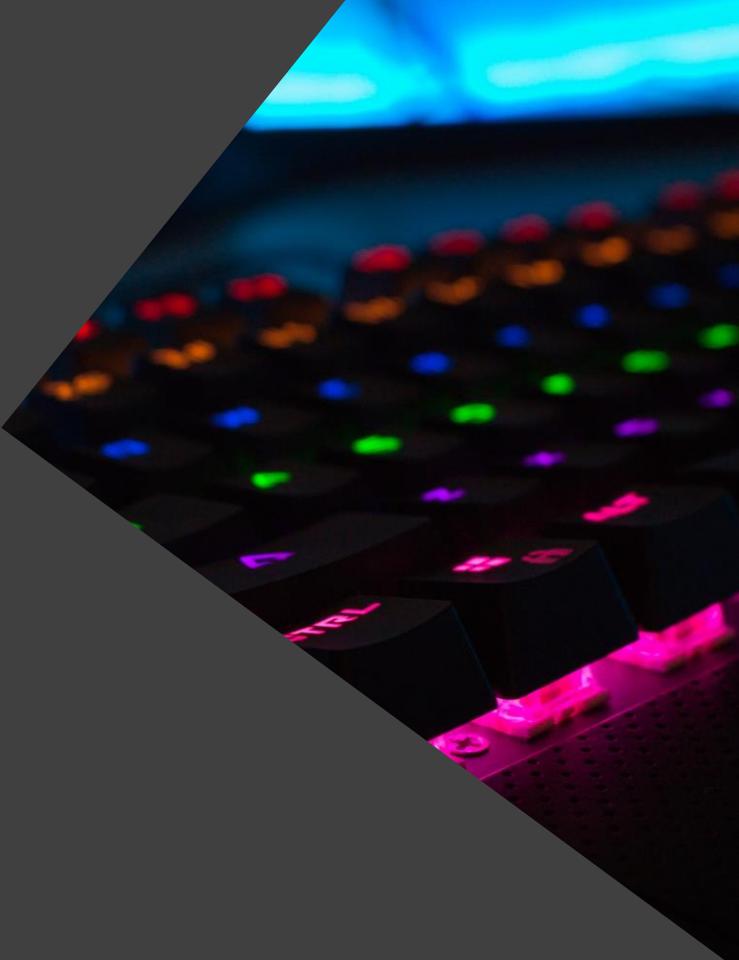


**The
Alan Turing
Institute**

The Turing Way
**A handbook for
reproducible research**
Kirstie Whitaker



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

The Turing Way is:

- a book
- a community
- a global collaboration
- a whole tonne of work



Rachael Ainsworth



Becky Arnold



Louise Bowler



Sarah Gibson



Patricia Hererich



James Hetherington



Rosie Higman



Anna Krystalli



Catherine Lawrence



Alex Morley

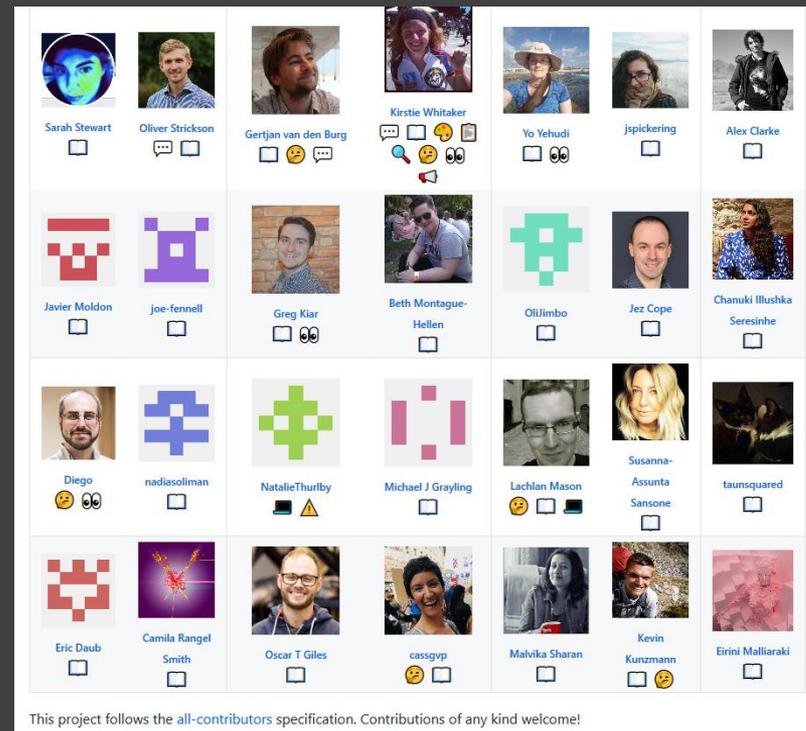
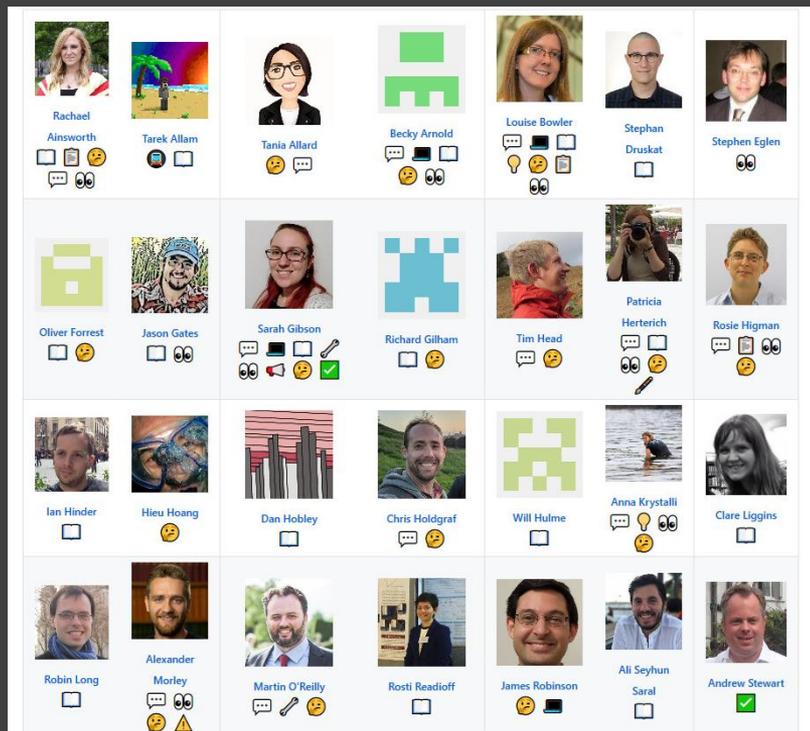


Martin O'Reilly



Binder Team

Thank you to all our contributors



This project follows the [all-contributors](https://allcontributors.org/) specification. Contributions of any kind welcome!

<https://github.com/alan-turing-institute/the-turing-way#contributors>
<https://allcontributors.org/docs/en/emoji-key>

#HackSTIR #TuringWay @kirstie_
<https://doi.org/10.5281/zenodo.3515937>

An introduction to me



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>



Picture credit: Chris Gorgolewski
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Academic errors have real world effects

	B	C	I	J	K	L	M
2			Real GDP growth				
3			Debt/GDP				
4	Country	Coverage	30 or less	30 to 60	60 to 90	90 or above	30 or less
26			3.7	3.0	3.5	1.7	5.5
27	Minimum		1.6	0.3	1.3	-1.8	0.8
28	Maximum		5.4	4.9	10.2	3.6	13.3
29							
30	US	1946-2009	n.a.	3.4	3.3	-2.0	n.a.
31	UK	1946-2009	n.a.	2.4	2.5	2.4	n.a.
32	Sweden	1946-2009	3.6	2.9	2.7	n.a.	6.3
33	Spain	1946-2009	1.5	3.4	4.2	n.a.	9.9
34	Portugal	1952-2009	4.8	2.5	0.3	n.a.	7.9
35	New Zealand	1948-2009	2.5	2.9	3.9	-7.9	2.6
36	Netherlands	1956-2009	4.1	2.7	1.1	n.a.	6.4
37	Norway	1947-2009	3.4	5.1	n.a.	n.a.	5.4
38	Japan	1946-2009	7.0	4.0	1.0	0.7	7.0
39	Italy	1951-2009	5.4	2.1	1.8	1.0	5.6
40	Ireland	1948-2009	4.4	4.5	4.0	2.4	2.9
41	Greece	1970-2009	4.0	0.3	2.7	2.9	13.3
42	Germany	1946-2009	3.9	0.9	n.a.	n.a.	3.2
43	France	1949-2009	4.9	2.7	3.0	n.a.	5.2
44	Finland	1946-2009	3.8	2.4	5.5	n.a.	7.0
45	Denmark	1950-2009	3.5	1.7	2.4	n.a.	5.6
46	Canada	1951-2009	1.9	3.6	4.1	n.a.	2.2
47	Belgium	1947-2009	n.a.	4.2	3.1	2.6	n.a.
48	Austria	1948-2009	5.2	3.3	-3.8	n.a.	5.7
49	Australia	1951-2009	3.2	4.9	4.0	n.a.	5.9
50							
51			4.1	2.8	2.8	=AVERAGE(L30:L44)	

<https://statmodeling.stat.columbia.edu/2013/04/16/memo-to-reinhart-and-rogo-off-i-think-its-best-to-admit-your-errors-and-go-on-from-there>
<https://www.bbc.co.uk/news/magazine-22223190>

#HackSTIR #TuringWay @kirstie_
<https://doi.org/10.5281/zenodo.3515937>

Academic errors have real world effects

	B	C	I	J	K	L	M
2	Real GDP growth						
3	Debt/GDP						
4	Country	Coverage	30 or less	30 to 60	60 to 90	90 or above	30 or less
26			3.7	3.0	3.5	1.7	5.5
27	Minimum		1.6	0.3	1.3	-1.8	0.8
28	Maximum		5.4	4.9	10.2	3.6	13.3
29							
30	US	1946-2009	n.a.	3.4	3.3	-2.0	n.a.
31	UK	1946-2009	n.a.	2.4	2.5	2.4	n.a.
32	Sweden	1946-2009	3.6	2.9	2.7	n.a.	6.3
33	Spain	1946-2009	1.5	3.4	4.2	n.a.	9.9
34	Portugal	1952-2009	4.8	2.5	0.3	n.a.	7.9
35	New Zealand	1948-2009	2.5	2.9	3.9	-7.9	2.6
36	Netherlands	1956-2009	4.1	2.7	1.1	n.a.	6.4
37	Norway	1947-2009	3.4	5.1	n.a.	n.a.	5.4
38	Japan	1946-2009	7.0	4.0	1.0	0.7	7.0
39	Italy	1951-2009	5.4	2.1	1.8	1.0	5.6
40	Ireland	1948-2009	4.4	4.5	4.0	2.4	2.9
41	Greece	1970-2009	4.0	0.3	2.7	2.9	13.3
42	Germany	1946-2009	3.9	0.9	n.a.	n.a.	3.2
43	France	1949-2009	4.9	2.7	3.0	n.a.	5.2
44	Finland	1946-2009	3.8	2.4	5.5	n.a.	7.0
45	Denmark	1950-2009	3.5	1.7	2.4	n.a.	5.6
46	Canada	1951-2009	1.9	3.6	4.1	n.a.	2.2
47	Belgium	1947-2009	n.a.	4.2	3.1	2.6	n.a.
48	Austria	1948-2009	5.2	3.3	-3.8	n.a.	5.7
49	Australia	1951-2009	3.2	4.9	4.0	n.a.	5.9
50							
51			4.1	2.8	2.8	=AVERAGE(L30:L44)	

BBC Sign in News Sport Weather iPlayer Sounds

NEWS

Home UK World Business Politics Tech Science Health Family & Education

Magazine

Reinhart, Rogoff... and Herndon: The student who caught out the profs

By Ruth Alexander
BBC News

© 20 April 2013

f t Share

This week, economists have been astonished to find that a famous academic paper often used to make the case for austerity cuts contains major errors. Another surprise is that the mistakes, by two eminent Harvard professors, were spotted by a student doing his homework.

It's 4 January 2010, the Marriott Hotel in Atlanta. At the annual meeting of the American Economic Association, Professor Carmen Reinhart and the former chief economist of the International Monetary Fund, Ken Rogoff, are presenting a research paper called Growth in a Time of Debt.



<https://statmodeling.stat.columbia.edu/2013/04/16/memo-to-reinhart-and-rogooff-i-think-its-best-to-admit-your-errors-and-go-on-from-there>
<https://www.bbc.co.uk/news/magazine-22223190>

#HackSTIR #TuringWay @kirstie_
<https://doi.org/10.5281/zenodo.3515937>

The humans are the
hardest part of
reproducibility



		Data	
		Same	Different
Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

<https://the-turing-way.netlify.com/reproducibility/03/definitions.html>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Is not considered
for promotion

Held to higher
standards than
others

Publication bias
towards novel
findings

Barriers to reproducible research

Requires
additional
skills

Plead the 5th

Support additional
users

Takes time

<https://doi.org/10.6084/m9.figshare.5537101>
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

The Turing Institute



<https://www.turing.ac.uk/news/enigma-machine-goes-display-alan-turing-institute>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>



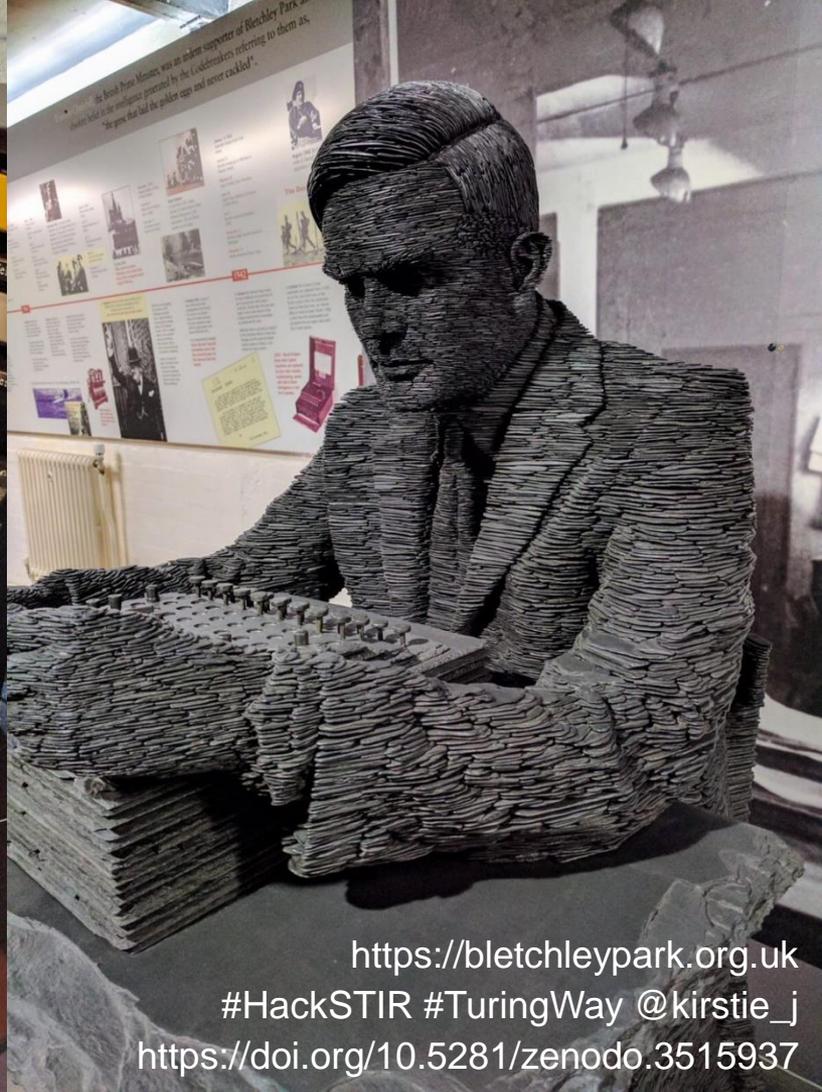
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>



<https://www.bbc.co.uk/programmes/p0704h04>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>



<https://bletchleypark.org.uk>
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

University network



THE UNIVERSITY
of EDINBURGH



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

The Institute's partners and collaborators



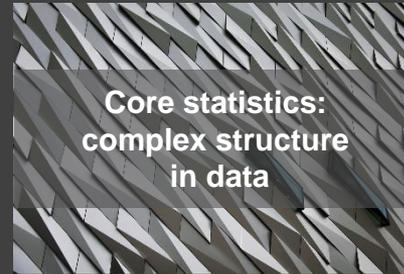
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Challenges

Advance data science and artificial intelligence to...



Core capabilities



Martin O'Reilly

“Make reproducible research too easy not to do.”



<https://www.turing.ac.uk/people/researchers/martin-oreilly>
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Martin O'Reilly

“Make reproducible
research too easy not to
do.

Do you need a biscuit?”



<https://www.turing.ac.uk/people/researchers/martin-oreilly>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Martin O'Reilly

“Make reproducible research too easy not to do.

Do you need a biscuit?

If we can't do it here, we can't do it at all.”



<https://www.turing.ac.uk/people/researchers/martin-oreilly>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

The Turing Way



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

1. Introduction

2. Reproducibility

3. Open Research

4. Version Control

5. Collaborating on GitHub/GitLab

6. Research Data Management

7. Reproducible Environments

8. Testing

9. Reviewing

10. Continuous Integration

11. Reproducible Research with
Make

12. Risk Assessment

Welcome to the Turing Way

The Turing Way is a lightly opinionated guide to reproducible data science.

Our goal is to provide all the information that researchers need at the start of their projects to ensure that they are easy to reproduce at the end.

This also means making sure PhD students, postdocs, PIs and funding teams know which parts of the “responsibility of reproducibility” they can affect, and what they should do to nudge data science to being more efficient, effective and understandable.

A bit more background

Reproducible research is necessary to ensure that scientific work can be trusted. Funders and publishers are beginning to require that publications include access to the underlying data and the analysis code. The goal is to ensure that all results can be independently verified and built upon in future work. This is sometimes easier said than done. Sharing these research outputs means understanding data management, library sciences, software development, and continuous integration techniques: skills that are not widely taught or expected of academic researchers and data scientists.

The Turing Way is a handbook to support students, their supervisors, funders and journal editors

<https://the-turing-way.netlify.com/introduction/introduction>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

1. Introduction

2. Reproducibility

3. Open Research

4. Version Control

5. Collaborating on GitHub/GitLab

6. Research Data Management

7. Reproducible Environments

8. Testing

9. Reviewing

10. Continuous Integration

11. Reproducible Research with Make

12. Risk Assessment



Welcome to the Turing Way

The Turing Way is a lightly opinionated guide to reproducible data science.

Our goal is to provide all the information that researchers need at the start of their projects to ensure that they are easy to reproduce at the end.

This also means making sure PhD students, postdocs, PIs and funding teams know which parts of the “responsibility of reproducibility” they can affect, and what they should do to nudge data science to being more efficient, effective and understandable.

A bit more background

Reproducible research is necessary to ensure that scientific work can be trusted. Funders and publishers are beginning to require that publications include access to the underlying data and the analysis code. The goal is to ensure that all results can be independently verified and built upon in future work. This is sometimes easier said than done. Sharing these research outputs means understanding data management, library sciences, software development, and continuous integration techniques: skills that are not widely taught or expected of academic researchers and data scientists.

The Turing Way is a handbook to support students, their supervisors, funders and journal editors

<https://the-turing-way.netlify.com/introduction/introduction>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

1. Introduction

2. Reproducibility

3. Open Research

4. Version Control

5. Collaborating on GitHub/GitLab

6. Research Data Management

7. Reproducible Environments

8. Testing

9. Reviewing

10. Continuous Integration

11. Reproducible Research with Make

12. Risk Assessment



Welcome to the Turing Way

The Turing Way is a lightly opinionated guide to reproducible data science.

Our goal is to provide all the information that researchers need at the start of their projects to ensure that they are easy to reproduce at the end.

This also means making sure PhD students, postdocs, PIs and funding teams know which parts of the “responsibility of reproducibility” they can affect, and what they should do to nudge data science to being more efficient, effective and understandable.

A bit more background

Reproducible research is necessary to ensure that scientific work can be trusted. Funders and publishers are beginning to require that publications include access to the underlying data and the analysis code. The goal is to ensure that all results can be independently verified and built upon in future work. This is sometimes easier said than done. Sharing these research outputs means understanding data management, library sciences, software development, and continuous integration techniques: skills that are not widely taught or expected of academic researchers and data scientists.

The Turing Way is a handbook to support students, their supervisors, funders and journal editors



<https://the-turing-way.netlify.com/introduction/introduction>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Is not considered
for promotion

Held to higher
standards than
others

Publication bias
towards novel
findings

Barriers to reproducible research

Requires
additional
skills

Plead the 5th

Support additional
users

Takes time

<https://doi.org/10.6084/m9.figshare.5537101>
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>



#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Wilkinson et al., Sci. Data, 2016. doi: 10.1038/sdata.2016.18

Catherine Lawrence

“We should ensure all our processes for running programmes are FAIR.

- Findable (intranet)
- Accessible (EDI)
- Interoperable across programmes and projects
- Reusable (bus factor)”



<https://www.turing.ac.uk/people/business-team/catherine-lawrence>

#HackSTIR #TuringWay @kirstie_j

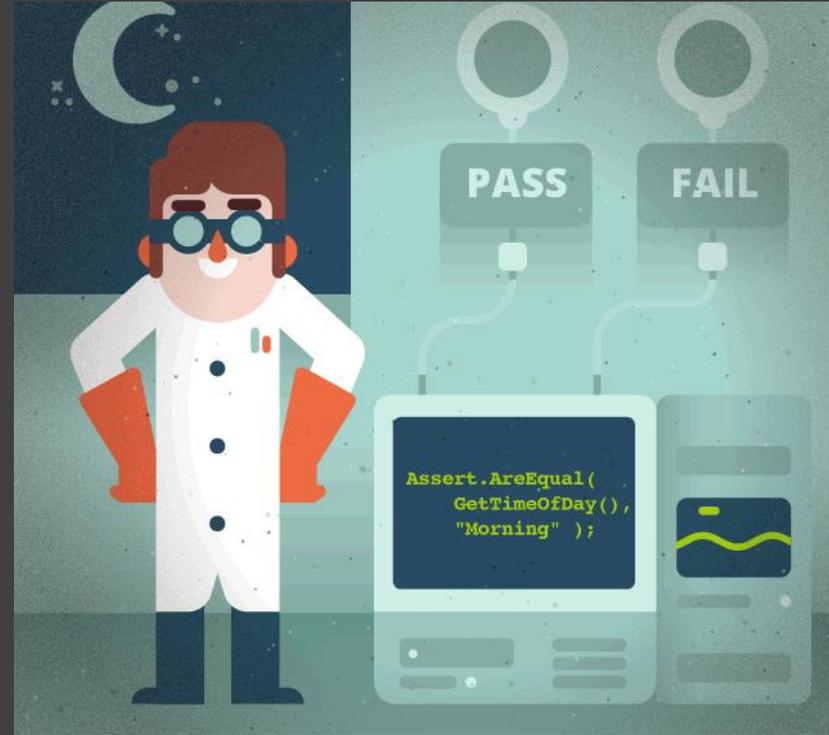
<https://doi.org/10.5281/zenodo.3515937>

Testing for research

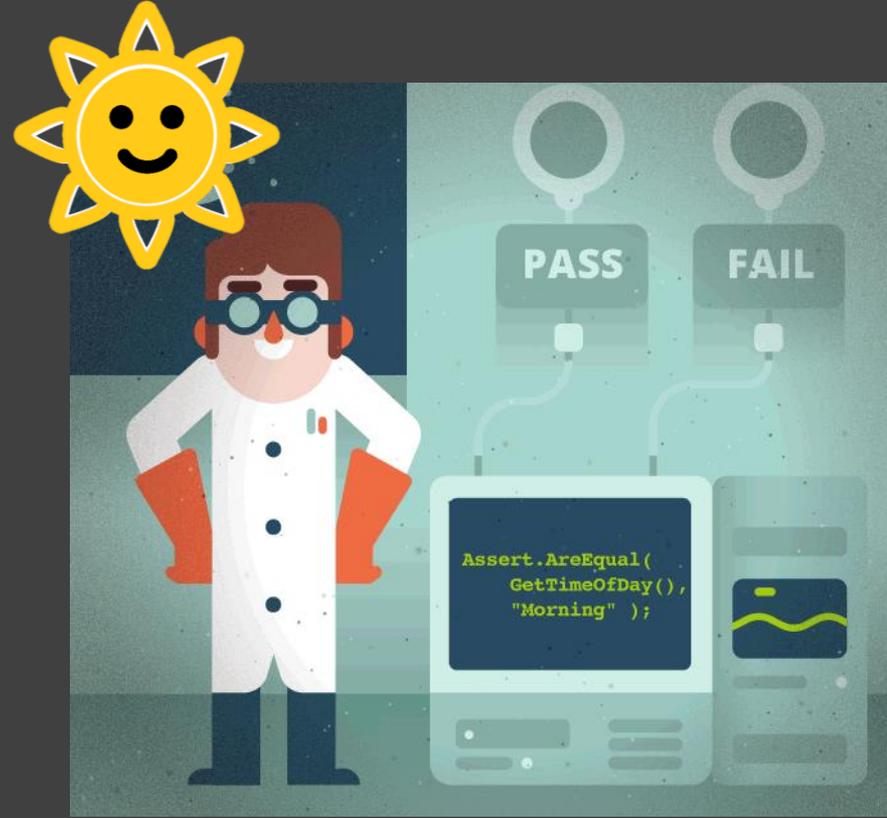


#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Is your code doing what
you think its doing?

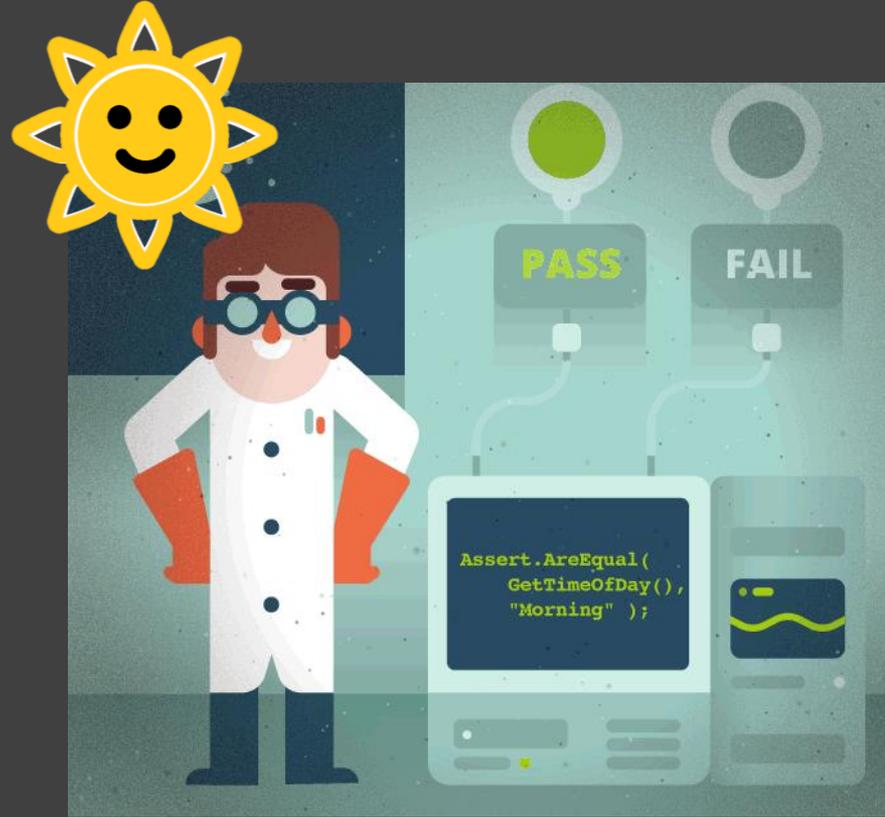


Is your code doing what
you think its doing?



Is your code doing what
you think its doing?

```
Assert.AreEqual(  
    GetTimeOfDay(),  
    "Morning" )
```



Is your code doing what
you think its doing?

```
Assert.AreEqual(  
    GetTimeOfDay(),  
    "Morning" )
```



Louise Bowler

“Add a test before you change anything.”



<https://www.turing.ac.uk/people/researchers/louise-bowler>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Louise Bowler

“Add a test before you change anything.

Particularly if you’re just going to tidy up your code before sharing it.”



<https://www.turing.ac.uk/people/researchers/louise-bowler>

#HackSTIR #TuringWay @kirstie_j

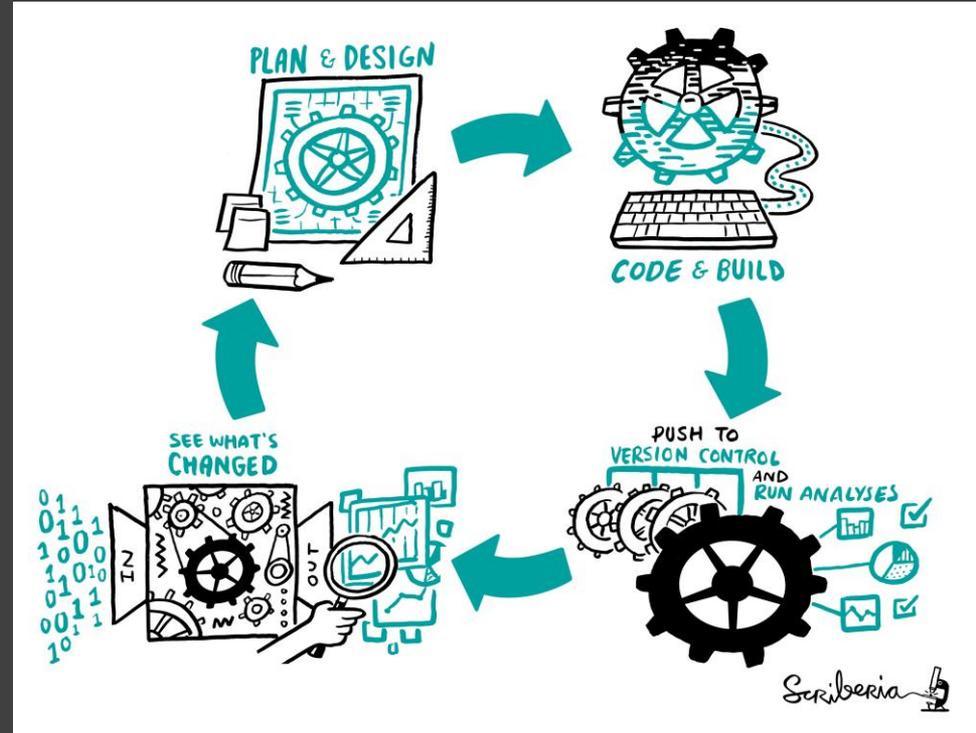
<https://doi.org/10.5281/zenodo.3515937>

Continuous Analysis



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

- Plan and design your experiment
- Write down those steps in code
- Push to version control and run the analyses
 - Traditionally done on the cloud, but the important part is that all steps are run every time
- Test to see what's changed



Dashboard Changelog Documentation Help

Search all repositories

My Repositories Running (0/0) +

alan-turing-institute/PosteriorB # 98

Duration: 2 hrs 11 min 35 sec
Finished: about 9 hours ago

alan-turing-institute/signatures # 1

Duration: 1 min 41 sec
Finished: about 12 hours ago

bids-standard/bids-specificatio # 506

Duration: 32 sec
Finished: a day ago

alan-turing-institute / signatures-psychiatry

build unknown

Current Branches Build History Pull Requests More options

lab-add-synth-data Add travis config - #1 passed Restart build

Commit 023d957 Ran for 1 min 41 sec
Compare e63a607 . . 023d957 about 12 hours ago
Branch lab-add-synth-data

Louise Bowler

Python: 2.7

Job log View config

<https://github.com/alan-turing-institute/signatures-psychiatry>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

[https://the-turing-way.netlify.com/
continuous_integration/continuous_integration.html](https://the-turing-way.netlify.com/continuous_integration/continuous_integration.html)



Dashboard

Search all repositories

My Repositories Running (0/0) +

alan-turing-institute/Posterior # 98

Duration: 2 hrs 11 min 35 sec

Finished: about 9 hours ago

alan-turing-institute/signatures # 1

Duration: 1 min 41 sec

Finished: about 12 hours ago

bids-standard/bids-specification # 50

Duration: 32 sec

Finished: a day ago

Job log View config

Remove log Raw log

```

1 Worker information
6 Build system information
413
414 docker stop/waiting
416
417 $ git clone --depth=50 --branch=lab-add-synth-data https://github.com/alan-turing-institute
427
428 $ source ~/virtualenv/python2.7/bin/activate
429 $ python --version
430 Python 2.7.14
431 $ pip --version
432 pip 9.0.1 from /home/travis/virtualenv/python2.7.14/lib/python2.7/site-packages (python 2.7)
433 $ pip install -r requirements.txt
518 $ pytest -v
519 ===== test session starts =====
520 platform linux2 -- Python 2.7.14, pytest-4.4.1, py-1.5.2, pluggy-0.11.0 -- /home/travis/virtualenv/python2.7.14
/bin/python
521 cachedir: .pytest_cache
522 rootdir: /home/travis/build/alan-turing-institute/signatures-psychiatry
523 collected 4 items
524
525 test_synthetic.py::test_pairwise_group_classification_synth[239673-expected_values0] PASSED [ 25%]
526 test_synthetic.py::test_pairwise_group_classification_synth[425769-expected_values1] PASSED [ 50%]
527 test_synthetic.py::test_pairwise_group_classification_synth[772192-expected_values2] PASSED [ 75%]
528 test_synthetic.py::test_pairwise_group_classification_synth_defaults PASSED [100%]
529
530 ===== 4 passed in 33.00 seconds =====
531 The command "pytest -v" exited with 0.
532
533
534
535 Done. Your build exited with 0.

```



build unknown

More options

Restart build

Remember Raw log

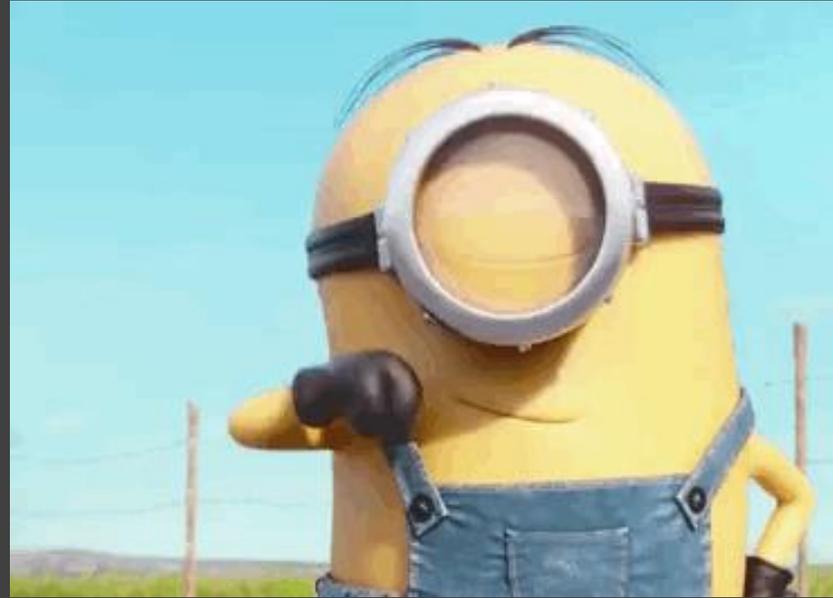
<https://github.com/alan-turing-institute/signatures-psychiatry>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

https://the-turing-way.netlify.com/continuous_integration/continuous_integration.html

- Run the analysis from start to finish as you work
- Many times tests will fail as expected: you're developing the analysis!
- Sometimes tests will fail unexpectedly
- CI makes you be explicit about what has changed



1. Introduction

2. Reproducibility

3. Open Research

4. Version Control

5. Collaborating on GitHub/GitLab

6. Research Data Management

7. Reproducible Environments

8. Testing

9. Reviewing

10. Continuous Integration

11. Reproducible Research with

Make

12. Risk Assessment



Welcome to the Turing Way

The Turing Way is a lightly opinionated guide to reproducible data science.

Our goal is to provide all the information that researchers need at the start of their projects to ensure that they are easy to reproduce at the end.

This also means making sure PhD students, postdocs, PIs and funding teams know which parts of the “responsibility of reproducibility” they can affect, and what they should do to nudge data science to being more efficient, effective and understandable.

A bit more background

Reproducible research is necessary to ensure that scientific work can be trusted. Funders and publishers are beginning to require that publications include access to the underlying data and the analysis code. The goal is to ensure that all results can be independently verified and built upon in future work. This is sometimes easier said than done. Sharing these research outputs means understanding data management, library sciences, software development, and continuous integration techniques: skills that are not widely taught or expected of academic researchers and data scientists.

The Turing Way is a handbook to support students, their supervisors, funders and journal editors

<https://the-turing-way.netlify.com/introduction/introduction>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Becky Arnold

“There are a lot of things you need to know before you can jump into continuous integration.

Version control is a prerequisite for pretty much everything.”



<https://software.ac.uk/about/fellows/becky-arnold>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

1. Introduction
2. Reproducibility
3. Open Research
4. Version Control
5. Collaborating on GitHub/GitLab
6. Research Data Management
7. Reproducible Environments
8. Testing
9. Reviewing
10. Continuous Integration
11. Reproducible Research with Make
12. Risk Assessment

Continuous integration

Prerequisite	Importance	Notes
Experience with the command line	Necessary	A tutorial on working via the command line can be found here
Version control	Necessary	See the chapter on this for more information
Testing	Very helpful	See the chapter on this for more information
Reproducible computational environments	Necessary	See the chapter on this for more information, particularly the sections on YAML files and containers

Table of contents

- [Summary](#)
- [How this will help you/ why this is useful](#)
 - [What are continuous delivery and continuous deployment?](#)
- [What is Travis and how does it work?](#)
- [Setting up continuous integration with Travis](#)
 - [Basic steps](#)

https://the-turing-way.netlify.com/continuous_integration/continuous_integration.html

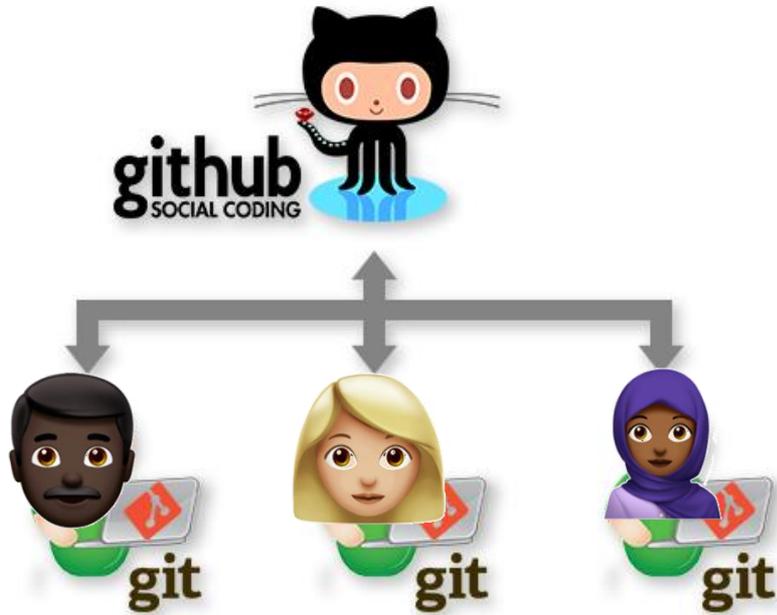
#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Version control



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

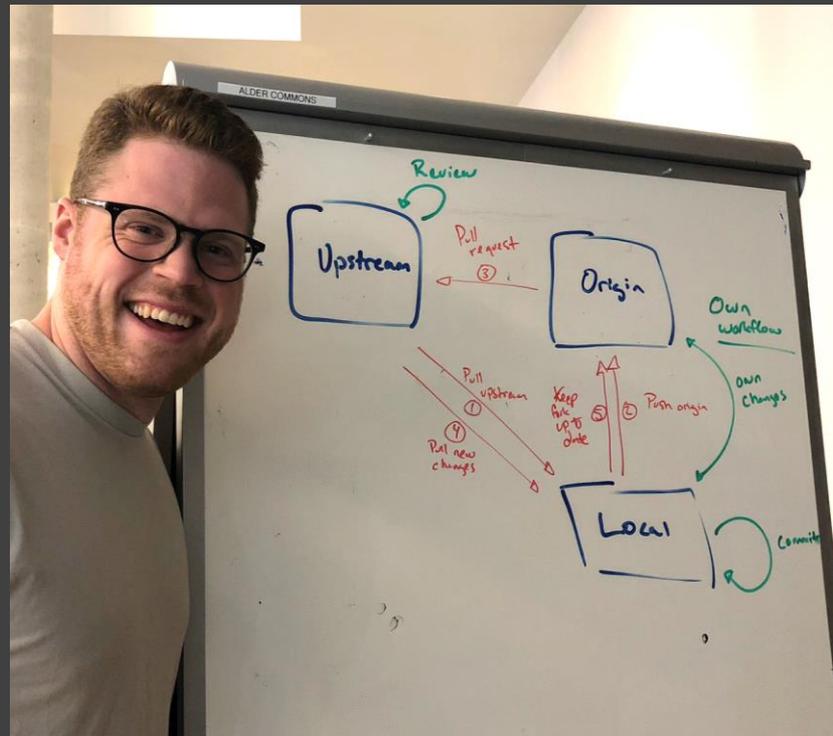
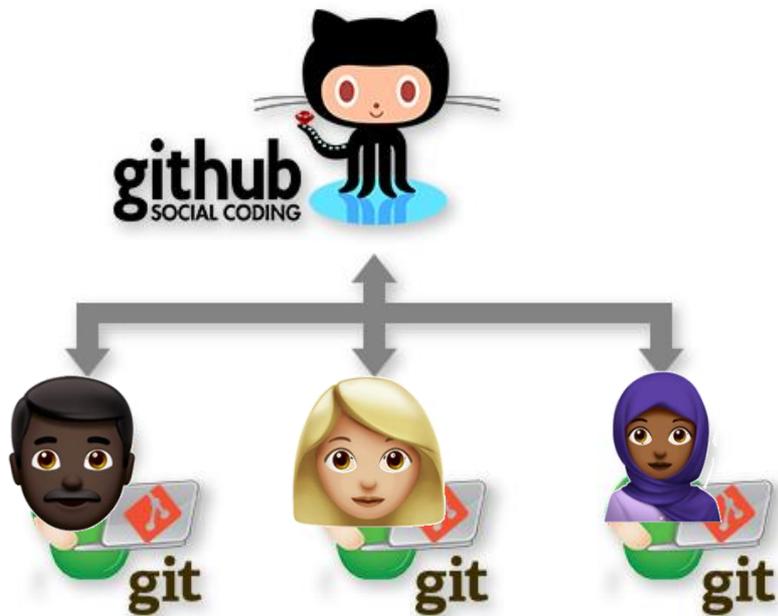


https://the-turing-way.netlify.com/collaborating_github/collaborating_github.html

https://the-turing-way.netlify.com/version_control/version_control.html

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>



https://the-turing-way.netlify.com/collaborating_github/collaborating_github.html

https://the-turing-way.netlify.com/version_control/version_control.html

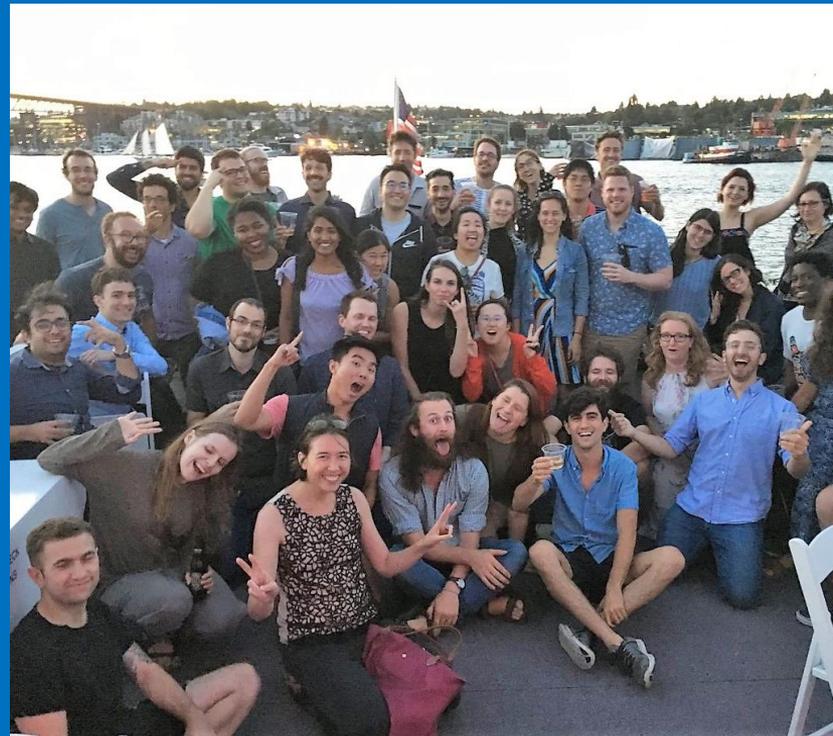
<https://neurohackademy.org>

#OHBM2019 #OpenForAll @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Neurohackademy

“Every hackathon should have a gong that you can ring when you complete your first pull request.”

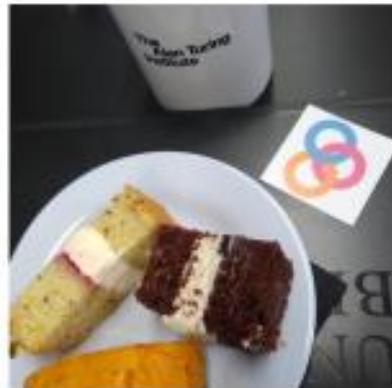


<https://neurohackademy.org>
[#HackSTIR](#) [#TuringWay](#) [@kirstie_j](#)
<https://doi.org/10.5281/zenodo.3515937>

Workshops & trainings



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>



<https://github.com/alan-turing-institute/the-turing-way/tree/master/workshops>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Rosie Higman

“There’s no point in running events when you’re only preaching to the choir. We need to show researchers the selfish reasons to follow our recommendations.”



<https://rosiehigman.wordpress.com>
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Book Dashes

- Manchester and London
- 13 selected people to contribute to the book
- 1:3 support ratio: mentored support to contribute expertise



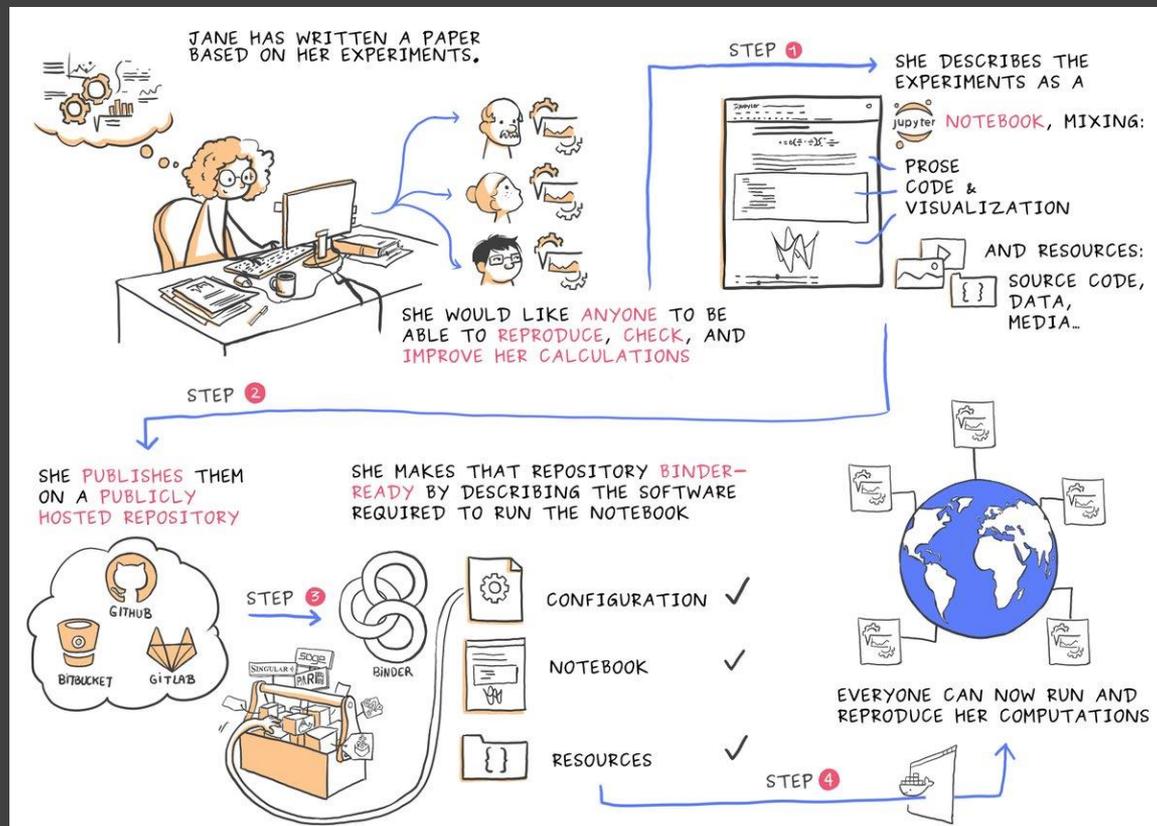
[https://github.com/
alan-turing-institute/the-turing-way/
blob/master/
workshops/book-dash/
book-dash-\[mcr|ldn\]-report.md](https://github.com/alan-turing-institute/the-turing-way/blob/master/workshops/book-dash/book-dash-[mcr|ldn]-report.md)

#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Turing Way & Binder



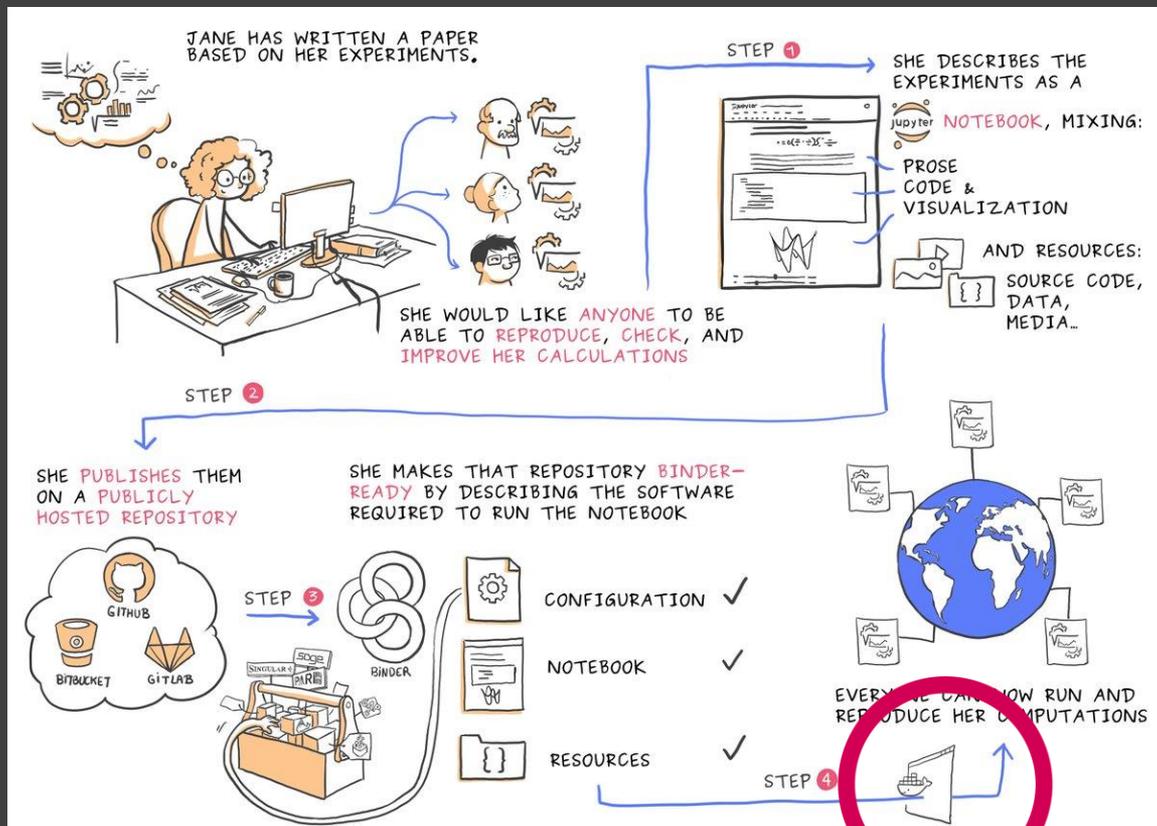
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>



Courtesy of Juliette Taka: <https://twitter.com/mybinderteam/status/1082556317842264064>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

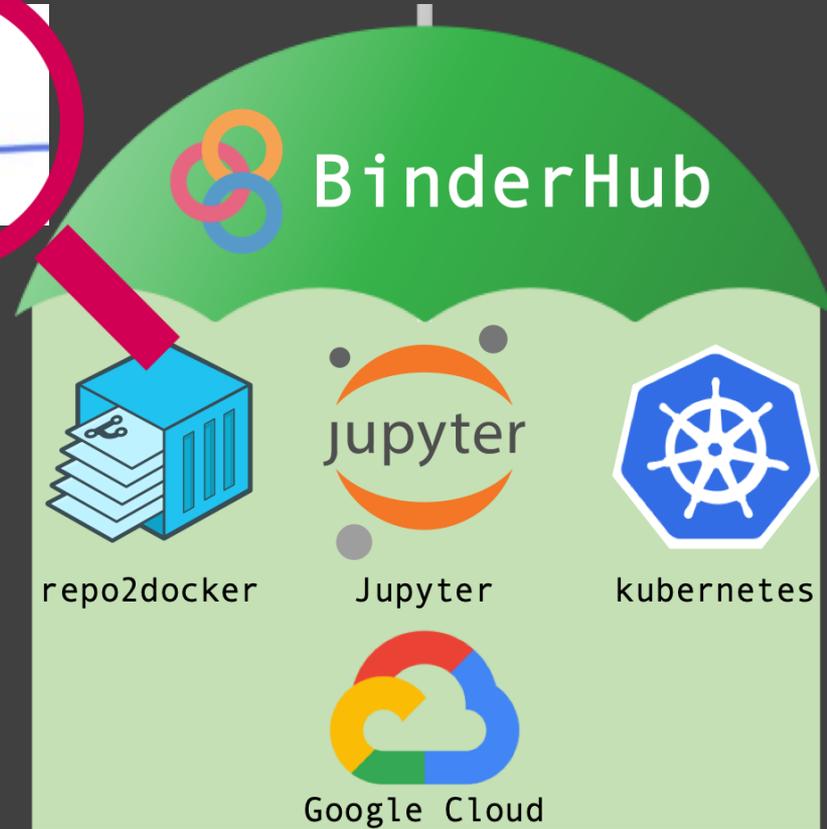
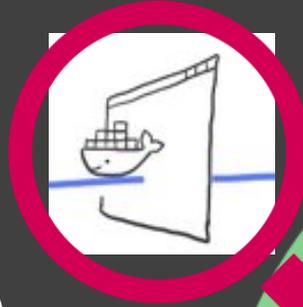


Courtesy of Juliette Taka: <https://twitter.com/mybinder-etc/status/1082556317842264064>

#Hack4IR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

- Coordinate cloud computing resources with Kubernetes (k8s)
- Make it easy for users to access with a JupyterHub
- Set up the environment from your GitHub repository



<https://binderhub.readthedocs.io>
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Sarah Gibson

“It took me a while to feel like I knew enough to contribute to Binder. But the team are always so excited to have my input. Its really motivating to be part of such a welcoming community.”



<https://www.turing.ac.uk/people/researchers/sarah-gibson>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

- Check analysis on my phone
- Share the responsibility with busy PIs
- Requires version control, capturing environment and new build for each change



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>



Table of Contents

Getting started with Binder

[Getting started with Binder](#)
[Common usage patterns in Binder](#)

How to...

[Choose languages for your environment](#)
[Configure the user interface](#)
[Generate custom launch badges for your Binder repository](#)
[Track repository data on mybinder.org](#)
[Share JupyterLab](#)

What is mybinder.org?

mybinder.org is a single deployment of a BinderHub instance, managed by the Binder community. It serves as both a public service and a demonstration of the BinderHub technology, though it is by no means the only BinderHub in existence. If you're interested in deploying your own BinderHub for your own uses, please see the [BinderHub documentation](#) and don't hesitate to reach out to the [Binder community](#).

For more information, check out [About mybinder.org](#).

Is mybinder.org free to use?

Yes! Though note that it has relatively [limited computational resources](#).

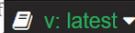
How much does running mybinder.org cost?

Great question! If you're interested in the technical costs of running **mybinder.org**, we publish a semi-up-to-date dataset of our costs at the [binder-data](#) repository. In addition, you can explore these costs with the binder link below!

 launch binder

How can mybinder.org be free to use?

On this page

What is a Binder?
What is the Binder community?
What is BinderHub?
What is **mybinder.org**?
Is **mybinder.org** free to use?
How much does running **mybinder.org** cost?
How can **mybinder.org** be free to use?
How much memory am I given when using Binder?
How long will my Binder session last?
Can I use mybinder.org for a live demo or workshop?
How does mybinder.org ensure user privacy?
How secure is mybinder.org?
Where can I report a security issue?
Can I push data from my Binder session back to my repository?
Can I put my configuration files outside the root of ?
What factors influence how long it takes a Binder session to start?
Will repos with fewer notebooks launch faster? Should I split my

<https://mybinder.readthedocs.io/en/latest/faq.html#how-much-does-running-mybinder-org-cost>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Thanks to [Google Cloud](#) and [OVH](#) for sponsoring our computers 🙌!



Starting repository: [jupyterhub/binder-billing/master](#)

New to Binder? Check out the [Binder Documentation](#) for more information

Build logs

[show](#)

Here's a non-interactive preview on [nbviewer](#) while we start a server for you. Your binder will open automatically when it is ready.

A screenshot of the nbviewer interface. On the left is the Jupyter logo and the text "jupyter nbviewer". In the center, the repository name "binder-billing" is displayed in a search bar. On the right, there are links for "JUPYTER", "FAQ", and icons for GitHub and Binder.

<https://mybinder.readthedocs.io/en/latest/faq.html#how-much-does-running-mybinder-org-cost>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

The image shows a Jupyter Notebook window titled "analyze_data.ipynb" in a "Launcher" environment. The notebook contains four code cells. The first cell imports pandas, glob, numpy, and altair, with a comment about the classic notebook interface and a width setting of 600. The second cell loads data from a GitHub URL. The third cell groups the data by project_id and date, calculates costs, and resamples to weekly costs. The fourth cell starts an altair chart.

```
[1]: import pandas as pd
from glob import glob
import numpy as np

import altair as alt
# If you're using the classic notebook interface, uncomment and run the below command
# alt.renderers.enable('notebook')

width = 600
```

Load data

```
[2]: url_data = "https://github.com/jupyterhub/binder-data/raw/master/billing/data/proc/data.json"
data = pd.read_json(url_data, orient='split')
data['date'] = pd.to_datetime(data['date'])
```

Cost by day

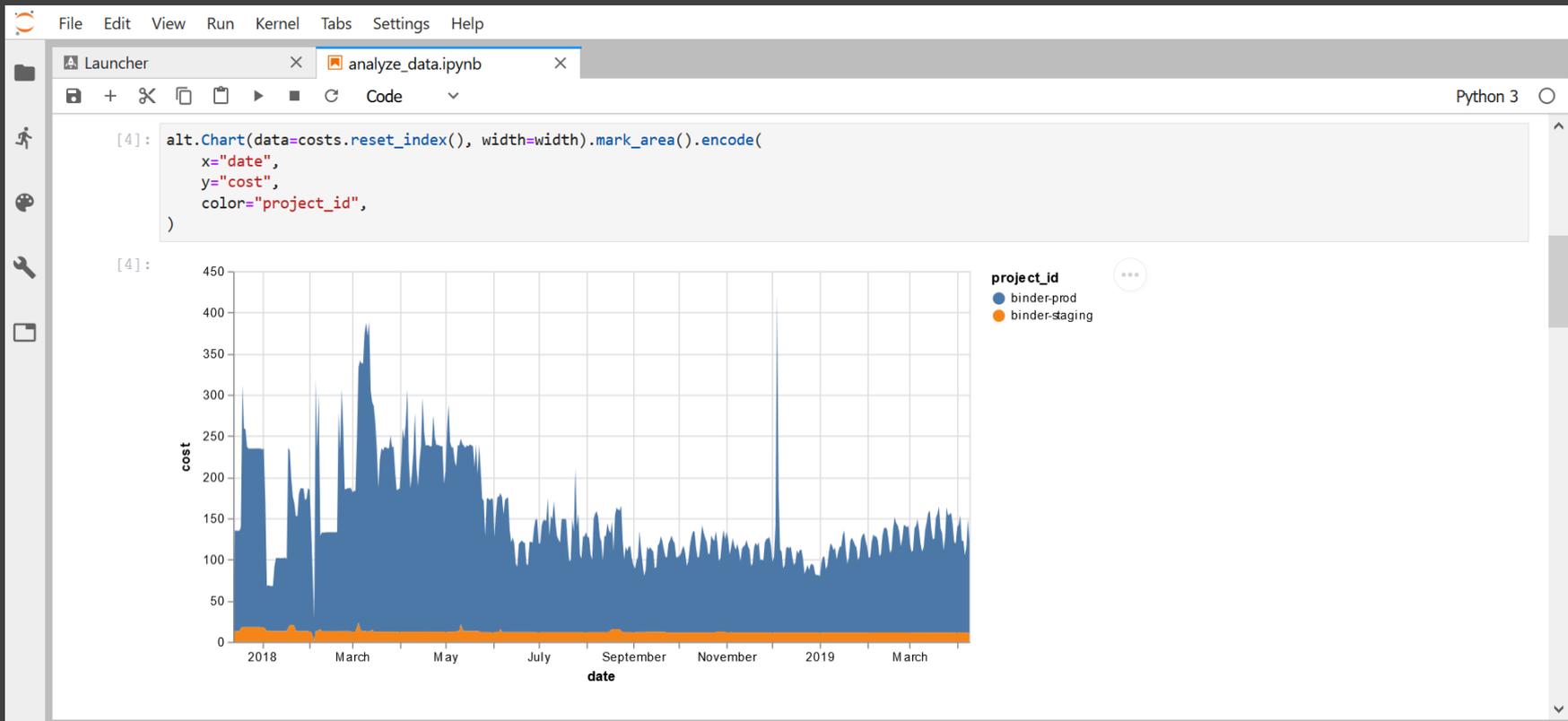
```
[3]: costs = data.groupby(['project_id', 'date']).sum()['cost']
costs = costs.reset_index()
weekly_costs = costs.groupby("project_id").resample("w", on="date").sum().reset_index()
```

```
[4]: alt.Chart(data=costs.reset_index(), width=width).mark_area().encode(
```

<https://mybinder.readthedocs.io/en/latest/faq.html#how-much-does-running-mybinder-org-cost>

#HackSTIR #TuringWay @kirstie_j

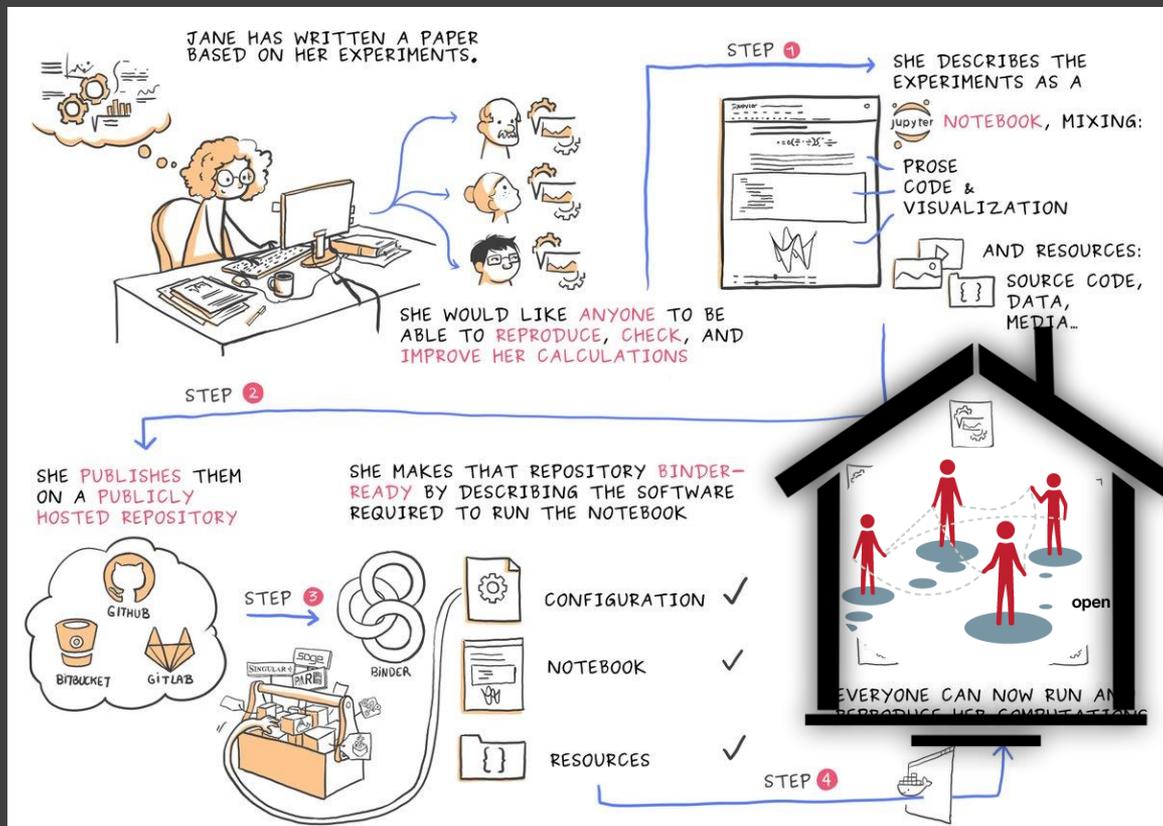
<https://doi.org/10.5281/zenodo.3515937>



<https://mybinder.readthedocs.io/en/latest/faq.html#how-much-does-running-mybinder-org-cost>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>



Courtesy of Juliette Taka: <https://twitter.com/mybinderteam/status/1082556317842264064>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

The Alan Turing Institute



Loading repository (can take 30s or more to load): sgibson91/branchLSTM/sgibson91python-runtime-patch

New to Binder? Check out the [Binder Documentation](#) for more information

Build logs

[show](#)

Here's a non-interactive preview on [nbviewer](#) while we start a server for you. Your binder will open automatically when it is ready.



JUPYTER

FAQ



branchLSTM

sgibson91python-runtime-patch

<https://github.com/kochkinaelena/branchLSTM> (on Turing Way Hub)

#HackSTIR #TuringWay @kirstie_j

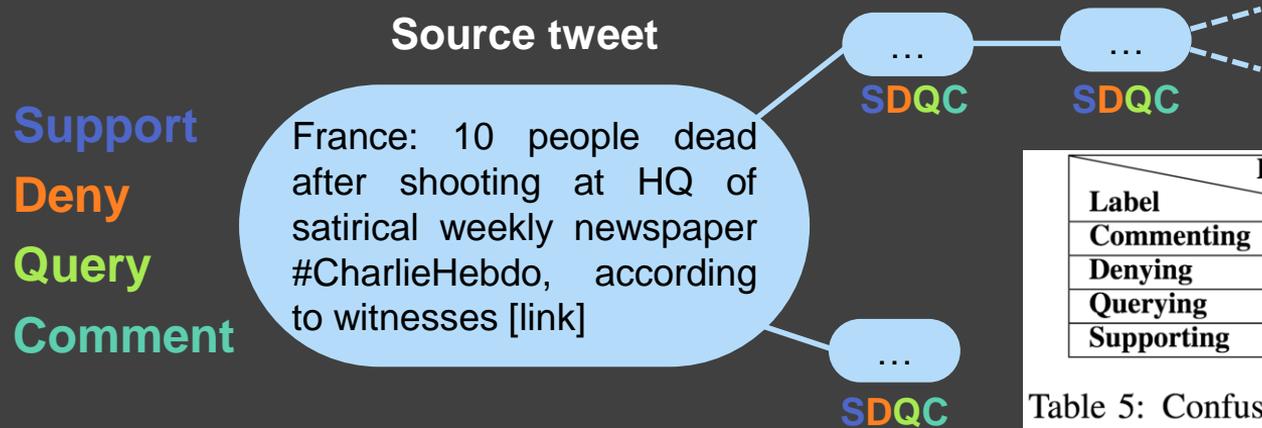
<https://doi.org/10.5281/zenodo.3515937>

Champion: Elena Kochkina



Turing at SemEval-2017 Task 8: Sequential Approach to Rumour Stance Classification with Branch-LSTM

Elena Kochkina, Maria Liakata, Isabelle Augenstein



	Prediction			
Label	C	D	Q	S
Commenting	760	0	12	6
Denying	68	0	1	2
Querying	69	0	36	1
Supporting	67	0	1	26

Table 5: Confusion matrix for testing set predictions

The screenshot shows a JupyterLab environment. On the left is a file browser with a table of files and folders. On the right is a console window titled 'Console 1' showing the output of running a Python script.

Name	Last Modified
dev_data	15 days ago
downloaded_data	15 days ago
output	15 days ago
scorer	15 days ago
src	15 days ago
tokenizers	in a few seconds
badwords.txt	15 days ago
bestparams_GN.txt	15 days ago
depth_analysis.py	15 days ago
environment.yml	15 days ago
LICENSE	15 days ago
outer.py	15 days ago
postBuild	15 days ago
predict.py	15 days ago
preprocessing.py	15 days ago
README.md	15 days ago
requirements.txt	15 days ago
subtaska.json	15 days ago
subtaskb.json	15 days ago
training.py	15 days ago

```
Python 2.7.15 | packaged by conda-forge | (default, Feb 28 2019, 04:00:11)
Type "copyright", "credits" or "license" for more information.

IPython 5.8.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

[*]: %run preprocessing.py

[nltk_data] Downloading package punkt to /home/jovyan...
[nltk_data] Unzipping tokenizers/punkt.zip.
Loading the model
```

<https://github.com/kochkinaelena/branchLSTM> (on Turing Way Hub)

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

The screenshot shows a JupyterLab environment. On the left is a file browser with a table of files and folders. On the right is a console window showing the output of a Python script execution.

Name	Last Modified
dev_data	15 days ago
downloaded_data	15 days ago
output	15 days ago
saved_data	seconds ago
scorer	15 days ago
src	15 days ago
tokenizers	2 minutes ago
badwords.txt	15 days ago
bestparams_GN.txt	15 days ago
depth_analysis.py	15 days ago
environment.yml	15 days ago
LICENSE	15 days ago
outer.py	15 days ago
postBuild	15 days ago
predict.py	15 days ago
preprocessing.py	15 days ago
README.md	15 days ago
requirements.txt	15 days ago
subtaska.json	15 days ago
subtaskb.json	15 days ago
training.py	15 days ago

```
Python 2.7.15 | packaged by conda-forge | (default, Feb 28 2019, 04:00:11)
Type "copyright", "credits" or "license" for more information.

IPython 5.8.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

[1]: %run preprocessing.py

[nltk_data] Downloading package punkt to /home/jovyan...
[nltk_data] Unzipping tokenizers/punkt.zip.
Loading the model
Done!
```

<https://github.com/kochkinaelena/branchLSTM> (on Turing Way Hub)

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

The screenshot shows a JupyterLab environment. On the left is a file browser with a table of files and their last modified dates. On the right is a console window showing the execution of two Python scripts: `preprocessing.py` and `outer.py`.

Name	Last Modified
dev_data	15 days ago
downloaded_data	15 days ago
output	15 days ago
saved_data	10 minutes ago
scorer	15 days ago
src	15 days ago
tokenizers	11 minutes ago
badwords.txt	15 days ago
bestparams_GN.txt	15 days ago
depth_analysis.py	15 days ago
environment.yml	15 days ago
LICENSE	15 days ago
outer.py	15 days ago
postBuild	15 days ago
predict.py	15 days ago
preprocessing.py	15 days ago
README.md	15 days ago
requirements.txt	15 days ago
subtaska.json	15 days ago
subtaskb.json	15 days ago
training.py	15 days ago

```
Python 2.7.15 | packaged by conda-forge | (default, Feb 28 2019, 04:00:11)
Type "copyright", "credits" or "license" for more information.

IPython 5.8.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

[1]: %run preprocessing.py

[nltk_data] Downloading package punkt to /home/jovyan...
[nltk_data] Unzipping tokenizers/punkt.zip.
Loading the model
Done!

[*]: %run outer.py

Loading best set of model parameters from output/bestparams_semeval2017.txt ...

{'learn_rate': 0.001, 'num_dense_layers': 2, 'num_lstm_units': 100, 'num_dense_units': 500, 'mb_size': 100, 'num_lstm_layers': 2, 'rng_seed': 364, 'num_epochs': 30, 'l2reg': 0.0}
Retrain model on train+dev set and evaluate on testing set
```

<https://github.com/kochkinaelena/branchLSTM> (on Turing Way Hub)

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

File Edit View Run Kernel Tabs Settings Help

Console 1

```
(Time: 154177550543516)
[3]: %run depth_analysis.py
      trials.txt is not available

      --- Table 4 ---
      Number of tweets per depth and performance at each of the depths
```

Depth	# tweets	# Support	# Deny	# Query	# Comment	Accuracy	MacroF	Support	Deny	Query
0	28	26	2	0	0	0.929	0.481	0.963	0.000	0.000
1	704	61	60	81	502	0.696	0.436	0.192	0.088	0.660
2	128	3	6	7	112	0.805	0.318	0.000	0.000	0.385
3	60	2	1	5	52	0.817	0.307	0.000	0.000	0.333
4	41	0	0	3	38	0.927	0.481	0.000	0.000	0.000
5	27	1	0	1	25	0.926	0.321	0.000	0.000	0.000
6+	61	1	2	9	49	0.803	0.223	0.000	0.000	0.000

```

      --- Table 5 ---
      Confusion matrix
```

Lab \ Pred	Comment	Deny	Query	Support
Comment	667	5	62	44
Deny	58	3	4	6
Query	30	0	72	4
Support	52	0	4	38

<https://github.com/kochkinaelena/branchLSTM> (on Turing Way Hub)

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Elena Kochkina

“How would I have known that it would be different on a different machine?! I only have access to the university HPC to run deep learning analyses.”



<https://warwick.ac.uk/fac/sci/dcs/people/research/mapmbc>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Gertjan van den Burg

“The fun part of data science is the modelling. Being able to read in information from a csv file should not be the hardest part.”



<https://gertjanvandenburgh.com>
[#HackSTIR](#) [#TuringWay](#) [@kirstie_j](#)
<https://doi.org/10.5281/zenodo.3515937>

alan-turing-institute / CleverCSVDemo

Unwatch 6 Star 0 Fork 1

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Insights

No description, website, or topics provided.

23 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find File Clone or download

GjvdBurg add more examples and clarify Latest commit 8304aaF a day ago

data	add more examples and clarify	a day ago
images	add qr code with link to repo	12 days ago
CSV_dialect_detection_with_CleverCSV.ipynb	add more examples and clarify	a day ago
CSV_dialect_detection_with_CleverCSV.md	add more examples and clarify	a day ago
LICENSE	Add makefile and create the notebook from Markdown	7 days ago
Makefile	Add makefile and create the notebook from Markdown	7 days ago
README.md	Add binder thingy to Readme	13 days ago
requirements.txt	add termcolor dependency	6 days ago

README.md

CleverCSV Demonstration

[launch binder](#)

This repository contains a demonstration of CleverCSV, a Python package for robust handling of non-standard (*messy*) CSV files. It is based on the work [Wrangling Messy CSV Files by Detecting Row and Type Patterns](#) by Gertjan van den Burg, Alfredo Nazabal, and Charles Sutton.

– <https://github.com/alan-turing-institute/CleverCSVDemo>

#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

alan-turing-institute / CleverCSVDemo

Unwatch 6 Star 0 Fork 1

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Insights

No description, website, or topics provided.

23 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find File Clone or download

GjvdBurg add more examples and clarify Latest commit 8304aaF a day ago

data	add more examples and clarify	a day ago
images	add qr code with link to repo	12 days ago
CSV_dialect_detection_with_CleverCSV.ipynb	add more examples and clarify	a day ago
CSV_dialect_detection_with_CleverCSV.md	add more examples and clarify	a day ago
LICENSE	Add makefile and create the notebook from Markdown	7 days ago
Makefile	Add makefile and create the notebook from Markdown	7 days ago
README.md	Add binder thingy to Readme	13 days ago
requirements.txt	add termcolor dependency	6 days ago

README.md

CleverCSV Demonstration

[launch binder](#)

This repository contains a demonstration of CleverCSV, a Python package for robust handling of non-standard (*messy*) CSV files. It is based on the work [Wrangling Messy CSV Files by Detecting Row and Type Patterns](#) by Gertjan van den Burg, Alfredo Nazabal, and Charles Sutton.

- <https://github.com/alan-turing-institute/CleverCSVDemo>
- “Wrangling Messy CSV Files by Detecting Row and Type Patterns”
arXiv:1811.11242

#HackSTIR #TuringWay @kirstie_
<https://doi.org/10.5281/zenodo.3515937>

CSV dialect detection with CleverCSV

Author: [Gertjan van den Burg](#)

In this note we'll show some examples of using CleverCSV, a package for handling messy CSV files. We'll start with a motivating example and then show some other files where CleverCSV shines. CleverCSV was developed as part of a research project on automating data wrangling. It achieves an accuracy of 97% on over 9300 real-world CSV files and improves the accuracy on messy files by 21% over standard tools.

Handy links:

- [Paper on arXiv](#)
- [CleverCSV on GitHub](#)
- [CleverCSV on PyPI](#)
- [Reproducible Research Repo](#)

IMDB Movie data

Alice is a data scientist who would like to analyse the movie ratings on IMDB for movies of different genres. She found [a dataset shared by a user on Kaggle](#) that contains information of over 14,000 movies. Great!

The data is stored in a CSV file, which is a very common data format for sharing tabular data. The first few lines of the file look like this:



```
In [1]: %xmode Minimal
import pandas as pd
df = pd.read_csv('./data/imdb.csv')
```

Exception reporting mode: Minimal

ParserError: Error tokenizing data. C error: Expected 44 fields in line 66, saw 46

Oh, that doesn't work. Maybe there's something wrong with the file? Let's try opening it with the Python CSV reader:

```
In [2]: import csv
with open('./data/imdb.csv', 'r', newline='') as fid:
    dialect = csv.Sniffer().sniff(fid.read())
    print("Detected delimiter = %r, quotechar = %r" % (dialect.delimiter, dialect.quotechar))
    fid.seek(0)
    reader = csv.reader(fid, dialect=dialect)
    rows = list(reader)

print("Loaded %i rows." % len(rows))
```

Detected delimiter = ' ', quotechar = ""

Loaded 13928 rows.

Huh, that's strange, Python thinks the *space* is the delimiter and loads 13928 rows, but the file should contain 14,762 rows according to the documentation. What's going on here?

Huh, that's strange, Python thinks the *space* is the delimiter and loads 13928 rows, but the file should contain 14,762 rows according to the documentation. What's going on here?

It turns out that on the 65th line of the file, there's a movie with the title `Dr. Seltsam\, oder wie ich lernte\, die Bombe zu lieben (1964)` (the German version of Dr. Strangelove). The title has commas in it, that are escaped using the `\` character! Why are CSV files so hard? 😞

CleverCSV to the rescue!

CleverCSV detects the dialect of CSV files much more accurately than existing approaches, and it is therefore robust against these kinds of format variations. It even has a wrapper that works with DataFrames!

```
In [3]: from ccsv.wrappers import csv2df

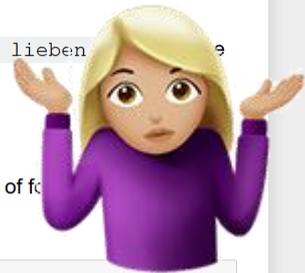
df = csv2df('./data/imdb.csv')
df
```

Out [3]:

	fn	tid	title	wordsInTitle	url	imdbRating	ratingCount	duration	year	type	...	News
0	titles01/tt0012349	tt0012349	Der Vagabund und das Kind (1921)	der vagabund und das kind	http://www.imdb.com/title/tt0012349/	8.4	40550.0	3240.0	1921.0	video.movie	...	0
1	titles01/tt0015864	tt0015864	Goldrausch (1925)	goldrausch	http://www.imdb.com/title/tt0015864/	8.3	45319.0	5700.0	1925.0	video.movie	...	0
2	titles01/tt0017136	tt0017136	Metropolis (1927)	metropolis	http://www.imdb.com/title/tt0017136/	8.4	81007.0	9180.0	1927.0	video.movie	...	0
3	titles01/tt0017925	tt0017925	Der General (1926)	der general	http://www.imdb.com/title/tt0017925/	8.3	37521.0	6420.0	1926.0	video.movie	...	0
			Lichter der	lichter der gro	http://www.imdb.com							

Huh, that's strange, Python thinks the *space* is the delimiter and loads 13928 rows, but the file should contain 14,762 rows according to the documentation. What's going on here?

It turns out that on the 65th line of the file, there's a movie with the title `Dr. Seltsam\, oder wie ich lernte\, die Bombe zu lieben` (German version of Dr. Strangelove). The title has commas in it, that are escaped using the `\` character! Why are CSV files so hard? 😞



CleverCSV to the rescue!

CleverCSV detects the dialect of CSV files much more accurately than existing approaches, and it is therefore robust against these kinds of fo even has a wrapper that works with DataFrames!

```
In [3]: from ccsv.wrappers import csv2df

df = csv2df('./data/imdb.csv')
df
```

Out [3]:

	fn	tid	title	wordsInTitle	url	imdbRating	ratingCount	duration	year	type	...	News
0	titles01/tt0012349	tt0012349	Der Vagabund und das Kind (1921)	der vagabund und das kind	http://www.imdb.com/title/tt0012349/	8.4	40550.0	3240.0	1921.0	video.movie	...	0
1	titles01/tt0015864	tt0015864	Goldrausch (1925)	goldrausch	http://www.imdb.com/title/tt0015864/	8.3	45319.0	5700.0	1925.0	video.movie	...	0
2	titles01/tt0017136	tt0017136	Metropolis (1927)	metropolis	http://www.imdb.com/title/tt0017136/	8.4	81007.0	9180.0	1927.0	video.movie	...	0
3	titles01/tt0017925	tt0017925	Der General (1926)	der general	http://www.imdb.com/title/tt0017925/	8.3	37521.0	6420.0	1926.0	video.movie	...	0
			Lichter der	lichter der gro	http://www.imdb.com							

df

			Episode 2005)	episode	http://www.imdb.com/title/tt0675644/									
14757	titles04/index.html.9992	tt0675644	"Playhouse 90" The Miracle Worker (TV Episode ...	playhouse the miracle worker tv episode	http://www.imdb.com/title/tt0675644/	7.3	8.0	5400.0	1957.0	video.episode	...	0		
14758	titles04/index.html.9994	tt0679222	"Private Screenings" Robert Mitchum and Jane R...	private screenings robert mitchum and jane rus...	http://www.imdb.com/title/tt0679222/	7.0	20.0	3600.0	1996.0	video.episode	...	0		
14759	titles04/index.html.9995	tt0680064	"Providence" All the King's Men (TV Episode 2002)	providence all the king s men tv episode	http://www.imdb.com/title/tt0680064/	NaN	NaN	3600.0	2002.0	video.episode	...	0		
14760	titles04/index.html.9997	tt0681024	"QI" Adam (TV Episode 2003)	qi adam tv episode	http://www.imdb.com/title/tt0681024/	7.6	89.0	1800.0	2003.0	video.episode	...	0		

14761 rows × 44 columns

Hooray! 🎉

How does it work? CleverCSV searches the space of all possible dialects of a file, and computes a *data consistency measure* that quantifies how much the resulting table "looks like real data". The consistency measure combines patterns of row lengths in the parsing result and the data type of the resulting cells. This mimicks how a human would identify the dialect. If you're wondering why this problem is hard, it's because every dialect will give you *some* table, but not necessarily the correct one. More details can be found [in the paper](#).

Gertjan van den Burg

“The fun part of data science is the modelling. Being able to read in information from a csv file should not be the hardest part.

There is no AI. I am the AI.”



<https://gertjanvandenburgh.com>
[#HackSTIR](#) [#TuringWay](#) [@kirstie_j](#)
<https://doi.org/10.5281/zenodo.3515937>

CSV dialect detection with CleverCSV

Author: [Gertjan van den Burg](#)

In this note we'll show some examples of using CleverCSV, a package for handling messy CSV files. We'll start with a motivating example and then show some other files where CleverCSV shines. CleverCSV was developed as part of a research project on automating data wrangling. It achieves an accuracy of 97% on over 9300 real-world CSV files and improves the accuracy on messy files by 21% over standard tools.

Handy links:

- [Paper on arXiv](#)
- [CleverCSV on GitHub](#)
- [CleverCSV on PyPI](#)
- [Reproducible Research Repo](#)



IMDB Movie data

Alice is a data scientist who would like to analyse the movie ratings on IMDB for movies of different genres. She found [a dataset shared by a user on Kaggle](#) that contains information of over 14,000 movies. Great!

The data is stored in a CSV file, which is a very common data format for sharing tabular data. The first few lines of the file look like this:

<https://github.com/alan-turing-institute/CleverCSVDemo>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Repository for reproducibility of the CSV file project

[reproducible-research](#) [reproducible-paper](#) [reproducibility](#) [reproducible-science](#) [csv-files](#) [csv](#) [csv-parsing](#)

27 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find File Clone or download

GjjvdBurg Simplify makefile Latest commit 348911c on 29 Nov 2018

data	add data dir placeholder	5 months ago
design	Fix indent	5 months ago
results/test	Replace absolute path by relative path	5 months ago
scripts	Make normal form output the same as the other detectors	5 months ago
.gitmodules	initial commit	5 months ago
LICENSE	Add the license	5 months ago
Makefile	Simplify makefile	5 months ago
README.md	Simplify makefile	5 months ago
requirements.txt	Add missing package	5 months ago
urls_github.json	Update GitHub data urls to direct links	5 months ago
urls_ukdata.json	initial commit	5 months ago

README.md

CSV Wrangling

This is the repository for reproducing the experiments in the paper:

[Wrangling Messy CSV files by Detecting Row and Type Patterns](#)

by G.J.J. van den Burg, A. Nazabal and C. Sutton.

– https://github.com/alan-turing-institute/CSV_Wrangling

– “Wrangling Messy CSV Files by Detecting Row and Type Patterns”

arXiv:1811.11242

#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Repository for reproducibility of the CSV file project

[reproducible-research](#) [reproducible-paper](#) [reproducibility](#) [reproducible-science](#) [csv-files](#) [csv](#) [csv-parsing](#)

27 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request

Create new file Upload files Find File Clone or download

GjvdBurg Simplify makefile Latest commit 348911c on 29 Nov 2018

data	add data dir placeholder	5 months ago
design	Fix indent	5 months ago
results/test	Replace absolute path by relative path	5 months ago
scripts	Make normal form output the same as the other detectors	5 months ago
.gitmodules	initial commit	5 months ago
LICENSE	Add the license	5 months ago
Makefile	Simplify makefile	5 months ago
README.md	Simplify makefile	5 months ago
requirements.txt	Add missing package	5 months ago
urls_github.json	Update GitHub data urls to direct links	5 months ago
urls_ukdata.json	initial commit	5 months ago

README.md

CSV Wrangling

This is the repository for reproducing the experiments in the paper:

[Wrangling Messy CSV files by Detecting Row and Type Patterns](#)

by G.J.J. van den Burg, A. Nazabal and C. Sutton.

– https://github.com/alan-turing-institute/CSV_Wrangling

– “Wrangling Messy CSV Files by Detecting Row and Type Patterns”

arXiv:1811.11242

#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

The Turing Way

1. Introduction
2. Reproducibility
3. Open Research
4. Version Control
5. Reproducible Environments
6. Testing
7. Reviewing
8. Continuous Integration
9. Research Data Management
10. Reproducible Research with Make

What is Make

Make is a build automation tool. It uses a configuration file called a Makefile that contains the *rules* for what to build. Make builds *targets* using *recipes*. Targets can optionally have *prerequisites*. Prerequisites can be files on your computer or other targets. Make determines what to build based on the dependency tree of the targets and prerequisites (technically, this is a [directed acyclic graph](#)). It uses the *modification time* of prerequisites to update targets only when needed.

Why use Make for Reproducible Research?

There are several reasons why Make is a good tool to use for reproducible research:

1. Make is available on many platforms
2. Make is easy to learn
3. Makefiles are text files, which makes them easy share and keep in version control.
4. Many people are already familiar with Make
5. Using Make doesn't exclude using other tools such as Travis, Docker, etc.

Learn Make by Example

One of the things that might scare people off from using Make is that existing Makefiles can seem daunting and it may seem difficult to tailor to your own needs. In this hands-on tutorial we will intentionally construct a Makefile for a real data analysis project. The idea is to explain different

<https://the-turing-way.netlify.com/make/make.html>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Case studies

- Show that it can be done
- Provide templates and starting points
- Inspire



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

A global collaboration



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Patricia Herterich

“What really sets The Turing Way apart is HOW we’re writing the book. The focus on community, the commitment to transparency and working open right from the beginning is an exciting (and terrifying) new way of working.”



<https://rd-alliance.org/users/patricia-herterich>
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Open Leadership Principles



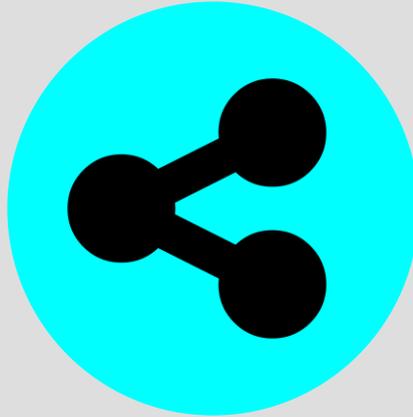
Understanding

You make the work accessible and clear

Read more

<https://mozilla.github.io/olm-whitepaper>

moz://a



Sharing

You make the work easy to adapt, reproduce, and spread



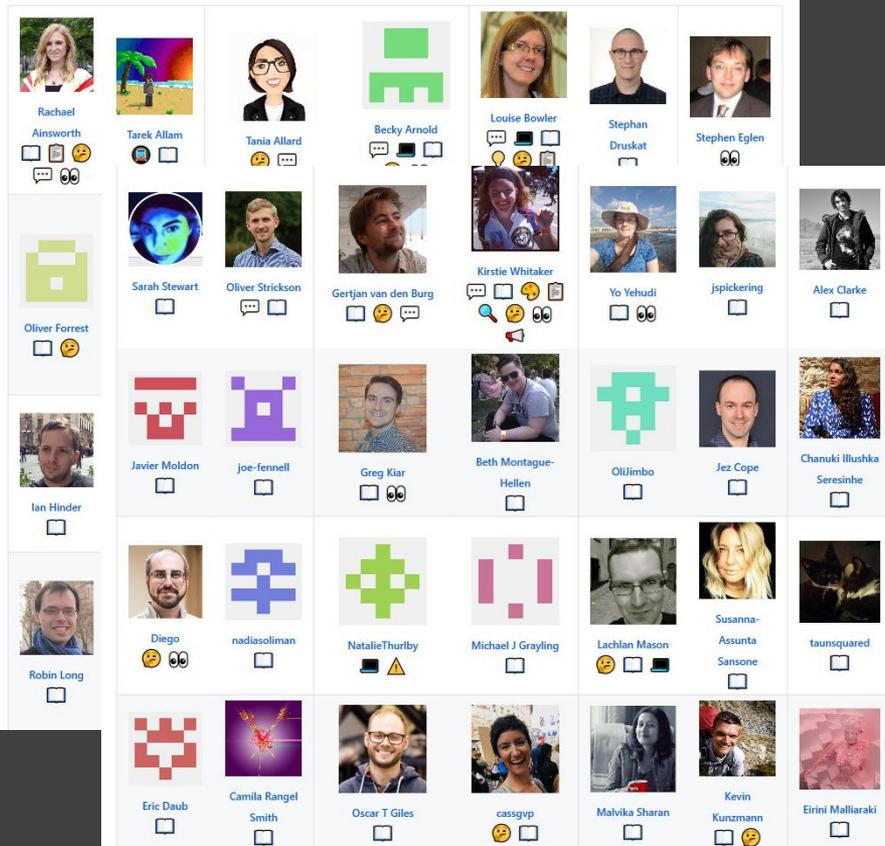
Participation & Inclusion

You build shared ownership and agency to make the work inviting and sustainable for all.

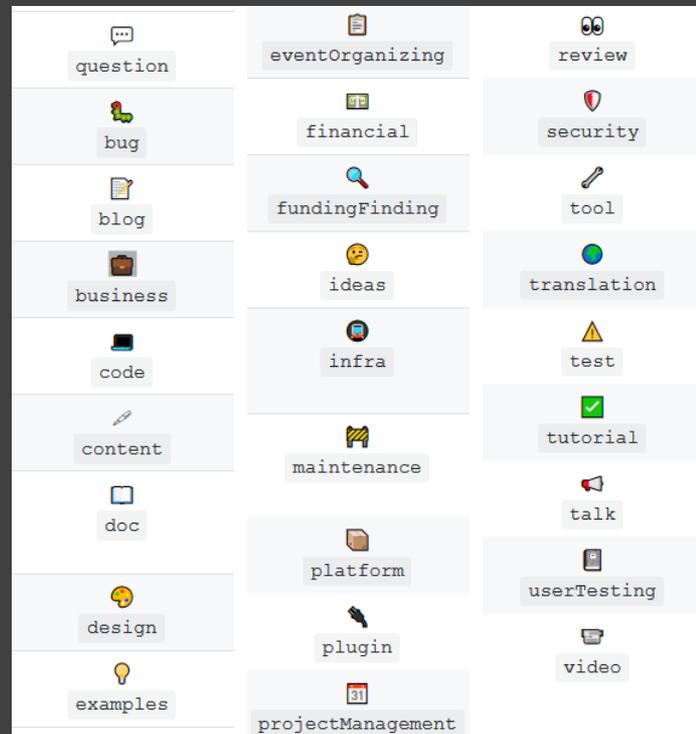
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Contributors

Thanks goes to these wonderful people (emoji key):



This project follows the [all-contributors](#) specification. Contributions of any kind welcome!



<https://github.com/alan-turing-institute/the-turing-way>

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

<https://github.com/alan-turing-institute/the-turing-way#contributors>

<https://allcontributors.org/docs/en/emoji-key>

Rachael Ainsworth

“Personas and pathways exercises let me reflect on what people are finding difficult about contributing to The Turing Way. The project can only reach its potential if it is easy for a diverse constellation of contributions.”



<https://ainsworth.github.io>
#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>



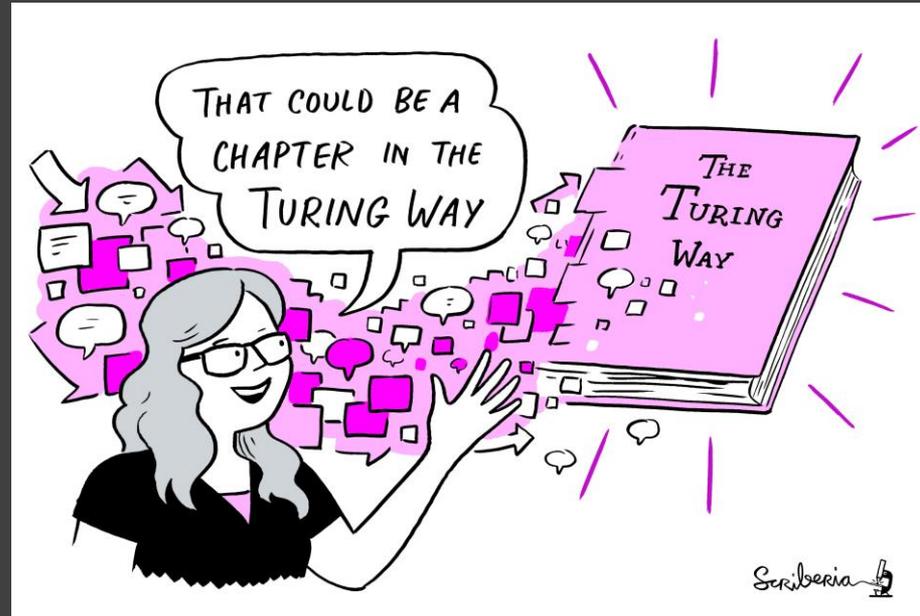
The future



#HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>

Funding extension

- Expand scope to all data science practices
 - Reproducibility
 - Scoping and designing a data science project
 - Ethics
 - Communication and visualisation
 - Collaborative working



[https://github.com/
alan-turing-institute/the-turing-way/
blob/master/project_management/
tps-funding-application-20190429.md](https://github.com/alan-turing-institute/the-turing-way/blob/master/project_management/tps-funding-application-20190429.md)

#HackSTIR #TuringWay @kirstie_j

<https://doi.org/10.5281/zenodo.3515937>

Metrics for success

- 20 new chapters
- 100 authors
- 200 contributors
- 1000 mailing list subscribers
- 50 first pull requests
- 20 new contributors to other open source projects

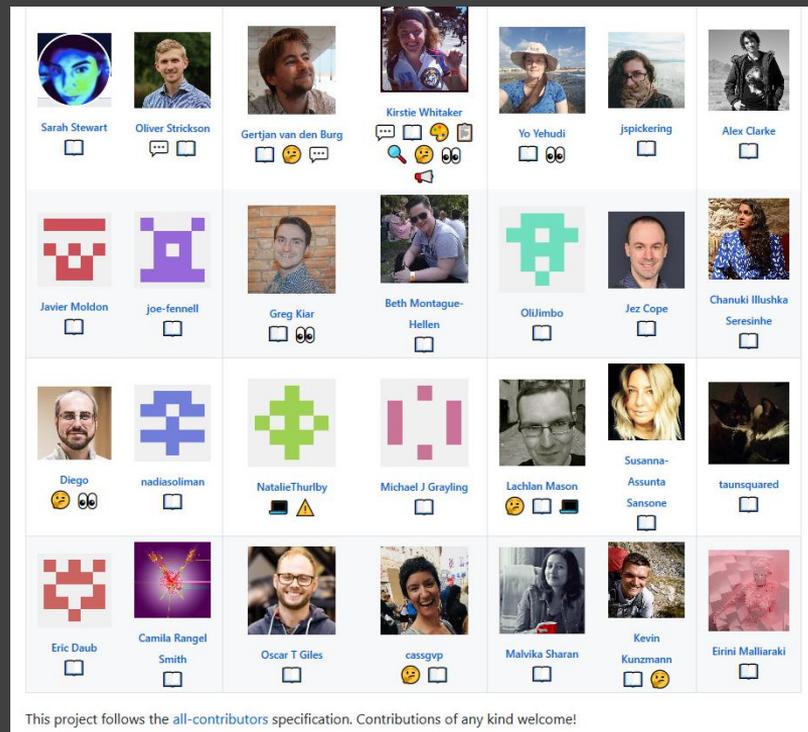
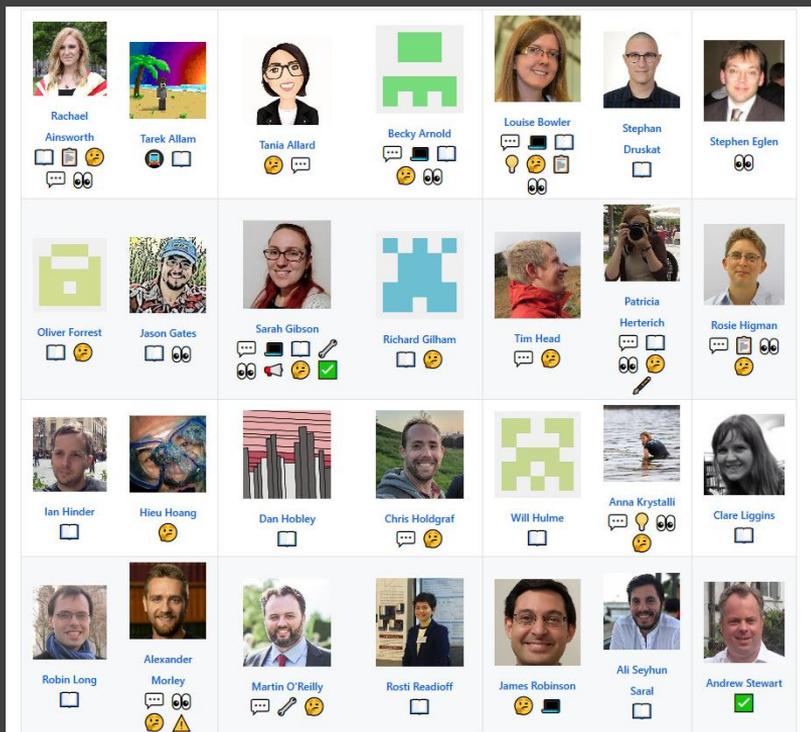


Collaboration cafes

- 1st and 3rd Wednesdays of each month
- All remote participation
 - Zoom call
 - Pomodoro technique
 - Breakout rooms for mentored contributions
- Everyone welcome



Thank you to current (& future) contributors



This project follows the [all-contributors](#) specification. Contributions of any kind welcome!

<https://github.com/alan-turing-institute/the-turing-way#contributors>
<https://allcontributors.org/docs/en/emoji-key>

#HackSTIR #TuringWay @kirstie_
<https://doi.org/10.5281/zenodo.3515937>

Thank you

The
Alan Turing
Institute



- Book: <https://the-turing-way.netlify.com>
- Newsletter: <https://tinyletter.com/TuringWay>
- GitHub: <https://github.com/alan-turing-institute/the-turing-way>
- Chat: <https://gitter.im/alan-turing-institute/the-turing-way>
- Next Collaboration Café: 30 October at 8am UK time
- This work was supported by The UKRI Strategic Priorities Fund under the EPSRC Grant EP/T001569/1, particularly the "Tools, Practices and Systems" theme within that grant, and by The Alan Turing Institute under the EPSRC grant EP/N510129/1.
- Unsplash photos by Adolfo Felix, James Pond, Jose Alejandro Cuffia, Kinson Leung, Mateo Vrbnjak, Mimi thian, Omar Albeik, Perry Grone, Toa Heftiba, Tomasz Frankows, Wilmer Martinez. Noun Project icons by Aybige, Luis Prado, Edward Boatman, Becris, Rose Alice Design, Hyemm.work.
- Original artwork by Scriberia: <https://doi.org/10.5281/zenodo.3332807> #HackSTIR #TuringWay @kirstie_j
<https://doi.org/10.5281/zenodo.3515937>