

LABORATORY NOTEBOOKS AND REPRODUCIBLE RESEARCH

Arnaud Legrand



February 2021, Journées DKM, Irisa

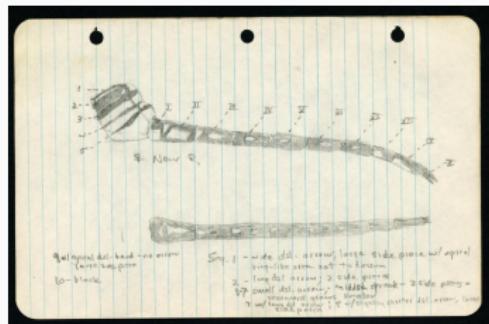


FIELD AND LABORATORY NOTEBOOKS

Social Sciences, Ecology, Biology



Chemistry, Physics, Biology



Robert Henry Gibbs, Jr.,
ichthyologist (1929 – 1988)



Emil Heinrich du Bois-Reymond,
electrophysiology (1818 – 1896)

PROS AND CONS

Pros

- Intuitive
- Convenient (lightweight)
- Sketching, equations, thoughts
- Difficult to temper

Cons

- Attachments
- Large amount of information
- Indexing and Searching
- Sharing, No backup

PROS AND CONS

Pros

- Intuitive
- Convenient (lightweight)
- Sketching, equations, thoughts
- Difficult to temper

Cons

- Attachments
- Large amount of information
- Indexing and Searching
- Sharing, No backup

Toward digital notebooks

Electronic Laboratory Notebooks and Laboratory Information Management Systems

- Batch and Sample management
- Instrument integration
- Stock management (e.g., chemical substances)
- Native handling of digital objects



PROS AND CONS

Pros

Cons

- Attachments

Example: eLabFTW

The screenshot shows the eLabFTW web application interface. At the top, there is a navigation bar with links for EXPERIMENTS, DATABASE, TEAM, SEARCH, and DOCUMENTATION. Below the navigation bar, the page title is "Experiments". There are several filters at the top: "Filter status" dropdown, "Order by" dropdown, "Sort" dropdown, and a page size selector "15". A "Create" button is located on the right.

The main content area displays a grid of experiment cards. Each card contains the experiment name, a brief description, status, date, project information, and a "details" icon.

- Some microscopy experiment: Next step: analyze data. Status: RUNNING, Date: 2020.04.30, Project: SIMA, Tags: video1.
- Another experiment: Status: SUCCESS, Date: 2020.04.30, Project: SIMA.
- Making fake experiments to do a screenshot: Status: SUCCESS, Date: 2020.04.30, Project: elabweb, elabftw.net.
- Looking at cells: Status: SUCCESS, Date: 2020.04.30.
- A failed experiment: Status: FAIL, Date: 2020.04.30, Project: sima.
- 无标题: Status: SUCCESS, Date: 2020.04.30.
- Senza titolo: Status: SUCCESS, Date: 2020.04.27.
- lock unlock: Status: SUCCESS, Date: 2020.04.22.
- mathjax I: Status: SUCCESS, Date: 2020.04.21.
- が無効になっています。: Status: SUCCESS, Date: 2020.04.20.
- Processing of Sample #392A: Status: SUCCESS, Date: 2020.04.16, Project: X.
- analysis of sample #2343 from this company: Status: NEED TO BE REDONE, Date: 2020.04.16, Project: X.
- mathjax test $\frac{d}{dx} x\dot{x} = \dot{x}^2 + x\ddot{x}$: Status: SUCCESS, Date: 2020.04.16.
- Model test: Status: RUNNING, Date: 2020.04.07, Project: null, non, sed, omnis, sit.
- Qui pariatur aut architecto fuga.: Status: NEED TO BE REDONE, Date: 2020.03.27, Project: labore, minus, inventore, qui, ta'g.

At the bottom center, there is a "NEXT PAGE" button.

PROS AND CONS

Pros

Cons

- Attachments

Example: eLabFTW

Intuitive

eLabFTW EXPERIMENTS DATABASE TEAM SEARCH DOCUMENTATION

Experiments

Back to listing

2020.04.30 Success Team User

project SIMA video1

Some microscopy experiment

Next step: analyze data

Goal:

Show what an experiment looks like for elabftw.net.

Sample #	Concentration	Absorbance	Density	Something
1	20 μM	0.239	3.4	102
2	30 μM	0.193	4.1	1339

You can have colors too.

And use **bold**, *italics*, underline like you would do in a normal text processing application. Have some mathjax: $x^y z$ is x^{y^z} , and $x^y \{y^z\}$ is x^{y^z} .

This is a list:

- List item 1
- List item 2
- Now show an image in the text



2/11

WHY DO WE DO THIS ?

Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

WHY DO WE DO THIS ?

Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex

Authors



WHY DO WE DO THIS ?

Different Reproducibility Concerns in Modern Science

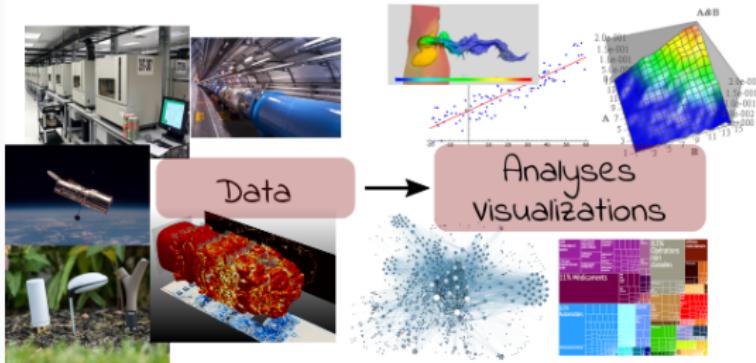
Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex

Authors



WHY DO WE DO THIS ?

Different Reproducibility Concerns in Modern Science

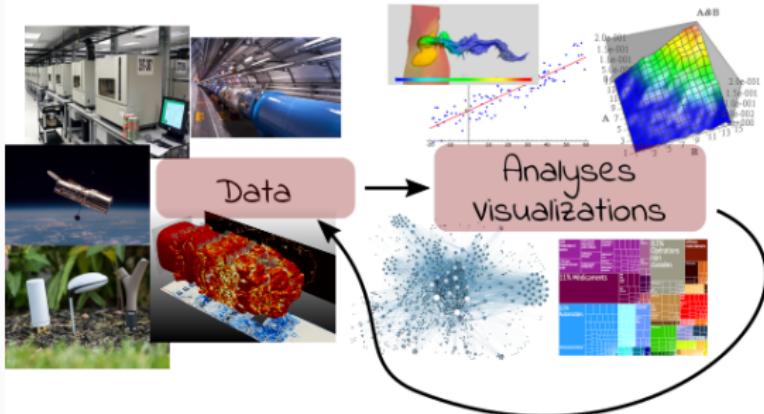
Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex

Authors



WHY DO WE DO THIS ?

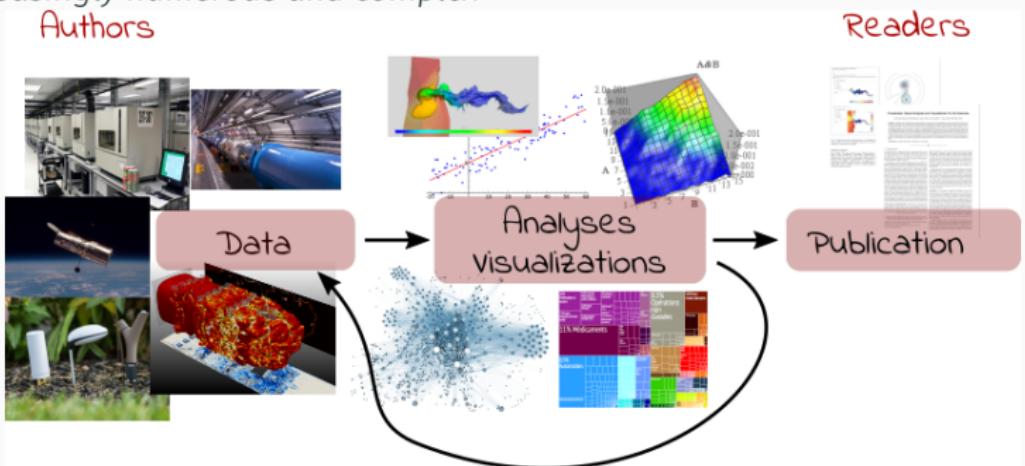
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



WHY DO WE DO THIS ?

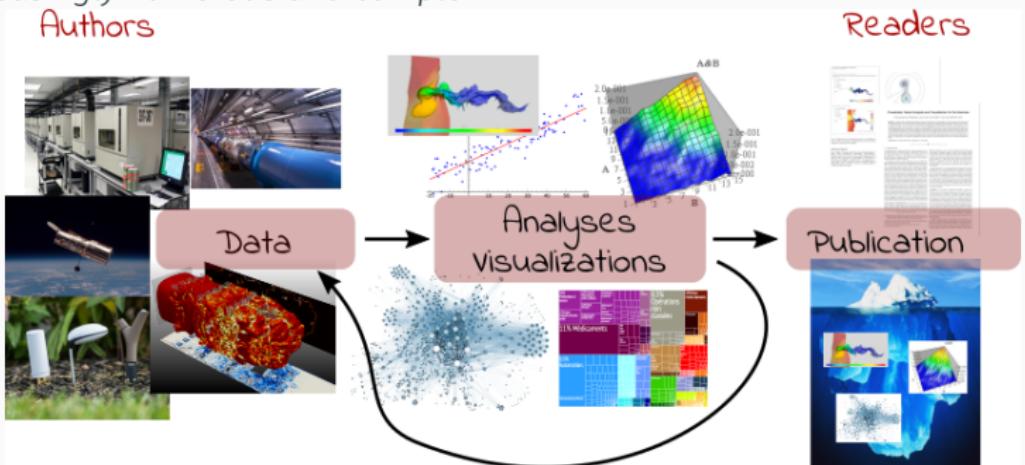
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



WHY DO WE DO THIS ?

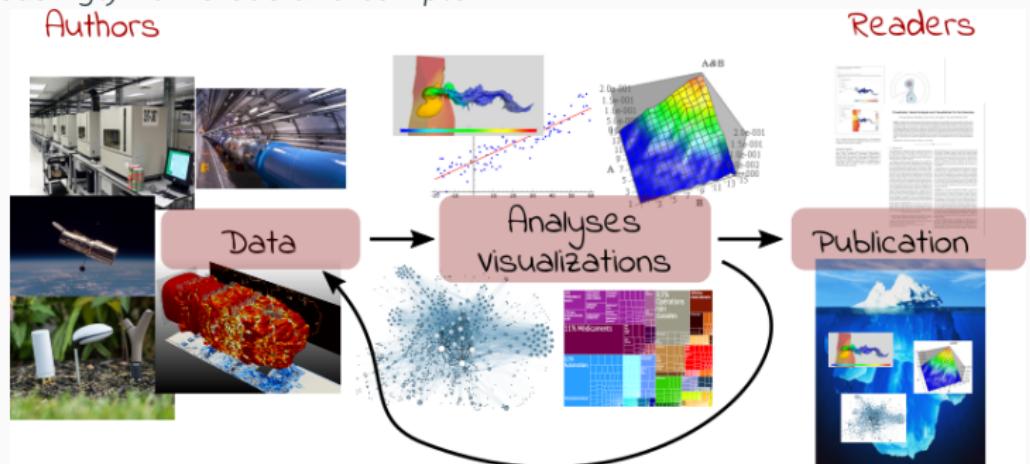
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

The processing steps between raw observations and findings have gotten increasingly numerous and complex



Reproducible Research = Bridging the Gap by working Transparently

REPRODUCIBLE RESEARCH PRACTICES

"REPRODUCIBLE RESEARCH": FIRST APPEARANCE

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 meaning 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).

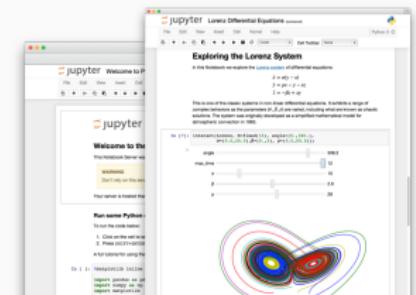
- make incremental improvements in electronic-document software
- seek partners for broadening standards (and making incremental improvements).

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.

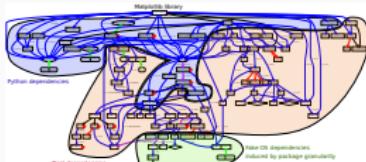
Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display 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 systems. Naturally we cannot include licensed software, but with the exception

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

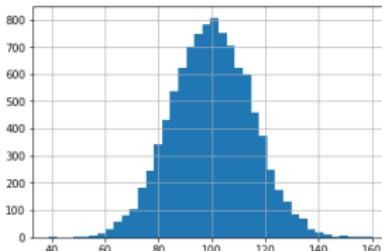
3.141592653589793

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 window titled "example_ip" (modified). It contains three code cells:

- In [1]:**

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

 Output: Mais calculé avec la [méthode des aiguilles de Buffon](#) (https://fr.wikipedia.org/wiki/Aiguille_de_Buffon), on obtient d'abord 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
```

 Output: 3.1437198694998765
- 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()
```

 Output: A histogram showing a normal distribution centered at 100 with a standard deviation of 15.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

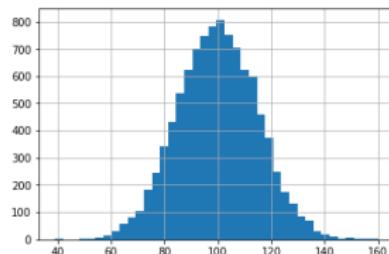
3.141592653589793

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
```

3.1437198694998765

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

Mon ordinateur m'indique que $\pi$ 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 obtient aussi comme approximation : 3.141592653589793

In [2]: import numpy as np N = 1000000 x = np.random.uniform(size=N, low=0, high=2) theta = np.random.uniform(size=N, low=0, high=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 $\pi$ (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()
```

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

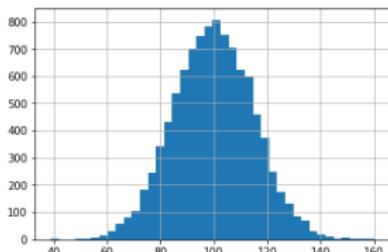
3.141592653589793

Mais calculé avec la [méthode des aiguilles de Buffon](#), on obtient 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

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]:** Calculates a numerical approximation of pi using a Monte Carlo simulation with 1000000 points.
- In [3]:** Plots a histogram of 100 random numbers between 0 and 1, showing a bell-shaped distribution centered at 0.5.

Annotations with red arrows point from the text "Code" in the first section to the code blocks in the notebook, and from the text "Document initial dans son environnement" to the top of the screenshot.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

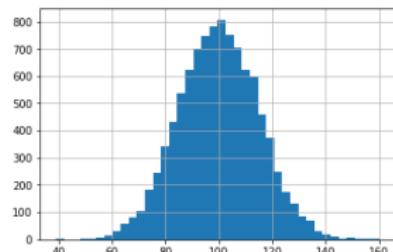
3.141592653589793

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 méthode des épingles de Buffon (https://fr.wikipedia.org/wiki/Algille\_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  $\pi$  (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

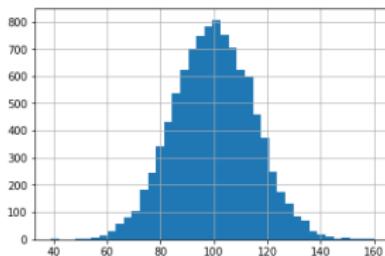
3.141592653589793

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

A screenshot of a Jupyter Notebook interface. The top bar shows 'jupyter example_pi' and 'Python 3'. The notebook has three cells:

- In [1]:** Prints the value of pi: `3.141592653589793`. A note says: "Mon ordinateur m'indique que \$\pi\$ vaut approximativement".
- In [2]:** Calculates the area under a curve using the Monte Carlo method. It imports numpy, generates random numbers, calculates theta, and sums the results. The output is `3.1437198694098765`. A note says: "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]:** Plots a histogram of 100,000 random numbers. The x-axis ranges from 40 to 160, and the y-axis ranges from 0 to 800. The distribution is approximately normal.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

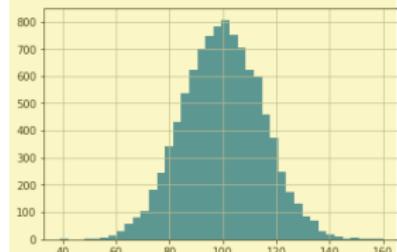
Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
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 it with theBuffon's needle method.
- In [2]:** Generates random points (x, y) and calculates the ratio of points below the unit circle to the total number of points, which approximates pi.
- In [3]:** Plots a histogram of 100,000 random numbers between -1 and 1, showing a symmetric bell-shaped distribution centered at 0.

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

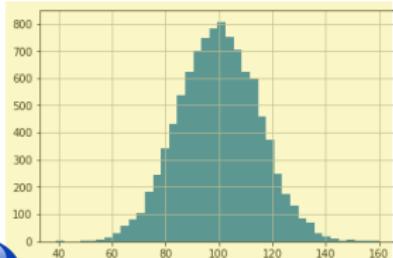
Mais calculé avec la **méthode des aiguilles de Buffon**, on obtiendrait comme approximation :

```
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

```
import matplotlib
```

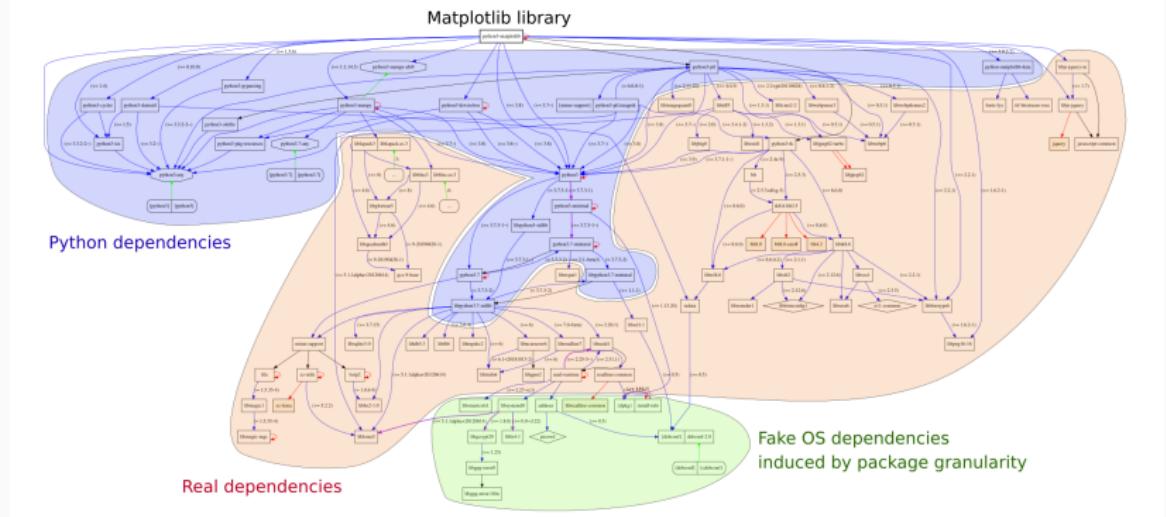
Package: python3-matplotlib
Version: 2.1.1-2
Depends: python3-dateutil, python-matplotlib-data (>= 2.1.1-2),
python3-pyparsing (>= 1.5.6), python3-six (>= 1.10), python3-tz,
libjs-jquery, libjs-jquery-ui, python3-numpy (>= 1:1.13.1),
python3-numpy-abi9, python3 (<< 3.7), python3 (>= 3.6~),
python3-cycler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

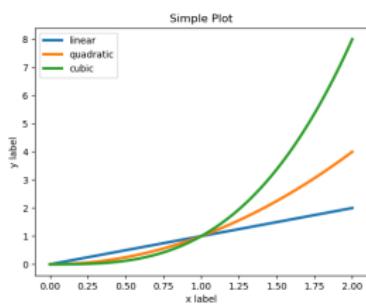
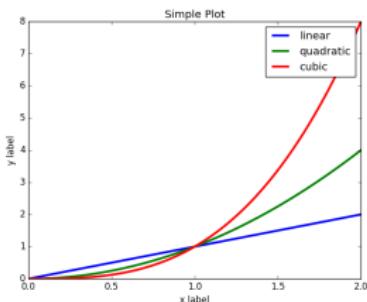
```
python2 -c "print(10/3)"  
python3 -c "print(10/3)"
```

3
3.333333333333335

TOOL 2: FIGHTING SOFTWARE ENVIRONMENTS NIGHTMARE

Python and its rapidly evolving environment

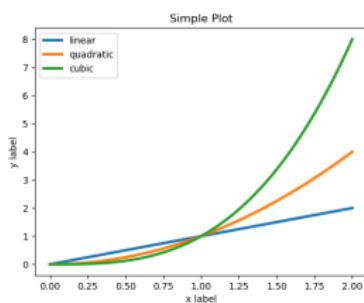
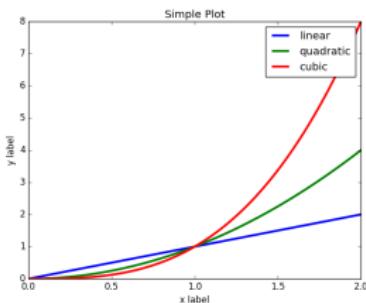
```
python2 -c "print(10/3)"  
python3 -c "print(10/3)"
```



TOOL 2: FIGHTING SOFTWARE ENVIRONMENTS NIGHTMARE

Python and its rapidly evolving environment

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

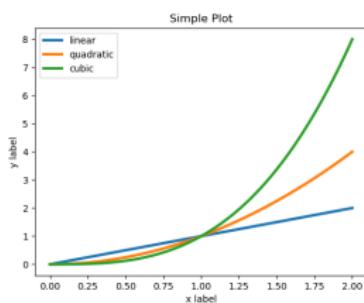
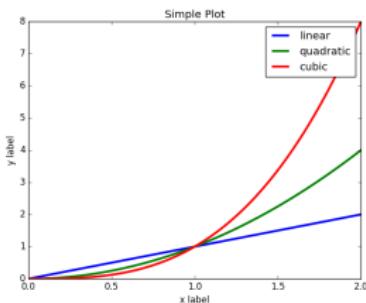


Cortical Thickness Measurements (PLOS ONE, June 2012): *FreeSurfer: differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*

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.*



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

CONCLUSION

CHANGING RESEARCH PRACTICES (RIGOR AND TRANSPARENCY)

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 RESEARCH PRACTICES (RIGOR AND TRANSPARENCY)

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

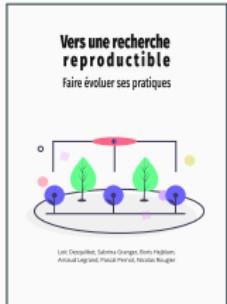
1. I will teach my graduate students about reproducibility
2. All our research code (and writing) is under version control
3. We will always carry out verification and validation
4. We will share data, plotting script & figure under CC-BY
5. We will upload the preprint to arXiv at the time of submission 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 Research Practices

1. Train yourself and your students: RR, statistics, experiments
 - Soft. Engineering, Statistics, and Reproducible Research in the curricula
 - 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 RESOURCES ☺



A non-technical introduction to reproducibility issues (in French)

- Loïc Desquillet, Sabrina Granger, Boris Hejblum, Pascal Pernot, Nicolas Rougier

MOOC Reproducible Research: Methodological principles for a transparent science, Learning Lab Inria

- Konrad Hinsen, Christophe Pouzat, Alexandre Hocquet
- 3rd Edition: March 2020 – March 2022
- MOOC RR "Advanced" (2021?)

