

REPRODUCIBLE RESEARCH: A PERSPECTIVE

Arnaud Legrand

Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP

SBAC-PAD, October 2019



COMMON BELIEFS

- RR mainly allows to fight scientific misconduct (e.g., fraud).
That's nice but I'm honest so just let me do my work!

COMMON BELIEFS

- RR mainly allows to fight scientific misconduct (e.g., fraud).
That's nice but I'm honest so just let me do my work!
- RR is all about re-executing the same code, even if the code is stupid and makes wrong computation. It's pointless!

COMMON BELIEFS

- RR mainly allows to fight scientific misconduct (e.g., fraud).
That's nice but I'm honest so just let me do my work!
- RR is all about re-executing the same code, even if the code is stupid and makes wrong computation. It's pointless!
- My student has done everything with org-jupyter-studio-mode.
Now he's gone and I can't reuse what he did. See, what's the point? RR does not help!

COMMON BELIEFS

- RR mainly allows to fight scientific misconduct (e.g., fraud).
That's nice but I'm honest so just let me do my work!
- RR is all about re-executing the same code, even if the code is stupid and makes wrong computation. It's pointless!
- My student has done everything with org-jupyter-studio-mode. Now he's gone and I can't reuse what he did. See, what's the point? RR does not help!
- RR is about controlling and checking everything, which slows down the scientific discovery process. Changing the way we work and publish may be harmful!

PAST

Claerbout & Karrenbach, meeting of the Society of Exploration Geophysics, 1992

Electronic Documents Give Reproducible Research a New Meaning

RE1.3

Jon F. Claerbout and Martin Karrenbach, Stanford Univ.

SUMMARY

A revolution in education and technology transfer follows from the marriage of word processing and software command scripts. In this marriage an author attaches to every figure caption a pushbutton or a name tag usable to recalculate the figure from all its data, parameters, and programs. This provides a new level of reproducibility in computer documents.

In 1990, we set this sequence of goals:

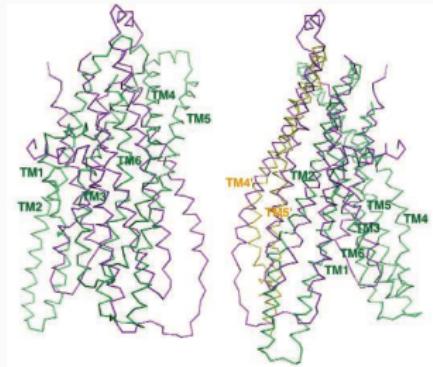
- Learn how to merge a publication with its underlying computational analysis.
- Teach researchers how to prepare a document in a form where they themselves can reproduce their own research results a year or more later by "pressing a single button".
- Learn how to leave finished work in a condition where coworkers can reproduce the calculation including the final illustration by pressing a button in its caption.
- Prepare a complete copy of our local software environment so that graduating students can take their work away with them to other sites, press a button, and reproduce their Stanford work.
- Merge electronic documents written by multiple authors (SEP reports).

- make incremental improvements in electronic-document software
- seek partners for broadening standards (and making incremental improvements).

Our basic goal is reproducible research. The electronic document is our means to this end. In principle, reproducibility in research can be achieved without electronic documents and that is how we started. Our first nonelectronic reproducible document was a textbook in which the paper document contained the name of a program script in every figure caption. The program scripts were organized by book chapter and section so they could be correlated to an accompanying magnetic tape dump of the file system. The magnetic tape also contained all the necessary data to feed the program script.

Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display seismic images of the earth's interior, but to reach wider audiences, Figure 1 shows a satellite weather picture which the pushbutton will animate as seen on commercial television. We include all our plot software as well as freely available software from many sources, including compilers and the L^AT_EX word processing system. Naturally some software includes licensed software, but with the exception

UNFORTUNATE MISTAKES



Geoffrey Chang (Scripps, UCSD) works on crystallography and studies the structure of cell membrane proteins. He specialized in structures of **multidrug resistant transporter proteins in bacteria**: MsbA de Escherichia Choli (Science, 2001), Vibrio cholera (Mol. Biology, 2003), Salmonella typhimurium (Science, 2005)

2006: Inconsistencies reveal a programming mistake

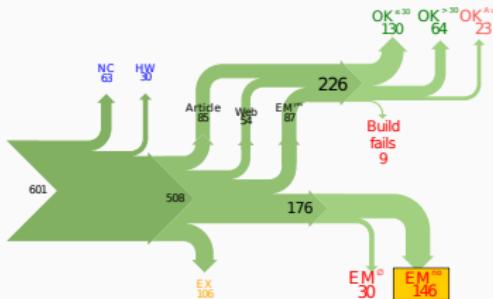
a homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the protein structure.

5 retractions that motivate improved software engineering practices in computational biology

THE "REPRODUCIBILITY CRISIS"

"clinical trials in oncology have the highest failure rate [only 5% are licensed] compared with other therapeutic areas [...] difficulty to repeat experiments even in the laboratory of the original investigator"

Begley and Ellis, Nature, 2012. *Raise standards for preclinical cancer research*



8 ACM conferences and 5 journals
EM^{no}= **the code cannot be provided**
Collberg, Christian et al., *Measuring Reproducibility in Computer Systems Research*, 2015

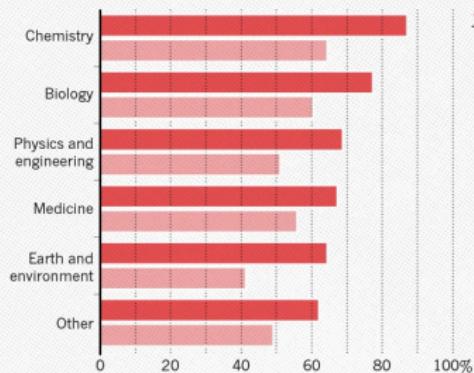
The scientific process demands the highest standards of **quality**, **ethics** and **rigor**.

WHY ARE SCIENTIFIC STUDIES SO DIFFICULT TO REPRODUCE?

HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?

Most scientists have experienced failure to reproduce results.

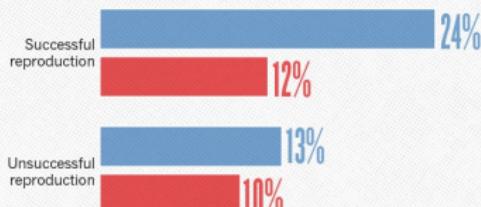
- Someone else's
- My own



HAVE YOU EVER TRIED TO PUBLISH A REPRODUCTION ATTEMPT?

Although only a small proportion of respondents tried to publish replication attempts, many had their papers accepted.

- Published
- Failed to publish



1,500 scientists lift the lid on reproducibility,
Nature, May 2016

Social causes

- Fraud, conflict of interest (pharmaceutic, ...)
- No incentive to reproduce/check our own work (afap), nor the work of others (big results!), nor to allow others to check (competition)
- Peer review does not scale: 1+ million articles per year!

Methodological or technical causes

- The many biases, (apophenia, confirmation, ...), HARKing: bad designs
- Selective reporting, weak analysis (statistics, data processing, *computational errors)
- Lack of information, code/raw data unavailable

DIFFERENT REPRODUCIBILITY CONCERNS

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance, ...

Computational fluid dynamics numerical issues

Authors

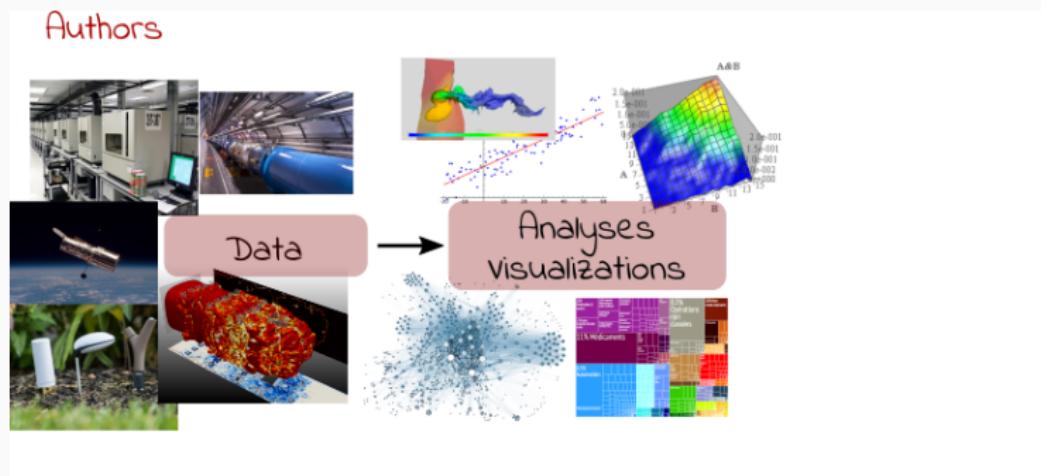


DIFFERENT REPRODUCIBILITY CONCERN

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance, ...

Computational fluid dynamics numerical issues

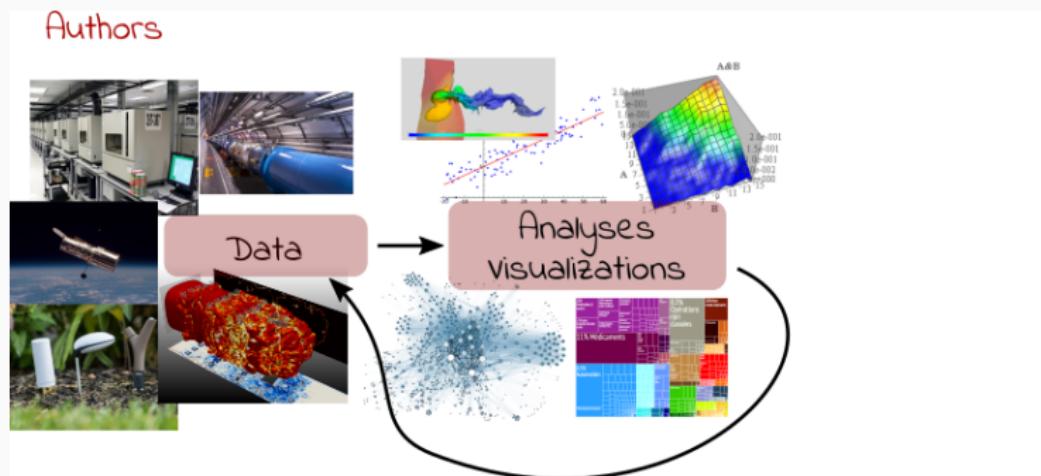


DIFFERENT REPRODUCIBILITY CONCERN

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance, ...

Computational fluid dynamics numerical issues

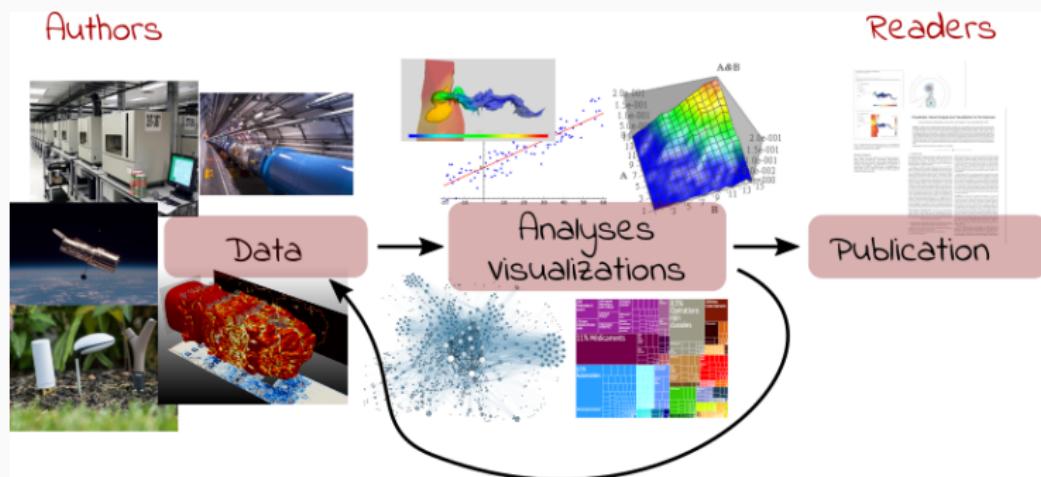


DIFFERENT REPRODUCIBILITY CONCERN

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance, ...

Computational fluid dynamics numerical issues

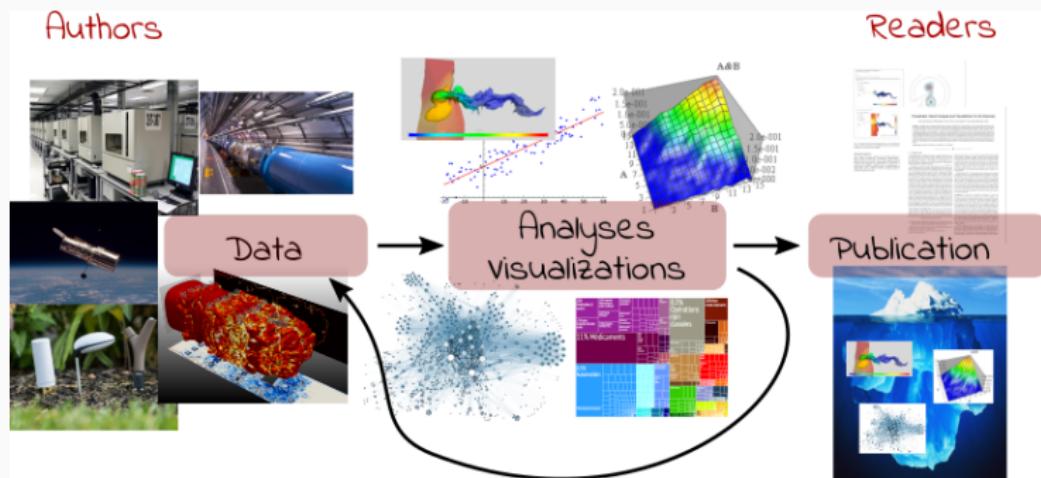


DIFFERENT REPRODUCIBILITY CONCERN

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance, ...

Computational fluid dynamics numerical issues

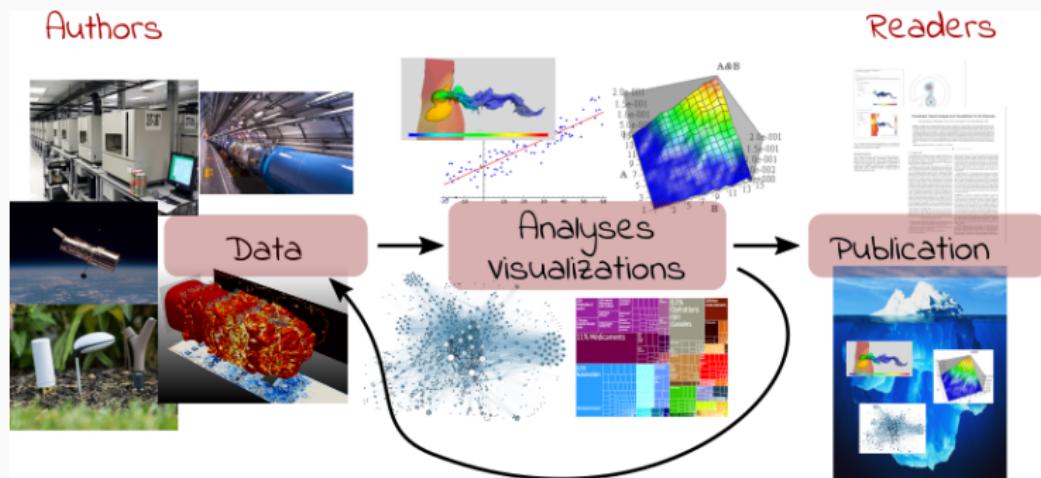


DIFFERENT REPRODUCIBILITY CONCERN

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance, ...

Computational fluid dynamics numerical issues

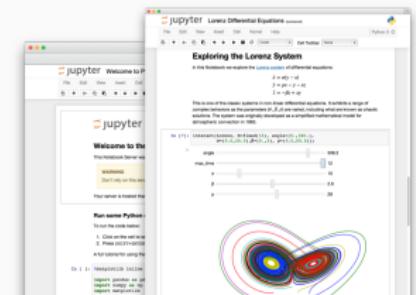


Reproducible Research = Bridging the Gap by working Transparently

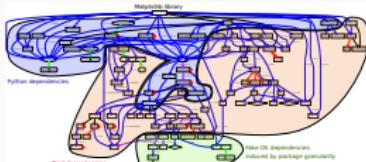
PRESENT

EXISTING TOOLS, EMERGING STANDARDS

Notebooks and workflows



Software environments



Sharing platforms



TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

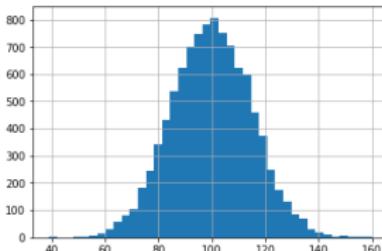
3.141592653589793

Mais calculé avec la méthode des [aiguilles de Buffon](#), on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

3.1437198694098765

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).



TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** # Un document computationnel
- In [1]:** A code cell containing:

```
from math import *
print(pi)
3.141592653589793
```
- Out [1]:** The output 3.141592653589793, followed by a note: "Mais calculé avec la [méthode des aiguilles de Buffon](#) (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtientrait comme approximation :".
- In [2]:** A code cell containing:

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1))/N
```
- Out [2]:** The output 3.14371986944998765, followed by a note: "On peut inclure des formules mathématiques comme $\sqrt{2/\pi} \exp(-x^2/2)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation...)."
- In [3]:** A code cell containing:

```
%matplotlib inline
import matplotlib.pyplot as plt
mu, sigma = 100, 15
x = mu + sigma*np.random.randn(10000)
plt.hist(x,40)
plt.grid(True)
plt.show()
```
- Out [3]:** A histogram showing a bell-shaped distribution centered around 100.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

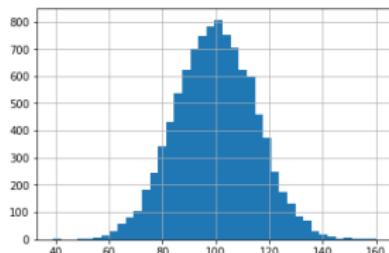
3.141592653589793

Mais calculé avec la [méthode des aiguilles de Buffon](#), on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1))/N
```

3.14371986944998765

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation...).



TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

Jupyter example_pl (modified)

Un document computationnel

Mais ordinateur m'indique que π vaut "approximativement"

```
In [1]:
```

```
from math import *
print(pi)
3.141592653589793
```

Mais calculé avec la [méthode des aiguilles de Buffon](#) (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme approximation :

```
In [2]:
```

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=np.pi/2)
2/(sum((x+np.sin(theta))>1))/N
```

Out[2]: 3.14371986944998765

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).

```
In [3]:
```

```
%matplotlib inline
import matplotlib.pyplot as plt
mu, sigma = 100, 15
x = mu + sigma*np.random.randn(10000)
plt.hist(x,40)
plt.grid(True)
plt.show()
```

800
700
600
500
400
300
200
100
0

40 60 80 100 120 140 160

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

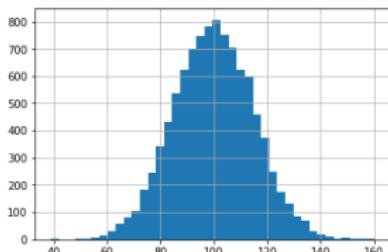
3.141592653589793

Mais calculé avec la [méthode des aiguilles de Buffon](#), on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=np.pi/2)
2/(sum((x+np.sin(theta))>1))/N
```

3.14371986944998765

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).



TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

The screenshot shows a Jupyter Notebook interface with three code cells:

- In [1]:** A single-line print statement that outputs the value of pi.
- In [2]:** A more complex script that generates random points (x, y) and calculates an approximation of pi based on the ratio of points in a unit square falling within a quarter circle.
- In [3]:** A histogram plot of a normal distribution with mean 100 and standard deviation 15.

Annotations with red arrows point from the text "Code" and "Document computationnel" to the code cells in the notebook.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

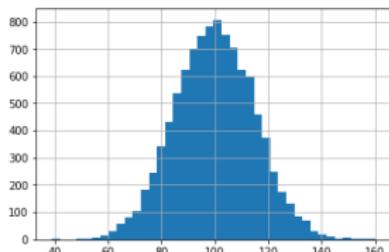
3.141592653589793

Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2*(sum((x+np.sin(theta))>1))/N
```

3.14371986949098765

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).



TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

Un document computationnel

```
In [1]:  
from math import *  
print(pi)  
3,141592653589793
```

Mais calculé avec la `_methode_des_aiguilles_de_Buffon` (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtiendrait comme approximation :

```
In [2]:  
import numpy as np  
N = 1000000  
x = np.random.uniform(size=N, low=0, high=1)  
theta = np.random.uniform(size=N, low=0, high=pi/2)  
2*(sum((x*np.sin(theta))>1))/N  
Out[2]: 3,1437198694098765
```

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).

```
In [3]:  
%matplotlib inline  
import matplotlib.pyplot as plt  
  
mu, sigma = 100, 15  
x = mu + sigma*np.random.randn(10000)  
  
plt.hist(x, 99)  
plt.grid(True)  
plt.show()
```

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

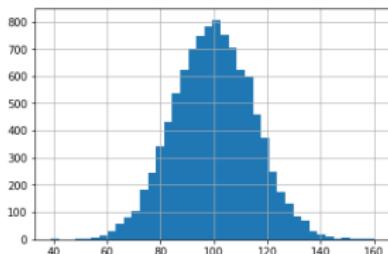
Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```
import numpy as np  
N = 1000000  
x = np.random.uniform(size=N, low=0, high=1)  
theta = np.random.uniform(size=N, low=0, high=pi/2)  
2*(sum((x*np.sin(theta))>1))/N
```

3.1437198694098765

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et

des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).



TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

The screenshot shows a Jupyter Notebook interface with three code cells:

- In [1]:** Prints the value of pi (3.141592653589793) and includes a note about calculating pi with Buffon's needle method.
- In [2]:** Generates random points (x, y) and calculates the ratio of points below the unit circle to total points, which is approximately 0.7854.
- In [3]:** Plots a histogram of 100,000 random numbers between 0 and 100, showing a bell-shaped distribution centered around 50.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

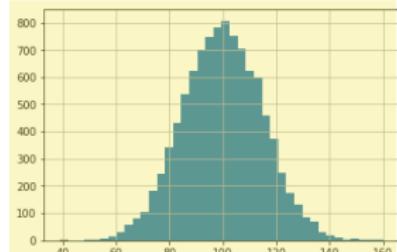
Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1))/N
```

3.1437198694098765

Export →

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).



TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING

Document initial dans son environnement

The screenshot shows a Jupyter Notebook interface with three code cells:

- In [1]:** Prints the value of pi (3.141592653589793) and includes a note about calculating pi with Buffon's needle method.
- In [2]:** Generates a sample of random points and calculates the ratio of points below a diagonal line to the total number of points, which is used to approximate pi.
- In [3]:** Plots a histogram of the generated data, showing a bell-shaped curve centered around 100.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

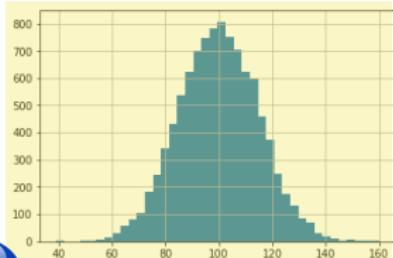
Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1))/N
```

3.1437198694098765

Export

On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).



TOOL 2: FIGHTING SOFTWARE ENVIRONMENTS NIGHTMARE

What is hiding behind a simple

```
1 import matplotlib
```

```
Package: python3-matplotlib
Version: 2.1.1-2
Depends: python3-dateutil, python-matplotlib-data (>= 2.1.1-2),
python3-pyparsing (>= 1.5.6), python3-six (>= 1.10), python3-tz,
libjs-jquery, libjs-jquery-ui, python3-numpy (>= 1:1.13.1),
python3-numpy-abi9, python3 (<< 3.7), python3 (>= 3.6~),
python3-cycler (>= 0.10.0), python3:any (>= 3.3.2-2~), libc6 (>=
2.14), libfreetype6 (>= 2.2.1), libgcc1 (>= 1:3.0), libpng16-16 (>=
1.6.2-1), libstdc++6 (>= 5.2), zlib1g (>= 1:1.1.4)
```

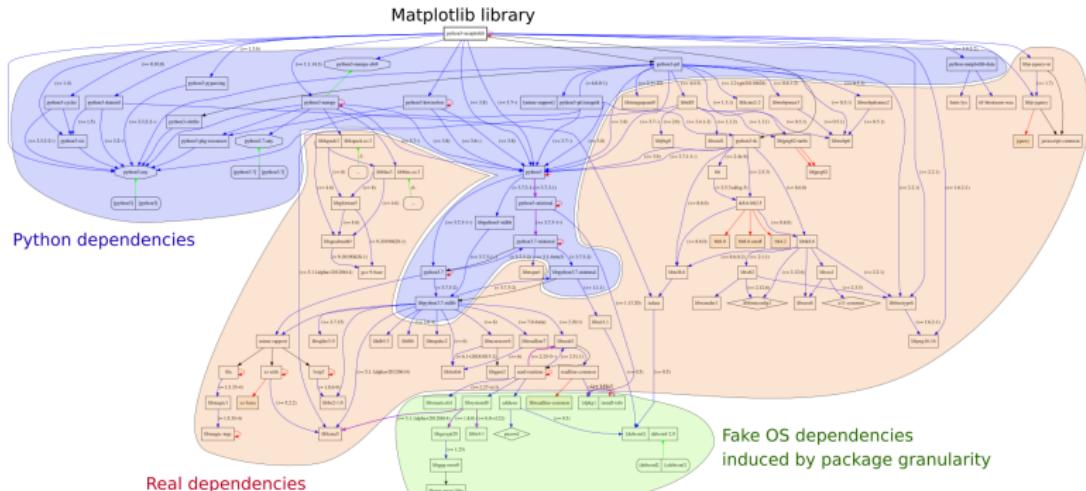
TOOL 2: FIGHTING SOFTWARE ENVIRONMENTS NIGHTMARE

What is hiding behind a simple

1

```
import matplotlib
```

Package: python3-matplotlib



TOOL 2: FIGHTING SOFTWARE ENVIRONMENTS NIGHTMARE

Python and its rapidly evolving environment

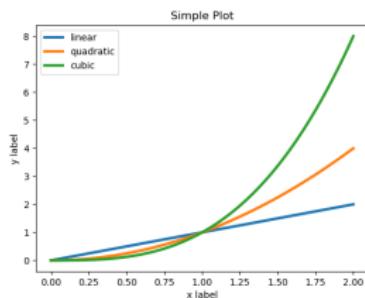
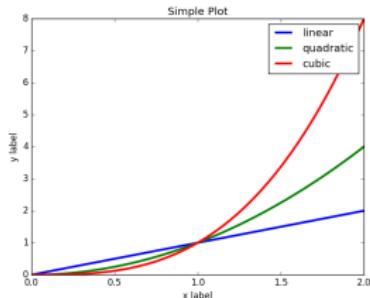
```
1 python2 -c "print(10/3)"  
2 python3 -c "print(10/3)"
```

```
3  
3.333333333333335
```

TOOL 2: FIGHTING SOFTWARE ENVIRONMENTS NIGHTMARE

Python and its rapidly evolving environment

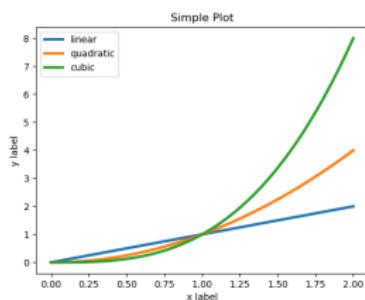
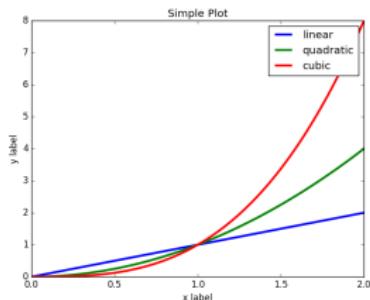
```
1 python2 -c "print(10/3)"  
2 python3 -c "print(10/3)"
```



TOOL 2: FIGHTING SOFTWARE ENVIRONMENTS NIGHTMARE

Python and its rapidly evolving environment

```
1 python2 -c "print(10/3)"  
2 python3 -c "print(10/3)"
```

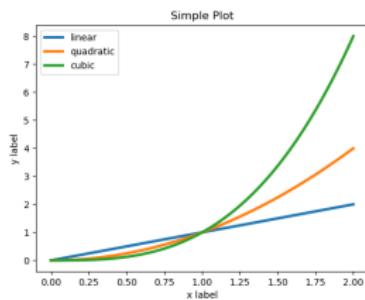
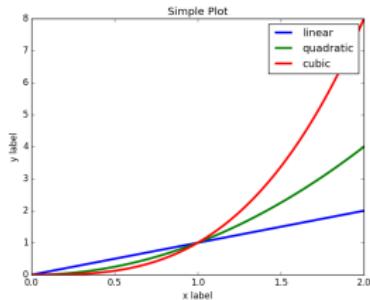


Cortical Thickness Measurements (PLOS ONE, June 2012): FreeSurfer: differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.

TOOL 2: FIGHTING SOFTWARE ENVIRONMENTS NIGHTMARE

Python and its rapidly evolving environment

```
1 python2 -c "print(10/3)"  
2 python3 -c "print(10/3)"
```



Cortical Thickness Measurements (PLOS ONE, June 2012): FreeSurfer: differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.



TOOL 3: FIGHTING INFORMATION LOSS WITH ARCHIVES

D. Spinellis. The Decay and Failures of URL References. CACM, 46(1), Jan 2003.

The half-life of a referenced URL is approximately 4 years from its publication date.

P. Habibzadeh. Decay of References to Web sites in Articles Published in General Medical Journals: Mainstream vs Small Journals". Applied Clinical Informatics. 4 (4), 2013

half life ranged from 2.2 years in EMHJ to 5.3 years in BMJ

TOOL 3: FIGHTING INFORMATION LOSS WITH ARCHIVES

D. Spinellis. The Decay and Failures of URL References. CACM, 46(1), Jan 2003.

The half-life of a referenced URL is approximately 4 years from its publication date.

P. Habibzadeh. Decay of References to Web sites in Articles Published in General Medical Journals: Mainstream vs Small Journals". Applied Clinical Informatics. 4 (4), 2013

half life ranged from 2.2 years in EMHJ to 5.3 years in BMJ

Article archives arXiv.org HAL
archives-ouvertes.fr

Data archives figshare zenodo

Software Archive Software Heritage

 or  = awesome collaborations ≠ archive

CHANGING PRACTICES

Manifesto: "*I solemnly pledge*" (WSSSPE, Lorena Barba, FAIR)

1. I will teach my graduate students about reproducibility
2. All our research code (and writing) is under version control
3. We will always carry out verification and validation
4. We will share data, plotting script & figure under CC-BY
5. We will upload the preprint to arXiv at the time of submission of a paper
6. We will release code at the time of submission of a paper
7. We will add a "Reproducibility" declaration at the end of each paper
8. I will keep an up-to-date web presence

Soft. Engineering, Statistics, and Reproducible Research in the **curricula**



- Webinars on RR 2016-2017
- MOOC on RR (3rd edition planned for January 2020)
- Book on RR in June 2019

Artifact evaluation and ACM badges



Major conferences

- Supercomputing: Artifact Description (AD) **mandatory**, Artifact Evaluation (AE) still **optional**, Double blind vs. RR
- NeurIPS, ICLR: **open reviews**, reproducibility challenge



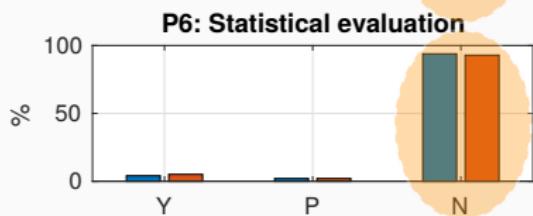
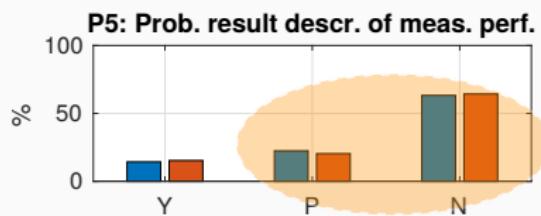
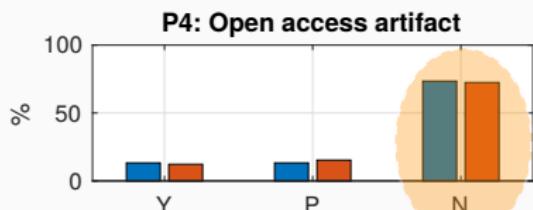
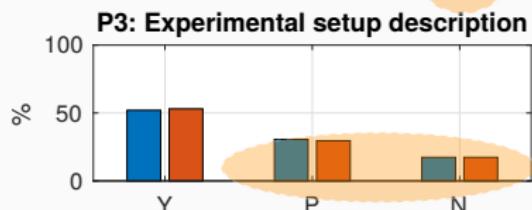
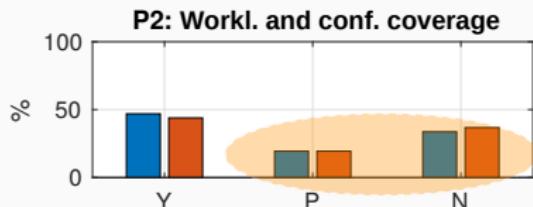
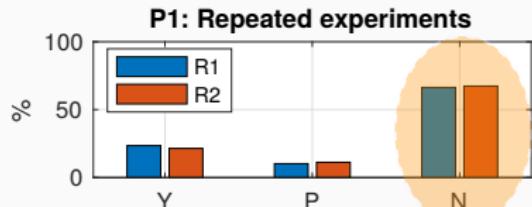
Joelle Pineau @ NeurIPS'18

- ACM SIGMOD 2015-2019, ...

Mentalities are evolving people care, make stuff available, errors are found and fixed

KEY CONCERN FOR OUR COMMUNITY (ROOM FOR IMPROVEMENT)

- Awareness of Experiments and Statistics How are cloud performance currently obtained and reported?, March 2019



FUTURE

PUBLISH OR PERISH (OK, THIS IS PAST AND PRESENT)

- Goodhart's Law: Are Academic Metrics Being Gamed?, M. Fire 2019
 - AI: over 1,000 ranked journals ($\times 10$ in 15 years)
 - Shorter papers with increasing self references
 - More and more papers without any citation
 - Sharp increase in the number of new authors publishing at a much faster rate given their career age
- The Truth, The Whole Truth, and Nothing But the Truth: A Pragmatic Guide to Assessing Empirical Evaluations, TOPLAS 2016



PUBLISH OR PERISH (OK, THIS IS PAST AND PRESENT)

- Goodhart's Law: Are Academic Metrics Being Gamed?, M. Fire 2019
 - AI: over 1,000 ranked journals ($\times 10$ in 15 years)
 - Shorter papers with increasing self references
 - More and more papers without any citation
 - Sharp increase in the number of new authors publishing at a much faster rate given their career age
- The Truth, The Whole Truth, and Nothing But the Truth: A Pragmatic Guide to Assessing Empirical Evaluations, TOPLAS 2016



REPRODUCIBLE RESEARCH = TRANSPARENCY

To err is human.

Good research requires time and resources

1. **Train yourself and your students:** RR, statistics, experiments
 - Beware of checklists and norms
 - Understand what's at stake
2. **Change the norm:** make publication practices evolve
3. **Incentive:** consider RR/open science when hiring/promoting
4. **Prepare the Future:** Toward **literate experimentation?**
 - Reuse, reuse, reuse!
 - Shared and controled testbeds (e.g., Grid'5000/SILECS)
 - How to share Experiments ?

SOME ADVERTISING

TEN YEARS REPRODUCIBILITY CHALLENGE

RESCIENCE SPECIAL ISSUE
FREE TO READ - FREE TO PUBLISH



Would you dare to run the code from your past self?
(the one that does not answer mail)

SUBMISSION DEADLINE 01/04/2020
<http://rescience.github.io/ten-years>
In association with Inria, CQRS, Software Heritage, Rescience, Comité pour la Science Ouverte, DRFST Bordeaux & Mission de la pédagogie et du numérique pour l'enseignement supérieur.



3rd Edition: ≈ Feb. 2020

A new MOOC: "Advanced R"

- Software environment control (Docker)
- Scientific workflow (snakemake)
- Managing data (HDF5, archiving)

October 2020 ?

<http://rescience.github.io/ten-years/>

17/100