The Turing Way and other approaches to **reproducible** and **generalizable** research

in deep learning and cognitive neuroscience research

by **Martina Vilas** (she/her)

@martinagvilas

about me

about me

→ **cognitive computational neuroscience** PhD student

Roig Lab

Poeppel Lab

about me

- **cognitive computational neuroscience** PhD student
- core contributor of **The Turing Way**

Roig Lab

Poeppel Lab

what is reproducible research?

what is reproducible research?

same analytic steps on the same dataset produces same answer

		data	
		same	different
analysis	same	reproducible	replicable
	different	robust	generalisable

https://the-turing-way.netlify.app/reproducible-research/overview/overview-definitions.html

@martinagvilas

why is it important?

why is it important?

➔ **trust** other's work and ensure **continuity** of research

why is it important?

- ➔ **trust** other's work and ensure **continuity** of research
- ➔ avoid **misinformation**

why is it important?

- ➔ **trust** other's work and ensure **continuity** of research
- ➔ avoid **misinformation**
- ➔ easier **collaboration** and **review**

why is it important?

- ➔ **trust** other's work and ensure **continuity** of research
- ➔ avoid **misinformation**
- ➔ easier **collaboration** and **review**
- ➔ more **efficient analysis** and manuscript **writing**

https://the-turing-way.netlify.app/reproducible-research/overview/overview-benefit.html#rr-overview-benefits

SHARE

IN DEPTH | COMPUTER SCIENCE

Artificial intelligence faces reproducibility crisis

Matthew Hutson

+ See all authors and affiliations

Science 16 Feb 2018:
Vol. 359, Issue 6377, pp. 725-726
DOI: 10.1126/science.359.6377.725

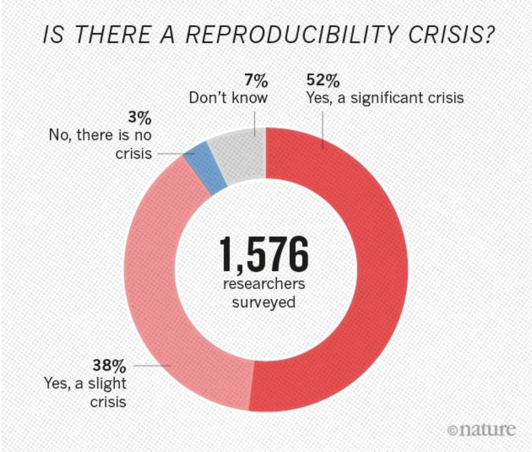
Published: 27 August 2015

Over half of psychology studies fail reproducibility test

Monya Baker

Nature (2015) | Cite this article

5170 Accesses | 42 Citations | 1256 Altmetric | Metrics



readme.md

Case studies

The term "case studies" is used here in a general sense to describe any study of reproducibility. A *reproduction* is an attempt to arrive at comparable results with identical data using computational methods described in a paper. A *refactor* involves refactoring existing code into frameworks and other reproducibility best practices while preserving the original data. A *replication* involves generating new data and applying existing methods to achieve comparable results. A *robustness test* applies various protocols, workflows, statistical models or parameters to a given data set to study their effect on results, either as a follow-up to an existing study or as a "bake-off". A *census* is a high-level tabulation conducted by a third party. A *survey* is a questionnaire sent to practitioners. A *case narrative* is an in-depth first-person account. An *independent discussion* utilizes a secondary independent author to interpret the results of a study as a means to improve inferential reproducibility.

Study	Field	Approach	Size
Glasziou et al 2008	Medicine	Census	80 studies
Baggerly & Coombes 2009	Cancer biology	Refactor	8 studies

barriers for reproducible research

barriers for reproducible research

publication bias towards novel findings

is not considered for **promotion**

held to **higher standards** than others

requires **additional skills**

takes **time**

support additional users

Kirstie Whitaker (2020). http://doi.org/10.5281/zenodo.3959180

@martinagvilas

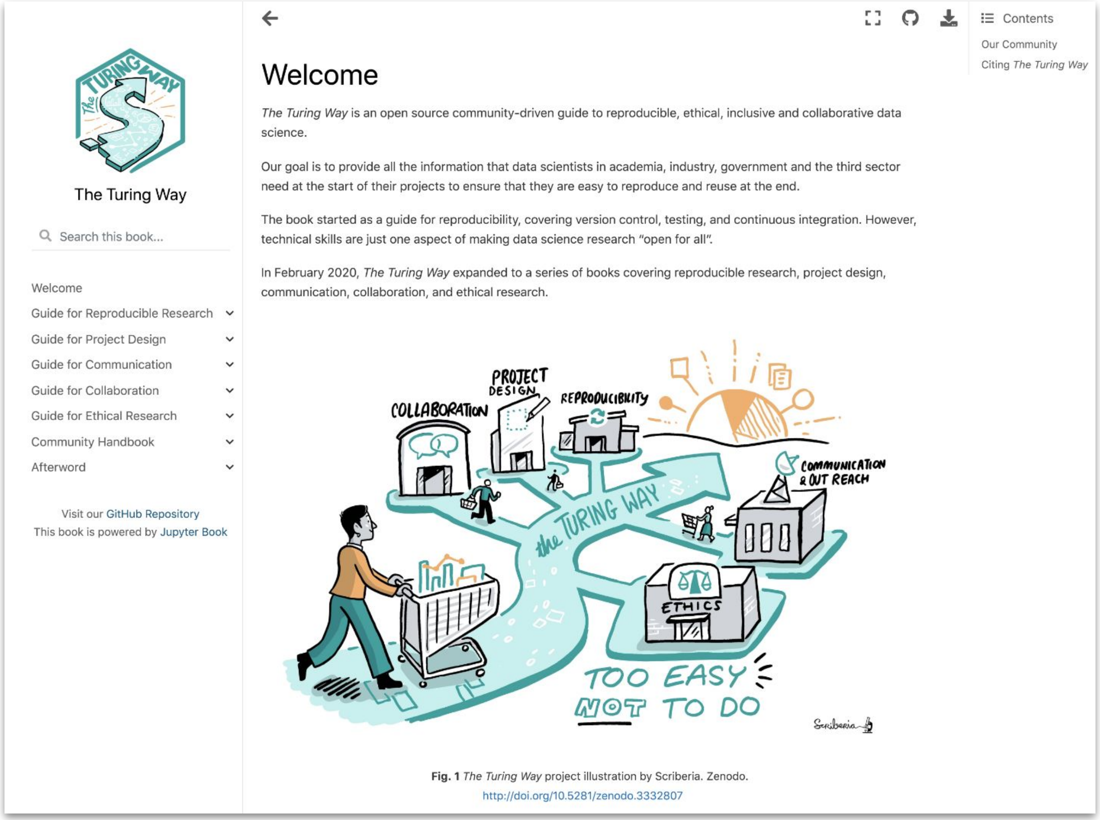
The Turing Way

The Turing Way

online guide to

- reproducible
- ethical
- inclusive
- collaborative

data science



The Turing Way

Search this book...

Welcome
Guide for Reproducible Research
Guide for Project Design
Guide for Communication
Guide for Collaboration
Guide for Ethical Research
Community Handbook
Afterword

Visit our GitHub Repository
This book is powered by Jupyter Book

Contents
Our Community
Citing *The Turing Way*

Welcome

The Turing Way is an open source community-driven guide to reproducible, ethical, inclusive and collaborative data science.

Our goal is to provide all the information that data scientists in academia, industry, government and the third sector need at the start of their projects to ensure that they are easy to reproduce and reuse at the end.

The book started as a guide for reproducibility, covering version control, testing, and continuous integration. However, technical skills are just one aspect of making data science research "open for all".

In February 2020, *The Turing Way* expanded to a series of books covering reproducible research, project design, communication, collaboration, and ethical research.

Fig. 1 *The Turing Way* project illustration by Scriberia. Zenodo. http://doi.org/10.5281/zenodo.3332807

https://the-turing-way.netlify.app/

@martinagvilas

The Turing Way

online guide to

- reproducible
- ethical
- inclusive
- collaborative

data science

Dr. Kirstie Whitaker

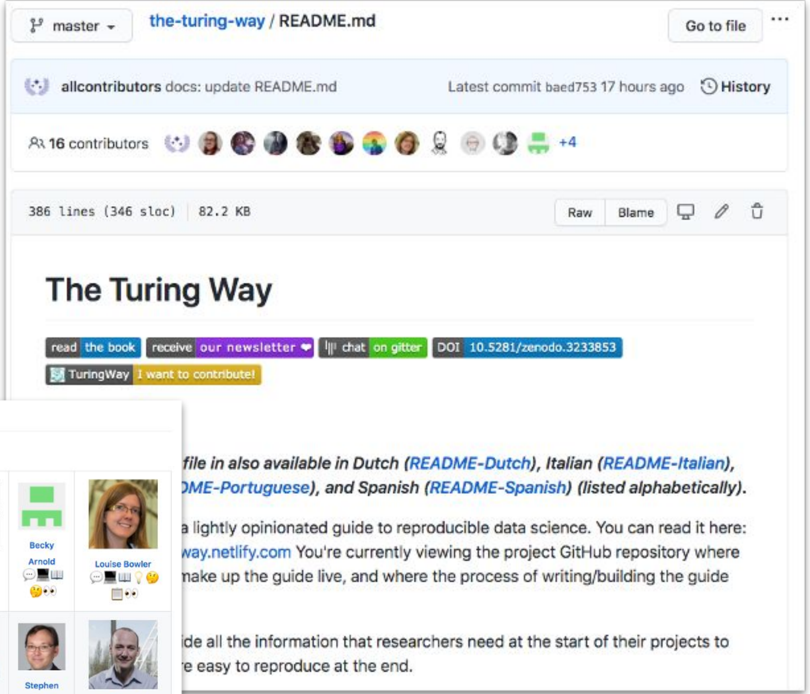
Programme Director for Tools, Practices and Systems

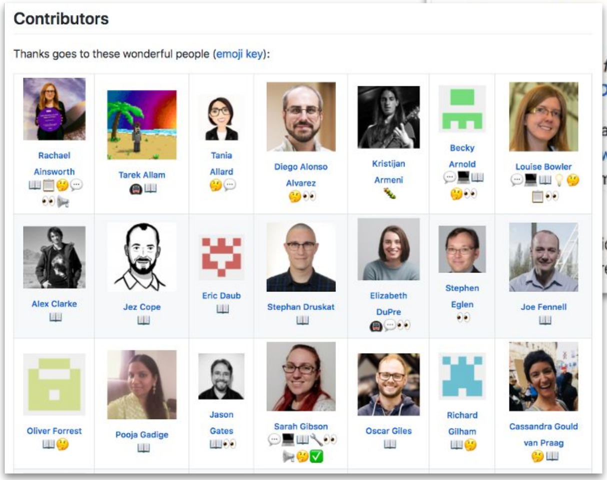
The Alan Turing Institute

The Turing Way

also...

- an open source project





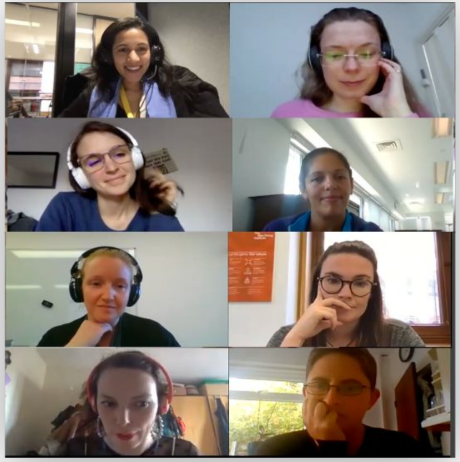
https://the-turing-way.netlify.app/

@martinagvilas

The Turing Way

also...

- a community



online collaboration cafés
and community calls

book dash events

The Turing Way

- Twitter:

twitter.com/turingway

- Newsletter:

tinyletter.com/TuringWay

- GitHub:

github.com/alan-turing-institute/the-turing-way

- Slack:

https://tinyurl.com/jointuringwayslack

The Turing Way

Search this book...

Guide for Reproducible Research

This guide covers topics related to skills, tools and best practices for research reproducibility.

The Turing Way defines reproducibility in data research as data and code being available to fully rerun the analysis.

There are several definitions of reproducibility in use, and we discuss these in more detail in the Definitions section of this chapter. While it is absolutely fine for us each to use different words, it will be useful for you to know how *The Turing Way* defines *reproducibility* to avoid misunderstandings when reading the rest of the handbook.

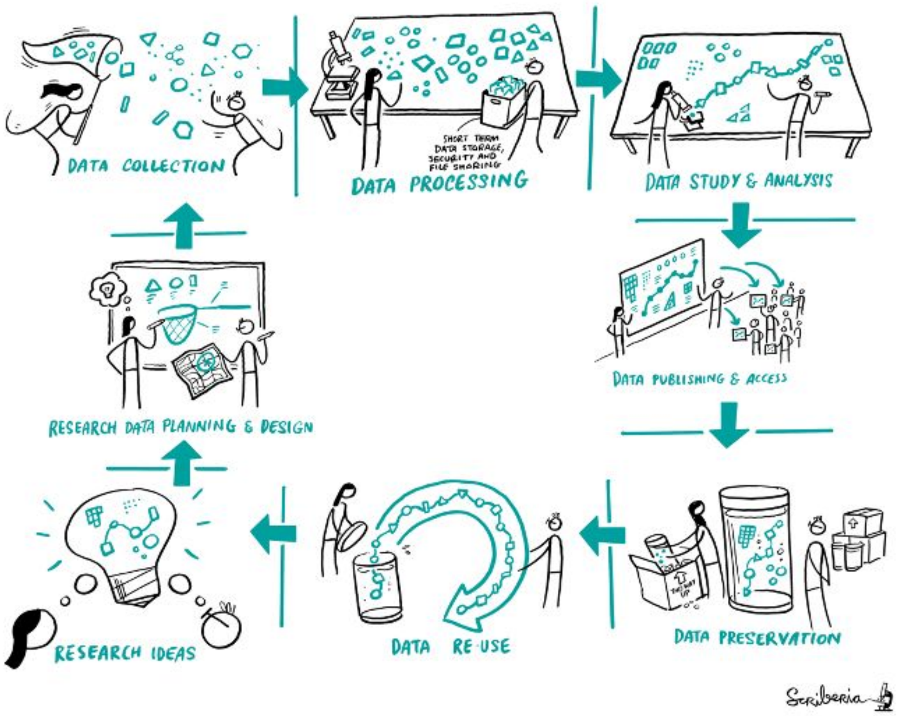
Fig. 2 *The Turing Way* project illustration by Scriberia. Zenodo. http://doi.org/10.5281/zenodo.3332807

The Turing Way started by defining reproducibility in the context of this handbook, laying out its importance for science and scientists, and providing an overview of the common concepts, tools and resources. The first few chapters were on version control, testing, and reproducible computational environments. Since the start of this project in 2019, many additional chapters have been written, edited, reviewed, read and promoted by over 100 contributors.

We welcome your contributions to improve these chapters, add other important concepts in reproducibility, and

The Turing Way

Search this book...

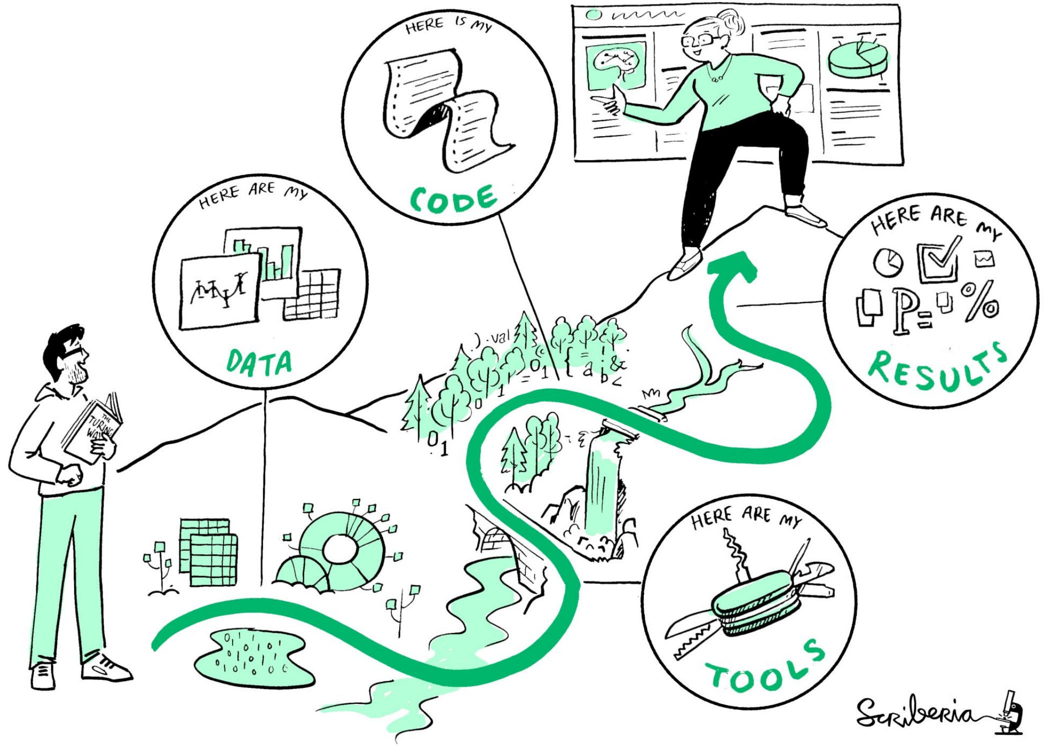


https://the-turing-way.netlify.app/reproducible-research/reproducible-research.html

@martinagvilas

computational reproducibility

computational reproducibility



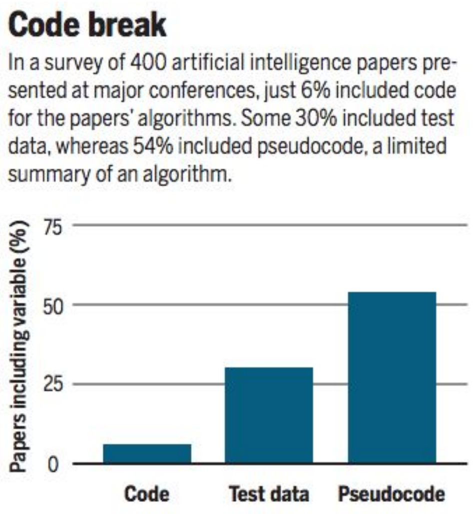
@martinagvilas

computational reproducibility

→ **share** code and data

computational reproducibility

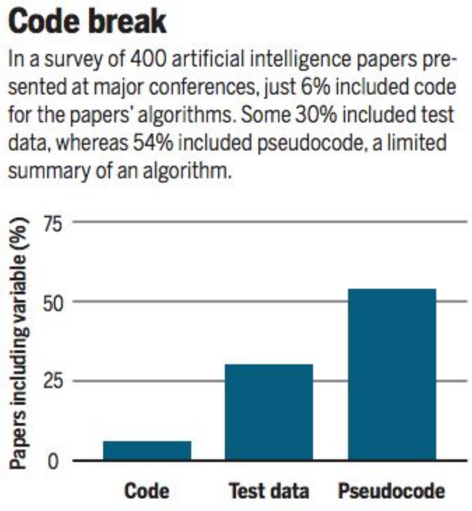
➔ **share** code and data

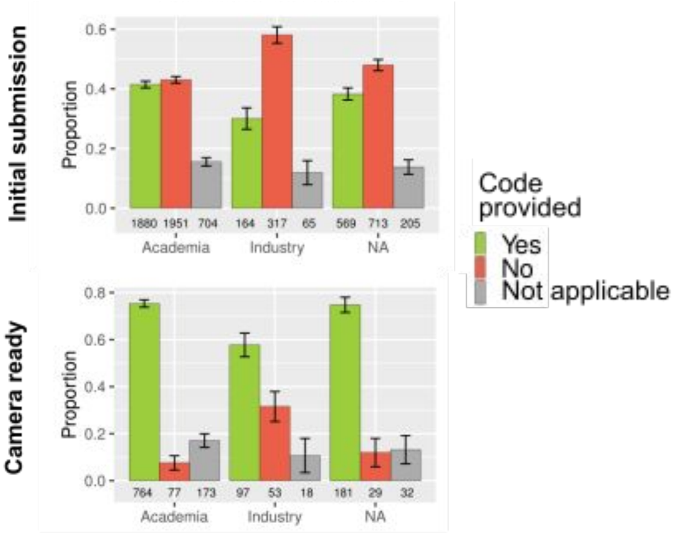


Hutson (2018) Artificial intelligence faces reproducibility crisis

computational reproducibility

→ **share** code and data



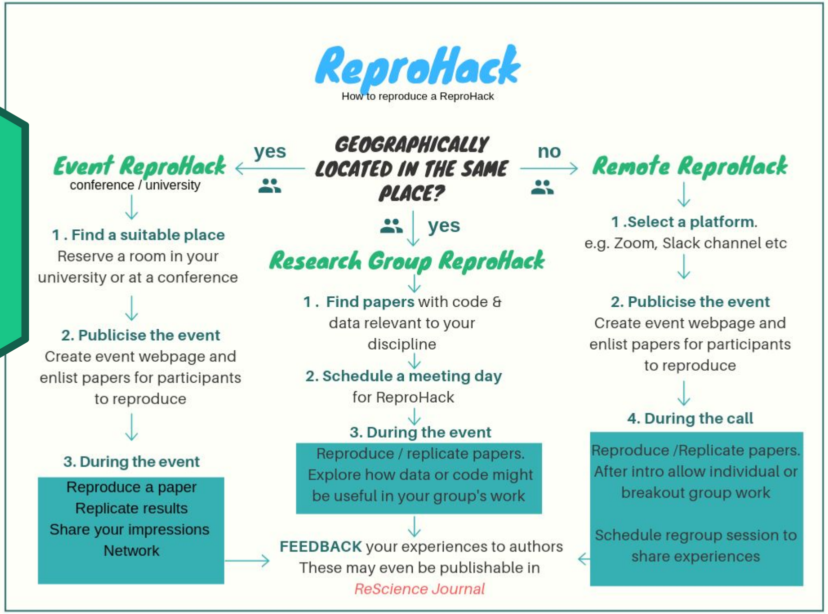


Pineau et al. (2021)Improving Reproducibility in Machine Learning Research

@martinagvilas

computational reproducibility

→ **share** code and data



computational reproducibility

→ **share** code and data

is not enough!

@martinagvilas

computational reproducibility

➔ capture, make executable and share the **computational environment**

computational reproducibility

➔ capture, make executable and share the **computational environment**

- operating system
- installed software (and its version)
- hardware

computational reproducibility

→ capture, make executable and share the **computational environment**

- operating system
- installed software (and its version)
- hardware

computational reproducibility

➔ use a **version control** system

computational reproducibility

- use a **version control** system
  - records changes to a file or set of files over time
  - provides access to any specific version

https://git-scm.com/doc

computational reproducibility

→ use a **version control** system

- records changes to a file or set of files over time
- provides access to any specific version

git

- changes are recorded using **snapshots**
- **distributed** version control system

computational reproducibility

- ➔ provide **good documentation** of how to reproduce the results

computational reproducibility

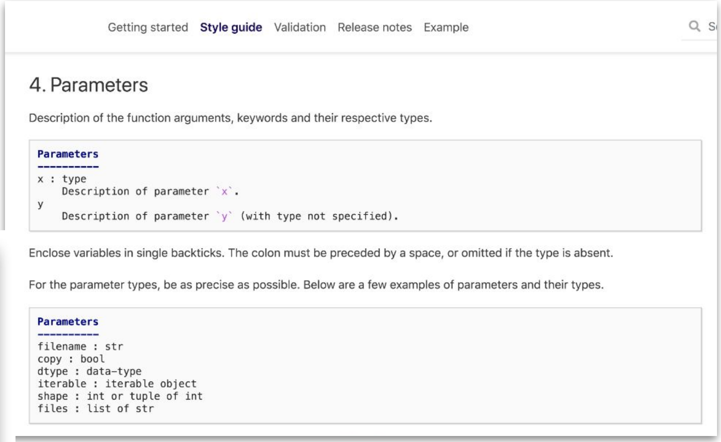
➔ provide **good documentation** of how to reproduce the results

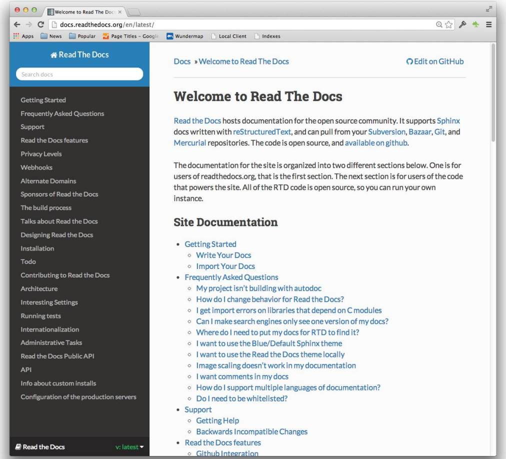
- step-by-step
- examples
- tutorials
- docstrings

computational reproducibility

➔ provide **good documentation** of how to reproduce the results

- step-by-step
- examples
- tutorials
- docstrings





@martinagvilas

computational reproducibility

- ➔ follow a **code style guide**

computational reproducibility

➔ follow a **code style guide**

- set of conventions of how to format your code
- e.g.
  - ✓ indentation
  - ✓ comments
  - ✓ imports
  - ✓ naming

computational reproducibility

→ follow a **code style guide**

- set of conventions of how to format your code
- e.g.
  - ✓ indentation
  - ✓ comments
  - ✓ imports
  - ✓ naming

PEP 8

```
# Correct:
i = i + 1
submitted += 1
x = x*2 - 1
hypot2 = x*x + y*y
c = (a+b) * (a-b)
```

```
# Wrong:
i=i+1
submitted +=1
x = x * 2 - 1
hypot2 = x * x + y * y
c = (a + b) * (a - b)
```

computational reproducibility

➔ **test** the code

computational reproducibility

➔ **test** the code

“You should not skip writing tests because you are short on time, you should write tests because you are short on time”

https://the-turing-way.netlify.app/reproducible-research/testing.html

computational reproducibility

→ **test** the code

“You should not skip writing tests because you are short on time, you should write tests because you are short on time”

```
expected_n_rows = 3
assert data.shape[0] == expected_n_rows, "shape mismatch"
```

```
---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
<ipython-input-3-c9f3f4600ddd> in <module>
      1 expected_n_rows = 3
----> 2 assert data.shape[0] == expected_n_rows, "shape mismatch"

AssertionError: shape mismatch
```

```
$ pytest
=========================== test session starts ============================
platform linux -- Python 3.x.y, pytest-6.x.y, py-1.x.y, pluggy-0.x.y
cachedir: $PYTHON_PREFIX/.pytest_cache
rootdir: $REGENDOC_TMPDIR
collected 1 item

test_sample.py F                                                     [100%]

================================= FAILURES =================================
_______________________________ test_answer ________________________________

    def test_answer():
>       assert inc(3) == 5
E       assert 4 == 5
E        +  where 4 = inc(3)

test_sample.py:6: AssertionError
========================= short test summary info ==========================
FAILED test_sample.py::test_answer - assert 4 == 5
============================ 1 failed in 0.12s =============================
```

computational reproducibility

- ➔ make the project **open source**

https://the-turing-way.netlify.app/reproducible-research/open/open-source.html

computational reproducibility

➔ make the project **open source**

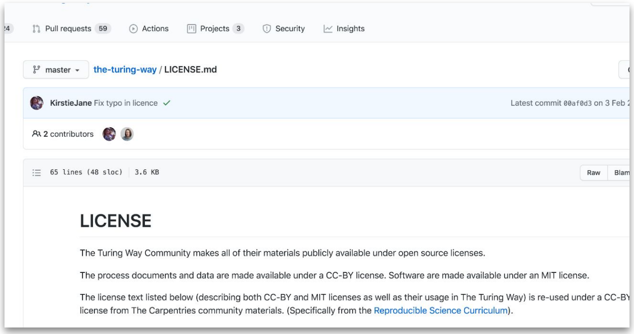
- use a software hosting platform

https://the-turing-way.netlify.app/reproducible-research/open/open-source.html
https://the-turing-way.netlify.app/reproducible-research/licensing.html

computational reproducibility

➔ make the project **open source**

- use a software hosting platform
- add a license

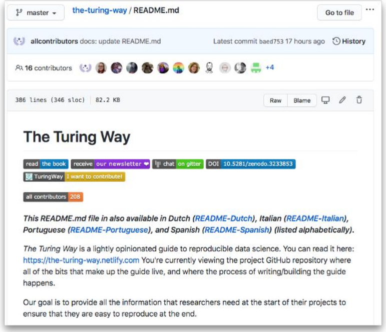


https://the-turing-way.netlify.app/reproducible-research/open/open-source.html
https://the-turing-way.netlify.app/reproducible-research/licensing.html

computational reproducibility

➔ make the project **open source**

- use a software hosting platform
- add a license
- provide community files



https://the-turing-way.netlify.app/reproducible-research/open/open-source.html
https://the-turing-way.netlify.app/reproducible-research/licensing.html

@martinagvilas

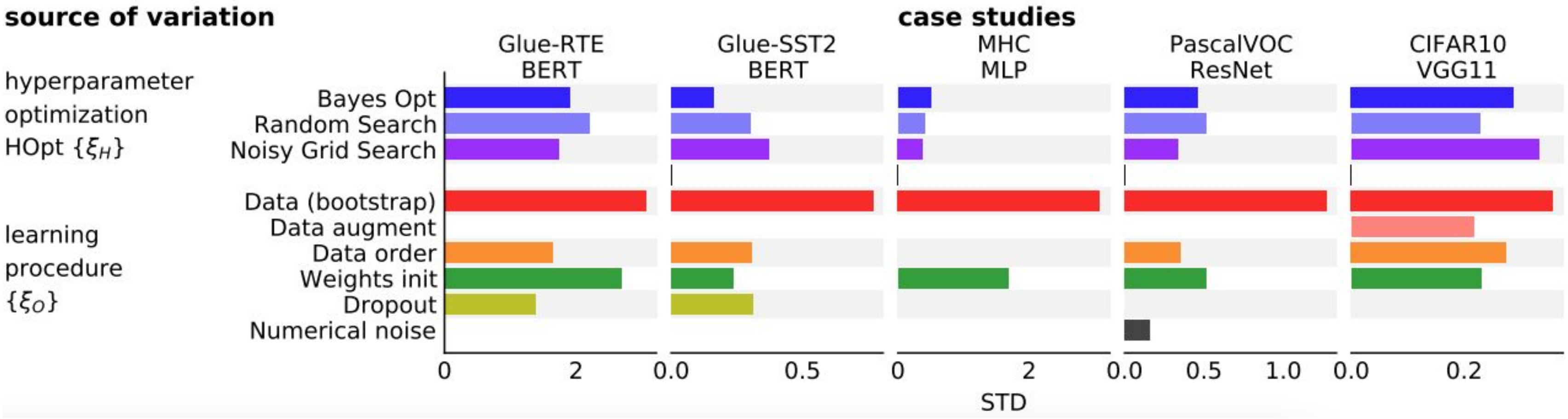
computational reproducibility

- → **share** code and data
- → capture, make executable and share the **computational environment**
- → use a **version control** system
- → provide **good documentation** of how to reproduce the results
- → follow a **code style guide**
- → **test** the code
- → make the project **open source**
- → **ETC!**

@martinagvilas

reproducibility in ML

reproducibility in ML



Bouthillier et al. (2021). Accounting for Variance in Machine Learning Benchmarks.

@martinagvilas

tools

tools

Out-of-the-box Reproducibility: A Survey of Machine Learning Platforms

Richard Isdahl
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway

Odd Erik Gundersen
Department of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
odderik@ntnu.no

Koustuv Sinha — about blog activities projects publications

Tools

Updated : 21st December, 2020

	Practice	Tools
1	Config Management	Hydra, OmegaConf, Pytorch Lightning
2	Checkpoint Management	Pytorch Lightning, TestTube
3	Logging	Tensorboard, Comet.ML, Weights & Biases, MLFlow, Visdom, Neptune
4	Seed	*Check best practices below*
-	Experiment Management	Pytorch Lightning, MLFlow, Determined.AI
5	Versioning	Github, Gitlab, Replicate.AI
6	Data Management	DVC, CML, Replicate.AI
7	Data analysis	Jupyter Notebook, papermill, JupyterLab, Google Colab
8	Reporting	Matplotlib, Seaborn , Pandas, Overleaf
9	Dependency Management	pip, conda, Poetry, Docker, Singularity, repo2docker
10	Open Source Release	Squash Commits, Binder
11	Effective Communication	ML Code Completeness Checklist, ML Reproducibility Checklist
12	Test and Validate	AWS, GCP, CodeOcean

Gundersen & Kjensmo (2019). Out-of-the-box Reproducibility: A Survey of Machine Learning Platforms.
https://www.cs.mcgill.ca/~ksinha4/practices_for_reproducibility/

logging

mlflow

Concepts

MLflow Tracking is organized around the concept of *runs*, which are executions of some piece of data science code. Each run records the following information:

Code Version

Git commit hash used for the run, if it was run from an MLflow Project.

Start & End Time

Start and end time of the run

Source

Name of the file to launch the run, or the project name and entry point for the run if run from an MLflow Project.

Parameters

Key-value input parameters of your choice. Both keys and values are strings.

Metrics

Key-value metrics, where the value is numeric. Each metric can be updated throughout the course of the run (for example, to track how your model's loss fun
MLflow records and lets you visualize the metric's full history.

Artifacts

Output files in any format. For example, you can record images (for example, PNGs), models (for example, a pickled scikit-learn model), and data files (for exa
artifacts.

https://mlflow.org/docs/latest/tracking.html

packaging

MLproject File

You can get more control over an MLflow Project by adding an `MLproject` file, which is a text file in YAML syntax, to the project's root directory. The following is an example of an `MLproject` file:

```
name: My Project

conda_env: my_env.yaml
# Can have a docker_env instead of a conda_env, e.g.
# docker_env:
#    image:  mlflow-docker-example

entry_points:
  main:
    parameters:
      data_file: path
      regularization: {type: float, default: 0.1}
    command: "python train.py -r {regularization} {data_file}"
  validate:
    parameters:
      data_file: path
    command: "python validate.py {data_file}"
```

The file can specify a name and a Conda or Docker environment, as well as more detailed information about each entry point. Specifically, each entry point defines a command to run an parameters to pass to the command (including data types).

Specifying an Environment

This section describes how to specify Conda and Docker container environments in an `MLproject` file. `MLproject` files cannot specify *both* a Conda environment and a Docker environment.

Conda environment

Include a top-level **`conda_env`** entry in the **`MLproject`** file. The value of this entry must be a *relative* path to a Conda environment YAML file within the MLflow project's directory. In following example:

```
conda_env: files/config/conda_environment.yaml
```

`conda_env` refers to an environment file located at **<MLFLOW_PROJECT_DIRECTORY>/files/config/conda_environment.yaml**, where **<MLFLOW_PROJECT_DIRECTORY>** is the path to the MLflow project's root directory.

Docker container environment

https://mlflow.org/docs/latest/projects.html

@martinagvilas

challenges

challenges

ML Reproducibility Challenge 2020 and Spring 2021

Welcome to the ML Reproducibility Challenge 2020! This is already the fourth edition of this event (see V1, V2, V3), and we are excited this year to announce that we are broadening our coverage of conferences and papers to cover several new top venues, including: NeurIPS, ICML, ICLR, ACL, EMNLP, CVPR and ECCV.

The primary goal of this event is to encourage the publishing and sharing of scientific results that are reliable and reproducible. In support of this, the objective of this challenge is to investigate reproducibility of papers accepted for publication at top conferences by inviting members of the community at large to select a paper, and verify the empirical results and claims in the paper by reproducing the computational experiments, either via a new implementation or using code/data or other information provided by the authors.

All submitted reports will be peer reviewed and shown next to the original papers on Papers with Code. Reports will be peer-reviewed via OpenReview. Every year, a small number of these reports, selected for their clarity, thoroughness, correctness and insights, are selected for publication in a special edition of the journal ReScience. (see J1, J2).

OpenReview.net Search OpenReview...

ML Reproducibility Challenge 2020

RC2020

TBD Mar 12 2021 https://paperswithcode.com/rc2020 reproducibility.challenge@gmail.com

Spring 2021

Submission Start: Oct 05 2020 12:00AM UTC-0, End: Jan 30 2021 11:59AM UTC-0

Accepted for ReScience | Submissions

[Re] Satellite Image Time Series Classification with Pixel-Set Encoders and Temporal Self-Attention
Maja Schneider, Marco Körner
06 Dec 2020 (modified: 01 Apr 2021) RC2020 Readers: Everyone 4 Replies
Show details

Reimplementation of FixMatch and Investigation on Noisy (Pseudo) Labels and Confirmation Errors of FixMatch
Ci Li, Ruibo Tu, Hui Zhang
06 Dec 2020 (modified: 01 Apr 2021) RC2020 Readers: Everyone 4 Replies
Show details

[Reproducibility Report] Rigging the Lottery: Making All Tickets Winners
Varun Sundar, Rajat Vadiraj Dwaraknath
22 Jan 2021 (modified: 03 Apr 2021) RC2020 Readers: Everyone 3 Replies
Show details

[Re] Can gradient clipping mitigate label noise?
David Mizrahi, Oğuz Kaan Yüksel, Aiday Marlen Kyzy
31 Jan 2021 (modified: 08 Apr 2021) RC2020 Readers: Everyone 4 Replies
Show details

checklists

checklists

The Machine Learning Reproducibility Checklist (v2.0, Apr.7 2020)

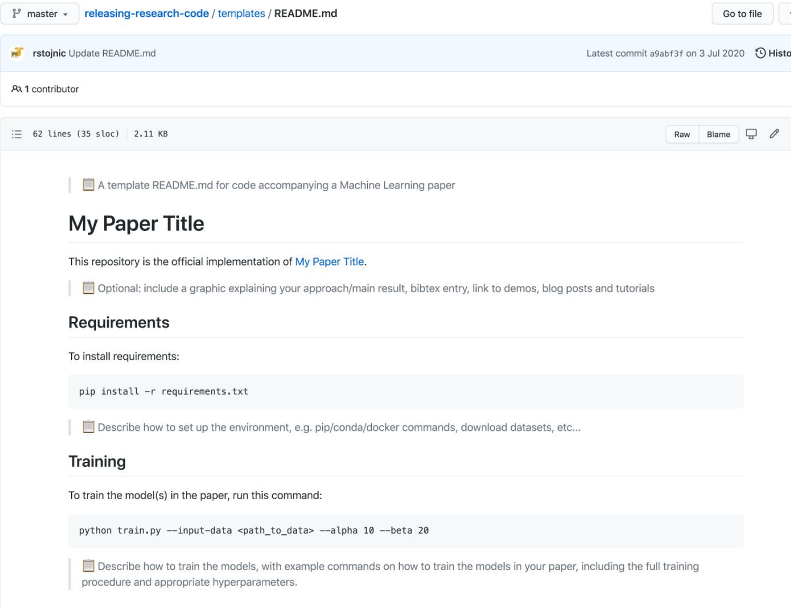
For all **models** and **algorithms** presented, check if you include:
- ❑ A clear description of the mathematical setting, algorithm, and/or model.
- ❑ A clear explanation of any assumptions.
- ❑ An analysis of the complexity (time, space, sample size) of any algorithm.

For any **theoretical claim**, check if you include:
- ❑ A clear statement of the claim.
- ❑ A complete proof of the claim.

For all **datasets** used, check if you include:
- ❑ The relevant statistics, such as number of examples.
- ❑ The details of train / validation / test splits.
- ❑ An explanation of any data that were excluded, and all pre-processing step.
- ❑ A link to a downloadable version of the dataset or simulation environment.
- ❑ For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.

For all shared **code** related to this work, check if you include:
- ❑ Specification of dependencies.
- ❑ Training code.
- ❑ Evaluation code.
- ❑ (Pre-)trained model(s).
- ❑ README file includes table of results accompanied by precise command to run to produce those results.

For all reported **experimental results**, check if you include:
- ❑ The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results.
- ❑ The exact number of training and evaluation runs.
- ❑ A clear definition of the specific measure or statistics used to report results.
- ❑ A description of results with central tendency (e.g. mean) & variation (e.g. error bars).
- ❑ The average runtime for each result, or estimated energy cost.
- ❑ A description of the computing infrastructure used.


master releasing-research-code / templates / README.md Go to file

rstojnic Update README.md Latest commit a9abf3f on 3 Jul 2020 Histo

1 contributor

62 lines (35 sloc) 2.11 KB Raw Blame

> 📋 A template README.md for code accompanying a Machine Learning paper

My Paper Title

This repository is the official implementation of My Paper Title.

> 📋 Optional: include a graphic explaining your approach/main result, bibtex entry, link to demos, blog posts and tutorials

Requirements

To install requirements:

```
pip install -r requirements.txt
```

> 📋 Describe how to set up the environment, e.g. pip/conda/docker commands, download datasets, etc...

Training

To train the model(s) in the paper, run this command:

```
python train.py --input-data <path_to_data> --alpha 10 --beta 20
```

> 📋 Describe how to train the models, with example commands on how to train the models in your paper, including the full training procedure and appropriate hyperparameters.

https://github.com/paperswithcode/releasing-research-code/blob/master/templates/README.md
https://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist.pdf

@martinagvilas

beyond reproducibility

beyond reproducibility

		data	
		same	different
analysis	same	reproducible	replicable
	different	robust	generalisable

https://the-turing-way.netlify.app/reproducible-research/overview/overview-definitions.html

@martinagvilas

model vs. **instantiation** of the model

model vs. **instantiation** of the model

↓

what we try to estimate

in science

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

@martinagvilas

model vs. **instantiation** of the model

"specific set of (trained) parameter values for a given model"

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

model vs. **instantiation** of the model

"useful as a probe to better understand a model"

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

model vs. **instantiation** of the model

"*Conclusions on a model that are limited to a single instance are very weak.*"

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

model vs. instantiation of the model

"Conclusions on a model that are limited to a ***single instance*** *are very weak."*

poor generalizability of **scientific claim**

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

≠ types of scientific claims

↓

≠ types of generalizability checks

@martinagvilas

≠ scientific aims

≠ scientific aims

DNN research → build a deep learning model that achieves best performance

≠ scientific aims

DNN research → build a deep learning model that achieves best performance

build a deep learning model to understand how the human brain implements cognitive functions ← **CCN research**

CCN ***mechanistic*** claim

CCN *mechanistic* claim

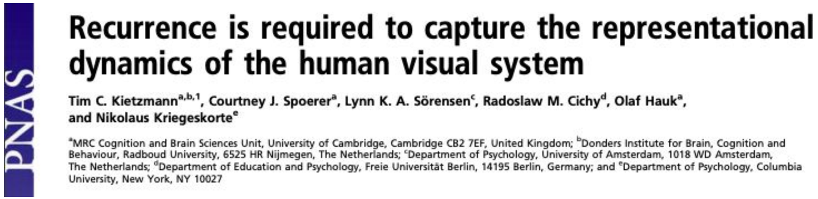
➔ how the brain **computes** information

CCN *mechanistic* claim

➔ how the brain **computes** information

e.g. inspect:

1. network architecture
2. learning goal (objective function)
3. learning update rule


PNAS

Recurrence is required to capture the representational dynamics of the human visual system

Tim C. Kietzmann[a,b,1], Courtney J. Spoerer[a], Lynn K. A. Sörensen[c], Radoslaw M. Cichy[d], Olaf Hauk[a], and Nikolaus Kriegeskorte[e]

[a]MRC Cognition and Brain Sciences Unit, University of Cambridge, Cambridge CB2 7EF, United Kingdom; [b]Donders Institute for Brain, Cognition and Behaviour, Radboud University, 6525 HR Nijmegen, The Netherlands; [c]Department of Psychology, University of Amsterdam, 1018 WD Amsterdam, The Netherlands; [d]Department of Education and Psychology, Freie Universität Berlin, 14195 Berlin, Germany; and [e]Department of Psychology, Columbia University, New York, NY 10027

Richards et al. (2019). *A deep learning framework for neuroscience.*

CCN ***representational*** claim

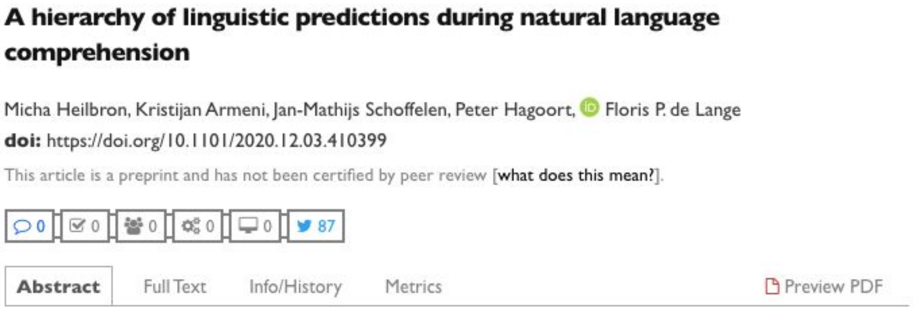
CCN *representational* claim

- ➔ **what** information is represented in the brain

CCN *representational* claim

→ **what** information is represented in the brain

e.g. dnn for feature generation


A hierarchy of linguistic predictions during natural language comprehension

Micha Heilbron, Kristijan Armeni, Jan-Mathijs Schoffelen, Peter Hagoort, Floris P. de Lange

doi: https://doi.org/10.1101/2020.12.03.410399

This article is a preprint and has not been certified by peer review [what does this mean?].

Abstract Full Text Info/History Metrics Preview PDF

CCN *representational* claim

➔ **what** information is represented in the brain

e.g. dnn for feature generation

➔ **how** information is represented in the brain

@martinagvilas

CCN ***representational*** claim

➔ **what** information is represented in the brain

e.g. dnn for feature generation

➔ **how** information is represented in the brain

e.g. representational geometries with RSA

The temporal evolution of conceptual object representations revealed through models of behavior, semantics and deep neural networks

B.B. Bankson [*,1], M.N. Hebart [1], I.I.A. Groen, C.I. Baker

Section on Learning and Plasticity, Laboratory of Brain and Cognition, National Institute of Mental Health, National Institutes of Health, Bethesda, MD 20892, USA

@martinagvilas

steps for generalizability

1. make the scientific claim very clear

2. determine which sources of variation should not affect the scientific claim

3. investigate the generalizability of the claim under those irrelevant conditions

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

steps for generalizability

1 make the scientific claim very clear

→ e.g. recurrent connections are needed for human visual processing

determine which sources of variation should not affect the scientific claim 2

3 investigate the generalizability of the claim under those irrelevant conditions

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

steps for generalizability

1 make the scientific claim very clear

→ e.g. recurrent connections explain feedback mechanisms in visual cortex during visual segmentation tasks

determine which sources of variation should not affect the scientific claim 2

3 investigate the generalizability of the claim under those irrelevant conditions

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

steps for generalizability

1 make the scientific claim very clear

determine which sources of variation should not affect the scientific claim 2

→ e.g. computational environment, initialization, test/train split, data order

→ task, dataset, objective function

3 investigate the generalizability of the claim under those irrelevant conditions

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

steps for generalizability

1 make the scientific claim very clear

determine which sources of variation should not affect the scientific claim 2

3 investigate the generalizability of the claim under those irrelevant conditions

Bouthillier et al. (2019). *Unreproducible research is reproducible.*

@martinagvilas

3 **investigate the generalizability** of the claim under those irrelevant conditions

3 **investigate the generalizability** of the claim under those irrelevant conditions

➔ compute every variation, or randomize it

3 **investigate the generalizability** of the claim under those irrelevant conditions

- ➔ compute every variation, or randomize it
- ➔ average variations

3 **investigate the generalizability** of the claim under those irrelevant conditions

- ➔ compute every variation, or randomize it
- ➔ average variations
- ➔ report distribution over variations

3 **investigate the generalizability** of the claim under those irrelevant conditions

- compute every variation, or randomize it
- average variations
- report distribution over variations
- use variations as observations for statistical analysis

3 **investigate the generalizability** of the claim under those irrelevant conditions

- → compute every variation, or randomize it
- → average variations
- → report distribution over variations
- → use variations as observations for statistical analysis
- → systematically study how the behavior of the model changes

summary

1. further work is needed to ensure research reproducibility across scientific fields
2. reproducibility is more than sharing your code and data
3. use as many computational reproducibility tools as possible
4. but also think about replicable, robust and generalizable research
5. each scientific field has its own reproducibility and generalizability challenges, even if they use the same analytical tool

thank you!

Acknowledgements:

- Kirstie Whitaker (@kirstie_j), Project Lead
- Malvika Sharan (@malvikasharan), Community Manager
- *The Turing Way* community, friends & collaborators

Useful links:

- Book: the-turing-way.netlify.com
- Twitter: twitter.com/turingway
- Newsletter: tinyletter.com/TuringWay
- GitHub: github.com/alan-turing-institute/the-turing-way
- Slack: https://tinyurl.com/jointuringwayslack
- Artwork by Scriberia: https://doi.org/10.5281/zenodo.3332808