

# HLMA408: Traitement des données

Préambule, informations, etc.

**Joseph Salmon**

<http://josephsalmon.eu>

Université de Montpellier



# Sommaire

Informations pratiques

Conseils numériques: pour le cours et au-delà

Python pour la science des données

# Sommaire

## Informations pratiques

L'équipe pédagogique

Modalités de contrôle de connaissance (MCC)

Ressources et prérequis

Conseils numériques: pour le cours et au-delà

Python pour la science des données

# Sommaire

## Informations pratiques

L'équipe pédagogique

Modalités de contrôle de connaissance (MCC)

Ressources et prérequis

Conseils numériques: pour le cours et au-delà

Python pour la science des données

# Enseignant: cours magistral

- **Joseph Salmon :**

- ▶ Situation actuelle: Professeur à l'université de Montpellier
- ▶ Précédemment: Paris Diderot-Paris 7, Duke University, Télécom ParisTech, University of Washington
- ▶ Spécialités: statistiques en grande dimension, optimisation, agrégation, traitement des images
- ▶ Bureau: 415, Bat. 9

Contact:

Joseph Salmon

✉ [joseph.salmon@umontpellier.fr](mailto:joseph.salmon@umontpellier.fr)

🌐 <http://josephsalmon.eu>

**Github:** @josephsalmon



**Twitter:** @salmonjsph



# Enseignants: TDs/TPs

- **Thierry Mignon :**

- ▶ Situation actuelle : Maître de conf. Université de Montpellier
- ▶ Spécialités : Mathématiques (géométrie)
- ▶ Email : *thierry.mignon@umontpellier.fr*
- ▶ Bureau : Bat. 9

# Enseignants: TDs/TPs

- **Pierre-Louis Montagard :**

- ▶ Situation actuelle : Maître de conf. Université de Montpellier
- ▶ Spécialités : Mathématiques (topologie algébrique)
- ▶ Email : *[pierre-louis.montagard@umontpellier.fr](mailto:pierre-louis.montagard@umontpellier.fr)*
- ▶ Bureau : Bat. 9

# Enseignants: TDs/TPs

- **Raphaël Paegelow :**

- ▶ Situation actuelle : Doctorant à l'université de Montpellier
- ▶ Spécialités : Mathématiques (Algèbre, Géométrie)
- ▶ Email : *raphael.paegelow@ens-lyon.fr*
- ▶ Bureau : Bat. 9

# Enseignants: TDs/TPs

- **Tristan Roget :**

- ▶ Situation actuelle : ATER à l'université de Montpellier
- ▶ Spécialité : Mathématiques (dynamique de populations)
- ▶ Email : *tristan.roget@gmail.com*
- ▶ Bureau : Bat. 9

# Enseignants: TDs/TPs

- **Colin Thomas :**

- ▶ Situation actuelle: Doctorant au CIRAD
- ▶ Spécialité : Ecologie (modélisation)
- ▶ Email : *colin.thomas@cirad.fr*
- ▶ Bureau : CIRAD

# Enseignants: TDs/TPs

- **Tiffany Cherchi:**

- ▶ Situation actuelle : ATER à l'université de Montpellier
- ▶ Spécialité : Probabilités
- ▶ Email : *tiffany.cherchi@umontpellier.fr*
- ▶ Bureau : Bat. 9

# Sommaire

## Informations pratiques

L'équipe pédagogique

Modalités de contrôle de connaissance (MCC)

Ressources et prérequis

Conseils numériques: pour le cours et au-delà

Python pour la science des données

# Calendrier de validation

**Note finale = 100%CC** (cf. syllabus pour le détails)

- ▶ Quiz : **25%** (**17/03/2021**) - à faire sur Moodle
- ▶ TP Noté : **35%** (du 12/04/2021 au **16/04/2021**, 23h59)
- ▶ Quiz: **40%** (**05/05/2021**) - à faire sur Moodle

Rem. : le TP noté sera mis en ligne le lundi soir, 18h, et sera à rendre pour le vendredi 23h59.

Rem. : rattrapage oral/visio

 le travail sera un rendu individuel!!! Les copies identiques verront leurs notes partagées en autant de doublons.

# Notation pour le TP noté

## Détails de la notation du TP (note sur 20) :

- ▶ Qualité des réponses aux questions: **14** pts
- ▶ Qualité de rédaction et d'orthographe: **1** pt
- ▶ Qualité des graphiques (légendes, couleurs, précision): **1** pt
- ▶ Style PEP8 valide<sup>(1)</sup> : **2** pts
- ▶ Qualité d'écriture du code (nom de variable clair, commentaires utiles, code synthétique, etc.): **1** pt
- ▶ Notebook reproductible / absence de bug (e.g., *Restart & Run all* fonctionne correctement): **1** pt

## Pénalités :

- ▶ Envoi par mail : **zéro**
- ▶ Retard : **zéro** (sauf excuse validée par l'administration)

---

<sup>(1)</sup><https://openclassrooms.com/fr/courses/4425111-perfectionnez-vous-en-python/4464230-assimilez-les-bonnes-pratiques-de-la-pep-8>

# Bonus

2 pts supplémentaires sur la note finale pour toute contribution à l'amélioration des cours (présentations, codes, widgets, etc.)

Contraintes :

- ▶ seule la première amélioration reçue est “rémunérée”
- ▶ déposer un fichier au format **.md**<sup>(2)</sup> (de taille <10 ko) en créant une fiche sur Moodle, dans la section “**Bonus - Proposition d'amélioration**”
- ▶ détailler précisément (ligne de code, page des présentations, etc.) l'amélioration proposée, ce qu'elle corrige et/ou améliore
- ▶ pour les fautes d'orthographe : proposer au minimum 5 corrections par contribution
- ▶ chaque élève ne peut gagner que 2 points maximum

---

<sup>(2)</sup><https://fr.wikipedia.org/wiki/Markdown>

# Sommaire

## Informations pratiques

L'équipe pédagogique

Modalités de contrôle de connaissance (MCC)

Ressources et prérequis

Conseils numériques: pour le cours et au-delà

Python pour la science des données

# Ressources en ligne

Site du cours: <http://josephsalmon.eu/HLMA408.html>

- ▶ Slides (MAJ au fil de l'eau)
- ▶ Widgets (sur l'aléa notamment)
- ▶ Notebooks associés (MAJ au fil de l'eau)
- ▶ Feuilles de TDs (MAJ au fil de l'eau)
- ▶ Feuilles de TPs (MAJ au fil de l'eau)
- ▶ Polycopié (Python / statistiques descriptives)

Moodle : <https://moodle.umontpellier.fr/course/view.php?id=440>

- ▶ Rendu en ligne des TPs
- ▶ Bonus
- ▶ Quiz

## Prérequis - à revoir ( $\approx$ HLMA314)

- ▶ Bases de **probabilités** : probabilité, densité, espérance, loi des grands nombres, théorème central limite  
Lecture: Foata et Fuchs (1996)
  
- ▶ Bases de l'**algèbre linéaire** : normes, produit scalaire, matrices, diagonalisation  
Lecture: Horn et Johnson (1994)

## Prérequis - à revoir ( $\approx$ HLMA314)

- ▶ Bases de **probabilités** : probabilité, densité, espérance, loi des grands nombres, théorème central limite  
Lecture: Foata et Fuchs (1996)
  
- ▶ Bases de l'**algèbre linéaire** : normes, produit scalaire, matrices, diagonalisation  
Lecture: Horn et Johnson (1994)

# Objectifs de la partie modélisation du cours

Objectifs : utilisation des probabilités et des statistiques pour le traitement et la visualisation de données

- ▶ méthodes basiques de statistiques descriptives
- ▶ aspects élémentaires de la quantification de l'aléatoire
- ▶ introduction aux tests et aux intervalles de confiance
- ▶ modèles linéaires univariés
- ▶ Analyse de variance, ANOVA ( : *Analysis of variance*)

# Objectif de la partie numérique du cours

Objectifs : utilisation de Python pour le traitement et la visualisation de données

- ▶ méthodes basiques de programmation et d'algorithmique
- ▶ bibliothèques de méthodes numériques (`numpy`, `scipy`)
- ▶ bibliothèques pour le traitement des bases de données (`pandas`)
- ▶ bibliothèques pour la visualisation (`matplotlib`, `seaborn`)
- ▶ introduire des bonnes pratiques numériques valables pour tous les langages (lisibilité, documentation)

# Sommaire

Informations pratiques

Conseils numériques: pour le cours et au-delà

Python pour la science des données

- ▶ Tutoriel de connexion aux machines de l'UM (x2go):

<https://moodle.umontpellier.fr/course/view.php?id=8975>

- ▶ Une fois connecté utiliser anaconda3 pour lancer un jupyter-notebook; détails en section 2.3.1 du poly<sup>(3)</sup>

Rem. : premier TP principalement sur la prise en main, mais à préparer en amont, avec les notebooks fournis sur le site du cours

Rem. : alternative possible pour utiliser des notebooks "distants"  
colab <https://colab.research.google.com/>

---

<sup>(3)</sup><http://josephsalmon.eu/enseignement/Montpellier/HLMA310/IntroPython.pdf>

# Python sur votre machine personnelle

Conseil: Privilégier **Conda** / **Anaconda** / **mini Conda** (tous OS)



: pas d'aide des enseignants sur ce point; entraidez-vous!

Format de rendu des TPs: `jupyter notebook`, extension **.ipynb**

Rem. : en TP, choisissez l'environnement que vous préférez tant qu'il est fonctionnel (packages, versions, etc.)

## Conseils généraux pour l'année

- ▶ Adoptez des règles d'écriture de code et tenez-vous y!  
**Exemple** : **PEP8** pour Python (utiliser **AutoPEP8**, ou pour les notebooks [jupyter-autopep8](#) sur votre propre machine)
- ▶ Utilisez **Markdown** (.md) pour les parties rédigées du .ipynb

"A (wo)man must have a code."  
- Bunk

Source : [The Wire](#), épisode 7, saison 1.

- ▶ Apprenez de bons exemples (ouvrez les codes sources!):  
<http://jakevdp.github.io/>,  
<https://www.statsmodels.org/stable/index.html>, etc.

# Éditeurs de texte (pas seulement pour Python)

jupyter notebooks : excellents pour des courts projets (TPs)

Pour la suite de votre carrière:

IPython + éditeur de texte avancé: projets / codes longs

Éditeurs recommandés :

- ▶ **Visual Studio Code**
- ▶ Atom
- ▶ Sublime Text
- ▶ vim (fort coût d'entrée, déconseillé)
- ▶ emacs (fort coût d'entrée, déconseillé)

Bénéfices : coloration syntaxique, auto-complétion du code, débogueur graphique, warning PEP8, etc.

# Environnement intégré (optionnel)

**VSCode** : <https://code.visualstudio.com/>

- ▶ coloration syntaxique
- ▶ auto-complétion
- ▶ vérification de code dans l'environnement
- ▶ débogueur graphique
- ▶ intégration avec gestionnaires de versions (git)
- ▶ gestion des environnements virtuels (VirtualEnv)
- ▶ gestion des tests, etc.

# Sommaire

Informations pratiques

Conseils numériques: pour le cours et au-delà

**Python pour la science des données**

Choix de Python: introduction et motivation

Écosystème Python: les librairies

# Sommaire

Informations pratiques

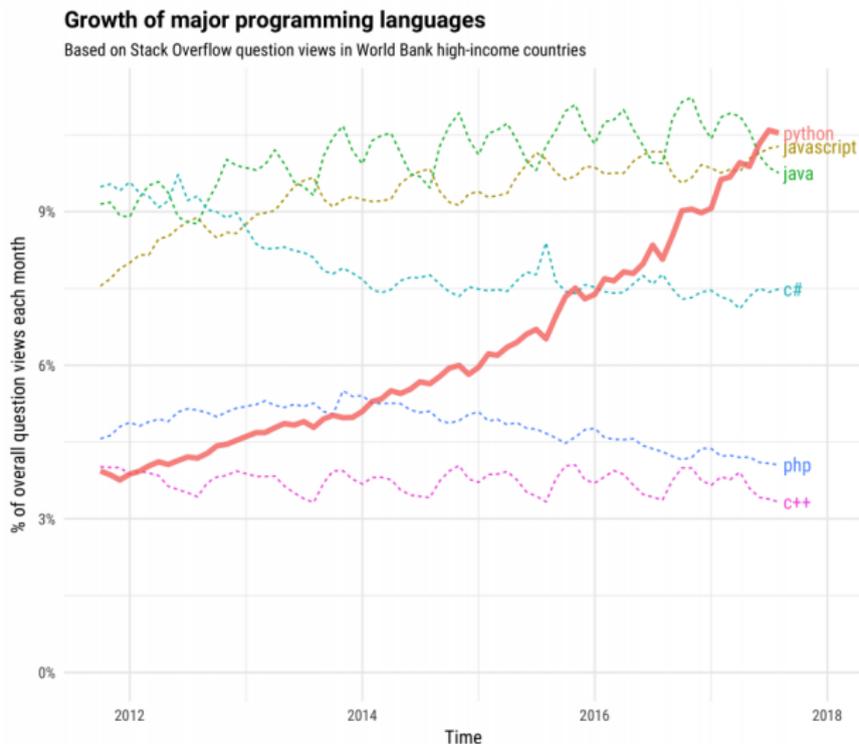
Conseils numériques: pour le cours et au-delà

**Python pour la science des données**

Choix de Python: introduction et motivation

Écosystème Python: les librairies

# Popularité de Python sur Stackoverflow

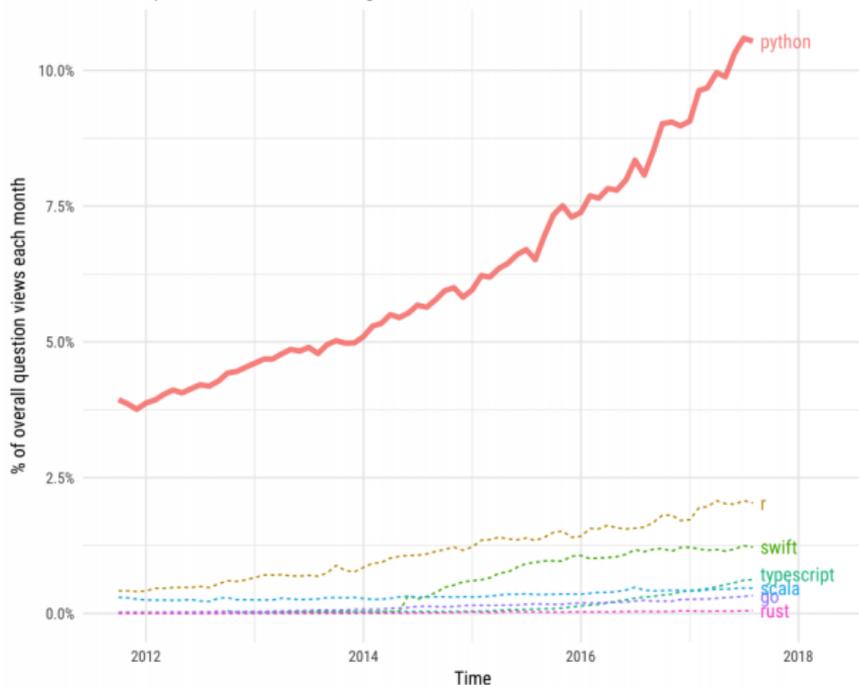


Source : <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

# Popularité de Python sur Stackoverflow

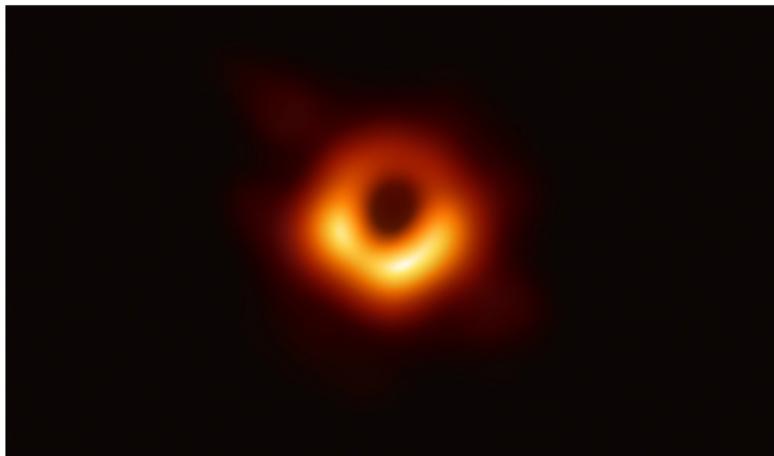
## Python compared to smaller, growing technologies

Based on question traffic in World Bank high-income countries



Source : <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

# Python et médias: 1<sup>re</sup> image d'un trou noir (EHT Collaboration, 2019)

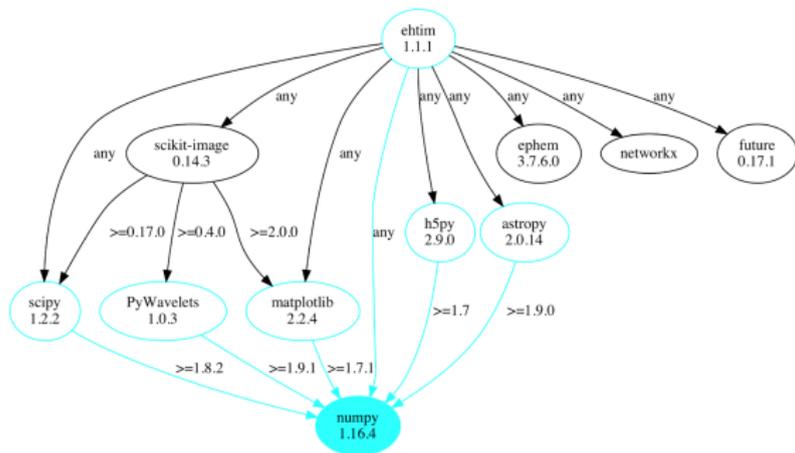


Trou Noir M87

Sources : Event Horizon Telescope Collaboration (2019)

▶ <https://numpy.org/case-studies/blackhole-image/>

# Python et médias: 1<sup>re</sup> image d'un trou noir (EHT Collaboration, 2019)

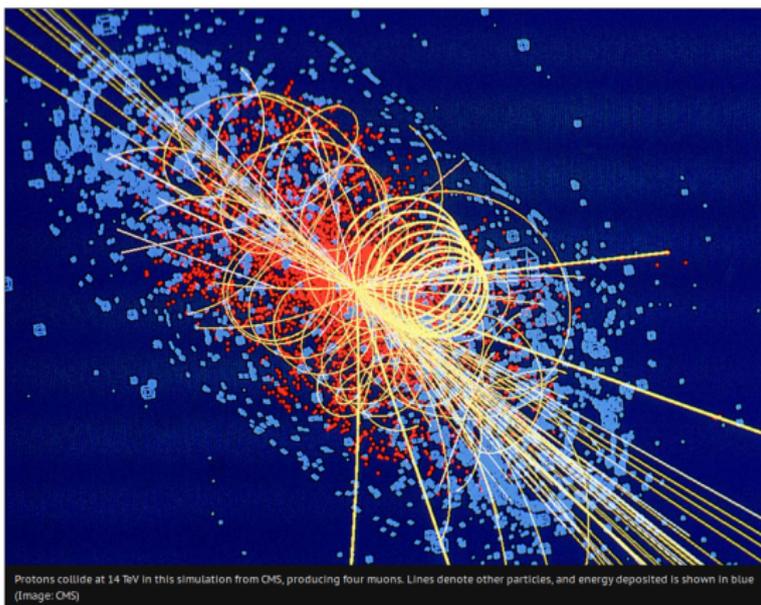


Dépendance logicielle des packages

Sources : Event Horizon Telescope Collaboration (2019)

► <https://numpy.org/case-studies/blackhole-image/>

# Python et médias: découverte du Boson de Higgs (CERN, 2012)

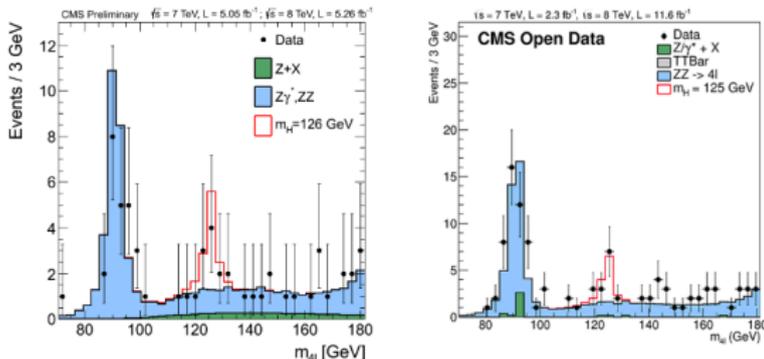


Collisions

## Sources :

- ▶ <https://home.cern/fr/science/physics/higgs-boson>
- ▶ <https://cms.cern/news/observing-higgs-over-one-petabyte-new-cms-open-data>

# Python et médias: découverte du Boson de Higgs (CERN, 2012)



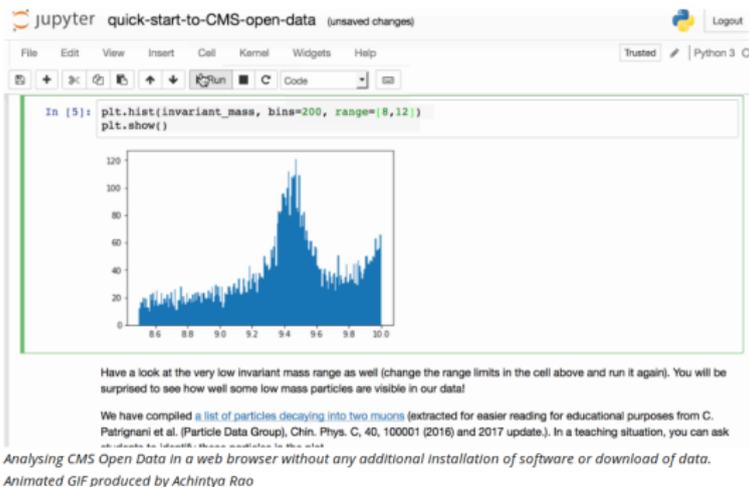
Left: The official CMS plot for the “Higgs to four leptons” channel, shown on the day of the Higgs discovery announcement. Right: A similar plot produced by Nur Zulaiha Johari et al. using CMS Open Data from 2011 and 2012. Although the plots appear similar, the analysis with CMS Open Data uses more data (at 8 TeV and overall) than the official CMS one from the original discovery but is a lot less sophisticated and is not scrutinised by the wider CMS community of experts.

Matplotlib (pour la visualisation)

Sources :

- ▶ <https://home.cern/fr/science/physics/higgs-boson>
- ▶ <https://cms.cern/news/observing-higgs-over-one-petabyte-new-cms-open-data>

# Python et médias: découverte du Boson de Higgs (CERN, 2012)



Jupyter notebook (pour la présentation)

## Sources :

- ▶ <https://home.cern/fr/science/physics/higgs-boson>
- ▶ <https://cms.cern/news/observing-higgs-over-one-petabyte-new-cms-open-data>

# Python dans l'académique: aspect enseignement

## **Python est au programme des classes préparatoires aux grandes écoles (depuis 2013)**

“Depuis la réforme des programmes de 2013, l'informatique est présente dans les programmes de CPGE à deux niveaux. Un tronc commun à chacune des trois filières MP, PC et PSI se donne pour objectif d'apporter aux étudiants la maîtrise d'un certain nombre de concepts de base : conception d'algorithmes, choix de représentations appropriées des données, etc. à travers l'apprentissage du langage Python.”

Source : <https://info-llg.fr/>

# Python dans l'académique: aspect recherche

Exemple d'une package populaire d'apprentissage automatique  
( : *machine learning*) : scikit-learn

## Scikit-learn: Machine learning in Python

[F Pedregosa](#), [G Varoquaux](#), [A Gramfort](#)... - [Journal of machine ...](#), 2011 - [jmlr.org](#)

**Scikit-learn** is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level ...

☆ 99 **Cité 11668 fois** [Autres articles](#) [Les 31 versions](#) 

Source : Google Scholar (8/09/2018)

- ▶ ≈ 1 000 pages de documentation
- ▶ ≈ 500 000 utilisateurs les 30 derniers jours (fin 2017)
- ▶ ≈ 42 000 000 pages vues sur le site (2017)

Source : Alexandre Gramfort (INRIA - Parietal)

# Python dans l'académique: aspect recherche

Exemple d'une package populaire d'apprentissage automatique  
( : *machine learning*) : scikit-learn

## Scikit-learn: Machine learning in Python

[F Pedregosa](#), [G Varoquaux](#), [A Gramfort](#)... - [Journal of machine ...](#), 2011 - [jmlr.org](#)

**Scikit-learn** is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level ...

☆ 99 **Cité 11668 fois** [Autres articles](#) [Les 31 versions](#) 

Source : Google Scholar (8/09/2018)

- ▶  $\approx$  1 000 pages de documentation
- ▶  $\approx$  500 000 utilisateurs les 30 derniers jours (fin 2017)
- ▶  $\approx$  42 000 000 pages vues sur le site (2017)

Source : Alexandre Gramfort (INRIA - Parietal)

# Explication du succès de Python

Proverbe (récent):

 : Python; *the second best language for everything!*

 : Python; *le deuxième meilleur langage pour tout!*

Autres bénéfices de Python:

- ▶ langage compact (5X plus compact que Java ou C++)
- ▶ ne requiert pas d'étape — potentiellement longue — de compilation (comme le C)  $\implies$  débogage plus facile
- ▶ portabilité sur les systèmes d'exploitation courants (Linux, MacOS, Windows)

En résumé : Python = excellent “couteau suisse” numérique

## Les limites (car il en existe!)

- ▶ vitesse d'exécution souvent inférieure vs. C/C++, Fortran (langage compilé de plus bas niveau)
- ▶ langage permissif, un programme peut s'exécuter malgré des "erreurs" non dépistées; **vigilance donc!**
- ▶ historiquement les statisticiens utilis(ai)ent R ... (cela évolue!)

# Sommaire

Informations pratiques

Conseils numériques: pour le cours et au-delà

**Python pour la science des données**

Choix de Python: introduction et motivation

Écosystème Python: les librairies

# Popularisation récente: richesse des bibliothèques “libres” ( : *open source*)<sup>(4)</sup>

Apprentissage automatique ( : *Machine learning*) :

- ▶ `sklearn` (2010)
- ▶ `tensorflow` (2015)

Traitement du langage ( : *Natural Language Processing*) :

- ▶ `nltk` (2001)

---

<sup>(4)</sup> ce n'est pas le cas de matlab par exemple; dont le coût = plusieurs dizaines de milliers d'euros pour l'université

# Popularisation récente: richesse des bibliothèques “libres” ( : *open source*)<sup>(4)</sup>

Apprentissage automatique ( : *Machine learning*) :

- ▶ sklearn (2010)
- ▶ tensorflow (2015)

Traitement du langage ( : *Natural Language Processing*) :

- ▶ nltk (2001)

Traitement des images ( : *image processing*) :

- ▶ skimage (2009)

---

<sup>(4)</sup> ce n'est pas le cas de matlab par exemple; dont le coût = plusieurs dizaines de milliers d'euros pour l'université

# Popularisation récente: richesse des librairies “libres” ( : *open source*)<sup>(4)</sup>

Apprentissage automatique ( : *Machine learning*) :

- ▶ sklearn (2010)
- ▶ tensorflow (2015)

Traitement du langage ( : *Natural Language Processing*) :

- ▶ nltk (2001)

Traitement des images ( : *image processing*) :

- ▶ skimage (2009)

Développement web :

- ▶ django (2005)
- ▶ etc.

---

<sup>(4)</sup> ce n'est pas le cas de matlab par exemple; dont le coût = plusieurs dizaines de milliers d'euros pour l'université

# Popularisation récente: richesse des bibliothèques “libres” ( : *open source*)<sup>(4)</sup>

Apprentissage automatique ( : *Machine learning*) :

- ▶ sklearn (2010)
- ▶ tensorflow (2015)

Traitement du langage ( : *Natural Language Processing*) :

- ▶ nltk (2001)

Traitement des images ( : *image processing*) :

- ▶ skimage (2009)

Développement web :

- ▶ django (2005)
- ▶ etc.

---

<sup>(4)</sup> ce n'est pas le cas de matlab par exemple; dont le coût = plusieurs dizaines de milliers d'euros pour l'université

# Librairies indispensables en Python (pour ce cours)

## ▶ Numpy

[https://github.com/agramfort/liesse\\_telecom\\_paristech\\_python/blob/master/2-Numpy.ipynb](https://github.com/agramfort/liesse_telecom_paristech_python/blob/master/2-Numpy.ipynb)

<https://www.labri.fr/perso/nrougier/from-python-to-numpy/index.html>

## ▶ Scipy :

[https://github.com/agramfort/liesse\\_telecom\\_paristech\\_python/blob/master/3-Scipy.ipynb](https://github.com/agramfort/liesse_telecom_paristech_python/blob/master/3-Scipy.ipynb)

## ▶ Matplotlib :

<https://www.labri.fr/perso/nrougier/teaching/matplotlib/matplotlib.html>

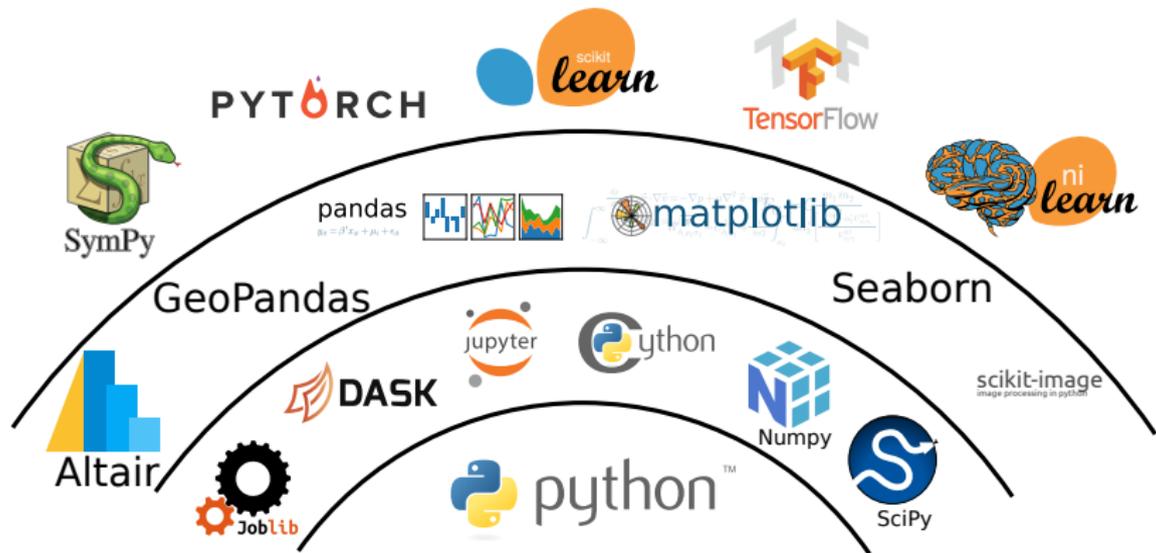
<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003833>

## ▶ Pandas : <https://github.com/jorisvandenbossche/pandas-tutorial>

## ▶ Statsmodels : <http://www.statsmodels.org/stable/index.html>

Tutos de Jake Vanderplas: [Reproducible Data Analysis in Jupyter](#)

# Écosystème Python: Panorama partiel et partiel



## Livres et ressources en ligne complémentaires

Statistiques Holmes et Huber, Modern statistics for modern biology (2018)

Statistiques Nolan et Speed, Stat labs: mathematical statistics through applications (2001)

Général / Science des données: Guttag, Introduction to Computation and Programming (2016)

Science des données: J. Van DerPlas, With Application to Understanding Data (2016), Statistical Rethinking: A Bayesian Course with Examples in R and Stan R. McElreath (2015)

Python: <http://www.scipy-lectures.org/>

Visualisation (sous R): <https://serialmentor.com/dataviz/>

Rem. : plus de références dans le syllabus

# Biographie du jour : Aaron Swartz<sup>(5)</sup>



- ▶ Hactiviste américain (1986-2013)
- ▶ Créateur du format de fichier Markdown (.md)
- ▶ Mise au point des licences *Creative Commons* (CC)
- ▶ ...

Pour aller plus loin: Documentaire de Brian Knappenberger, *The Internet's Own Boy: The Story of Aaron Swartz* (2014)

---

<sup>(5)</sup>[https://fr.wikipedia.org/wiki/Aaron\\_Swartz](https://fr.wikipedia.org/wiki/Aaron_Swartz)

# Bibliographie I

- ▶ Foata, D. and A. Fuchs. *Calcul des probabilités: cours et exercices corrigés*. Masson, 1996.
- ▶ Guttag, J. V. *Introduction to Computation and Programming Using Python: With Application to Understanding Data*. MIT Press, 2016.
- ▶ Holmes, S. and W. Huber. *Modern statistics for modern biology*. Cambridge University Press, 2018.
- ▶ Horn, R. A. and C. R. Johnson. *Topics in matrix analysis*. Corrected reprint of the 1991 original. Cambridge: Cambridge University Press, 1994, pp. viii+607.
- ▶ McElreath, R. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Chapman & Hall/CRC Texts in Statistical Science. Chapman and Hall/CRC, 2015.
- ▶ Nolan, D. and T. P. Speed. *Stat labs: mathematical statistics through applications*. Springer Science & Business Media, 2001.

# Bibliographie II

- ▶ VanderPlas, J. *Python Data Science Handbook*. O'Reilly Media, 2016.