

# HLMA408: Traitement des données

Statistiques descriptives

**Joseph Salmon**

<http://josephsalmon.eu>

Université de Montpellier



# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

Introduction: Grossesse, cigarettes et nouveaux nés

Éléments de cours

# 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

- **Emmanuel Bonnet:**

- ▶ Situation actuelle: Doctorant à l'université de Montpellier
- ▶ Spécialités: Bio-statistiques
- ▶ Email: *emmanuel.bonnet@umontpellier.fr*
- ▶ Bureau : Institut Universitaire de Recherche Clinique (IURC)

# Enseignants: TDs/TPs

- **Pierre-Louis Montagard:**

- ▶ Situation actuelle: Maître de conf. à l'université de Montpellier
- ▶ Spécialités: Mathématiques (topologie algébrique)
- ▶ Email: *pierre-louis.montagard@umontpellier.fr*
- ▶ Bureau: 421, Bat. 9

# Enseignants: TDs/TPs

- **Thi-Thuy Bui:**

- ▶ Situation actuelle: Doctorante à l'université de Montpellier
- ▶ Spécialités: Probabilité
- ▶ Email: *thi-thuy.bui@umontpellier.fr*
- ▶ Bureau: TBD, Bat. 9

# Enseignants: TDs/TPs

- **Florent Bascou:**

- ▶ Situation actuelle: Doctorant à l'université de Montpellier
- ▶ Spécialités: Statistiques
- ▶ Email: *[florent.bascou@umontpellier.fr](mailto:florent.bascou@umontpellier.fr)*
- ▶ Bureau: 128, Bat. 9

# Ressources en ligne

Informations principales: site du cours

<http://josephsalmon.eu/HLMA408.html>

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

<http://josephsalmon.eu/enseignement/Montpellier/HLMA310/IntroPython.pdf>

Rendu en ligne des TPs: Moodle


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




# Calendrier de validation

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

- ▶ TP1: 10% (25/02/2020) - à rendre sur Moodle le jour même
- ▶ TP2: 20% (10/03/2020) - à rendre sur Moodle le jour même
- ▶ TP3: 30% (7/04/2020) - à rendre sur Moodle le jour même
- ▶ Quiz: 40% (29/04/2020)

 : les TPs seront mis en ligne la veille (les lundis soirs, 18h), et seront à rendre pour le mardi même 18h.

 : pas de rattrapage

# Notation pour les TPs notés

## Détails de la notation des TPs (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, etc.)

## Contraintes:

- ▶ seule la première amélioration reçue est "rémunérée"
- ▶ déposer un fichier d'extension **.txt** (taille <10 ko) en créant une fiche sur Moodle, 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

# Prérequis - à revoir seul

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

# Prérequis - à revoir seul

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

# Description 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
- ▶ librairies de méthodes numériques (`numpy`, `scipy`)
- ▶ librairies pour le traitement des bases de données (`pandas`)
- ▶ librairies pour la visualisation (`matplotlib`, `seaborn`)

# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

Introduction: Grossesse, cigarettes et nouveaux nés

Éléments de cours

# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

Introduction: Grossesse, cigarettes et nouveaux nés

Éléments de cours



# Aspects algorithmiques: quelques conseils

Python: installation fonctionnelle sur les machines de l'université,  
Utiliser anaconda3 et anaconda-navigator.

Informations détaillées sur le polycopié disponible ici:

<http://josephsalmon.eu/enseignement/Montpellier/HLMA310/IntroPython.pdf>

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

# Installation personnelle

Conseil d'installation de Python sur machines personnelles:  
Privilégier **Conda** / **Anaconda** / **mini Conda** (tous OS)



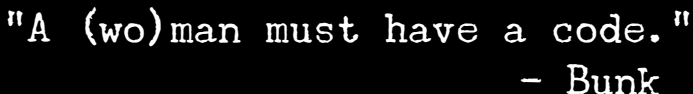
: pas d'aide des enseignants sur ce point; entraidez-vous!

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

Rem: en TP, amener votre ordinateur si vous préférez garder votre environnement (packages, versions, etc.)

## Conseils généraux pour l'année

- ▶ Adoptez des règles d'écriture de code et tenez-y vous!  
**Exemple: PEP8** pour Python (utiliser **AutoPEP8**, ou pour les notebooks <https://github.com/kenko000/jupyter-autopep8>)
- ▶ Utilisez **Markdown** (.md) pour les parties rédigées / comptes-rendus (e.g., markdown-preview-plus avec **Atom**)



"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.

TODO: ouvrir le fichier ".ipynb" associé (jupyter notebook)

# Éditeurs de texte (pas seulement pour Python)

jupyter notebooks: excellents pour des courts projets (TPs)

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.

# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

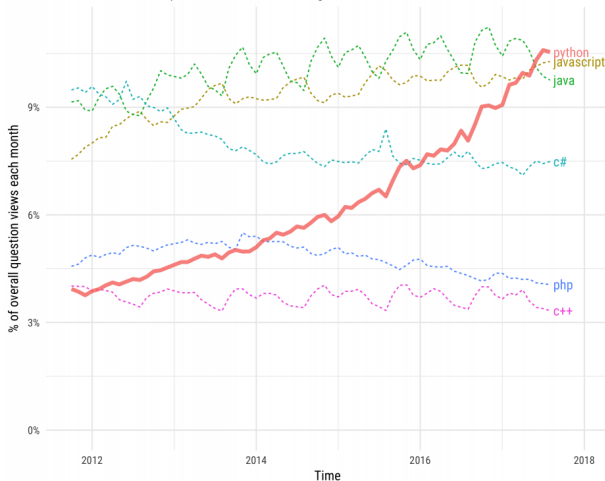
Introduction: Grossesse, cigarettes et nouveaux nés

Éléments de cours

# Popularité de Python sur Stackoverflow

## Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries

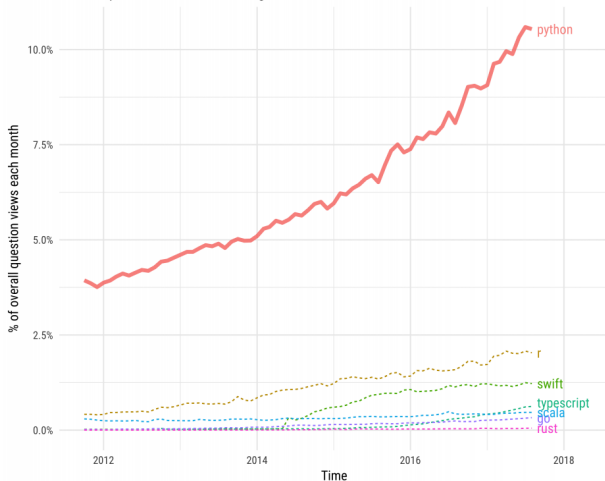


Source: <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>

# Popularité de Python sur Stackoverflow

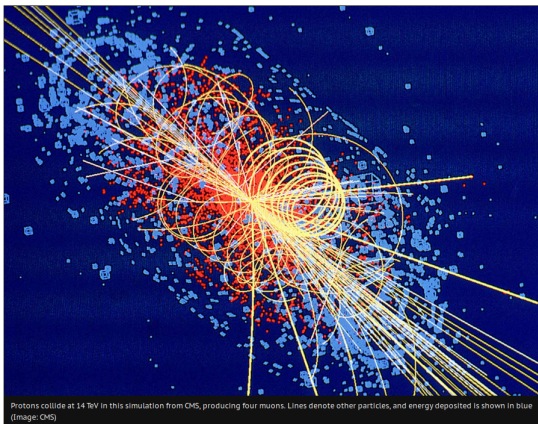
## Python compared to smaller, growing technologies

Based on question traffic in World Bank high-income countries



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

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



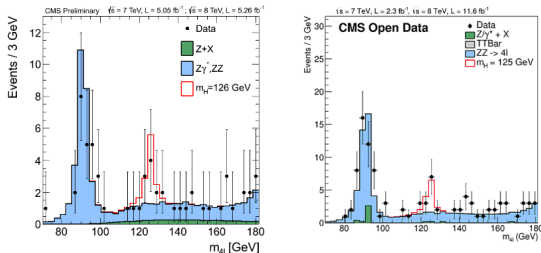
## Collisions

### Sources:

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



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



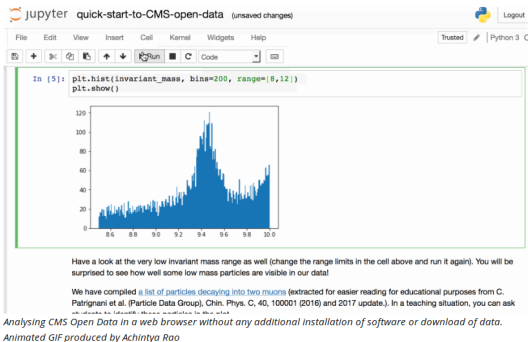
Left: The official CMS plot for the "Higgs to four leptons" channel, shown on the day of the Higgs discovery announcement. Right: A similar plot produced by Nur Zulaiha Jomhari et al. using CMS Open Data from 2011 and 2012. Although the plots appear similar, the analysis with CMS Open Data uses more 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 dans les 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 ...


☆ 77 **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)

# 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

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

**Écosystème Python: les librairies**

Introduction: Grossesse, cigarettes et nouveaux nés

Éléments de cours



# Popularisation récente: richesse des librairies “libres” ( : *open source*)<sup>(2)</sup>

Apprentissage automatique ( : *Machine learning*):

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

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

- ▶ nltk (2001)


---

<sup>(2)</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>(2)</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>(2)</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>(2)</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>(2)</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>(2)</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>(2)</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>

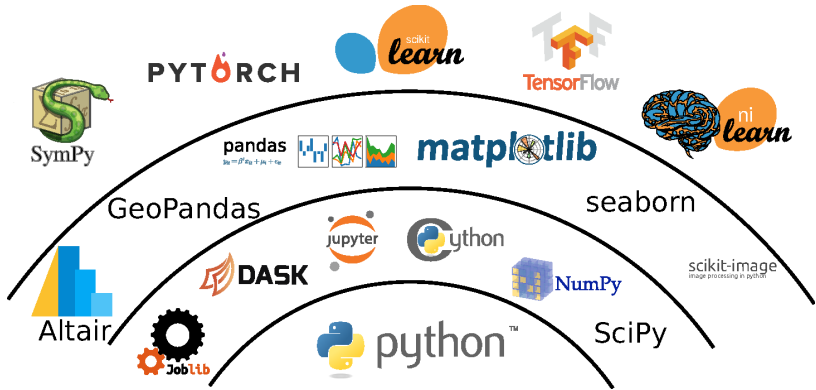
<https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003833>

## ► 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

# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

Introduction: Grossesse, cigarettes et nouveaux nés

Éléments de cours



# Objectifs

- ▶ Sur un jeu de données, comparer les poids à la naissance suivant le statut tabagique de la mère
- ▶ Revoir rapidement quelques outils de statistique descriptive
- ▶ Qu'est-ce qu'une loi normale ?

## Présentation des données babies23<sup>(3)</sup>

Poids Naissance	Statut Tabagique	...
120	0	...
113	0	...
128	1	...
123	0	...
108	1	...
136	0	...
138	0	...
132	0	...
⋮	⋮	...

- ▶ Poids (onces):  $1 \text{ g} \approx 0.035 \text{ once}$
- ▶ Statut :  
1, mère fumeuse  
0, mère non fumeuse
- ▶ Tableau entier :  
 $n = 1236$  observations
- ▶ Étude à l'œil nu impossible  
⇒ résumer les données par
  - ▶ quelques valeurs numériques
  - ▶ des graphiques parlants

---

<sup>(3)</sup><http://www.stat.berkeley.edu/users/statlabs/data/babies23.data>

# Table de fréquences croisées

Étude sur données complètes<sup>(4)</sup>: le taux de mortalité infantile chez les enfants nés de mères fumeuses est plus faible<sup>(5)</sup>:

**Table:** Taux de mortalité infantile en fonction du poids (g) à la naissance différencié selon le statut tabagique de la mère

Classe de poids	Non fumeur	Fumeur
< 1500	792 ‰	565 ‰
1500–2000	406 ‰	346 ‰
2000–2500	78 ‰	27 ‰
2500–3000	11.6 ‰	6.1 ‰
3000–3500	2.2 ‰	4.5 ‰
≥ 3500	3.8 ‰	2.6 ‰

## ► Des critiques / commentaires sur le tableau?

---

<sup>(4)</sup>ici on n'a qu'une sous-partie de l'ensemble des données

<sup>(5)</sup>D. Nolan and T. P. Speed. *Stat labs: mathematical statistics through applications*. Springer Science & Business Media, 2001.

## Corrigeons l'erreur ...

- ▶ Une autre étude préconise de travailler sur les poids à la naissance, après **standardisation**

$$\text{poids standardisé de l'obs} = \frac{\text{poids de l'obs} - \text{moyenne de toutes les obs}}{\text{écart-type de toutes les obs}}$$

- ▶ Cette standardisation est faite séparément pour les deux classes: pour les fumeurs et pour les non fumeurs
- ▶ Intérêt: comparer ce qui est comparable ! Ici les bébés de mères fumeuses ont (en général) un poids plus faible

## Corrigeons l'erreur ...

- Une autre étude préconise de travailler sur les poids à la naissance, après **standardisation**

$$\text{poids standardisé de l'obs} = \frac{\text{poids de l'obs} - \text{moyenne de toutes les obs}}{\text{écart-type de toutes les obs}}$$

- Cette standardisation est faite séparément pour les deux classes: pour les fumeurs et pour les non fumeurs
- Intérêt: comparer ce qui est comparable ! Ici les bébés de mères fumeuses ont (en général) un poids plus faible
- Ainsi on comparera le taux de mortalité d'un bébé pesant 2680g (fumeur) à celui pesant 3000g (non fumeur)

## Corrigeons l'erreur ...

- ▶ Une autre étude préconise de travailler sur les poids à la naissance, après **standardisation**

$$\text{poids standardisé de l'obs} = \frac{\text{poids de l'obs} - \text{moyenne de toutes les obs}}{\text{écart-type de toutes les obs}}$$

- ▶ Cette standardisation est faite séparément pour les deux classes: pour les fumeurs et pour les non fumeurs
- ▶ Intérêt: comparer ce qui est comparable ! Ici les bébés de mères fumeuses ont (en général) un poids plus faible
- ▶ Ainsi on comparera le taux de mortalité d'un bébé pesant 2680g (fumeur) à celui pesant 3000g (non fumeur)

## Effets "cachés"<sup>(6)</sup>

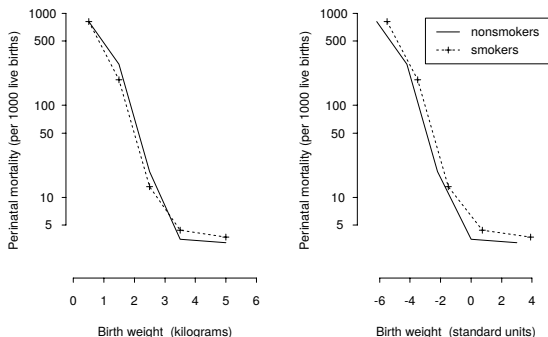


FIGURE 1.2. Mortality curves for smokers and nonsmokers by kilograms (left plot) and by standard units (right plot) of birth weight for the Missouri study (Wilcox [Wil93]).

- Il semblerait maintenant que les bébés de mères fumeuses aient un taux de mortalité plus élevé
- Faites attention aux effets cachés (**variables confondantes**)!

---

<sup>(6)</sup>D. Nolan and T. P. Speed. *Stat labs: mathematical statistics through applications*. Springer Science & Business Media, 2001.

# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

Introduction: Grossesse, cigarettes et nouveaux nés

## Éléments de cours

- Histogrammes et densités

- Résumés numériques

- Boîtes à moustache (Boxplot) et violons



# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

Introduction: Grossesse, cigarettes et nouveaux nés

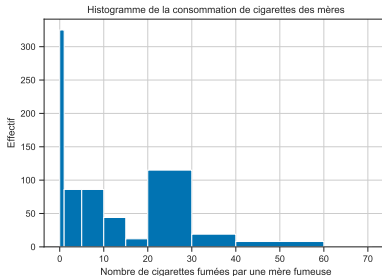
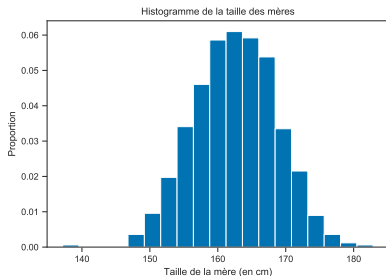
## Éléments de cours

- Histogrammes et densités

- Résumés numériques


- Boîtes à moustache (Boxplot) et violons

# Histogrammes



(Gauche) histogramme de la taille (cm) pour les mères (proportion)  
(Droite) histogramme du nombre de cigarettes fumées par jour  
pour les mères fumeuses (effectif)

# Qu'est-ce qu'un histogramme?

- ▶ Histogramme  $\neq$  diagramme en barre ( : *barplot*) !!!
- ▶ Décrit / estime la **distribution** : unimodalité, symétrie, étendue, ...
- ▶ Construction :
  - ▶ Axe horizontal : gradué (échelle des valeurs observées)
  - ▶ Axe vertical : **DENSITÉ!!!**  
de fréquence ou d'effectif

densité de fréquence de la classe  $k = \frac{\text{fréquence de la classe } k}{\text{longueur de la classe } k}$

densité d'effectif de la classe  $k = \frac{\text{effectif de la classe } k}{\text{longueur de la classe } k}$

- ▶ Attention à l'unité sur l'axe vertical (e.g., l'option `density=True/False` de `hist` en Matplotlib)

# Exemple de construction d'histogramme<sup>(7)</sup>

Nombre de cigarettes  
par jour pour les mères  
fumeuses:

Nb de cig.	% de fumeurs
0	46.76
1-5	12.37
5-10	12.37
10-15	6.33
15-20	1.72
20-30	16.55
30-40	2.73
40-60	1.15
60-	0.00
Total	100

- Problème de bords: le 5 appartient à quelle classe? à la deuxième!

Rem: toujours regarder l'aide pour savoir si `hist` est ouvert à droite ou à gauche (généralement :  $[a,b[$  )

- Hauteur du rectangle (cas densité):

$$h_0 = \frac{46.76}{1 \times 100} = 0.4676,$$

$$h_1 = \frac{12.37}{4 \times 100} = 0.0309,$$

$$\vdots = \vdots$$

$$h_{40} = \frac{1.15}{20 \times 100} = 0.00057,$$

---

<sup>(7)</sup>[https://matplotlib.org/api/\\_as\\_gen/matplotlib.pyplot.hist.html](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.hist.html)

## Exemple de construction d'histogramme<sup>(7)</sup>

Nombre de cigarettes  
par jour pour les mères  
fumeuses:

Nb de cig.	% de fumeurs
0	46.76
1-5	12.37
5-10	12.37
10-15	6.33
15-20	1.72
20-30	16.55
30-40	2.73
40-60	1.15
60-	0.00
Total	100

- Problème de bords: le 5 appartient à quelle classe? à la deuxième!

Rem: toujours regarder l'aide pour savoir si `hist` est ouvert à droite ou à gauche (généralement :  $[a,b[$  )

- Hauteur du rectangle (cas densité):

$$h_0 = \frac{46.76}{1 \times 100} = 0.4676,$$

$$h_1 = \frac{12.37}{4 \times 100} = 0.0309,$$

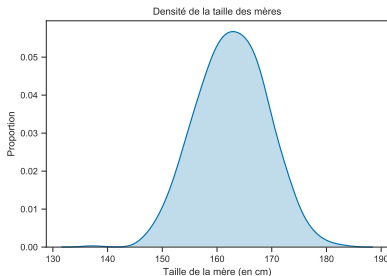
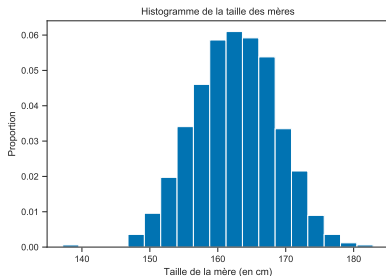
$$\vdots = \vdots$$

$$h_{40} = \frac{1.15}{20 \times 100} = 0.00057,$$


---

<sup>(7)</sup>[https://matplotlib.org/api/\\_as\\_gen/matplotlib.pyplot.hist.html](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.hist.html)

# Autres estimateurs de densité



(Gauche) histogramme — (droite) densité des tailles des mères

Rem: on parle d'estimateur à noyau<sup>(8),(9)</sup> de la densité  
( : *Kernel Density Estimator, KDE*)

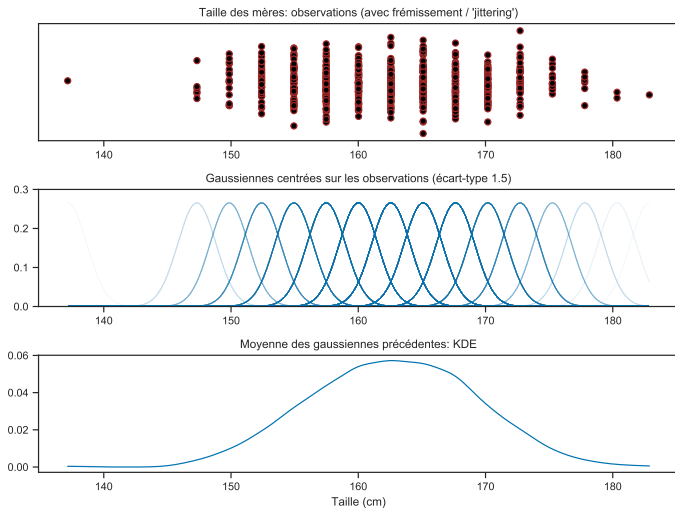
Rem: les deux figures bleues ont une aire égale à 1

---

<sup>(8)</sup> M. Rosenblatt. "Remarks on some nonparametric estimates of a density function". In: *Ann. Math. Statist.* 27 (1956), pp. 832–837.

<sup>(9)</sup> E. Parzen. "On estimation of a probability density function and mode". In: *Ann. Math. Statist.* 33 (1962), pp. 1065–1076.

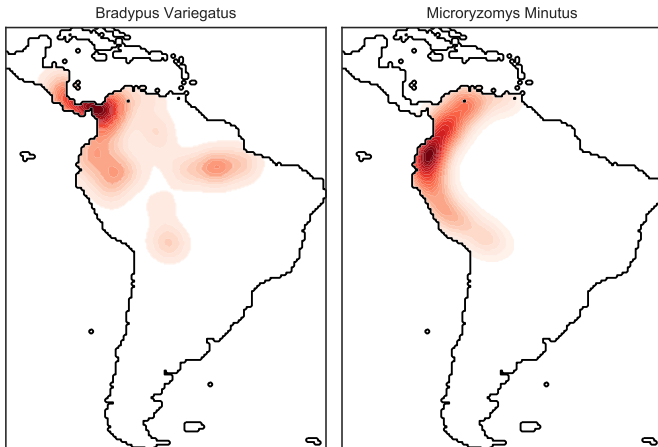
# Estimateur à noyau de la densité: aperçu



# Densité spatiale

- Quantité numérique grande lorsqu'il y a beaucoup d'observations dans une région de l'espace et petite sinon

**Exemple:** densité de population pour des espèces animales<sup>(10)</sup>



<sup>(10)</sup> [https://scikit-learn.org/stable/auto\\_examples/neighbors/plot\\_species\\_kde.html](https://scikit-learn.org/stable/auto_examples/neighbors/plot_species_kde.html)



# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

Introduction: Grossesse, cigarettes et nouveaux nés

## Éléments de cours

Histogrammes et densités

Résumés numériques

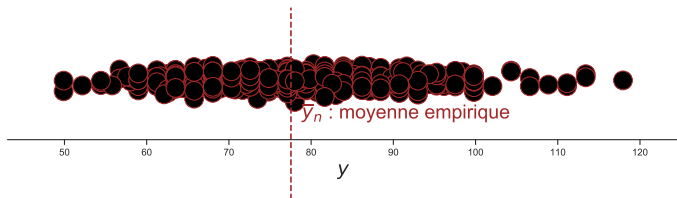
Boîtes à moustache (Boxplot) et violons

## Données étudiées

Pour les prochaines visualisation on utilise la taille des pères (en cm), tirées de la base de données babies23.data obtenues par:

```
df_babies = pd.read_csv("babies23.data",  
                        skiprows=38)  
df_babies['dht'] # dht stands for ``dads height''
```

# Moyenne (arithmétique)

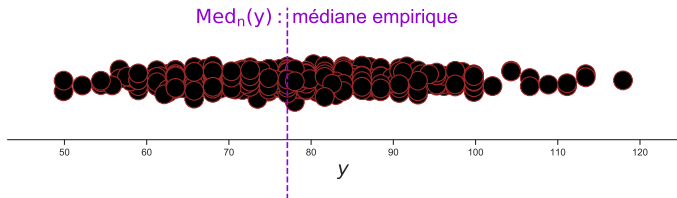


**Définition: Moyenne (arithmétique)**

$$\bar{y}_n = \frac{1}{n} \sum_{i=1}^n y_i$$

Notation:  $\mathbf{y} = (y_1, \dots, y_n)^\top$ , où le symbole  $\mathbf{y}^\top$  représente le transposé du vecteur  $\mathbf{y}$  (par convention on représente les vecteurs comme des colonnes :  $\mathbf{y} \in \mathbb{R}^n \iff \mathbf{y} \in \mathbb{R}^{n \times 1}$  )

# Médiane



On ordonne les  $y_i$  dans l'ordre croissant :  $y_{(1)} \leq y_{(2)} \leq \dots \leq y_{(n)}$

---

---

**Définition: Médiane**

---

---

$$\text{Med}_n(\mathbf{y}) = \begin{cases} \frac{y_{(\frac{n}{2})} + y_{(\frac{n}{2}+1)}}{2}, & \text{si } n \text{ est pair} \\ y_{(\frac{n+1}{2})}, & \text{si } n \text{ est impair} \end{cases}$$

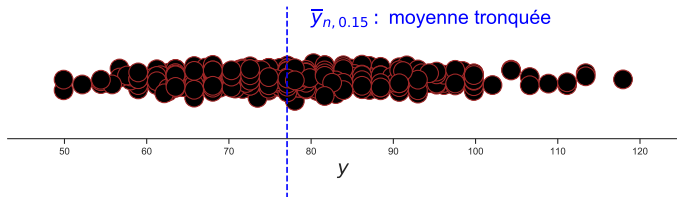
---

---

Rem: utile pour décrire le niveau de revenus dans une population

Rem: définition ambiguë : non unicité (idem pour les quantiles)

# Moyenne tronquée



Pour un paramètre  $\alpha$  (e.g.,  $\alpha = 15\%$ ), on calcule la moyenne en enlevant les  $\alpha\%$  plus grandes et plus petites valeurs

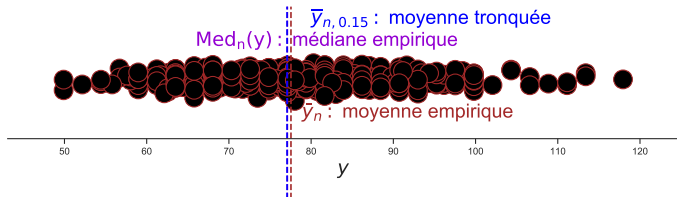
**Définition: Moyenne tronquée (à l'ordre  $\alpha$ )**


$$\bar{y}_{n,\alpha} = \bar{z}_n$$

où  $\mathbf{z} = (y_{(\lfloor \alpha n \rfloor)}, \dots, y_{(\lfloor (1-\alpha)n \rfloor)})$  est l'échantillon  $\alpha$ -tronqué

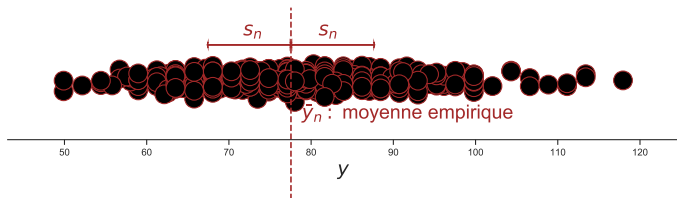
Rem:  $\lfloor u \rfloor$  est le nombre entier tel que  $\lfloor u \rfloor \leq u < \lfloor u \rfloor + 1$

# Moyenne vs médiane



- Les trois statistiques ne coïncident pas
- Moyennes tronquées et médianes sont robustes aux points atypiques ( : *outliers*), la moyenne non!

# Dispersion: variance et écart-type



---

---

## Définitions

---

---

**Variance :** 
$$\text{var}_n(\mathbf{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_n)^2 = \frac{1}{n} \|\mathbf{y} - \bar{y}_n \mathbf{1}_n\|^2$$

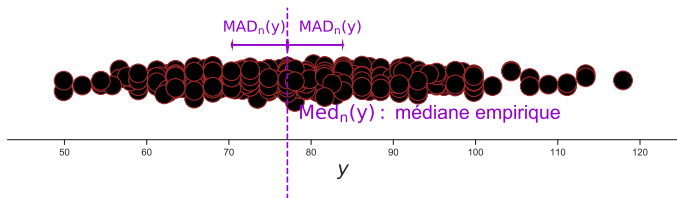
**Écart-type :** 
$$s_n(\mathbf{y}) = \sqrt{\text{var}_n(\mathbf{y})}$$

---

---

Notation: pour  $\mathbf{z} = (z_1, \dots, z_n)^\top$ ,  $\|\mathbf{z}\|^2 = \sum_{i=1}^n z_i^2$  ( $\|\mathbf{z}\|$  est la **norme** de  $\mathbf{z}$ ), et  $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$

# Dispersion: MAD



---

---

## Définition

---

---

**Déviati3n médiane absolue** (🇬🇧 : *Median Absolute Deviation*):

$$\text{MAD}_n(\mathbf{y}) = \text{Med}_n (|\text{Med}_n(\mathbf{y}) - \mathbf{y}|)$$

où  $\text{Med}_n(\mathbf{y})$  est la médiane de l'échantillon  $\mathbf{y} = (y_1, \dots, y_n)^\top$

---

---



# Covariances et corrélations empiriques

Soit deux échantillons  $\mathbf{x} = (x_1, \dots, x_n)^\top$  et  $\mathbf{y} = (y_1, \dots, y_n)^\top$

**Définition: covariance empirique**

$$\text{cov}_n(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}_n)(y_i - \bar{y}_n) = \frac{1}{n} \langle \mathbf{x} - \bar{x}_n \mathbf{1}_n, \mathbf{y} - \bar{y}_n \mathbf{1}_n \rangle$$

Notation:  $\mathbf{1}_n = (1, \dots, 1)^\top \in \mathbb{R}^n$  : vecteur constant

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i : \quad \text{produit scalaire}$$

**Définition: Coefficient de corrélation empirique**

$$\text{corr}_n(\mathbf{x}, \mathbf{y}) = \frac{\text{cov}_n(\mathbf{x}, \mathbf{y})}{\sqrt{\text{var}_n(\mathbf{x})} \sqrt{\text{var}_n(\mathbf{y})}} = \frac{\langle \mathbf{x} - \bar{x}_n \mathbf{1}_n, \mathbf{y} - \bar{y}_n \mathbf{1}_n \rangle}{\|\mathbf{x} - \bar{x}_n \mathbf{1}_n\| \|\mathbf{y} - \bar{y}_n \mathbf{1}_n\|}$$

Rem: on passe de covariance à corrélation en standardisant

# Standardisation

Soit un échantillon  $\mathbf{x} = (x_1, \dots, x_n)^\top \in \mathbb{R}^n$

---

---

**Définition: échantillon standardisé**

---

---

On note  $\tilde{\mathbf{x}}$  l'**échantillon standardisé** de  $\mathbf{x}$  obtenu comme suit

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - \bar{x}_n \mathbf{1}_n}{s_n(\mathbf{x})} \iff \tilde{x}_i = \frac{x_i - \bar{x}_n}{s_n(\mathbf{x})}, \quad \forall i \in \llbracket 1, n \rrbracket$$

où  $s_n(\mathbf{x}) = \sqrt{\text{var}_n(\mathbf{x})}$  sont l'écart type et  $\bar{x}_n$  la moyenne de l'échantillon

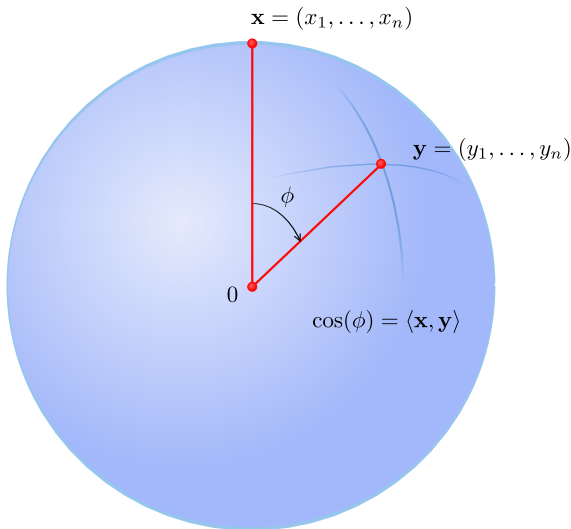
---

---

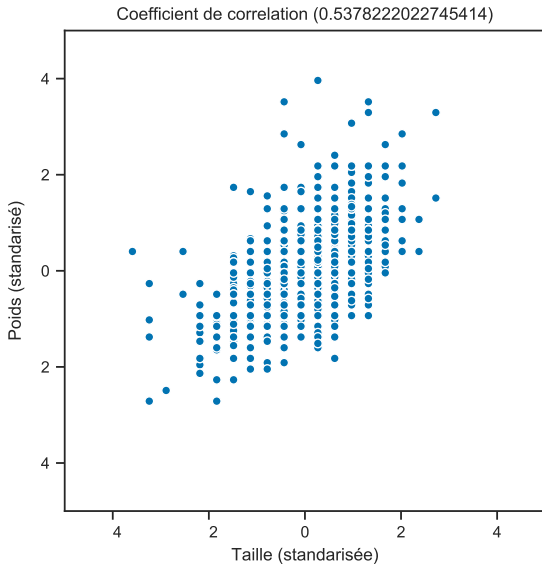
- ▶  $\tilde{\mathbf{x}}$  est
  - ▶ **centré** (moyenne nulle:  $\bar{\tilde{x}}_n = 0$ )
  - ▶ **réduit** (écart-type unitaire:  $s_n(\tilde{\mathbf{x}}) = 1$ )
- ▶  $\tilde{\mathbf{x}}$  est sans unité

# Interprétation de la corrélation:

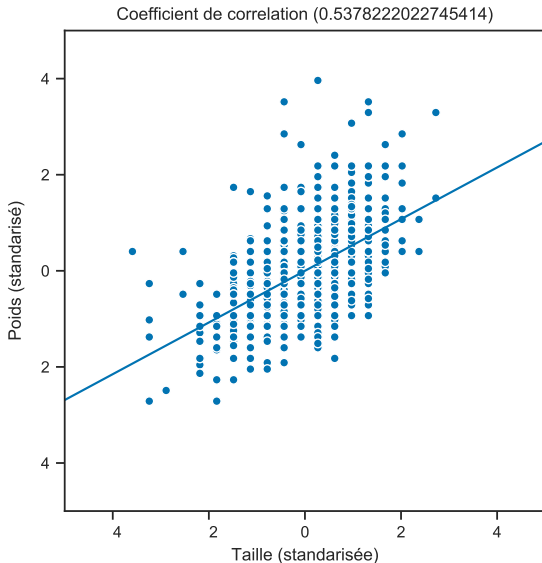
$$n = 3 \text{ et } \|\mathbf{x}\| = \|\mathbf{y}\| = 1$$



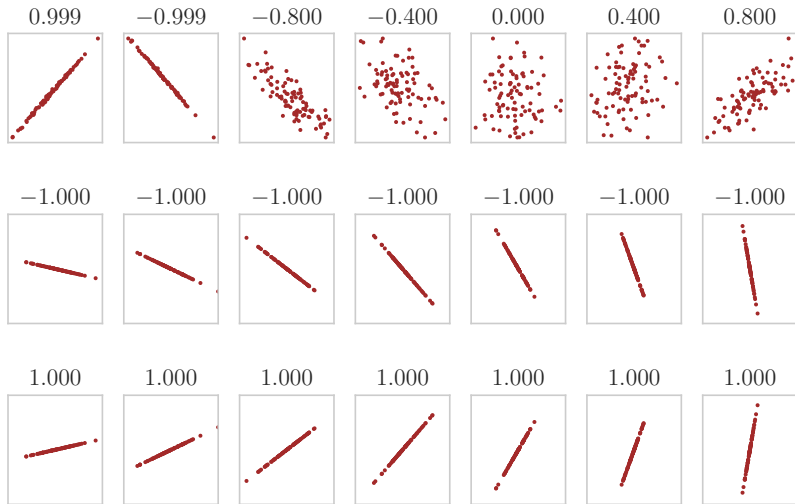
# Exemples de corrélations: taille du père / poids du père



# Exemples de corrélations: taille du père / poids du père

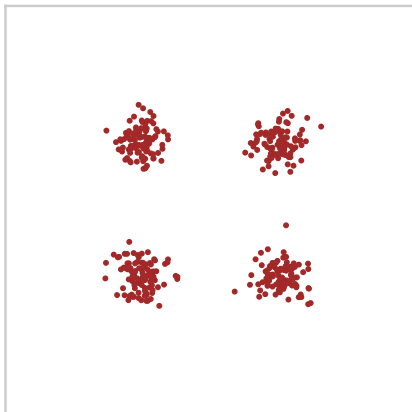


## Plus d'exemples



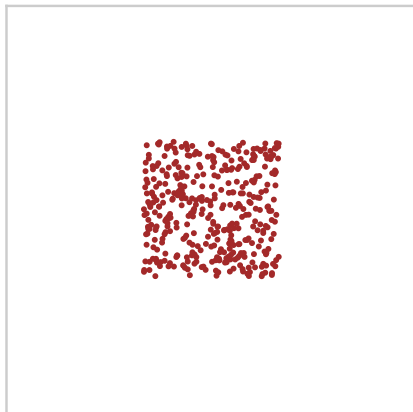
## Exemples de corrélations proches de zéro

Corrélation =  $-0.021$



## Exemples de corrélations proches de zéro

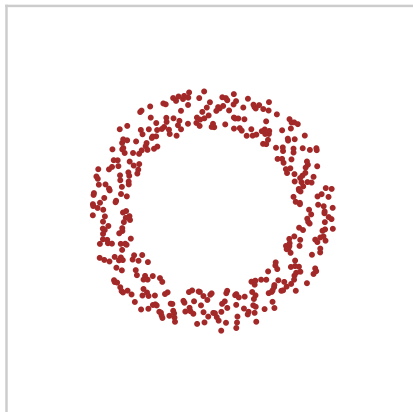
Corrélation = 0.007





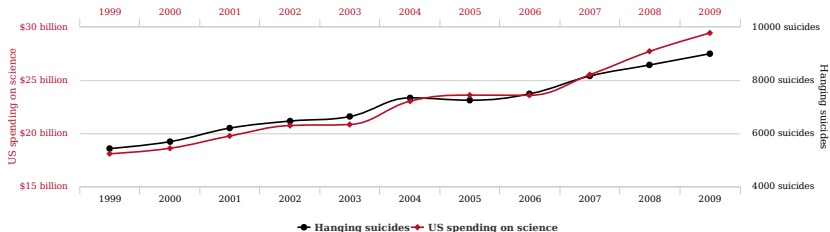
# Exemples de corrélations proches de zéro

Corrélation = 0.011



# Corrélation $\neq$ causalité

## US spending on science, space, and technology correlates with Suicides by hanging, strangulation and suffocation

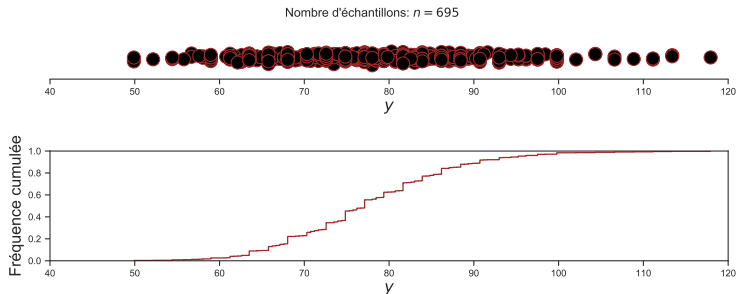


tylervigen.com

Corrélation: 0.9979

cf. <http://www.tylervigen.com/spurious-correlations>

# Fonction de répartition



---

---

**Définition: fonction de répartition**

---

---

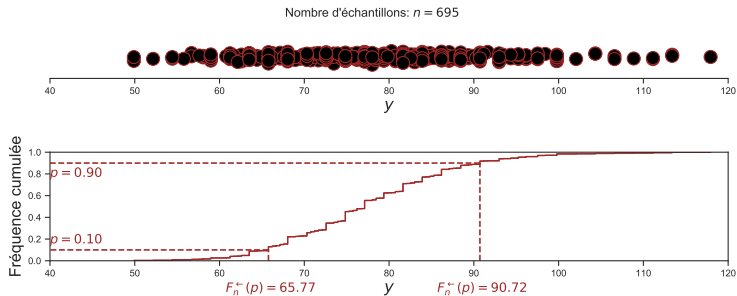
**Empirique :** 
$$F_n(u) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{y_i \leq u\}}$$

---

---

Interprétation: proportion d'observations sous un certain niveau

# Fonction quantile



---

---

**Définition: Quantile**

---

---

Pour  $p \in ]0, 1]$ ,  $F_n^{\leftarrow}(p) = \inf\{u \in \mathbb{R} : F_n(u) \geq p\}$

---

---

Rem: c'est l'inverse (généralisée) de la fonction de répartition; sa définition admet plusieurs conventions, cf. [percentile](#) in Numpy

# Quantiles

En bref: “le quantile d'ordre  $p$  est seuil tel que  $p \times 100\%$  des gens sont en dessous du seuil, et  $(1 - p) \times 100\%$  sont au-dessus”

- ▶ la médiane est le quantile d'ordre  $\frac{1}{2} = F_n^{\leftarrow}(\frac{1}{2})$
- ▶ le premier **quartile** ( $Q_1$ ) = quantile d'ordre  $\frac{1}{4} = F_n^{\leftarrow}(\frac{1}{4})$
- ▶ le troisième **quartile** ( $Q_3$ ) = quantile d'ordre  $\frac{3}{4} = F_n^{\leftarrow}(\frac{3}{4})$

Rem: de manière similaire on parle de déciles et de centiles


---

---

## Définition

---

---

L'**Écart interquartile** ( : *Interquartile range*), noté IQR, est défini comme étant l'écart entre le 3<sup>e</sup> quartile et le 1<sup>er</sup> quartile:

$$IQR = F_n^{\leftarrow}(\frac{3}{4}) - F_n^{\leftarrow}(\frac{1}{4})$$

---

---

## Quantiles (seconde définition)

Calcul de  $q_p(y)$ : quantile d'ordre  $p$  d'un échantillon  $y_1, \dots, y_n$ :  
 $y_{(1)} \leq y_{(2)} \leq \dots \leq y_{(n)}$ , il faut prendre l'indice  $j = \lfloor pn \rfloor$  (i.e.,  $j$  est le nombre entier juste avant  $pn$  si  $pn$  n'est pas entier)

$$q_p(y) = \begin{cases} \frac{1}{2}(y_j + y_{j+1}), & \text{si } j = pn \\ y_{j+1}, & \text{sinon} \end{cases}$$

**Exemple:**

- ▶  $n = 1000, p = \frac{1}{2} \implies j = 500 = \frac{1000}{2}$  et  
 $q_p(y) = \frac{1}{2}(y_{500} + y_{501})$
- ▶  $n = 1001, p = \frac{1}{2} \implies j = 500 \neq \frac{1001}{2}$  et  $q_p(y) = y_{501}$



cette convention<sup>(11)</sup> ne coïncide pas avec la précédente, ici on choisit le milieu de l'intervalle au lieu de l'extrémité gauche

---

<sup>(11)</sup>voir <https://fr.wikipedia.org/wiki/Quantile> pour d'autres conventions possibles

# Sommaire

Conseils numériques: pour le cours et au-delà

Choix de Python: introduction et motivation

Écosystème Python: les librairies

Introduction: Grossesse, cigarettes et nouveaux nés

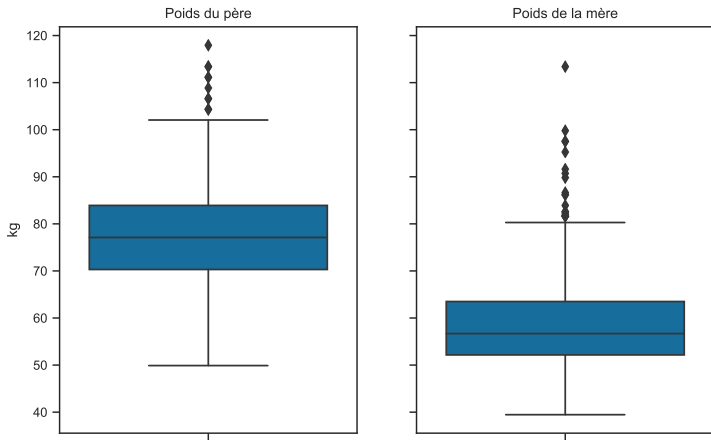
## Éléments de cours

Histogrammes et densités

Résumés numériques

Boîtes à moustache (Boxplot) et violons

## Exemple de boîte à moustache (🇬🇧 : *boxplot*)



Boxplot du poids des parents.



# Qu'est-ce qu'une boîte à moustache<sup>(12)</sup>?


- ▶ Représentation synthétique de la distribution d'une variable, similaire à l'histogramme, mais plus synthétique
- ▶ Utilité:
  - ▶ Comparer des distributions
  - ▶ Permet de détecter les "valeurs aberrantes"
  - ▶ Utile pour visualiser un grand nombre de variables (compact), et représenter des variables quantitatives en fonction de variables qualitatives

---

<sup>(12)</sup>R. McGill, J. W. Tukey, and W. A. Larsen. "Variations of box plots". In: *The American Statistician* 32.1 (1978), pp. 12–16.

# Construction d'une boîte à moustache

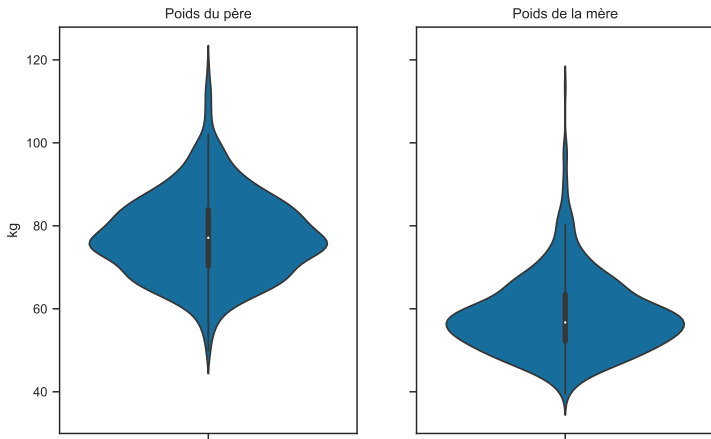
- ▶ la boîte est limitée par le 1<sup>er</sup> et le 3<sup>e</sup> quartiles
- ▶ elle est coupée en deux par la médiane
- ▶ les deux moustaches s'étendent de part et d'autre de la boîte sur une longueur (par défaut) de  $\frac{3}{2}$  fois l'écart inter-quartile

Rem: il y a parfois des modifications à la marge pour les cas extrêmes, e.g., affichage de points aberrants ( : *outliers*)<sup>(13)</sup>

---

<sup>(13)</sup> cf. [https://matplotlib.org/api/\\_as\\_gen/matplotlib.pyplot.boxplot.html](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.boxplot.html)

## Exemple de violons<sup>(14)</sup>



Violons du poids des parents  
(estimateur de densité pivoté et symétrisé)

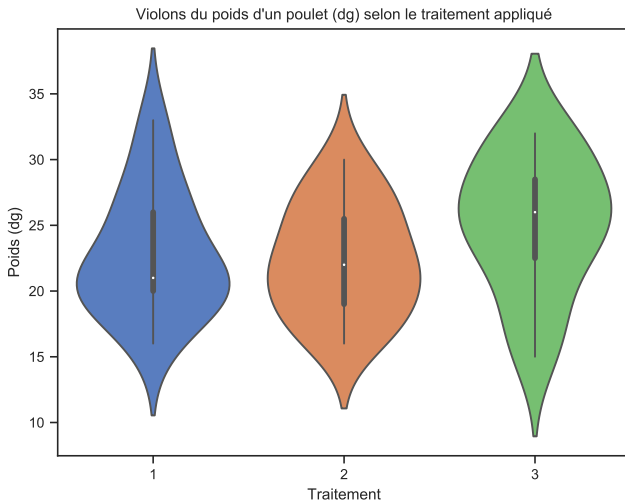
<sup>(14)</sup> J. L. Hintze and R. D. Nelson. "Violin plots: a box plot-density trace synergism". In: *The American Statistician* 52.2 (1998), pp. 181–184.

# Expérience: croissance des poussins

Scénario: des chercheurs veulent déterminer si parmi trois traitements possibles, il en existe un qui facilite la prise de poids des poussins. Données: ils disposent des résultats des traitements (avec trois températures d'incubation différentes) sur la croissance de  $n = 45$  poussins.

- ▶ Les 45 œufs sont répartis aléatoirement entre les trois types de traitements (15 | 15 | 15)
- ▶ Au bout d'un nombre de jours fixé à l'avance, on note la croissance (poids, en dg) des poussins et leur sexe

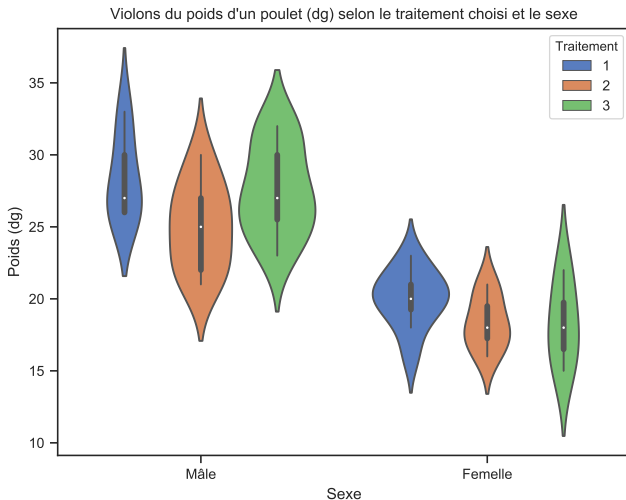
# Visualisation brute



Violons selon le type de traitement

Conclusion provisoire : le traitement 3 a le plus d'impact

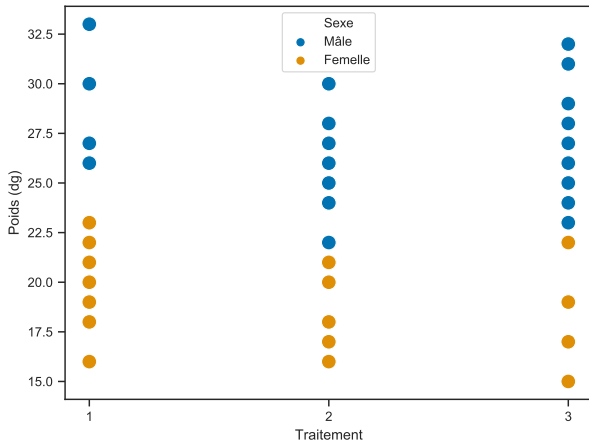
# Visualisation raffinée



Violons selon le type de traitement et le sexe

Conclusion : c'est le traitement 1 qui a le plus d'impact

# Explication



Répartition des poussins par sexe et par traitement

Conclusion : il y avait trop de femelles dans le traitement 1, et l'effet sexe a caché l'impact du traitement (groupe inhomogène)

# 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.
- ▶ Hintze, J. L. and R. D. Nelson. "Violin plots: a box plot-density trace synergism". In: *The American Statistician* 52.2 (1998), pp. 181–184.
- ▶ 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.



## Bibliographie II

- ▶ McGill, R., J. W. Tukey, and W. A. Larsen. “Variations of box plots”. In: *The American Statistician* 32.1 (1978), pp. 12–16.
- ▶ Nolan, D. and T. P. Speed. *Stat labs: mathematical statistics through applications*. Springer Science & Business Media, 2001.
- ▶ Parzen, E. “On estimation of a probability density function and mode”. In: *Ann. Math. Statist.* 33 (1962), pp. 1065–1076.
- ▶ Rosenblatt, M. “Remarks on some nonparametric estimates of a density function”. In: *Ann. Math. Statist.* 27 (1956), pp. 832–837.
- ▶ VanderPlas, J. *Python Data Science Handbook*. O'Reilly Media, 2016.