

# Reproductibilité et Cahiers de Laboratoire

Frédéric Suter

`frederic.suter@cc.in2p3.fr`

09 mars 2020

# Crédits

- ▶ Supports très largement inspirés du MOOC  
*Recherche reproductible: principes méthodologiques pour une science transparente*
  - ▶ Christophe Pouzat
    - ▶ CNRS/MAP5, Université Paris-Descartes
  - ▶ Arnaud Legrand
    - ▶ CNRS, équipe Inria Polaris, Université de Grenoble
  - ▶ Konrad Hinsen
    - ▶ CNRS, Centre de Biophysique Moléculaire à Orléans, Synchrotron SOLEIL
- ▶ Inria Learning Lab
  - ▶ Plate-forme FUN
  - ▶ Prochaine session à partir du 20 mars 2020

# Plan

Cahiers de notes / Cahiers de laboratoire

Document computationnel / Pour plus de reproductibilité

Analyse répliquable / Etude de cas

Conclusion

# Plan

Cahiers de notes / Cahiers de laboratoire

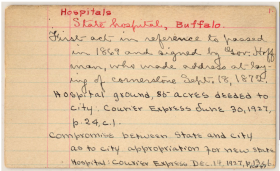
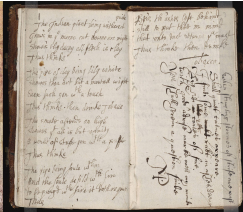
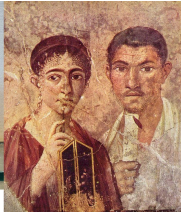
Document computationnel / Pour plus de reproductibilité

Analyse répliquable / Etude de cas

Conclusion



# Les supports de notes à travers les âges



# Galilée qui observe les lunes de Jupiter

Sc<sup>to</sup> Príncipe.

Galileo Galilei, Famulus. Servus della Ser.<sup>a</sup> V.<sup>a</sup> inuigilando  
 do studiuo, et de ogni spirito p. potere no solo sciscifare  
 alario che viene della liberta di Mathematicis nelle Scu-  
 la di Padova,

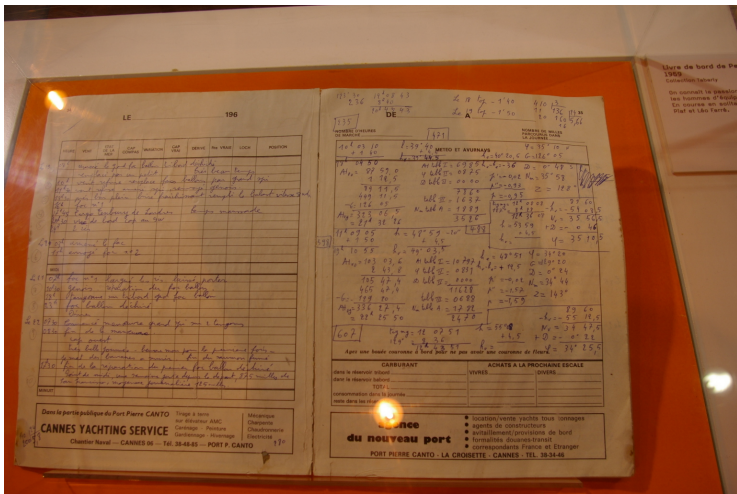
---

Inuere d'auore determinato di presentare al Sc<sup>to</sup> Príncipe  
 l'archile et il p. essere di formamenti inestabile, p. ogni  
 reggia et in breua marittima o terrestre spacio di tempo p. u-  
 sto nuovo artificio ne l'ingegno soggetto et usato a disposizione  
 di v. Ser.<sup>a</sup> L'ingegno canato nelle piu re d'ite speculazioni di  
 prospettiva in l'usaggio di scoprire l'opre et Vele dell' inuisis  
 p. v. acore et piu di tutto prima et ogni usanza rei et distinguendo  
 A numero et la qualita dei d'istelli giudicare la piu felice  
 p. all'opre et alla carca al amabilemente o alla fuga, o pure anco  
 nella usanza sperta vedere et partirla. Distinguire ogni suo  
 mo et propriamente.

1610. 7. di Gennaio



# Le Navigateur et son livre de bord : Éric Tabarly



Dans le port public du Port Pierre CANTO

**CANNES YACHTING SERVICE**

Chantier Naval - CANNES 06 - Tél. 38-48-85 - PORT P. CANTO

Trage à terre  
par électricité A.M.C.  
Cannage Peinture  
Carénage, Remorquage  
Electricité

Mécanique  
Chaudières  
Oubolisme  
Electricité

70

**CANBUBANT**

dans le réservoir habitué

dans le réservoir habitué

compréhension dans 15 minutes

note dans les 15

**ACHATS A LA PROMENADE ESCALE**

location/vente yachts tous usages

agents de constructeurs

équipementiers professionnels de bord

formalités douanes/transit

correspondants France et Etranger

**du nouveau port**

PORT PIERRE CANTO - LA CROISSETTE - CANNES - TEL. 38-34-66

## Quelques questions importantes à se poser

- ▶ Quel support utiliser ?
- ▶ Comment organiser sa prise de notes ?
- ▶ Comment gérer les évolutions ?
- ▶ Comment exporter ses notes ?
- ▶ Comment partager/diffuser ses notes ?

# Quel(s) support(s) matériel(s) pour les notes ?

Doit-on utiliser

- ▶ l'objet d'étude (comme annoter un livre)
- ▶ un ou des cahiers
- ▶ des fiches ou feuilles volantes stockées dans un classeur
- ▶ un ou des fichiers d'ordinateur
- ▶ des dessins ou photos
- ▶ des films
- ▶ ... ?

# Quel(s) support(s) matériel(s) pour les notes ?

Doit-on utiliser

- ▶ l'objet d'étude (comme annoter un livre)
- ▶ un ou des cahiers
- ▶ des fiches ou feuilles volantes stockées dans un classeur
- ▶ un ou des fichiers d'ordinateur
- ▶ des dessins ou photos
- ▶ des films
- ▶ ... ?

Privilégier si possible un support numérique pour profiter

- ▶ d'une plus grande flexibilité d'organisation, de réorganisation et de structuration
- ▶ d'outils d'archivage fiables
- ▶ d'outils d'indexation puissants.

# Comment s'y retrouver ?

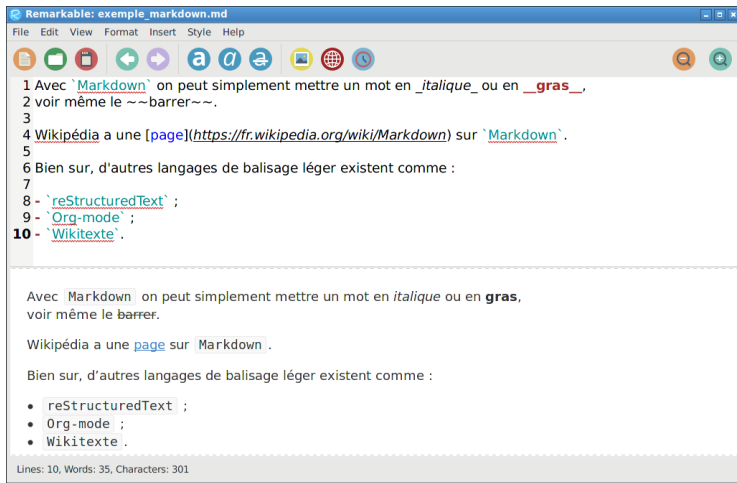
Les notes posent un problème d'organisation

- ▶ Comment imposer une structure à nos notes?
- ▶ Peut-on les indexer, si oui, comment ?
- ▶ Comment peut-on les rendre pérennes tout en les faisant évoluer ?

Langage de balisage léger

- ▶ fichiers texte attractifs pour la prise de notes
- ▶ écrire rapidement nos notes, avec n'importe quel éditeur, grâce à leur syntaxe simplifiée
- ▶ organiser nos notes en les structurant.

# L'exemple de Markdown



The screenshot shows a window titled "Remarkable: exemple\_markdown.md" with a menu bar (File, Edit, View, Format, Insert, Style, Help) and a toolbar. The main content area is split into two sections by a dashed line. The top section contains the raw Markdown source code, and the bottom section shows the rendered HTML output.

```
1 Avec `Markdown` on peut simplement mettre un mot en italique ou en gras,  
2 voir même le barrer.  
3  
4 Wikipédia a une [page](https://fr.wikipedia.org/wiki/Markdown) sur `Markdown`.  
5  
6 Bien sur, d'autres langages de balisage léger existent comme :  
7  
8 - `reStructuredText` ;  
9 - `Org-mode` ;  
10 - `Wikitexte`.
```

---

Avec Markdown on peut simplement mettre un mot en *italique* ou en **gras**,  
voir même le ~~barrer~~.

Wikipédia a une [page](https://fr.wikipedia.org/wiki/Markdown) sur Markdown .

Bien sur, d'autres langages de balisage léger existent comme :

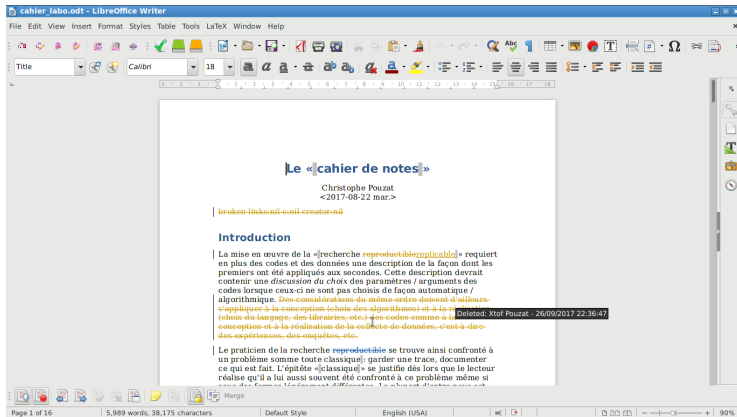
- reStructuredText ;
- Org-mode ;
- Wikitexte .

Lines: 10, Words: 35, Characters: 301

Markdown source (en haut) et sortie HTML (en bas)



# L'évolutivité avec traitement de texte



# L'évolutivité avec la gestion de version

RR\_MOOC/PITCHME.md at master · alegrand/RR\_MOOC - Conkeror

This repository Search Pull requests Issues Marketplace Explore

alegrand / RR\_MOOC Unwatch 5 Star 1 Fork 0

Code Issues Pull requests Projects Wiki Insights

Branch: master RR\_MOOC / slides-module1 / PITCHME.md Find file Copy path

**christophe-pouzat** Ajustements fins partie 2 module 1. 5a2951f an hour ago  
1 contributor

274 lines (176 sloc) 12.6 KB Raw Blame History

## C028AL-W3-S1

+++

### 1. Cahier de notes / cahier de laboratoire

- Nous utilisons tous des cahiers de notes
- Un aperçu historique de la prise de notes
- Du fichier texte au langage de balisage léger
- Pérennité et évolutivité des notes avec la gestion de version
- Les étiquettes et les logiciels d'indexation pour s'y retrouver

https://github.com/alegrand/RR\_MOOC/blob/master/slides-module1/PITCHME.md 16:35 (100, 0)  
Done

History for slides-module1/PITCHME.md - alegrand/RR\_MOOC - Conkeror




This repository Search Pull requests Issues Marketplace Explore

alegrand / RR\_MOOC Unwatch 5 Star 1 Fork 0




Code Issues 0 Pull requests 0 Projects 0 Wiki Insights -

History for RR\_MOOC / slides-module1 / PITCHME.md

Commits on Oct 2, 2017

-  **Ajustements fins partie 2 module 1.**  
christophe-pouzat committed an hour ago [5a2951f](#) [↔](#)
-  **Ajustements partie 2 module 1**  
christophe-pouzat committed 2 hours ago [5cb69f8](#) [↔](#)
-  **Module 1 partie 2 avec figures.**  
christophe-pouzat committed 2 hours ago [e6abc0f](#) [↔](#)

Commits on Sep 25, 2017

-  **J'enlève un zoom pas beau...**  
christophe-pouzat committed 7 days ago [707af5b](#) [↔](#)
-  **Avec des gros plans pour le module 1.**  
christophe-pouzat committed 7 days ago [11b6880](#) [↔](#)
-  **Chgts cosmétiques et othographiques W1\_S1.**  
[2001bd1](#) [↔](#)

[https://github.com/alegrand/RR\\_MOOC/commits/master/slides-module1/PITCHME.md](https://github.com/alegrand/RR_MOOC/commits/master/slides-module1/PITCHME.md) 16:38 (100, 0)

Done

12 slides-module1/PITCHME.md

```
@@ -37,7 +37,7 @@ Ici l'image d'une « enveloppe jaune » de Georges Simenon.
```

```
37 37
```

```
38 38 +
```

```
39 39
```

```
40 
```

```
+40 
```

```
41 41
```

```
42 42
```

```
43 43 +
```

```
@@ -156,6 +156,16 @@ D'après Frédéric Barbier dans l'« Histoire du livre » :
```

```
156 
```

```
157 157
```

```
158 158 +
```

```
159 +
```

```
+## Conclusions
```

```
160 +
```

```
161 +
```

```
162 +- comme il est rarement possible de se passer complètement d'un support papier, apprendre de nos brillants prédécesseurs devrait nous permettre de ne pas « réinventer la roue » ;
```

```
163 +- clairement nous avons néanmoins intérêt à utiliser autant que possible un support numérique pour profiter, en nous inspirant de ces mêmes prédécesseurs, :
```

```
164 + * d'une plus grande flexibilité d'organisation ;
```

## Avantages et inconvénients

- ▶ Solution sophistiquée (donc un peu plus difficile à maîtriser que les précédentes)
- ▶ Solution qui a fait ses preuves, en particulier dans un cadre collaboratif sur de grands projets (noyau Linux)
- ▶ Permet d'enregistrer des modifications sur plusieurs fichiers à la fois
- ▶ Une sauvegarde centralisée dont tous les membres du projet ont une copie intégrale

# Conclusion

- ▶ La prise de notes est la première étape vers la reproductibilité
- ▶ Toute prise de notes est bonne à prendre, **mais**
  - ▶ Certains outils sont plus efficaces que d'autres
  - ▶ Structurer et indexer est essentiel
  - ▶ Pérenniser et partager aussi!
- ▶ Quelques conseils
  - ▶ Support numériques
  - ▶ Langages de balisage léger
  - ▶ Gestion de version

# Plan

Cahiers de notes / Cahiers de laboratoire

Document computationnel / Pour plus de reproductibilité

Analyse répliquable / Etude de cas

Conclusion

# Le document computationnel – Les grandes lignes

What your research supposedly looks like:

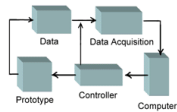


Figure 1. Experimental Diagram

What your research *actually* looks like:

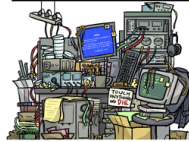


Figure 2. Experimental Mess

WWW.PHDCONICS.COM JURISE CHAN © 2008

1. Exemples récents d'études assez discutées
2. Pourquoi est-ce difficile ?
3. Le document computationnel : principe
4. Travailler avec les autres
5. Installation/Prise en main d'un outil
  - ▶ Rstudio
  - ▶ Org-Mode
6. Analyse comparée des différents outils



# Le document computationnel – Les grandes lignes

What your research supposedly looks like:

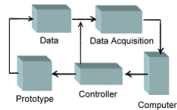


Figure 1. Experimental Diagram

What your research *actually* looks like:

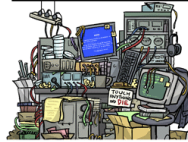


Figure 2. Experimental Mess

WWW.PHDCONICS.COM JURISE CHAN © 2008

1. Exemples récents d'études assez discutées
2. Pourquoi est-ce difficile ?
3. Le document computationnel : principe
4. Travailler avec les autres
5. Installation/Prise en main d'un outil
  - ▶ Rstudio
  - ▶ Org-Mode
6. Analyse comparée des différents outils

# Économie : politiques d'austérité (1/2)

2010

*Lorsque la dette extérieure brute atteint 60 pourcents du PIB, la croissance annuelle d'un pays diminue de deux pourcents.*

*[..] pour des niveaux de dette extérieure dépassant 90 pourcents du PIB, la croissance annuelle est à peu près divisée par deux.*

*– Reinhart et Rogoff: Growth in a Time of Debt*

## Économie : politiques d'austérité (2/2)

2013

*En utilisant leurs feuilles Excel, nous avons identifié des **erreurs de programmation**, des **exclusions** de certaines données, et des pondérations **statistiques non conventionnelles**.*

*– Herndon, Ash et Pollin*

*R&R combinent des données de siècles différents, des régimes de changes différents, des dettes privées et publiques, et des dettes exprimées en monnaies étrangères et nationales.*

*– Wray*

# IRM fonctionnelle

- ▶ 2010 : Bennett et al. et le saumon mort 😊
- ▶ 2016 : Eklund, Nichols, and Knutsson. A bug in fmri software could invalidate 15 years of brain research (*40 000 articles*)
- ▶ 2016 : Mais c'est plus subtil que ça. Nichols.  $\approx 3\ 600$  études concernées

Des méthodes statistiques à améliorer mais pas de remise en cause fondamentale.



## Crise de foi ?

- ▶ Oncologie : "*plus de la moitié des études publiées, même dans des journaux prestigieux, ne peuvent être reproduites en laboratoire industriel*"
- ▶ Psychologie : "*réplication d'une centaine d'articles seulement un tiers de résultats cohérents*"



Lanceurs d'alerte ou institutions malades ?

La remise en cause fait partie du processus scientifique

## Crise de foi ?

- ▶ Oncologie : "*plus de la moitié des études publiées, même dans des journaux prestigieux, ne peuvent être reproduites en laboratoire industriel*"
- ▶ Psychologie : "*réplication d'une centaine d'articles seulement un tiers de résultats cohérents*"



Lanceurs d'alerte ou institutions malades ?

La remise en cause fait partie du processus scientifique

Tout comme la rigueur et la transparence...

# Le document computationnel – Les grandes lignes

What your research supposedly looks like:

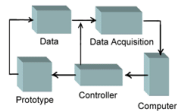


Figure 1. Experimental Diagram

What your research *actually* looks like:

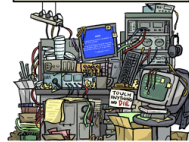


Figure 2. Experimental Mess

WWW.PHDCOMICS.COM JORGE GAMA © 2008

1. Exemples récents d'études assez discutées
2. **Pourquoi est-ce difficile ?**
3. Le document computationnel : principe
4. Installation/Prise en main d'un outil
  - ▶ Jupyter
  - ▶ Rstudio
  - ▶ Org-Mode
5. Travailler avec les autres
6. Analyse comparée des différents outils

# Le manque d'informations

Expliciter :

- ▶ Sources et données

*Données non disponibles = résultats difficiles à vérifier*

- ▶ Choix

*Choix non expliqués = choix suspicieux*

Le cahier de laboratoire peut vous aider



# L'ordinateur, source d'erreurs

- ▶ **Point and click** :
- ▶ **Les tableaux** : erreurs de programmation et de manipulation de données
  - ▶ Membrane-Associated Ring Finger (C3HC4) 1, E3 Ubiquitin Protein Ligase → MARCH1 → 2016-03-01 → 1456786800
  - ▶ 2310009E13 → 2.31E+19
- ▶ **Pile logicielle complexe**
- ▶ **Bug** : *Programmer, c'est dur !*

# L'informatique, seule responsable ?

## Le manque de rigueur et d'organisation

- ▶ Pas de backup
- ▶ Pas d'historique
- ▶ Pas de contrôle qualité

# Une dimension culturelle et sociale

*Article = version simplifiée de la procédure*

*Tracer toutes ces informations et les rendre disponibles = investissement conséquent*

Si personne n'exige/n'inspecte ces informations, à quoi bon s'embêter ?

# Tout rendre public ?

- ▶ Les *faiblesses* deviendraient évidentes
- ▶ Quelqu'un pourrait trouver une *erreur*
- ▶ Quelqu'un pourrait en tirer avantage à ma place
- ▶ *Les données peuvent être sensibles*

Donnons nous les moyens que tout soit inspectable à la demande

# Outils à éviter et alternatives

## ▶ Outils, formats, et services propriétaires

1. ~~Excel, Word, Evernote~~
  - ▶ Markdown, Org-mode, CSV, HDF5, ...
2. ~~SAS, Minitab, matlab, mathematica, ...~~
  - ▶ Scilab, R, Python, ...
3. ~~Dropbox, cahiers de labo en ligne propriétaires, ...~~
  - ▶ Framadrop, GitLab/GitHub, ...

## ▶ Outils "intuitifs"

- ▶ ~~tableurs, interfaces graphiques, exploration interactive~~
  - ▶ apprendre à se contrôler... 😊
  - ▶ R, Python, ...

# Changement de paradigme

1. Manque d'information, problème d'accès aux données
2. Erreurs de calcul
3. Manque de rigueur scientifique et technique



Expliciter augmente les chances de trouver les erreurs  
et de les éliminer

# Le document computationnel – Les grandes lignes

What your research supposedly looks like:

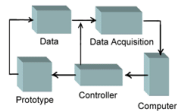


Figure 1. Experimental Diagram

What your research *actually* looks like:

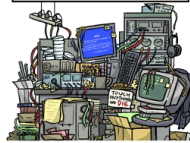
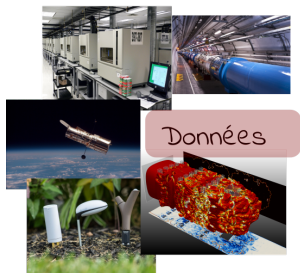


Figure 2. Experimental Mess

WWW.PHPCONICS.COM JURISE CHAN © 2008

1. Exemples récents d'études assez discutées
2. Pourquoi est-ce difficile ?
3. **Le document computationnel : principe**
4. Travailler avec les autres
5. Installation/Prise en main d'un outil
  - ▶ Rstudio
  - ▶ Org-Mode
6. Analyse comparée des différents outils

# La science moderne





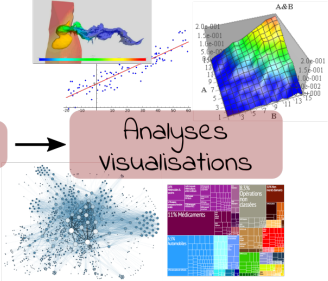
# La science moderne



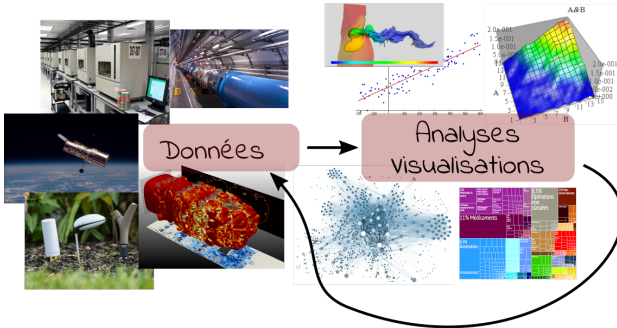
Données



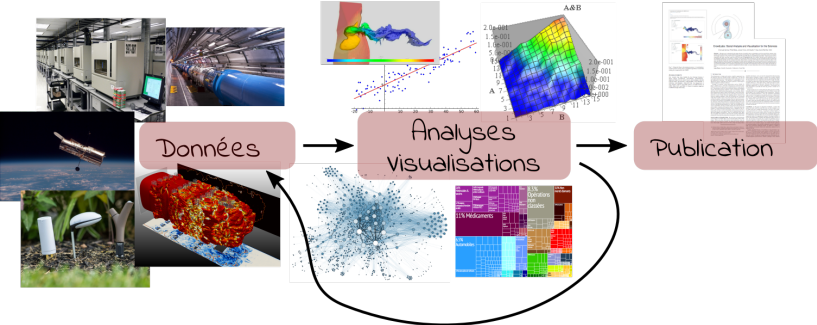
Analyses visualisations



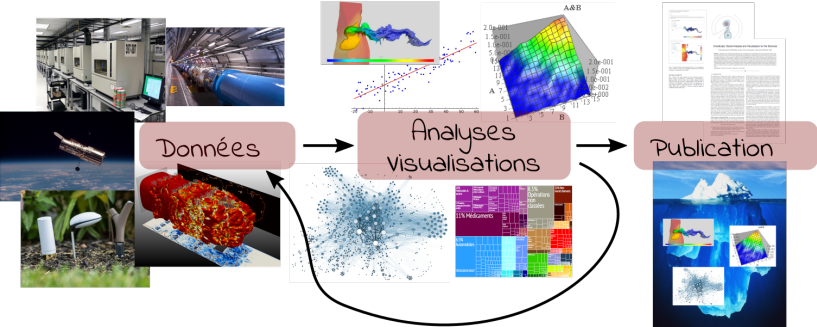
# La science moderne



# La science moderne



# La science moderne



# Objectifs méthodologiques

Garder trace afin de :

- ▶ **Inspecter** : justifier/comprendre
- ▶ **Refaire** : vérifier/corriger/réutiliser

# La vitrine... et l'envers du décor

## Un document computationnel

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

3.141592653589793

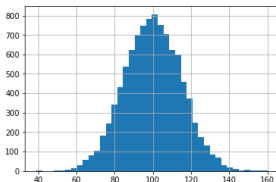
Mais calculé avec la **méthode** des [aiguilles de Buffon](#), on obtiendrait comme **approximation** :

```
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

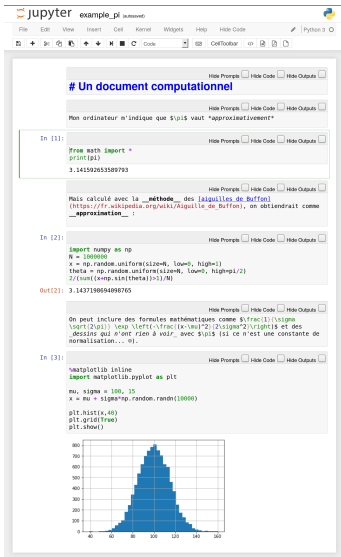
On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right)$  et

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



# La vitrine... et l'envers du décor

## Document initial dans son environnement



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

- Cell 1: A title "# Un document computationnel".
- Cell 2: A text block: "Mon ordinateur m'indique que  $\pi$  vaut 'approximativement'".
- Cell 3: A code cell with the following code:

```
In [1]: from math import *\nprint(pi)\n3.141592653589793
```
- Cell 4: A text block: "Mais calculé avec la 'méthode' des 'aiguilles de Buffon', on obtiendrait comme 'approximation' :".
- Cell 5: A code cell with the following code:

```
In [2]: import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(sum((x+np.sin(theta))>1)/N)\nOut[2]: 3.1437198694098765
```
- Cell 6: A text block: "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... ☺)".
- Cell 7: A code cell with the following code:

```
In [3]: %matplotlib inline\nimport matplotlib.pyplot as plt\nmu, sigma = 100, 10\nx = mu + sigma*np.random.randn(10000)\nplt.hist(x,40)\nplt.grid(True)\nplt.show()
```

The histogram in Cell 7 shows a normal distribution centered at 100, with a range from approximately 60 to 140.

## Document final

### Un document computationnel

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

3.141592653589793

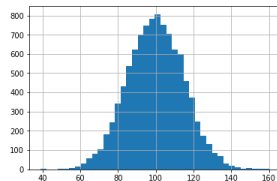
Mais calculé avec la *méthode* des *aiguilles de Buffon*, on obtiendrait comme *approximation* :

```
import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(sum((x+np.sin(theta))>1)/N)
```

3.1437198694098765

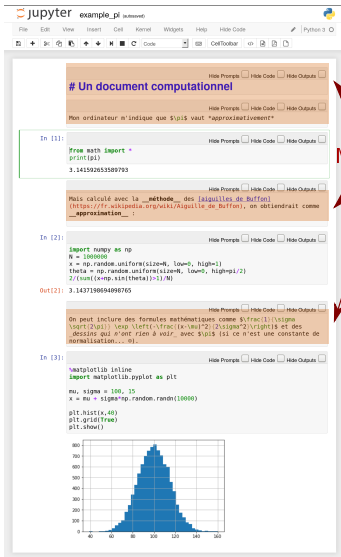
On peut inclure des formules mathématiques comme  $\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$  et

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



# La vitrine... et l'envers du décor

## Document initial dans son environnement



The screenshot shows a Jupyter Notebook window titled "example\_pt | jupyter". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Hide Code) and a toolbar. The notebook content is as follows:

```
# Un document computationnel
```

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

```
In [1]:
```

```
from math import *\nprint(pi)\n3.141592653589793
```

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

```
In [2]:
```

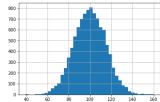
```
import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(sum((x+np.sin(theta))>1)/N)
```

Out[2]: 3.1437198694098765

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

```
In [3]:
```

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



Markdown

## Document final

### Un document computationnel

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

3.141592653589793

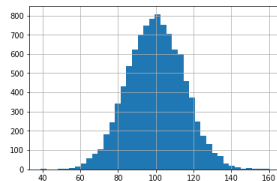
Mais calculé avec la *méthode des aiguilles de Buffon*, on obtiendrait comme *approximation* :

```
import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(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... ☺).





# La vitrine... et l'envers du décor

## Document initial dans son environnement

# Un document computationnel

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

3.141592653589793

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

```
import numpy as np
N = 100000
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... ☺).

```
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

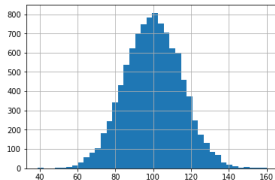
Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```
import numpy as np
N = 100000
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... ☺).



# La vitrine... et l'envers du décor

## Document initial dans son environnement

# Un document computationnel

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

```
In [1]: from math import *\nprint(pi)\n3.141592653589793
```

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

```
In [2]: import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(sum((x*np.sin(theta))>1)/N)\nOut [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\nimport matplotlib.pyplot as plt\nmu, sigma = 100, 15\nx = mu + sigma*np.random.randn(100000)\nplt.hist(x, 40)\nplt.grid(True)\nplt.show()
```

Résultats

## Document final

### Un document computationnel

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

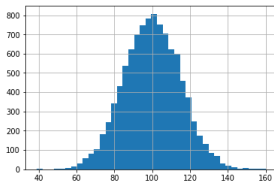
3.141592653589793

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

```
import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(sum((x*np.sin(theta))>1)/N)
```

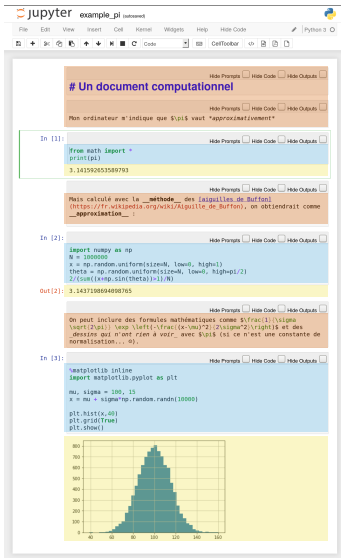
3.1437198694098765

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... ☺).



# La vitrine... et l'envers du décor

## Document initial dans son environnement



The screenshot shows a Jupyter Notebook window titled "example\_pi | jupyter". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Hide Code) and a toolbar. The notebook content is as follows:

- Cell 1:** A title cell with the text "# Un document computationnel".
- Cell 2:** A text cell containing the sentence "Mon ordinateur m'indique que  $\pi$  vaut 'approximativement'".
- Cell 3:** A code cell with the following Python code:

```
from math import *\nprint(pi)
```

The output of this cell is the value 3.141592653589793.
- Cell 4:** A text cell containing the sentence "Mais calculé avec la méthode des alguilles de Buffon (https://fr.wikipedia.org/wiki/Alguille\_de\_Buffon), on obtiendrait comme approximation :".
- Cell 5:** A code cell with the following Python code:

```
import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(sum((x*np.sin(theta))>1)/N)
```

The output of this cell is 3.1437198694098765.
- Cell 6:** A text cell containing the sentence "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... ☺)".
- Cell 7:** A code cell with the following Python code:

```
matplotlib inline\nimport matplotlib.pyplot as plt\nmu, sigma = 100, 15\nx = mu + sigma*np.random.randn(10000)\nplt.hist(x, 40)\nplt.grid(True)\nplt.show()
```

The output of this cell is a histogram showing a normal distribution centered at 100, with a peak frequency of approximately 800.

Export



## Document final

### Un document computationnel

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

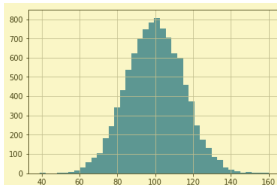
3.141592653589793

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

```
import numpy as np\nN = 1000000\nx = np.random.uniform(size=N, low=0, high=1)\ntheta = np.random.uniform(size=N, low=0, high=pi/2)\n2/(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... ☺).



# Les différents outils

1. Jupyter
2. Rstudio/knitR
3. Org mode

## Principes identiques

- 1 seul document  
(explications, code, resultats)
- Session
- Export

## Différences

- Syntaxe
- Interopérabilité
- Contrôle export

# Le document computationnel – Les grandes lignes

1. Exemples récents d'études assez discutées
2. Pourquoi est-ce difficile ?
3. Le document computationnel : principe
4. **Travailler avec les autres**
5. Installation/Prise en main d'un outil
  - ▶ Rstudio
  - ▶ Org-Mode
6. Analyse comparée des différents outils

# Préparer un document pour un journal

Pré-requis pour faire un pdf :

- ▶ Actuellement caché. En interne *pandoc*, *knitr* ou *emacs/org-mode*
- ▶  $\LaTeX$  installé  
Export *office/word* possible dans jupyter mais à configurer. Sinon export *html*...

Dans tous les cas :

- ▶ Besoin de cacher certaines cellules
- ▶ Utiliser le bon style

Produire un tel document demande d'avoir un environnement parfaitement configuré

# Convaincre vos co-auteurs

Face à cette complexité, plusieurs réactions :

1. Pas grave, c'est génial ! Je m'y mets !
2. Euh... c'est bien. Mais je n'ai pas le temps d'apprendre...
3. Un nouvel outil ? Jamais !

↪ différentes organisations possibles

## Option 1 : les co-auteurs enthousiastes

Il faudra **assurer le service après-vente** :

- ▶ Compatibilité avec les différents environnements
- ▶ Gérer cette complexité (Jupyter/Rstudio/Emacs, Git, ...)

C'est la meilleure façon de **s'assurer que tout est reproductible** et inspectable (et pas uniquement sur votre propre machine...)



## Option 2 : investissement a minima

Vos co-auteurs vous laissent gérer le code, les résultats mais adoptent votre style de document.

Ils peuvent :

- ▶ Éditer le texte de l'article (Markdown ou Org-Mode)

Ils ne peuvent pas :

- ▶ Recalculer
- ▶ Exporter et voir le document final

## Option 3 : les co-auteurs "réfractaires"

Les co-auteurs ne changent pas leurs habitudes

- ▶ Un document *computationnel* séparé produit tous les résultats et toutes les figures
- ▶ Un autre document (*classique*) inclut les figures générées

Mais tout est **conservé**, **documenté** et **recalculable** dans votre document computationnel !

# Publier / partager votre document

## Rpubs

- ▶ Parfait pour partage rapide, pas pérenne

## Dropbox et autres

- ▶ Pérérité, accès ??, ...

## Gitlab/Github/...

1. Rendre public (tout l'historique !)
2. Faire le ménage et archiver l'état courant dans un site compagnon

## Sites compagnons

- ▶ Runmycode, Éditeurs, ...
- ▶ Article : [HAL](#) ; code et données : [Figshare](#) / [Zenodo](#)

# Conclusion

Plusieurs modalités possibles en fonction de :

- ▶ vos co-auteurs
- ▶ vos contraintes techniques
- ▶ vos contraintes de confidentialité/copyright

# Petite démonstration par l'exemple

- ▶ Les conférences et sociétés savantes en informatique encouragent la reproductibilité
  - ▶ Ajout de *badges* sur les versions publiées
- ▶ Mon expérience : Europar 2019
  - ▶ Étude et simulation des jobs traités au CC-IN2P3
  - ▶ Article accepté → Soumission d'un **Experimental Artifact**
    - ▶ Document computationnel décrivant comment les résultats ont été obtenus
    - ▶ Évaluation séparée
- ▶ Utilisation de deux outils différents
  - ▶ Article: Emacs + org-mode + R
  - ▶ Artifact: RStudio + Rmd + R

## Improving Fairness in a Large Scale HTC System Through Workload Analysis and Simulation

Fabrice Averecké<sup>1</sup>, Dalila Khouch<sup>2,3</sup>, and Frédéric Suter<sup>1</sup>

<sup>1</sup> IN2P3 Computing Center / CNRS, Ecole Villesmiers, France  
{fabrice.averocke, frederic.suter}@cc.in2p3.fr  
<sup>2</sup> CNRS/IT - I.L., Paris, Cook Republic  
khouch@in2p3.fr



**Abstract.** Maximizing and analyzing the execution of a workload is at the core of the operation of data centers. It allows operators to verify that the operational objectives are satisfied or detect and react to any unexpected and unwanted behaviors. However, the scale and complexity of large workloads composed of millions of jobs executed each month on several thousands of nodes, often limit the depth of such an analysis. This may lead to overlook some phenomena that, while not harmful at a global scale, can be detrimental to a specific class of users.

In this paper, we illustrate such a situation by analyzing a large High Throughput Computing (HTC) workload trace coming from one of the largest academic computing centers in France. The data shows algorithms at the core of the batch scheduler ensures that all user groups are fairly provided with an amount of computing resources commensurate to their expressed needs. However, a deeper analysis of the produced schedules, especially of the job waiting traces, shows a certain degree of unfairness between user groups. We identify the reconfigurations of the queues and scheduling queues as the main root causes of this unfairness. We then propose a drastic reorganization of the system that aims at being more robust to the characteristics of the workload and at better balancing the waiting trace among user groups. We evaluate the impact of this reorganization through detailed simulations. The obtained results show that it still satisfies the main operational algorithm while significantly improving the quality of service experienced by formerly under-served users.

### 1 Introduction

### Companion to the article Improving Fairness in a Large Scale HTC System Through Workload Analysis and Simulation F. Averocké, D. Khouch, and F. Suter

#### 1 Data preparation

We start the data preparation by setting the origin of time which corresponds to the earliest submission received from the workload log provided by CC-IN2P3. This earliest submission is 2014-03-23 19:03:17.

This is done as follows to format the SHEF file (cf.:

- Remove the column
- Name the sub-workloads from the Eventcode Number/Job
- Add submission dates
- The date, day and hour of submission of a job
- The date interval when a job is running
- The date interval when a job is pending
- Remove unused information

```
format_workload = function(file) {
# Uses the SHEF format:
# format: [[EventCode Number] [Job Number] [Job Type] [ Job Date ]
submit[[1]] = c("Job Number", "Job Type", "Job Date",
               "Number of Submitted Processes", "Average CPU Time Used",
               "Number of Submitted Jobs", "Number of Submitted Nodes",
               "Submitted Average CPU Time", "Submitted Size", "Submitted Number",
               "Submitted Number of Nodes", "Submitted Job Number",
               "Job Date", "Job Pending Job Number")

# Uses the submitted information
dfWorkload[1,] = sample(SHEF.Eventcode Number, function(x)
df[[1]]["EventCode"] = df[2,] ["Job Number"]
# All information on date, day and hour of submission
df[[2,]]["Date"] = as.POSIXlt(df[[3,]]["Submitted Date"], format="%Y-%m-%d %H:%M:%S", origin="1970-01-01 00:00:00")
df[[3,]]["Date"] = as.POSIXlt(df[[4,]]["Date"], format="%Y-%m-%d %H:%M:%S", origin="1970-01-01 00:00:00")
df[[4,]]["Day"] = as.POSIXlt(df[[5,]]["Date"], format="%Y-%m-%d %H:%M:%S", origin="1970-01-01 00:00:00")
df[[5,]]["Run"] = as.POSIXlt(df[[6,]]["Date"], format="%Y-%m-%d %H:%M:%S", origin="1970-01-01 00:00:00")
df[[6,]]["Wait"] = as.POSIXlt(df[[7,]]["Date"], format="%Y-%m-%d %H:%M:%S", origin="1970-01-01 00:00:00")
# Determine the time interval when a job is pending
df[[7,]]["Waiting Submission"] = df[[5,]]["Date"] - df[[6,]]["Date"]
# Remove the useless information
df[[8,]] = df[[8,]]["Job Number"]
df[[9,]] = df[[9,]]["Submitted Date"]
df[[10,]] = df[[10,]]["Submitted Job Number"]
df[[11,]] = df[[11,]]["Submitted Job Number"]
df[[12,]] = df[[12,]]["Submitted Job Number"]
df[[13,]] = df[[13,]]["Submitted Job Number"]
df[[14,]] = df[[14,]]["Submitted Job Number"]
df[[15,]] = df[[15,]]["Submitted Job Number"]
df[[16,]] = df[[16,]]["Submitted Job Number"]
df[[17,]] = df[[17,]]["Submitted Job Number"]
df[[18,]] = df[[18,]]["Submitted Job Number"]
df[[19,]] = df[[19,]]["Submitted Job Number"]
df[[20,]] = df[[20,]]["Submitted Job Number"]
df[[21,]] = df[[21,]]["Submitted Job Number"]
df[[22,]] = df[[22,]]["Submitted Job Number"]
df[[23,]] = df[[23,]]["Submitted Job Number"]
df[[24,]] = df[[24,]]["Submitted Job Number"]
df[[25,]] = df[[25,]]["Submitted Job Number"]
df[[26,]] = df[[26,]]["Submitted Job Number"]
df[[27,]] = df[[27,]]["Submitted Job Number"]
df[[28,]] = df[[28,]]["Submitted Job Number"]
df[[29,]] = df[[29,]]["Submitted Job Number"]
df[[30,]] = df[[30,]]["Submitted Job Number"]
df[[31,]] = df[[31,]]["Submitted Job Number"]
df[[32,]] = df[[32,]]["Submitted Job Number"]
df[[33,]] = df[[33,]]["Submitted Job Number"]
df[[34,]] = df[[34,]]["Submitted Job Number"]
df[[35,]] = df[[35,]]["Submitted Job Number"]
df[[36,]] = df[[36,]]["Submitted Job Number"]
df[[37,]] = df[[37,]]["Submitted Job Number"]
df[[38,]] = df[[38,]]["Submitted Job Number"]
df[[39,]] = df[[39,]]["Submitted Job Number"]
df[[40,]] = df[[40,]]["Submitted Job Number"]
df[[41,]] = df[[41,]]["Submitted Job Number"]
df[[42,]] = df[[42,]]["Submitted Job Number"]
df[[43,]] = df[[43,]]["Submitted Job Number"]
df[[44,]] = df[[44,]]["Submitted Job Number"]
df[[45,]] = df[[45,]]["Submitted Job Number"]
df[[46,]] = df[[46,]]["Submitted Job Number"]
df[[47,]] = df[[47,]]["Submitted Job Number"]
df[[48,]] = df[[48,]]["Submitted Job Number"]
df[[49,]] = df[[49,]]["Submitted Job Number"]
df[[50,]] = df[[50,]]["Submitted Job Number"]
df[[51,]] = df[[51,]]["Submitted Job Number"]
df[[52,]] = df[[52,]]["Submitted Job Number"]
df[[53,]] = df[[53,]]["Submitted Job Number"]
df[[54,]] = df[[54,]]["Submitted Job Number"]
df[[55,]] = df[[55,]]["Submitted Job Number"]
df[[56,]] = df[[56,]]["Submitted Job Number"]
df[[57,]] = df[[57,]]["Submitted Job Number"]
df[[58,]] = df[[58,]]["Submitted Job Number"]
df[[59,]] = df[[59,]]["Submitted Job Number"]
df[[60,]] = df[[60,]]["Submitted Job Number"]
df[[61,]] = df[[61,]]["Submitted Job Number"]
df[[62,]] = df[[62,]]["Submitted Job Number"]
df[[63,]] = df[[63,]]["Submitted Job Number"]
df[[64,]] = df[[64,]]["Submitted Job Number"]
df[[65,]] = df[[65,]]["Submitted Job Number"]
df[[66,]] = df[[66,]]["Submitted Job Number"]
df[[67,]] = df[[67,]]["Submitted Job Number"]
df[[68,]] = df[[68,]]["Submitted Job Number"]
df[[69,]] = df[[69,]]["Submitted Job Number"]
df[[70,]] = df[[70,]]["Submitted Job Number"]
df[[71,]] = df[[71,]]["Submitted Job Number"]
df[[72,]] = df[[72,]]["Submitted Job Number"]
df[[73,]] = df[[73,]]["Submitted Job Number"]
df[[74,]] = df[[74,]]["Submitted Job Number"]
df[[75,]] = df[[75,]]["Submitted Job Number"]
df[[76,]] = df[[76,]]["Submitted Job Number"]
df[[77,]] = df[[77,]]["Submitted Job Number"]
df[[78,]] = df[[78,]]["Submitted Job Number"]
df[[79,]] = df[[79,]]["Submitted Job Number"]
df[[80,]] = df[[80,]]["Submitted Job Number"]
df[[81,]] = df[[81,]]["Submitted Job Number"]
df[[82,]] = df[[82,]]["Submitted Job Number"]
df[[83,]] = df[[83,]]["Submitted Job Number"]
df[[84,]] = df[[84,]]["Submitted Job Number"]
df[[85,]] = df[[85,]]["Submitted Job Number"]
df[[86,]] = df[[86,]]["Submitted Job Number"]
df[[87,]] = df[[87,]]["Submitted Job Number"]
df[[88,]] = df[[88,]]["Submitted Job Number"]
df[[89,]] = df[[89,]]["Submitted Job Number"]
df[[90,]] = df[[90,]]["Submitted Job Number"]
df[[91,]] = df[[91,]]["Submitted Job Number"]
df[[92,]] = df[[92,]]["Submitted Job Number"]
df[[93,]] = df[[93,]]["Submitted Job Number"]
df[[94,]] = df[[94,]]["Submitted Job Number"]
df[[95,]] = df[[95,]]["Submitted Job Number"]
df[[96,]] = df[[96,]]["Submitted Job Number"]
df[[97,]] = df[[97,]]["Submitted Job Number"]
df[[98,]] = df[[98,]]["Submitted Job Number"]
df[[99,]] = df[[99,]]["Submitted Job Number"]
df[[100,]] = df[[100,]]["Submitted Job Number"]
}
```

# Le document computationnel – Les grandes lignes

1. Exemples récents d'études assez discutées
2. Pourquoi est-ce difficile ?
3. Le document computationnel : principe
4. Travailler avec les autres
5. **Installation/Prise en main d'un outil**
  - ▶ **Rstudio**
  - ▶ **Org-Mode**
6. Analyse comparée des différents outils

# Prise en main de Rstudio (1/3)

## Installation

- ▶ Installer Rstudio

## Lancement

- ▶ Ouverture d'un document
- ▶ Description rapide
- ▶ Sauvegarde

## Exécution des blocs

- ▶ Exécution et récupération des résultats
- ▶ Ajout d'un bloc
- ▶ Attention à l'ordre!
  - ▶ notion de session, incohérences possibles
- ▶ Tout réexécuter depuis le début

# Prise en main de Rstudio (2/3)

## Raccourcis clavier et auto-complétion

- ▶ Raccourcis claviers
- ▶ Complétion R
- ▶ Folding

## Production et partage du document final

- ▶ Knit
- ▶ Partage à peu de frais via rpubs

## Contrôler la visibilité du code et des résultats

- ▶ Complétion (paramètres des blocs)



# Prise en main de Rstudio (3/3)

## Utiliser un style particulier

- ▶ pdf,  $\LaTeX$
- ▶ html
- ▶ word/office

Possibilité de faire du  $\LaTeX$  (R Sweave : `Rnw`) ou du html (R `html` : `Rhtml`) directement pour avoir un contrôle parfait.

## Utiliser d'autres langages

- ▶ Ajout et exécution d'un bloc Python
- ▶ Attention, pas de session !
  - ▶ Interaction uniquement via fichiers et dans de longs blocs

# Le document computationnel – Les grandes lignes

1. Exemples récents d'études assez discutées
2. Pourquoi est-ce difficile ?
3. Le document computationnel : principe
4. Travailler avec les autres
5. **Installation/Prise en main d'un outil**
  - ▶ Rstudio
  - ▶ **Org-Mode**
6. Analyse comparée des différents outils

# Prise en main Org Mode (1/3)

## Installation

- ▶ Installer Emacs – Org mode

## Lancement

- ▶ Ouverture d'un document
- ▶ Description rapide
  - ▶ Folding / Navigation / Restructuration
- ▶ Sauvegarde

# Prise en main Org Mode (2/3)

## Exécution des blocs

- ▶ Ajout d'un bloc R
- ▶ Exécution et récupération des résultats
- ▶ Attention à l'ordre
  - ▶ Notion de session
  - ▶ Incohérences possibles
  - ▶ Tout réexécuter depuis le début

## Raccourcis clavier

- ▶ Bloc expansion
  - ▶ R graphique / Python, Perl, ... / Shell session
- ▶ Plusieurs sessions, plusieurs langages !
- ▶ Communication entre langages possible

# Prise en main Org Mode (3/3)

## Production et partage du document final

- ▶ Git Commit
  - ▶ Attention aux fichiers produits
- ▶ Export
- ▶ Visibilité du code et des résultats
  - ▶ Sections cachées

## Utiliser un style particulier

- ▶ pdf,  $\LaTeX$
- ▶ html

# Le document computationnel

1. Exemples récents d'études assez discutées
2. Pourquoi est-ce difficile ?
3. Le document computationnel : principe
4. Travailler avec les autres
5. Prise en main d'un outil
  - ▶ Rstudio
  - ▶ Org-Mode
6. **Analyse comparée des différents outils**

# Un document computationnel, pour faire quoi ?

Un cours ou un tutoriel  $\rightsquigarrow$  Jupyter

- ▶ Facile à prendre en main
- ▶ Document dynamique

Un journal  $\rightsquigarrow$  org-mode

- ▶ Un seul auteur
- ▶ Organisation chronologique
- ▶ Étiquettes
- ▶ Notes, liens, code

# Un document computationnel, pour faire quoi ?

## Un cahier de laboratoire $\rightsquigarrow$ org-mode

- ▶ Organisation sémantique
- ▶ Conventions
- ▶ Plusieurs auteurs
- ▶ Étiquettes pour auteurs, expériences, etc.

## Un article reproductible $\rightsquigarrow$ org-mode

- ▶ Plusieurs auteurs
- ▶ Régénérer les figures
- ▶ Revenir aux sources



# Différences techniques

	Origine	Technologie	Utilisation	Navigation	Format	Article?
Jupyter	2001	Web App., Python	Facile	Limitée	JSON	Difficile
Rstudio/knitr	2011/2014	IDE, Java/R	Facile	Limitée	Rmd	Oui
Org-Mode	1976/2008	Editeur, EmacsLisp	Plus complexe	Puissante	Org	Oui

L'outil importe peu, ce qui importe, c'est :

- ▶ collecter l'information
- ▶ l'organiser et la rendre exploitable
- ▶ la rendre disponible

# Plan

Cahiers de notes / Cahiers de laboratoire

Document computationnel / Pour plus de reproductibilité

Analyse répliquable / Etude de cas

Conclusion

# Analyse répliquable – les grandes lignes

1. **Une analyse répliquable, c'est quoi?**
2. Étude de cas: l'incidence de syndromes grippaux
3. Importer les données
  - ▶ RStudio
  - ▶ OrgMode
4. Vérification et inspection
  - ▶ RStudio
  - ▶ OrgMode
5. Questions et réponses
  - ▶ RStudio
  - ▶ OrgMode

# L'analyse de données traditionnelle

résumé  
méthodologique

résultats

discussion

# L'analyse de données répliquable

code

explication

résultats

discussion

# Pourquoi faire répliquable?

- ▶ Facile à refaire si les données changent
- ▶ Facile à modifier
- ▶ Facile à inspecter et vérifier

# Analyse répliquable – les grandes lignes

1. Une analyse répliquable, c'est quoi?
2. **Étude de cas: l'incidence de syndromes grippaux**
3. Importer les données. Au choix:
  - ▶ RStudio
  - ▶ OrgMode
4. Vérification et inspection. Au choix:
  - ▶ RStudio
  - ▶ OrgMode
5. Questions et réponses. Au choix:
  - ▶ RStudio
  - ▶ OrgMode

# L'incidence de syndromes grippaux

## Taux d'incidence

- ▶ Rapport entre le nombre de nouveaux cas d'une pathologie observés pendant une période donnée et la population dont sont issus les cas (pendant cette même période)

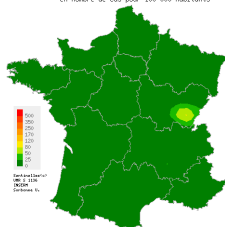
## Réseau Sentinelles

- ▶ <http://www.sentiweb.fr/>
- ▶ Collecte et conserve des données fournies par les médecins
- ▶ Données publiques
  - ▶ et donc analysables!

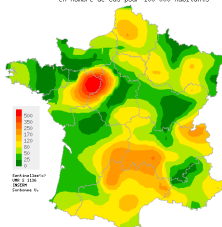


# Exemple d'utilisation des données

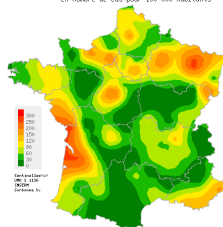
Syndrôme Grippeux Semaine 2019n18  
en nombre de cas pour 100 000 habitants



Syndrôme Grippeux Semaine 2019n01  
en nombre de cas pour 100 000 habitants



Diarrhée aiguë Semaine 2019n18  
en nombre de cas pour 100 000 habitants



# Objectifs

## Questions

1. Quelles années ont connu les épidémies les plus fortes?
2. Quelle est la fréquence d'épidémies faibles, moyennes, et fortes?

## Contraintes

- ▶ Aucune modification des données "à la main".
- ▶ Du code pour tout!

# Analyse répliquable – les grandes lignes

1. Une analyse répliquable, c'est quoi?
2. Étude de cas: l'incidence de syndromes grippaux
3. **Importer les données**
  - ▶ RStudio
  - ▶ OrgMode
4. Vérification et inspection
  - ▶ RStudio
  - ▶ OrgMode
5. Questions et réponses
  - ▶ RStudio
  - ▶ OrgMode

# Importer les données

## RStudio

- ▶ R
- ▶ Bibliothèque: `parsedate`

## OrgMode

- ▶ Python 3 pour préparer les données
- ▶ R pour l'analyse

Attention aux données manquantes

# Étapes à suivre

- ▶ Télécharger les données
  - ▶ incidence-PAY-3.csv
- ▶ Reporter le schéma des données dans le notebook
  - ▶ csv-schema-v1.json
- ▶ Vérifier (rapidement) le contenu de la table
- ▶ Chercher les données manquantes
- ▶ Extraire les colonnes d'intérêt (**week** et **inc**)

# Analyse répliquable – les grandes lignes

1. Une analyse répliquable, c'est quoi?
2. Étude de cas: l'incidence de syndromes grippaux
3. Importer les données
  - ▶ RStudio
  - ▶ OrgMode
4. **Vérification et inspection**
  - ▶ RStudio
  - ▶ OrgMode
5. Questions et réponses
  - ▶ RStudio
  - ▶ OrgMode

# Étapes à suivre

- ▶ Vérifier que les données de **week** sont au bon format
  - ▶ entier (voire avec 6 chiffres)
- ▶ Convertir pour faciliter l'analyse
  - ▶ R: utiliser le lundi
- ▶ Vérifier qu'on a bien 7 jours d'écart
- ▶ Inspecter les données
  - ▶ Tracer le graphe des taux d'incidence
  - ▶ Zoomer sur la période 2015-2019

# Analyse répliquable – les grandes lignes

1. Une analyse répliquable, c'est quoi?
2. Étude de cas: l'incidence de syndromes grippaux
3. Importer les données
  - ▶ Jupyter
  - ▶ RStudio
  - ▶ OrgMode
4. Vérification et inspection
  - ▶ Jupyter
  - ▶ RStudio
  - ▶ OrgMode
5. **Questions et réponses**
  - ▶ RStudio
  - ▶ OrgMode



# Étapes à suivre

- ▶ Les pics sont en hiver!
  - ▶ Considérer que les années vont de Août à Août
- ▶ Éliminer les années incomplètes
- ▶ Vérifier la consistance des nombres de semaines par an
- ▶ Tracer le graphes des incidences annuelles
- ▶ Identifier les épidémies les plus fortes
  - ▶ Avec une liste triée
  - ▶ En traçant un histogramme

## Les points clés à retenir

- ▶ Une analyse répliquable doit contenir **toutes les étapes** de traitement des données sous une forme **exécutable**.
- ▶ Il est important d'**expliquer** tous les choix qui peuvent influencer les résultats.
- ▶ Ceci nécessite d'exposer beaucoup de **détails techniques**, parce que c'est à ce niveau qu'on fait **le plus d'erreurs!**

# Plan

Cahiers de notes / Cahiers de laboratoire

Document computationnel / Pour plus de reproductibilité

Analyse répliquable / Etude de cas

**Conclusion**

# Conclusion – Ce qu'il faut retenir de ce cours

## Un véritable enjeu

- ▶ Méthodologie scientifique
- ▶ Inspectabilité et réutilisation

## Des outils existent

- ▶ Documents computationnels et Workflows, Suivi de version et Archives, Environnements logiciels, Intégration continue. . .
- ▶ Ces outils évoluent en permanence
- ▶ Choisissez ceux qui sont les plus adaptés à votre contexte
- ▶ Cherchez un compromis entre modernisme et pérenité

## Mettez en pratique, ne vous découragez pas!

- ▶ Prenez des notes rigoureusement
- ▶ Rendez l'information exploitable et accessible
- ▶ Améliorez petit à petit