WQD7001

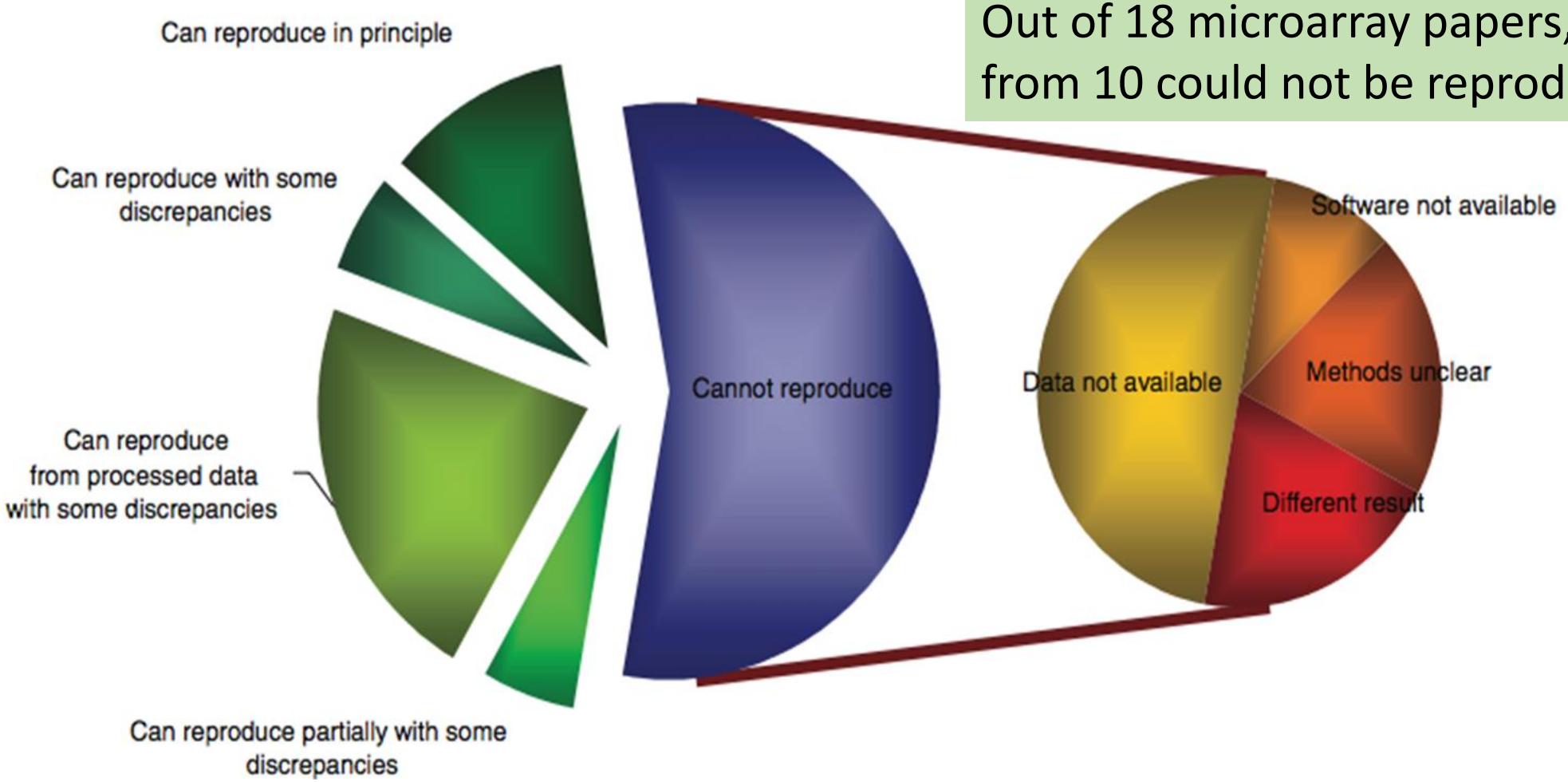
Reproducible Research in Data Science

Prepared by: Dr. Salimah Mokhtar

Have you ever tried to reproduce the results presented in a research paper?

Have you ever tried to reproduce your DS projects created last year?

Out of 18 microarray papers, results from 10 could not be reproduced



Essay

Why Most Published Research Findings Are False

John P. A. Ioannidis

PLOS | MEDICINE

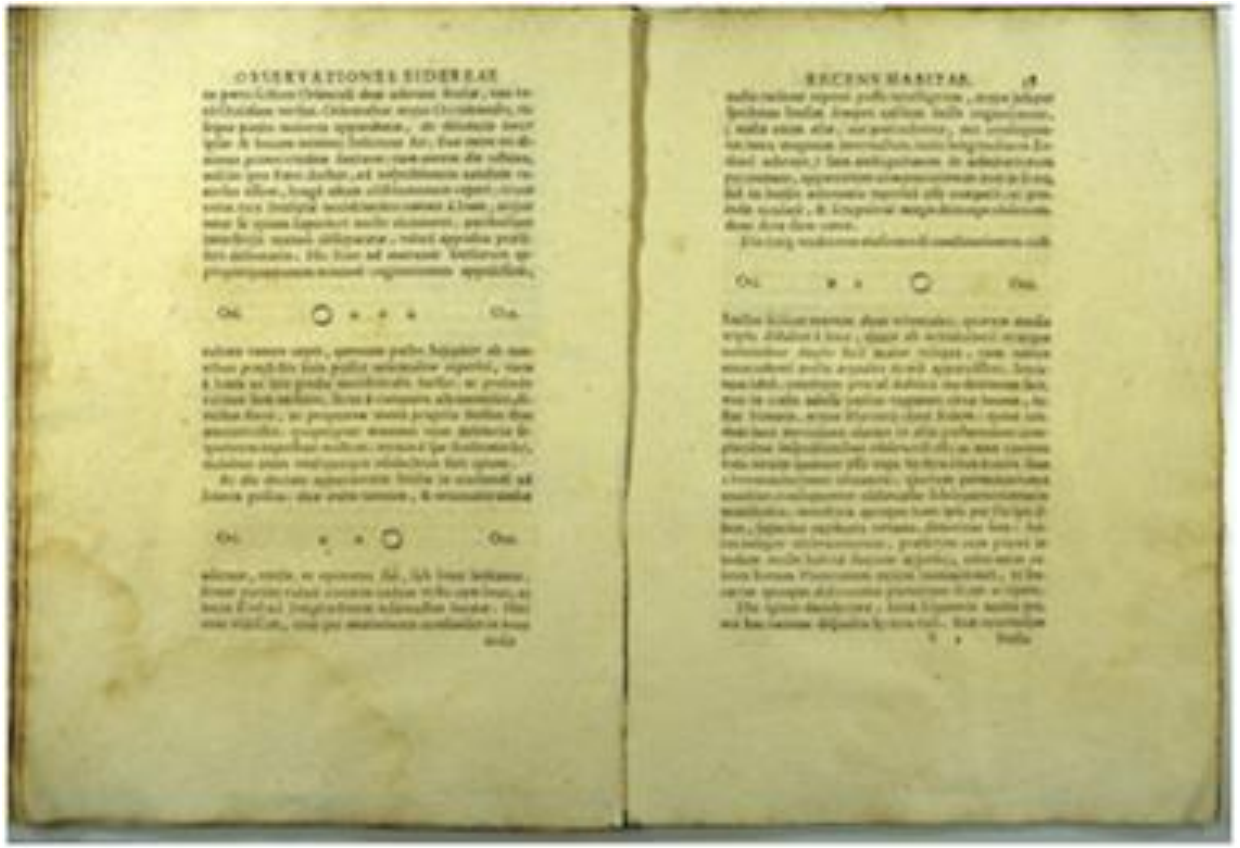
1. Ioannidis et al., (2009). Repeatability of published microarray gene expression analyses. *Nature Genetics* 41: 14
2. Ioannidis JPA (2005) Why Most Published Research Findings Are False. *PLoS Med* 2(8)

Lecture Coverage

- Scientific Reproducibility Crisis
- What is reproducible research?
- Why should data science team care about reproducibility?
- How can a data science team achieve reproducibility?
- Literate Programming
- Tools to Aid in Reproducible Research
- Reproducibility Skills
- A Hierarchy of Reproducibility
- What Problem Does Reproducibility Solve?
- Reproducibility Enhancement Principles

The Beginning of Reproducible Research

Galileo Galilei



Two pages from Galilei's Sidereus Nuncius (“The Starry Messenger” or “The Herald of the Stars”), Venice, 1610.

Galileo’s notes directly integrated his **data** (drawings of Jupiter and its moons), key **metadata** (timing of each observation, weather, and telescope properties), and **text** (methods, analysis, and conclusions)

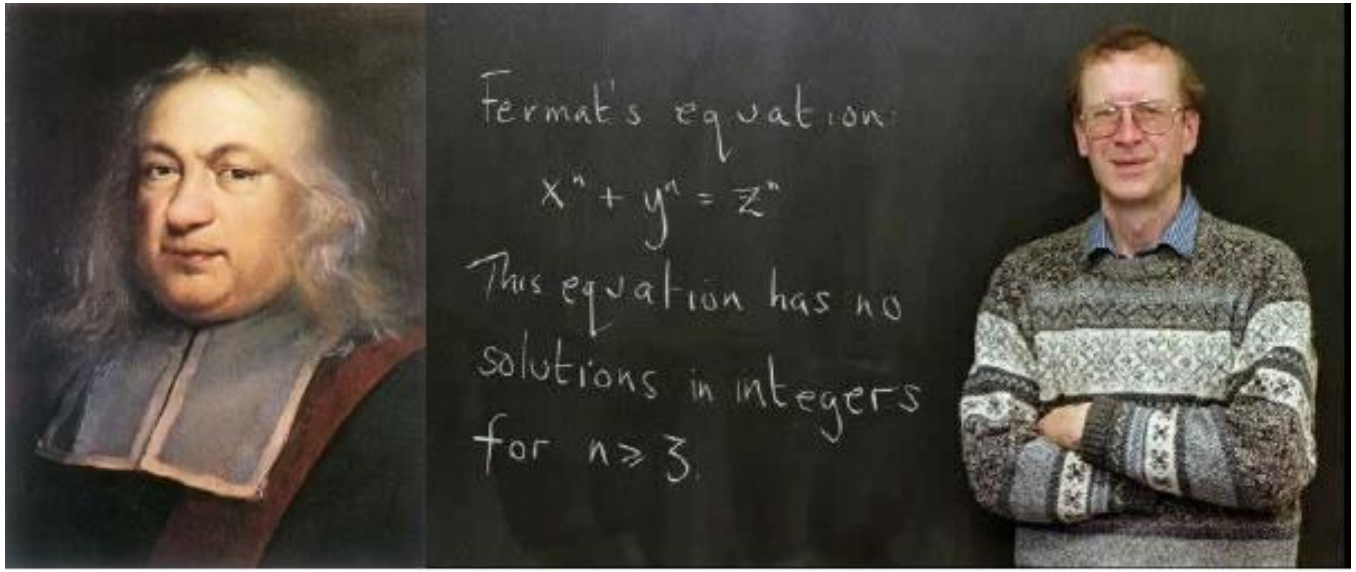
The Beginning of Irreproducible Research

Pierre de Fermat

$$a^n + b^n = c^n$$

- for $n > 2$, no three positive integers a, b, and c exist

Fermat's Last Theorem: It is impossible to separate a cube into two cubes, or a fourth power into two fourth powers, or in general, any power higher than the second into two like powers.



rpubs.com/mdozmorov/rr

Irreproducibility - Default Setting for all of Science

Is particularly common across the computational sciences.

Reasons for acceptance of the irreproducible work:

1. **Think "Big Picture"**. People are interested in the science, not the dull experimental setup, so don't describe it. If necessary, camouflage this absence with brief, high-level details of insignificant aspects of your methodology.

2. **Be abstract**. Pseudo-code is a great way of communicating ideas quickly and clearly while giving readers no chance to understand the subtle implementation details (particularly the custom toolchains and manual interventions) that actually make it work.

https://arxiv.org/pdf/1504.00062.pdf

Irreproducibility - Default Setting for all of Science

3. **Short and sweet.** Any limitations of your methods or proofs will be obvious to the careful reader, so there is no need to waste space on making them explicit. However much work it takes colleagues to fill in the gaps, you will still get the credit if you just say you have amazing experiments or proofs.

4. **The deficit model**. You're the expert in the domain, only you can define what algorithms and data to run experiments with. In the unhappy circumstance that your methods do not do well on community curated benchmarks, you should create your own bespoke benchmarks and use those (and preferably not make them available to others).

5. **Don't share**. Doing so only makes it easier for other people to scoop your research ideas, understand how your code actually works instead of why you say it does, or worst of all to understand that your code doesn't actually work at all.

Converging Trends

Two (competing?) conjectures:
1. Scientific research will become massively more computational,
2. Scientific computing will become dramatically more transparent.

These trends need to be addressed simultaneously:
Better transparency will allow people to run much more ambitious computational experiments.
And better computational experiment infrastructure will allow researchers to be more transparent.

WORKING LIFE

By Lorena A. Barba

The hard road to reproducibility

Early in my Ph.D. studies, my supervisor assigned me the task of running computer code written by a previous student who was graduated and gone. It was hell. I had to sort through many different versions of the code, saved in folders with a mysterious numbering scheme. There was no documentation and scarcely an explanatory comment in the code itself. It took me at least a year to run the code reliably, and more to get results that reproduced those in my predecessor's thesis. Now that I run my own lab, I make sure that my students don't have to go through that.

"My students and I continuously discuss and perfect our standards."

In 2012, I wrote a manifesto in which I committed to best practices for reproducibility. Today, a new student arriving in my group finds all of our research code in tidy repositories, where every change is recorded automatically. Version control is our essential technology for record keeping and collaboration. Whenever we publish a paper, we create a "reproducibility package," deposited online, which includes the data sets and all the code that is needed to recreate the analyses and figures. These are the practices that work for us as computational scientists, but the principles behind them apply regardless of discipline.

It takes new students some time to learn how to work to these standards, but we have documentation and training materials to make it as painless as possible. My students don't resent investing their time in this. They know that practices like ours are crucial for the integrity of the scientific endeavor. They also appreci-

additional details were documented in the supplementary materials. It was the very definition of reproducible research.

Three years of work and hundreds of runs with four different codes taught us just how many ways there are to go wrong! Failing to record the version of any piece of software or hardware, overlooking a single parameter, or glossing over a restriction on how to use another researcher's code can lead you astray.

We've found that we can only

documented. Every step of the way, what another researcher might ne our results (run our code with our

ars TECHNICA

SCIENTIFIC METHOD —

Nature Editorial: If you want reproducible science, the software needs to be open source

According to an editorial in *Nature*, all scientific code should be released ...

KYLE NIEMEYER - 2/27/2012, 4:00 AM

rb relies heavily on computer programs, which analyze

FUTURE TENSE THE CITIZEN'S GUIDE TO THE FUTURE. APRIL 15 2016 7:08 AM

FROM SLATE, NEW AMERICA, AND ASU

The Reproducibility Crisis Is Good for Science

Weak statistics are getting called out, and replication is gaining respect.

590 127 67

By Monya Baker

ms in psychology and b many published studies

Report from Dagstuhl Seminar 16041

Reproducibility of Data-Oriented Experiments in e-Science

Edited by

Juliana Freire[1], Norbert Fuhr[2], and Andreas Rauber[3]

1 New York University, US
2 Universität Duisburg-Ess
3 TU Wien, AT, rauber@if

Abstract

This report documents the progr
cibility of Data-Oriented Experi
experiments play an important ro
effectiveness and performance oft

How scientists are addressing the 'reproducibility problem'

April 25, 2016 by Deborah Berry, Georgetown University, The Conversation

In scientific research, repetition is good. Credit: w4nd3r0ast/flickr, CC BY-ND

Recently a friend of mine on Facebook posted a link whose headline quoted a scientist saying "Most cancer research is largely a fraud." The quote is both out of context and many decades old. But its appearance still makes a strong point: the general public has a growing distrust of science and research.

Health & Science

The new scientific revolution: Reproducibility at last

By Joel Achenbach January 27, 2015

Diederik Stapel, a professor of social psychology in the Netherlands, had been a rock-star scientist — regularly appearing on television and publishing in top journals. Among his striking discoveries was that people exposed to litter and abandoned objects are more likely to be bigoted.

And yet there was often something odd about Stapel's research. When students asked to see the data behind his work, he couldn't produce it readily. And colleagues would sometimes look at his data and think: It's beautiful. Too beautiful. Most scientists have messy data, contradictory data, incomplete data, ambiguous data. This data was too good to be true.

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video | For A

Archive > Volume 496 > Issue 7446 > Editorial > Article

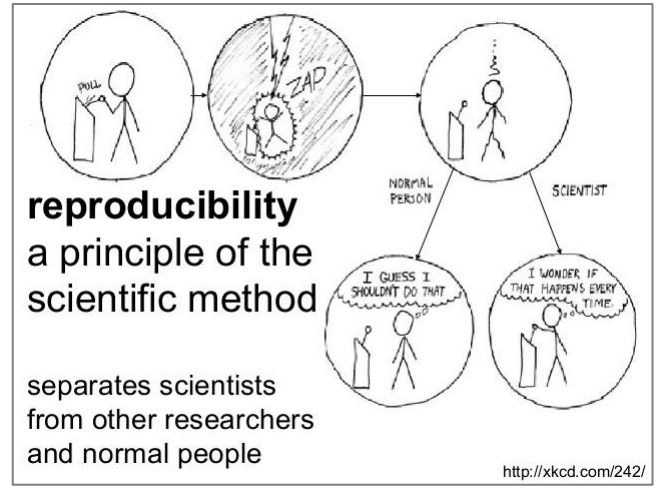
NATURE | EDITORIAL

Announcement: Reducing our irreproducibility

24 April 2013

PDF | Rights & Permissions

Over the past year, *Nature* has published a string of articles that highlight failures in the reliability and reproducibility of published research (collected and freely available at go.nature.com/huhbyr). The problems arise in laboratories, but journals such as this one compound them when they fail to exert sufficient scrutiny over the results that they publish, and when they do not publish enough information for other researchers to assess results properly.



reproducibility
a principle of the scientific method

separates scientists from other researchers and normal people

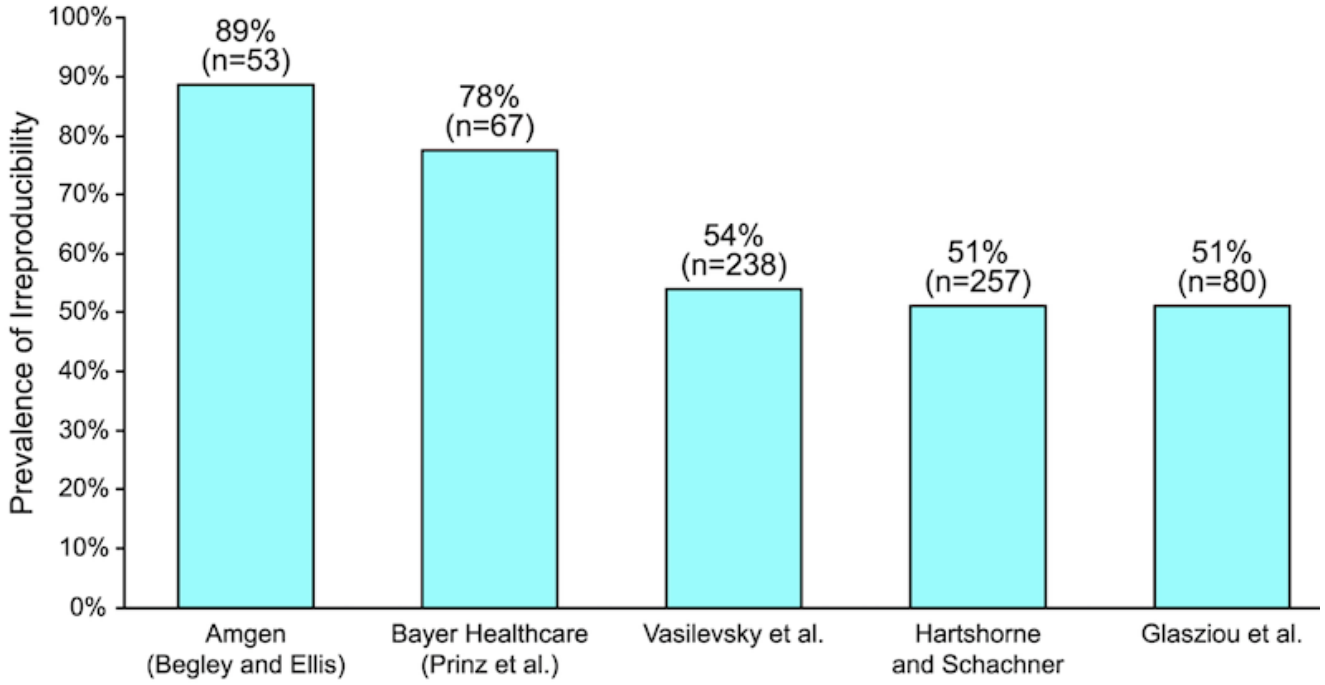
http://xkcd.com/242/

Scientific Reproducibility Crisis

Reproducibility crisis - the situation that a large percentage (between 51% and 89%) of the academic literature is not reproducible.

Reproducibility has also recently been found to be an issue in psychology and computer science.

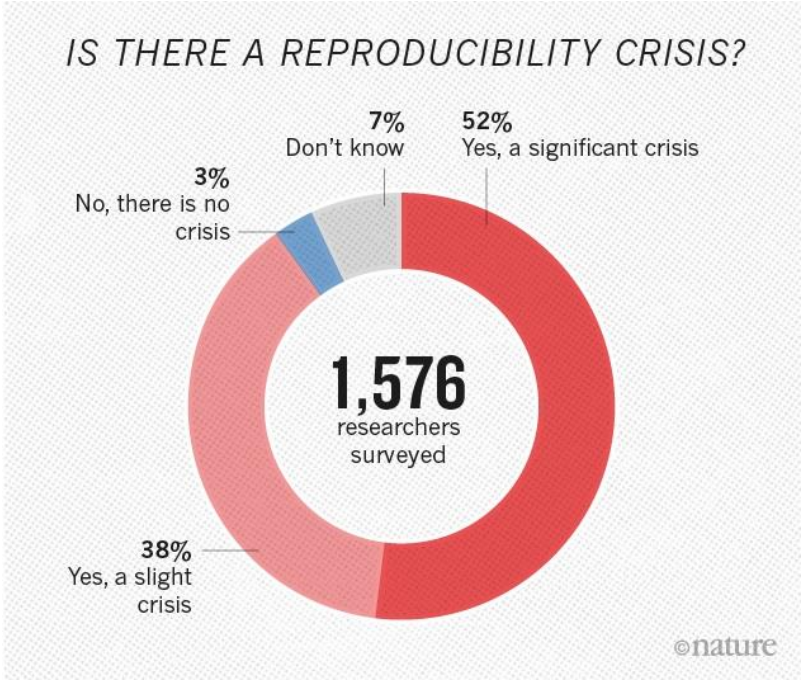
History of the reproducibility crisis



Reference: Leonard Freedman, Iain Cockburn, and Timothy Simcoe, "The Economics of Reproducibility in Preclinical Research." PLOS Biol 2015
http://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.1002165

Work highlighting the scale of the reproducibility crisis

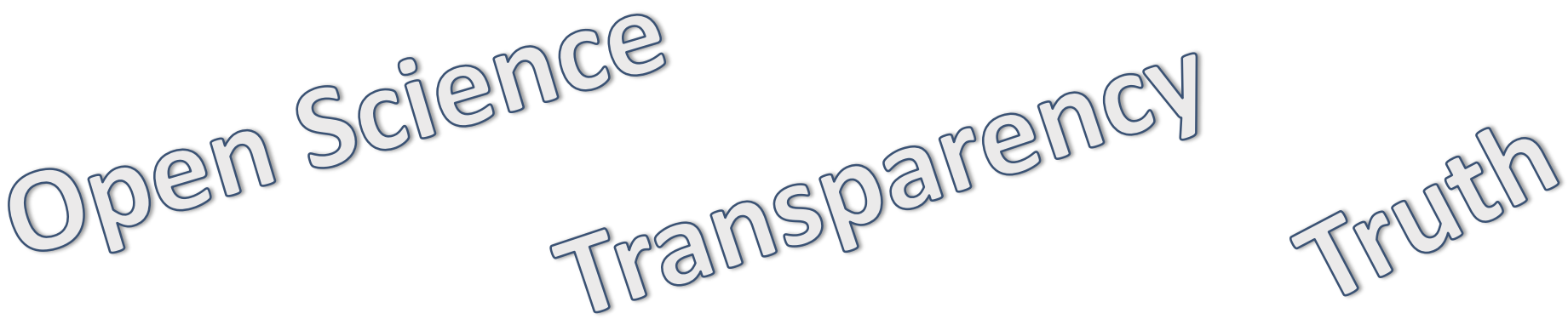
A project called "**The Reproducibility Initiative**" - focused on examining reproducibility levels in multiple scientific fields
http://validation.scienceexchange.com/#/reproducibility-initiative



Solution to Reproducibility Crisis

Open Research – scientific knowledge of all kinds should be openly shared as early as it is practical in the discovery process.

Therefore, need to reward the publication of research outputs along the entire process, NOT just each journal article as it is published.



Science

- Science is the systematic enterprise of gathering knowledge about the universe and organizing and condensing that knowledge into **testable** laws and theories.
- The success and credibility of science are anchored in the willingness of scientists to **independent testing** and **replication** by other scientists.
- Scientific evidence is strengthened when important findings are replicated by independent investigators, data, methods, laboratories, instruments, etc.
- This requires the **complete** and **open exchange of data**, **procedures** and **materials**.

Scientific publications have at least two goals:

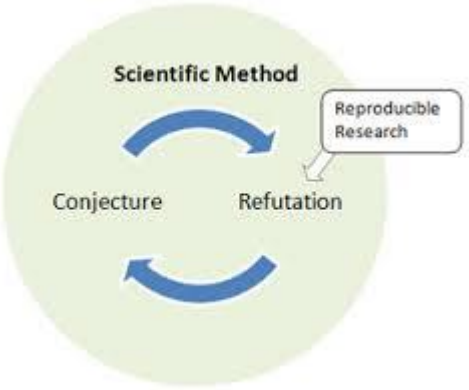
i. to announce a result.

ii. to convince readers that the result is correct.

Therefore

papers in experimental [and computational science] should describe the results and provide a clear enough protocol [or algorithm] to allow successful repetition and extension.

Jill Mesirov
Accessible Reproducible Research
Science 22 January 2010:
Vol. 327 no. 5964 pp. 415-416
DOI: 10.1126/science.1179653



Is there a reproducibility crisis in science? - Matt Anticole
https://ed.ted.com/lessons/is-there-a-reproducibility-crisis-in-science-matt-anticole

Putting the Science Back in Data Science

- One of key tenets of science is **reproducibility**.
- Truly “scientific” results can only be accepted by the community if they can be **clearly reproduced** and have undergone a **peer review process**.
- Information extraction from data is a science, and so the processes are expected to be **transparent**.
- With a little bit of **discipline** and the **right tooling**, data science teams can achieve reproducibility.

Steps in a Data Analysis

Arrange the steps in a logical order from 1 - 11

Clean the data

Interpret results

Challenge results

Create reproducible code

Obtain the data

Determine what data you can access

Define the question

Synthesize / write up results

Statistical prediction / modeling

Exploratory data analysis

Define the ideal data set

Structure of Data Analysis

A suggested structure for data analysis has the following components:

- A **good question** (needs narrowing down, so that data that is required is not huge)
- What is the **ideal dataset**?
  - Solving a descriptive problem – use a sample of the entire population.
  - Inference –use smaller dataset to make a conclusion on a larger population.
  - Prediction – use a test and training set.
  - Casual analysis – use experimental data.
- The **real dataset**
  - Keep a reference to where it is sourced from.
- Cleaning the data
  - Understand how data was acquired, record down all cleaning steps.

Structure of Data Analysis (continue)

- Exploratory data analysis
  - Use the head function to display the first few lines of your data, pairwise correlations, hierarchical cluster analysis
- Statistical modelling and prediction
  - Fit data to a generalized statistical model
- Interpreting the results
  - Use appropriate language, key words include : correlation, association, prediction
- Challenge the findings
  - Be self-analytical or critical of the work done
- Synthesize the results
  - Telling a good story, steps need not be hierarchical
- Reproducible results
  - Preserve code and documentation using knitr

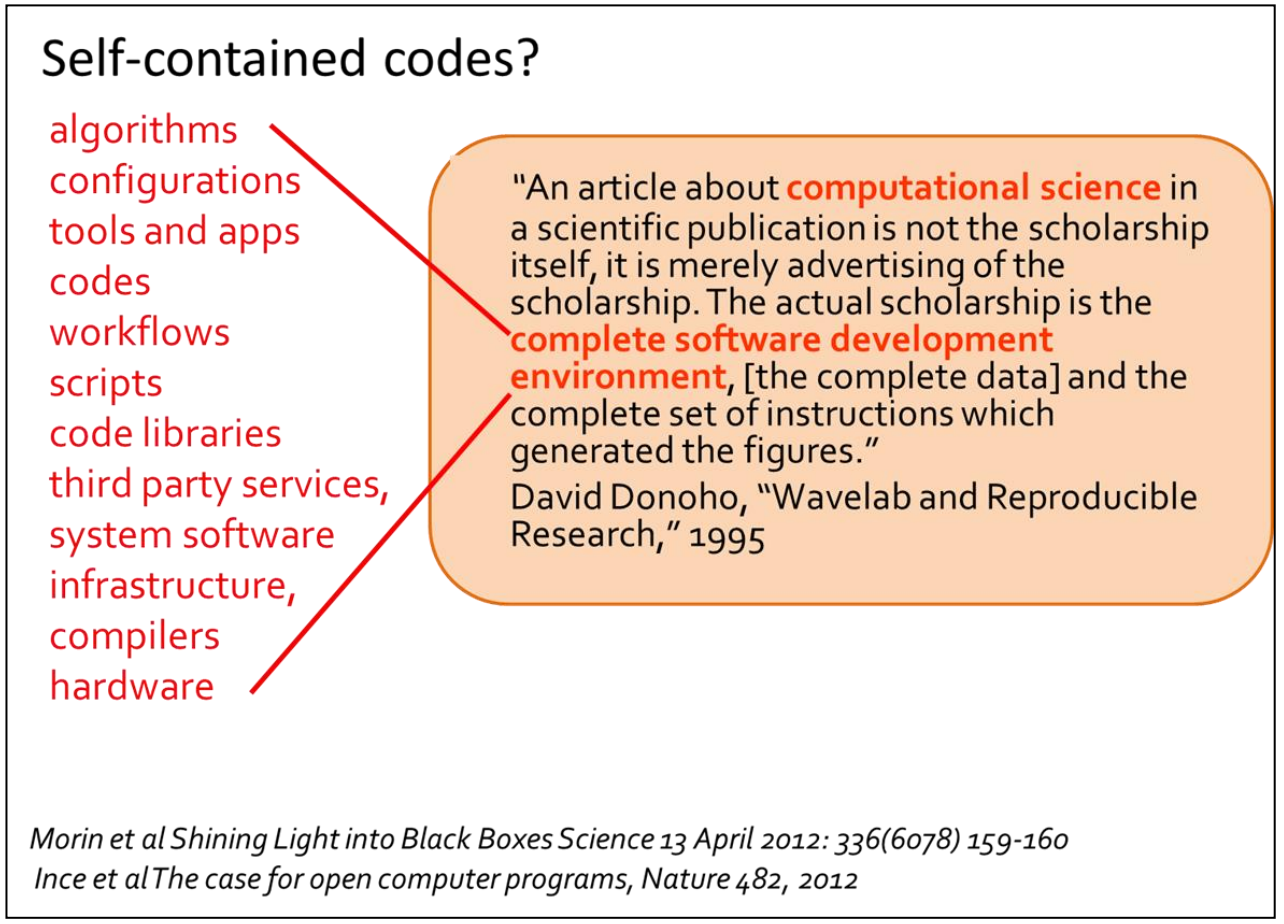
What is Reproducible Research (RR)?

A study is said to be reproducible [Roger Peng], if

(1) the analytic data are made available;
(2) the statistical methods are fully described and computer code is made available;
(3) documentation for both data and methods made available; and
(4) standard methods of distribution are employed.

RR is about communicating what you have exactly to others.

Reproducibility is critical for studies where the likelihood for full replication within a relevant time window is very low.



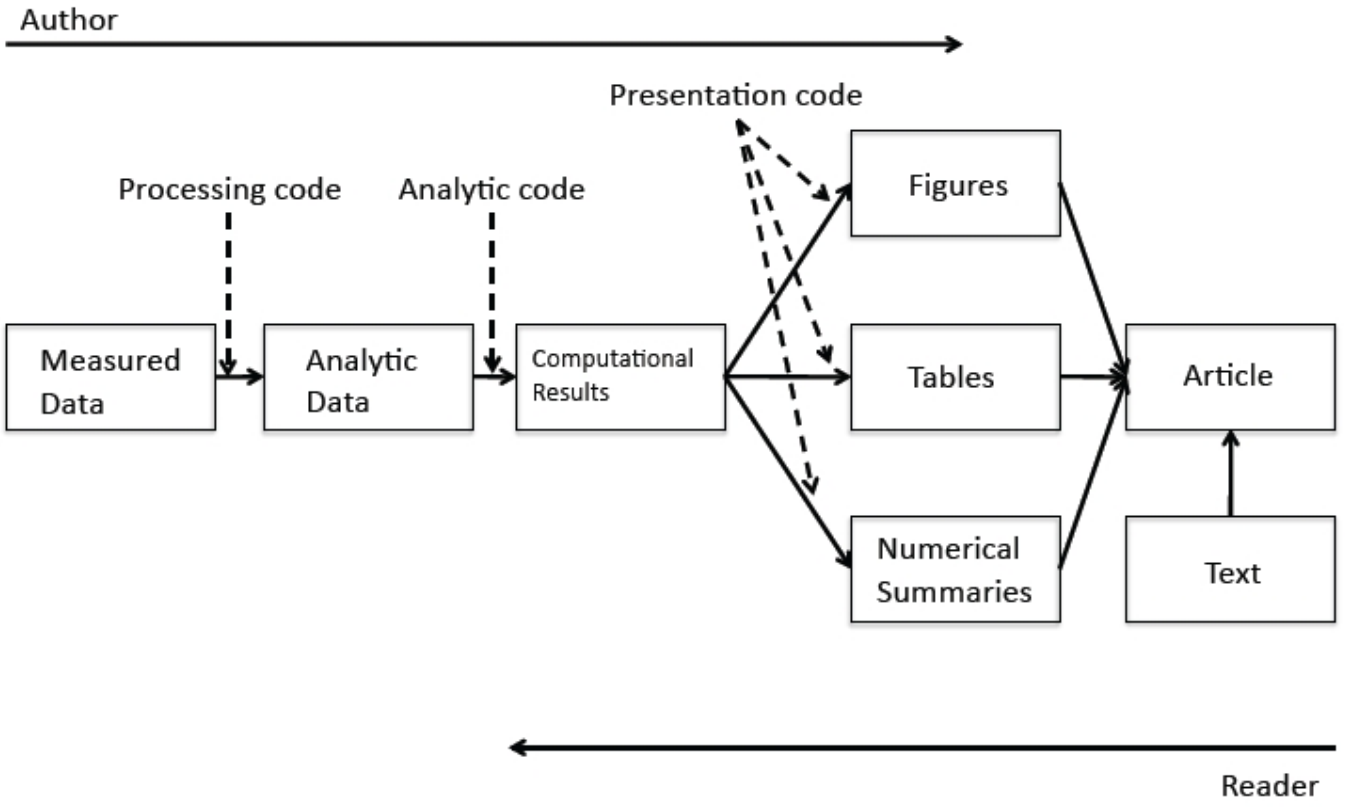
Computational reproducibility - all details of computation (code and data) are made routinely available to others.

What is Reproducible Research?

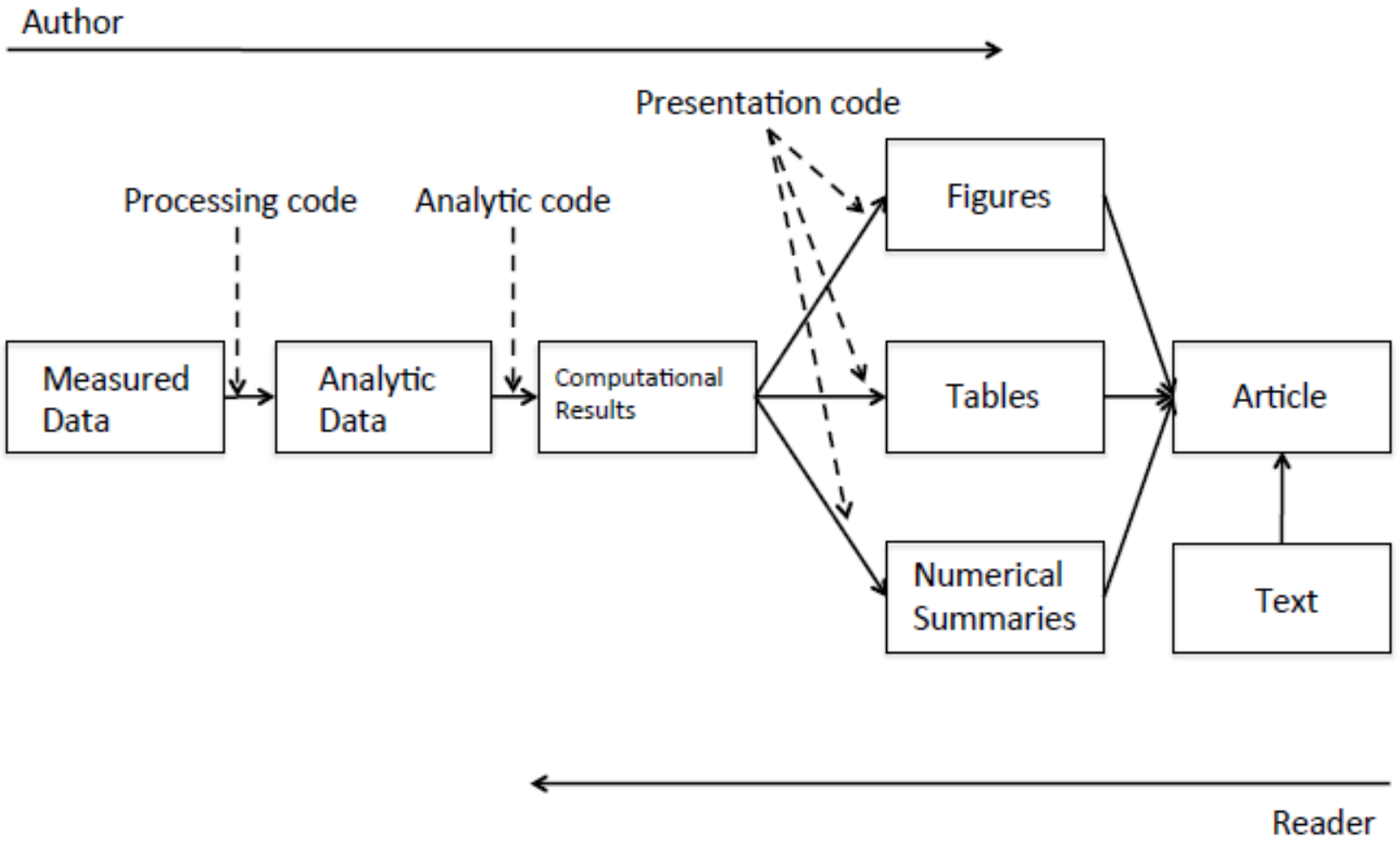
Reproducible research is the ultimate **standard** for strengthening scientific evidence by independent:

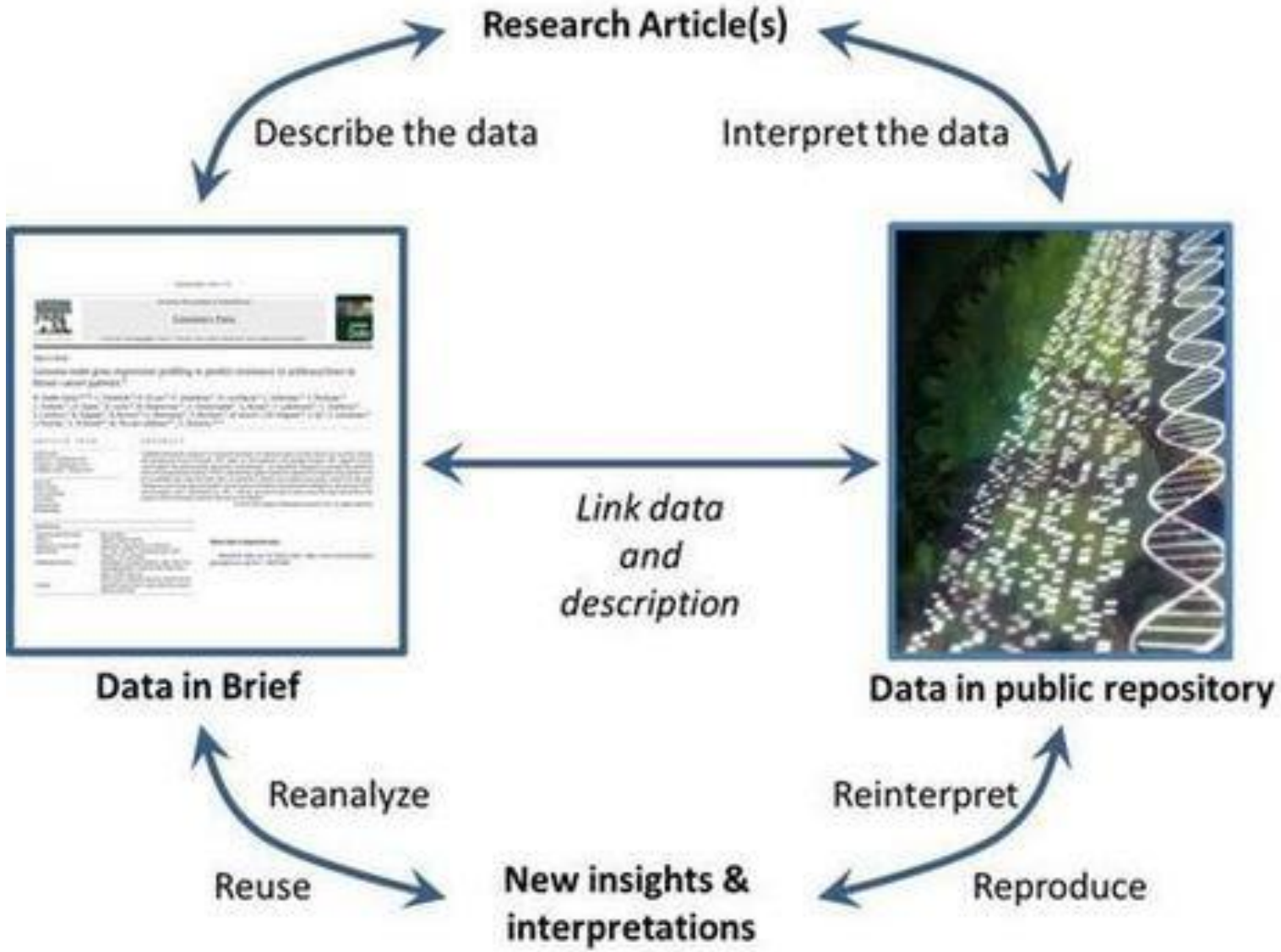
- Investigators
- Data
- Analytical methods
- Laboratories
- Instruments

Goal is to expose the reader to more of the research workflow.



Adopting RR in your data science project




Research Article(s)
Describe the data
Interpret the data
Link data and description
Data in Brief
Data in public repository
Reanalyze
Reuse
New insights & interpretations
Reinterpret
Reproduce

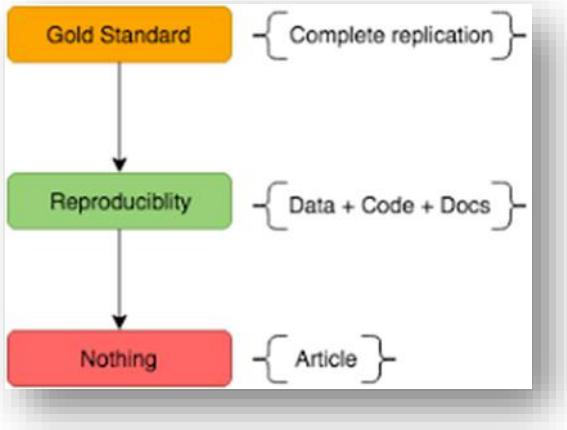
Replication vs. Reproducibility

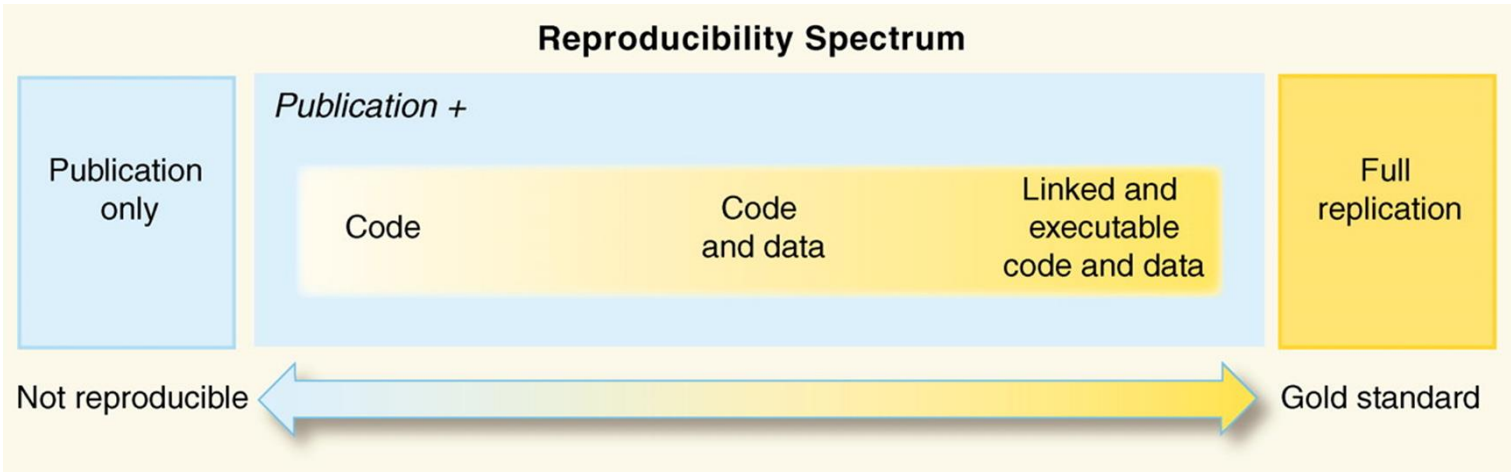
- **Replication**: The confirmation of results and conclusions from one study obtained independently in another is considered the scientific gold standard.

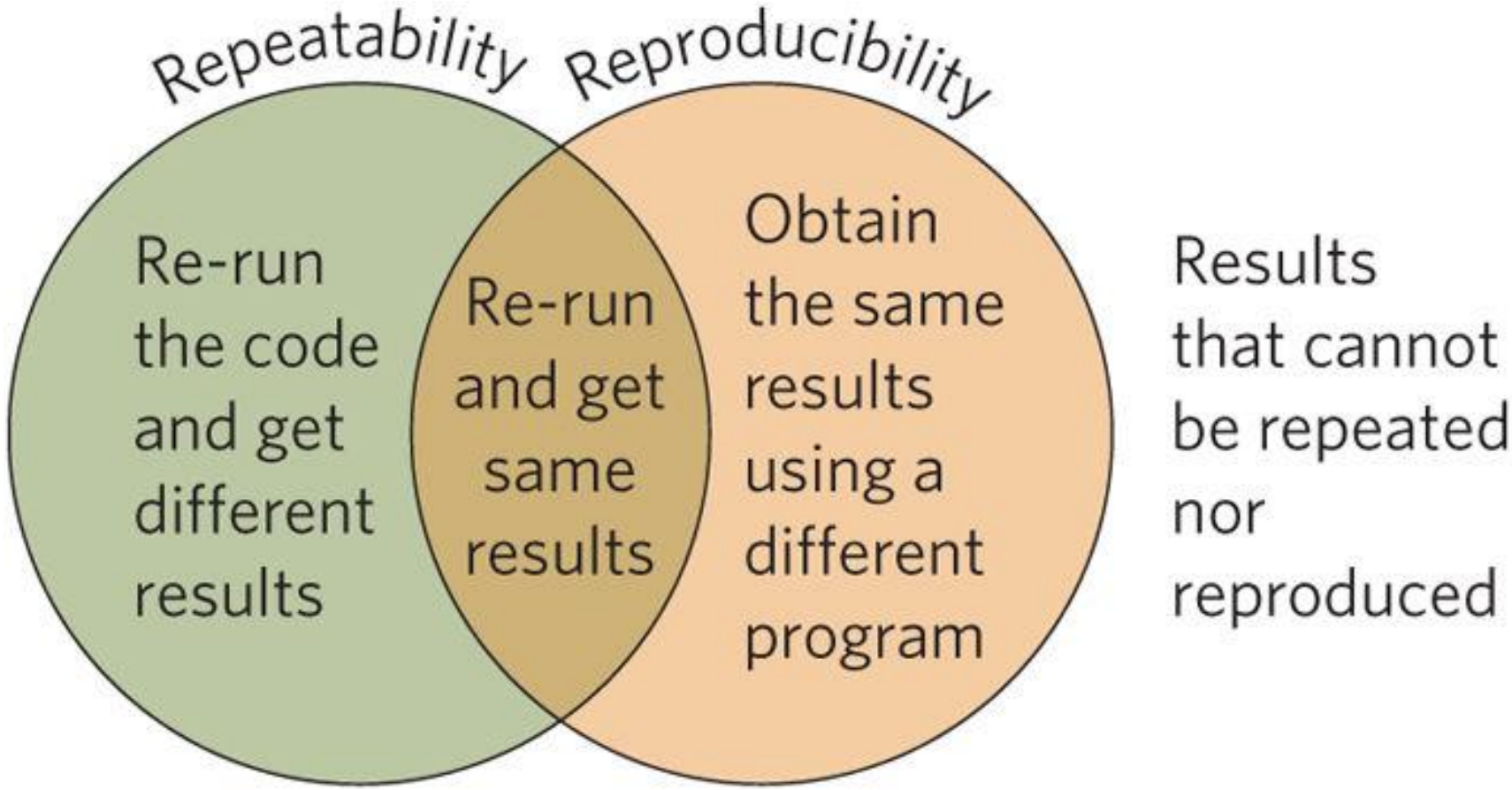
"Again, and Again, and Again …" **BR Jasny et. al.** Science, 2011. 334(6060) pp. 1225 DOI: 10.1126/science.334.6060.1225

- **Some studies can't be replicated**: too big, too costly, too time consuming, one time event, rare samples.
- **Reproducibility**: minimum standard for assessing the value of scientific claims, particularly when full independent replication of a study is not feasible.

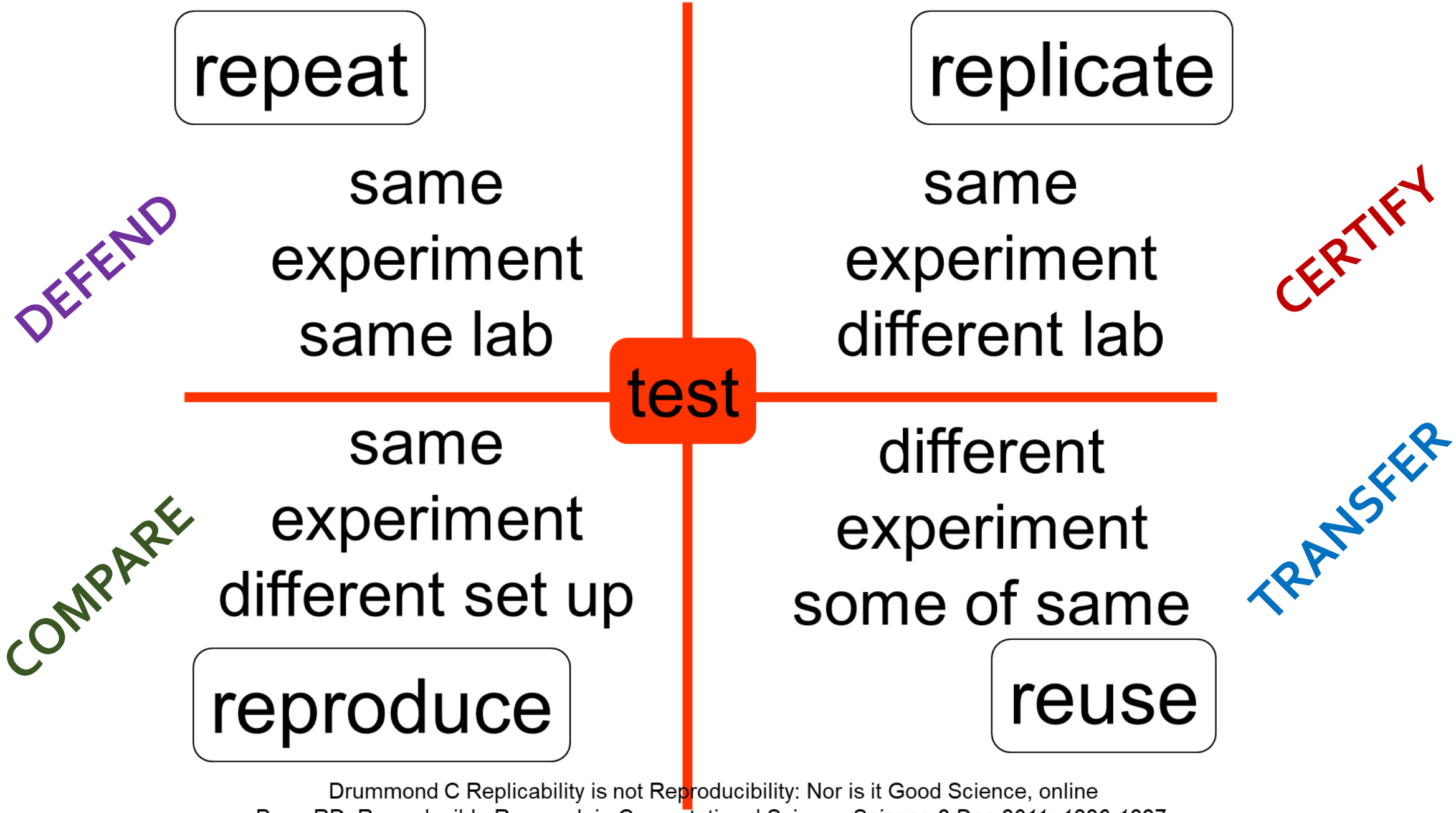
"Reproducible Research in Computational Science". **RD Peng** Science, 2011. 334 (6060) pp. 1226-1227 DOI: 10.1126/science.1213847







Source: https://www.nature.com/articles/ngeo2283#f1



Drummond C Replicability is not Reproducibility: Nor is it Good Science, online
Peng RD, Reproducible Research in Computational Science *Science 2 Dec 2011: 1226-1227.*

Replication vs. Reproducibility

Replicability ⟷ **Reproducibility**

Reproduction of the original results using the same tools.

By the original author on the same machine.

By someone in the same lab / using different machine.

By someone in a different lab.

Reproduction using different software, but with access to the original code

Completely independent reproduction based on only text description, without access to the original code

Degrees of Reproducibility

0 - The results cannot be reproduced by an independent researcher.

1 - The results cannot seem to be reproduced by an independent researcher.

2 - The results could be reproduced by an independent researcher, requiring extreme effort.

3 - The results can be reproduced by an independent researcher, requiring considerable effort.

4 - The results can be easily reproduced by an independent researcher with at most 15 minutes of user effort, requiring some proprietary source packages (MATLAB, SAS, etc.)

5 - The results can be easily reproduced by an independent researcher with at most 15 min of user effort, requiring only standard, freely available tools (C compiler, R, Python, etc.)

Reference: P. Vandewalle, J. Kovacevic and M. Vetterli, "Reproducible Research in Signal Processing - What, Why, and How," IEEE Signal Processing Magazine, 2009
https://infoscience.epfl.ch/record/136640/files/VandewalleKV09.pdf

Terminology

Reviewable Research - The descriptions of the research methods can be independently assessed and the results judged credible. (This includes both traditional peer review and community review, and does not necessarily imply reproducibility.)

Replicable Research - Tools are made available that would allow one to duplicate the results of the research, for example by running the authors' code to produce the plots shown in the publication. (Here tools might be limited in scope, e.g., only essential data or executables, and might only be made available to referees or only upon request.)

Confirmable Research - The main conclusions of the research can be attained independently without the use of software provided by the author. (But using the complete description of algorithms and methodology provided in the publication and any supplementary materials.)

Auditable Research - Sufficient records (including data and software) have been archived so that the research can be defended later if necessary or differences between independent confirmations resolved. The archive might be private, as with traditional laboratory notebooks.

Open or Reproducible Research - Auditable research made openly available. This comprised well-documented and fully open code and data that are publicly available that would allow one to (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

Source: https://icerm.brown.edu/tw12-5-rcem/icerm_report.pdf

Tracing Origin of Reproducible Research

- **Robert Boyle** (1660) – a discovery should be reproducible before being accepted as scientific knowledge.
- **Donald Knuth** (1984) – Literate programming.
- **Jon Claerbout** (1990) – Coined the term RR

Introduced these practices in his Stanford Exploration Project (SEP) lab working on geophysics research.

- **Buckheit and Donoho** (1995) at Stanford
  - Applied RR to wavelet research using MATLAB.
  - Developed the WaveLab toolbox to reproduce the results from a set of their publications.

"Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do."

Donald E. Knuth. Literate Programming. The Computer Journal, 27(2):97-l11, May 1984.

J. Claerbout, "Electronic documents give reproducible research a new meaning," in Proc. 62nd Ann. Int. Meeting of the Soc. of Exploration Geophysics,1992, pp. 601–604.

J. B. Buckheit and D. L. Donoho. (1995). "WaveLab and reproducible research," Dept. of Statistics, Stanford Univ., Tech. Rep. 474.

Why should data science team care about reproducibility?

Collaboration:

- Data science, and science in general for that matter, is a collaborative endeavor.
- No data scientist knows all relevant modeling techniques and analyses
- A data scientist, you should always be concerned about how you share your results with your colleagues and how you collaborate on analyses/models.
- Specifically, you should share your work and deploy your products in a way that allows others to do exactly what you did, with the same data you used, to produce the same result.
- Otherwise, your team will not be able to capitalize on its collective knowledge, and advances within the team will only be advanced and understood by individuals.

https://www.oreilly.com/ideas/putting-the-science-back-in-data-science

Why should data science team care about reproducibility?

Creativity:

- How do you know if a new model is performing better than an old model? How can you properly justify adding creative sophistication or complexity to analyses?
- If analyses are reproducible, data science teams can: (1) concretely determine how new analyses compare to old analyses because the old analyses can be exactly reproduced and the new analyses can be run against the known previous data; and (2) clearly see which analyses performed poorly in the past to avoid repeating mistakes.

Why should data science team care about reproducibility?

Compliance:

- As more and more statistical, machine learning, and artificial intelligence applications make decisions that directly impact users, there will be more and more public pressure to explain and reproduce results.
- The EU is already demanding a “right to an explanation” for many algorithmically generated, user-impacting decisions.
- How could such an explanation be given or an audit trail be established without a clearly understood and reproducible workflow that let to the results?

How can a data science team achieve reproducibility?

- Strive for and celebrate simple, interpretable solutions.
- No reproducibility, no deployment
  - Data scientists are working to create a culture of data-driven decision-making. If the application breaks without an explanation (likely because unable to reproduce the results), people will lose confidence in the application and stop making decisions based on the results of the application.
- Version your data
- Know your provenance
- Write it down

Data provenance is the documentation of where a piece of data comes from and the processes and methodology by which it was produced.

https://www.oreilly.com/ideas/putting-the-science-back-in-data-science

Literate Programming

by Donald E. Knuth (Stanford, California: Center for the Study of Language and Information, 1992), xvi+368pp.
(CSLI Lecture Notes, no. 27.)
ISBN 0-937073-80-6
Japanese translation by Makoto Arisawa, *Bungeiteki Programming* (Tokyo: ASCII Corporation, 1994), 463pp.

Literate programming is a methodology that combines a programming language with a documentation language, thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language. The main idea is to treat a program as a piece of literature, addressed to human beings rather than to a computer. The program is also viewed as a hypertext document, rather like the World Wide Web. (Indeed, I used the word WEB for this purpose long before CERN grabbed it!) This book is an anthology of essays including my early papers on related topics such as structured programming, as well as the article in *The Computer Journal* that launched Literate Programming itself. The articles have been revised, extended, and brought up to date.

Evolution of Literate Programming

HTML - HyperText Markup Language, used to create web pages. Developed in 1993

LaTeX – a typesetting system for production of technical/scientific documentation, PDF output. Developed in 1994

Sweave – a tool that allows embedding of the R code in LaTeX documents, PDF output. Developed in 2002

reStructuredText - a markup syntax that plays well with Python and Sphinx documentation. Developed in 2002

Markdown – a lightweight markup language for plain text formatting syntax. Easily converted to HTML. Developed in 2004

Literate Programming

- Literate programming can be a useful way to put text, code, data, output all in one document.
- Original idea comes from **Don Knuth**.
- An article is a stream of **text** and **code**.
- Analysis code is divided into text and code "**chunks**".
- Presentation code formats results (tables, figures, etc.)
- Article text explains what is going on.

Literate Programming

- Literate programs are **weaved** to produce human--readable documents and **tangled** to produce machine--readable documents.
- Literate programming is a general concept. We need:
  - ✓A documentation language
  - ✓A programming language
- The original **Sweave** system developed by Friedrich Leisch used LaTeX and R.
- **Knitr** supports a variety of documentation languages.

Literate Programming: Pros

The Pros

- Text and code all in one place, logical order.
- Data, results automatically updated to reflect external changes.
- Code is live------automatic “regression test” when building a document

The Cons

- Text and code all in one place; can make documents difficult to read, especially if there is a lot of code.
- Can substantially slow down processing of documents (although there are tools to help).

How Do I Make My Work Reproducible?

- Decide to do it (ideally from the start)
- Keep track of things, perhaps with a version control system to track snapshots/changes
- Use software whose operation can be coded
- Don't save output
- Save data in non-proprietary formats

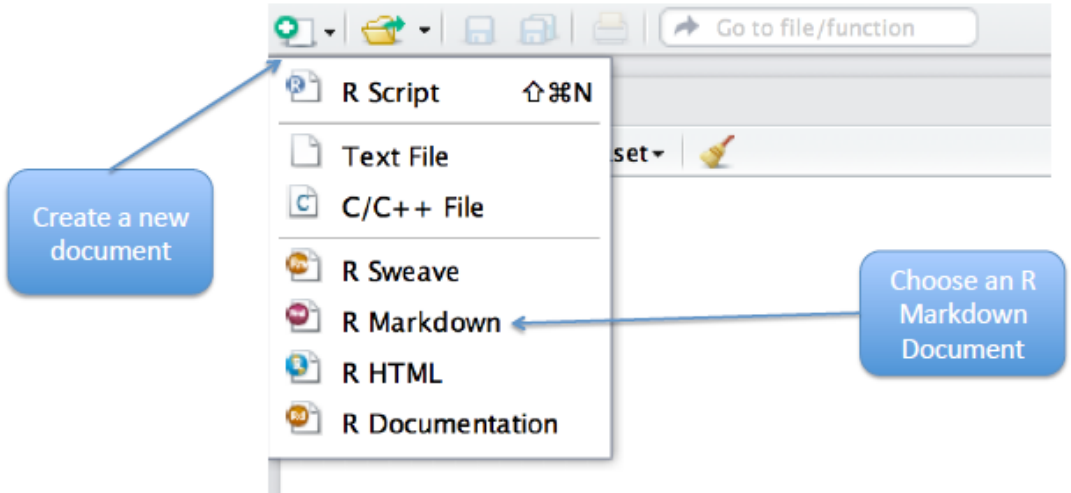
The most important is the mindset, when starting, that the end product will be reproducible.

– Keith Baggerly

What is knitr?

- An R package written by Yihui Xie (while he was a grad student at Iowa State)
  - Available on CRAN
- Supports RMarkdown, LaTeX, and HTML as documentation languages
- Can export to PDF, HTML
- Built right into RStudio for your convenience

My First knitr Document



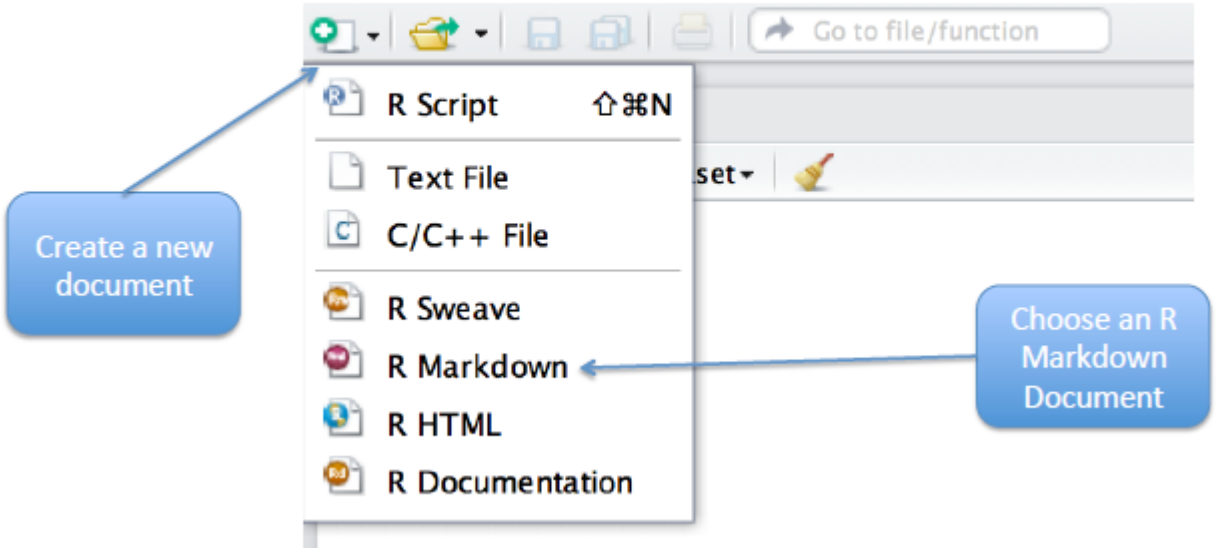
What is knitr Good For?

- Manuals
- Short/medium-length technical documents
- Tutorials
- Reports (esp. if generated periodically)
- Data preprocessing documents/summaries

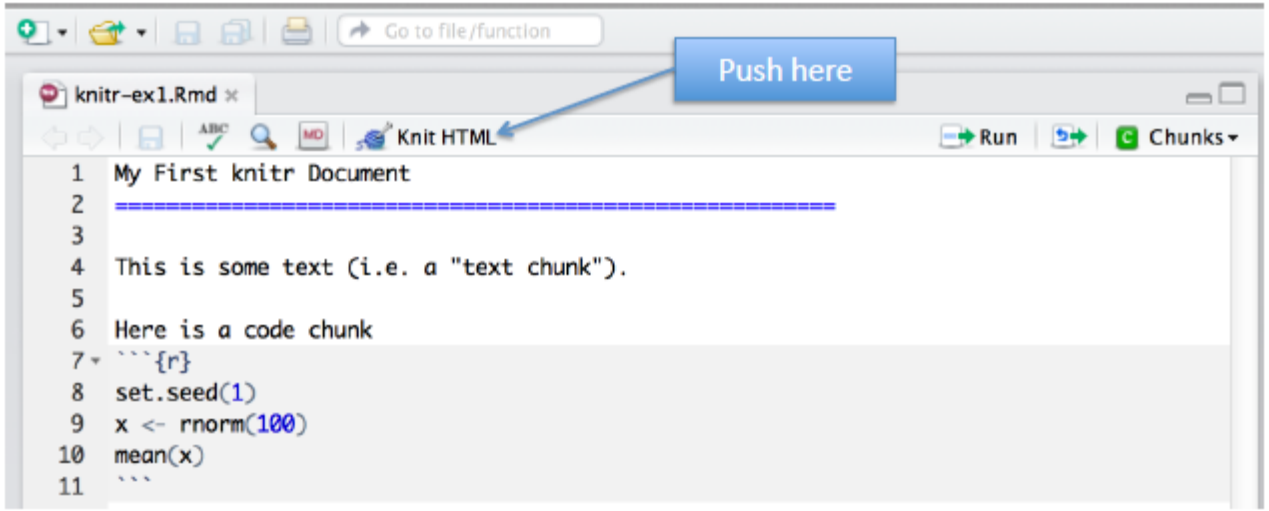
What is knitr NOT Good For?

- Very long research articles
- Complex time-consuming computations
- Documents that require precise formatting

My First knitr Document



Processing a knitr Document




```
1  My First knitr Document
2  ==========
3
4  This is some text (i.e. a "text chunk").
5
6  Here is a code chunk
7  ```{r}
8  set.seed(1)
9  x <- rnorm(100)
10 mean(x)
11 ```
```

Start of code chunk

End of code chunk

HTML Output

My First knitr Document

This is some text (i.e. a "text chunk").

Here is a code chunk

```
set.seed(1)
x <- rnorm(100)
mean(x)
```

Code input

```
## [1] 0.1089
```

Numerical output

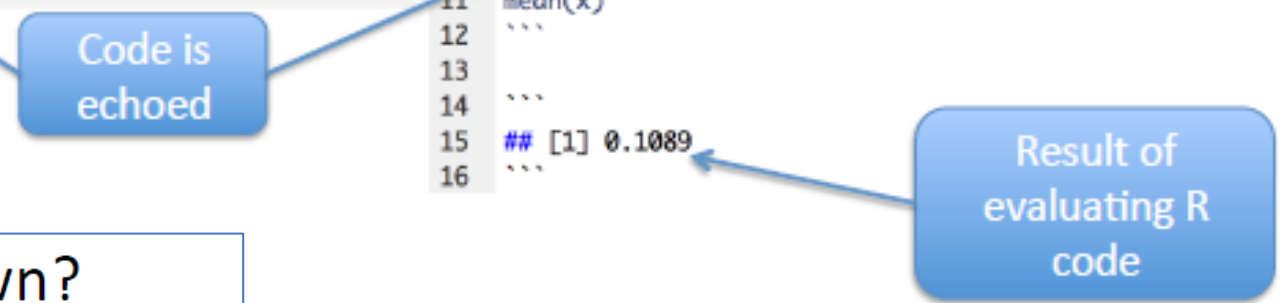
What knitr Produces: Markdown

RMarkdown Document

```
1  My First knitr Document
2  ==========================================================
3
4  This is some text (i.e. a "text chunk").
5
6  Here is a code chunk
7  ```{r}
8  set.seed(1)
9  x <- rnorm(100)
10 mean(x)
11 ```
```

Markdown Document (generated)

```
1  My First knitr Document
2  ==========================================================
3
4  This is some text (i.e. a "text chunk").
5
6  Here is a code chunk
7
8  ```r
9  set.seed(1)
10 x <- rnorm(100)
11 mean(x)
12 ```
13
14 ```
15 ## [1] 0.1089
16 ```
```



What is Markdown?

- A simplified version of “markup” languages
- No special editor required
- Simple, intuitive formatting elements
- Complete information available at http://goo.gl/MUt9i5

Processing of knitr Documents (what happens under the hood)

- You write the RMarkdown document (.Rmd)
- knitr produces a Markdown document (.md)
- knitr converts the Markdown document into HTML (by default)
- .Rmd → .md → .html
- You should NOT edit (or save) the .md or .html documents until you are finished

Another Example

```
# My First knitr Document
Roger D. Peng

## Introduction

This is some text (i.e. a "text chunk"). Here is a code chunk.
```{r simulation,echo=FALSE}
set.seed(1)
x <- rnorm(100)
mean(x)
```
```

Level 1 heading

Level 2 heading

Do not echo code

Output

My First knitr Document

Roger D. Peng

Introduction

This is some text (i.e. a “text chunk”). Here is a code chunk.

```
## [1] 0.1089
```

Hiding Results

````
# My First knitr Document
Roger D. Peng

## Introduction

This is some text (i.e. a "text chunk"). Here is a code chunk but it doesn't print
anything!
```{r simulation,echo=FALSE,results="hide"}
set.seed(1)
x <- rnorm(100)
mean(x)
```
````

Output

My First knitr Document

Roger D. Peng

Introduction

This is some text (i.e. a “text chunk”). Here is a code chunk but it doesn't print anything!

Inline Text Computations

```
# My First knitr Document

## Introduction

```{r computetime,echo=FALSE}
time <- format(Sys.time(), "%a %b %d %X %Y")
rand <- rnorm(1)
```

The current time is `r time`. My favorite random number is `r rand`.
```

Inline Text Computations

My First knitr Document

Introduction

The current time is Wed Sep 04 16:42:09 2013. My favorite random number is 1.1829.

Incorporating Graphics

```
## Introduction

Let's first simulate some data.
```{r simulatedata,echo=TRUE}
x <- rnorm(100); y <- x + rnorm(100, sd = 0.5)
```

Here is a scatterplot of the data.
```{r scatterplot,fig.height=4}
par(mar = c(5, 4, 1, 1), las = 1)
plot(x, y, main = "My Simulated Data")
```
```

Adjust figure height

What knitr Produces in HTML

```
<body>
<h2>Introduction</h2>

<p>Let&#39;s first simulate some data.</p>

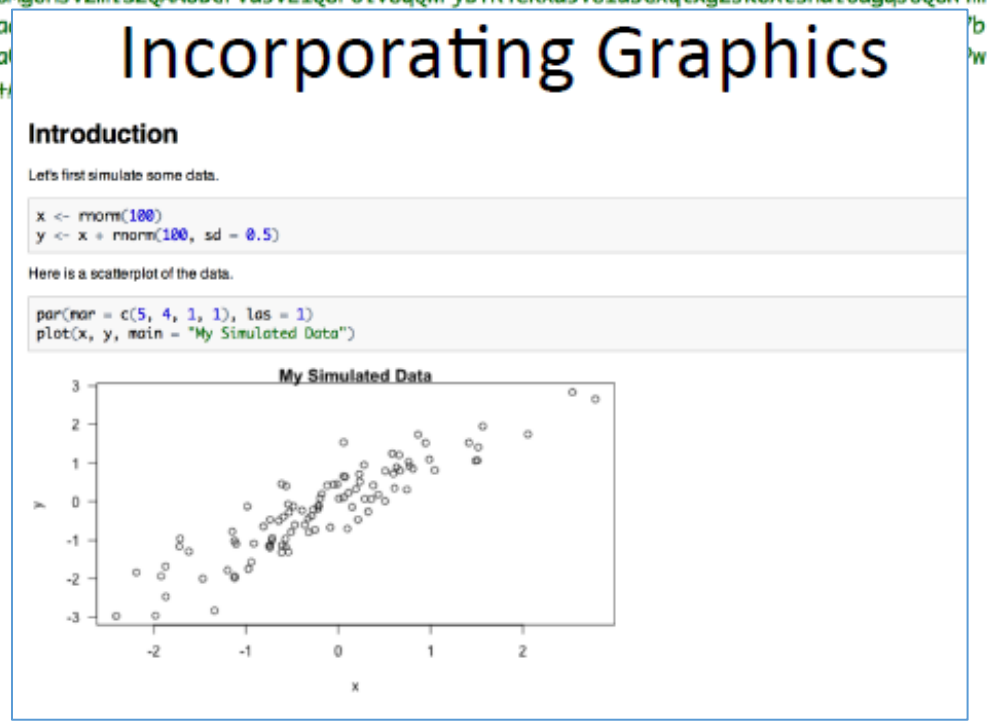
<pre><code class="r">x <- rnorm(100)
y <- x + rnorm(100, sd = 0.5)
</code></pre>

<p>Here is a scatterplot of the data.</p>

<pre><code class="r">par(mar = c(5, 4, 1, 1), las = 1)
plot(x, y, main = &quot;My Simulated Data&quot;)
</code></pre>

<p><img src="
lDQ1BJQ0MgUHJvZmlsZQAAOBGFVd9v21QUPolvUqQWPyBYR4eKxa9VU1u5GxqtxgZJk6XtShal6dgqJOQ6N4m
pGwfb6ba
RD1fabWa
G4VHX0Z+
```

Image is embedded in HTML



Making Tables with xtable

````
## Introduction

```{r fitmodel}
library(datasets)
data(airquality)
fit <- lm(Ozone ~ Wind + Temp + Solar.R, data = airquality)
```

Here is a table of regression coefficients.
```{r showtable,results="asis"}
library(xtable)
xt <- xtable(summary(fit))
print(xt, type = "html")
```
````

Making Tables with xtable

Introduction

```
library(datasets)
data(airquality)
fit <- lm(Ozone ~ Wind + Temp + Solar.R, data = airquality)
```

Here is a table of regression coefficients.

```
library(xtable)
xt <- xtable(summary(fit))
print(xt, type = "html")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-64.3421	23.0547	-2.79	0.0062
Wind	-3.3336	0.6544	-5.09	0.0000
Temp	1.6521	0.2535	6.52	0.0000
Solar.R	0.0598	0.0232	2.58	0.0112

Setting Global Options

- Sometimes we want to set options for **every** code chunk that are different from the defaults
- For example, we may want to suppress all code echoing and results output
- We have to write some code to set these global options

Setting Global Options

```
## Introduction

```{r setoptions,echo=FALSE}
opts_chunk$set(echo = FALSE, results = "hide")
```

First simulate data
```{r simulatedata,echo=TRUE}
x <- rnorm(100); y <- x + rnorm(100, sd = 0.5)
```

Here is a scatterplot of the data.

```{r scatterplot,fig.height=4}
par(mar = c(5, 4, 1, 1), las = 1)
plot(x, y, main = "My Simulated Data")
```
```

Set default to NOT echo code

Override default

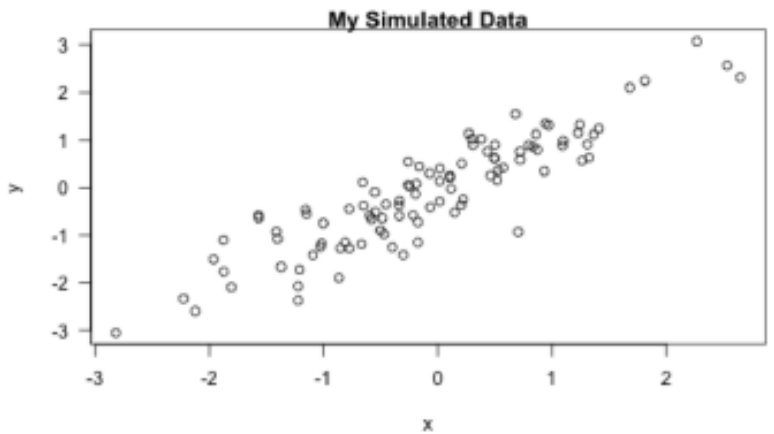
Don't echo code here

Introduction

First simulate data

```
x <- rnorm(100)
y <- x + rnorm(100, sd = 0.5)
```

Here is a scatterplot of the data.



How to Implement Reproducible Research?

Significant **infrastructure** for conducting and distributing reproducible research:

- Tools for researchers, developers
  - Software tools
- Repositories for datasets
  - Servers,
  - Websites
- Rights framework for datasets
  - Protocols, and
  - Standards for exchanging data, methods,
- Other necessary information.

Reproducible Research 101

- Begin with the final product in mind
- Use literate programming (self-documenting code)
- Keep history of changes via code versioning and sharing
- Get basic statistics right
- Set stringent cutoffs, correct p-values for multiple testing
- Be critical, consider batch effects, visualize, do sanity checks, use random controls, cross-validation
- Follow reporting guidelines.

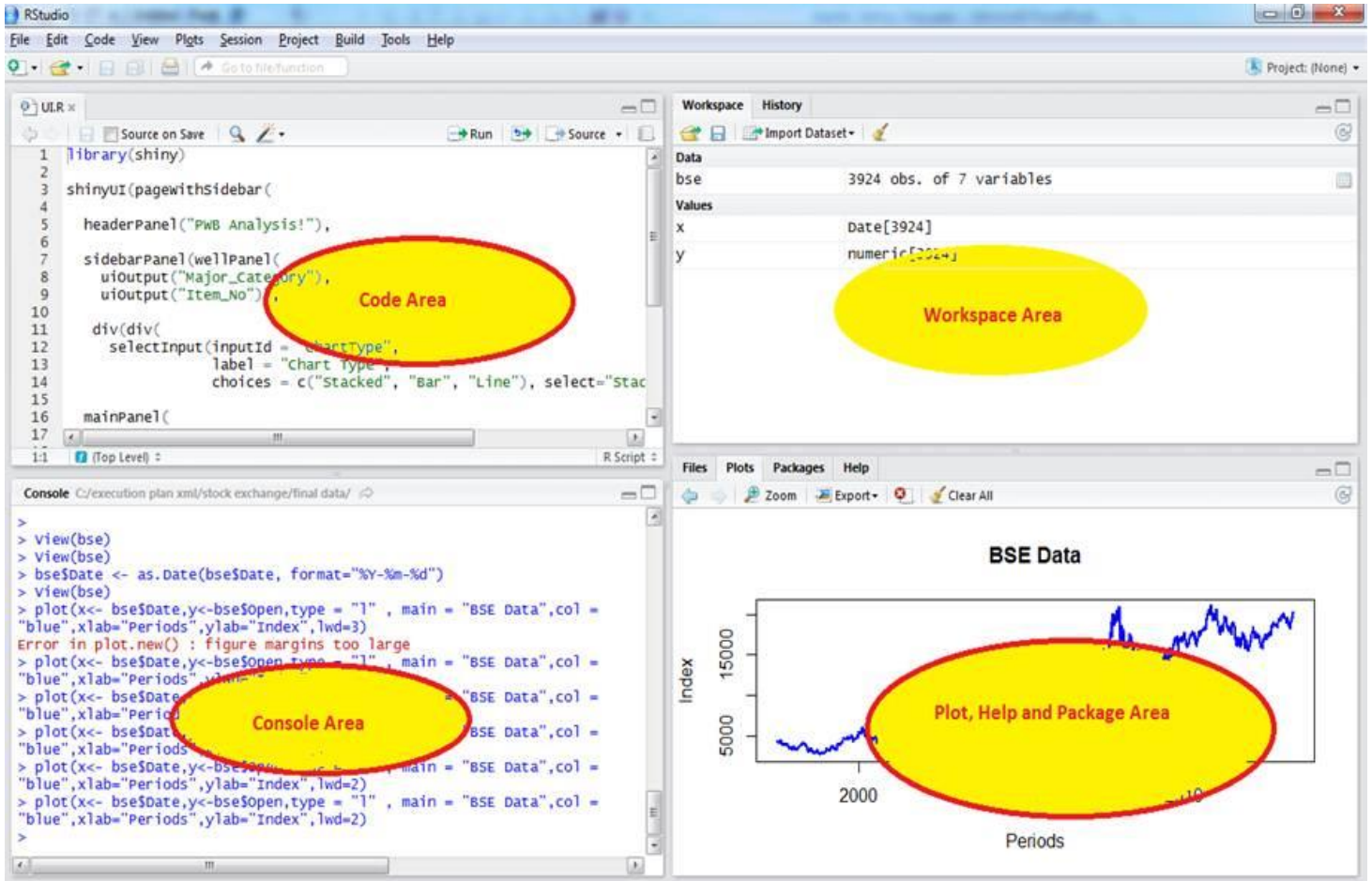
Tools to Aid in Reproducible Research

(1) Literate programming, authoring, and publishing tools.

- Enable users to write and publish documents that integrate the text and figures seen in reports with code and data used to generate both text and graphical results.
- Tools that enable literate programming include both
  - programming language-specific tools such as WEB, Sweave, and knitR
  - programming-language independent tools such as Dexy, Lepton, and noweb.
- Other authoring environments include SHARE, Doxygen, Sphinx, CWEB, and the Collage Authoring Environment.


Welcome to RStudio
Software, education, and services for the R community


Shiny
RStudio
ggplot2
dplyr
rmarkdown
knitr
lubridate
stringr
devtools
tidyr
%>%
packrat
ggvis
readr


RStudio
File Edit Code View Plots Session Project Build Tools Help
Project: (None)
ui.R
Source on Save
Run
Source
1 library(shiny)
3 shinyUI(pageWithSidebar(
5 headerPanel("PWB Analysis!"),
7 sidebarPanel(wellPanel(
8 uiOutput("Major_Category"),
9 uiOutput("Item_No"),
11 div(div(
12 selectInput(inputId = "ChartType",
13 label = "Chart Type",
14 choices = c("Stacked", "Bar", "Line"), select="Stac
16 mainPanel(
(Top Level)
R Script
Code Area
Workspace
History
Import Dataset
Data
bse 3924 obs. of 7 variables
Values
x Date[3924]
y numeric[3924]
Workspace Area
Console C:/execution plan xml/stock exchange/final data/
> view(bse)
> view(bse)
> bse$Date <- as.Date(bse$Date, format="%Y-%m-%d")
> view(bse)
> plot(x<- bse$Date,y<-bse$Open,type = "l" , main = "BSE Data",col = "blue",xlab="Periods",ylab="Index",lwd=3)
Error in plot.new() : figure margins too large
Console Area
Files Plots Packages Help
Zoom
Export
Clear All
BSE Data
Index
5000
15000
2000
Periods
Plot, Help and Package Area

Tools to Aid in Reproducible Research

(2) Tools that define and execute structured computation and track provenance

- Provenance refers to the tracking of chronology and origin of research objects, such as data, source code, figures, and results.
- Tools that record provenance of computations include VisTrails, Kepler, Taverna, Sumatra, Pegasus, Galaxy, Workflow4ever, and Madagascar.

Tools to Aid in Reproducible Research

(3) Integrated tools for version control and collaboration

- Tools that track and manage work as it evolves facilitate reproducibility among a group of collaborators.
- With the advent of version control systems (e.g., Git, Mercurial, SVN, CVS), it has become easier to track the investigation of new ideas, and collaborative version control sites like Github, Google Code, BitBucket, and Sourceforge enable such ideas to be more easily shared.

Tools to Aid in Reproducible Research

(4) Tools that express computations as notebooks

- These tools represent sequences of commands and calculations as an interactive worksheet with pretty printing and integrated displays, decoupling content (the data, calculations) from representation (PDF, HTML, shell console), so that the same research content can be presented in multiple ways.
- Examples include both closed-source tools such as MATLAB (through the publish and app features), Maple, and Mathematica, as well as open-source tools such as IPython, Sage, RStudio (with knitr), and TeXmacs.

Tools to Aid in Reproducible Research

(5) Tools that capture and preserve a software environment

- New tools make it possible to exactly capture the computational environment and pass it on to someone who wishes to reproduce a computation.
- VirtualBox, VMWare, or Vagrant can be used to construct a virtual machine image containing the environment.
- Application virtualization tools, such as CDE (Code, Data, and Environment), attach themselves to the computational process in order to find and capture software dependencies.
- Computational environments can also be constructed and made available in the cloud, using Amazon EC2, Wakari, RunMyCode.org and other tools.

Reproducibility Skills

- ✓version control and use of online repositories,
- ✓modern programming practice including unit testing and regression testing,
- ✓maintaining “notebooks” or “research compendia”,
- ✓recording the provenance of final results relative to code and/or data,
- ✓numerical / floating point reproducibility and nondeterminism,
- ✓reproducibility on parallel systems,
- ✓dealing with large datasets,
- ✓dealing with complicated software stacks and use of virtual machines,
- ✓documentation and literate programming,
- ✓IP and licensing issues, proper citation and attribution.

Learn more

Reference: https://icerm.brown.edu/tw12-5-rcem/icerm_report.pdf

A Hierarchy of Reproducibility

- **Good**: Use code. Minimize pointing and clicking (RStudio). Mention availability of code.
- **Better**: Use version control. Help yourself keep track of changes, fix bugs and improve project management (RStudio & Git & GitHub or BitBucket).
- **Best**: Use embedded narrative and code to explicitly link code, text and data, save yourself time, save reviewers time, improve your code. (RStudio & Git & GitHub or BitBucket & Rmarkdown & knitR & data repository).

What Problem Does Reproducibility Solve?

What we get:

- ✓Transparency
- ✓Data Availability
- ✓Software / Methods Availability
- ✓Improved Transfer of Knowledge

What we do not get:

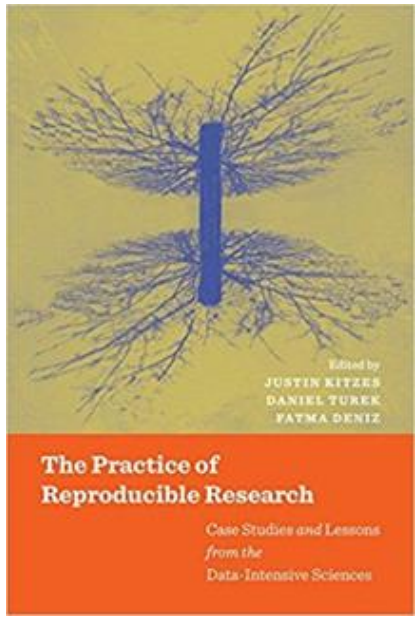
- ➢ Validity / Correctness of the analysis

An analysis can be reproducible and still be wrong.
We want to know “can we trust this analysis?”
Does requiring reproducibility deter bad analysis?

Reproducibility Enhancement Principles

1. **To facilitate reproducibility**, share the data, software, workflows, and details of the computational environment in open repositories.
2. **To enable discoverability**, persistent links should appear in the published article and include a permanent identifier for data, code, and digital artifacts upon which the results depend.
3. **To enable credit for shared digital scholarly objects**, citation should be standard practice.
4. **To facilitate reuse**, adequately document digital scholarly artifacts.
5. Journals should conduct a **Reproducibility Check** as part of the publication process and enact the TOP Standards at level 2 or 3.
6. Use **Open Licensing** when publishing digital scholarly objects.
7. Funding agencies should instigate **new research programs** and pilot studies

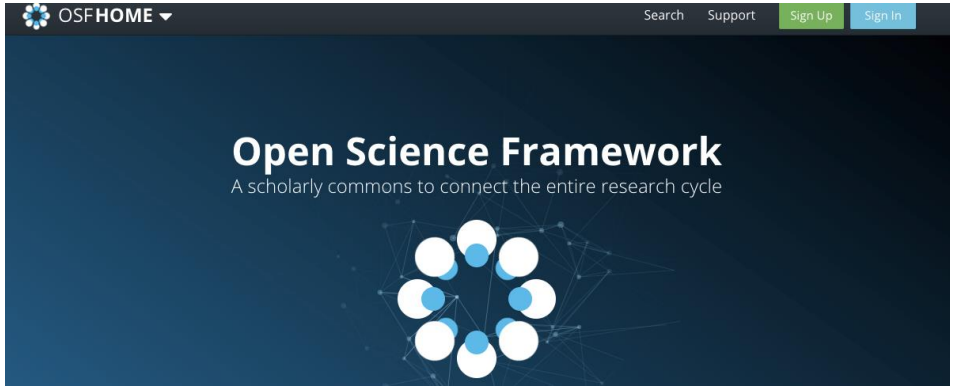
References



Site about reproducible research

http://reproducibleresearch.net/
http://reproducibleresearch.net/links/

https://www.plos.org/publications



https://osf.io/

Conclusion

- Reproducible research
  - A way to improve our daily work, with immediate benefits.
  - An opportunity to think about our practices.
  - A research field of its own.
- Many solutions and tools are now ready for use.
- It is important to promote a culture change that will integrate computational reproducibility into the research process.
- Reproducible will be the default in future.

Acknowledgement

Roger Peng

Mikhail Dozmorov

Victoria Stodden

- https://www.oreilly.com/ideas/putting-the-science-back-in-data-science

https://www2.stat.duke.edu/courses/Fall15/sta112.01/post/app_ex/app_Twitter_election_results.html


THANKS FOR YOUR ATTENTION