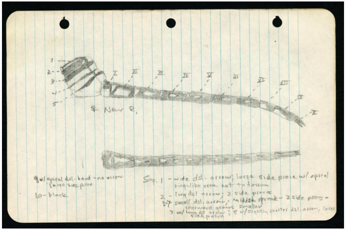
Laboratory Notebooks and Reproducible Research

Arnaud Legrand

February 2021, Journées DKM, Irisa

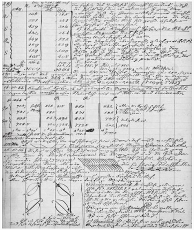
Field and Laboratory notebooks

Social Sciences, Ecology, Biology



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

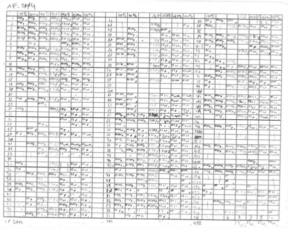
Chemistry, Physics, Biology



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

Field and Laboratory notebooks

Social Sciences, Ecology, Biology



Marie-Claude Quidoz, Centre d'Ecologie Fonctionnelle et Evolutive

Chemistry, Physics, Biology

Contemporary laboratory notebook in Neurology

Essentially no evolution throughout the last century

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

- 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

CENTER FOR OPEN SCIENCE

Pros and Cons

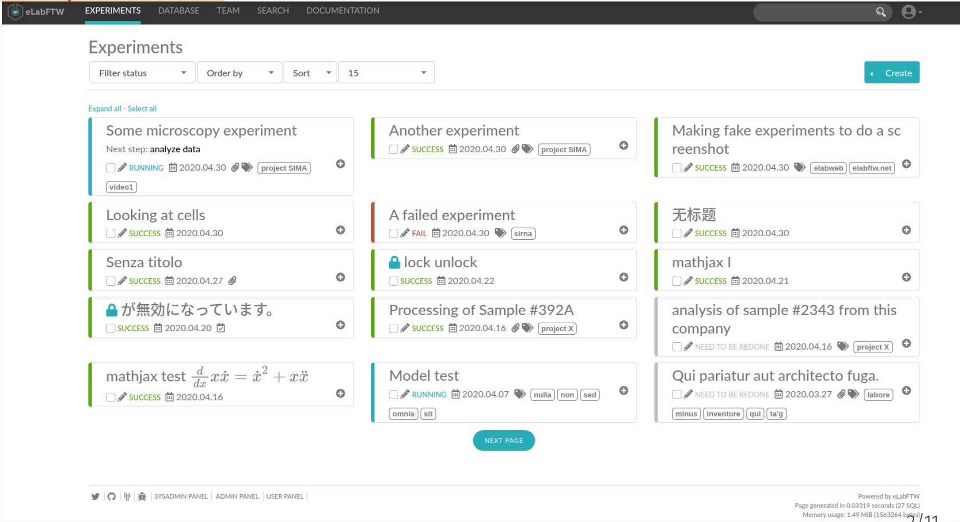
Pros

- Intuitive

Cons

- Attachments

Example: eLabFTW



Pros and Cons

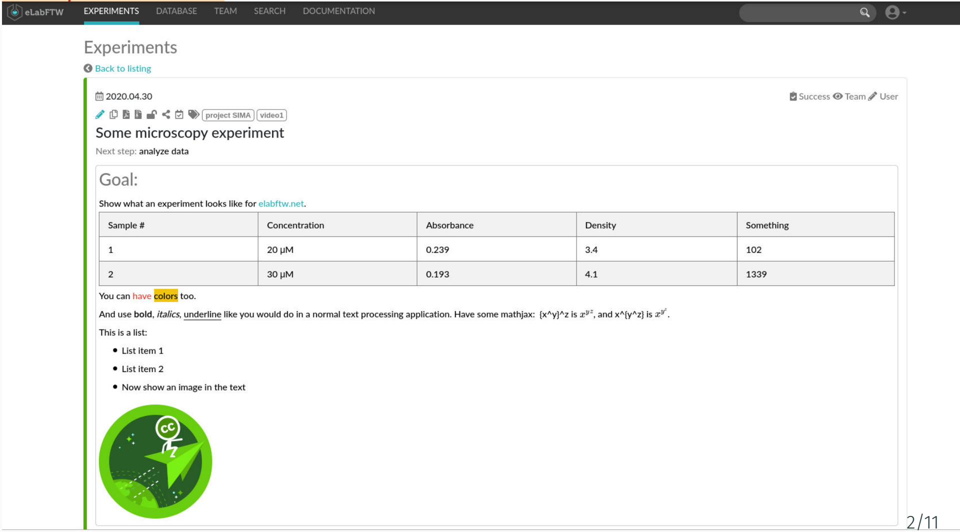
Pros

- Intuitive

Cons

- Attachments

Example: eLabFTW



Why do we do this ?

Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, … methodology, statistics

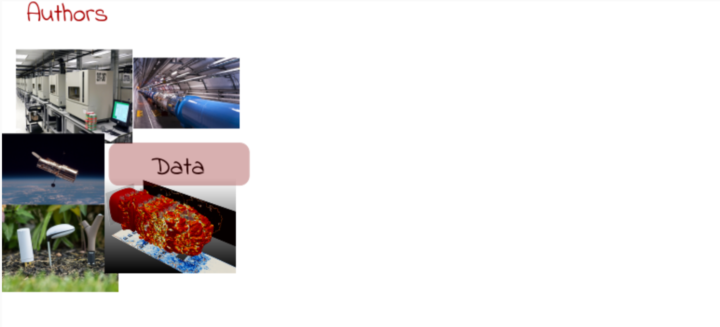
Genomics software engineering, computational reproducibility, provenance

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*



Why do we do this ?

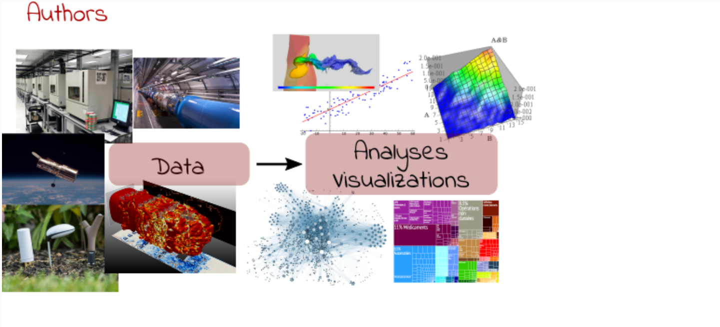
Different Reproducibility Concerns in Modern Science

Social Sciences, Oncology, ... methodology, statistics

Genomics software engineering, computational reproducibility, provenance

Computational fluid dynamics numerical issues

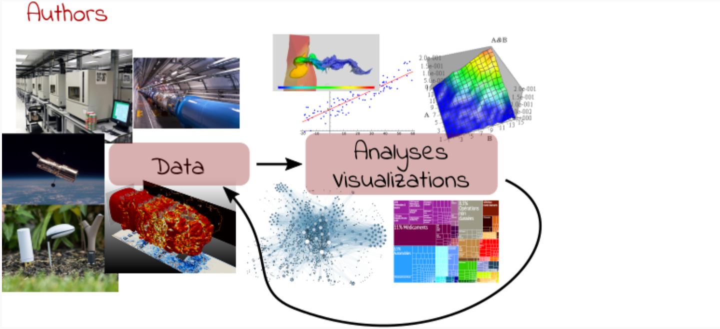
The processing steps between raw observations and findings have gotten increasingly numerous and complex



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*

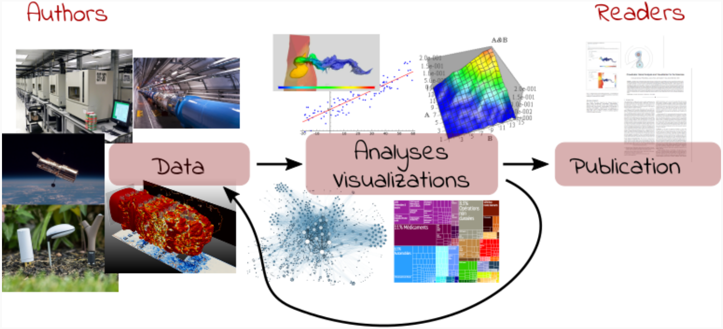


Why do we do this ?

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

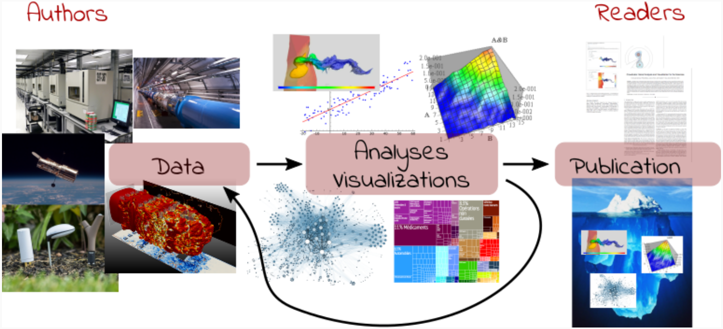


Why do we do this ?

Different Reproducibility Concerns in Modern Science

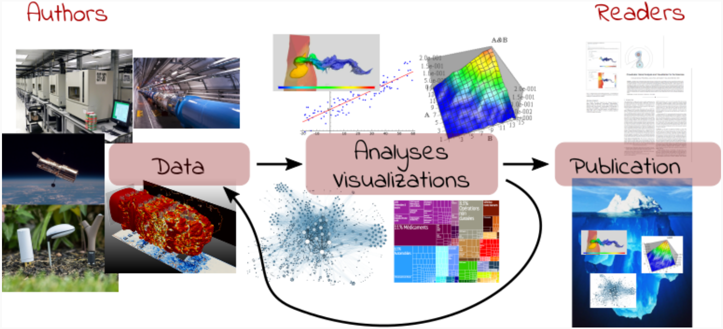
- **Social Sciences, Oncology, ...** methodology, statistics
- **Genomics** software engineering, computational reproducibility, provenance
- **Computational fluid dynamics** numerical issues

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



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 provid[illegible]finition of reproducibility in computa-

In 1990, we set this sequence of goals:

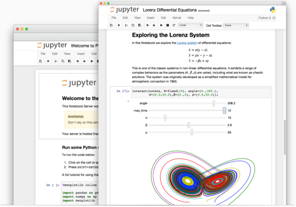
- Learn how to merge a publication with its underlying computational analysis.
- Teach researchers how to prepare a document in a form where they themselves can reproduce their own research results a year or more later by "pressing a single button".
- Learn how to leave finished work in a condition where 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.

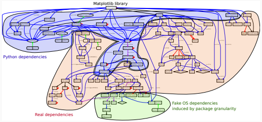
Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display 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 commerical television. We include all our plot software as well as freely available software from many sources, including compilers and the LaTeX word processing system. Naturally

Existing Tools, Emerging Standards

Notebooks and workflows



Software environments



Sharing platforms

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

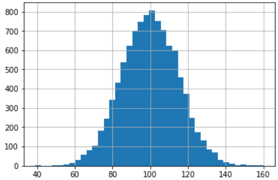
```
3.141592653589793
```

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

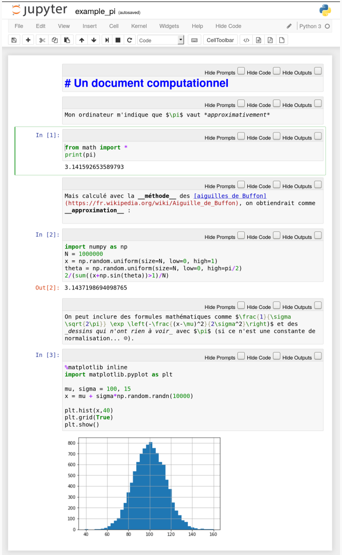
```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

```
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... ☺).



Document initial dans son environnement



jupyter example_pi (autosaved)

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

```
# 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 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

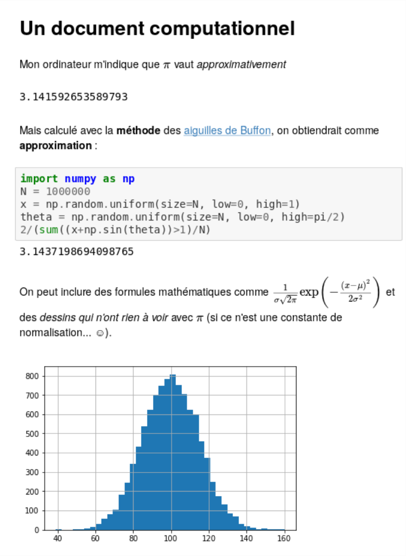
```
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp \left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des _dessins qui n'ont rien à voir_ avec $\pi$ (si ce n'est une constante de normalisation... ☺).
```

In [3]:
```
%matplotlib inline
import matplotlib.pyplot as plt

mu, sigma = 100, 15
x = mu + sigma*np.random.randn(10000)

plt.hist(x,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** :

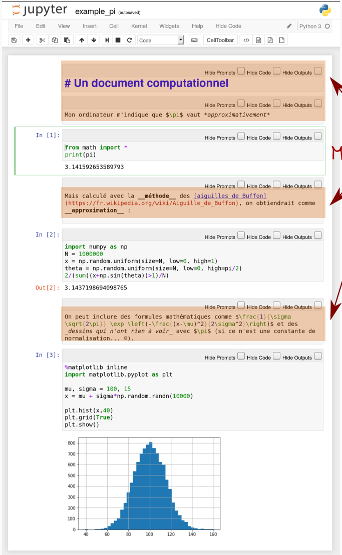
```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

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... ☺).

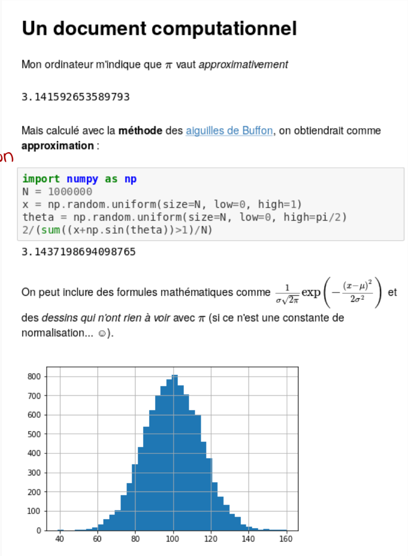
Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement



Mark Down

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** :

```python
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)
```

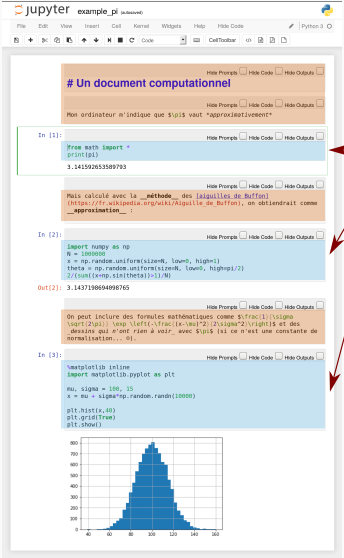
3.1437198694098765

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... ☺).

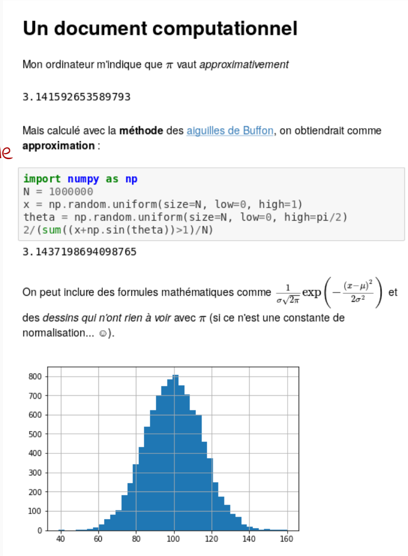
Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement

Document final

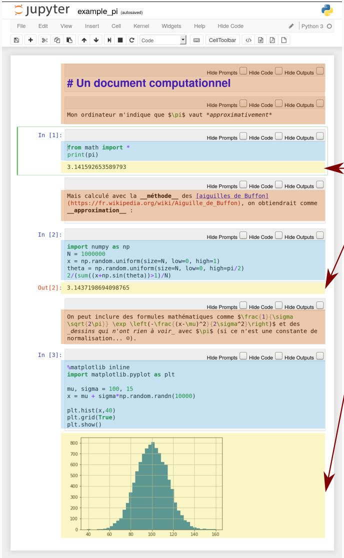


Code



Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement



Résultats

Document final

Un document computationnel

Mon ordinateur m'indique que π vaut *approximativement*

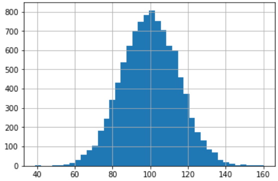
```
3.141592653589793
```

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

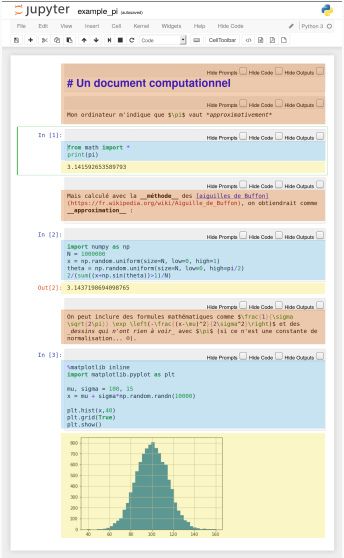
```python
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x+np.sin(theta))>1)/N)
```

```
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... ☺).

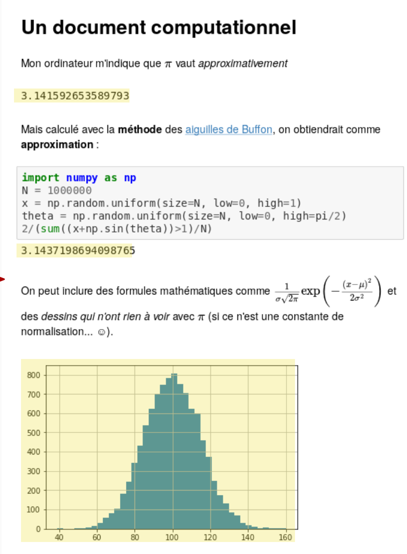


Document initial dans son environnement



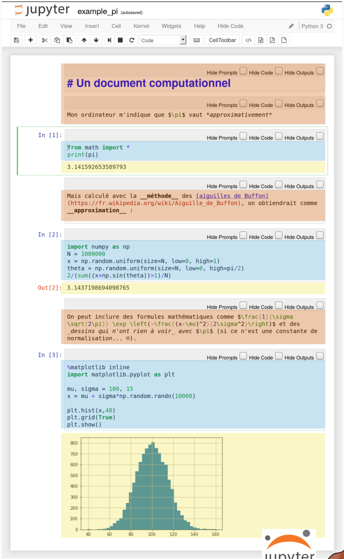
Export

Document final



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

```
# 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 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... ☺).
```

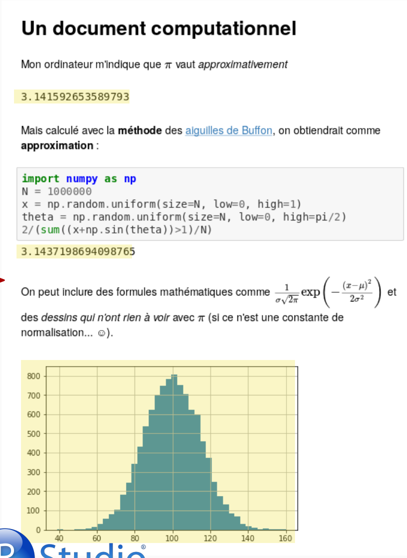
In [3]:
```
%matplotlib inline
import matplotlib.pyplot as plt

mu, sigma = 100, 15
x = mu + sigma*np.random.randn(10000)

plt.hist(x,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... ☺).

What is hiding behind a simple

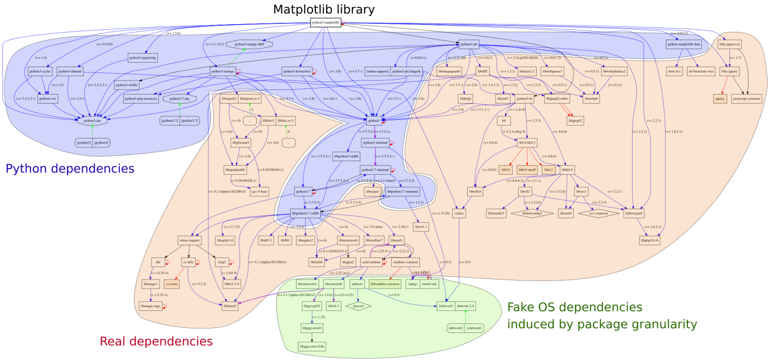
```python
import matplotlib
```

```
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)
```

What is hiding behind a simple

```python
import matplotlib
```

Package: python3-matplotlib



Python and its rapidly evolving environment

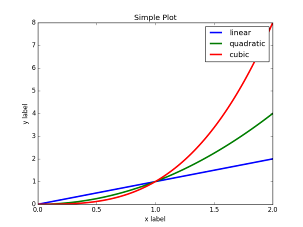
```
python2 -c "print(10/3)"
python3 -c "print(10/3)"
```

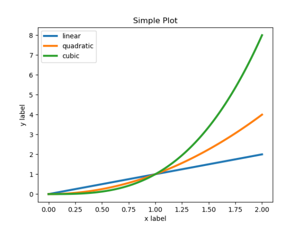
```
3
3.3333333333333335
```

Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

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

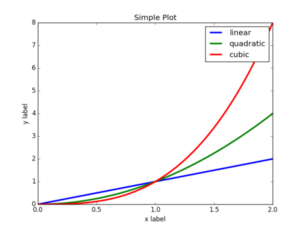


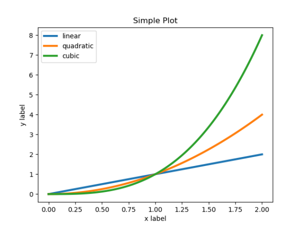


Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

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



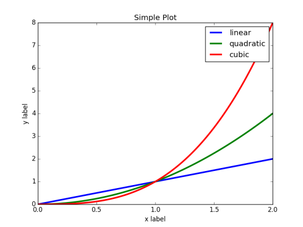


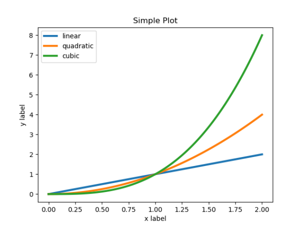
Cortical Thickness Measurements (PLOS ONE, June 2012): *FreeSurfer: differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*

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 $\neq$ archive

Conclusion

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 Desquilbet, 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?)