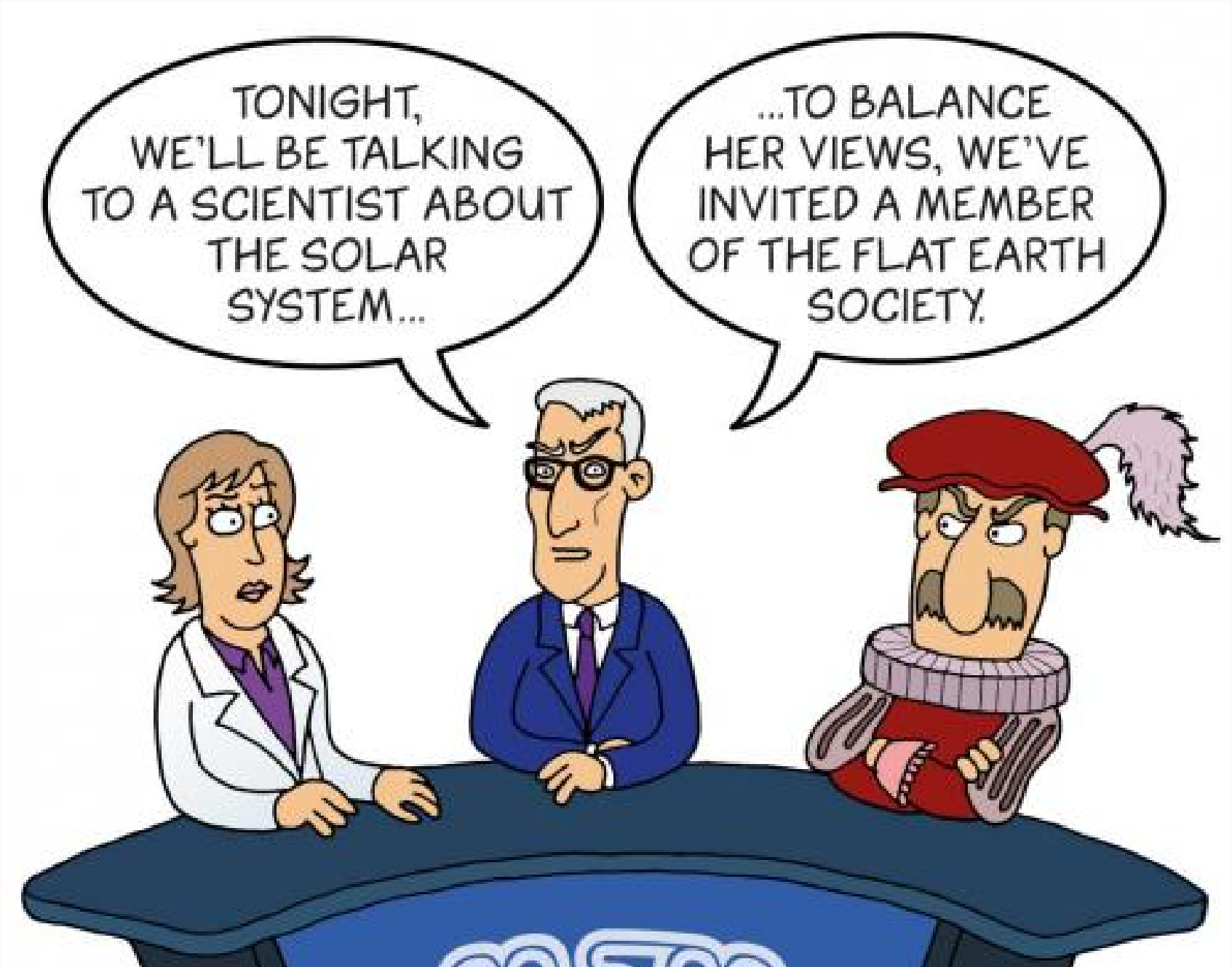
# Reproducible Research and Benchmarking

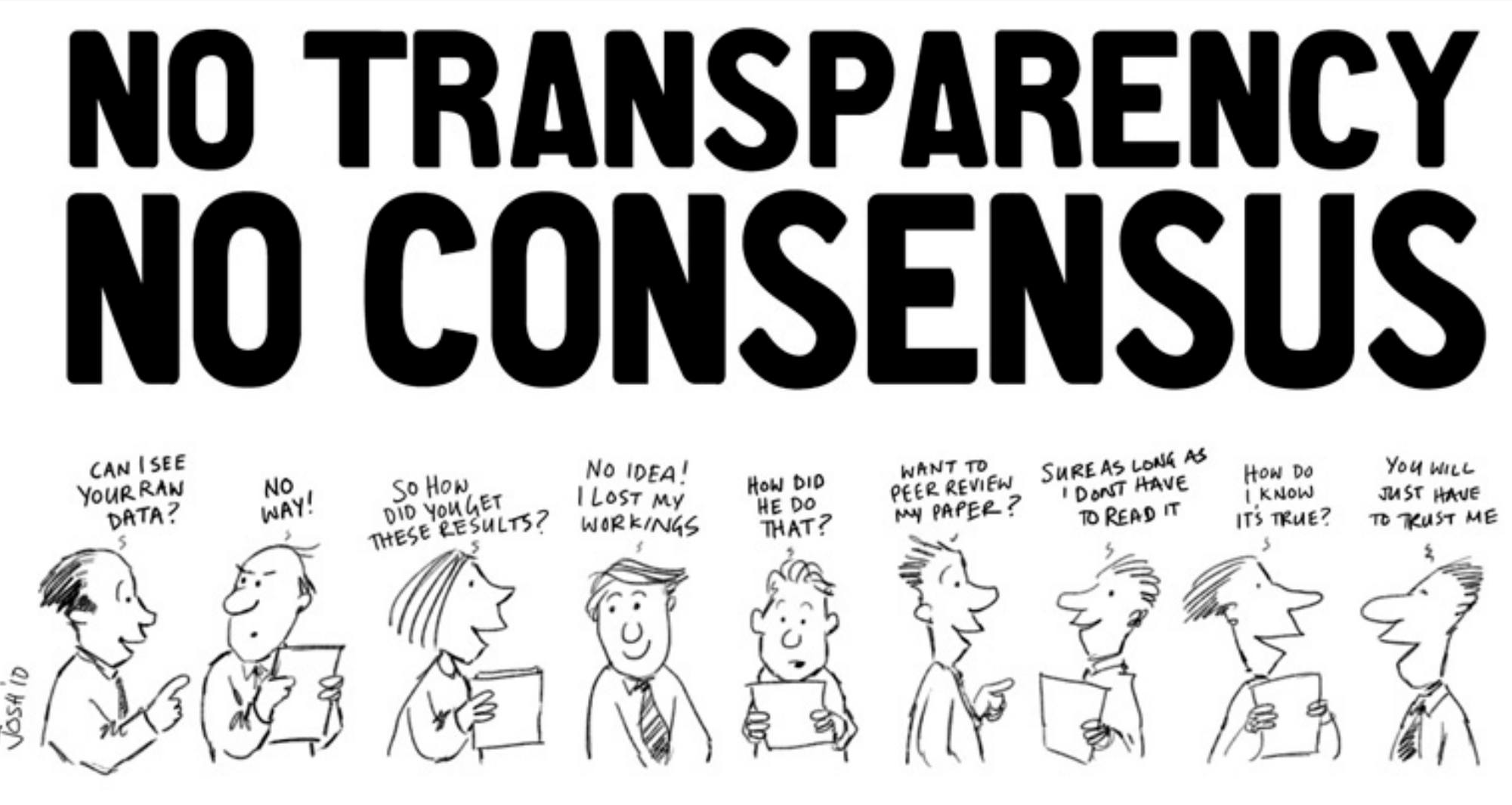
Arnaud Legrand

Inria foresight seminar
Optimization, Machine Learning and Statistical Methods
October 2023

# Scientific Consensus vs. Democracy and Freedom of Speech



# Scientific Consensus



# Reproducible Research ∼ Open Science ?

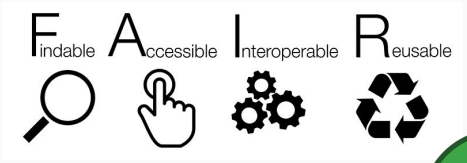
Plan National pour la Science Ouverte (BSN ⇝ CoSO)

- CNRS, Inria, INRAE, ...
- Many flavors: *Citizen Science*

Main pillars:

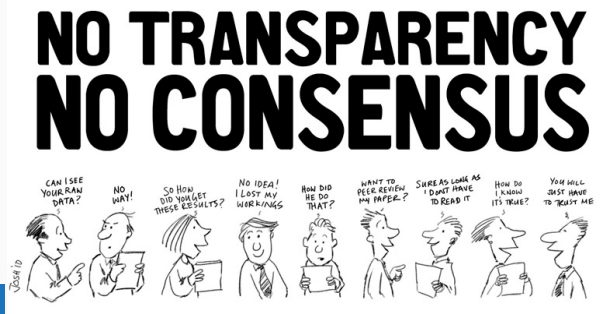
1. Open access

2. Open data




3. Open source
   - *Open hardware*

4. **Open methodology** (Reproducible Research)
   - *Open-notebook science*
   - *Open science infrastructures*
5. Open peer review (avoid collusion)
6. Open educational resources


Obviously making code/data available for the reproduction of results from published papers has become the new norm

# Inria *Learning* team's reproducibility concerns (1/2)

Statistics (Misuse, *external* reproducibility issues)

- **CELESTE**: bound false positive error rate while maximizing power, robustness
- **MODAL**: ML performance assessment (~~misleading conclusions~~)

# Inria *Learning* team's reproducibility concerns (1/2)

**Statistics** (Misuse, *external* reproducibility issues)

- **CELESTE**: bound false positive error rate while maximizing power, robustness
- **MODAL**: ML performance assessment (~~misleading conclusions~~)

**Mathematics** Are mathematicians concerned with reproducibility ?

# Inria *Learning* team's reproducibility concerns (1/2)

**Statistics** (Misuse, *external* reproducibility issues)

- **CELESTE**: bound false positive error rate while maximizing power, robustness
- **MODAL**: ML performance assessment (~~misleading conclusions~~)

**Mathematics** Are mathematicians concerned with reproducibility ?

- **GEOSTAT**: *Reproducing kernel Hilbert spaces* 😉
  - Yet complex scientific software (multiscale, turbulence, propagate uncertainty, quantify noise)

# Inria *Learning* team's reproducibility concerns (1/2)

**Statistics** (Misuse, *external* reproducibility issues)

- **CELESTE**: bound false positive error rate while maximizing power, robustness
- **MODAL**: ML performance assessment (~~misleading conclusions~~)

**Mathematics** Are mathematicians concerned with reproducibility ?

- **GEOSTAT**: *Reproducing kernel Hilbert spaces* 😉
  - Yet complex scientific software (multiscale, turbulence, propagate uncertainty, quantify noise)
- **SCOOL**: *Software development is not in our genes in Scool and we did not see how we might contribute in a useful manner until recently*
  - *rlberry*: *a Reinforcement Learning library for Research and Education Reaction of PhD students to have an open-source library with state-of-the-art implementations of RL algorithms*

# Inria *Learning* team's reproducibility concerns (1/2)

**Statistics** (Misuse, *external* reproducibility issues)

- **CELESTE**: bound false positive error rate while maximizing power, robustness
- **MODAL**: ML performance assessment (~~misleading conclusions~~)

**Mathematics** Are mathematicians concerned with reproducibility ?

- **GEOSTAT**: *Reproducing kernel Hilbert spaces* 😉
  - Yet complex scientific software (multiscale, turbulence, propagate uncertainty, quantify noise)
- **SCOOL**: *Software development is not in our genes in Scool and we did not see how we might contribute in a useful manner until recently*
  - *rlberry*: *a Reinforcement Learning library for Research and Education Reaction of PhD students to have an open-source library with state-of-the-art implementations of RL algorithms*
- **SIERRA**: *potentially long, non-intuitive, and technical, proofs*
  - $\rightsquigarrow$ *computer-aided proofs, techniques gathered in a software package,* PEPit*, for helping producing and reproducing convergence proofs*

# Inria *Learning* team's reproducibility concerns (2/2)

## Machine Learning/Computer Science/Software

- **INOCS**, **REALOPT**: RR not mentioned but some software made available, lots of industrial collaborations, some members quite involved in open science

# Inria *Learning* team's reproducibility concerns (2/2)

## Machine Learning/Computer Science/Software

- **INOCS**, **REALOPT**: RR not mentioned but some software made available, lots of industrial collaborations, some members quite involved in open science
- **BONUS**, **STATIFY**: making software available, encouraging their community through conferences

# Inria *Learning* team's reproducibility concerns (2/2)

## Machine Learning/Computer Science/Software

- INOCS, REALOPT: RR not mentioned but some software made available, lots of industrial collaborations, some members quite involved in open science
- BONUS, STATIFY: making software available, encouraging their community through conferences
- MAASAI: making software and algorithms of others available
  - POT (optimal transport): *implementations of more than 40 research papers providing new solvers*

# Inria *Learning* team's reproducibility concerns (2/2)

## Machine Learning/Computer Science/Software

- INOCS, REALOPT: RR not mentioned but some software made available, lots of industrial collaborations, some members quite involved in open science
- BONUS, STATIFY: making software available, encouraging their community through conferences
- MAASAI: making software and algorithms of others available
  - POT (optimal transport): *implementations of more than 40 research papers providing new solvers*
- RANDOPT, TAU, OCKHAM: benchmarking frameworks and organizing challenges
  - Target = Fair comparison and Animation of the community
  - COCO, BenchOpt, Codalab

# Different Reproducibility Concerns in Modern Science

**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

# Different Reproducibility Concerns in Modern Science

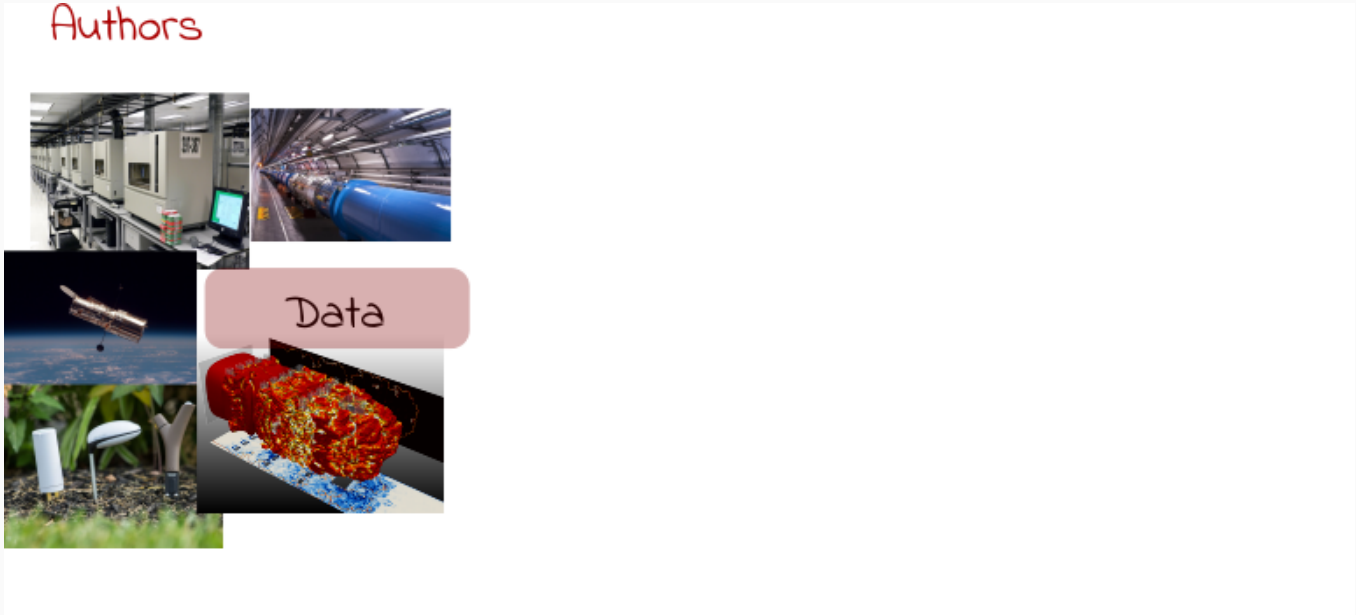
**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

***Artificial Intelligence*** most of the above 😉

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



# Different Reproducibility Concerns in Modern Science

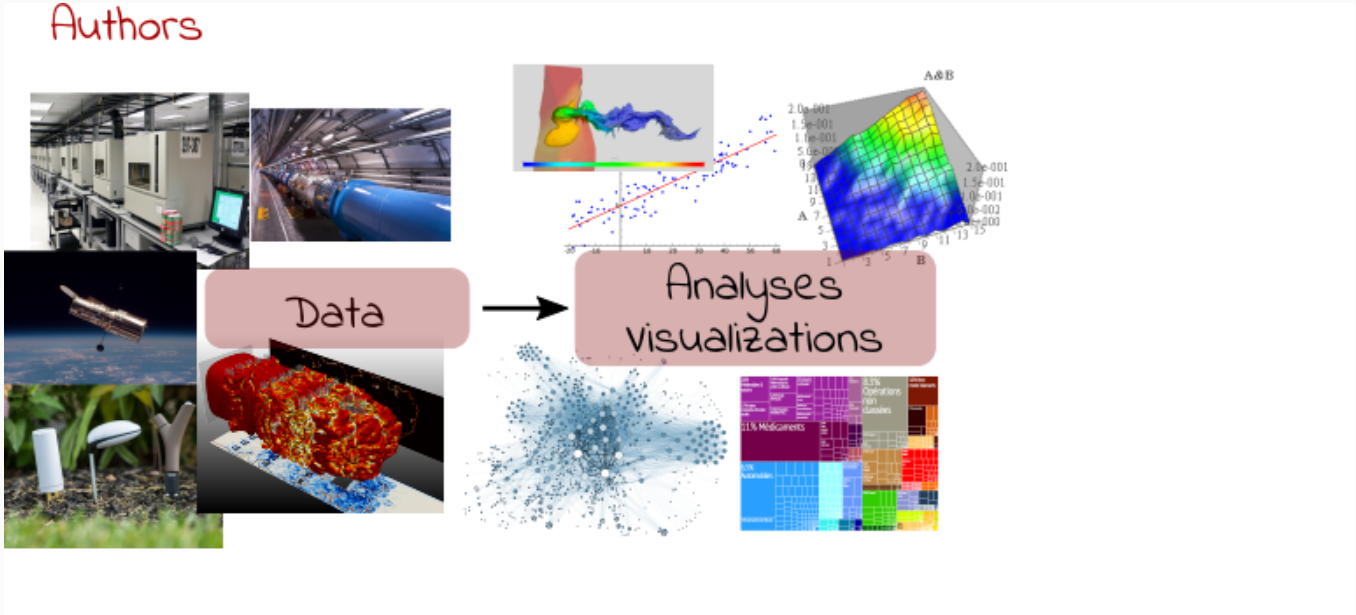
**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

***Artificial Intelligence*** most of the above 😉

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



# Different Reproducibility Concerns in Modern Science

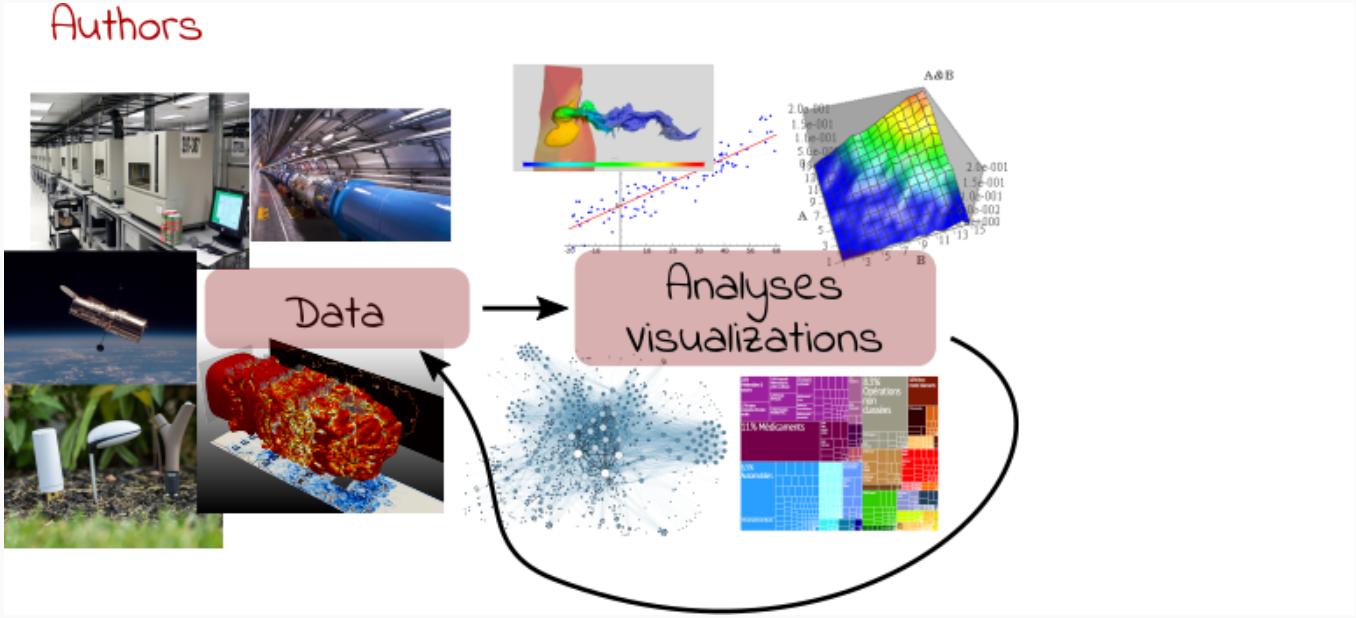
**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

***Artificial Intelligence*** most of the above 😉

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



# Different Reproducibility Concerns in Modern Science

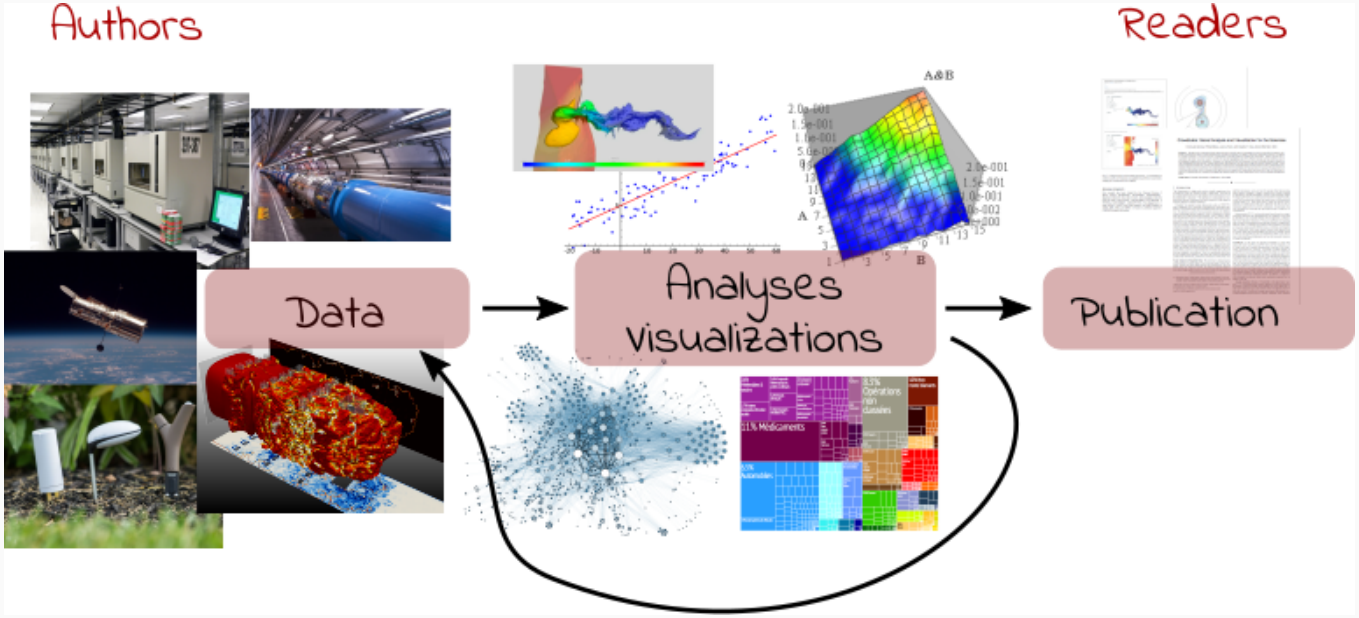
**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

***Artificial Intelligence*** most of the above 😉

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



# Different Reproducibility Concerns in Modern Science

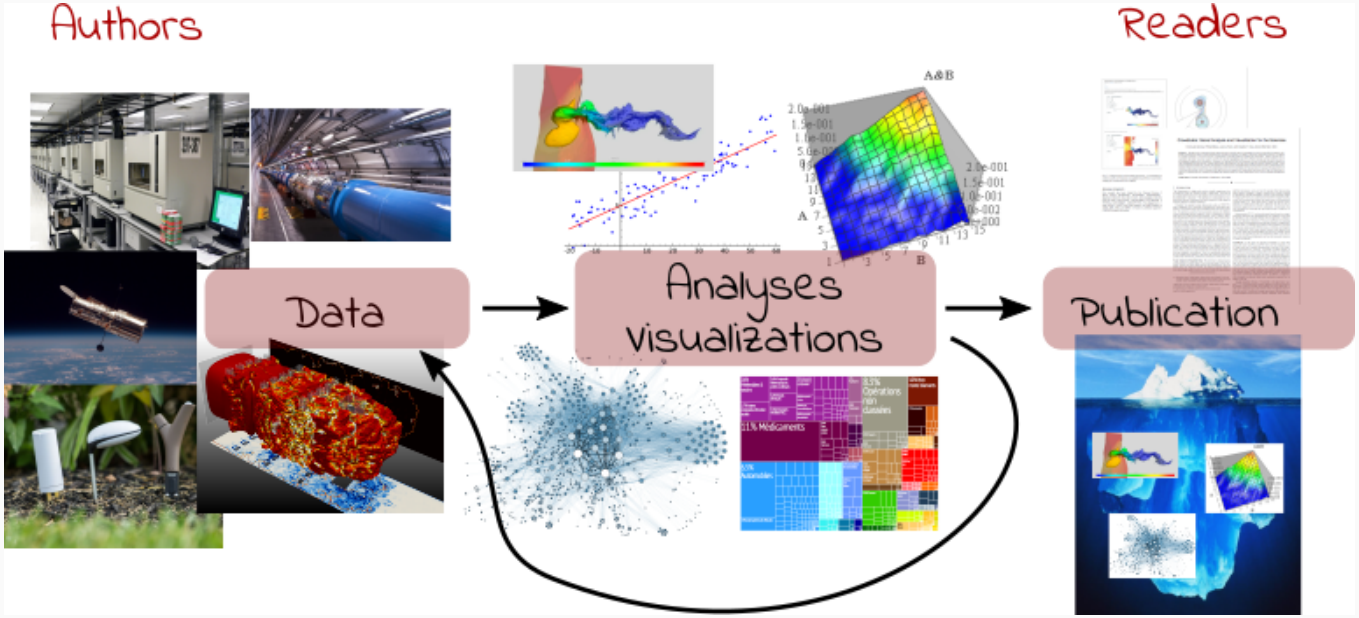
**Social Sciences, Oncology, ...** methodology, statistics, pre-registration

**Genomics** software engineering, computational reproducibility, provenance

**Computational fluid dynamics** numerical issues

***Artificial Intelligence*** most of the above 😉

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




Reproducible Research = Bridging the Gap by working Transparently

# Reproducibility (glossary may vary)

Many (conflicting) definitions (*replicability*, *repeatability*, *reproducibility*, *corroboration*)

| | (*new data,* *same person, independant researcher*) | | |
|---|---|---|---|
| **experimental** reproducibility | similar input (data) + similar experimental protocol | → | **similar results**[1] |
| **statistical** reproducibility | different input (data) + same analysis | → | **same conclusions**[2] |
| **computational** reproducibility | same input (data) + same code/software + same software environment | → | **exact same results**[3] |

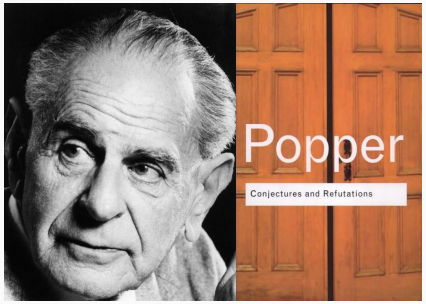
**Reproducible Research = A way of doing science so that scientific experiments, discoveries, results, etc. can be easily reproduced (done again), to be confirmed, or to be built on for the next study.**

– Courtesy G. Durrif, 2021

[1] Up-to measurement variability and precision
[2] Independently from (random) sampling variability (fight bias)
[3] Bitwise

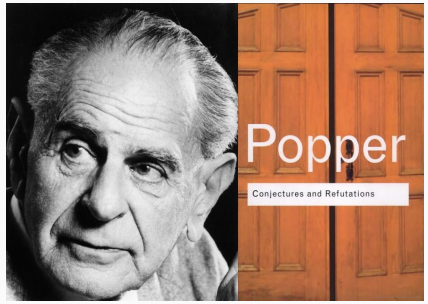
# Reproducibility of experimental results: the *hallmark of science*

1934: **Karl Popper** puts the notions of falsifiability and crucial experiment as the hallmark of science




- If no experiment can be set up to disprove your theory, it is not science
- Good experiments discriminate good theories from bad ones
- Non-reproducible single occurrences are of no significance to science

# Reproducibility of experimental results: the *hallmark of science*




1934: **Karl Popper** puts the notions of falsifiability and crucial experiment as the hallmark of science

- If no experiment can be set up to disprove your theory, it is not science
- Good experiments discriminate good theories from bad ones
- Non-reproducible single occurrences are of no significance to science

## An ideal rather than the norm

Popper's proposal works well for Physics from the 18th century but is not so simple for many other domains:

- Theory of evolution
- Spotting a SuperNova
- Particle Physics (a single LHC)
- Biology (every animal does not behave in the same way)
- Anthropology (impact on people from a remote culture)

# Reproducibility: a *core value of science*

1. Universality: Science aims for objective findings, accessible to anyone

   Reproducibility acts as a Universality/Robustness control

2. Incremental: We build on each others work but everybody makes mistakes

   Methods, biases, ... How to discriminate sound ~~theories~~ experiments from bad ones? 😉

   Reproducibility acts as a Quality control

# Reproducibility: a *core value of science*

1. Universality: Science aims for objective findings, accessible to anyone

   Reproducibility acts as a Universality/Robustness control

2. Incremental: We build on each others work but everybody makes mistakes

   Methods, biases, ... How to discriminate sound ~~theories~~ experiments from bad ones? 😉

   Reproducibility acts as a Quality control

   *Today the computer is just as important a tool for chemists as the test tube. Simulations are so realistic that they predict the outcome of traditional experiments...*

   *– Nobel Comity (Chemistry), 2013*

But, scientific practices have greatly evolved, in particular since we rely on computers, statistics, machine learning, complex black boxes, ...

# Reproducibility: both an internal and external problem for CS/ML/...

# Computers...

How computers broke science –

and what we can do about it

> *Most modern science is so complicated, and most journal articles so brief, it's impossible for the article to include details of many important methods and decisions made by the researcher as he analyzed his data on his computer. How, then, can another researcher judge the reliability of the results, or reproduce the analysis?*

– Ben Marwick,
The conversation, 2015

Point-and-click procedures are rampant but they hinder reproducibility.

Spreadsheets are generalized and intensively uses in biology:

- `Membrane-Associated Ring Finger (C3HC4) 1`,
  `E3 Ubiquitin Protein Ligase` → `MARCH1` → 2016-03-01 → 1456786800
- `2310009E13` → 2.31E+19

And more recently, we had the COVID tracing failure.

# ... and Statistics/Machine Learning

Machine Learning: Trouble at the lab, The Economist 2013

> *According to some estimates, three-quarters of published scientific papers in the field of machine learning are bunk because of this "overfitting".*
>
> *– Alex "Sandy" Pentland*

The Reproducibility Crisis in ML-based science (Princeton workshop 2022)

> *Reproducibility failures in ML-based science are systemic. We found 20 reviews across 17 scientific fields (medicine, neuroimaging, autism diagnosis, genomics, computer security, ...) that find errors in a total of 329 papers that use ML-based science and in some cases leading to wildly overoptimistic conclusion. [...] complex ML models don't perform substantively better than decades-old LR models.*
>
> *Data leakage: spurious relationship between the independent variables and the target variable that arises as an artifact of the data collection, sampling, or pre-processing strategy.*
>
> *– S. Kapoor and A. Narayanan*

# This is about Computational Science. Should mathematicians care?

Computer Science is young and inherits from Mathematics, Engineering, Linguistic, Nat. Sciences, ...

Purely theoretical scientists whose practice is close to mathematics *may* not be concerned (can't publish a math article without releasing the proofs).

# This is about Computational Science. Should mathematicians care?

Computer Science is young and inherits from Mathematics, Engineering, Linguistic, Nat. Sciences, ...

Purely theoretical scientists whose practice is close to mathematics *may* not be concerned (can't publish a math article without releasing the proofs).

Yet, incoherencies are common, especially in a fast moving field:

- E.g., definitions/concepts in book/article A and B are *slightly different* and the resulting theorems cannot be mixed
- Have a look at Vladimir Voevodsky's talk in 2014 at Princeton 😉
  - $\rightsquigarrow$ computer-aided proofs in **SIERRA**
- ERC Nano bubbles: how, when and why does science fail to correct itself?

  Flagging incorrect nucleotide sequence reagents in biomedical papers: To what extent does the leading publication format impede automatic error detection? (Labbe et al., 2020)

# Emerging Tools and Practices

# Main Challenges

```
my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

## Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

# Main Challenges

```
my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

## Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

## Tracking software environment

- `my_code` depends on a dozen of libraries, which depend on dozens of libraries
- `my_code` was compiled with `clang 1:9.0-49.1` and `-O3 -funroll-loops -fno-strict-aliasing -finline-functions ...`

# Main Challenges

```
my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

## Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

## Tracking software environment

- `my_code` depends on a dozen of libraries, which depend on dozens of libraries
- `my_code` was compiled with `clang 1:9.0-49.1` and `-O3 -funroll-loops -fno-strict-aliasing -finline-functions ...`

## Tracking parameters and data

- `*.dat`? Ooh, you ran this in `data/2091293-AJXQ37`?
- Wasn't `mymap.dat` updated since then?
- That was for `foo.csv`. What about `bar.csv`? Is it reproducible?

# Main Challenges

```
my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

## Tracking code version

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

## Tracking software environment

- `my_code` depends on a dozen of libraries, which depend on dozens of libraries
- `my_code` was compiled with `clang 1:9.0-49.1` and `-O3 -funroll-loops -fno-strict-aliasing -finline-functions ...`

## Tracking parameters and data

- `*.dat`? Ooh, you ran this in `data/2091293-AJXQ37`?
- Wasn't `mymap.dat` updated since then?
- That was for `foo.csv`. What about `bar.csv`? Is it reproducible?

## Tracking the process (on short/long term)

- Why did I run this? What did I learn from it? I remember doing this but when?

# Main Challenges

```
my_code --cfg=magical_param:0.94572 '*.dat' --output foo.csv
```

**Tracking code version**

- `my_code` is revision `21b95ecfa0911d6ca87668482b11ab9498edd8f3`

**Tracking software environment**

- `my_code` depends on a dozen of libraries, which depend on dozens of libraries
- `my_code` was compiled with `clang 1:9.0-49.1` and `-O3 -funroll-loops -fno-strict-aliasing -finline-functions ...`

**Tracking parameters and data**

- `*.dat`? Ooh, you ran this in `data/2091293-AJXQ37`?
- Wasn't `mymap.dat` updated since then?
- That was for `foo.csv`. What about `bar.csv`? Is it reproducible?

**Tracking the process** (on short/long term)

- Why did I run this? What did I learn from it? I remember doing this but when?

**Handle complex sequences and reuse results** (leverage supercomputers)

# Tool 1: Computational Notebooks/Litterate Programming

## Un document computationnel

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

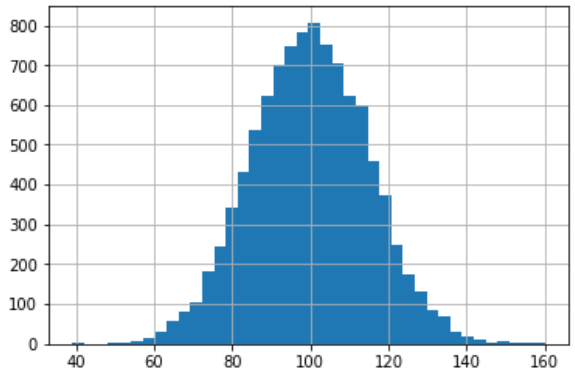
```
3.141592653589793
```

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 $\pi$ (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... ☺).
```

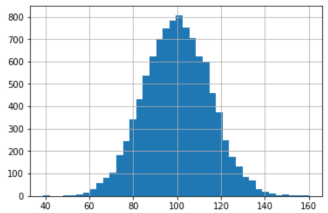
In [3]:

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 $\pi$ vaut *approximativement*

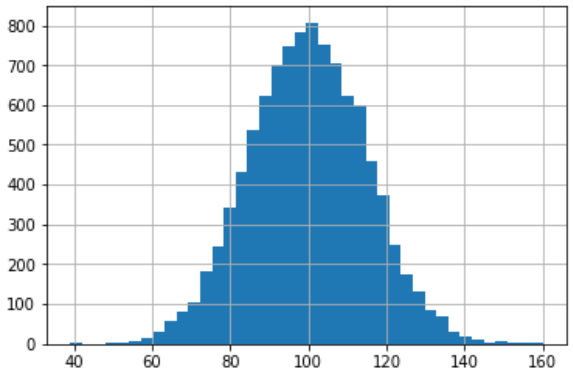
```
3.141592653589793
```

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
```

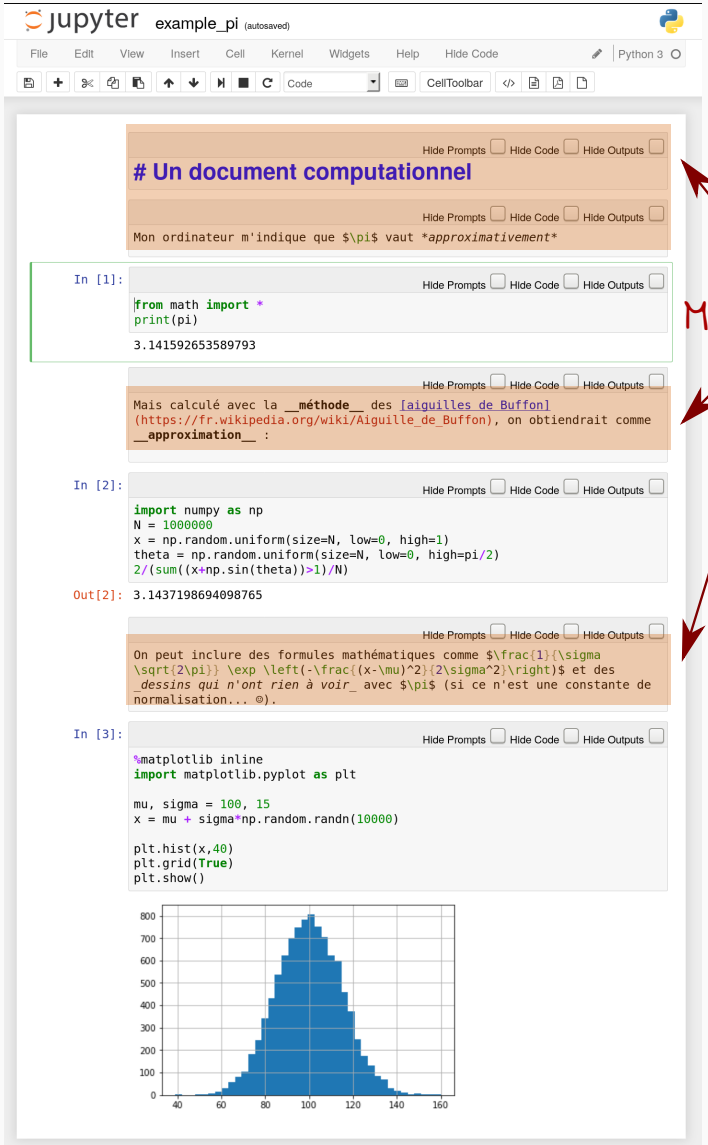
On peut inclure des formules mathématiques comme $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des *dessins qui n'ont rien à voir* avec $\pi$ (si ce n'est une constante de normalisation... ☺).



# Tool 1: Computational Notebooks/Litterate Programming

Document initial dans son environnement

Document final




Mark Down

## Un document computationnel

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

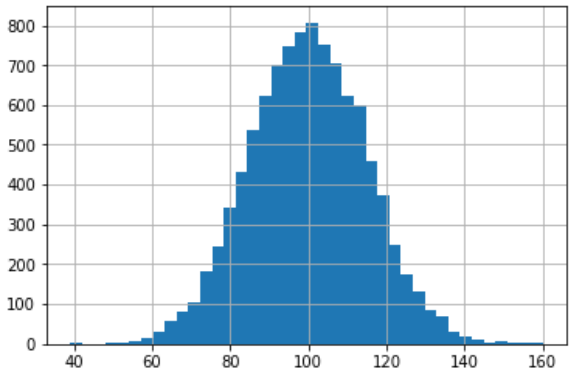
```
3.141592653589793
```

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 $\pi$ (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... ☺).
```

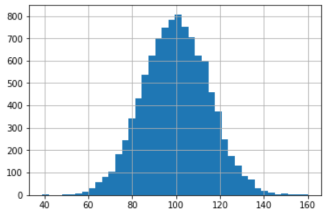
In [3]:

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()
```




Code

Document final

## Un document computationnel

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

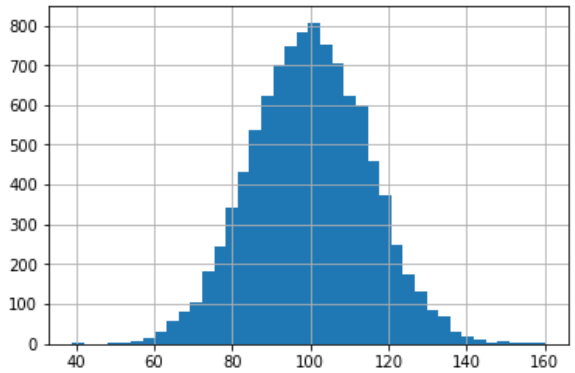
```
3.141592653589793
```

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 $\pi$ (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... ☺).
```

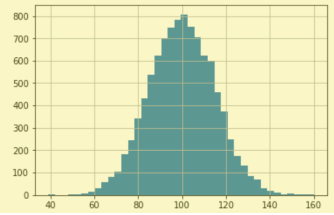
In [3]:

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 $\pi$ vaut *approximativement*

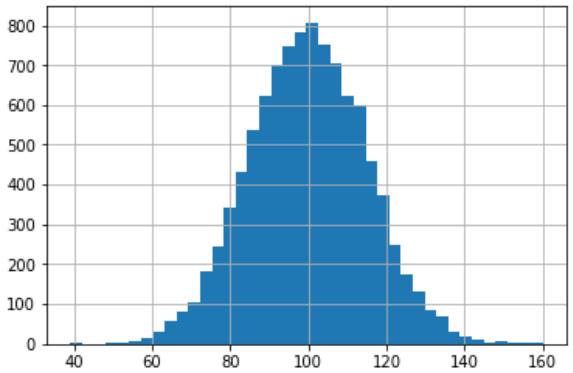
```
3.141592653589793
```

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 $\pi$ (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... ☺).
```

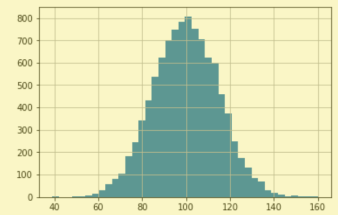
In [3]:

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 $\pi$ vaut *approximativement*

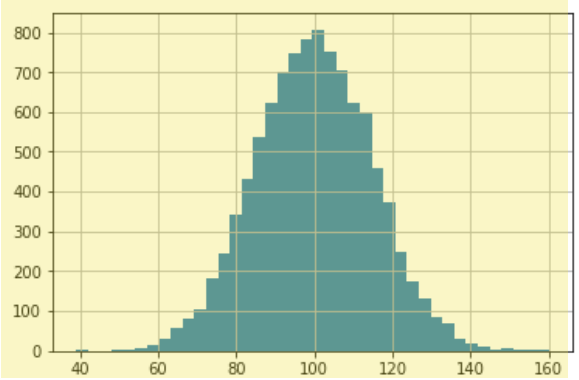
```
3.141592653589793
```

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 $\pi$ (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... ☺).
```

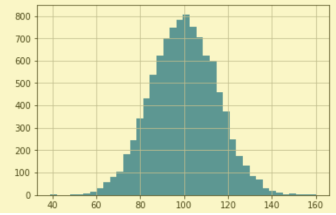
In [3]:

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 $\pi$ vaut *approximativement*

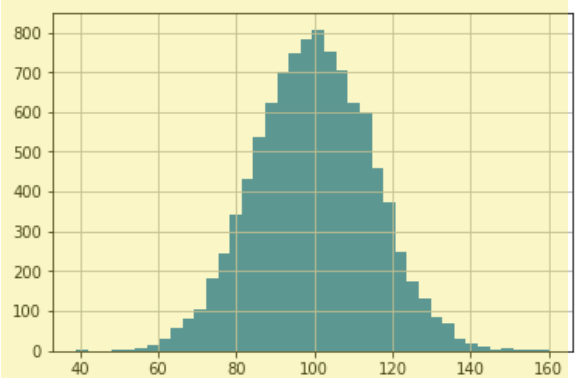
```
3.141592653589793
```

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



# Tracking the Exploration Process with a Notebook

The *REPL* (Read–eval–print loop) vs. *Notebook* vs *IDE* debate

- In the beginning was the Mathematica (1988) and the Maple (1989) notebooks, which allow to tell a story (litterate programming)

# Tracking the Exploration Process with a Notebook

The *REPL* (Read–eval–print loop) vs. *Notebook* vs *IDE* debate

- In the beginning was the Mathematica (1988) and the Maple (1989) notebooks, which allow to tell a story (litterate programming)
- Then IJulia, IPython, and IR merged into the Jupyter notebook 😎
  - The coolest kid on the block without ~~IDE/Structure~~, ~~Interactive collaboration/Versioning~~, ~~Software Environment Control~~, ~~Easy setup and use of Computing resources~~ !!! ☹

# Tracking the Exploration Process with a Notebook

The *REPL* (Read–eval–print loop) vs. *Notebook* vs *IDE* debate

- In the beginning was the Mathematica (1988) and the Maple (1989) notebooks, which allow to tell a story (litterate programming)
- Then IJulia, IPython, and IR merged into the Jupyter notebook 😎
  - The coolest kid on the block without ~~IDE/Structure~~, ~~Interactive collaboration/Versioning~~, ~~Software Environment Control~~, ~~Easy setup and use of Computing resources~~ !!! ☹️

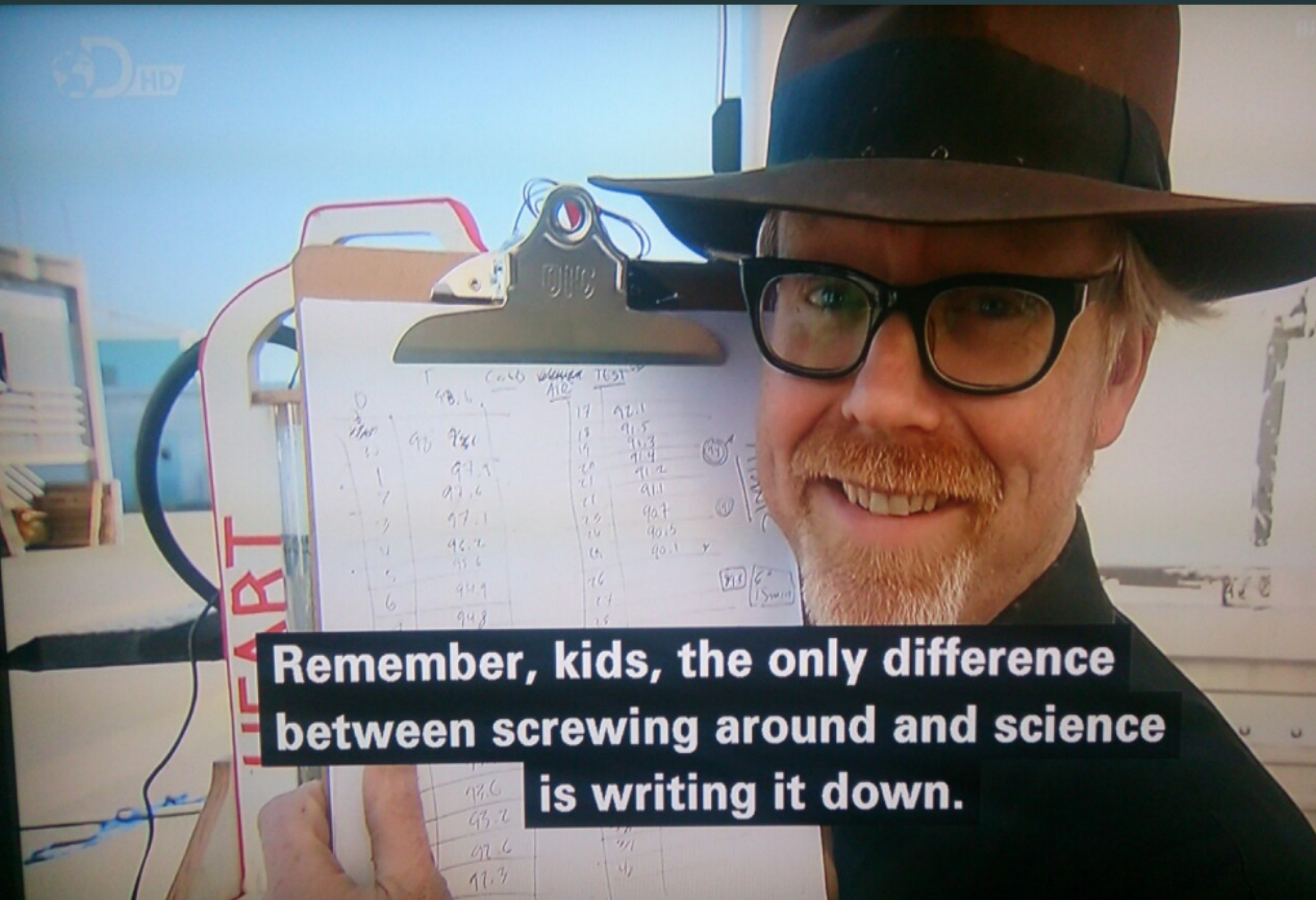
Now we have:

- JupyterLab, Binder, JupyterHub
  - Guix-Jupyter
- IDE: Rstudio (not just R), Emacs, VSstudio (Jupyter∼backend)
- CodeOcean *showroom, ~~interactive~~*
- CoCalc/SAGE notebooks, Kaggle, Google Colab, DeepNote

  *real-time, versioning, custom environment*
- fast.ai/nbdev

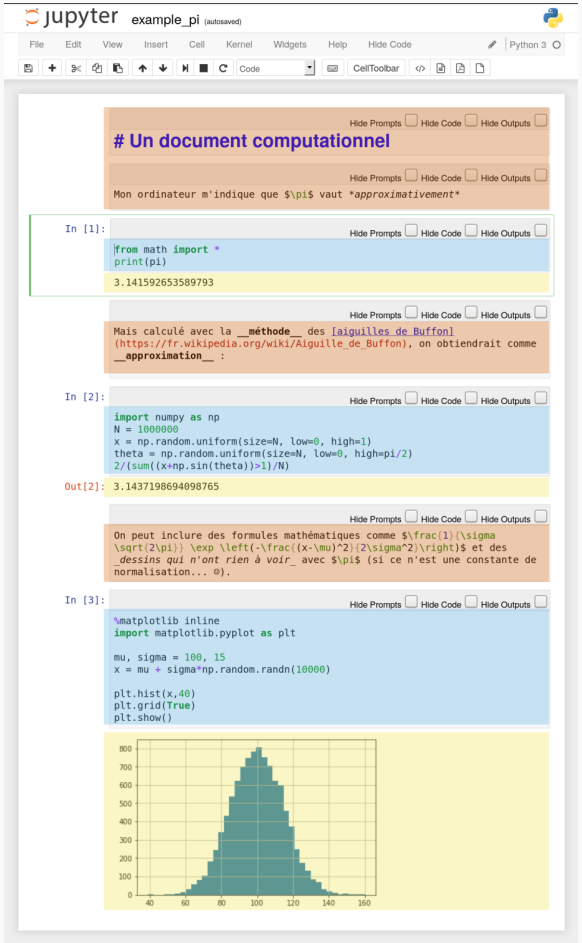
  *merge conflict, module export, test*
- Beaker, Count 😐 ???

# Tool 1 bis: Laboratory Notebooks, Computational Documents



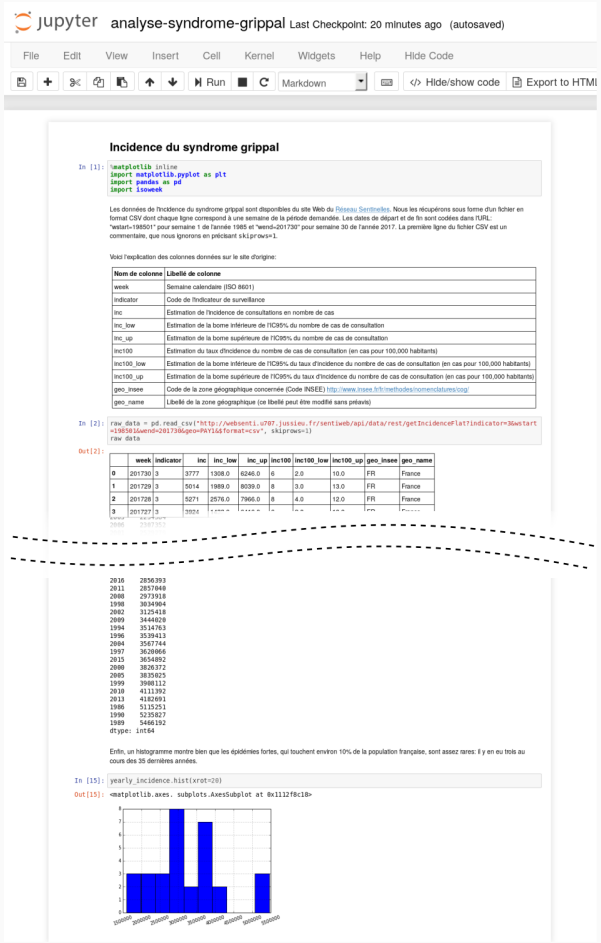
# Tool 1 ter: Workflows

Notebooks are no panacea and do not help developing clean code



# Tool 1 ter: Workflows

Notebooks are no panacea and do not help developing clean code



jupyter analyse-syndrome-grippal Last Checkpoint: 20 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Hide Code

Run Markdown Hide/show code Export to HTML

**Incidence du syndrome grippal**

```
In [1]: %matplotlib inline
        import matplotlib.pyplot as plt
        import pandas as pd
        import isoweek
```

Les données de l'incidence du syndrome grippal sont disponibles du site Web du Réseau Sentinelles. Nous les récupérons sous forme d'un fichier en format CSV dont chaque ligne correspond à une semaine de la période demandée. Les dates de départ et de fin sont codées dans l'URL : "sstart=198501" pour semaine 1 de l'année 1985 et "send=201730" pour semaine 30 de l'année 2017. La première ligne du fichier CSV est un commentaire, que nous ignorons en précisant `skiprows=1`.

Voici l'explication des colonnes données sur le site d'origine:

| Nom de colonne | Libellé de colonne |
|---|---|
| week | Semaine calendaire (ISO 8601) |
| indicator | Code de l'indicateur de surveillance |
| inc | Estimation de l'incidence de consultations en nombre de cas |
| inc_low | Estimation de la borne inférieure de l'IC95% du nombre de cas de consultation |
| inc_up | Estimation de la borne supérieure de l'IC95% du nombre de cas de consultation |
| inc100 | Estimation du taux d'incidence du nombre de cas de consultation (en cas pour 100,000 habitants) |
| inc100_low | Estimation de la borne inférieure de l'IC95% du taux d'incidence du nombre de cas de consultation (en cas pour 100,000 habitants) |
| inc100_up | Estimation de la borne supérieure de l'IC95% du taux d'incidence du nombre de cas de consultation (en cas pour 100,000 habitants) |
| geo_insee | Code de la zone géographique concernée (Code INSEE) http://www.insee.fr/fr/methodes/nomenclatures/cog/ |
| geo_name | Libellé de la zone géographique (ce libellé peut être modifié sans préavis) |

```
In [2]: raw_data = pd.read_csv("http://websenti.u707.jussieu.fr/sentiweb/api/data/rest/getIncidenceFlat?indicator=3&sstart
        =198501&send=201730&geo=PAYS1&format=csv", skiprows=1)
        raw_data
```

Out[2]:

| | week | indicator | inc | inc_low | inc_up | inc100 | inc100_low | inc100_up | geo_insee | geo_name |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 201730 | 3 | 3777 | 1308.0 | 6246.0 | 6 | 2.0 | 10.0 | FR | France |
| 1 | 201729 | 3 | 5014 | 1989.0 | 8039.0 | 8 | 3.0 | 13.0 | FR | France |
| 2 | 201728 | 3 | 5271 | 2576.0 | 7966.0 | 8 | 4.0 | 12.0 | FR | France |
| 3 | 201727 | 3 | 3924 | | | | | | | |

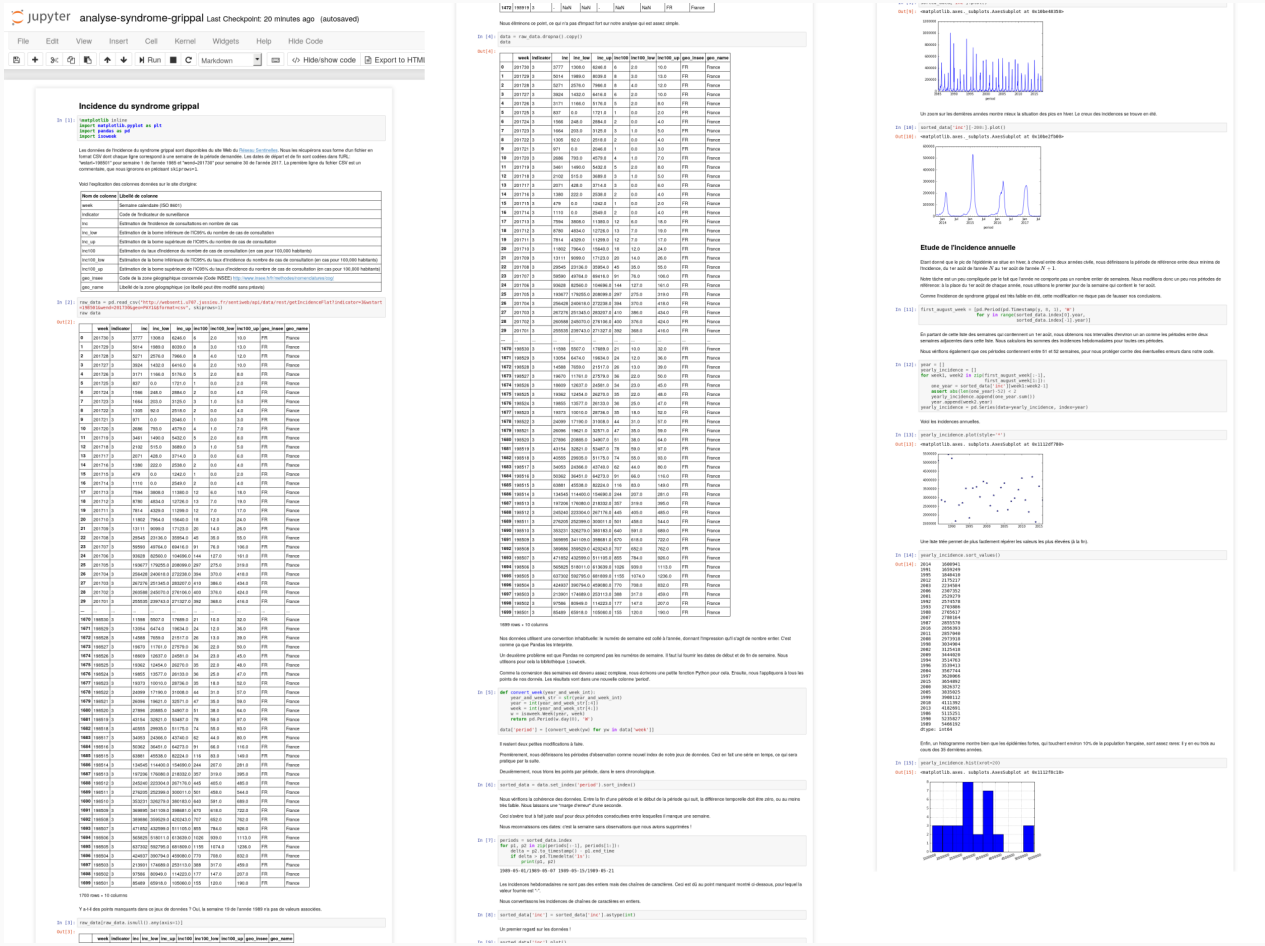
```
2016    2856393
2011    2857040
2008    2973918
1998    3034904
2002    3125418
2009    3444028
1994    3514763
1996    3539413
2004    3567744
1997    3620066
2015    3654892
2000    3826372
2005    3835025
1999    3908112
2010    4111392
2013    4182691
1986    5115251
1990    5235827
1989    5466192
dtype: int64
```

Enfin, un histogramme montre bien que les épidémies fortes, qui touchent environ 10% de la population française, sont assez rares: il y en eu trois au cours des 35 dernières années.

```
In [15]: yearly_incidence.hist(xrot=20)
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1112f8c18>
```

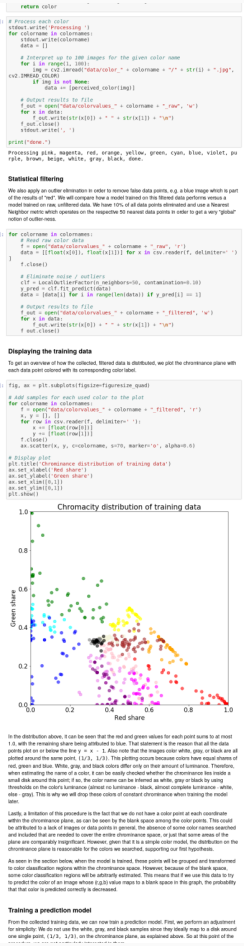
# Tool 1 ter: Workflows

Notebooks are no panacea and do not help developing clean code



# Tool 1 ter: Workflows

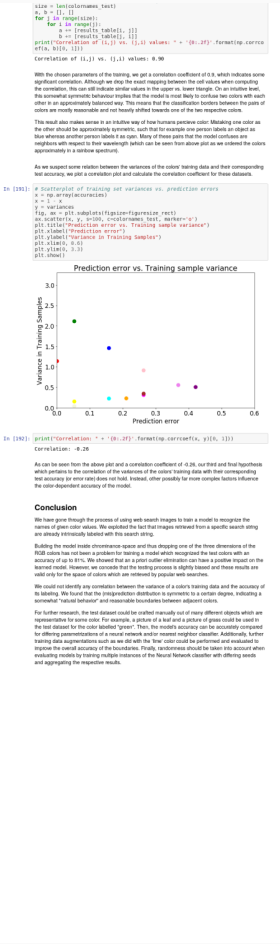
## Notebooks are no panacea and do not help developing clean code

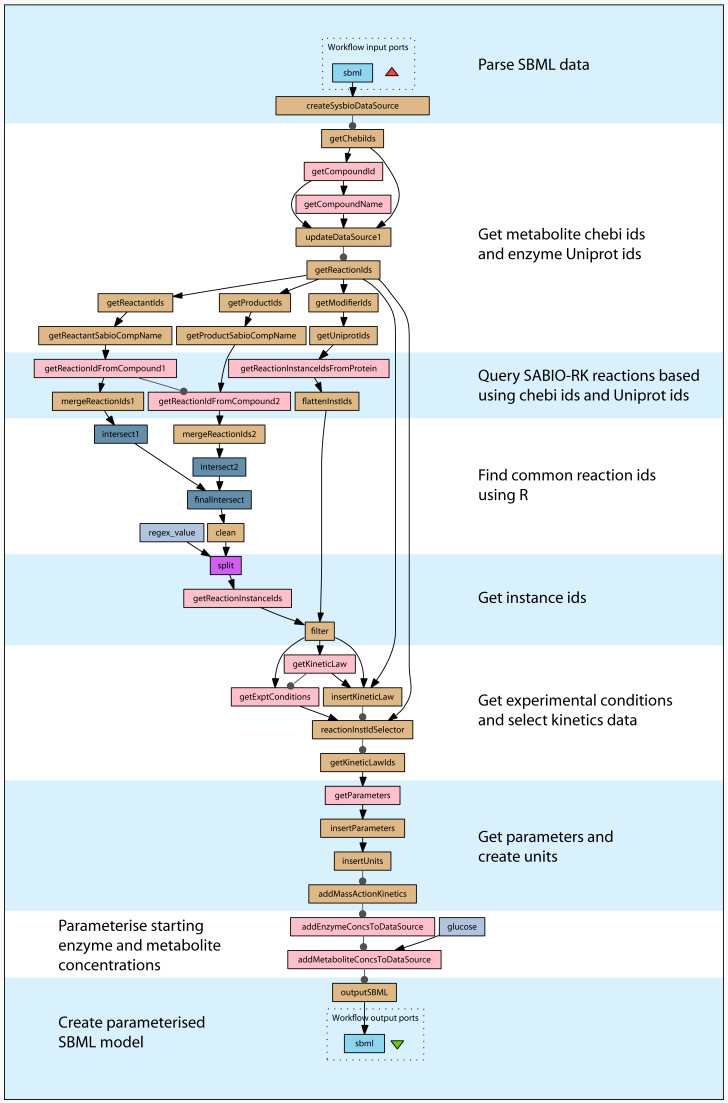


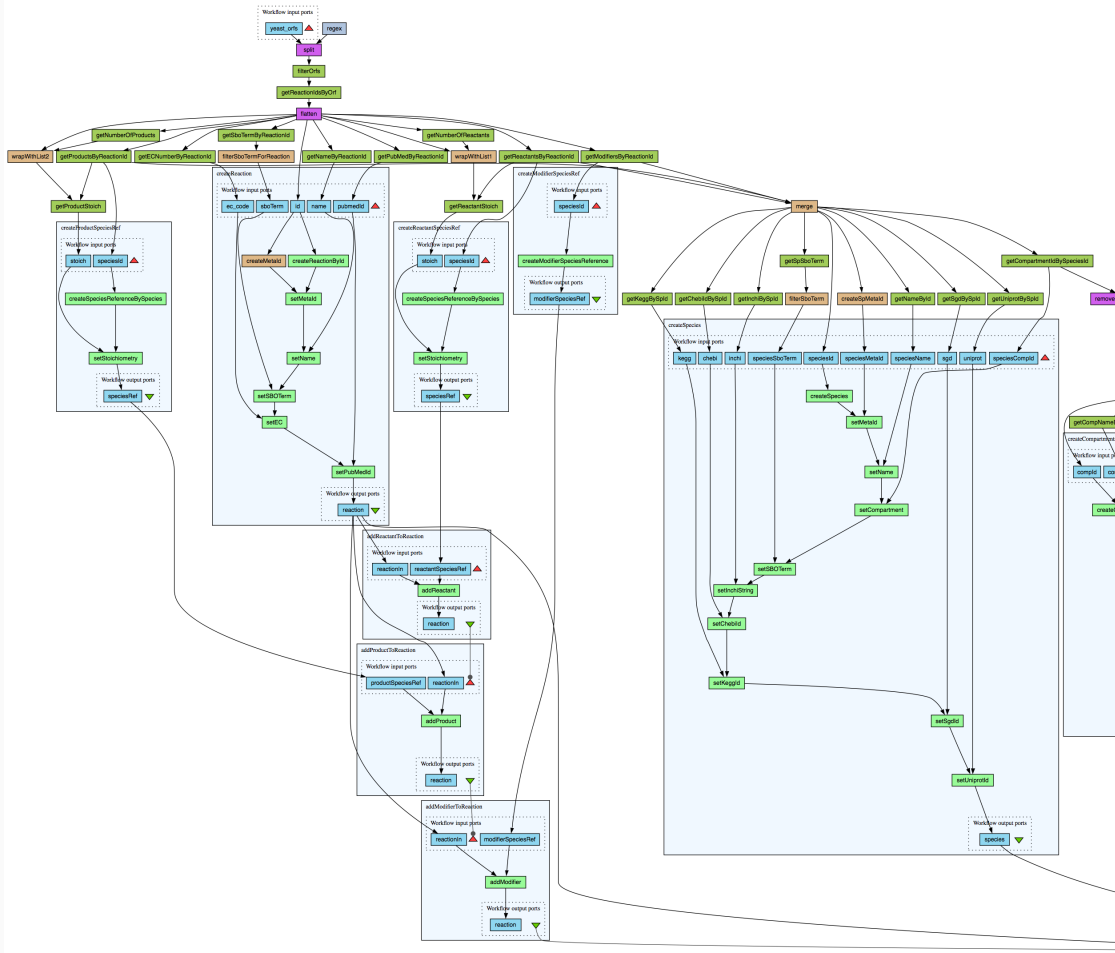




# Tool 1 ter: Workflows



# Tool 1 ter: Workflows

Workflows:

- Clearer high-level view
- Explicit composition of codes and data movement
- Safer sharing, reusing, and execution
- Notebooks are a variant that is both impoverished and richer
  - No simple/mature path from a notebook to a workflow
    - SoS Polyglot Notebook/Workflow System

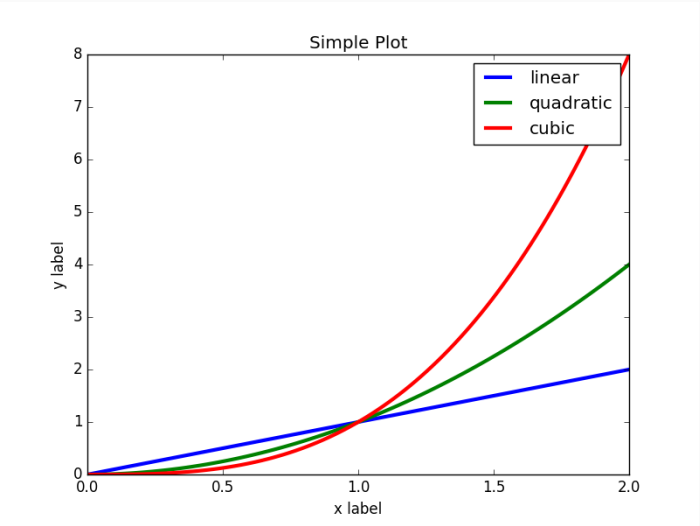
Examples:

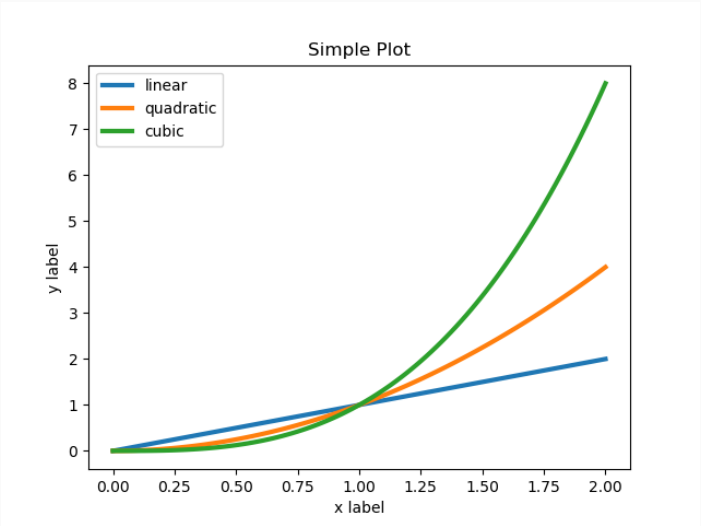
- Galaxy, Kepler, Taverna, Pegasus, Collective Knowledge, VisTrails
- Light-weight: `make`, dask, drake, swift, `snakemake`, ...

# Software dependencies: horror stories

# Software dependencies: horror stories

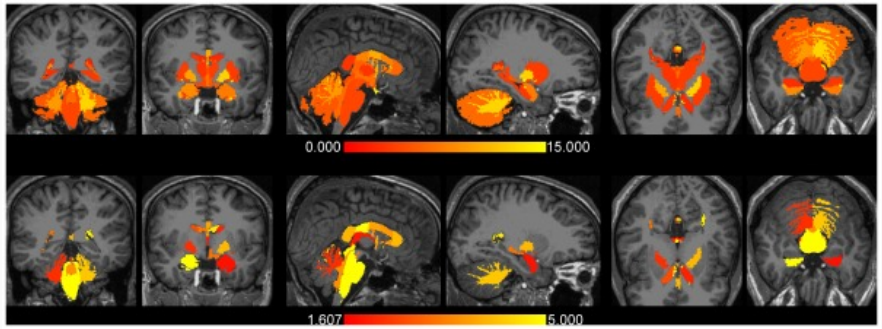
- Software environment evolution



# Software dependencies: horror stories

- Software environment evolution
- OS heterogeneity

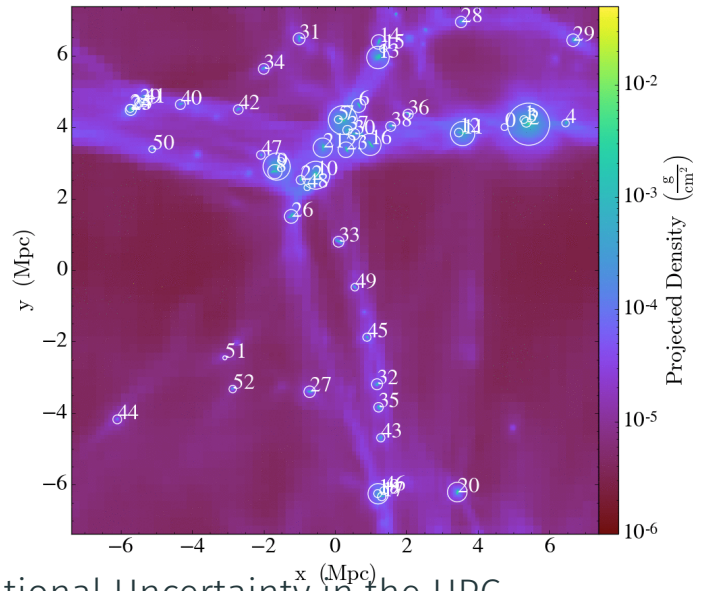



The Effects of FreeSurfer Version, Workstation Type, and Macintosh Operating System Version on Anatomical Volume and Cortical Thickness Measurements (PLOS ONE, 2012)

> *Significant differences in volume and cortical thickness were revealed across FreeSurfer versions:*
>
> - *volume:* *8.8 ± 6.6% (range 1.3-64.0%)*
> - *cortical thickness:* *2.8 ± 1.3% (range 1.1-7.7%)*
>
> *About a factor two smaller differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.*
>
> *In the context of an ongoing study, users are discouraged to update to a new major release of either FreeSurfer or operating system.*
>
> *Formal assessment of the accuracy of FreeSurfer is desirable.*

# Software dependencies: horror stories

- Software environment evolution
- OS heterogeneity
- Impact of the compiler

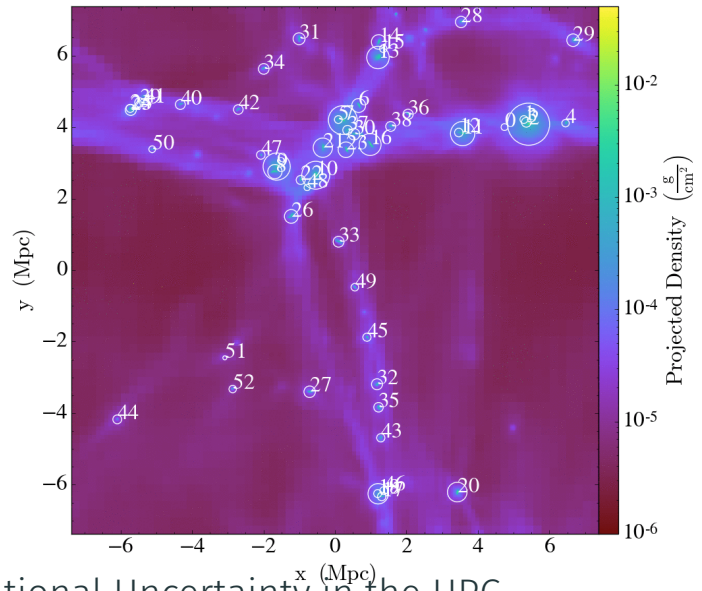



Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler | Optim. | Largest Halo Avg Mass. | Std. Err | Walltime |
|---|---|---|---|---|
| gcc@6.2.0 | None | 2.273E 46 | 1.069E 44 | 22h |

# Software dependencies: horror stories

- Software environment evolution
- OS heterogeneity
- Impact of the compiler

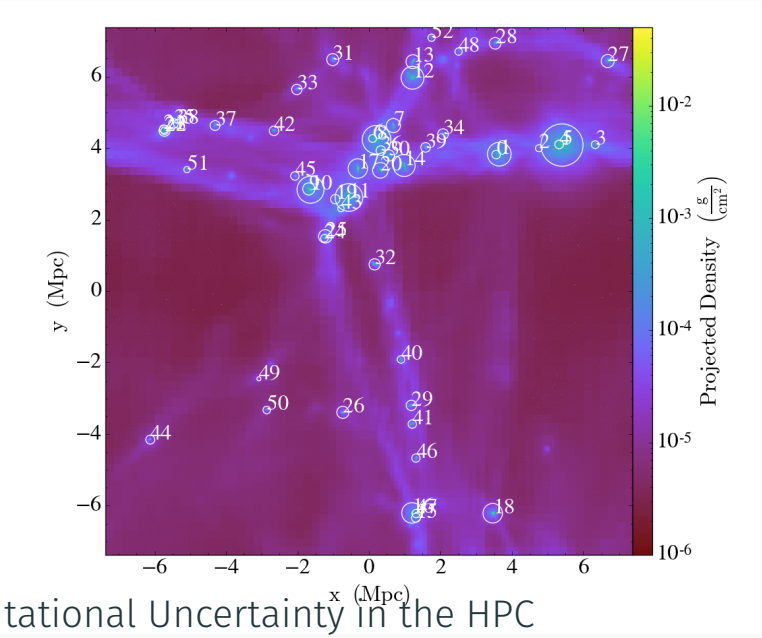



Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler | Optim. | Largest Halo Avg Mass. | Std. Err | Walltime |
|---|---|---|---|---|
| gcc@6.2.0 | None | 2.273E 46 | 1.069E 44 | 22h |

# Software dependencies: horror stories

- Software environment evolution
- OS heterogeneity
- Impact of the compiler

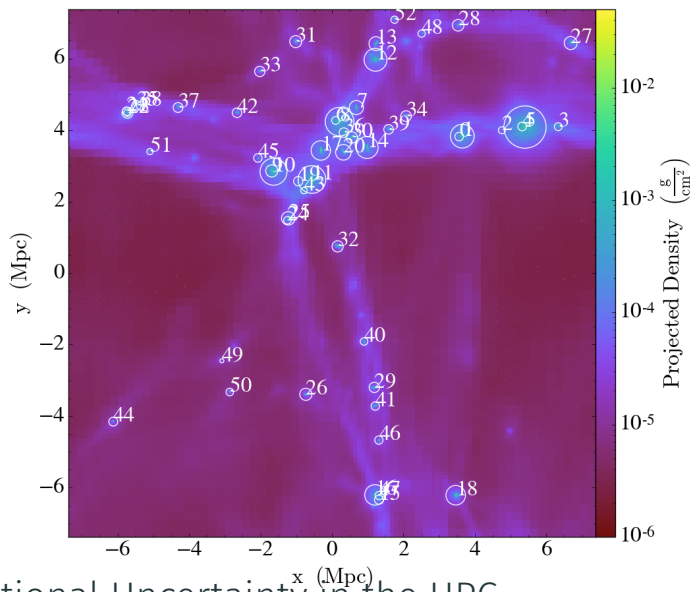



Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler | Optim. | Largest Halo Avg Mass. | Std. Err | Walltime |
|---|---|---|---|---|
| gcc@6.2.0 | None | 2.273E 46 | 1.069E 44 | 22h |
| gcc@6.2.0 | Normal | 2.266E 46 | 1.218E 44 | 10h |
| gcc@6.2.0 | High | 2.275E 46 | 1.199E 44 | 9h |

# Software dependencies: horror stories

- Software environment evolution
- OS heterogeneity
- Impact of the compiler



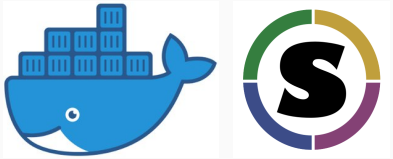

Assessing Reproducibility: An Astrophysical Example of Computational Uncertainty in the HPC Context (ResCuE-HPC, 2018)

| Compiler | Optim. | Largest Halo Avg Mass. | Std. Err | Walltime |
|---|---|---|---|---|
| gcc@6.2.0 | None | 2.273E 46 | 1.069E 44 | 22h |
| gcc@6.2.0 | Normal | 2.266E 46 | 1.218E 44 | 10h |
| gcc@6.2.0 | High | 2.275E 46 | 1.199E 44 | 9h |
| intel@16.0.3 | None | 22.71 E 46 | 1.587E 44 | 39h |
| intel@16.0.3 | Normal | 43.30 E 46 | 1.248E 44 | 7h |
| intel@16.0.3 | High | 2.268E 46 | 1.414E 44 | 6h |
| cce@8.5.5 | Low | 43.11 E 46 | 1.353E 44 | 16h |
| cce@8.5.5 | Normal | 2.271E 46 | 1.261E 44 | 6h |
| cce@8.5.5 | High | 2.272E 46 | 1.341E 44 | 5h |

# Tool 2: Containers and Package Managers

The good


The bad



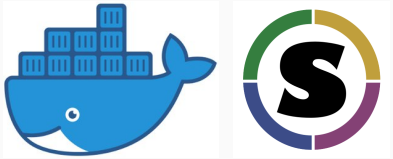
The uggly


Automatic tracking

# Tool 2: Containers and Package Managers

The good


The bad



The uggly


## Automatic tracking

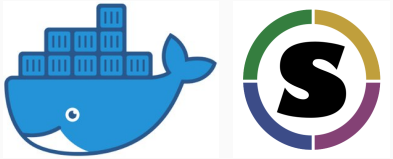
## Containers

- Pros: Lightweight, Good isolation, Easy to use
  - Running as easy as `docker run <img> <cmd>`
  - Building images: `docker build -f <Dockerfile>`
  - Sharing through the Docker Hub: `docker pull/push <img>`

# Tool 2: Containers and Package Managers

The good


The bad



The uggly


## Automatic tracking

## Containers

- Pros: Lightweight, Good isolation, Easy to use
- Cons: Opaque, Container build is generally not reproducible
  - Recipes rarely follow *reproducible good practices*

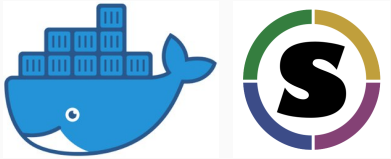
```
1  FROM ubuntu:20.04
2  RUN apt-get update
3       && apt-get upgrade -y
4       && apt-get install -y ...
```

- Choose a stable image (and the smallest possible)
- Include only the necessary libraries (e.g. no graphics libs)
- Avoid system updates (instead freeze sources)

# Tool 2: Containers and Package Managers

The good


The bad



The uggly


**Automatic tracking**

**Containers**

- Pros: Lightweight, Good isolation, Easy to use
- Cons: Opaque, Container build is generally not reproducible

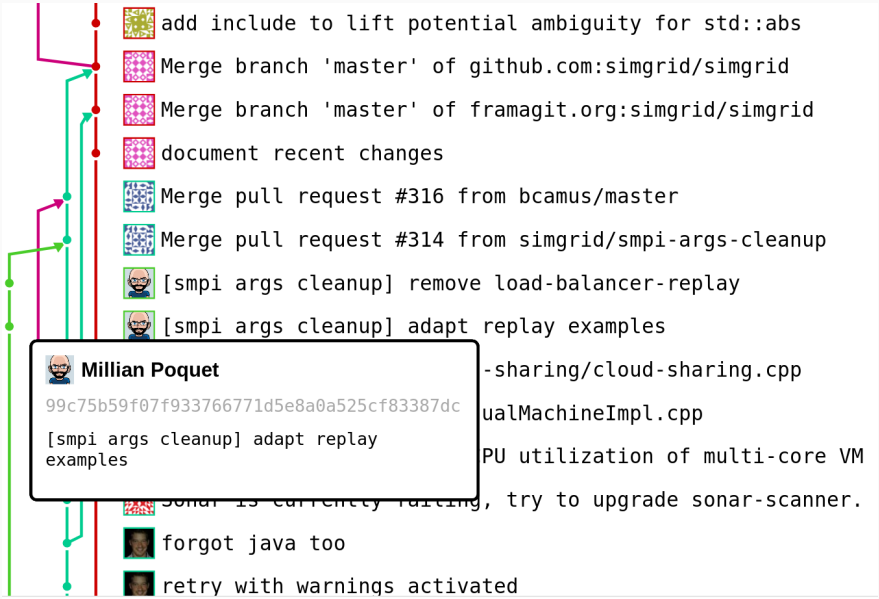
**Package managers (the uggly and the good)**

- Language specific: `pip/pipenv/virtualenv`, `conda`, `CRAN/Bioconductor`
  - Limits: version management, durability, permeable, language centric
- **GUIX/NiX = Full-fledged functional package manager**
  - Native support for environment (*à la git*)
  - Isolation through `--pure`
  - Recompile from source (cache recommended)

# Tool 3: Version Control and Forge

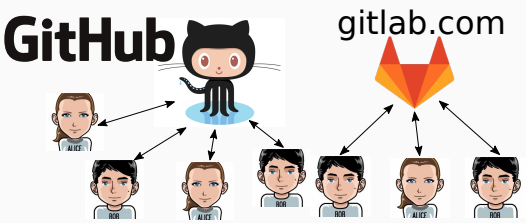
## Git = version control

- Developed in 2005 by Linus Torvalds for the kernel development
- Local and efficient rollbacks
- Distributed: everyone has a full copy of the history



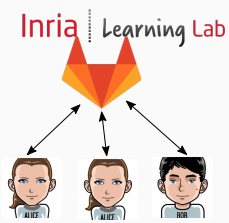

## GitHub, GitLab, and Co

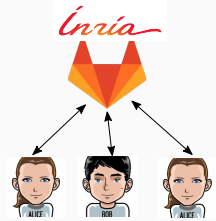
- Free hosting of public projects, social network

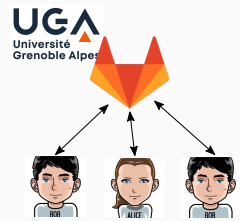


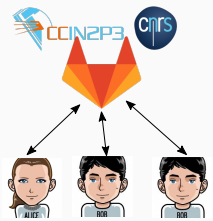







## Limitation

- Managing large data: ~~Git LFS~~ `Git Annex` (or DataLad)

# Tool 3bis: Fighting Information Loss with Archives

 or  = awesome collaborations ($\neq$ archive)

- D. Spinellis. *The Decay and Failures of URL References*. CACM, 46(1), 2003
  *The half-life of a referenced URL is approximately 4 years*
- 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*
- Discontinuated forges: Code Space, Gitorious, Google code, Inria Gforge

# Tool 3bis: Fighting Information Loss with Archives

or

= awesome collaborations (≠ archive)

- D. Spinellis. *The Decay and Failures of URL References*. CACM, 46(1), 2003
  *The half-life of a referenced URL is approximately 4 years*
- 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*
- Discontinuated forges: Code Space, Gitorious, Google code, Inria Gforge

Article archives arXiv.org HAL archives-ouvertes.fr

Data archives figshare zenodo

Software Archive Software Heritage

INTERNET ARCHIVE

Collect/Preserve/Share

Separation between articles, code, and data is not so simple though

## Reinventing the square wheel

Data Version Control with DVC

- ~~`git annex`~~ but special `.dvc` files with information about local/remote storage
- ~~`snakemake`~~ but Data Pipelines trough a `dvc.yaml` task description
- A basic Workflow Management System with little support for running pipelines in parallel

NetFlix (100,000+ jobs per day)

- Notebooks as functions with papermill (Parameterized Notebooks)
- Run in docker environments using MESON (home made workflow orchestration/scheduler)
- Storage in S3, Dashboard like usage to generate daily reports

DeepKit, PolyAxon, Pachyderm, Kono Lord, Have mercy!

- Versionned data without `git`, a centralized location (no conflict, allows removal,...)

Best tool I've seen so far: `git-annex` / `datalad`

# Trend #5: delegating the experiment to a third party

**CASCAD** certification agency for scientific code & data (for economists)

**Continuous Integration** Running experiments through CI

- Main concern: invisibility $\rightsquigarrow$ crazy resource consumption

# Conclusion, challenges

# Concluding thoughts

- *The Structure of Scientific Revolutions* – Thomas Kuhn

For 20 years, reproducibility issues have been a rempant problem, addressed by few isolated groups

# Concluding thoughts

▶ *The Structure of Scientific Revolutions* – Thomas Kuhn

For 20 years, reproducibility issues have been a rempant problem, addressed by few isolated groups  Now, it is everywhere (ACM, ACM EIG, COARA, NIPS, ICML) and can be seen both as a disruption, a threat, or an opportunity, ...

- Benchmarking and challenges are one option
  - positive (federating the community)
  - negative (HPL, NAS PB)
  - sometimes limited gain
  - sometimes inadequate

# Concluding thoughts

▶ *The Structure of Scientific Revolutions* – Thomas Kuhn

For 20 years, reproducibility issues have been a rempant problem, addressed by few isolated groups Now, it is everywhere (ACM, ACM EIG, COARA, NIPS, ICML) and can be seen both as a disruption, a threat, or an opportunity, ...

- Benchmarking and challenges are one option
  - positive (federating the community)
  - negative (HPL, NAS PB)
  - sometimes limited gain
  - sometimes inadequate

▶ *Anything goes* – Paul Feyerabend

- Tension between original/groundbreaking research and strengthening foundations Imposing RR globally and too soon can be deleterious

# Concluding thoughts

▶ *The Structure of Scientific Revolutions* – Thomas Kuhn

For 20 years, reproducibility issues have been a rempant problem, addressed by few isolated groups Now, it is everywhere (ACM, ACM EIG, COARA, NIPS, ICML) and can be seen both as a disruption, a threat, or an opportunity, ...

- Benchmarking and challenges are one option
  - positive (federating the community)
  - negative (HPL, NAS PB)
  - sometimes limited gain
  - sometimes inadequate

▶ *Anything goes* – Paul Feyerabend

- Tension between original/groundbreaking research and strengthening foundations Imposing RR globally and too soon can be deleterious

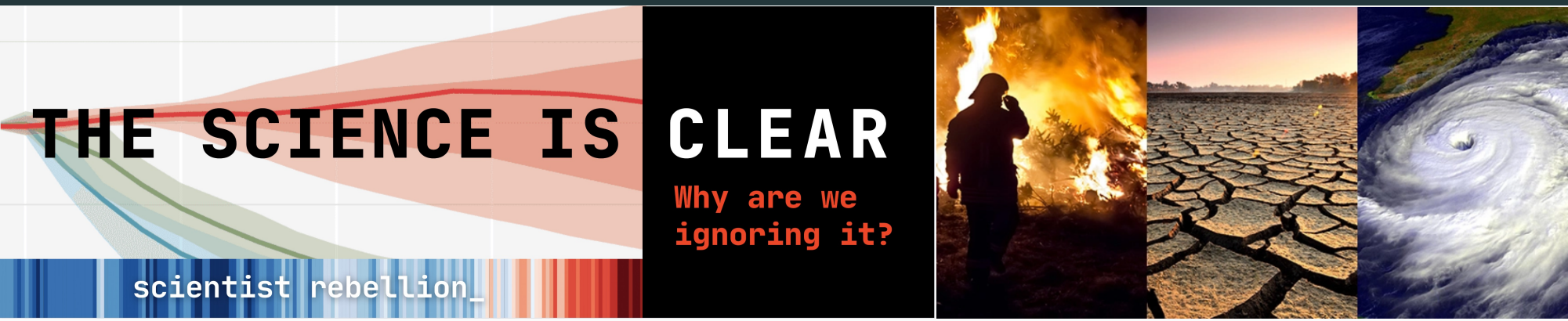
It has become insanely complex! Level up with our PhD students.

- MOOC Reproducible Research: Methodological principles for a transparent science
- 3rd Edition: 2020 – 2024 (16,000+) | MOOC "Advanced RR" Feb. 2024
- Slow down and keep things simple!

***Monster models imply non-reproducible science, without even mentioning sustainability concerns and applicability concerns*** – *TAU*

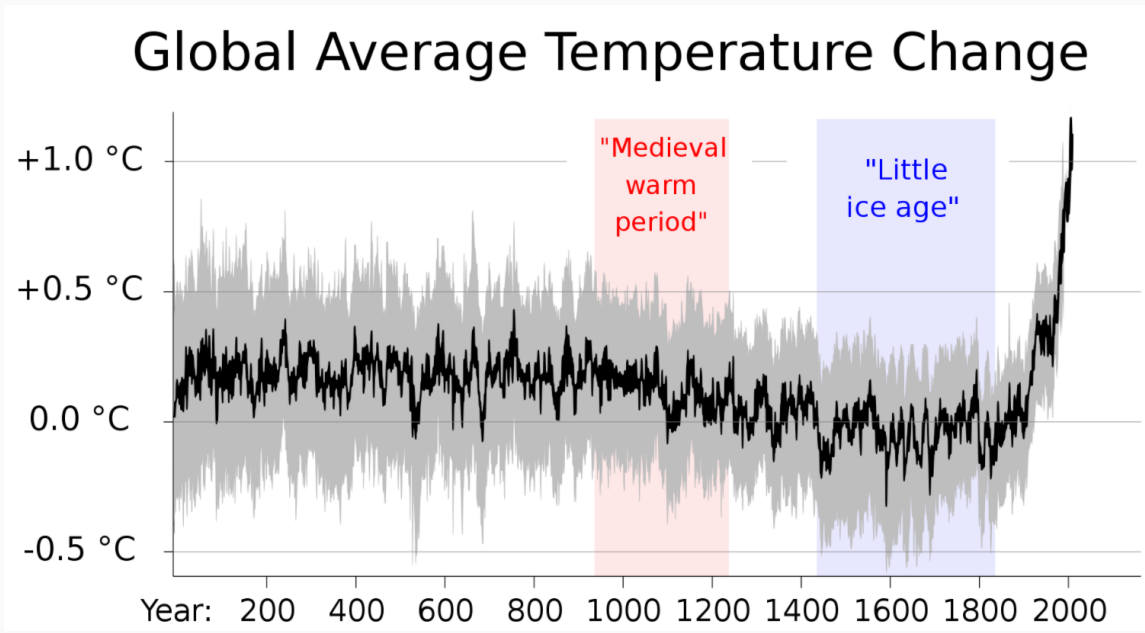
# The Elephant in the Room: Climate Change



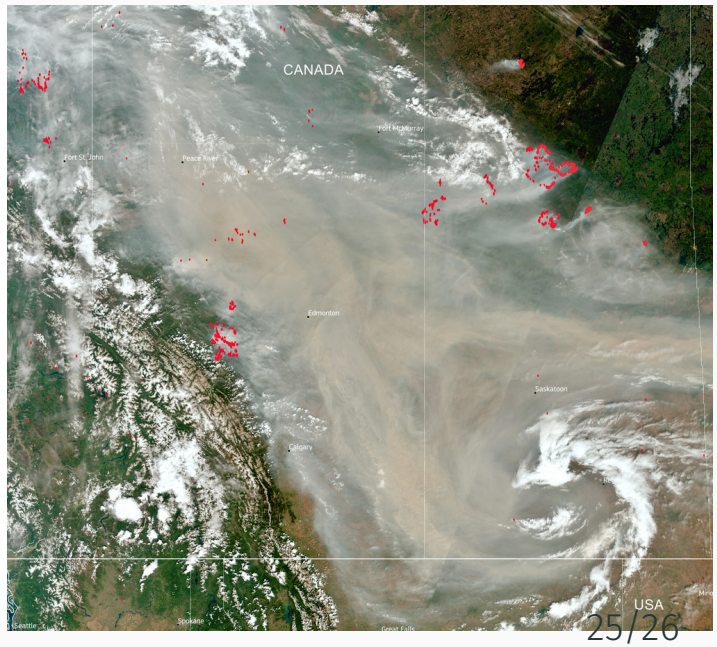



IPCC, IPBES, https://climate.nasa.gov/

1. Global climate change is **not** a future problem



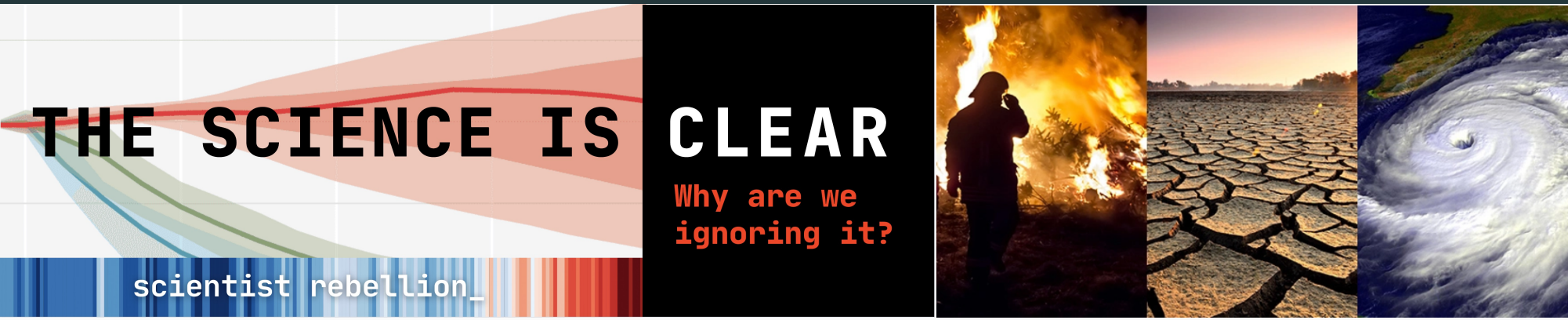

https://en.wikipedia.org/wiki/Global_temperature_record




2023 Alberta wildfires ($> 1$ Mha)

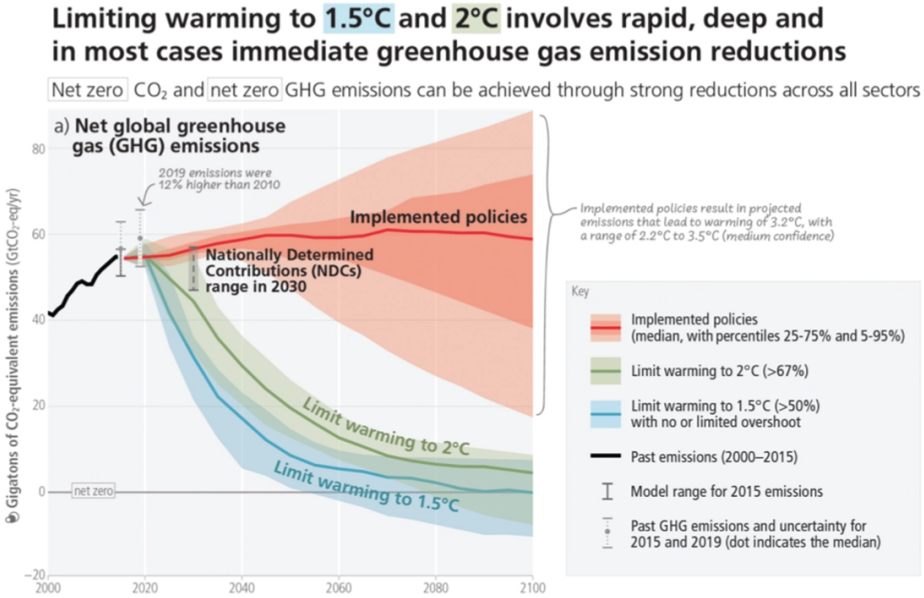
# The Elephant in the Room: Climate Change



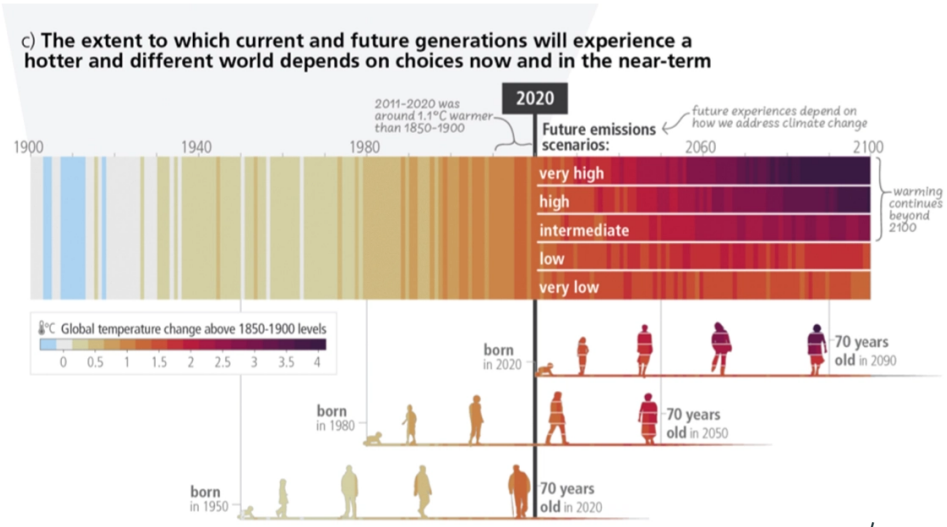



IPCC, IPBES, https://climate.nasa.gov/

1. Global climate change is **not** a future problem
2. It is **entirely** due to human activity



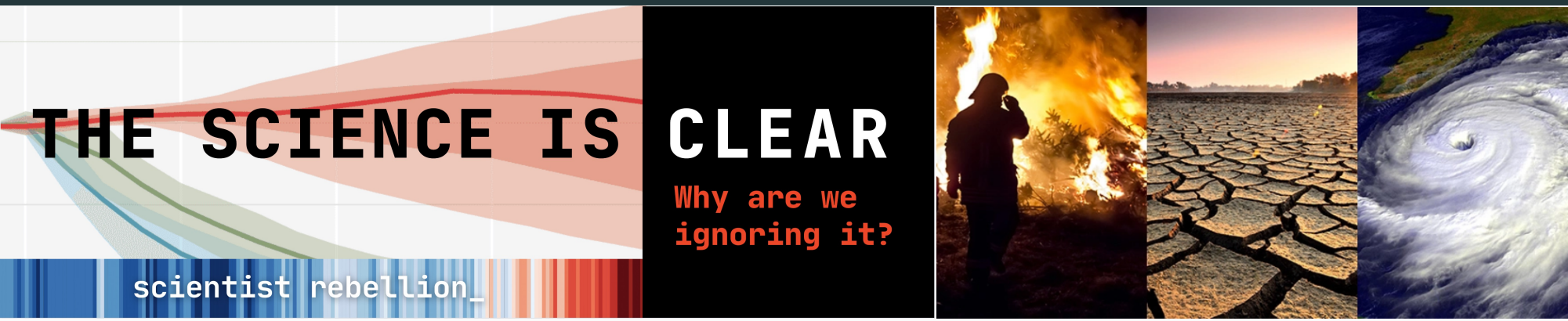

Paris Agreement'15 ∼ Net Zero by 2050




Latest IPCC report

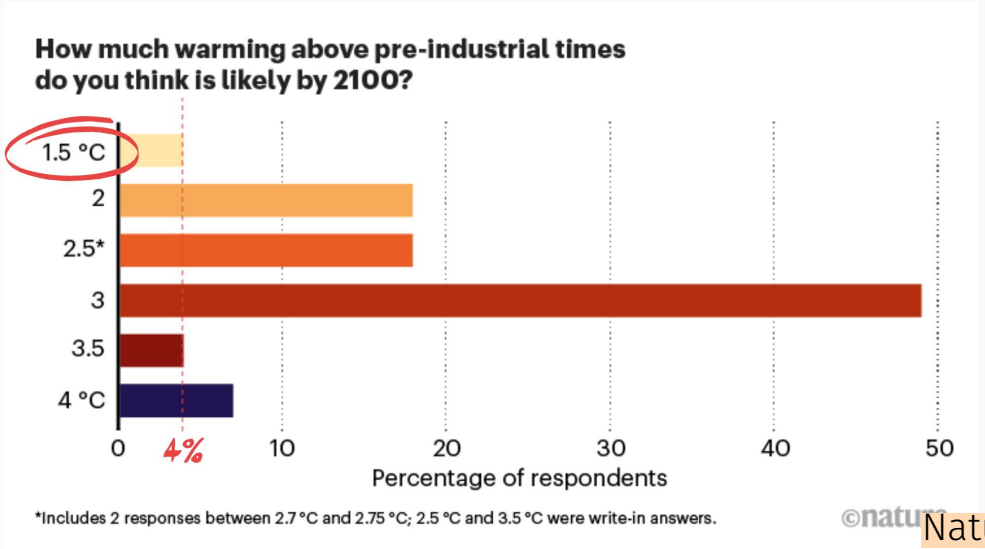
# The Elephant in the Room: Climate Change





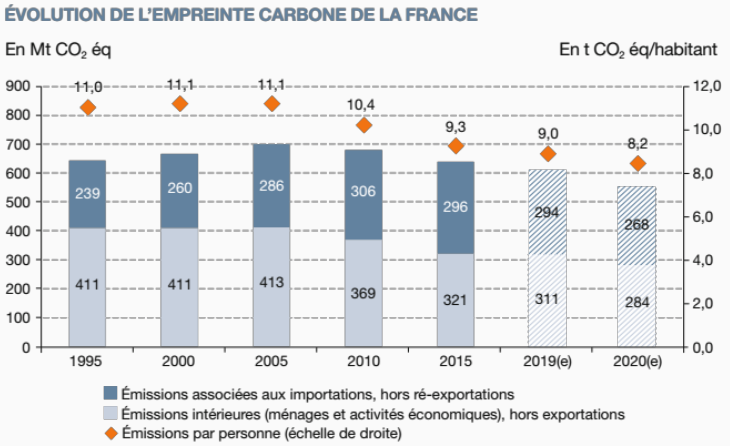

IPCC, IPBES, https://climate.nasa.gov/

1. Global climate change is **not** a future problem
2. It is **entirely** due to human activity
3. *9 out of 10 IPCC scientists believe overshoot is likely*

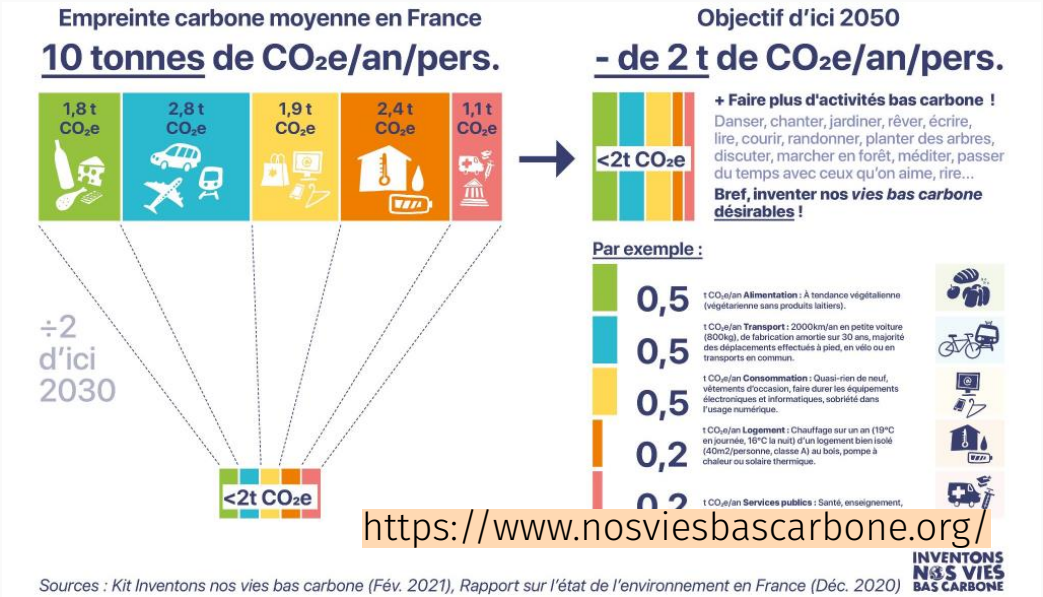



Nature survey, Nov. 2021

## Put aside biodiversity loss, pollution, freshwater, land system change...

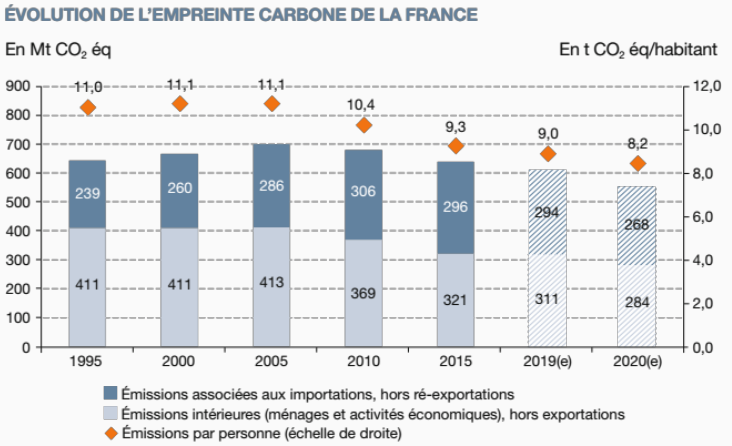


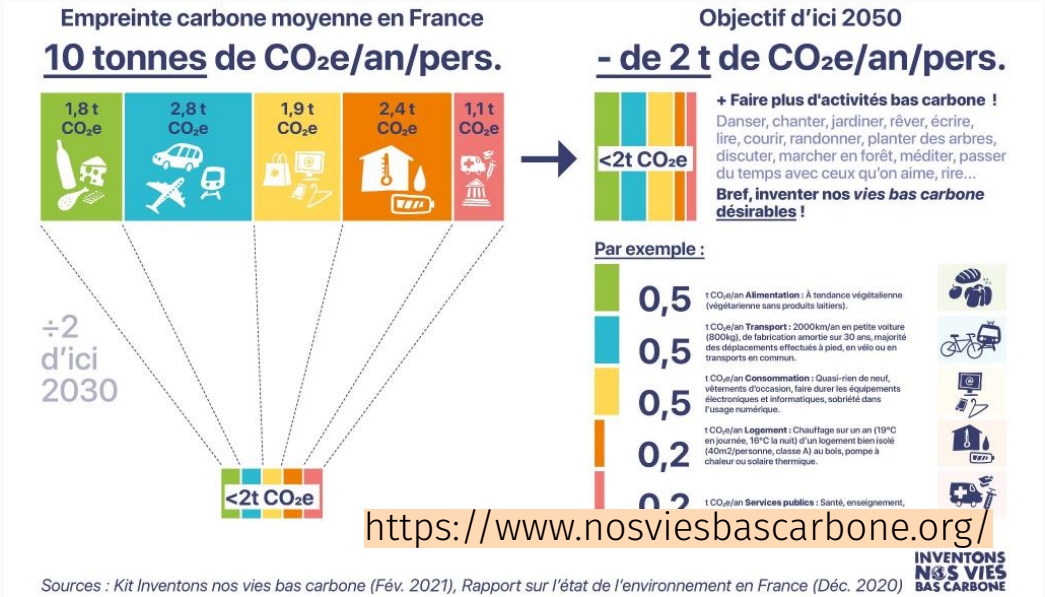

*(e) = estimations.*
*Note : l'empreinte carbone porte sur les trois principaux gaz à effet de serre ($CO_2$, $CH_4$, $N_2O$). En 2021, la méthodologie a été ajustée afin de mieux tenir compte de l'évolution des coûts du pétrole brut, du gaz et du charbon ; l'ensemble de la série a ainsi été révisée, l'essentiel des ajustements portant sur les émissions importées de $CH_4$.*
*Champ : périmètre Kyoto (métropole et outre-mer appartenant à l'UE).*
***Sources :** Citepa ; AIE ; FAO ; Douanes ; Eurostat ; Insee. Traitement : SDES, 2021*




*Sources : Kit Inventons nos vies bas carbone (Fév. 2021), Rapport sur l'état de l'environnement en France (Déc. 2020)*

## Put aside biodiversity loss, pollution, freshwater, land system change…



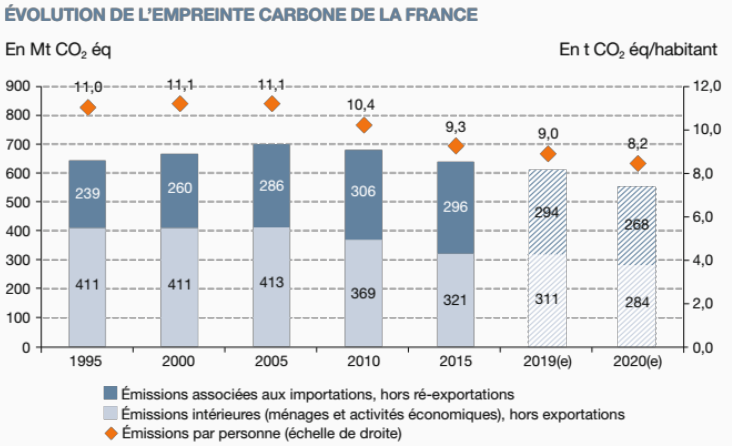



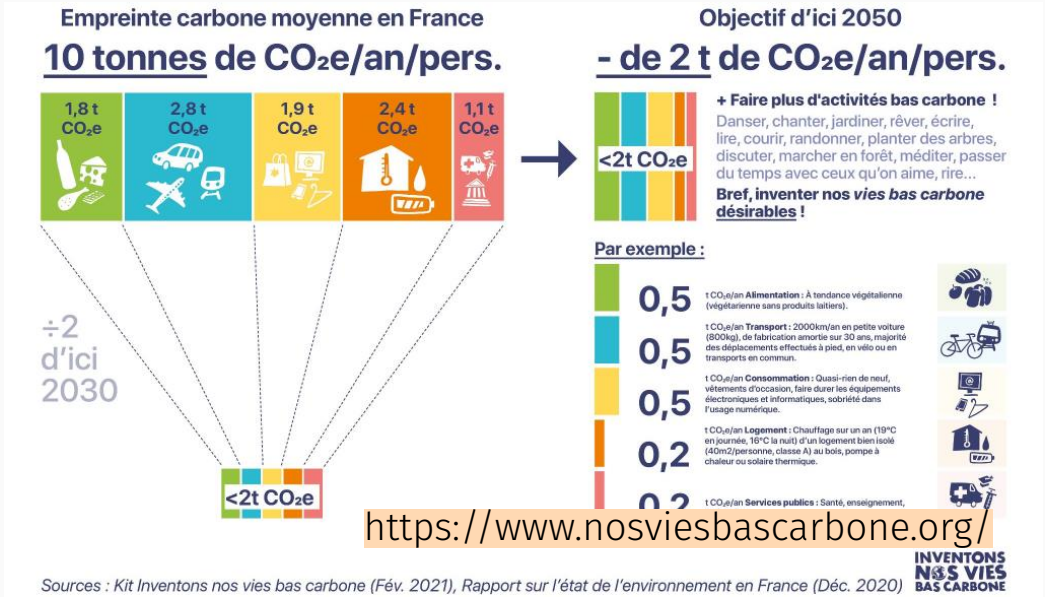

https://www.nosviesbascarbone.org/

## French government response

- *Verdissement de l'industrie: « pause » sur les normes environnementales*
- *Loi de programmation militaire (+41%)*
- *Nous devons préparer la France à une élévation de la température de 4 °C*
- Academia ? PEPR 5G, Cloud, NUMPEX, Quantique, IA, Agroécologie et numérique

## Put aside biodiversity loss, pollution, freshwater, land system change...




## French government response

- *Verdissement de l'industrie: « pause » sur les normes environnementales*
- *Loi de programmation militaire (+41%)*
- *Nous devons préparer la France à une élévation de la température de 4 °C*
- Academia ? PEPR 5G, Cloud, NUMPEX, Quantique, IA, Agroécologie et numérique

## ~~Several~~ scenarios on the table

- What will research/CS look like/be used for in such a world?
- Energy optimization/saving ≠ sobriety and frugality