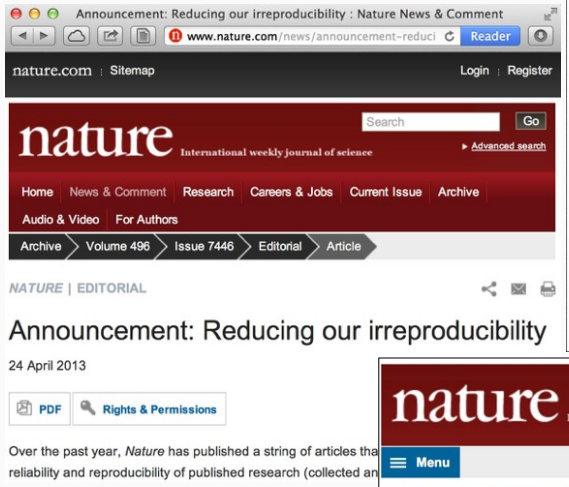
Reproducibility Crisis and Open Science

Arnaud Legrand
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP

International Week, Grenoble, January 2020

Public Evidence for a Lack of Reproducibility

- J.P. Ioannidis. *Why Most Published Research Findings Are False* PLoS Med. 2005.
- *Lies, Damned Lies, and Medical Science*, The Atlantic. Nov, 2010
- *Reproducibility: A tragedy of errors*, Nature, Feb 2016.
- Steen RG, Retractions in the scientific literature: is the incidence of research fraud increasing?. J. Med. Ethics 37, 2011



Courtesy V. Stodden, SC, 2015

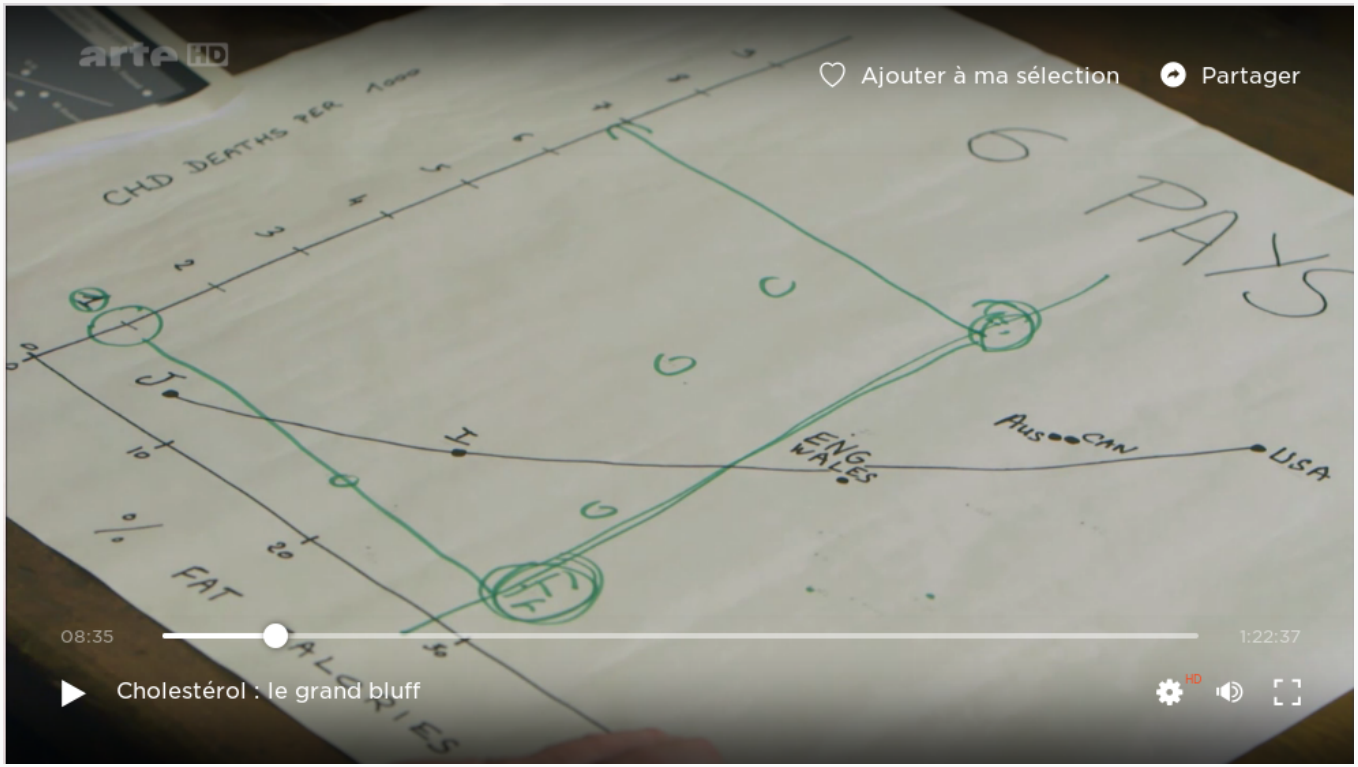
Correlation and Causation

A "vivid" debate: Cholesterol and Statins

Cholesterol: le grand bluff (Arte, 18/10/2016 @ 20h50)

A "vivid" debate: Cholesterol and Statins

Cholesterol: le grand bluff (Arte, 18/10/2016 @ 20h50)



"Careful" selection of data and influence from the industry ☹

But that's not what I want to illustrate now... Even if data hadn't been removed, could we really conclude something from such data?

Inference: Correlation and Causation

It may be the case that two random variables X and Y are dependent

- E.g., Let's pick a student at random and measure its *DrinkingHabit* and its *TestScore*

  In general, the more a student drinks the more his test goes down 🙂

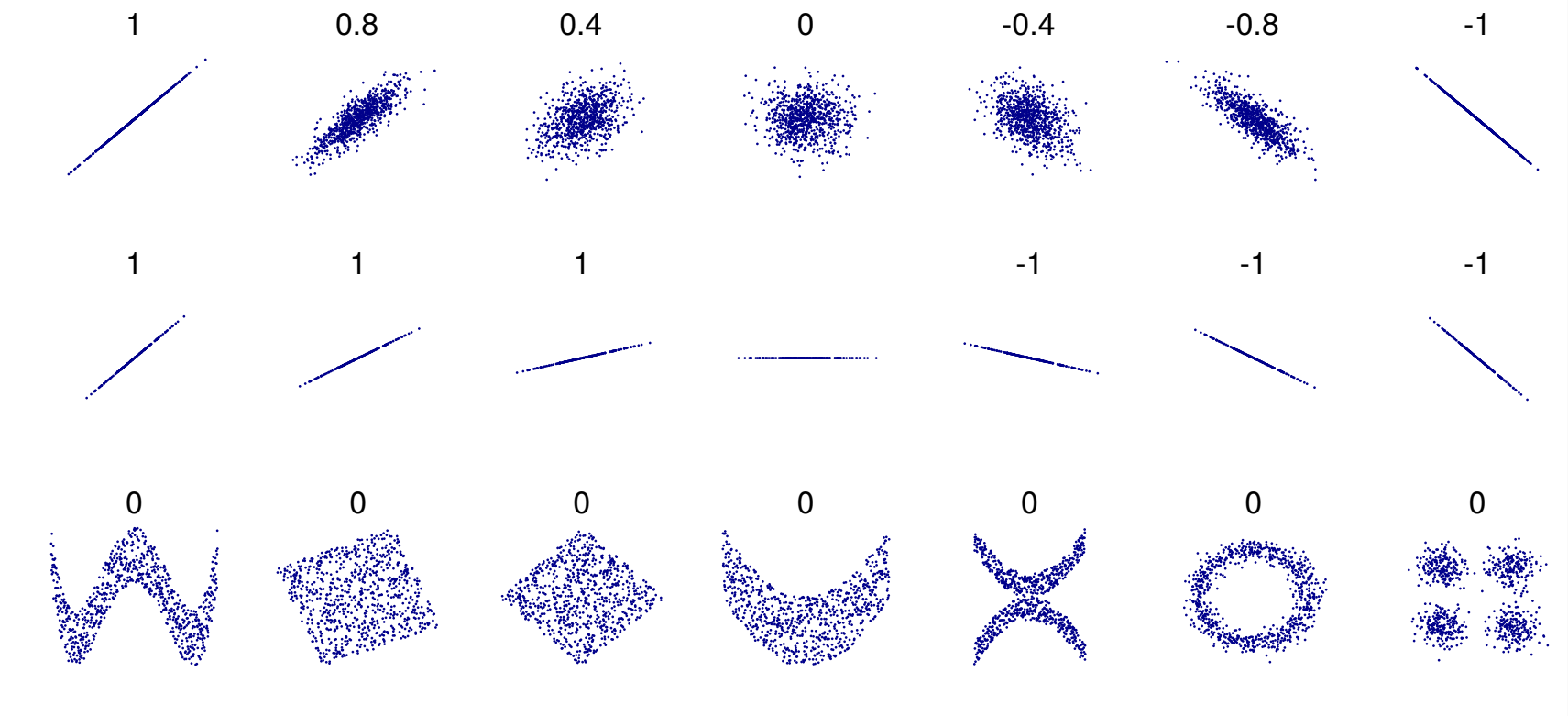
The correlation of two variables X and Y is defined as:

$$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

- The correlation is symmetrical ($\text{corr}(X, Y) = \text{corr}(Y, X)$)
- The correlation is in $[-1, 1]$
- $\text{corr}(Y, X) = 1$ or $-1 \Rightarrow$ perfectly linear relationship
- X independent of $Y \Rightarrow \text{corr}(X, Y) = 0$
- Y grows when X grows $\Rightarrow \text{corr}(X, Y) > 0$

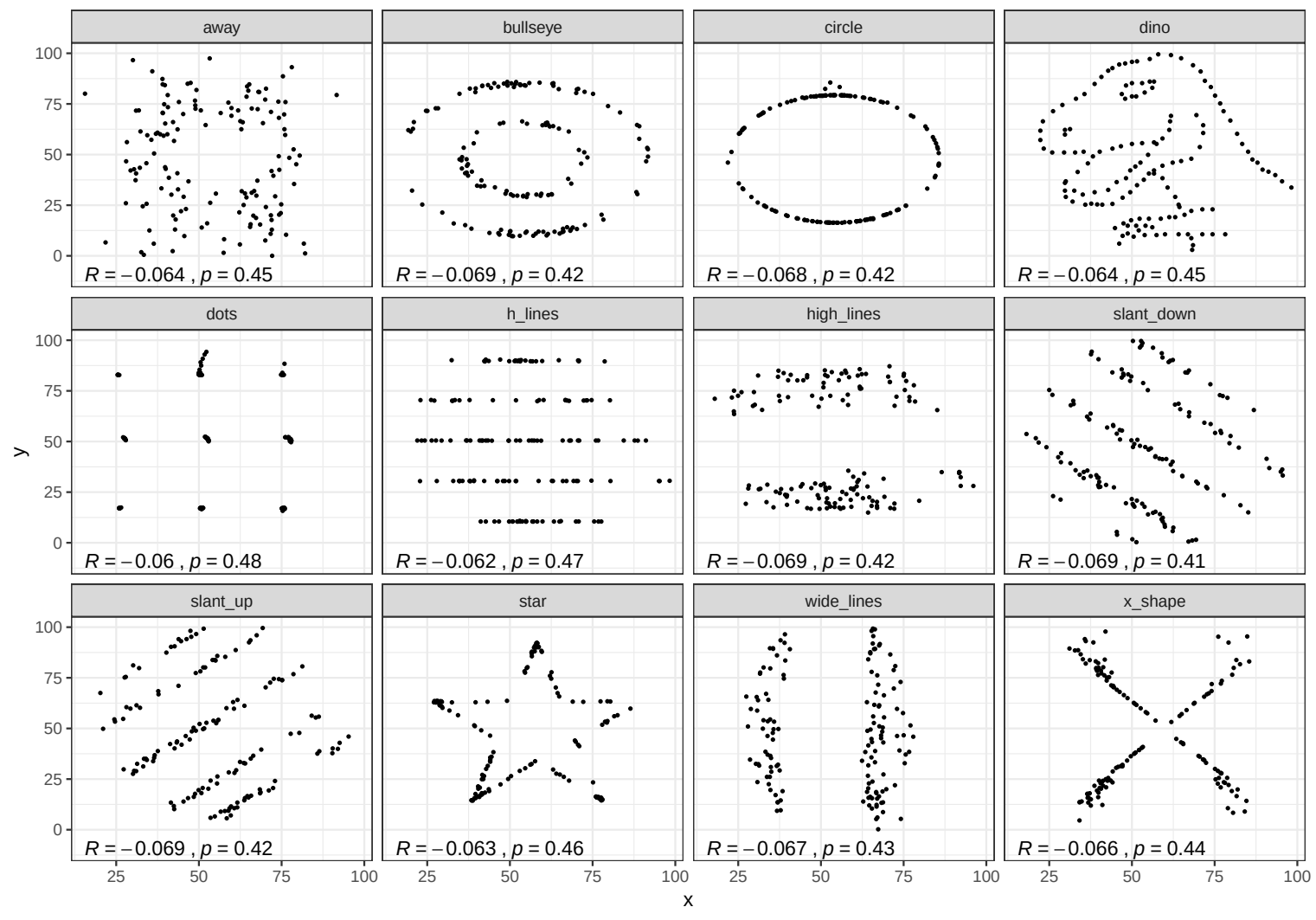
It is thus very tempting to use sample correlation as a way of knowing whether some variables are dependant

Scatter plot and correlation



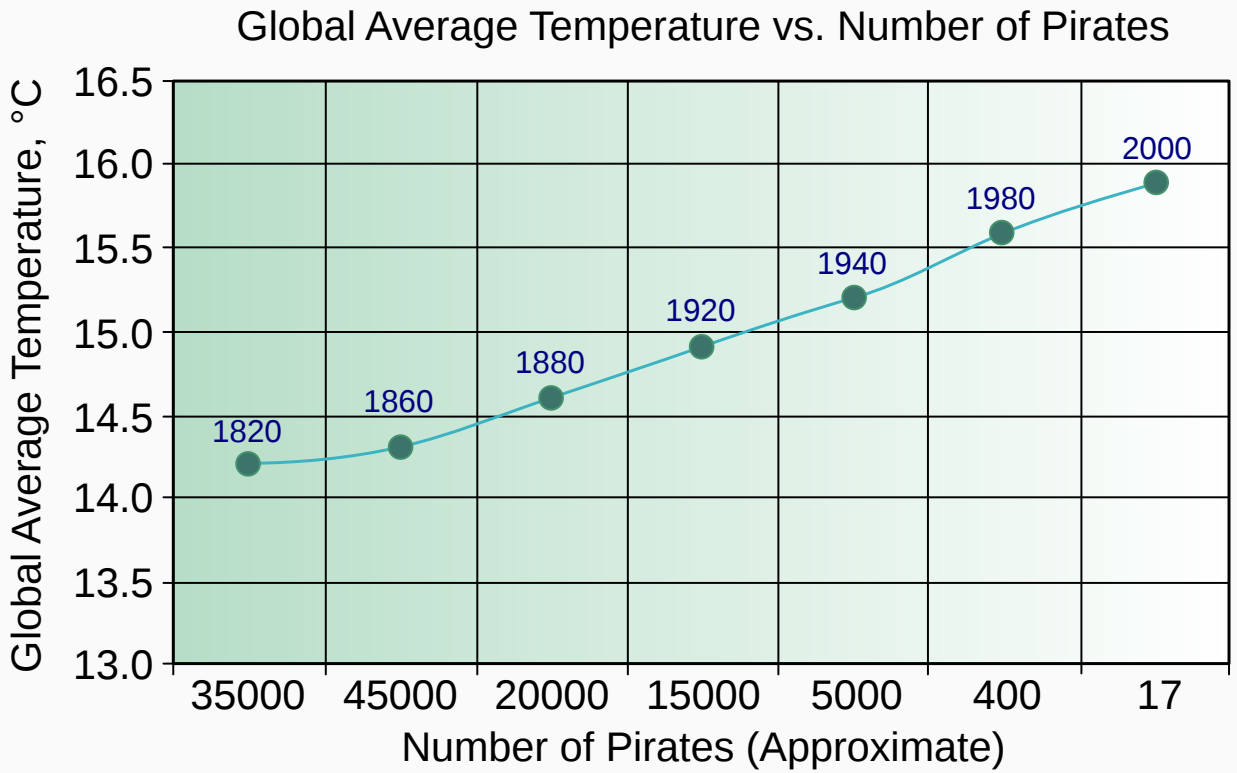
Non-linear relations or hidden variables are not be well trapped by correlation

The DataSaurus



https://www.autodeskresearch.com/publications/samestats

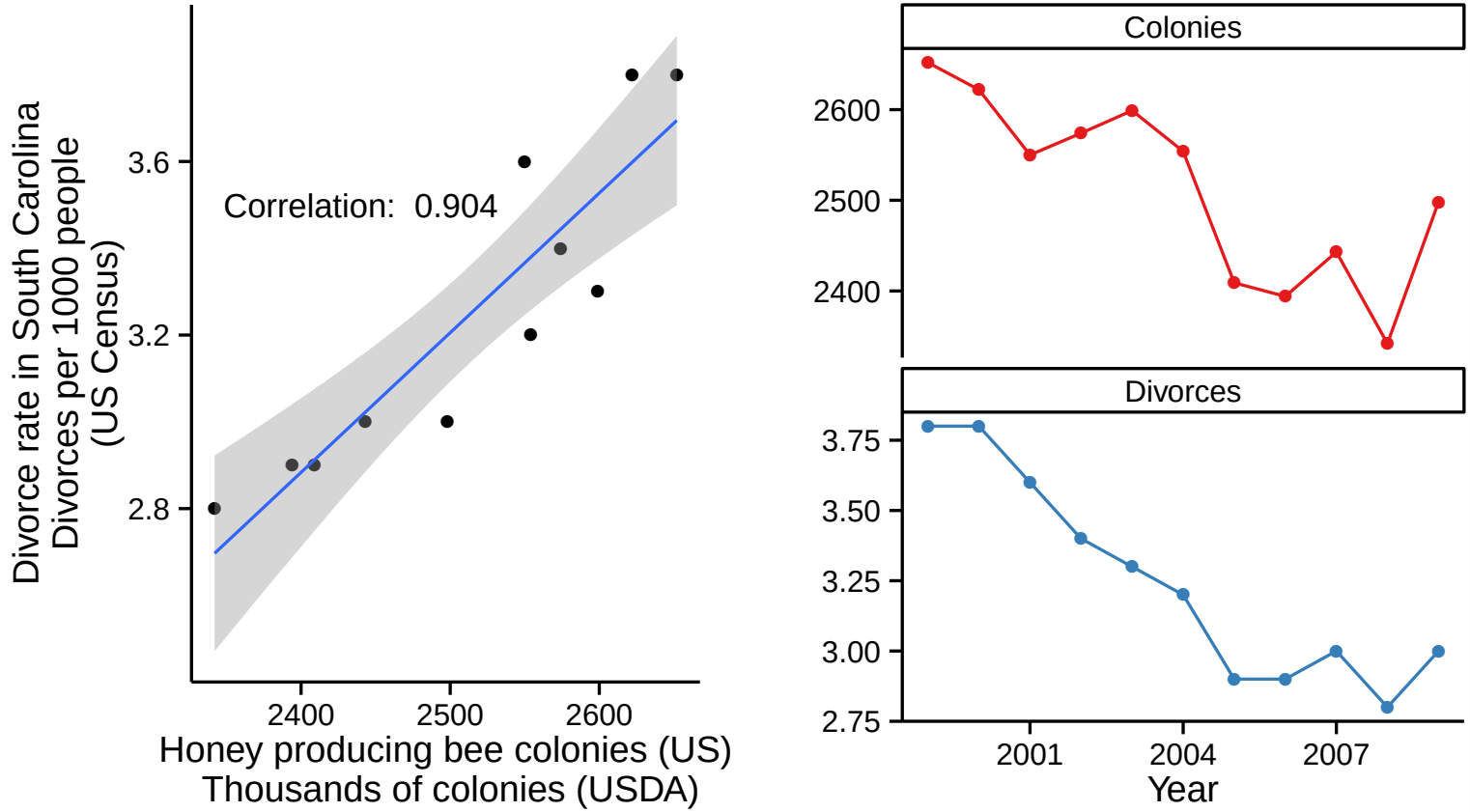
Correlation does not imply Causation



Mikhail Ryazanov (talk) - PiratesVsTemp.svg.
Licensed under CC BY-SA 3.0 via Wikimedia Commons

- 2 variables can be strongly correlated to a third one (e.g., year)
- Btw, what is wrong with this figure? 😉

Observational vs. Experimental Data Illustration



Source: *Spurious correlations*. For the good of the US society, we should try to get rid of honey bees 😉

The Deluge of Spurious Correlations in Big Data

The Deluge of Spurious Correlations in Big Data, by C. Calude and G. Longo, Foundations of Science, March 2016)

Is Data science is the end of science ?

- Powerful algorithms can now explore huge databases and find therein correlations and regularities.
- Properly defining ”meaning” or ”content” of such correlations is very difficult. But do we need to ?

Ergodic Theory

- Almost every trajectory (even deterministic and chaotic) will eventually iterate in a similar way
- Regularity is expected but it does not mean that prediction can be done

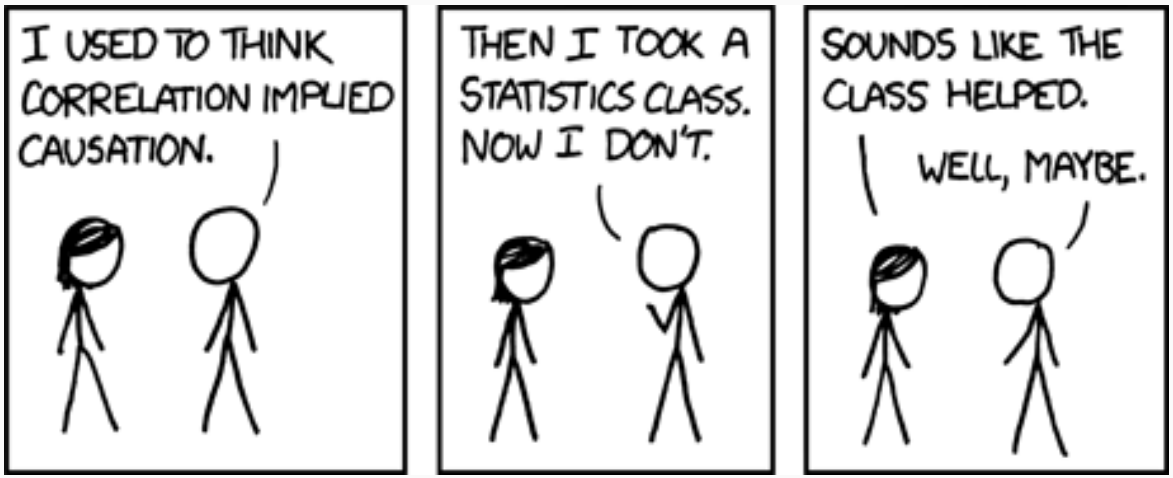
Ramsey Theory: Any sufficiently long string contains an arithmetic progression

- 0, 1, 1, 0, 0, 1, 1, 0, 0
- 0, 1, 1, 0, 0, 1, 1, 0, 1

Correlation does not imply Causation

For any two correlated events, A and B, the following relationships are possible:

- A causes B (direct causation) 
- A causes B and B causes A (bidirectional or cyclic causation) 
- A causes C which causes B (indirect causation) 
- B causes A; (reverse causation) 
- A and B are consequences of a common cause, but do not cause each other 
- There is no connection between A and B; it is a ”coincidence” 
  - But designed experiments can help you ruling this option out



Open Science

Scientific Misconduct ? What are the Consequences ?

The Duke University scandal with scientific misconduct on lung cancer

- *Nature Medicine* - 12, (2006) Genomic signatures to guide the use of chemotherapeutics, by Anil Potti and 16 other researchers from Duke and USF
- Major commercial labs licensed it and were about to start using it before two statisticians discovered and publicized its faults

Dr. Baggerly and Dr. Coombes found errors almost immediately. Some seemed careless — moving a row or a column over by one in a giant spreadsheet — while others seemed inexplicable. The Duke team shrugged them off as "clerical errors."

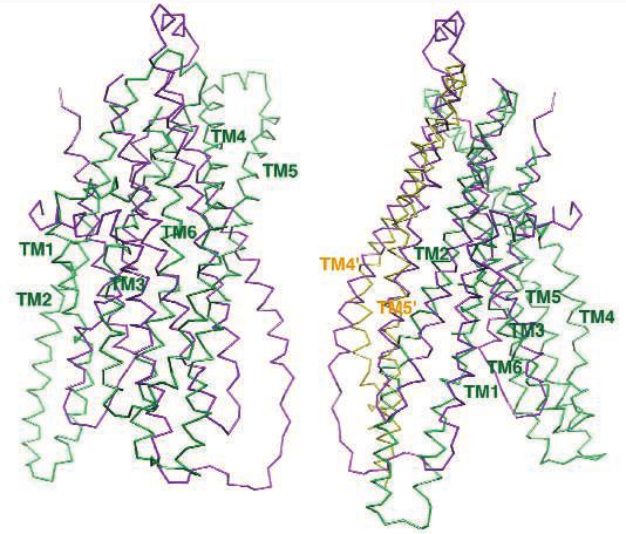
The Duke researchers continued to publish papers on their genomic signatures in prestigious journals. Meanwhile, they started three trials using the work to decide which drugs to give patients.

- Retractions: January 2011. Ten papers that Potti coauthored in prestigious journals were retracted for varying reasons

Bad science is deleterious

- It is used to backup stupid politics, it affects people's life, ...
- It blurs the frontier between scientists and crooks

Unfortunate Mistakes



Geoffrey Chang (Scripps, UCSD) works on crystalography and studies the structure of cell membrane proteins.
He specialized in structures of multidrug resistant transporter proteins in bacteria:
MsbA de Escheria 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.*

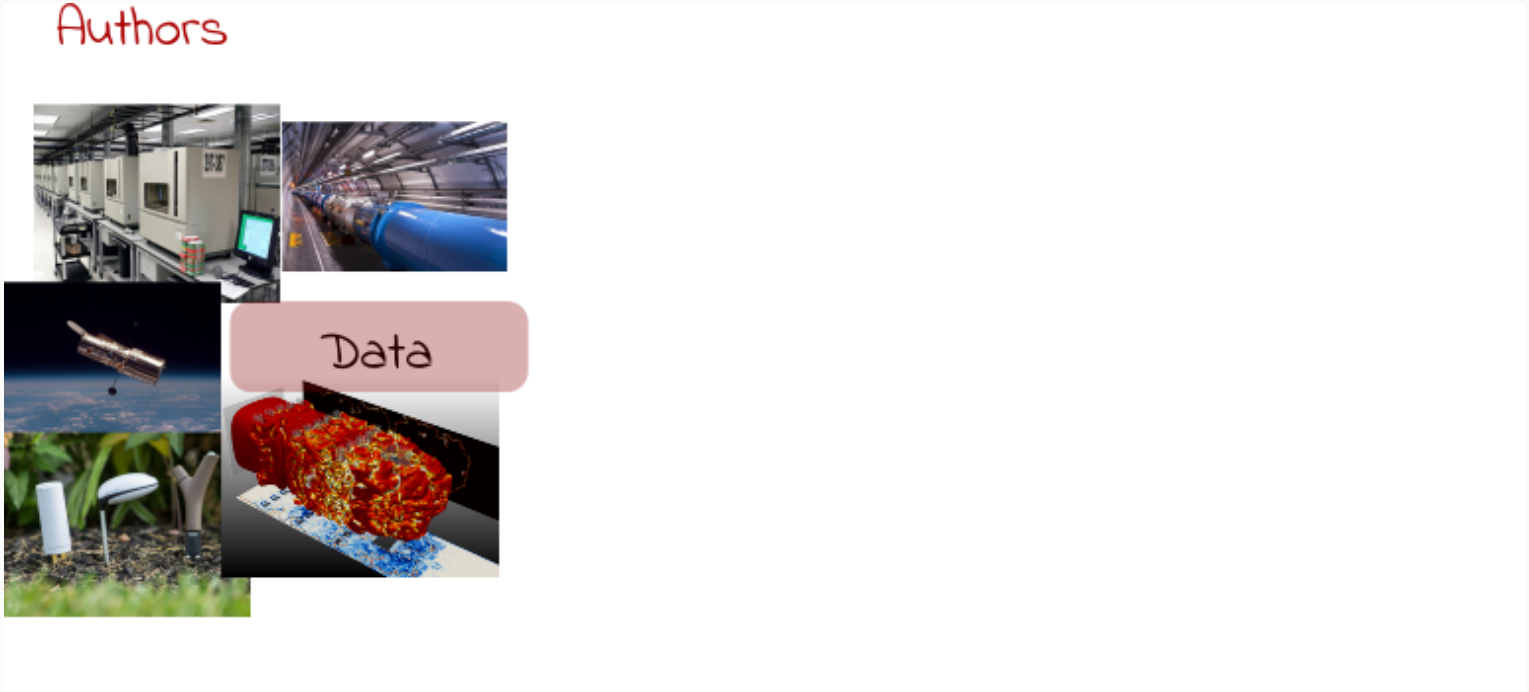
5 retractations that motivate improved software engineering practices in computational biology

Different Reproducibility Concerns

Social Sciences, Oncology, ... methodology, statistics

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

Computational fluid dynamics numerical issues

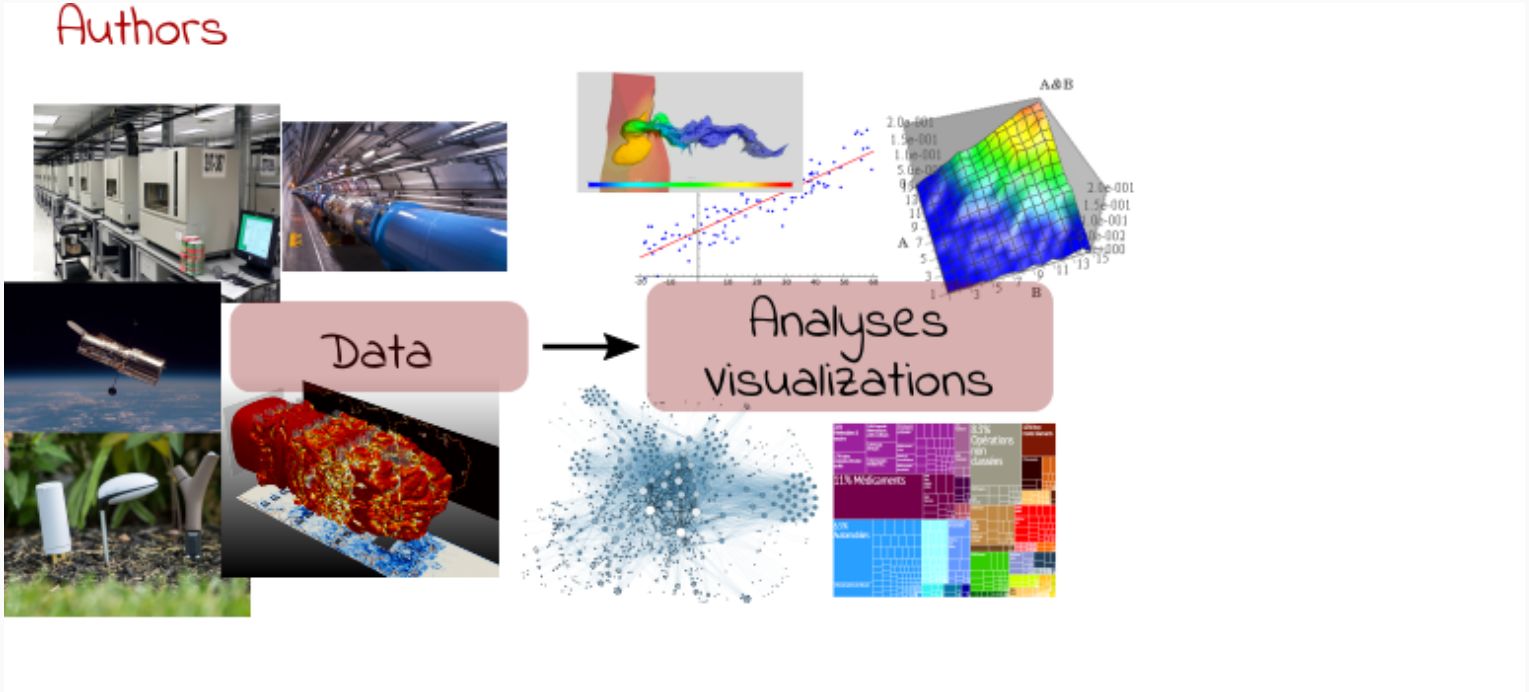


Different Reproducibility Concerns

Social Sciences, Oncology, ... methodology, statistics

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

Computational fluid dynamics numerical issues

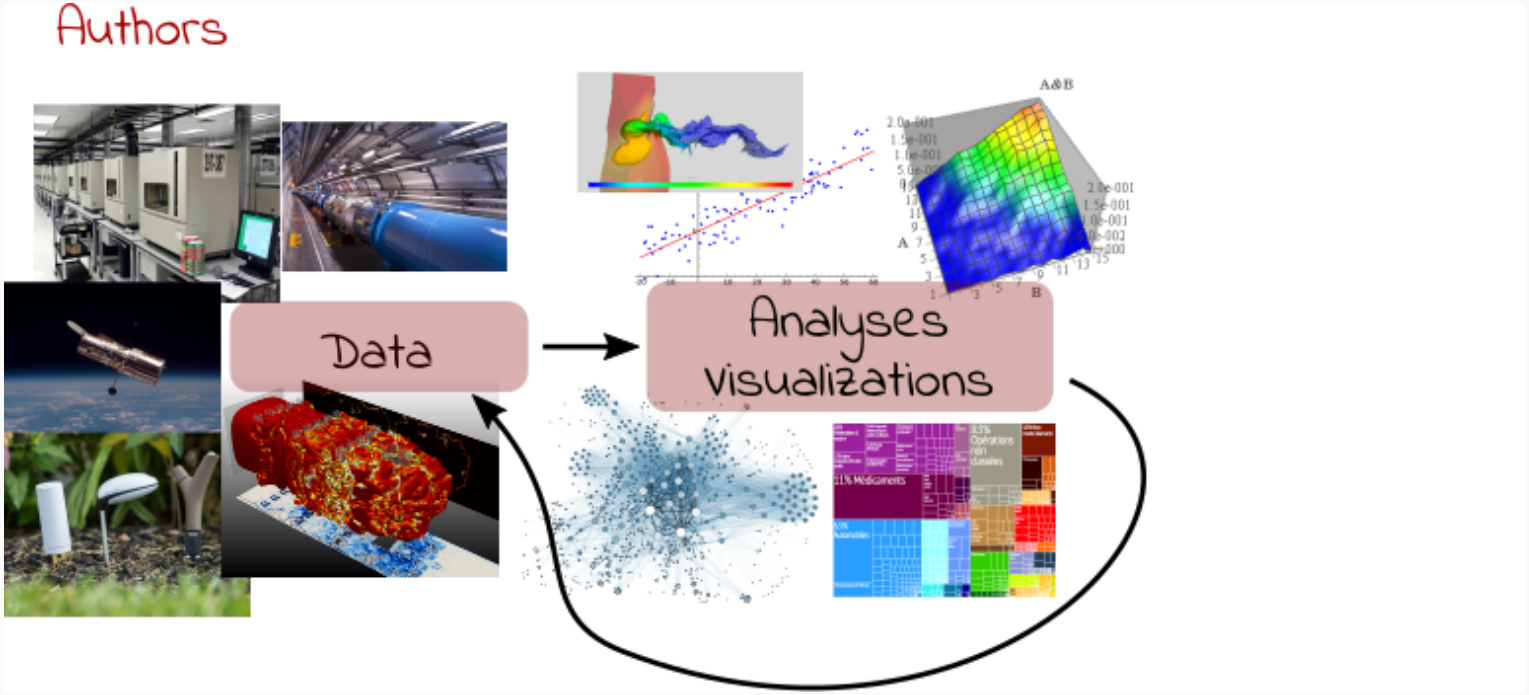


Different Reproducibility Concerns

Social Sciences, Oncology, ... methodology, statistics

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

Computational fluid dynamics numerical issues

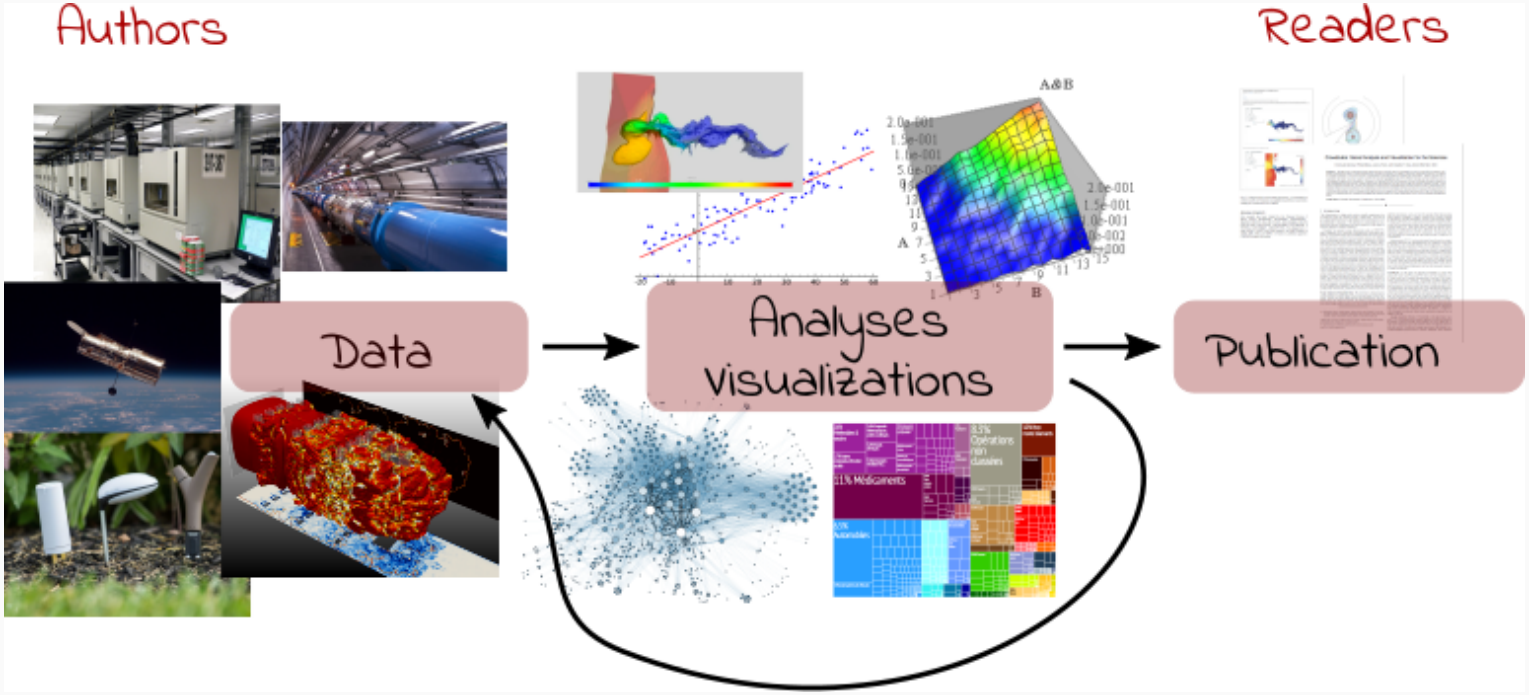


Different Reproducibility Concerns

Social Sciences, Oncology, ... methodology, statistics

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

Computational fluid dynamics numerical issues

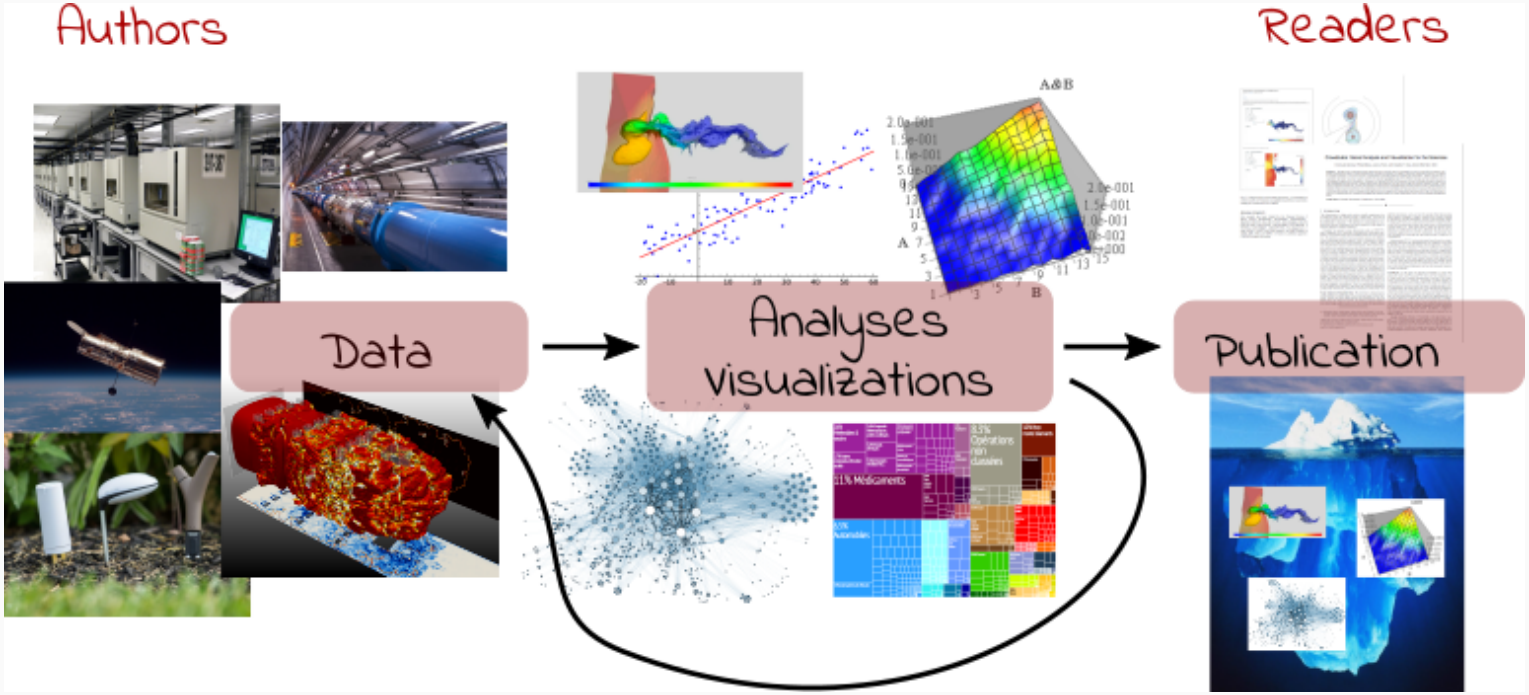


Different Reproducibility Concerns

Social Sciences, Oncology, ... methodology, statistics

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

Computational fluid dynamics numerical issues

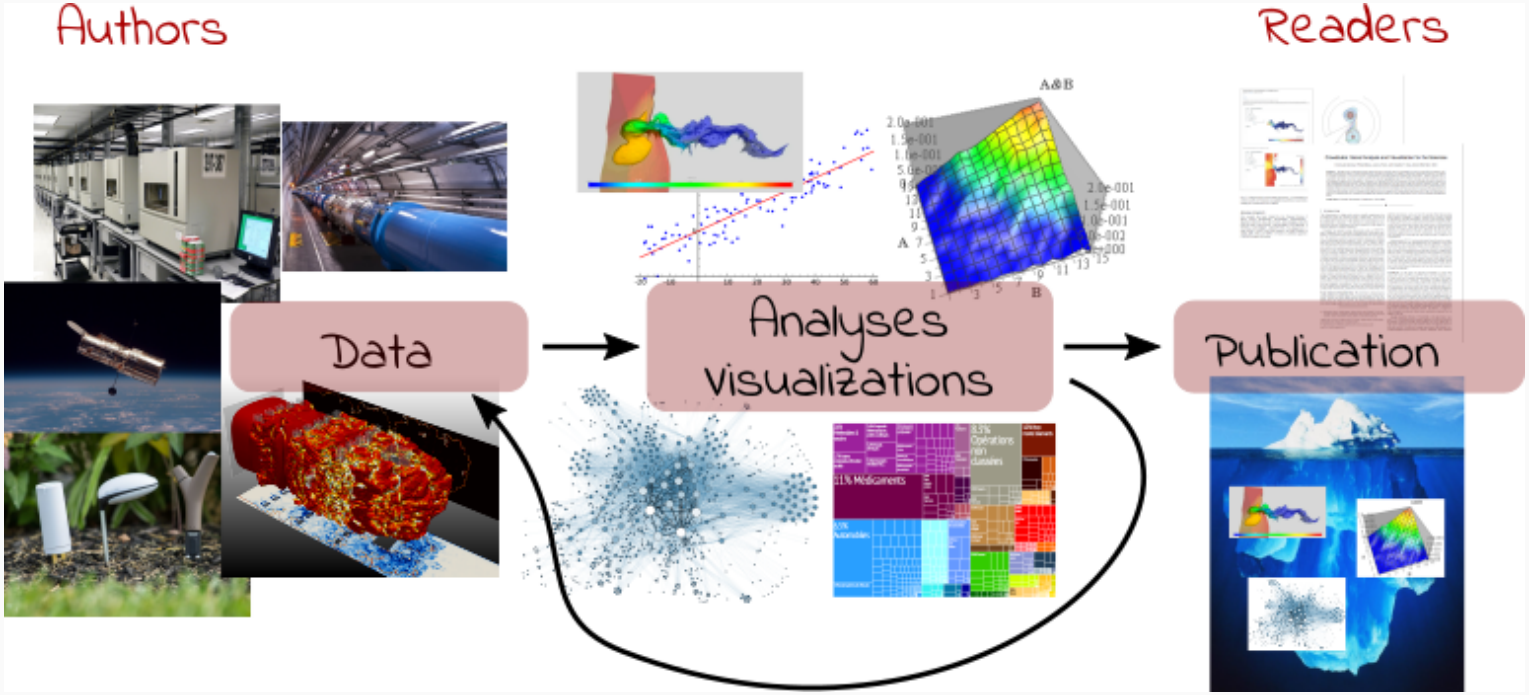


Different Reproducibility Concerns

Social Sciences, Oncology, ... methodology, statistics

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

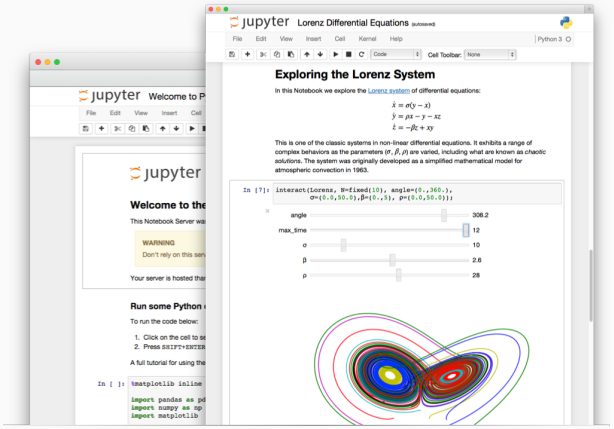
Computational fluid dynamics numerical issues



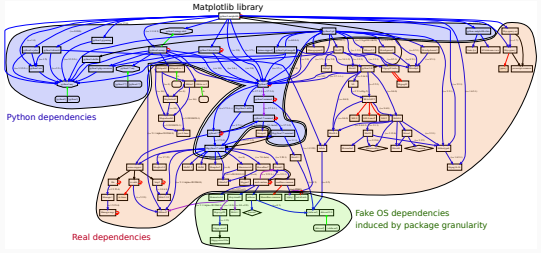
Reproducible Research = Bridging the Gap by working Transparently

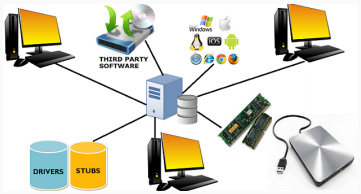
Existing Tools, Emerging Standards

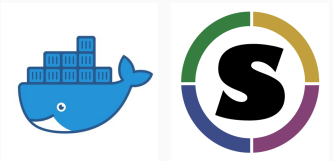
Notebooks and workflows



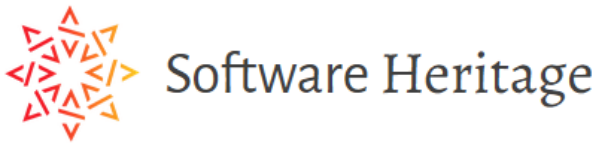
Software environments







Sharing platforms



Tool 1: Computational Notebooks/Litterate Programming

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

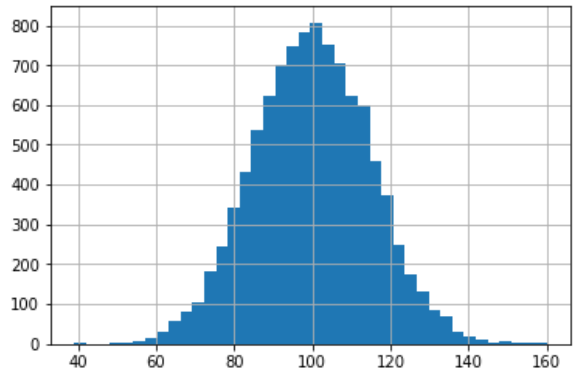
```
3.141592653589793
```

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

jupyter example_pi (autosaved)

File Edit View Insert Cell Kernel Widgets Help Hide Code Python 3

Code CellToolbar

Hide Prompts Hide Code Hide Outputs

```
# Un document computationnel
```

Hide Prompts Hide Code Hide Outputs

```
Mon ordinateur m'indique que $\pi$ vaut *approximativement*
```

In [1]:

Hide Prompts Hide Code Hide Outputs

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

```
3.141592653589793
```

Hide Prompts Hide Code Hide Outputs

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

In [2]:

Hide Prompts Hide Code Hide Outputs

```
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

Hide Prompts Hide Code Hide Outputs

```
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 $\pi$ (si ce n'est une constante de
normalisation... ☺).
```

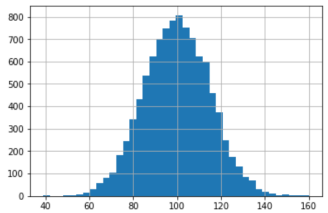
In [3]:

Hide Prompts Hide Code Hide Outputs

```
%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()
```



Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

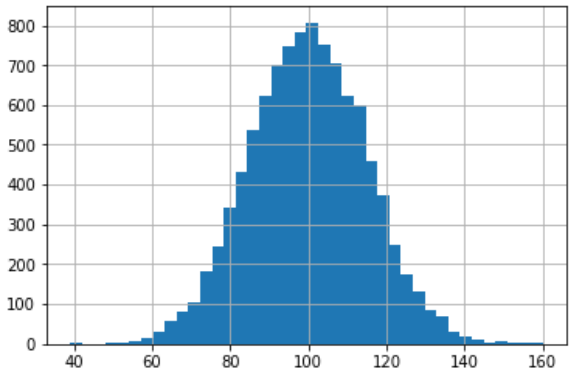
```
3.141592653589793
```

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
```

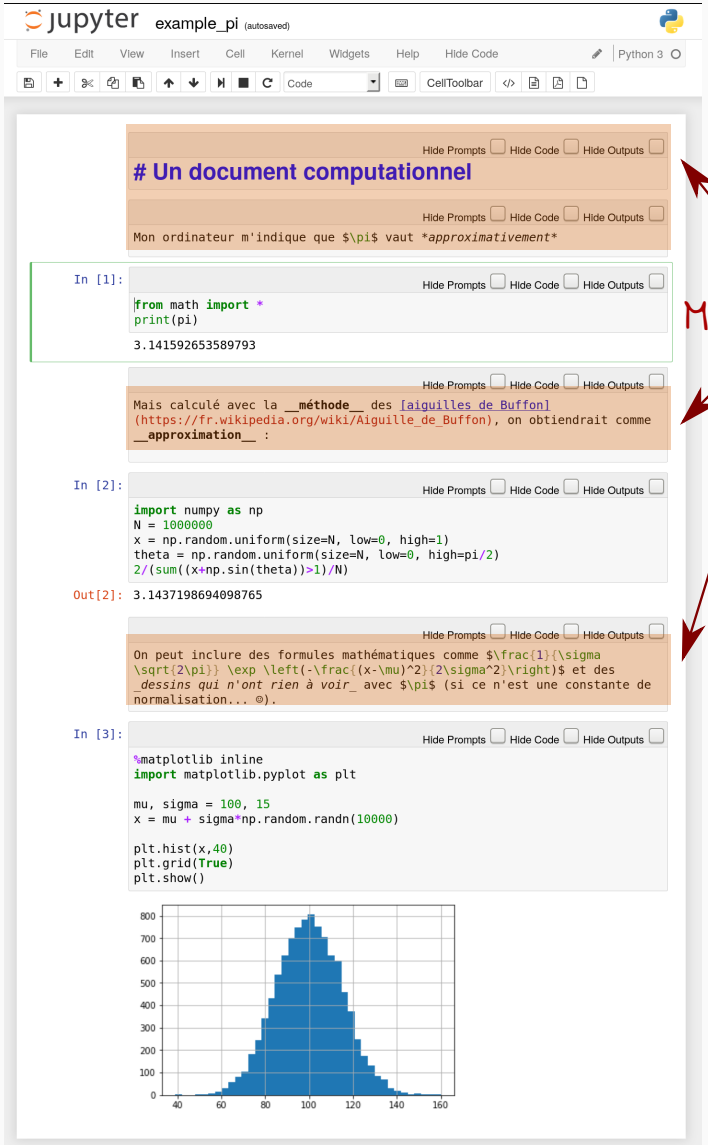
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec π (si ce n'est une constante de normalisation... ☺).



Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement

Document final



Mark Down

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

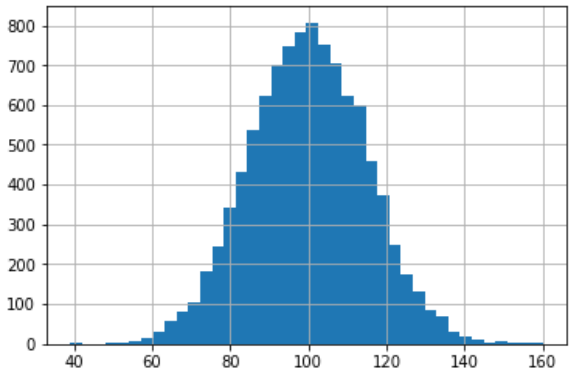
```
3.141592653589793
```

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)
```

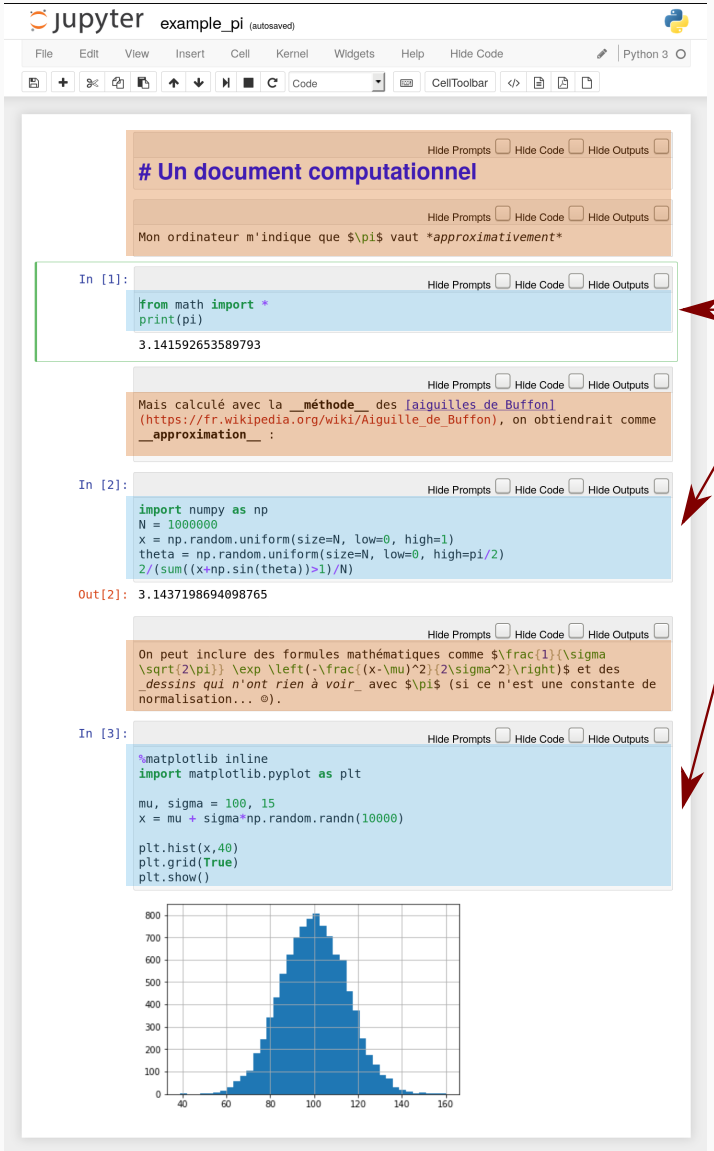
```
3.1437198694098765
```

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



Code

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

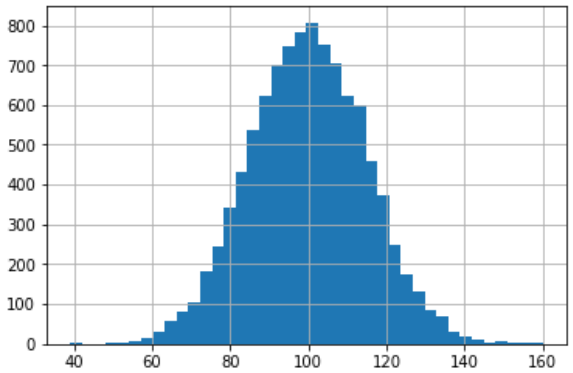
```
3.141592653589793
```

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

jupyter example_pi (autosaved)

File Edit View Insert Cell Kernel Widgets Help Hide Code Python 3

Code CellToolbar

Hide Prompts Hide Code Hide Outputs

```
# Un document computationnel
```

Hide Prompts Hide Code Hide Outputs

```
Mon ordinateur m'indique que $\pi$ vaut *approximativement*
```

In [1]:

Hide Prompts Hide Code Hide Outputs

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

```
3.141592653589793
```

Hide Prompts Hide Code Hide Outputs

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

In [2]:

Hide Prompts Hide Code Hide Outputs

```
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

Hide Prompts Hide Code Hide Outputs

```
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 $\pi$ (si ce n'est une constante de
normalisation... ☺).
```

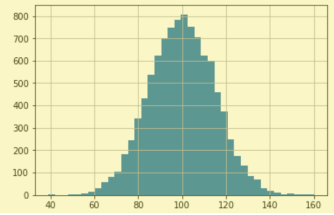
In [3]:

Hide Prompts Hide Code Hide Outputs

```
%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()
```



Résultats

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

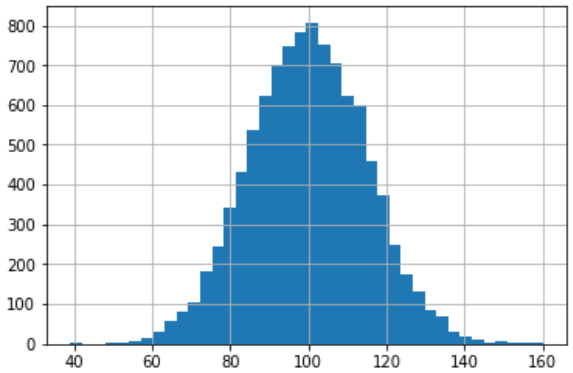
```
3.141592653589793
```

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

jupyter example_pi (autosaved)

File Edit View Insert Cell Kernel Widgets Help Hide Code Python 3

Code CellToolbar

Hide Prompts Hide Code Hide Outputs

```
# Un document computationnel
```

Hide Prompts Hide Code Hide Outputs

```
Mon ordinateur m'indique que $\pi$ vaut *approximativement*
```

In [1]:

Hide Prompts Hide Code Hide Outputs

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

```
3.141592653589793
```

Hide Prompts Hide Code Hide Outputs

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

In [2]:

Hide Prompts Hide Code Hide Outputs

```
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

Hide Prompts Hide Code Hide Outputs

```
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 $\pi$ (si ce n'est une constante de
normalisation... ☺).
```

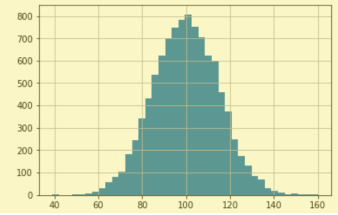
In [3]:

Hide Prompts Hide Code Hide Outputs

```
%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()
```



Export

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

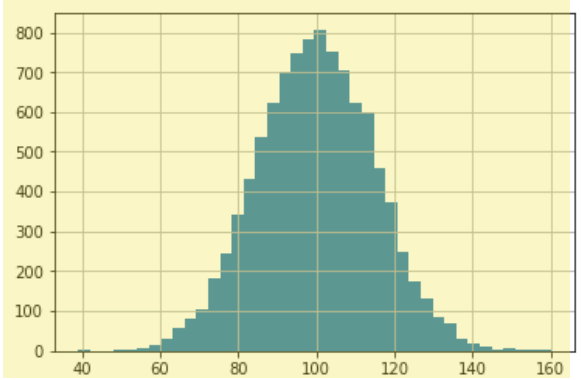
```
3.141592653589793
```

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

jupyter example_pi (autosaved)

File Edit View Insert Cell Kernel Widgets Help Hide Code Python 3

Code CellToolbar

Hide Prompts Hide Code Hide Outputs

```
# Un document computationnel
```

Hide Prompts Hide Code Hide Outputs

```
Mon ordinateur m'indique que $\pi$ vaut *approximativement*
```

In [1]:

Hide Prompts Hide Code Hide Outputs

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

```
3.141592653589793
```

Hide Prompts Hide Code Hide Outputs

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

In [2]:

Hide Prompts Hide Code Hide Outputs

```
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

Hide Prompts Hide Code Hide Outputs

```
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 $\pi$ (si ce n'est une constante de
normalisation... ☺).
```

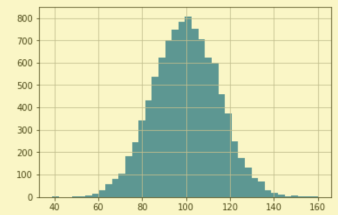
In [3]:

Hide Prompts Hide Code Hide Outputs

```
%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()
```



Export

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

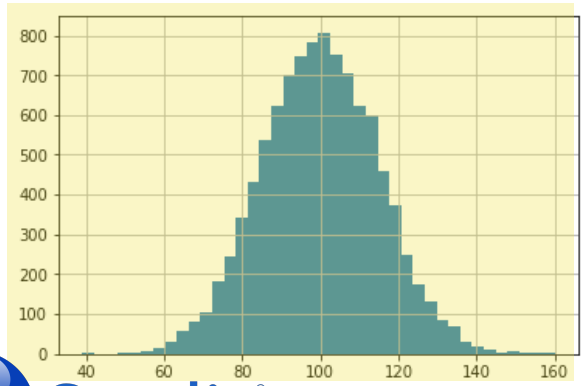
```
3.141592653589793
```

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 2: Fighting Software Environments Nightmare

What is hiding behind a simple

```
import matplotlib
```

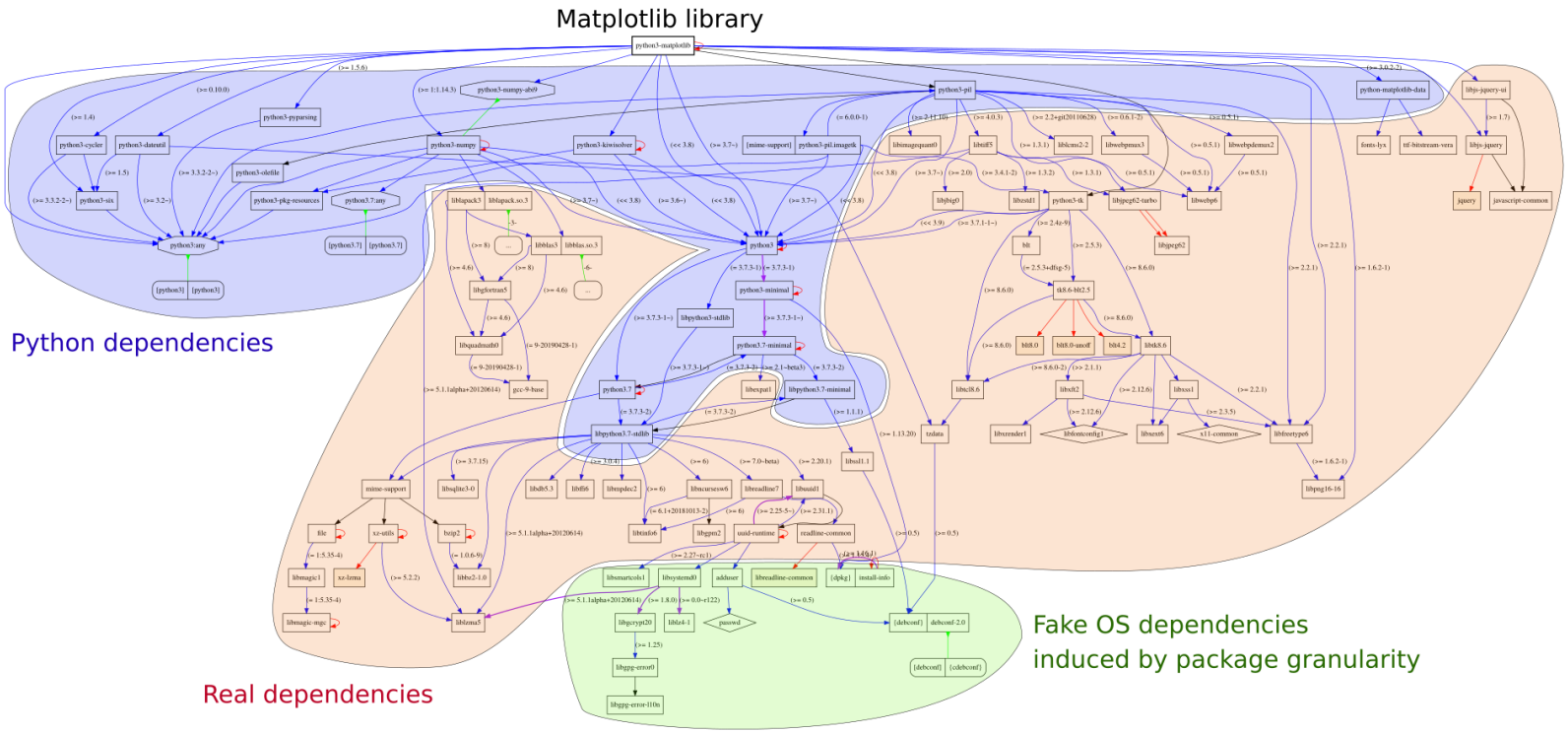
```
Package: python3-matplotlib
Version: 2.1.1-2
Depends: python3-dateutil, python-matplotlib-data (>= 2.1.1-2),
python3-pyparsing (>= 1.5.6), python3-six (>= 1.10), python3-tz,
libjs-jquery, libjs-jquery-ui, python3-numpy (>= 1:1.13.1),
python3-numpy-abi9, python3 (<< 3.7), python3 (>= 3.6~),
python3-cycler (>= 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

```
import matplotlib
```

`Package: python3-matplotlib`



Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

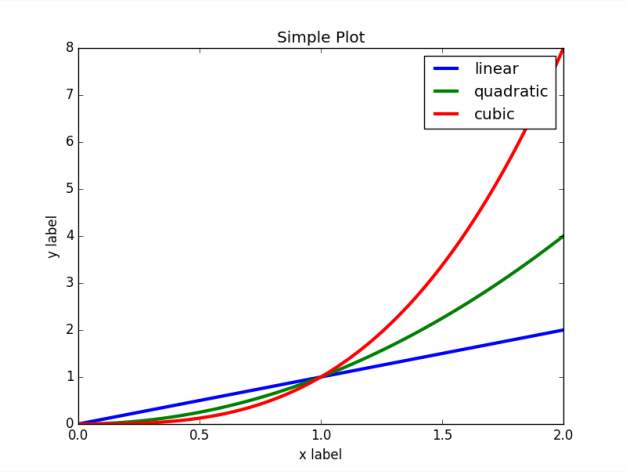
```
python2 -c "print(10/3)"
python3 -c "print(10/3)"
```

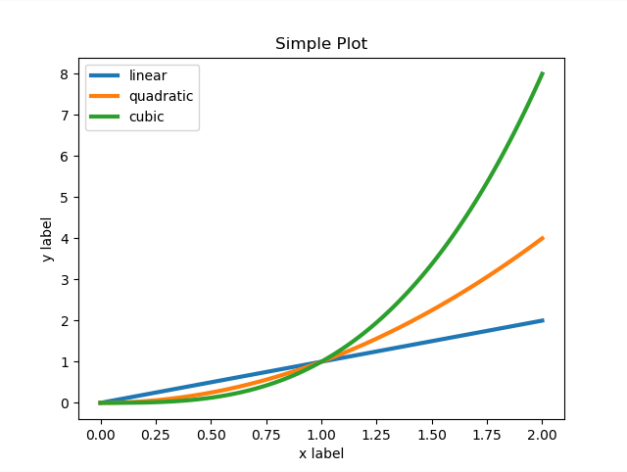
```
3
3.3333333333333335
```

Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

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

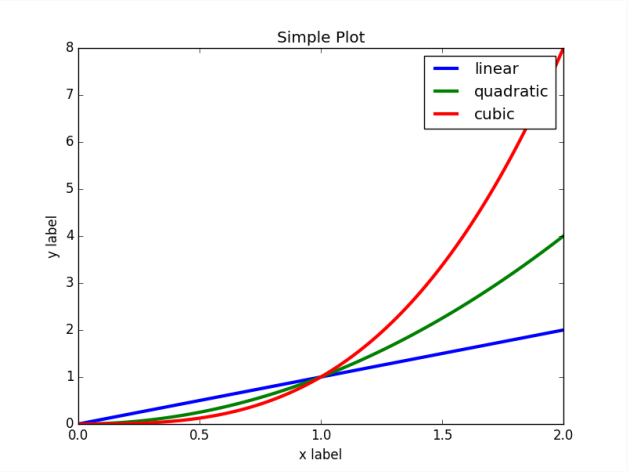


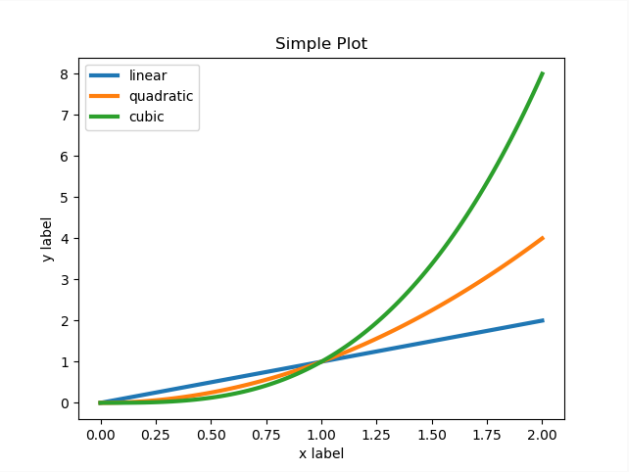


Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

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



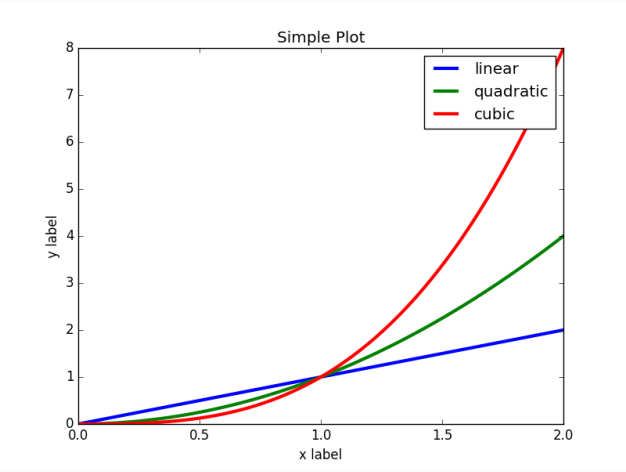


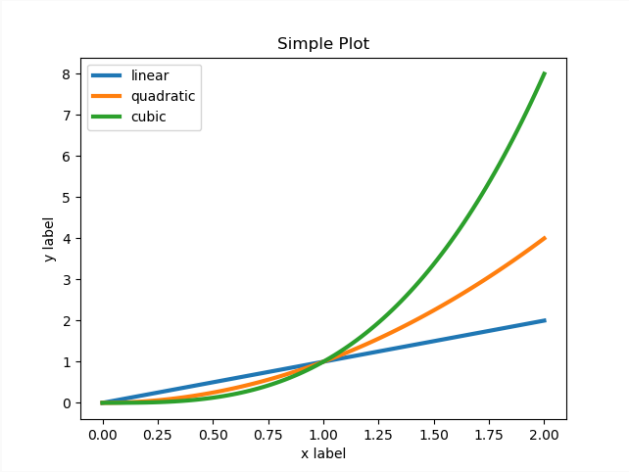
Cortical Thickness Measurements (PLOS ONE, June 2012): *FreeSurfer*: *differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*

Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

```
python2 -c "print(10/3)"
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.*

ReproZip

Guix

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

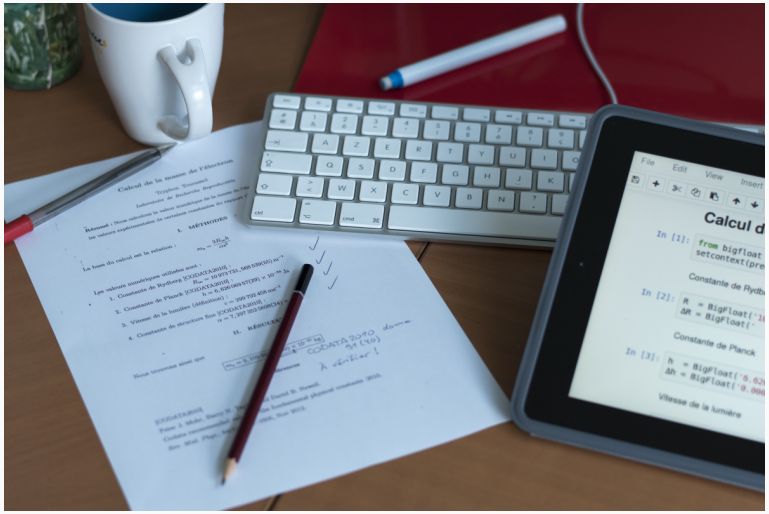


or

= awesome collaborations ≠ archive

Changing Research Practices

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



- Book on RR *Vers une recherche reproductible: Faire évoluer ses pratiques*
- **MOOC on RR** (3rd edition Feb. 2020)
- A new "Advanced RR" MOOC (Oct. 2020)
  - Software environment control (Docker)
  - Scientific workflow (snakemake)
  - Managing data (HDF5, archiving)

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

Changing Publishing Practices

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, Most Reproducible Paper Award...

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

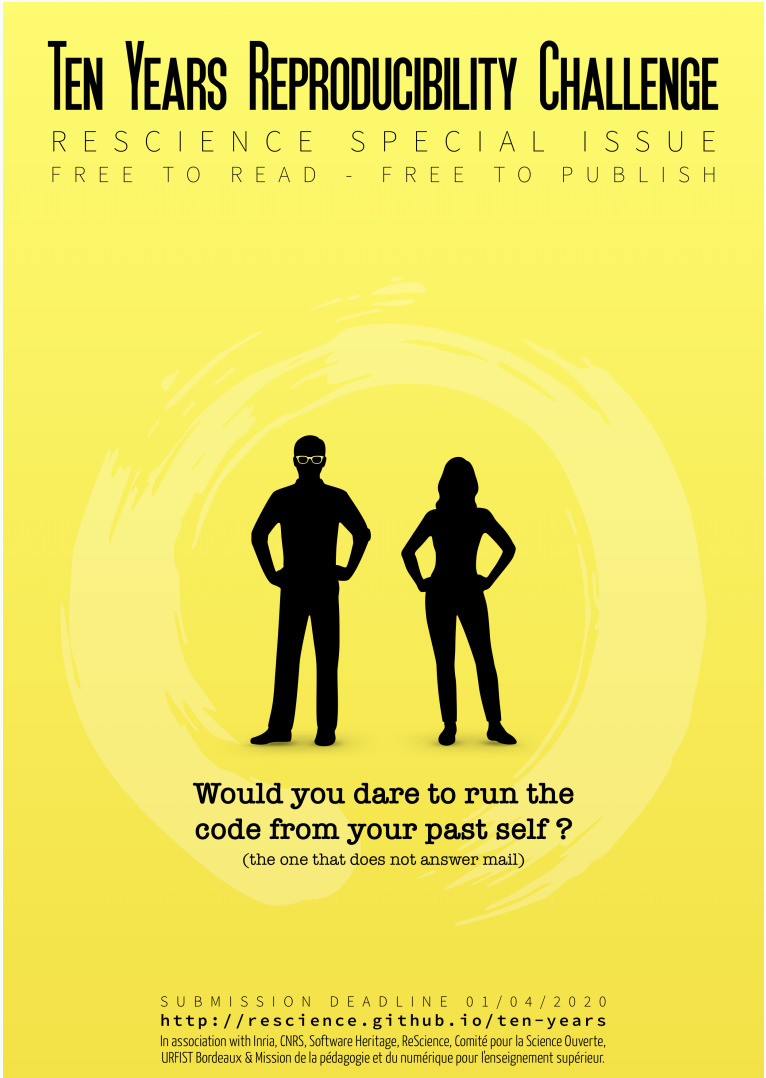
Reproducible Research = Rigor and Transparency

To err is human.

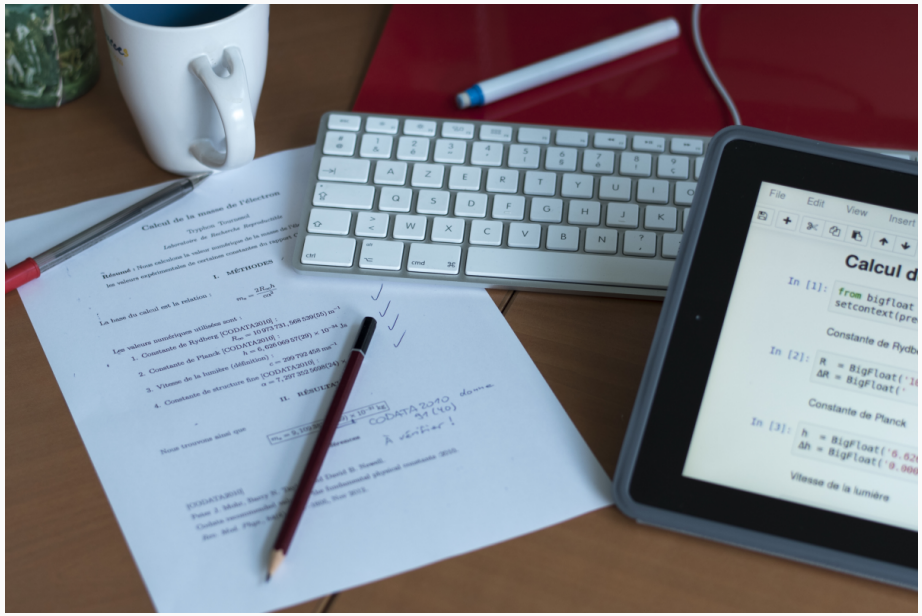
Good research requires time and resources

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

Some Advertising



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



3rd Edition: ≈ March 2020

A new MOOC: "Advanced RR"

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

October 2020 ?