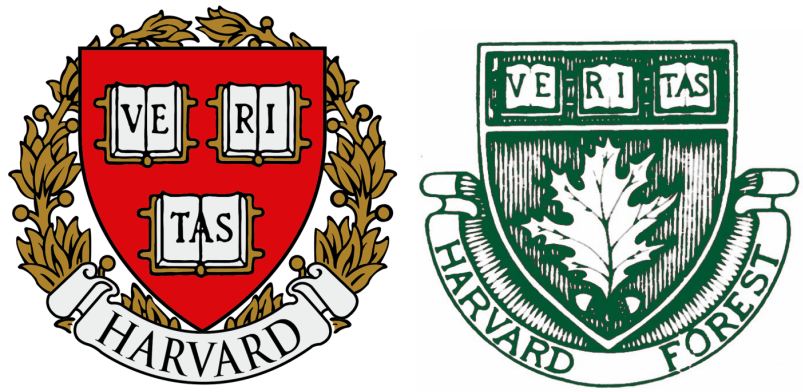
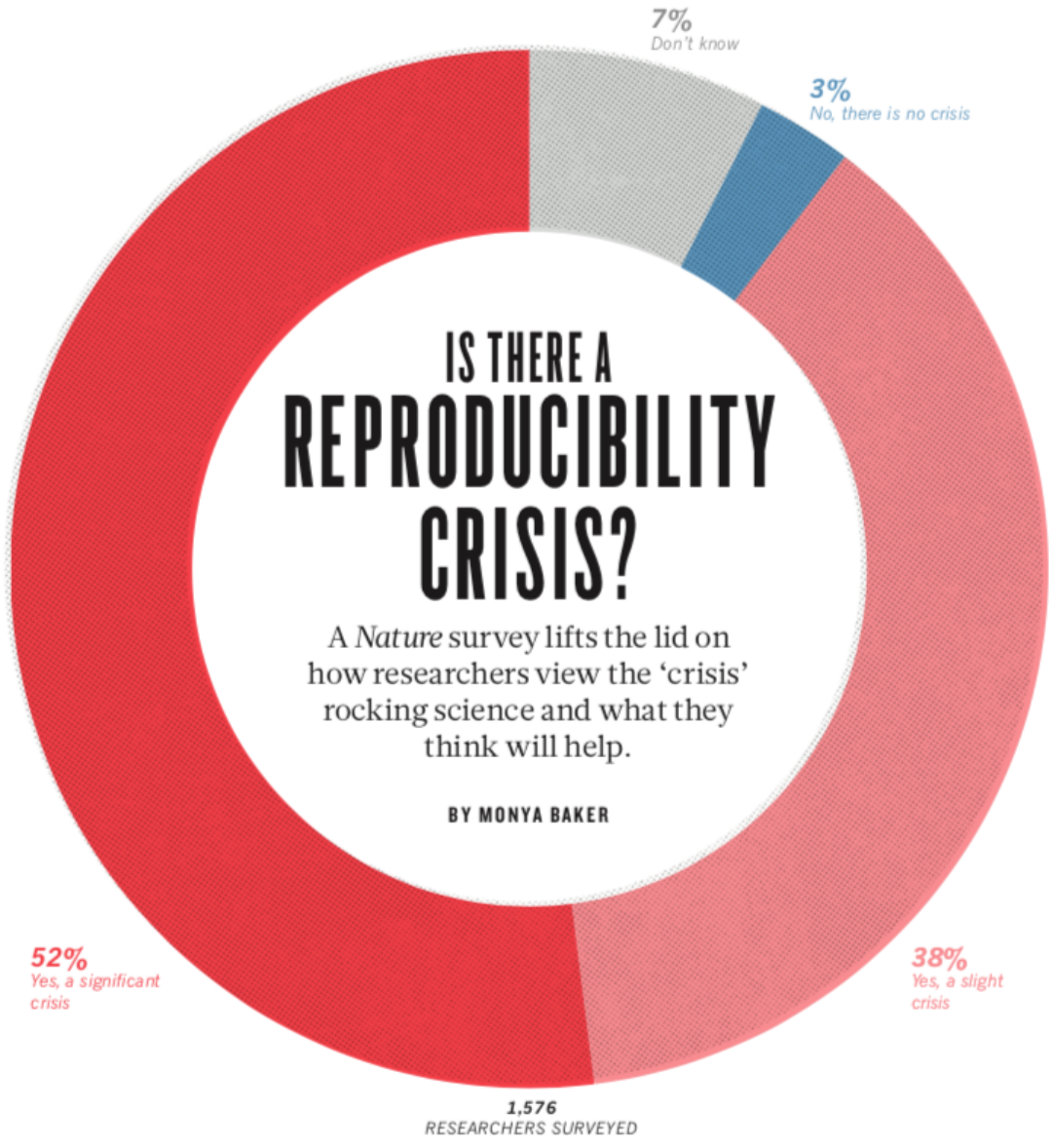
Opportunity in the Reproducibility Crisis

Computational tools to improve scientific benefaction

Matthew K. Lau, PhD



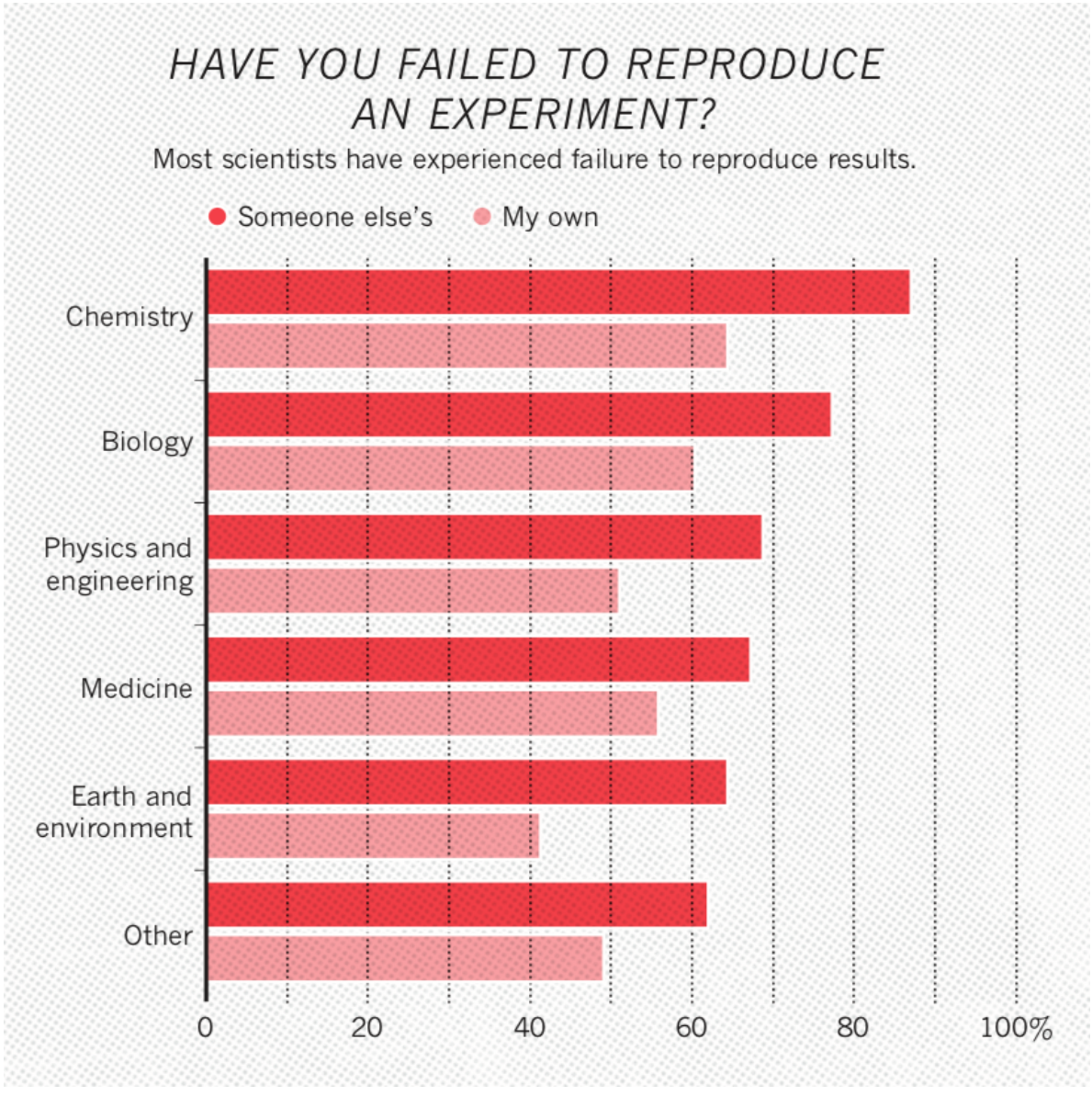
START



IS THERE A REPRODUCIBILITY CRISIS?

A *Nature* survey lifts the lid on how researchers view the 'crisis' rocking science and what they think will help.

BY MONYA BAKER


HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?
Most scientists have experienced failure to reproduce results.
Someone else's
My own
Chemistry
Biology
Physics and engineering
Medicine
Earth and environment
Other
0
20
40
60
80
100%

So, how bad is it, really?

Harvard Calls for Retraction of Dozens of Studies by Noted Cardiac Researcher

Some 31 studies by Dr. Piero Anversa contain fabricated or falsified data, officials concluded. Dr. Anversa popularized the idea of stem cell treatment for damaged hearts.

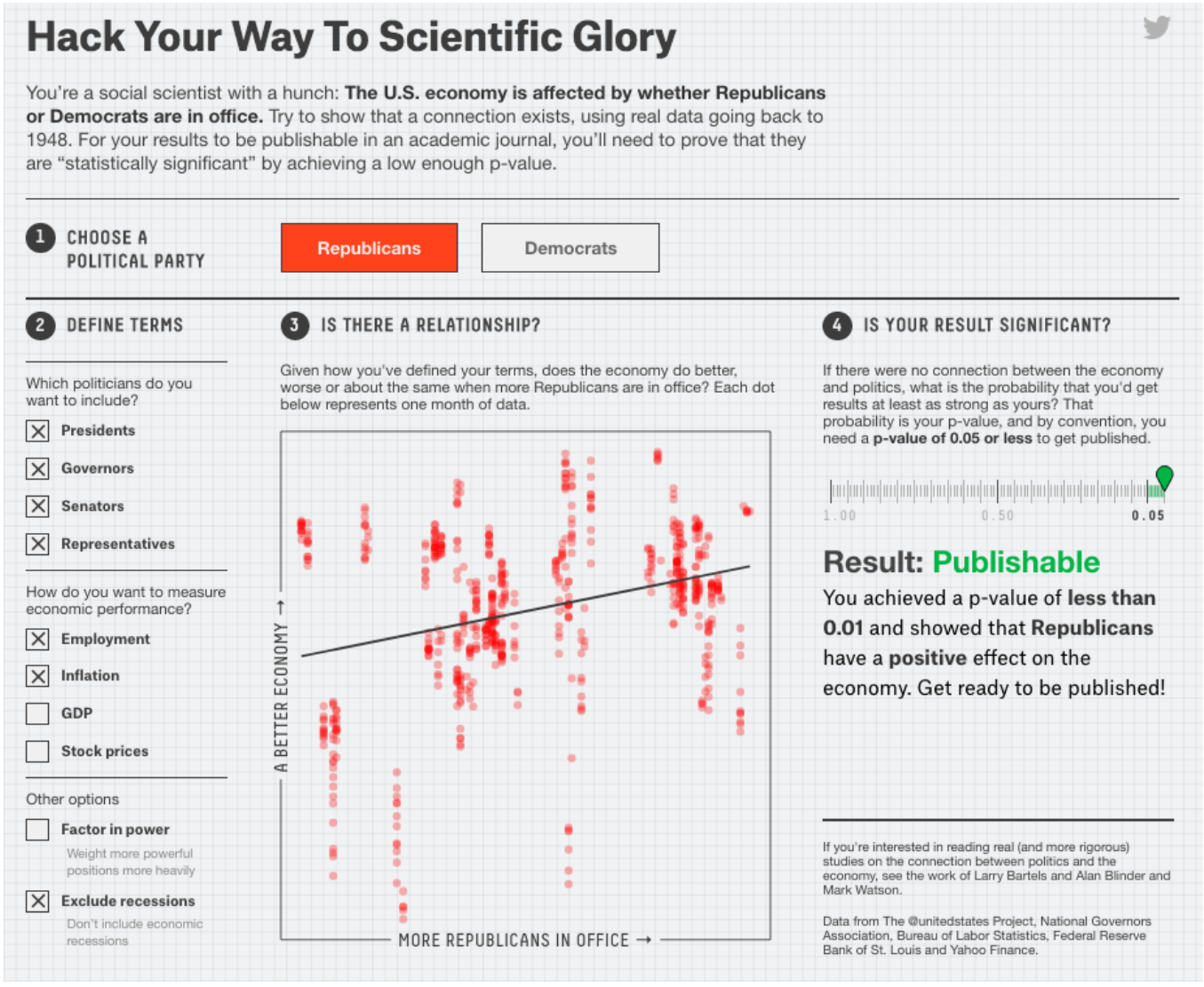
What are the factors contributing to the crisis?

Statistic's Contribution

$p = 0.05 = 1/20$

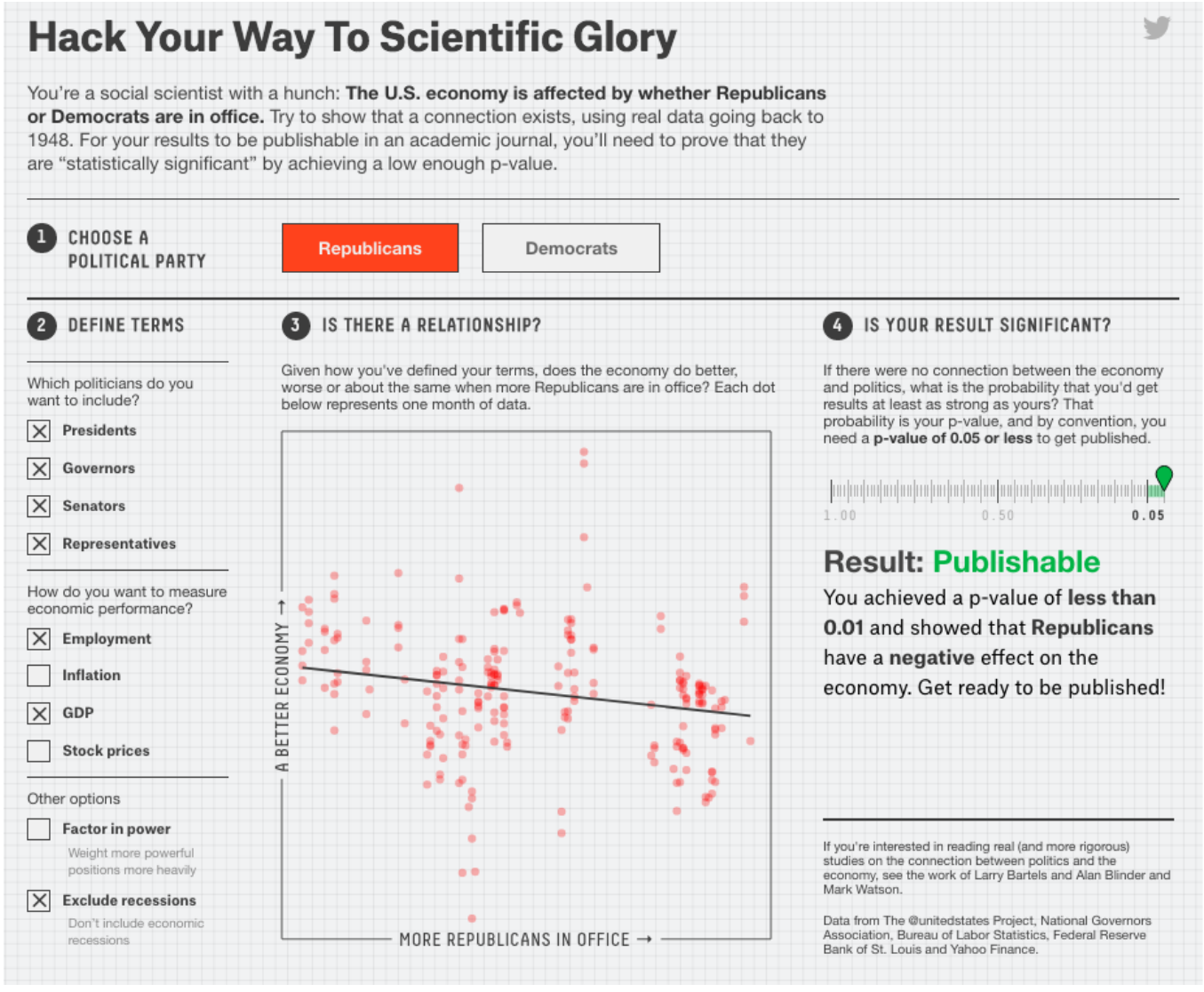
Statistic's Contribution

One in twenty



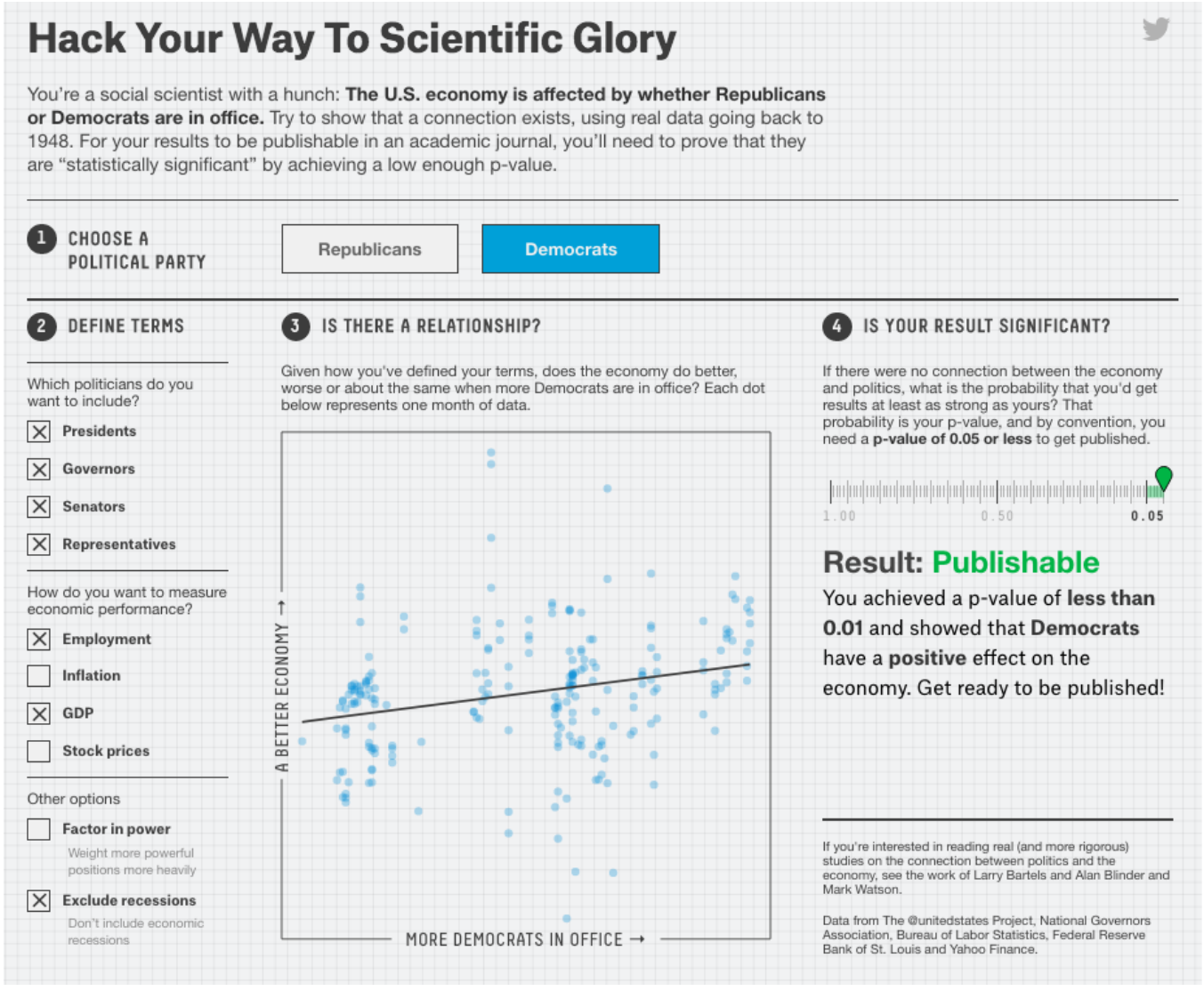
Aschwanden and King 2015

fivethirtyeight.com/features/science-isnt-broken



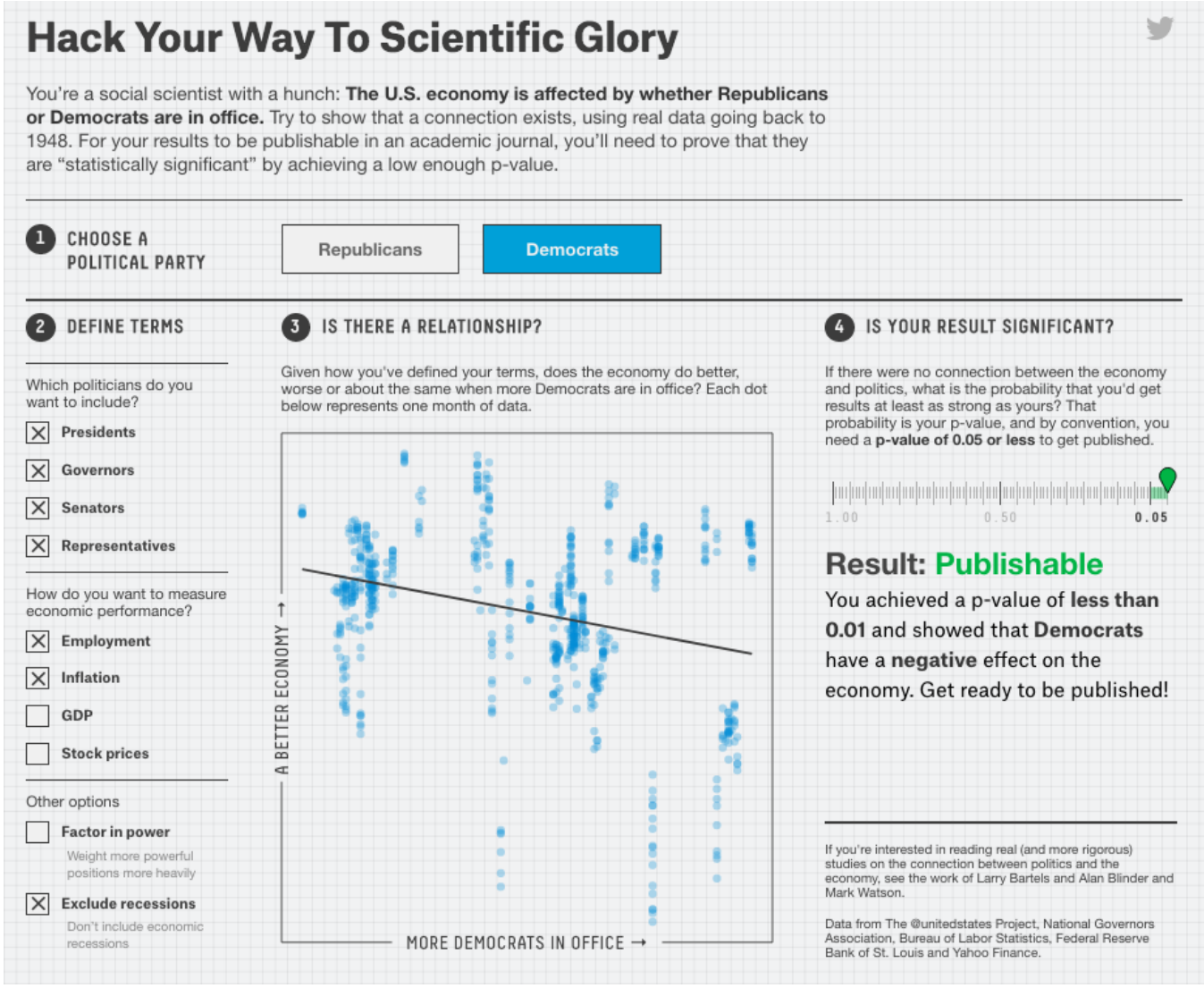
Aschwanden and King 2015

fivethirtyeight.com/features/science-isnt-broken



Aschwanden and King 2015

fivethirtyeight.com/features/science-isnt-broken

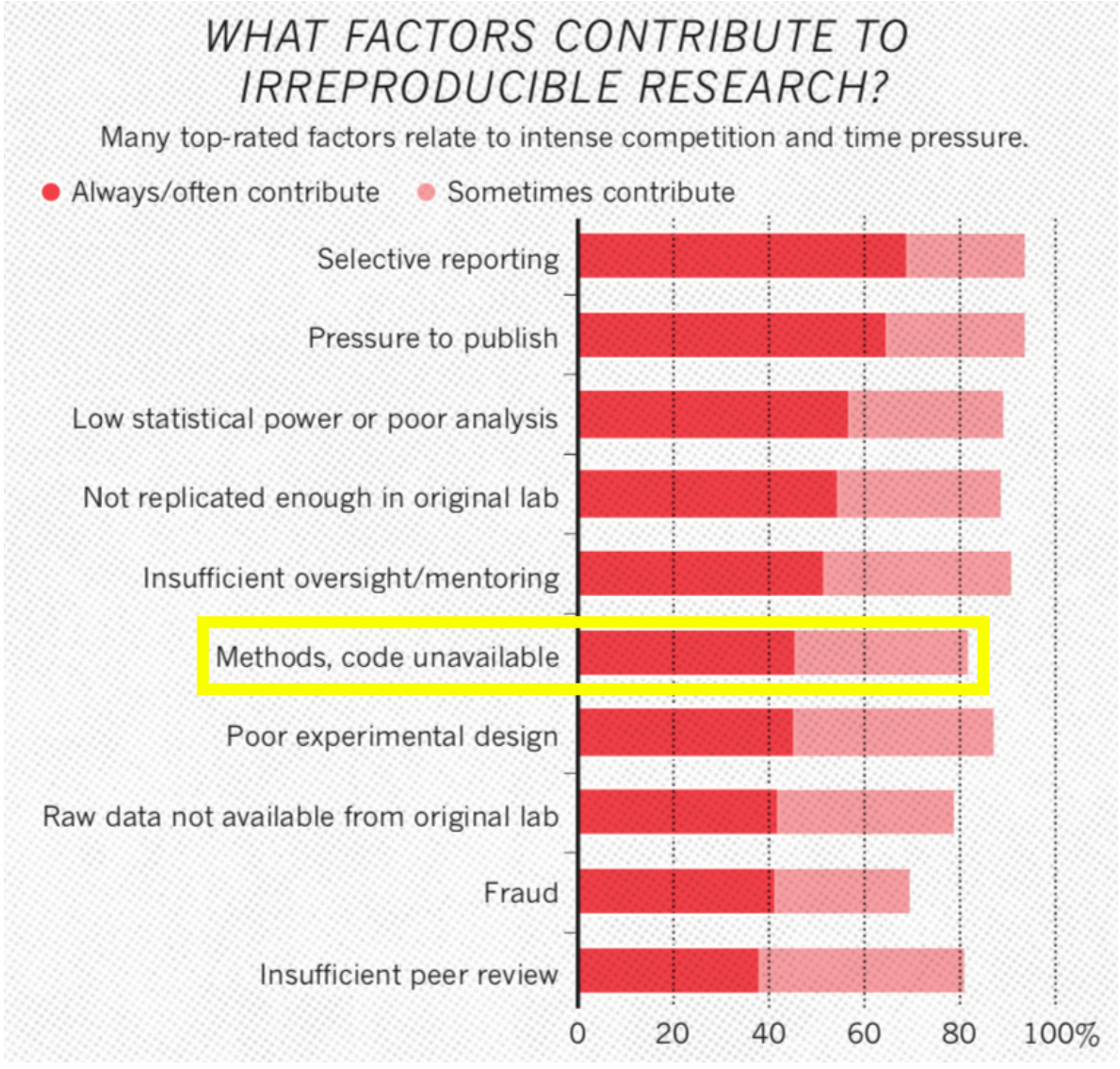


Aschwanden and King 2015

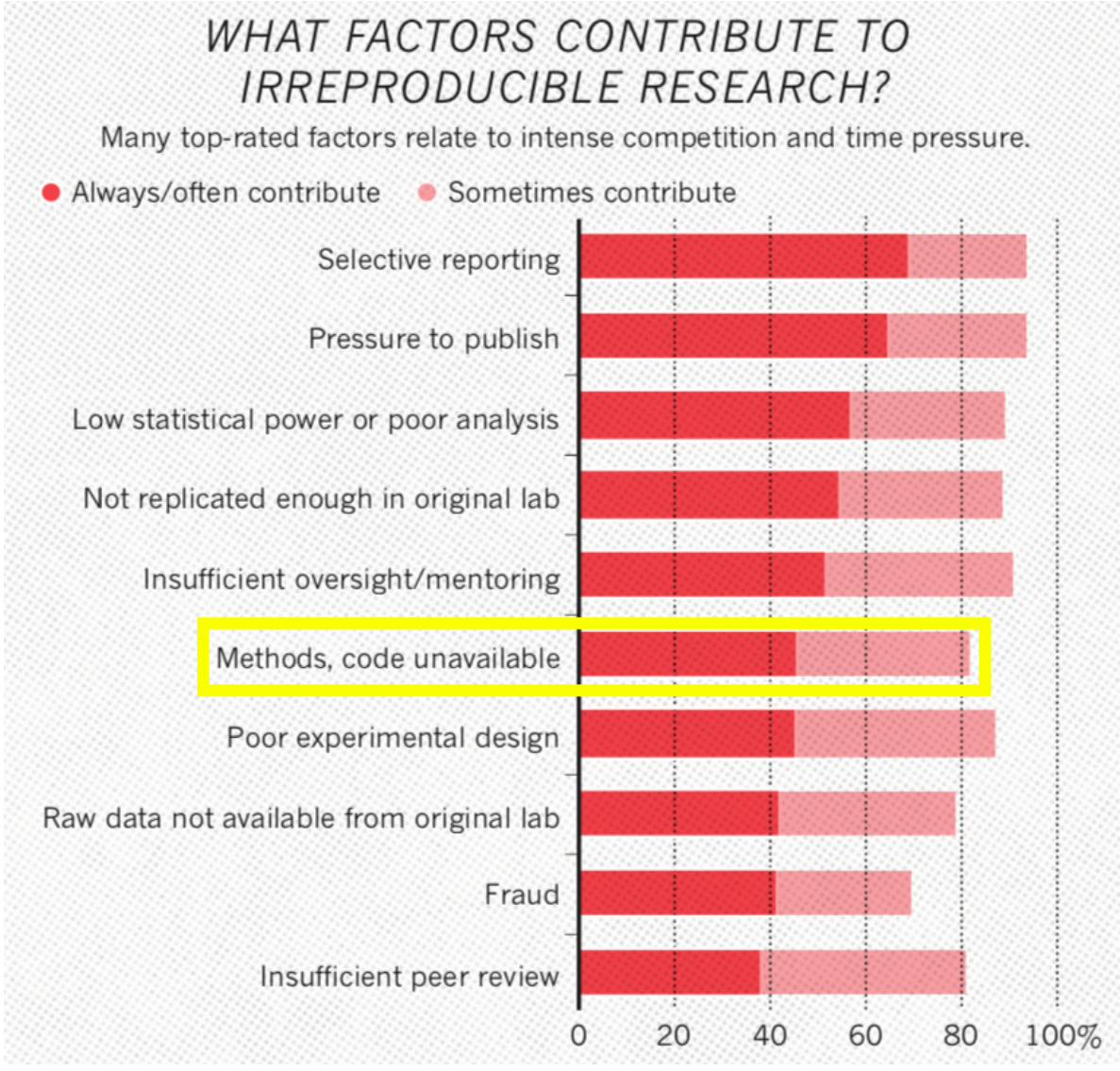
fivethirtyeight.com/features/science-isnt-broken

Pleasevote!

What are the factors contributing to the crisis?



Computation is playing a significant role



Computation is playing a significant role

PNAS PNAS PNAS

An empirical analysis of journal policy effectiveness for computational reproducibility

Victoria Stodden[a,1], **Jennifer Seiler**[b], **and Zhaokun Ma**[b]

[a]School of Information Sciences, University of Illinois at Urbana–Champaign, Champaign, IL 61820; and [b]Department of Statistics, Columbia University, New York, NY 10027

Edited by David B. Allison, Indiana University Bloomington, Bloomington, IN, and accepted by Editorial Board Member Susan T. Fiske January 9, 2018 (received for review July 11, 2017)

A key component of scientific communication is sufficient information for other researchers in the field to reproduce published findings. For computational and data-enabled research, this has often been interpreted to mean making available the raw data from which results were generated, the computer code that generated the findings, and any additional information needed such as workflows and input parameters. Many journals are revising author guidelines to include data and code availability. This work evaluates the effectiveness of journal policy that requires the data and code necessary for reproducibility be made available postpublication by the authors upon request. We assess the effectiveness of such a policy by (*i*) requesting data and code from authors and (*ii*) attempting replication of the published findings. We chose a random sample of 204 scientific papers published in the journal *Science* after the implementation of their policy in February 2011. We found that we were able to obtain artifacts from 44% of our sample and were able to reproduce the findings for 26%. We find this policy—author remission of data and code postpublication upon request—an improvement over no policy, but currently insufficient for reproducibility.

reproducible research | data access | code access | reproducibility policy | open science

putational reproducibility of published results. We use a survey instrument to test the availability of data and code for articles published in *Science* in 2011–2012. We then use the scientific communication standards from the 2012 Institute for Computational and Experimental Research in Mathematics (ICERM) workshop report to evaluate the reproducibility of articles for which artifacts were made available (11). We then assess the impact of the policy change directly, by examining articles published in *Science* in 2009–2010 and comparing artifact ability to our postpolicy sample from 2011–2012. Finally, we discuss possible improvements to journal policies for enabling reproducible computational research in light of our results.

Results

We emailed corresponding authors in our sample to request the data and code associated with their articles and attempted to replicate the findings from a randomly chosen subset of the articles for which we received artifacts. We estimate the artifact recovery rate to be 44% with a 95% bootstrap confidence interval of the proportion $[0.36, 0.50]$, and we estimate the replication rate to be 26% with a 95% bootstrap confidence interval $[0.20, 0.32]$.

Motivation: Journal Policy Impacts

Table 1. Responses to emailed requests (n = 180)

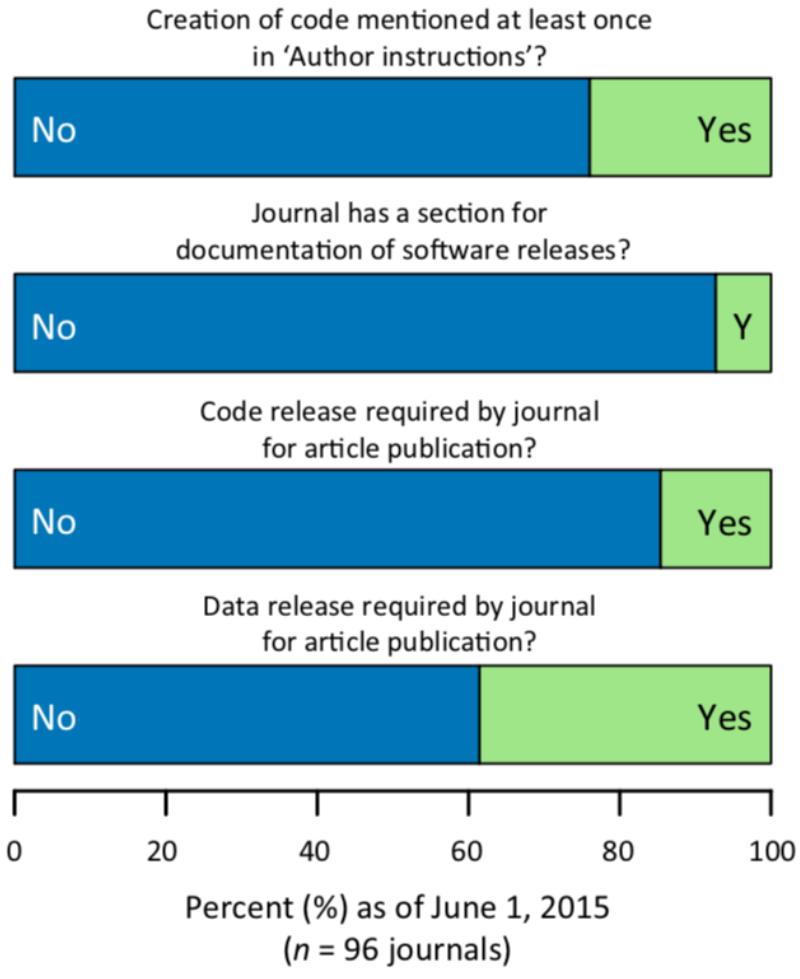
Type of response	Count	Percent, %
Did not share data or code:		
Contact another person	20	11
Asked for reasons	20	11
Refusal to share	12	7
Directed back to supplement	6	3
Unfulfilled promise to follow up	5	3
Impossible to share	3	2
Shared data and code	65	36
Email bounced	3	2
No response	46	26

Motivation: Journal Policy Impacts

Table 4. Classification of reproducibility effort (n = 22)

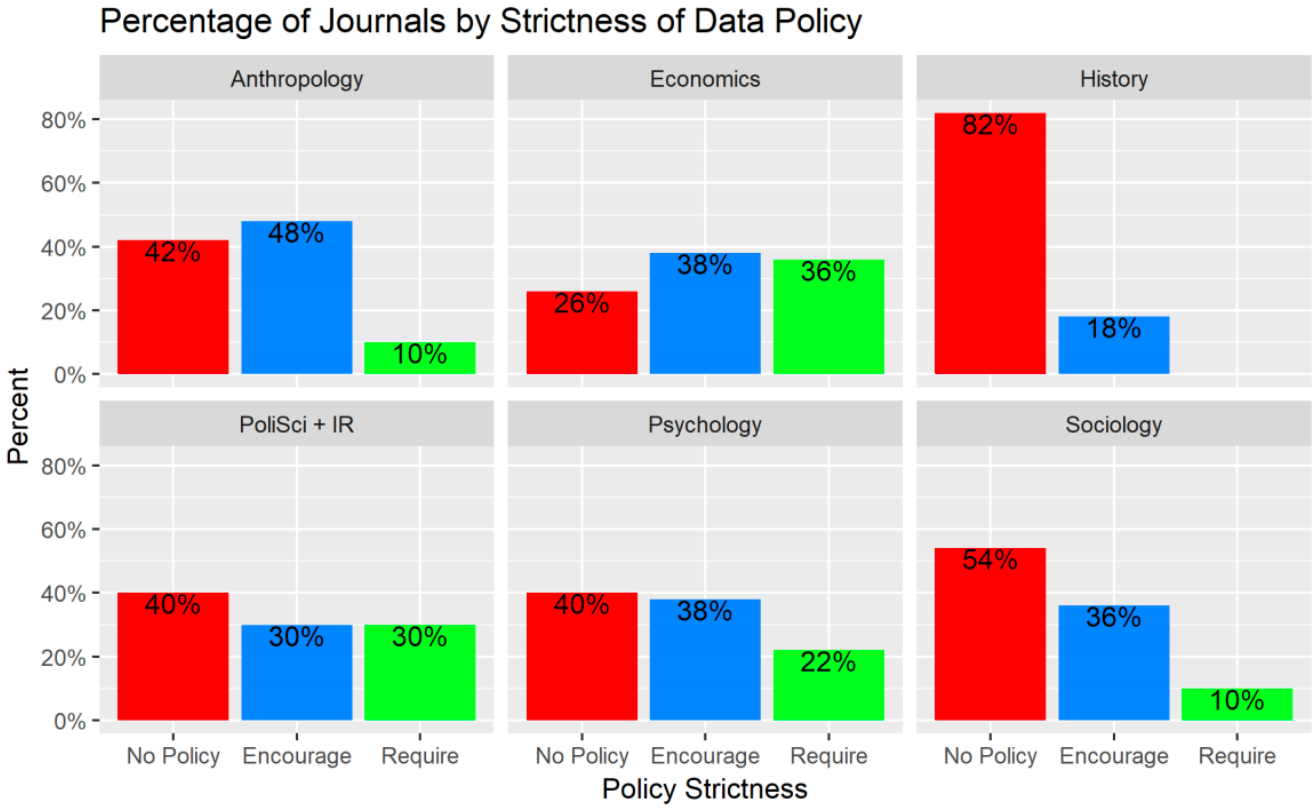
Classification	Percent, %
Impossible to reproduce (missing essential code, data, or methodology)	5
Nearly impossible to reproduce (specialized hardware, intense computation requirements, sensitive data, human study, or other unavoidable reasons)	14
Difficult to reproduce because of unavoidable inherent complexity (e.g., requiring 300 million Markov chain Monte Carlo steps on each dataset, or needing months to do runs)	14
Reproducible with substantial tedious effort (e.g., individual download of a large number of datasets, hand coding of data into a new format, i.e., from an image, many archiving steps required)	5
Reproducible with substantial intellectual effort (e.g., methods well defined but required some knowledge of jargon or understanding of the field; or down the rabbit hole references to past articles required to reproduce; etc.)	5
Could reproduce with fairly substantial skill and knowledge (e.g., required GPU programing abilities to run code that wasn't given; translating complex models into MATLAB code; pseudo code with functions not detailed described in text into code; missing scripts)	23
Reproducible after tweaking (e.g., missing parameters required fiddling to find, missing modified code lines, missing arguments required for differing architecture; missing minor method step)	5
Minor difficulty in reproducing (e.g., installing a specialized library, converting to a different computational system)	18
Straightforward to reproduce with minimal effort	14

Motivation: Ecology Journal Policies



Meeslan, Heer and White 2016 *Trends in Eco Evo*

Motivation: Social Science Journal Policies



Crosas et al. 2018 *SocArXiv*

Crisis and Opportunity

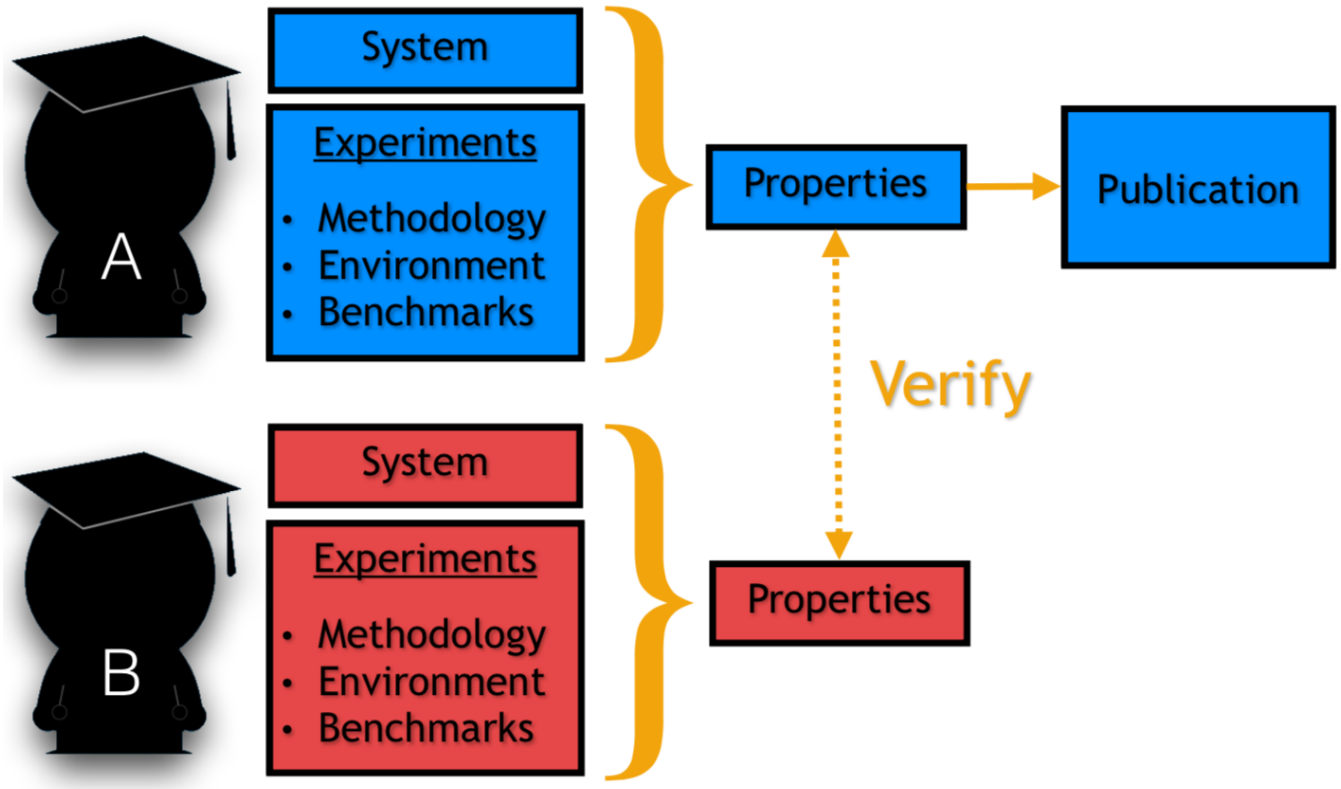
The Chinese use two brush strokes to write the word 'crisis.' One brush stroke stands for danger; the other for opportunity. In a crisis, be aware of the danger–but recognize the opportunity.

– John F. Kennedy

Benefaction: the real benefit

$$f(\mathit{benefaction}) = \mathit{science}$$

Opportunity: Benefaction not just reproducibility

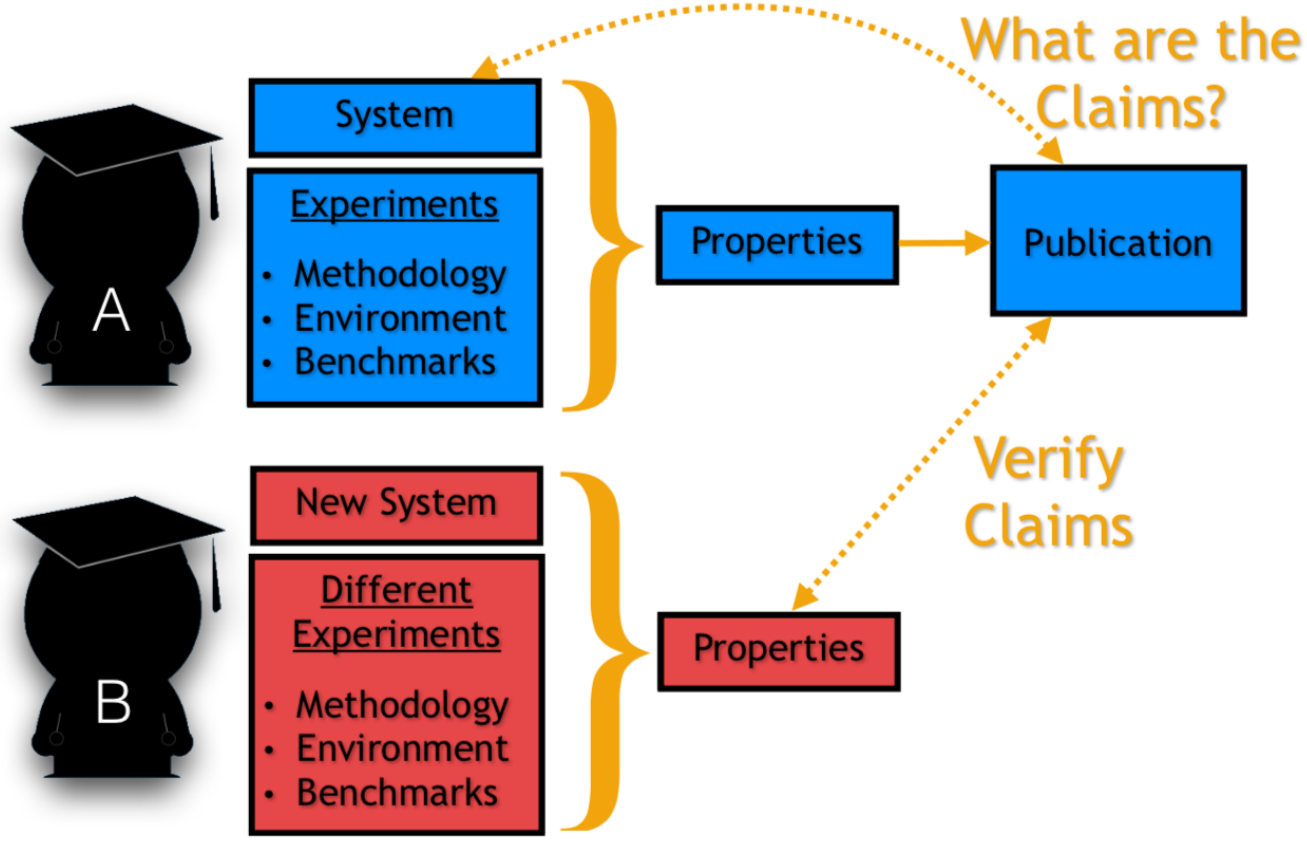


Repetition

Colberg et al. 2015 Comm ACM

Opportunity: Benefaction not just reproducibility

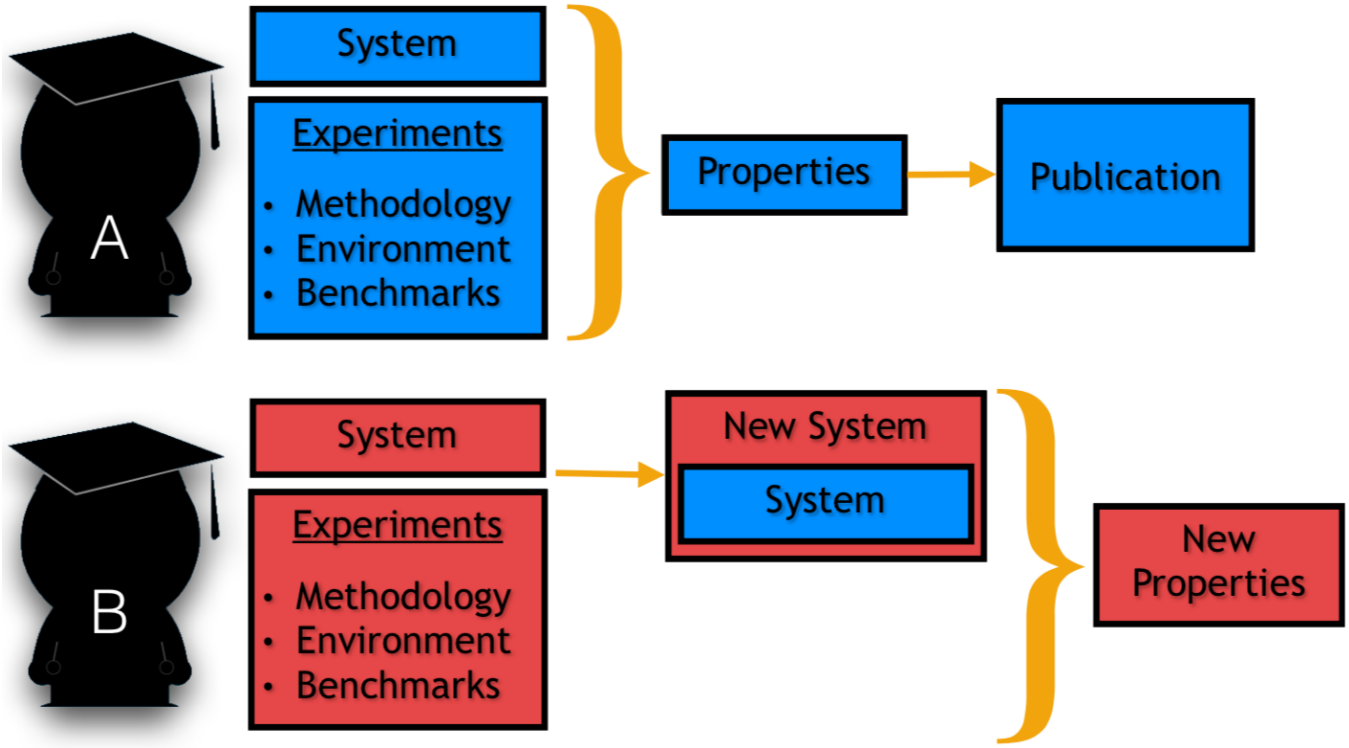
Reproduction



Colberg et al. 2015 Comm ACM

Opportunity: Benefaction not just reproducibility

Benefaction



Colberg et al. 2015 Comm ACM

What can we do?

Top Three List

1. Learn a programming language (statistical, command)

Top Three List

1. Learn a programming language (statistical, command)
2. Make your workflow transparent (data, code and notes)

Top Three List

1. Learn a programming language (statistical, command)
2. Make your pipelines transparent (data, code and notes)
3. Share and take credit!

Programming Languages

- *R*: free, open-source, designed for analysis
- *python*: also free and open-source, designed for more general computation
- *BASH*: command language, glues software together

Programming Languages

- *R*: free, open-source, designed for analysis
- *python*: also free and open-source, designed for more general computation
- *BASH*: command language
- *Ruby*, *Java*, *C++*, *MatLab*, *Octave*, *Stata*, and many more.

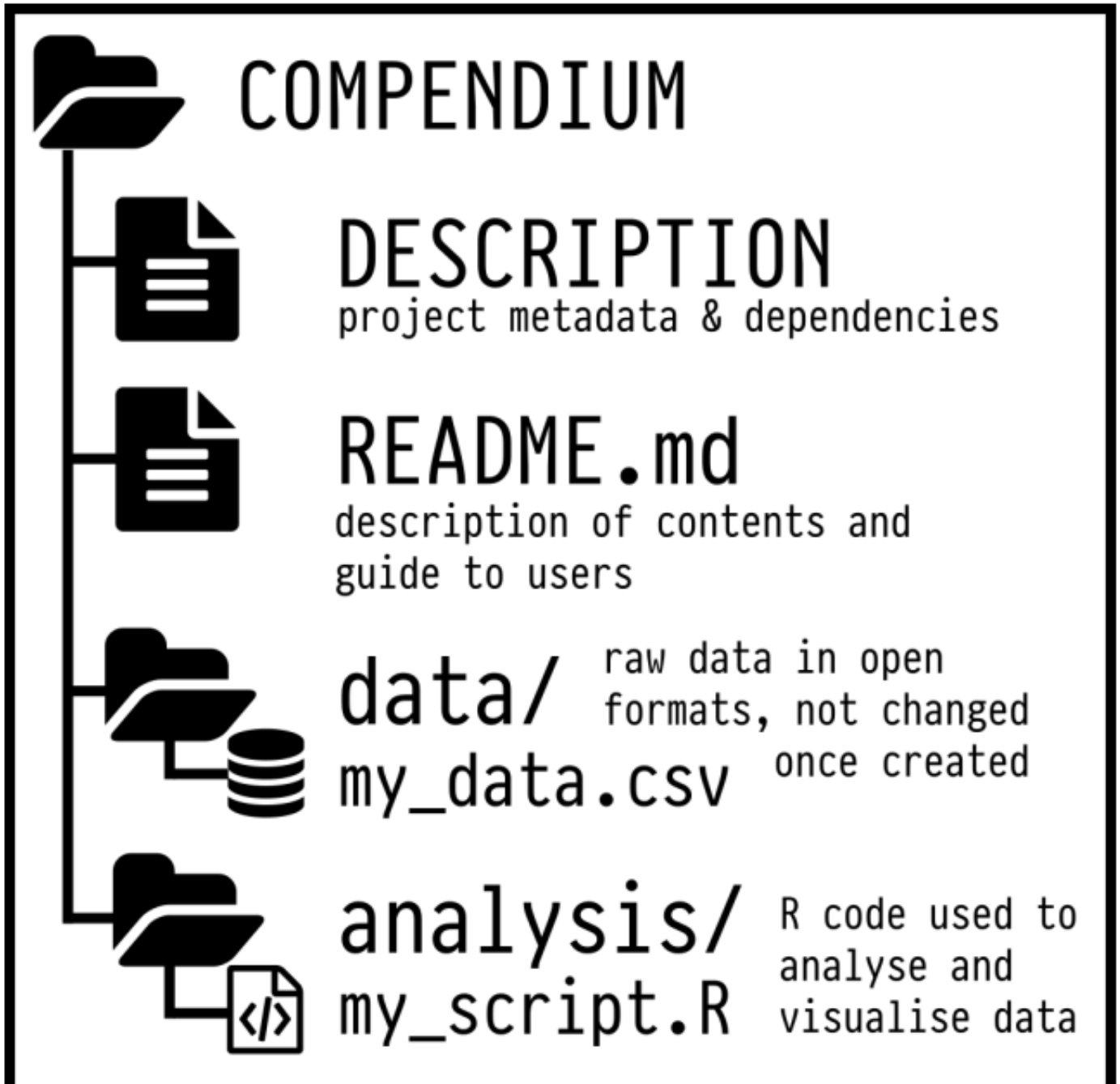
Transparency

- Consistent project architecture

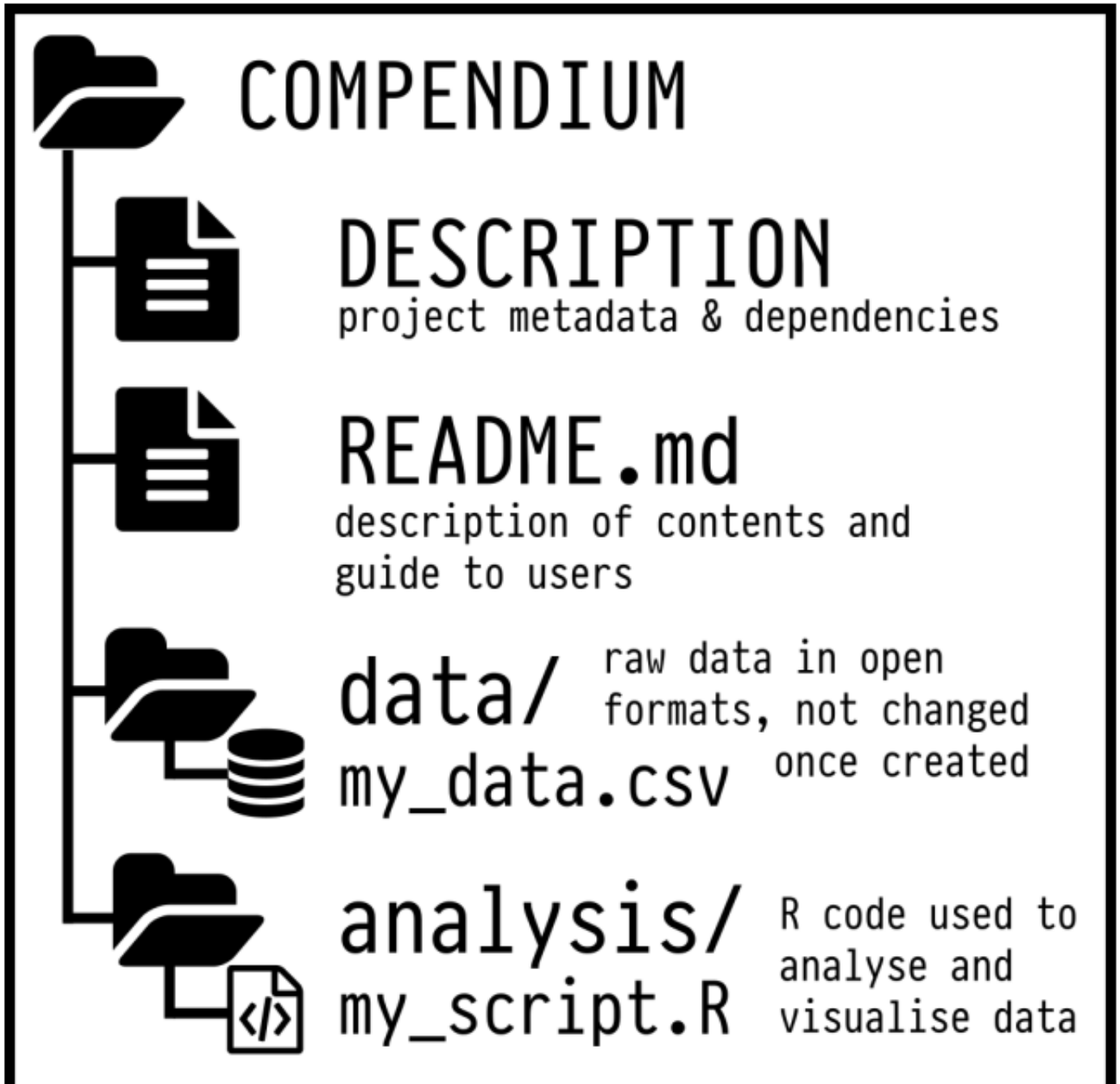
Transparency

- Project architecture
- Version control

Transparency: Project Architecture



Transparency: Project Architecture



Transparency: Version Control

Showing 2 changed files with 173 additions and 19 deletions. Unified Split

BIN +22.9 KB talk/img/ostrich.jpg View

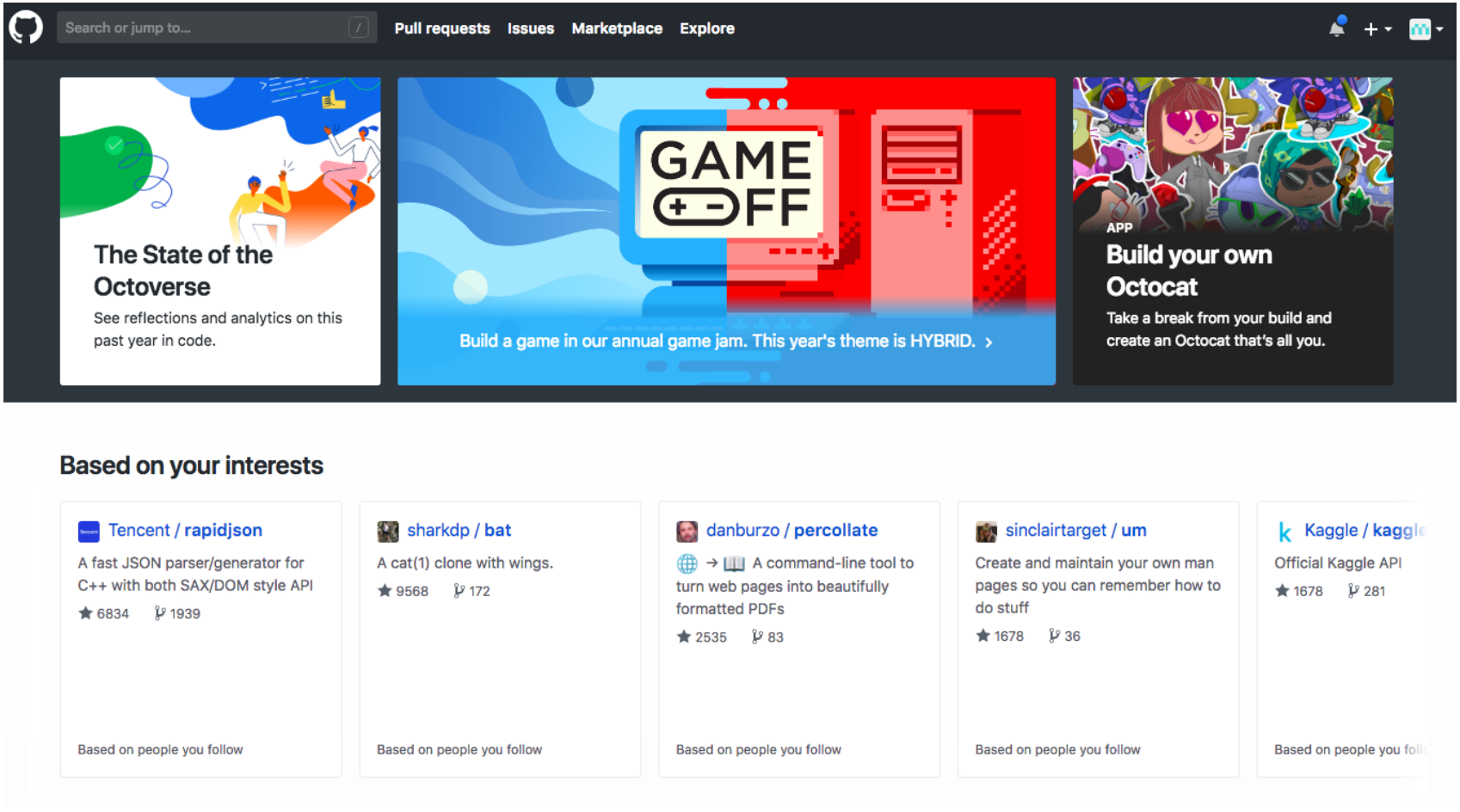
Binary file not shown.

192 talk/talk.Rmd View

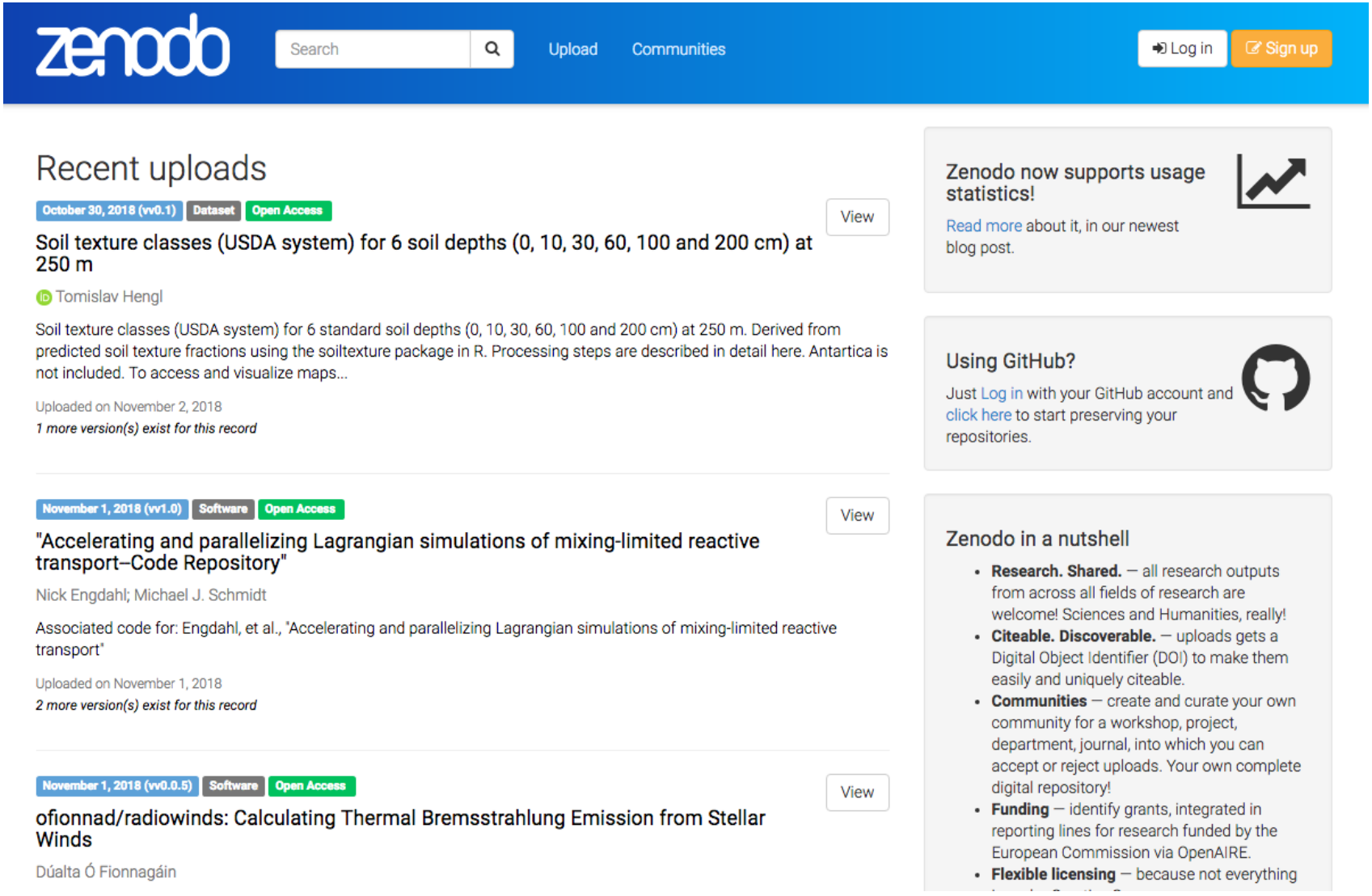
```
@@ -39,7 +39,9 @@ header-includes:
 - "A scientist, a reviewer and an editor walk into a bar..."
 - Crosas, White et al., Poisot, Marwick

- 3. Tools for improving benefaction not just reproducibility (15 min)
+ 3. Tools and resources for improving benefaction not just reproducibility (15 min)
+
+

 *ReproSci Top 10 Checklist*

@@ -65,6 +67,7 @@ header-includes:
 - Capsules (Code Ocean, ReproZip, encapsulator)
 - Literate programming (Rmarkdown, Jupyter, OverLeaf, RStudio)
 - Leaks in the Pipeline: Not recorded information, pseudo-non-determinism
+ - ??? (ROSA, CodeDepends, others)

 4. Let the computers do the work! (10 min)

@@ -90,13 +93,158 @@ intelligently and get them to do the right things for us

 ![](img/Baker2016_3.png)
```

Transparency: Version Control

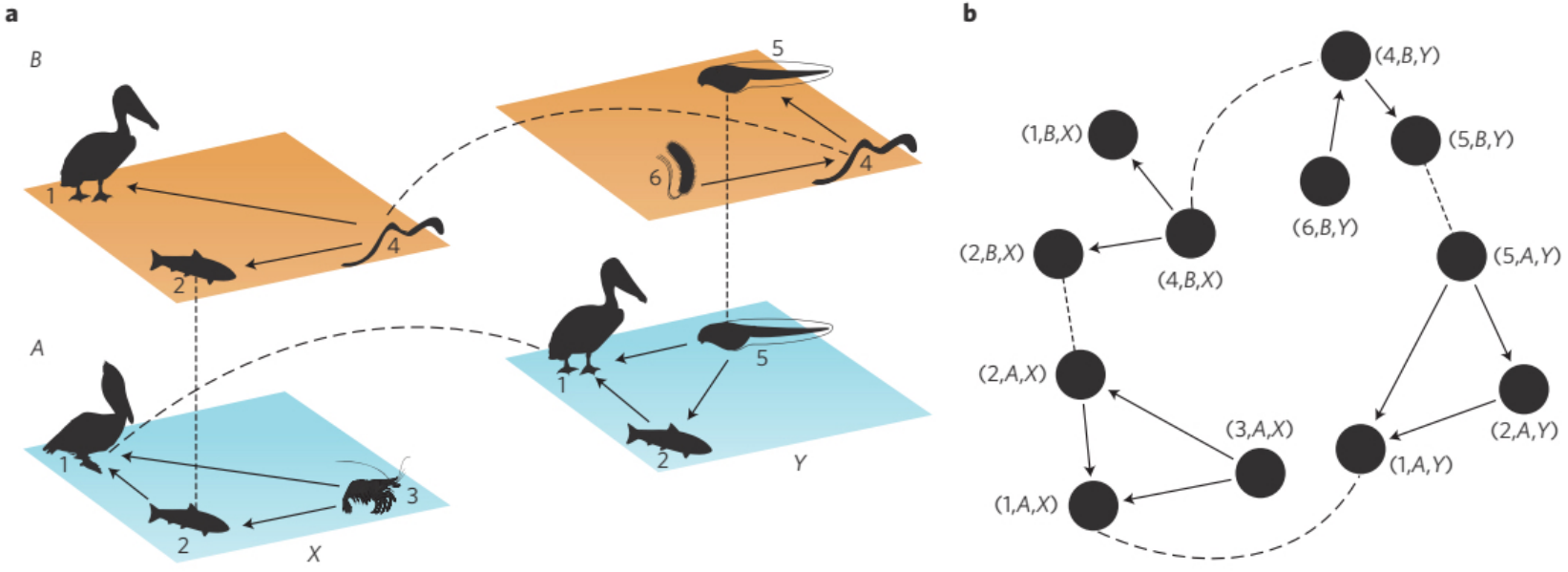
www.github.com



Transparency: Version Control



Reality



Reality

www.nature.com/scientificdata

SCIENTIFIC DATA

OPEN

Comment: If these data could talk

Thomas Pasquier[1], Matthew K. Lau[2], Ana Trisovic[3,4], Emery Boose[2], Ben Couturier[3], Mercè Crosas[5], Aaron M. Ellison[2], Valerie Gibson[4], Chris Jones[4] & Margo Seltzer[1]

Received: 12 April 2017
Accepted: 24 July 2017
Published: xx xxx 2017

In the last few decades, data-driven methods have come to dominate many fields of scientific inquiry. Open data and open-source software have enabled the rapid implementation of novel methods to manage and analyze the growing flood of data. However, it has become apparent that many scientific fields exhibit distressingly low rates of repeatability and reproducibility. Although there are many dimensions to this issue, we believe that there is a lack of formalism used when describing end-to-end published results, from the data source to the analysis to the final published results. Even when authors do their best to make their research and data accessible, this lack of formalism reduces the clarity and efficiency of reporting, which contributes to issues of reproducibility. Data provenance aids both repeatability and reproducibility through *systematic* and *formal* records of the relationships among data sources, processes, datasets, publications and researchers.

Reality: Common Ground

Reality

Most scientists don't want to produce software, they want to do science.

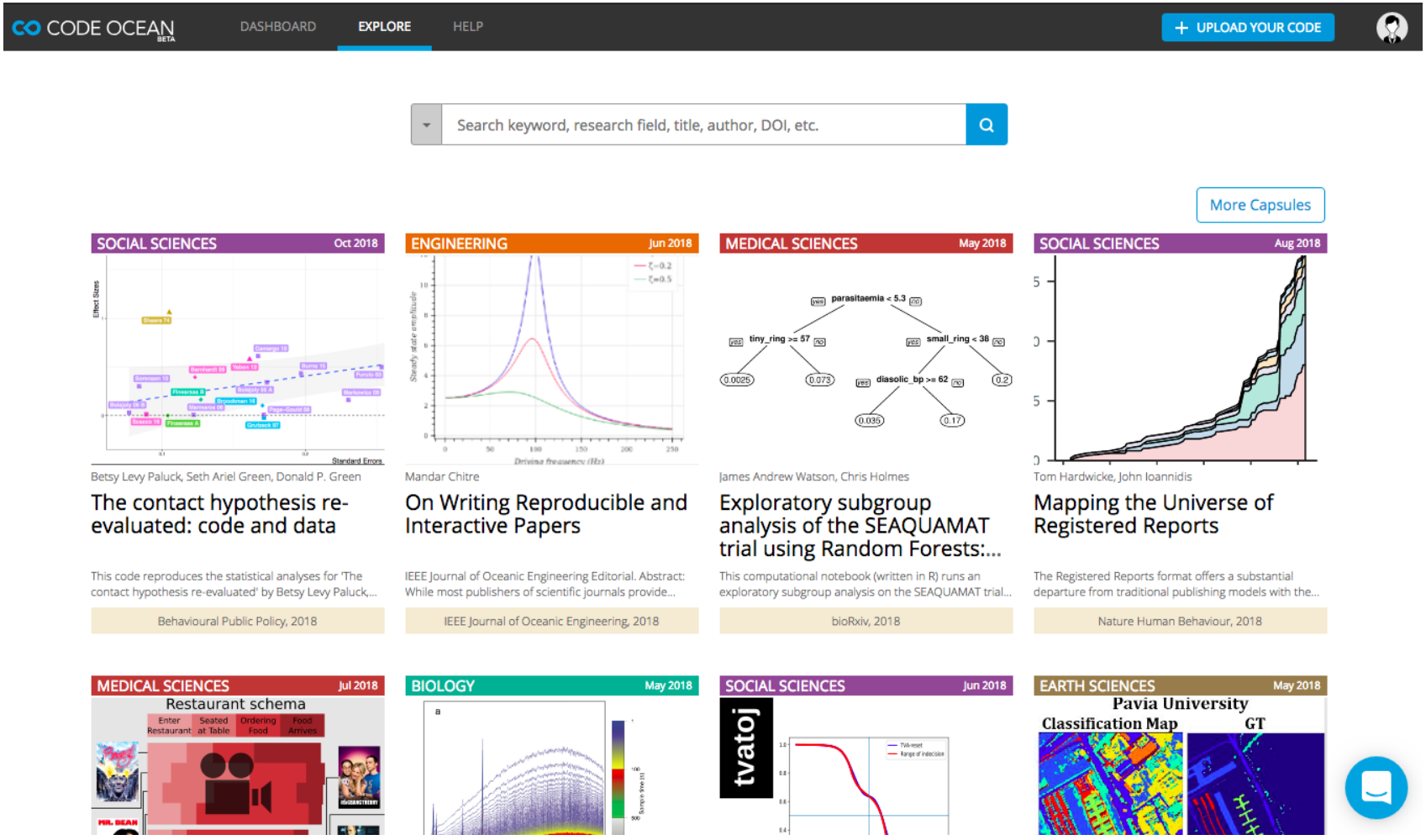
Wild-wild West of Statistical Software

R-ube Goldberg Coding

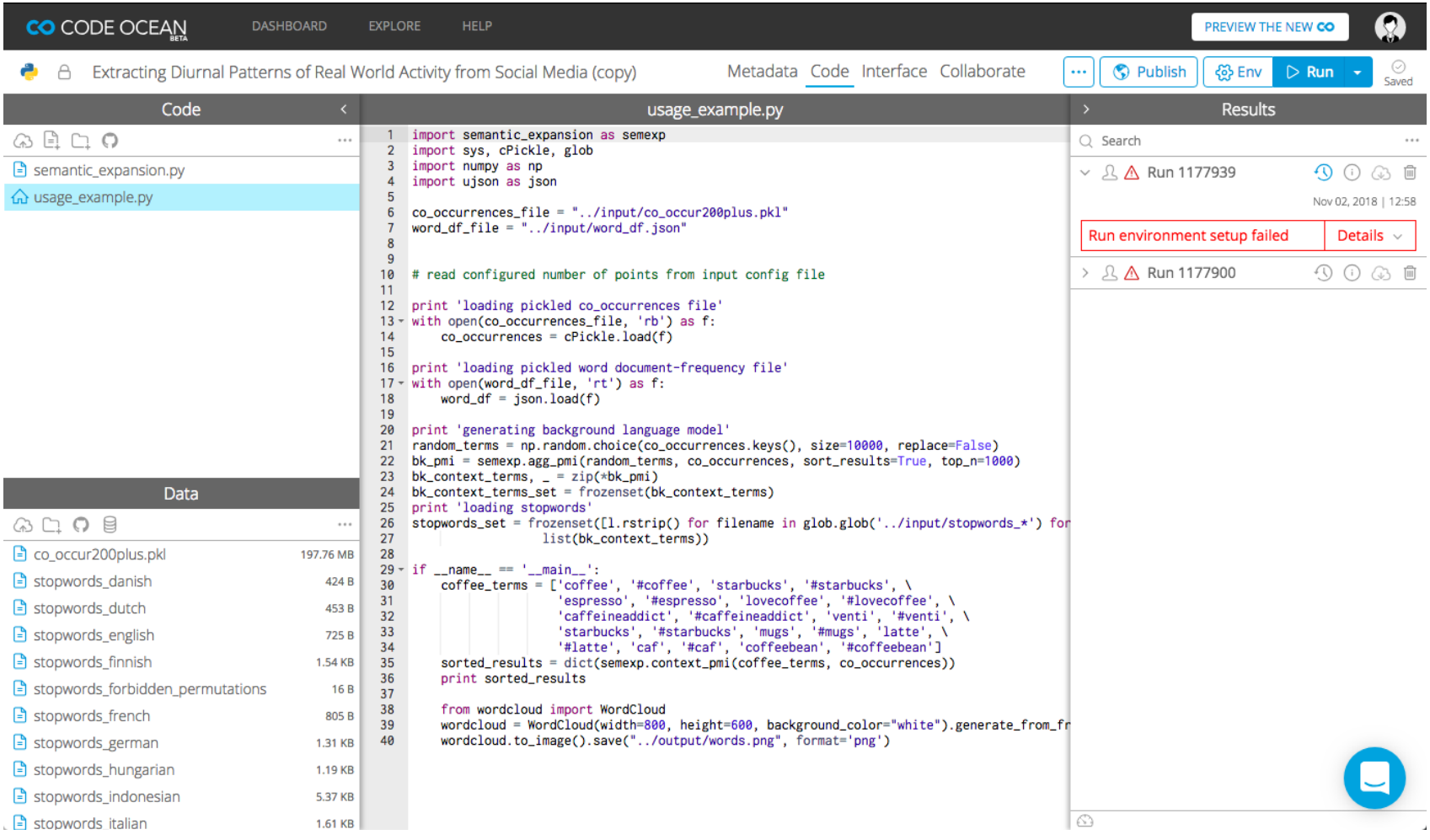
Code Cleaning and Encapsulation

Let's get the computer to work for us and decrease error rates and increase sharing.

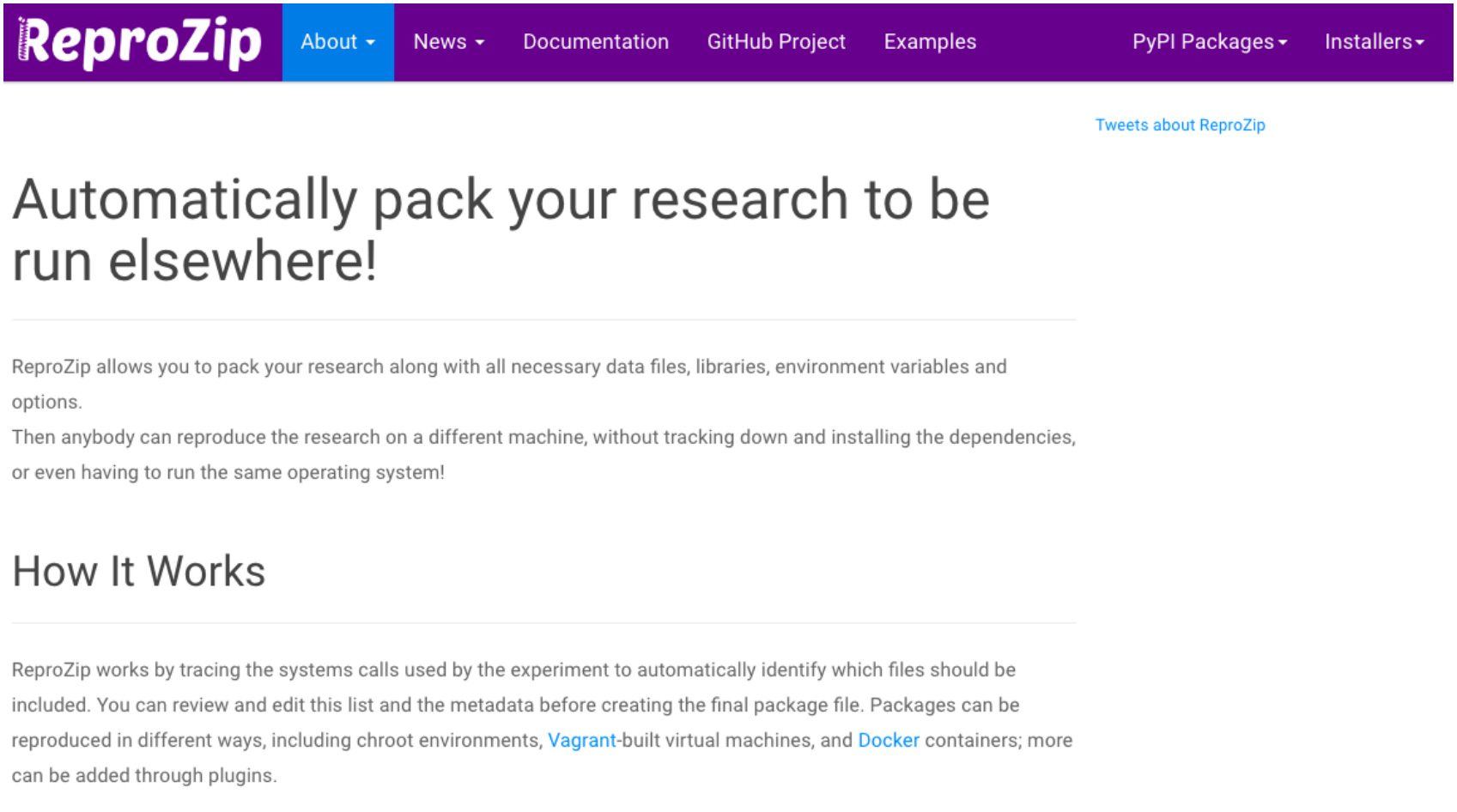
Code Cleaning and Encapsulation



Code Cleaning and Encapsulation



Code Cleaning and Encapsulation



Code Cleaning and Encapsulation

Sharing and Preserving Computational Analyses for Posterity with *encapsulator*

Thomas Pasquier
University of Cambridge

Matthew K. Lau and Xueyuan Han
Harvard University

Elizabeth Fong and Barbara S. Lerner
Mount Holyoke College

Emery R. Boose, Mercè Crosas, Aaron M. Ellison, and Margo Seltzer
Harvard University

Editors: Lorena A. Barba, labarba@gwu.edu; George K. Thiruvathukal, gkt@cs.luc.edu

Reproducibility has become a recurring topic of discussion in many scientific disciplines.[1] Although it might be expected that some studies will be difficult to reproduce, recent conversations highlight important aspects of the scientific endeavor that could be improved to facilitate reproducibility. Open data and open source software are two important parts of a concerted effort to achieve reproducibility.[2] However, multiple publications point out these approaches' shortcomings,[3,4] such as the identification of dependencies, poor documentation of the installation processes, "code rot," failure to capture dynamic inputs, and technical barriers.

In prior work,[5] we pointed out that open data and open source software alone are insufficient to ensure reproducibility, as they do not capture information about the computational execution, that is, the "process" and context that produced the results using the data and code. In keep-

Code Cleaning and Encapsulation

RClean: Simplify code based on specified results.

Lau 2018 *CRAN*

Encapsulator: generate a cleaned capsule.

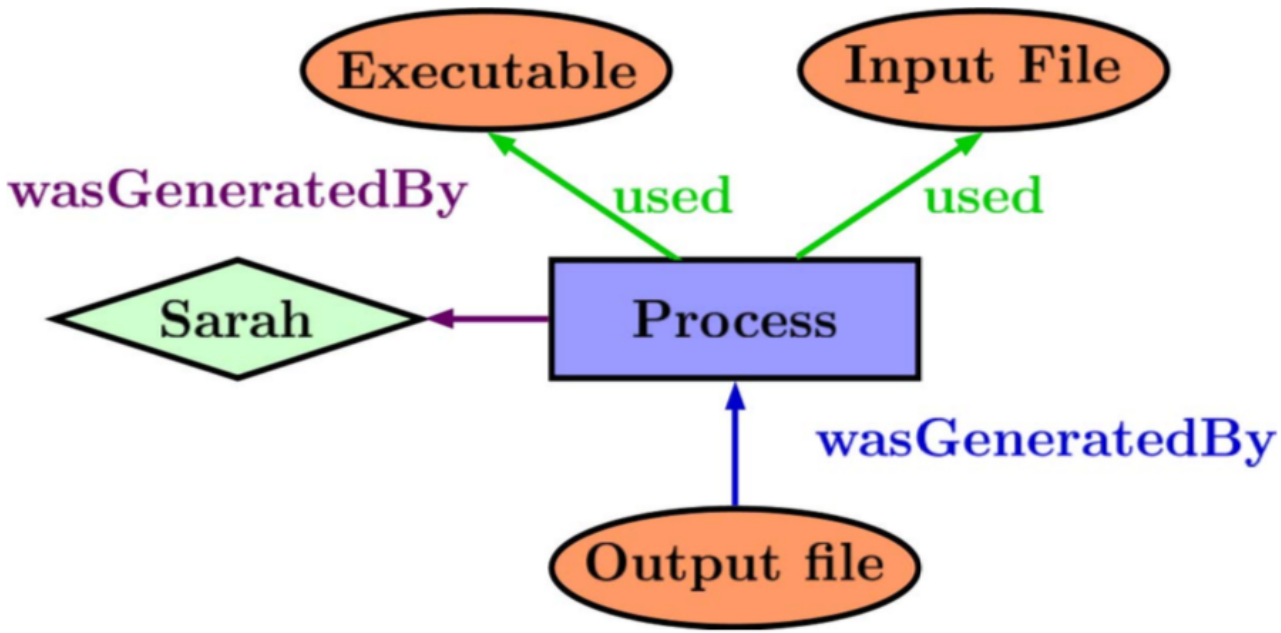
Pasquier et al. 2017 *IEEE CISE*

- Capsule = all necessary software and data
- Cleaned = organize files, remove non-essential code and re-format

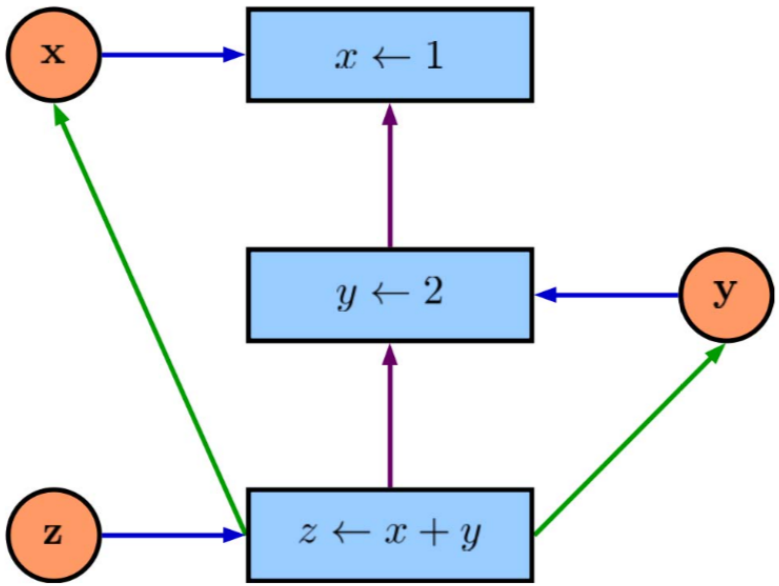
Tools: Encapsulator

- near stream-of-consciousness coding that follows a train of thought in script development,
- output to console that is not written to disk,
- intermediate objects that are abandoned,
- library and new data calls throughout the script,
- output written to disk but not used in final documents,
- code that is not modularized,
- code that is syntactically correct but not particularly comprehensible.

Encapsulation: Under-the-hood



Rclean and Encapsulator



Encapsulation: Under-the-hood

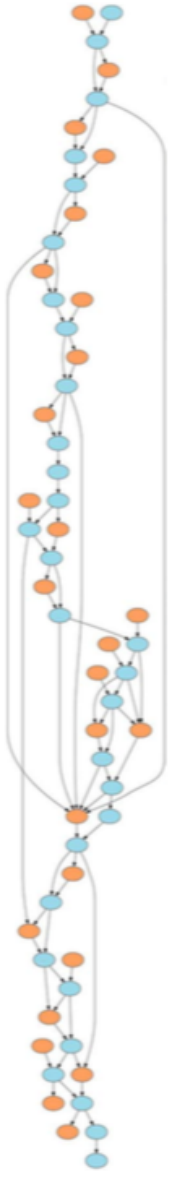


Figure 4. Provenance graph corresponding to a small R script (approximately 60 lines of code).

Encapsulation: Under-the-hood

Example: Messycode

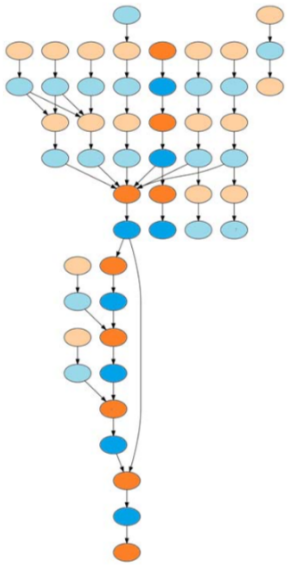


Figure 5. Data dependency transformation of the provenance graph shown in Figure 4.

Tools: Encapsulator

Basic Usage (current paradigm):

1. Code as usual in your normal environment while recording provenance
2. Run encapsulator from the console
3. List desired results

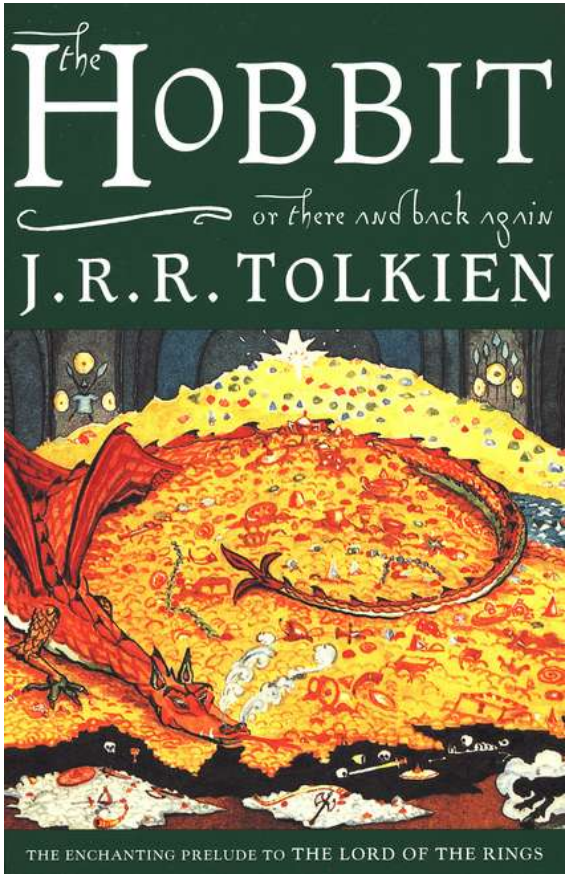
Tools: Encapsulator

Basic Usage (current paradigm):

1. Code as usual in your normal environment while recording provenance
2. Run encapsulator from the console
3. List desired results

Product = Capsule containing essential code and data organized following project best practices inside a virtual machine

** Software should not limit science **


the HOBBIT
or there and back again
J.R.R. TOLKIEN
THE ENCHANTING PRELUDE TO THE LORD OF THE RINGS

Computation will not replace good scientific thought or practice.

Computation will not replace good scientific thought or practice.

But hopefully it can help.

Thanks for listening!

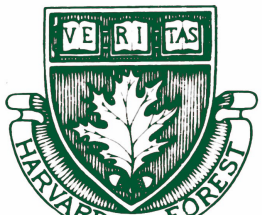
Resources:

- Databases: *Github Data*, *Data Dryad*, *Figshare*, *Dataverse*
- Open Source Communities: *ROpenSci*, *RStudio*, *Center for Open Science*, *Transparency in Ecology and Evolution*
- Tools: *Reprozip*, *CodeOcean*, *Rclean*

Contact Info:

- **Email: *matthewklau@fas.harvard.edu***
- **Github: MKLau**

Much of this work was supported by NSF SSI-1450277 (End-to-End Provenance) and ACI-1448123 (Citation++). More details are available at https://projects.iq.harvard.edu/provenance-at-harvard



UNIVERSITY OF CAMBRIDGE