level 5 **Nirvana**
Profile	**Manual** and **interactive** analysis	Some **scripts**	**Version controlled** scripts	Scripts **combined** in single workflow	Single command reproduction **of it all**
	Reliance on researcher's **memory**	**Comments** in README, inline or some notebook	**Workflow management** or well-structured scripts	Minimal steps to spin up **environment** using containers	Whole lifecycle is **documented** and **continuously tested**
	Probably **not** reproducible	Reproducible if not **too long ago**	Reproducible on the **same** infrastructure by **same** person	Reproducible on/for **different** machines or collaborators	Reproducible for **anyone** after publication
Tools	• Spreadsheets with colours • Manual notes • Ad-hoc scripts • File sharing via email or personal clouds	• Basic Git usage • Simple README files • Installing software as needed • Basic ELN usage • Cloud storage w/o clear structure	• GitHub/GitLab • Conda/Virtualenv • Lightweight workflow scripts (bash scripts or Makefiles) • Basic containerization • ELN integrated with version control (e.g., openBIS, labkey)	• Full containerization • Advanced workflow managers (Snakemake, Nextflow, CWL) • Data versioning (e.g., DVC, git-annex) or persistent repositories for large datasets • Continuous integration (GitHub Actions, GitLab CI, Jenkins) • Metadata-rich ELN	• Coordination (web) application • Comprehensive CI/CD • Integrated repositories with DOIs (e.g., Zenodo, Dryad) • Automated documentation generation (e.g., Sphinx, MkDocs) • Seamless lab-to-publication pipeline
Pros & cons	Extremely difficult to **revisit** or **share** analyses.	Difficult to reproduce results if environment or code **changes**.	Colleagues (with some effort) can reproduce analyses if **they follow** step-by-step instructions.	Colleagues (and sometimes even external collaborators) can reproduce results with **minimal setup**.	Multiple collaborators or future researchers can **reproduce and extend** the work almost effortlessly.
	High risk of **irrecoverable** errors or lost steps.	Documentation might be inconsistent or **quickly outdated**.	The workflow is robust to small changes, and results are traceable, **relies on good practice** by researcher.	**Checks** reduce human error, though complex configurations can still hide subtle bugs.	CI and end-to-end automation **catch errors** early and often.

Where are now? Where could we possibly go to? What is needed?

	Level 1 **Ephemeral Work**	Level 2 **Carpe Diem**	Level 3 **Sweet Spot**	Level 4 **Assembly Line**	Level 5 **Nirvana**
Profile	**Manual** and **interactive** analysis	Some **scripts**	**Version controlled** scripts	Scripts **combined** in single workflow	Single command reproduction **of it all**
	Reliance on researcher's **memory**	**Comments** in README, inline or some notebook	**Workflow management** or well-structured scripts	Minimal steps to spin up **environment** using containers	Whole lifecycle is **documented** and **continuously tested**
	Probably **not** reproducible	Reproducible if not **too long ago**	Reproducible on the **same** infrastructure by **same** person	Reproducible on/for **different** machines or collaborators	Reproducible for **anyone** after publication
Tools	Spreadsheets with colours • Manual notes • Ad-hoc scripts • File sharing via email or personal clouds	• Basic Git usage • Simple README files • Installing software as needed • Basic ELN usage • Cloud storage w/o clear structure	GitHub/GitLab • Conda/Virtualenv • Lightweight workflow scripts (bash scripts or Makefiles) • Basic containerization • ELN integrated with version control (e.g., openBIS, labkey)	• Full containerization • Advanced workflow managers (Snakemake, Nextflow, CWL) • Data versioning (e.g., DVC, git-annex) or persistent repositories for large datasets • Continuous integration (GitHub Actions, GitLab CI, Jenkins) Metadata-rich ELN	• Coordination (web) application • Comprehensive CI/CD • Integrated repositories with DOIs (e.g., Zenodo, Dryad) • Automated documentation generation (e.g., Sphinx, MkDocs) • Seamless lab-to-publication pipeline
Pros & cons	Extremely difficult to **revis**[illegible] **share** analyses.	[illegible] environment or code **changes**.	[illegible] (with some effort) can reproduce analyses if **they follow** step-by-step instructions.	Colleagues (and sometimes even external collaborators) can reproduce results with **minimal setup**.	Multiple collaborators or future researchers can **reproduce and extend** the work almost effortlessly.
	High risk of **irrecoverable** errors or lost steps.	Documentation might be inconsistent or **quickly outdated**.	The workflow is robust to small changes, and results are traceable, **relies on good practice** by researcher.	**Checks** reduce human error, though complex configurations can still hide subtle bugs.	CI and end-to-end automation **catch errors** early and often.

Tools **Attitudes**

How would you rate your own level now?

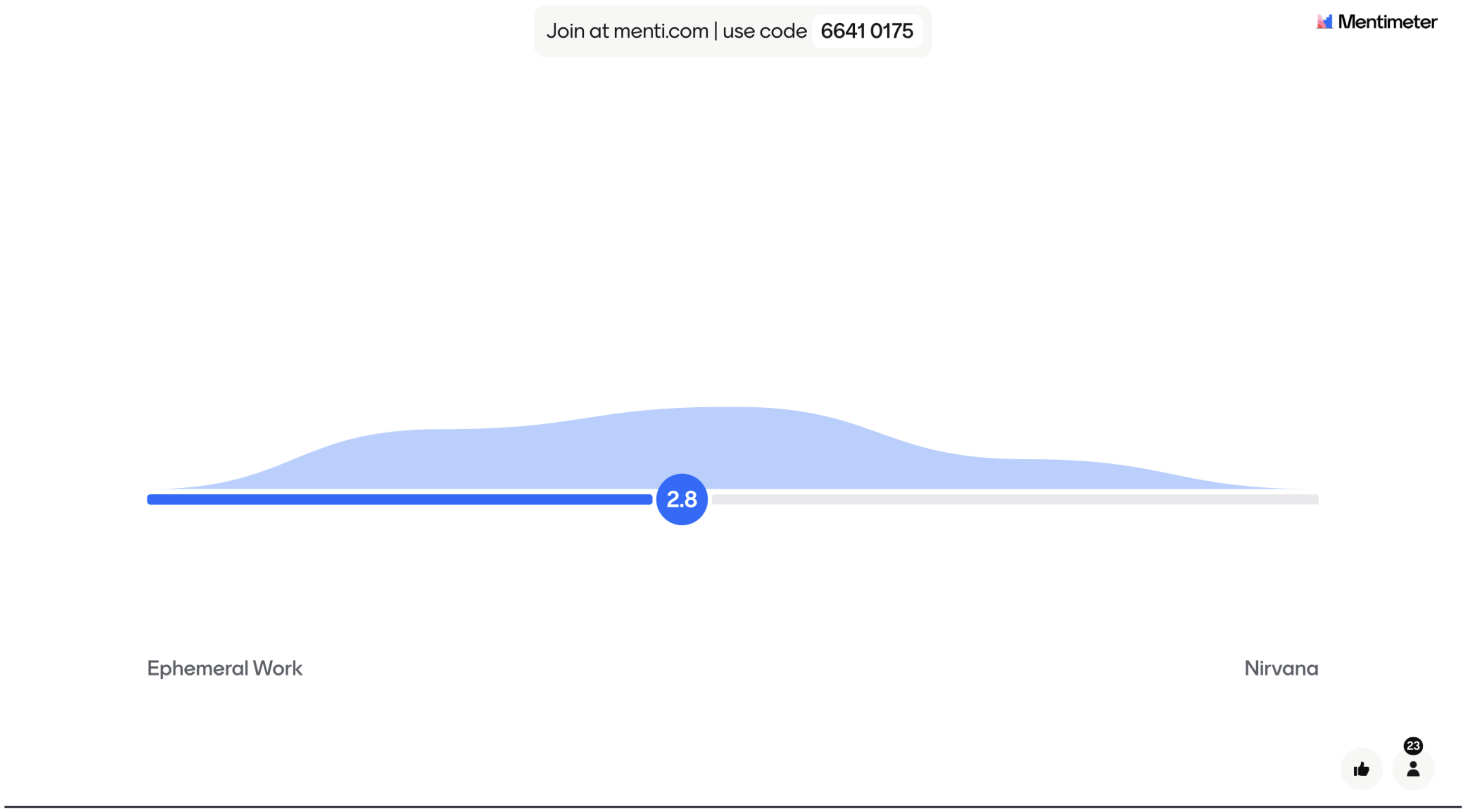
Move the scale on your phones or laptops!

Level 1 **Ephemeral Work**	Level 2 **Carpe Diem**	Level 3 **Sweet Spot**	Level 4 **Assembly Line**	Level 5 **Nirvana**
Manual and **interactive** analysis	Some **scripts**	**Version controlled** scripts	Scripts **combined** in single workflow	Single command reproduction **of it all**
Reliance on researcher's **memory**	**Comments** in README, inline or some notebook	**Workflow management** or well-structured scripts	Minimal steps to spin up **environment** using containers	Whole lifecycle is **documented** and **continuously tested**
Probably **not** reproducible	Reproducible if not **too long ago**	Reproducible on the **same** infrastructure by **same** person	Reproducible on/for **different** machines or collaborators	Reproducible for **anyone** after publication

Mentimeter

Join at menti.com | use code 6641 0175

How would you rate your own level now?



Join at menti.com | use code 6641 0175

Reasons for reproducible research & development

- Increasing **complexity** of questions asked
- **Size** of data & Switzerland
- **Observational** vs controlled experimental **data** for research
- Enabling **data-intensive** methods like AI
- Tackling **regulatory challenges** of medical and clinical research
- Facilitating adoption of **FAIR principles**

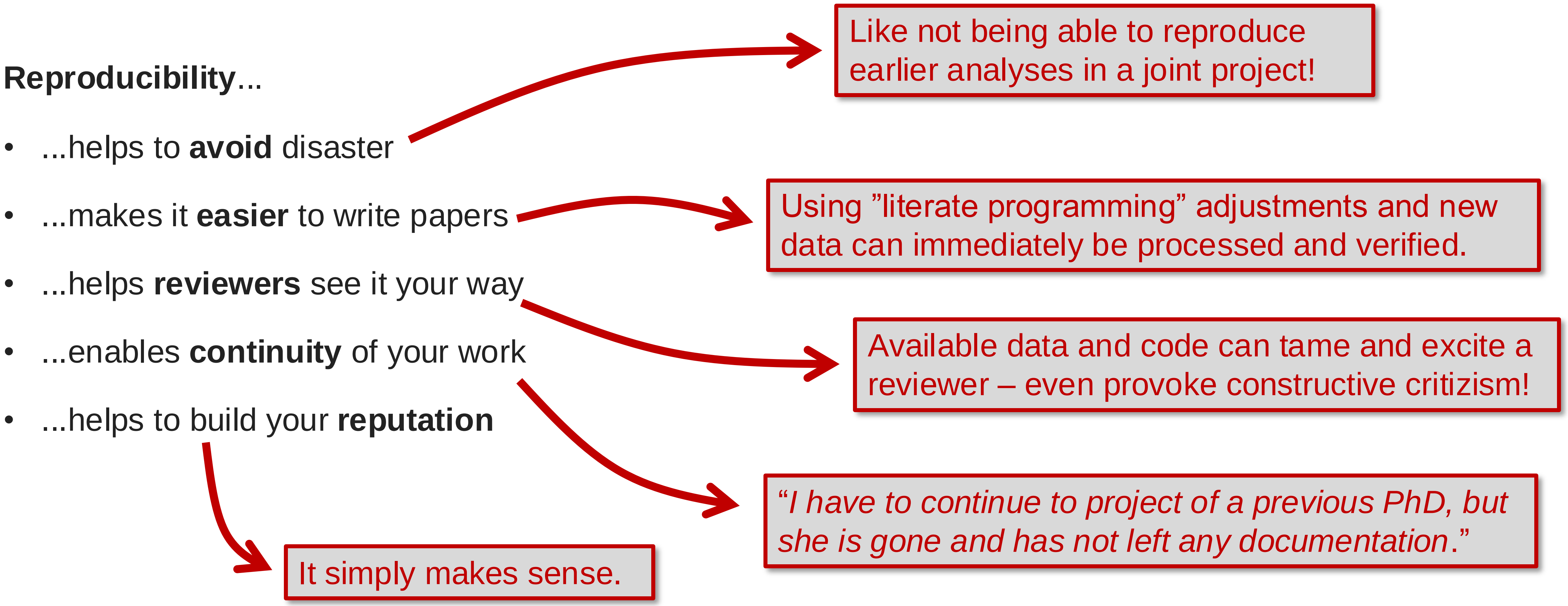
There is still a lot of work to do...

[1] Citation needed.

zero 🦊 given...

[1] Icon from https://www.flaticon.com/

Why bother at all?



Reproducibility...

- ...helps to **avoid** disaster
- ...makes it **easier** to write papers
- ...helps **reviewers** see it your way
- ...enables **continuity** of your work
- ...helps to build your **reputation**

Like not being able to reproduce earlier analyses in a joint project!

Using ”literate programming” adjustments and new data can immediately be processed and verified.

Available data and code can tame and excite a reviewer – even provoke constructive critizism!

“*I have to continue to project of a previous PhD, but she is gone and has not left any documentation*.”

It simply makes sense.

[1] Markowetz, F. Five selfish reasons to work reproducibly. Genome Biol 16, 274 (2015). https://doi.org/10.1186/s13059-015-0850-7

[2] Grant, S., Corker, K. S., Mellor, D. T., Stewart, S. L. K., Cashin, A. G., Lagisz, M., … Nosek, B. A. (2025, February 3). TOP 2025: An Update to the Transparency and Openness Promotion Guidelines. https://doi.org/10.31222/osf.io/nmfs6_v2

How to get started?

One step at a time...

nexus

Personalized Health Technologies

ETH zürich

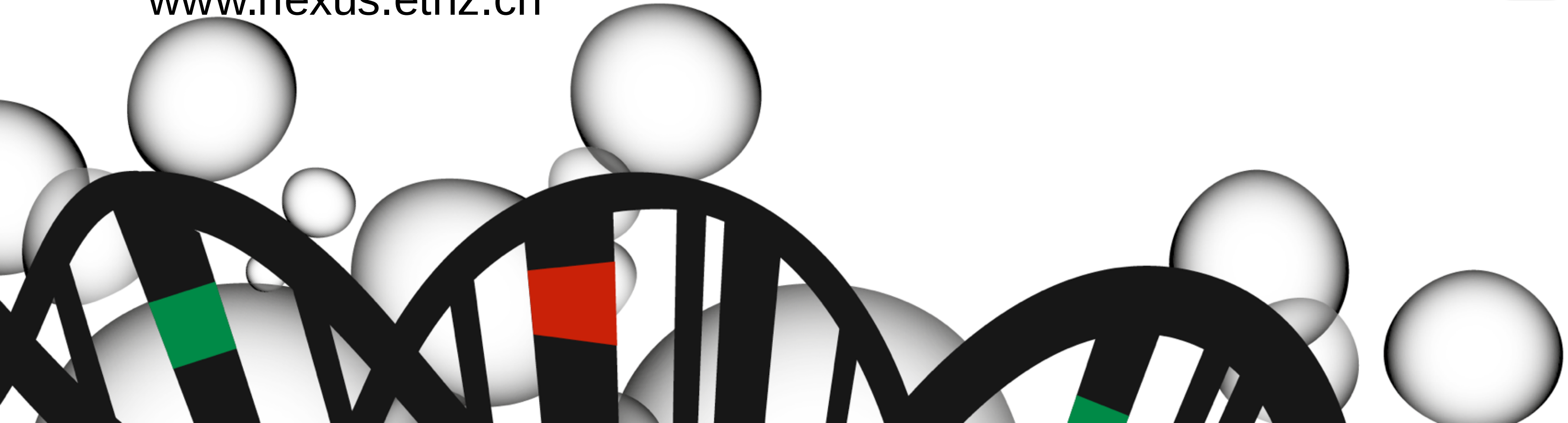
ETH zürich

nexus Personalized Health Technologies

Daniel Stekhoven
Director
stekhoven@nexus.ethz.ch

ETH Zürich
NEXUS Personalized Health Technologies
Wagistrasse 18
8952 Zürich (Schlieren)

www.nexus.ethz.ch



Appendix

How computational people do photo shootings...

How can we get to the next level?

	Level 1 **Ephemeral Work**	Level 2 **Carpe Diem**	Level 3 **Sweet Spot**	Level 4 **Assembly Line**	Level 5 **Nirvana**
Profile	**Manual** and **interacti**		**rolled** sc	**ned** in si	nd reproduction **of it**
	Reliance on researcher's **memory**	n READM some notebook	**anagemen** structured scripts	to spin up **environment** using containers	e is **documented** and **continuously tested**
	Probably **not** reprodu	f not **too**	n the **sa** y **same**	n/for **diff** ollaborato	r **anyone** after
Tools	• Spreadsheets with colours • Manual notes • Ad-hoc scripts • File sharing via em personal clouds	• Basic Git usage EADME fi oftware a usage age w/o	• GitHub/GitLab ualenv workflow s or Mak inerizatio ated with control (e.g., openBIS, labkey)	• Full containerization workflow e, Nextflo ing (e.g ersistent s for large • Continuous integration (GitHub tLab CI, J ch ELN	• Coordination (web) application nsive CI/CD epositories with enodo, Dryad) ocumentation (e.g., Sphinx, MkDocs) ab-to-publication
Pros & cons	Extremely difficult to **r** **share** analyses.	produce r environment or code **changes**.	ith some alyses if **t** step-by-step instructions.	d someti orators) c reproduce results with **minimal**	rators or future n **reproduce and** **extend** the work almost
	High risk of **irrecover** or lost steps.	n might r **quickly**	s robust esults a **es on go** researcher.	human x configu e bugs.	nd automation arly and often.

Script your manual steps.

Write a **basic README** or minimal documentation.

Start using **version control** for code tracking.

Record lab notes **digitally**.

Keep final data in a **consistent folder structure**.

Adopt **consistent environments.**

Automate key tasks using simple workflow scripting.

Automate key tasks using simple workflow scripting.

Migrate fully to an ELN.

Begin systematically archiving data.

Containerize your environment.

Use a **workflow manager** for structured pipelines.

Practice **systematic** research data management.

Attach DOIs to major datasets.

Clearly reference **lab protocols** (ELNs, SOPs) in the workflow documentation.

Integrate **CI** to confirm everything still works.

Expand **documentation** so others can plug-and-play.

Achieve a **single-command rebuild** of the entire project.

Automate **data** & **code deposition**.

Link **lab notes, protocols,** and **manuscript** in a single reference framework.