

# La programmation scientifique en Python

Pour l'analyse de réseaux

Émilien Schultz

[emilien.schultz@sciencespo.fr](mailto:emilien.schultz@sciencespo.fr)

médialab - SESSTIM

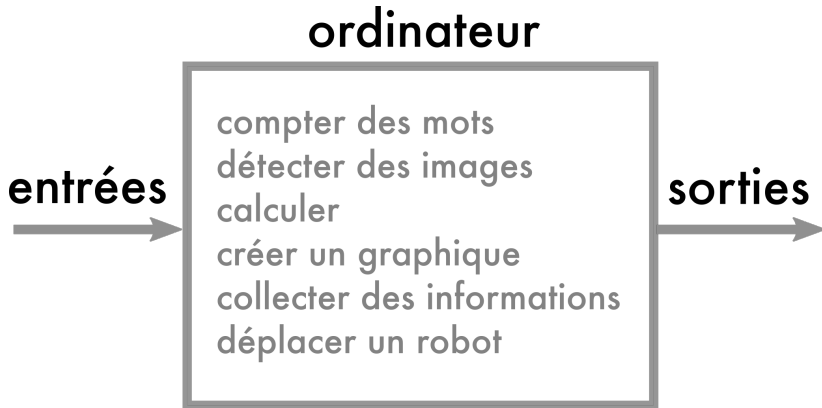
# Petit panorama rapide sur la programmation scientifique

1. Pourquoi programmer (en recherche) ?
2. Pourquoi Python ?



# Pourquoi programmer ?

**Programmer**[Définition pratique] : utiliser un ensemble de commandes (code) dans un langage (de programmation) pour faire réaliser (exécuter) à l'ordinateur des tâches.



# Cinquante nuance de programmation, dont la programmation scientifique

- ▶ Des *styles* de programmation différentes (paradigmes)
  - ▶ Impératif/Procédural
  - ▶ Orienté objet
  - ▶ ...
- ▶ Un usage spécifique pour la recherche : **la programmation scientifique**
  - ▶ Orientation **script** : réaliser des petites tâches spécifiques
  - ▶ Orientation **interactive** : tester et expérimenter
  - ▶ Orientation **recherche** : des outils spécifiques
- ▶ Usage compatible avec des logiciels et le reste des pratiques

# Script scientifique et *literate programming*

Une pratique largement orientée data science, *plus "légère"*, avec ses outils dédiés.

Intégration du code et du texte (Knuth, 1992) puis des résultats dans la *literate computing*.

## Casual Notebooks and Rigid Scripts: Understanding Data Science Programming

Krishna Subramanian, Nur Hamdan, Jan Borchers  
RWTH Aachen University  
52074 Aachen, Germany  
{krishna, hamdan, borchers}@cs.rwth-aachen.de

**Abstract**—Data workers are non-professional data scientists who often use scripting languages like R, Python, or MATLAB, and employ an exploratory programming workflow. Current IDEs offer them two main programming modalities: script files and computational notebooks. To understand how these modalities impact work practice, we conducted a study with 21 data workers, and a subsequent larger survey with 62 respondents. Through interviews, walkthroughs, and screen recordings, we collected information about their workflows. Our analysis shows a tension between scripts and computational notebooks. Scripts are more common, better support storage and execution of previous analyses, but hinder experimentation. Notebooks better suit the actual data science workflow, but can become easily unorganized. We discuss how this dual nature of modality usage leads to several issues that affect data workers' workflows, and discuss implications for the design of programming IDEs.

**Index Terms**—scripting languages, exploratory programming, programming interfaces, data science, notebooks

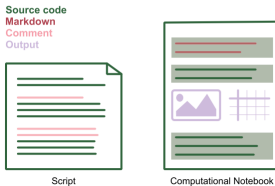
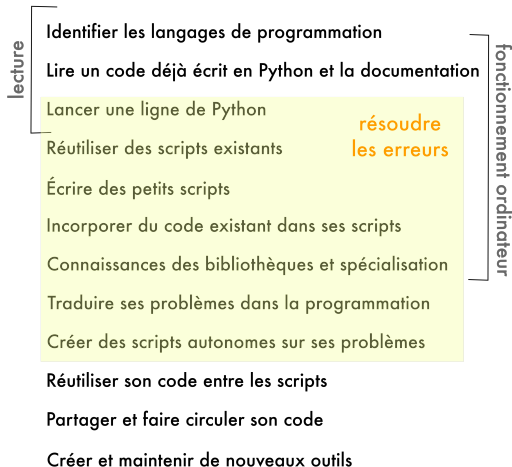


Fig. 1. Current scripting language IDEs support writing and executing code via two programming modalities: scripts (left) and computational notebooks (right). In this paper, we investigate how these modalities are used in data

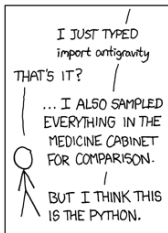
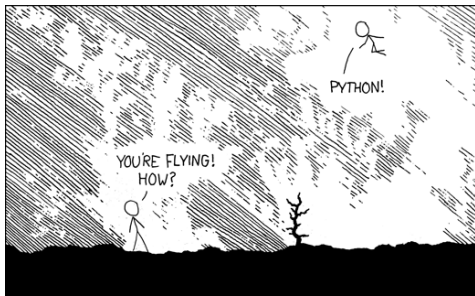
# En pratique : une diversité de niveaux de compétences

## Découvre la programmation



Contributeur •rice Open Source accompli • e

# Pourquoi Python ? Parce que tout est possible



<https://xkcd.com/353/>

# Propriétés de Python

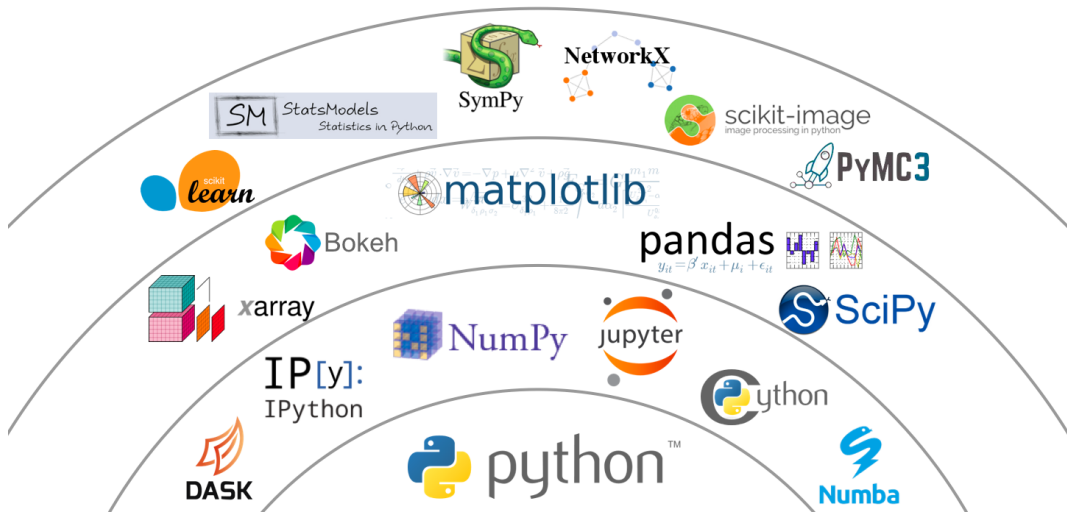
- ▶ Libre et interopérable (interprété)
- ▶ *versatile* par rapport aux manières de l'utiliser
- ▶ Pédagogique *by design*
- ▶ De nombreuses ressources / documentation
- ▶ Favorise les bonnes pratiques de programmation
- ▶ En croissance d'usage (recherche et privé)
- ▶ Un avenir brillant : enseigné dès le lycée



# Python, plus qu'un langage

## Python's Scientific Stack

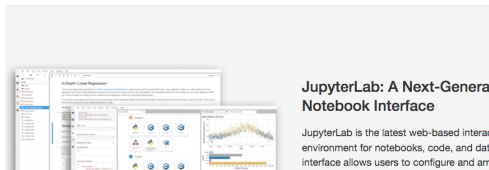
Jake Vanderplas PyCon 2017 Keynote



# De nombreux outils



Free software, open standards, and web services for interactive computing across all programming languages



Broken Barh



CSD Demo



EventCollection Demo



CapStyle



Curve with error band



Eventplot Demo



Plotting categorical variables



Errorbar limit selection



Filled polygon



Plotting the coherence of two signals



Errorbar subsampling



Fill Between and Alpha

- Lines, bars and markers
- Images, contours and fields
- Subplots, axes and figures
- Statistics
- Pie and polar charts
- Text, labels and annotations
- pyplot
- Color
- Shapes and collections
- Style sheets
- axes\_grid1
- axisartist
- Showcase
- Animation
- Event handling
- Front Page
- Miscellaneous
- 3D plotting
- Scales
- Specialty Plots
- Spines
- Ticks
- Units
- Embedding Matplotlib in graphical user interfaces
- Userdemo
- Widgets

Gallerie Matplotlib

## Mais pas le seul choix...

Convergence et divergences avec d'autres langages, R en premier lieu



### DATA SCIENTISTS



***R is the  
best  
language***



***Noooooo,  
Python is  
better***

### STATISTICIANS



***R is the  
best  
language***



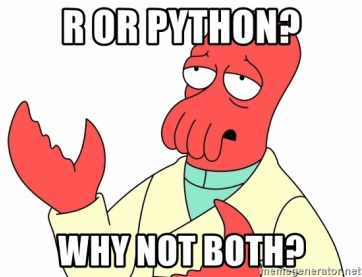
***I know***

Pourquoi choisir Python ?

# Python ou R? Python et R? Ou quoi encore?

- ▶ Python et R permettent la majorité des traitements associés à la collecte des données, au traitement, et à la visualisation, et évoluent en permanence.
- ▶ Python est davantage compris par les informaticiens et assimilés + secteur privé
- ▶ R excellent pour les statistiques
- ▶ Python est en avance pour les applications en machine learning
- ▶ Python permet de déployer
- ▶ Python semble avoir une meilleure logique de documentation

**Dans tous les cas, importance des ressources disponibles pour apprendre : collègues, etc.**



## S'y mettre ?

- ▶ Base du langage
- ▶ Découvrir les bibliothèques et les outils
- ▶ Se spécialiser dans certains bibliothèques

## Les obstacles

- ▶ Un outil parmi d'autres : **pas une baguette magique**
- ▶ Courbe d'apprentissage potentiellement longue (mais...)
- ▶ Avoir une idée de quoi en faire : quel imaginaire pratique ?
- ▶ Trouver des ressources locales : importance de la pratique



# Programmer $\neq$ Tout savoir

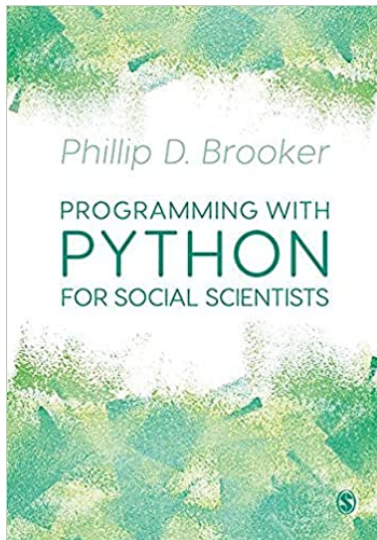
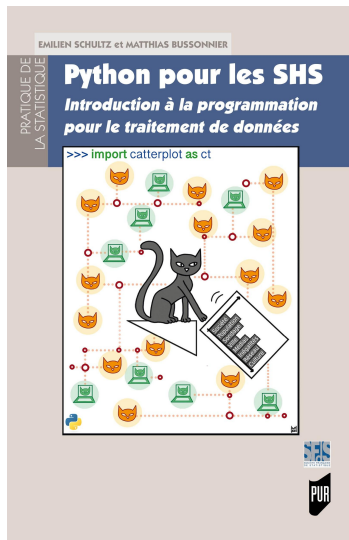
Apprendre à programmer signifie apprendre à potentiellement pouvoir utiliser de nombreux outils développés par des chercheurs.

Mais chaque domaine a ses savoirs spécifiques : *machine learning*, analyse de réseaux, textométrie, ...

La frontière peut être difficile à tracer.

- ▶ Réutilisation d'outils facilité
- ▶ Mais cela ne remplace pas une connaissance experte

# Ressources



<https://github.com/pyshs/ressources-pyshs>