

Python pour les SHS

Pourquoi faire (un peu) de programmation

Émilien Schultz

emilien.schultz@sciencespo.fr

médialab - SESSTIM

Ce que j'aimerai faire aujourd'hui

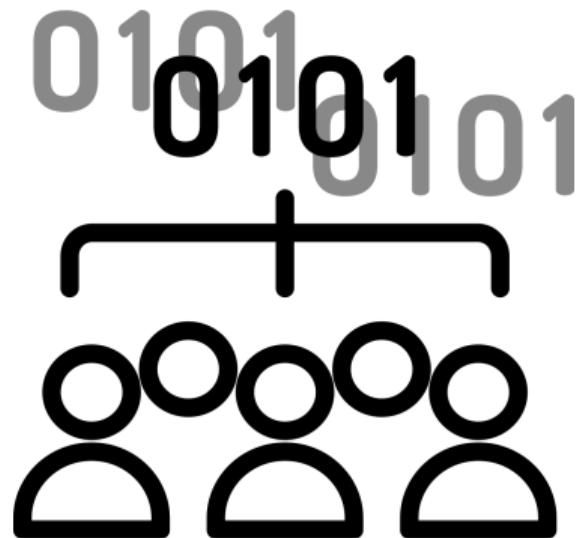
- ▶ Parler un peu de ce qu'est la programmation
- ▶ Parler un peu des bases du langage
- ▶ Montrer quelques usages utiles pour les données textuelles
- ▶ Échanger !

Le dépôt Github

<https://github.com/pyshs/tutorial-quali>

Répondre à 3 questions

1. Pourquoi programmer (en recherche) ?
2. Pourquoi Python ?
3. Quelle spécificité pour les SHS ?



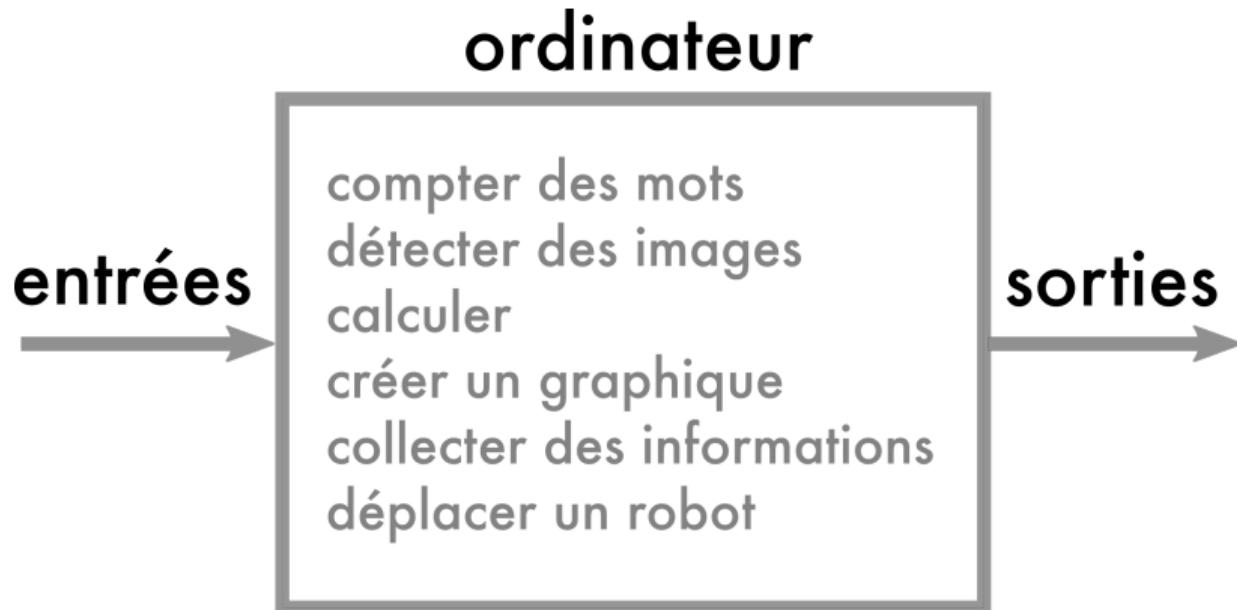
1. Pourquoi programmer ?

La numérisation de la recherche

- ▶ Traitement numérique comme point de passage obligé du•de la chercheur•se
 - ▶ *digital turn*
- ▶ Explosion de la disponibilité des données
 - ▶ *manipulation données*
- ▶ Courant profond et puissant de la science ouverte
 - ▶ *reproductibilité traitements*
- ▶ Apparition d'objets/méthodes liés aux pratiques numériques
 - ▶ *nouveaux terrain(s)*

Programmer !

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



Un **algorithme** est la description d'une suite d'étapes permettant d'obtenir un résultat à partir d'éléments fournis en entrée

Pour le faire : un ensemble de savoirs interdépendants

- ▶ Notions générales sur le fonctionnement d'un ordinateur (stockage, calcul, périphériques)
- ▶ Environnements spécifiques (OS et logiciels)
- ▶ Penser la logique des instructions : algorithmiques
 - ▶ ensemble ordonné d'instructions
- ▶ Exprimer ces instructions : langages de programmation
- ▶ Formats spécifiques des données
- ▶ Diversité d'outils/savoirs associés
 - ▶ Debugger

Les langages de programmation

Abstractions permettant de réaliser des opérations

- ▶ Des langages différents (plus ou moins abstraits)
- ▶ Des opérations partagées par tous les langages (opérations mathématiques)
- ▶ Infrastructure pour passer de l'opération à sa réalisation
 - ▶ Compilation (logiciel)
 - ▶ Interprétation

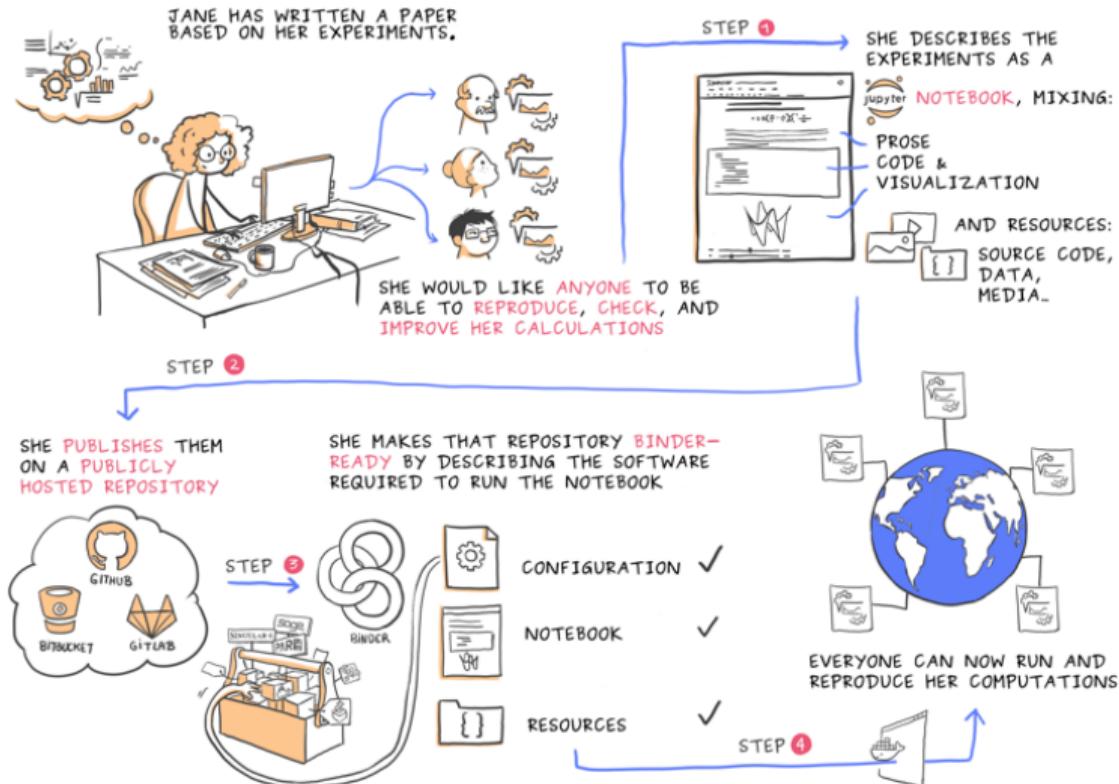
Cinquante nuance de programmation

- ▶ 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

Programmer pour la recherche : entre standardisation et adaptation

- ▶ créer un dialogue interactif avec l'ordinateur (exploration et stabilisation)
- ▶ formaliser des manipulations pour les partager (reproductibilité des traitements)
- ▶ adapter à des tâches non prévues par les logiciels (flexibilité)
- ▶ interconnecter des opérations sinon séparées (glue)

Plus généralement, les enjeux de reproductibilité



Open and reproducible scientific workflow using Jupyter notebook and related open-source tools. Source : Juliette Taka and Nicolas M. Thiery. Publishing reproducible logbooks explainer comic strip. Zenodo. DOI : 10.5281/zenodo.4421040 (2018).

2. Pourquoi Python ?

Un monde de langages

Liste de langages de programmation

49 langues

Article Discussion

Lire Modifier Modifier le code Voir l'historique

Le but de cette liste de langages de programmation est d'inclure tous les langages de programmation existants, qu'ils soient actuellement utilisés ou historiques, par ordre alphabétique. Ne sont pas listés ici les langages informatiques de représentation de données tels que XML, HTML, XHTML ou YAML. Un langage de programmation doit permettre d'écrire des algorithmes, mais il n'est pas nécessaire qu'il soit Turing-complet (par exemple Gallina, le langage de programmation de Coq, ne l'est pas).

Par ailleurs, cette liste répertorie les langages de programmation, et non leurs implémentations (par exemple, JRuby et IronRuby sont deux implémentations différentes du même langage Ruby).

Sommaire : Haut - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

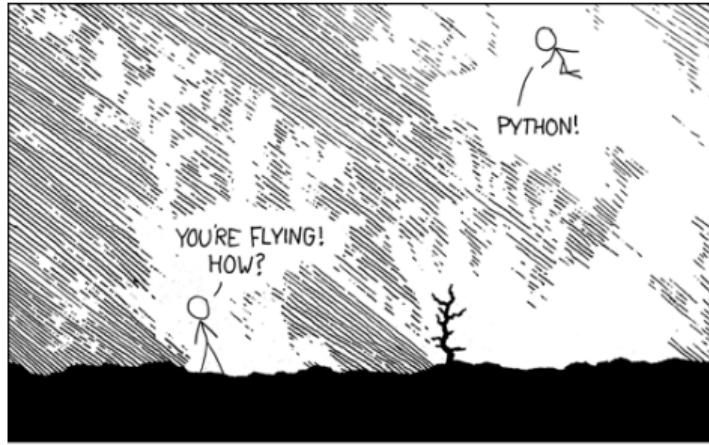
A [modifier | modifier le code]

- A+
- A++
- A# .NET
- A# (Axiom) (en)
- A-0 System
- ABAL
- ABAL++
- ABAP
- ABC
- ABCL/I
- ABCL/c+
- ABCL/R
- ABCL/R2
- Abel
- ABSET (en)
- ABSYS
- ALI
- Abundance
- ACC (programming language) (en)
- Accent
- ActForex
- Ace DASL
- ACT-III
- Ada
- Adenine
- Afrix
- Agora (programming language) (en)
- AIS Balise
- Aikido
- Alef
- Algebraic Logic Functional programming language (en)
- Algol 60
- Algol 68
- Algol W
- Alice (programming language) (en)
- Ambi
- Amiga E (en)
- AML
- AMOS
- AMPLE
- Anubis
- APDL
- APL
- AppleScript
- Arc
- Ariberion
- Arobase (langage)
- Assembleur
- ASP.NET
- ATS
- AUPL
- AutoHotkey
- AutoIt
- Averest
- awk
- axe parser
- Axum (programming language) (en)

B [modifier | modifier le code]

- B
- Bah-Lang
- BASIC
- BASICa
- Basic Nspire
- QuickBasic
- SmallBasic
- TI-Basic
- True Basic
- Turbo Basic
- Beef
- Befunge
- Bennu
- Bertrand
- BETA
- Bon
- Boo
- Boomerang
- Bosque
- Bourne shell (sh)

Tout est possible avec Python (sur un ordinateur)

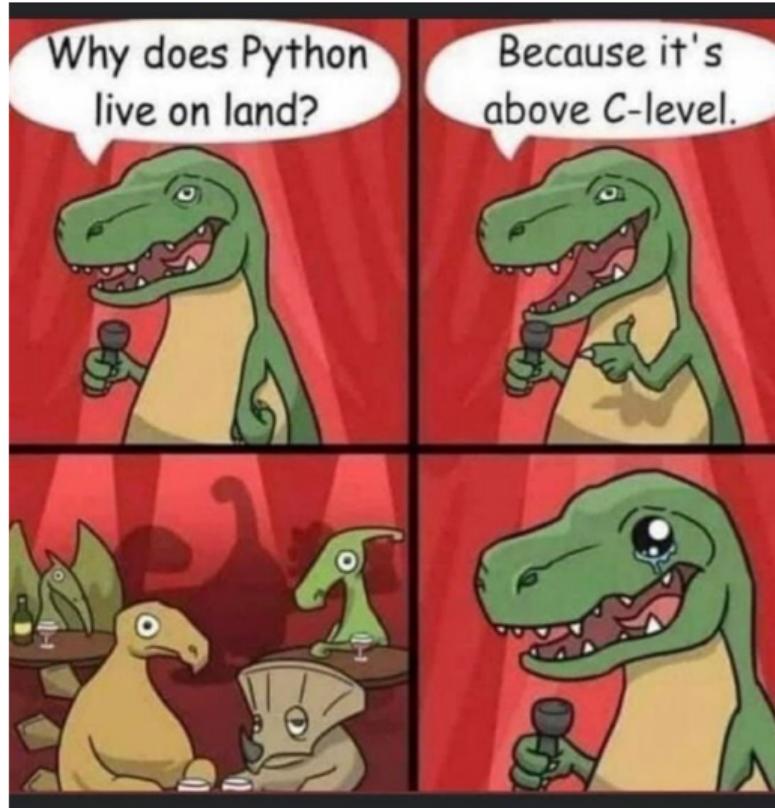


I LEARNED IT LAST
NIGHT! EVERYTHING
IS SO SIMPLE!
/ HELLO WORLD IS JUST
print "Hello, world!"

I DUNNO...
DYNAMIC TYPING?
WHITESPACE?
/ COME JOIN US!
PROGRAMMING
IS FUN AGAIN!
IT'S A WHOLE
NEW WORLD
UP HERE!
/ BUT HOW ARE
YOU FLYING?

I JUST TYPED
import antigravity
/ THAT'S IT?
/ ... I ALSO SAMPLED
EVERYTHING IN THE
MEDICINE CABINET
FOR COMPARISON.
/ BUT I THINK THIS
IS THE PYTHON.

Un langage de haut niveau



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



Le produit d'une histoire et d'un écosystème

Une histoire qui construit sur les autres langages :

<https://inference-review.com/article/the-origins-of-python>

"It makes sense to think of the realm of programming languages as an ecosystem in which languages occupy their own niches. FORTRAN's niche is high-performance scientific programming, involving heavy-duty numerical computation ; that of COBOL is administration, based on files of data records. The C language is designed for systems programming, originally developed specifically for the Unix operating system. Just as there is no such thing as a general-purpose transportation vehicle, a truly one-size-fits-all general-purpose programming language does not exist ; for a given highly specialized application domain it will always be possible to design a language tailored to, and better suited for, the specific needs of that domain [...] Python was originally designed to serve as a high-level scripting language for the Amoeba project. ABC was completely unsuitable for this purpose ; it lived in a world of its own, shielding its users—by design—from the outside world. Python was expressly designed to interface with that outside world."

The Zen of Python

Tim Peters a résumé les principes du BDFL en 19 aphorismes :

Beautiful is better than ugly.

Explicit is better than implicit.

Simple is better than complex.

Complex is better than complicated.

Flat is better than nested.

Sparse is better than dense.

Readability counts.

Special cases aren't special enough to break the rules.

Although practicality beats purity.

Errors should never pass silently.

Unless explicitly silenced.

In the face of ambiguity, refuse the temptation to guess.

There should be one— and preferably only one —obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than *right* now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

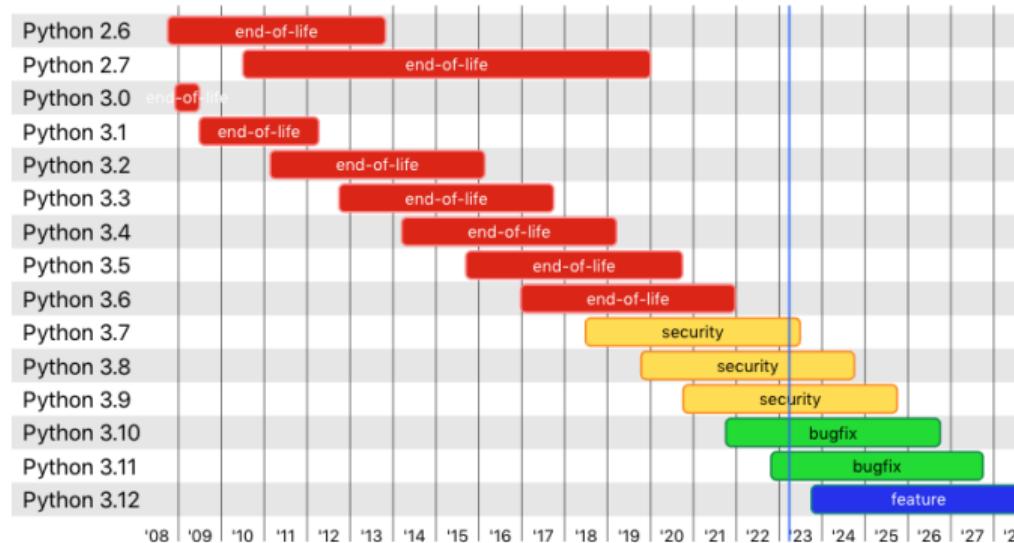
Namespaces are one honking great idea – let's do more of those !

Un langage qui a évolué et intégré les bonnes pratiques

Status of Python Versions

The main branch is currently the future Python 3.12, and is the only branch that accepts new features.
The latest release for each Python version can be found on the [download page](#).

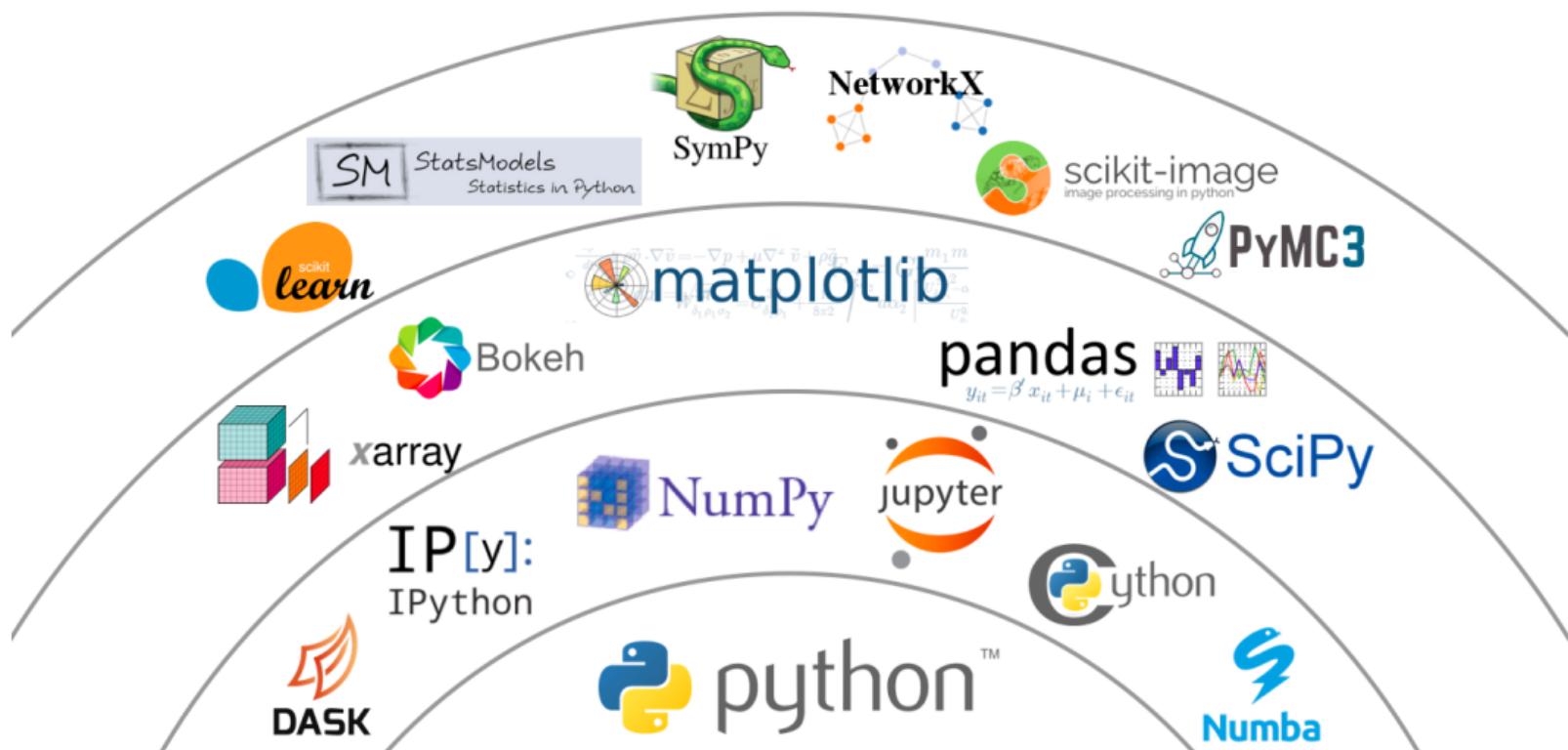
Python Release Cycle



Plus qu'un langage : un univers d'outils

Python's Scientific Stack

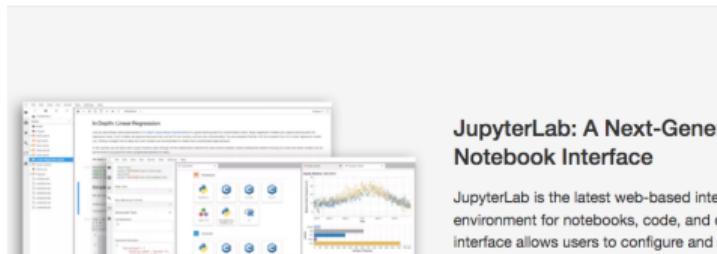
Jake Vanderplas PyCon 2017 Keynote



De nombreux outils



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



JupyterLab: A Next-Generation Notebook Interface

JupyterLab is the latest web-based interface for notebooks, code, and data. This interface allows users to configure and arrange workflows in



Gallerie Matplotlib

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

Des bibliothèques puissantes

learn Install User Guide API Examples Community More ▾

scikit-learn

Machine Learning in Python

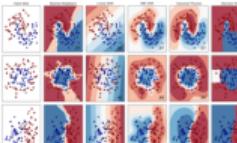
Getting Started Release Highlights for 1.0 GitHub

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

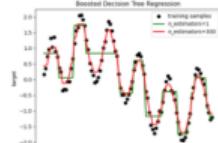


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...



Clustering

Automatic grouping of objects into sets.

Applications: Customer segmentation, Grouping experiment on digits.

Algorithms: k-Means, hierarchical clustering, mean-shift, and more...



Simple and efficient tools for predictive data analysis
Accessible to everybody, and reusable in contexts
Built on NumPy, SciPy, and matplotlib
Open source, commercially usable -

spaCy *Out now: spaCy v3.3

USAGE MODELS API UNIVERSE 23,242 Search docs

Industrial-Strength Natural Language Processing

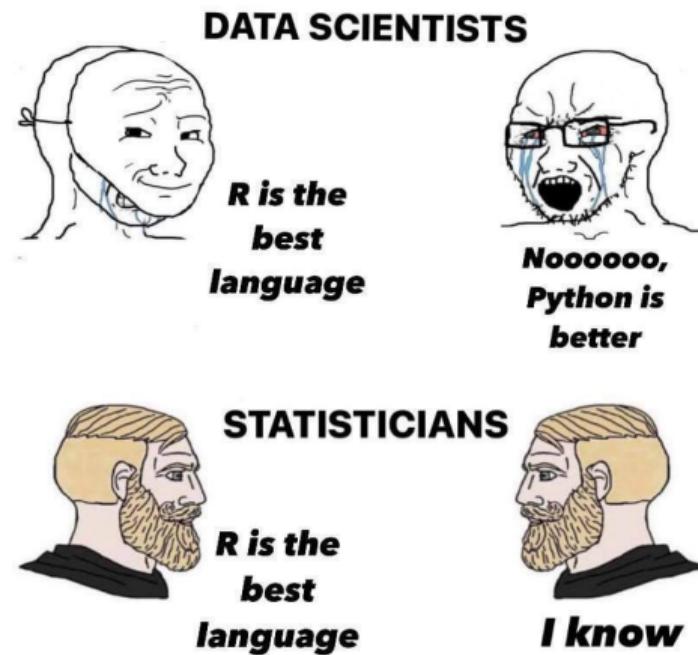
IN PYTHON

Get things done

spaCy is designed to help you do real work — to build real products, or gather real insights. The library respects your time, and tries to avoid wasting it. It's easy to install, and its API is simple and productive.

Mais pas le seul choix...

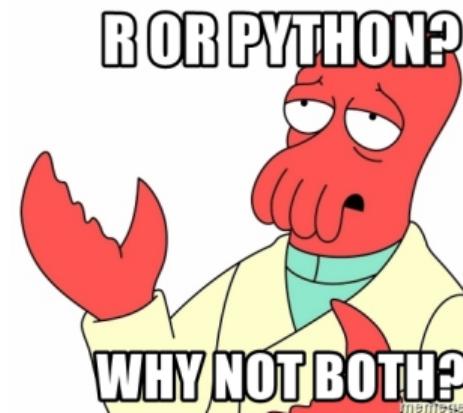
Convergence et divergences avec d'autres langages, R en premier lieu



Qui mène à la question centrale : dois-je choisir Python ?

Python ou R ? Python et R ? Tous ensemble dans la programmation scientifique

- ▶ 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, important d'initier d'un usage et importance des ressources disponibles pour apprendre : collègues, etc.

3. Les usages spécifiques des SHS ?

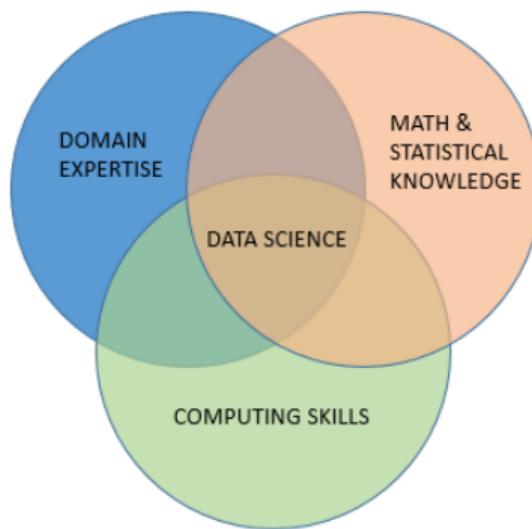
Des identités en transformation autour du numérique

Revenir à la poussière ? L'identité professionnelle des historiens et historiennes

Le livre d'Arlette Farge (1989) a connu un tel succès national et international qu'il semble avoir contribué à stabiliser la définition même du métier d'historien et d'historienne autour de celui ou celle qui noircit ses mains de poussière, qui « descend aux archives », etc. C'est la raison pour laquelle les médiations numériques sont très peu évoquées dans les remerciements de thèse, les blogs ou, plus simplement, les livres : historiens et historiennes seraient prisonniers de « faux récits de l'archive » qui le conduisent à valoriser la mise en scène du contact physique au document plutôt que la réalité du travail derrière l'écran ou la fouille via les moteurs de recherche⁸. Un certain « récit de l'archive », déphasé par rapport aux pratiques réelles, reste central dans la construction de l'identité professionnelle. La numérisation du métier est pourtant bien avancée : rares sont les gestes qui ne sont pas médiés par l'ordinateur ou l'instrument, scanner, téléphone ou encore appareil photo. Comment expliquer ce décalage entre récit de l'archive et pratiques concrètes ? Le déni de la numérisation du métier dans la présentation des coulisses des enquêtes historiques révèle la force des représentations qui lient empathie, imprégnation du passé et immersion dans des cartons de documents physiques. Quels seraient des récits d'archive plus proches des pratiques ?

L'autonomisation de la "data science"

- ▶ De plus en plus autonome comme littérature (manuels dédiés, beaucoup tournés vers l'opérationnel)
- ▶ Toujours relatif à des domaines spécifiques



Hétérogénéité des SHS

- ▶ Rôle central de la problématique (perspectivisme)
- ▶ Méthodologies très variées
- ▶ Données plus ou moins accessibles et normalisées
- ▶ Culture du numérique variable



Des dynamiques en cours

4. FOCUS SUR 3 OUTILS NUMÉRIQUES ET 3 LOGIQUES D'INNOVATION

Nous procédons à une analyse plus approfondie des 3 premiers outils les plus cités : Excel, R et Python. Leurs caractéristiques propres en font à la fois des « concurrents » et des outils complémentaires. Notre analyse tente d'évaluer si l'on peut trouver des profils de chercheurs, qui par leurs caractéristiques propres peuvent être associés à chacun de ces trois outils. Nous constatons que nous rencontrons trois configurations. Nous rencontrons l'innovation : en voie d'institutionnalisation (N. Alter, 2015) symbolisée par R; le logiciel institutionnalisé représenté par Excel; et la pratique en émergence avec Python.

Les utilisateurs de R (n = 244): la voie de l'institutionnalisation

Une moyenne d'âge des utilisateurs de R plus jeune

Les utilisateurs de R se caractérisent par une moyenne d'âge et un âge médian inférieur d'environ 4 ans à la POP. L'usage de R est lié à des chercheurs parmi les plus jeunes, les écarts étant sensibles pour les 35-45 ans et nettement plus marqués pour les chercheurs de moins de 35 ans.

Constats (à discuter)

- ▶ Une division persistante quanti/quali que la programmation permet de dépasser
- ▶ Des usages "discrets" plus que "computationnels" à identifier
- ▶ Programmation souvent ramenée aux statistiques (et à R)
- ▶ Encore peu de bibliothèques Python dédiées SHS (donc de la place pour en développer de nouvelles)
- ▶ Des usages encore peu stabilisés (Notebooks, etc.)

4. En pratique, ça sert à quoi ?

Cas : format de données

Passer d'un fichier *.html* à un *.txt* mis en forme pour l'ramuteq

Les Echos, no. 23183 événement, vendredi 20 mars 2020 813 mots, p. 3
Coronavirus
Aussi perdu dans 19 mars - lesechos.fr
2020
Les cliniques privées à la rescoussse SOLVEIG GODELUCK
En Alsace, où les hôpitaux publics sont débordés, les éti
Certains sont dans la tempête; d'autres l'attendent. Aло Faute de patients atteints du Covid-19. « Nous avons directeur général de la Fondation Saint-Vincent à Stras
Des lits transformés pour la réanimation
Ces disponibilités ont pourtant été signa pouvoir entrer dans le dispositif », plaide Christophe M
« Nous ne sommes pas sollicités à hauteur du service Samu : on oriente les malades vers le secteur public. L tous les deux jours, on a déprogrammé toutes nos opér
100.000 soins déprogrammés dans le privé lucratif



renforcement » dans d'autres. Le lendemain, le ministre de l'Intérieur a lui-même été infecté, a annoncé l'extension des tests de dépistage et a déclaré qu'il se lancerait dans le déconfinement Sophie Amsili et Tiffen Clémentine.

***** *num 618 *journal LeFigaro

«Pendant trois heures, Emmanuel Macron a pris connaissance à résultats obtenus par l'équipe du Pr Raoult», se réjouit la **Martine Wonner**, seule parlementaire LREM à «**Covid-19-Laissons les médecins prescrire**.» **LIRE AUSSI -** **Rapport** : les dessous d'une rencontre surprise. Cette psychologue, dont les positions souvent plus tranchées que celles de ses collègues, Elle s'était aussi engagée avec les écologistes, contre le tournoiement ouest de Strasbourg, dont l'énorme chantier a

 IRaMuTeQ

Ou encore : passer d'un fichier .pdf à un .txt pour faire du TAL

Cas : construire un réseau

Créer la bonne structure relationnelle (ici auteur/auteur) et l'exporter dans un format compatible avec Gephi

	A	B	C	D	E
1	ID	ANNEE	AUTEURS	TITRE	JOURNAL
2	35	1996	LEROUX A., BRETAGNOLLE V	Sex ratio var	Journal of Avia
3	37	1996	A RECODER	SALAMOLARD M., BRETAGNOL	
4	44	1998	ARROYO B.E., LEROUX A.B.A.	B Egg and clute journal	of Reptile
5	47	1998	de CORNUNIER T., BERNARD R.	Nmodification of Repro	
6	52	1999	ARROYO B.E., BRETAGNOLLE V	Breeding bird	Journal of Avia
7	55	1999	SALAMOLARD M., MOREAU C.	Habitat selec	Bird Study
8	58	2000	AMAR A., ARROYO B.E., BRETAGNOLLE V.	Poss fledgling	Ibis
9	59	2000	ARROYO B.E., DECORNILLIET J.	Sex and age	Condor
10	62	2000	GUILLEMIN M., HOUTTE S., FRIN	Activities and	Revue d'Ecolog
11	63	2000	JIGUET F., ARROYO B., BRETAGNOLLE V.	Matting	Behaviour
12	68	2000	SALAMOLARD M., BUTET A., LEB	Responses or	Ecology
13	69	2001	ARROYO B., MOUGEOU F., BREY	Colonial tree	Behavioural E
14	70	2001	CLERE E., BRETAGNOLLE V	Disponibilité	Revue d'Ecolog
15	71	2001	JIGUET F., BRETAGNOLLE V.	Behaviour	Behaviour



```
*rml version='1.0' encoding='utf-8'>
graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">
  <key id="d0" for="node" attr.name="name" attr.type="string"/>
  <key id="d5" for="node" attr.name="author" attr.type="string"/>
  <key id="d4" for="node" attr.name="cluster" attr.type="string"/>
  <key id="d2" for="node" attr.name="cat" attr.type="string"/>
  <key id="d1" for="node" attr.name="named" attr.type="long"/>
  <key id="d8" for="node" attr.name="label" attr.type="string"/>
  <key id="d3" for="edge" attr.name="weight" attr.type="double"/>
  <data key="d1">Sex ratio variations in broods of Montagu's harrier</data>
  <data key="d2">1999</data>
  <data key="d3"><Article></Article></data>
  <data key="d4">1</data>
  <data key="d5">1</data>
  <data key="d8">Sex ratio </data>
```



Cas : construction de tableaux adaptés

Produire des sorties de tableaux adaptés à l'objet (et possibilité ensuite d'aller sur Excel ou Latex)

```
Entrée [64]: var_ind = {"sexe":"1 - Sex","age2":"2 - Age","diplome":"3 - Education", "revenus":"4 - Incomes",
                     "PROXPARTI":"5 - Political orientation"}

t = {"COCONEL1":pyshs.tableau_croise_multiple(data1,"HC_c",var_ind,chi2=False)[["1 - HC effective","2 - HC not effect",
                     "COCONEL2":pyshs.tableau_croise_multiple(data2,"HC_c",var_ind,chi2=False)[["1 - HC effective","2 - HC not effect",
                     "COCONEL3":pyshs.tableau_croise_multiple(data3,"HC_c",var_ind,chi2=False)[["1 - HC effective","2 - HC not effect",
                     "TRACTRUST1":pyshs.tableau_croise_multiple(data4,"HC_c",var_ind,chi2=False)[["1 - HC effective","2 - HC not effect",
                     "TRACTRUST2":pyshs.tableau_croise_multiple(data5,"HC_c",var_ind,chi2=False)[["1 - HC effective","2 - HC not effect

t = pd.concat(t,axis=1)
t.applymap(lambda x : re.findall("\((.*?)%\)",x)[0])
```

Out[64]:

Variable	Modalités	COCONEL1		COCONEL2		COCONEL3		TRACTRUST1		TRACTRUST2	
		1 - HC effective	2 - HC not effective	1 - HC effective	2 - HC not effective	1 - HC effective	2 - HC not effective	1 - HC effective	2 - HC not effective	1 - HC effective	2 - HC not effective
1 - Sex	Femme	38.3	3.9	34.0	9.1	17.8	9.0	14.2	13.4	15.8	18.8
	Homme	36.8	7.4	27.2	13.6	21.6	14.7	19.5	19.0	16.2	29.1
	Total	37.6	5.6	30.8	11.3	19.6	11.7	16.7	16.1	16.0	23.9
2 - Age	17-34	36.7	8.9	27.8	15.4	16.8	14.7	14.6	20.4	14.4	25.8
	35-54	41.1	4.5	31.3	10.1	19.9	11.8	18.4	14.2	15.8	23.9
	55-79	36.8	4.0	33.3	10.2	23.3	8.9	17.7	16.7	18.8	20.1
3 - Education	70-100	33.3	4.5	31.0	8.4	19.1	9.6	14.9	11.8	15.5	25.7
	Total	37.6	5.6	30.8	11.3	19.6	11.7	16.7	16.1	16.0	23.9
	1 - inf bac	33.2	5.3	34.8	8.4	21.3	8.0	18.7	8.3	14.8	15.1
4 - Revenus	2 - bac	42.3	4.7	33.5	9.3	21.4	9.9	17.5	14.0	19.0	21.3
	3 - sup bac	37.5	6.1	27.0	13.9	17.5	15.0	15.0	22.0	15.2	30.8

Cas : collecte automatique de données

Twitter et l'API universitaire

```
Entrée [1]: import json
import pandas as pd
from searchtweets import ResultStream, gen_rule_payload, load_credentials,collect_results

Authentification

Entrée [2]: creds = load_credentials(filename='./credentials.yaml',
                                     yaml_key='search_tweets_api',
                                     env_overwrite=False)
             Grabbing bearer token from OAUTH

Requête

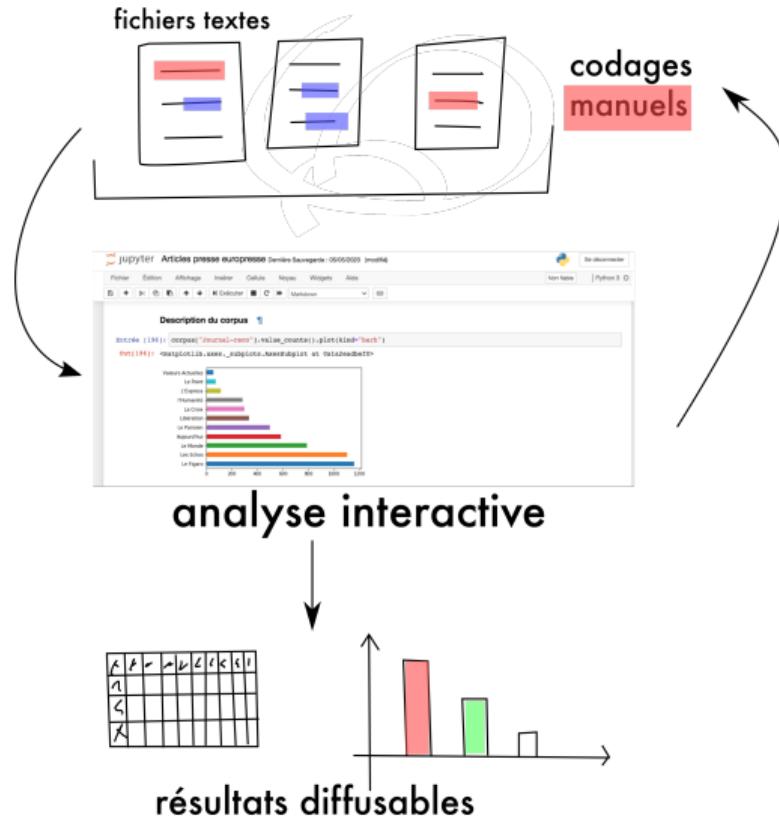
Entrée [3]: rule = gen_rule_payload("ANR lang:fr", results_per_call=50,
                                    from_date="201101210000",
                                    to_date="201102210000")
print(rule)
tweets = collect_results(rule,
                         max_results=1000,
                         result_stream_args=creds)
{"query": "ANR lang:fr", "maxResults": 50, "toDate": "201102210000", "fromDate": "201101210000"}

Entrée [4]: print(len(tweets))
pd.DataFrame([(i.created_at_datetim... for i in tweets))

136

Out[4]:
0 2011-02-20 18:21:50  'ANR Estée Lauder Advanced Night Repair sérum ...
1 2011-02-19 10:53:33 Recherches Partenariales et Innovation Biomédi...
2 2011-02-19 11:38:04 L'ANR propose une boîte à idées pour préparer ...
3 2011-02-18 10:28:41 A lire RT @CollectifPAPER La Cour des Comptes...
4 2011-02-18 10:26:09 La Cours des Comptes rappelle à l'ordre l'ANR ...
...
131 2011-01-25 07:52:30 Chaires d'excellence de l'ANR: accueil des che...
```

Cas : codage de matériel qualitatif



Outils dédiés facilement interfaçable



doccano

code quality doccano CI passing

doccano is an open source text annotation tool for humans. It provides annotation features for text classification, sequence labeling and sequence to sequence tasks. So, you can create labeled data for sentiment analysis, named entity recognition, text summarization and so on. Just create a project, upload data and start annotating. You can build a dataset in hours.

Demo

You can try the [annotation demo](#).

A screenshot of a web browser displaying the doccano annotation interface. The page title is "Annotations - doccano". The main content area shows a text document about Donald John Trump. Several words in the text are highlighted with colored boxes and underlined, indicating they have been annotated. To the right of the text is a table titled "PROJECTS" with one row. The table has two columns: "Key" and "Value".

Key	Value
id	4840772
Born	1946
Political party	Republican
Spouse	Melania Knauss
Parents	Fred Trump, Mary Anne MacLeod

Cas : diffuser ses outils à la communauté

The screenshot shows a project page for "pyshs 0.1.12". The top navigation bar includes a logo, a search bar labeled "Search projects", and links for "Help", "Sponsors", "Log in", and "Register". The main title "pyshs 0.1.12" is displayed with a green "Latest version" badge. Below the title, there's a "pip install pyshs" button and a release date "Released: Aug 8, 2021". The page content includes a brief description: "Module PySHS - Faciliter le traitement statistique en SHS". On the left, a sidebar titled "Navigation" lists "Project description" (selected), "Release history", and "Download files". Below that is a "Project links" section with a "Homepage" link. At the bottom, there's a "Statistics" section with links to "Libraries.io" and "Google BigQuery". The main content area is titled "Project description" and contains a section titled "Bibliothèque PySHS" with a detailed description of the library's purpose and a note about its Python alternative. It also mentions the current version (0.1.8). The "Contenu" section lists several features under "Traiter des données d'enquête par questionnaire".

Module PySHS - Faciliter le traitement statistique en SHS

Navigation

Project description

Release history

Download files

Project links

Homepage

Statistics

View statistics for this project via [Libraries.io](#), or by using our public [dataset on Google BigQuery](#)

Project description

Bibliothèque PySHS

La bibliothèque PySHS a pour but de réunir des outils utiles à un public de praticiens des sciences humaines et sociales francophones pour traiter des données. Elle a pour but de s'enrichir progressivement pour permettre à Python de devenir une alternative (réaliste) à R avec des fonctions facilement utilisable sur les opérations habituelles.

La version actuelle est la 0.1.8

Contenu

Traiter des données d'enquête par questionnaire

- Description d'un tableau de données
- Tri à plat et tableau croisé avec pondération
- Tableau croisant une variable dépendante avec une série de variables indépendantes, avec pondération
- Wrapper pour la régression logistique binomiale pondérée

Des applications qui se multiplient

Sociological Methods & Research

Impact Factor: 4.677 / 5-Year Impact Factor: 5.424 JOURNAL HOMEPAGE

Restricted access | Research article | First published online December 4, 2022

The Augmented Social Scientist: Using Sequential Transfer Learning to Annotate Millions of Texts with Human-Level Accuracy

Salomé Do, Étienne Ollion, and Rubing Shen | View all authors and affiliations

OnlineFirst | <https://doi.org/10.1177/00491241221134526>

Contents | Get access | Cite article | Share options | Information, rights and permissions | Metrics and citations

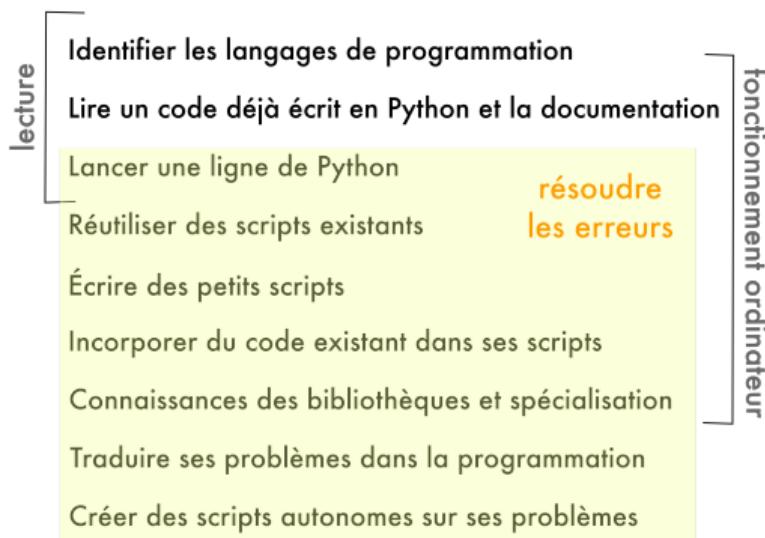
Abstract

The last decade witnessed a spectacular rise in the volume of available textual data. With this new abundance came the question of how to analyze it. In the social sciences, scholars mostly resorted to two well-established approaches, human annotation on sampled data on the one hand (either performed by the researcher, or outsourced to microworkers), and quantitative methods on the other. Each approach has its own merits - a potentially very fine-grained analysis for the former, a very scalable one for the latter - but the combination of these two properties has not yielded highly accurate results so far. Leveraging recent advances in sequential transfer learning, we demonstrate via an experiment that an expert can train a precise, efficient automatic classifier in a very limited amount of time. We also show that, under certain conditions, expert-trained models produce better annotations than humans themselves. We demonstrate these points using a classic research question in the sociology of journalism, the rise of a "horse race" coverage of politics. We conclude that recent advances in transfer learning help us augment ourselves when analyzing unstructured data.

5. S'y mettre !

Apprendre à programmer : pas une définition univoque

Découvre la programmation



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 ≠ Tout savoir

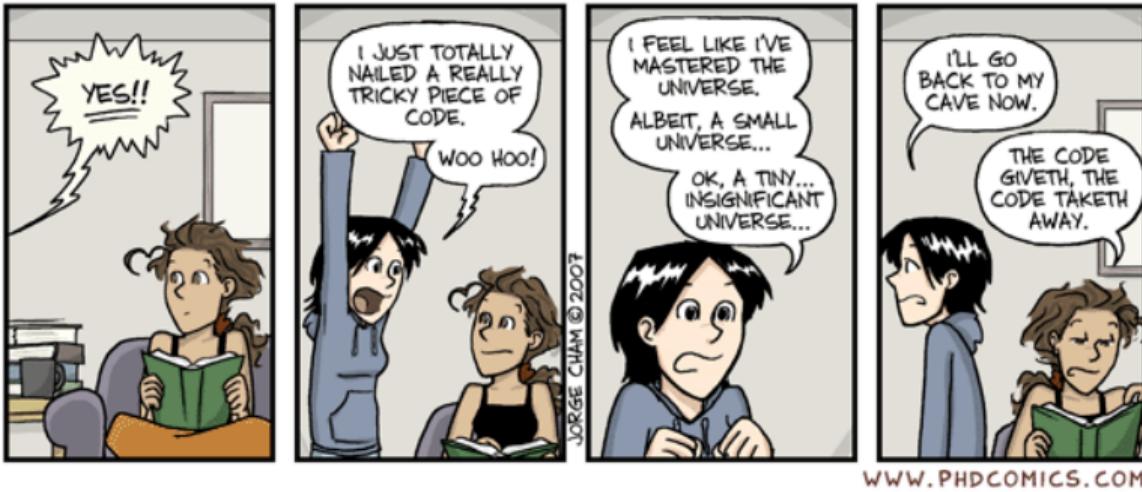
Apprendre à programmer signifie apprendre à potentiellement pouvoir utiliser de nombreux outils développés par des chercheurs.

Mais chaque domaine a ses savoirs spécifiques : *machine learning*, analyse de réseaux, textométrie, ... Il faut aussi assurer les savoirs théoriques associés à chaque outil.

La frontière peut être difficile à tracer.

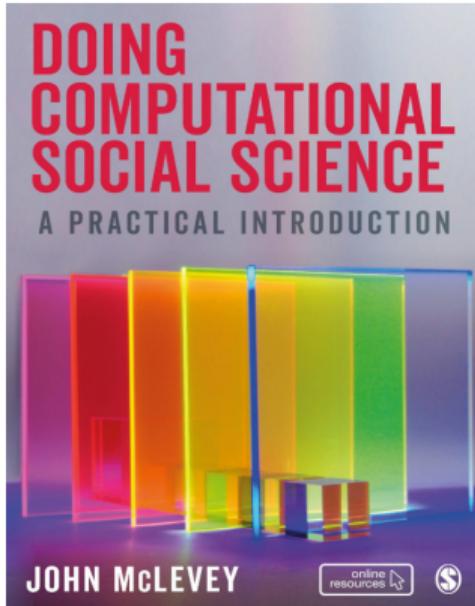
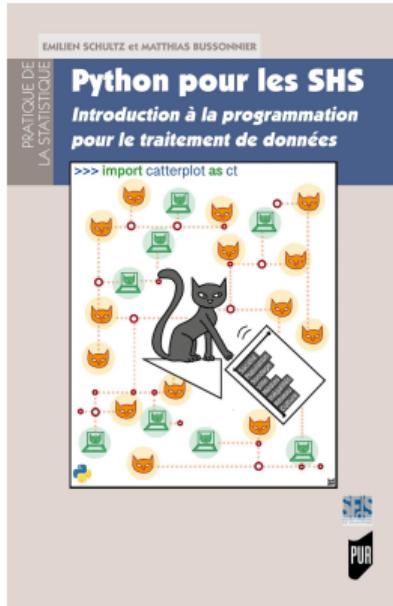
- ▶ Réutilisation d'outils facilité
- ▶ Mais cela ne replace pas une connaissance experte

Important de valoriser les petites victoires



Ressources pour être autonome

Prolifération de manuels et d'exemples : comment choisir ?

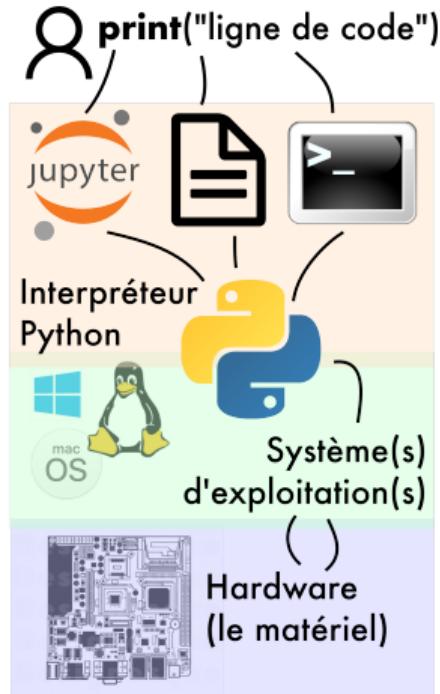


- ▶ Une liste de ressources : <https://github.com/pyshs/ressources-pyshs>
- ▶ Le cours de Melanie Walsh (EN) : <https://melaniewalsh.github.io/Intro-Cultural-Analytics/welcome.html>

Où écrire son code

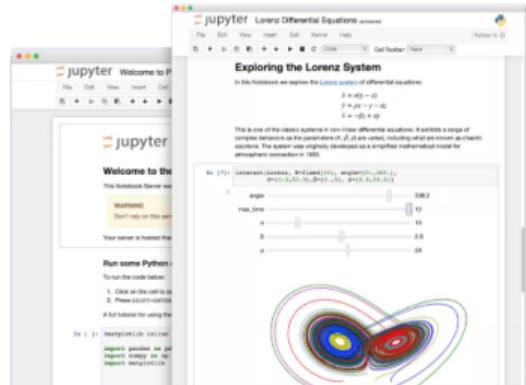
Trois manières d'exécuter un script :

- ▶ Dans un fichier texte (+ Integrated (I)DE)
- ▶ Dans le "logiciel" Python (interactivité)
- ▶ Dans un Notebook (Interactive (I)DE)



Notre choix : le Notebook Jupyter

Une philosophie générale : la programmation lettrée (*literate computing*).



Jupyter Notebook: The Classic Notebook Interface

The Jupyter Notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

[Try it in your browser](#)

[Install the Notebook](#)



Language of choice

Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala.



Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the [Jupyter Notebook Viewer](#).



Interactive output

Your code can produce rich, interactive output: HTML, images, videos, LaTeX, and custom MIME types.



Big data integration

Leverage big data tools, such as Apache Spark, from Python, R, and Scala. Explore that same data with pandas, scikit-learn, ggplot2, and TensorFlow.

IDE : du Notebook au Lab à autre

- ▶ I pour interactive
- ▶ I pour integrated

The screenshot shows the official Visual Studio Code website. At the top, there's a dark navigation bar with the Visual Studio Code logo, "Visual Studio Code", and links for "Docs", "Updates", "Blog", "API", "Extensions", "FAQ", and "Learn". To the right is a search icon. Below the bar, a banner announces "Version 1.78 is now available! Read about the new features and fixes from April." The main content area has a dark background. On the left, there's a sidebar with links: "Overview", "SETUP", "GET STARTED", "USER GUIDE", "SOURCE CONTROL", "TERMINAL", "LANGUAGES", and "NODEJS /". The main title "Jupyter Notebooks in VS Code" is centered above a detailed description of what Jupyter Notebooks are and how VS Code supports them. A list of features follows.

Jupyter Notebooks in VS Code

[Jupyter](#) (formerly IPython Notebook) is an open-source project that lets you easily combine Markdown text and executable Python source code on one canvas called a **notebook**. Visual Studio Code supports working with Jupyter Notebooks natively, and through [Python code files](#). This topic covers the native support available for Jupyter Notebooks and demonstrates how to:

- Create, open, and save Jupyter Notebooks
- Work with Jupyter code cells
- View, inspect, and filter variables using the Variable Explorer and Data Viewer
- Connect to a remote Jupyter server

Overview

SETUP

GET STARTED

USER GUIDE

SOURCE CONTROL

TERMINAL

LANGUAGES

NODEJS /

A avoir en tête

- ▶ Des avantages
 - ▶ Ludique et interactif
 - ▶ Avoir tous les éléments au même endroit
 - ▶ Partager son script
 - ▶ Très utilisé : "Ten computer codes that transformed science" (Nature, 2021)
<https://www.nature.com/articles/d41586-021-00075-2>
- ▶ Quelques limites
 - ▶ Orde d'exécution des cellules
 - ▶ Vite confus

Si vous voulez des critiques : I don't like notebooks.- Joel Grus

<https://www.youtube.com/watch?v=7jiPeIFXb6U>

Un environnement intégré



Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

Anaconda Distribution

Download

For Mac OS

Python 3.9 • 64-Bit Graphical Installer • 515 MB

Get Additional Installers



Open Source

Access the open-source software you need for projects in any field, from data visualization to robotics.



User-friendly

With our intuitive platform, you can easily search and install packages and create, load, and switch between environments.



Trusted

Our securely hosted packages and artifacts are methodically tested and regularly updated.

Exécutons un script déjà écrit !

Script "basique" qui demande une entrée textuelle et compte le nombre de mots de plus d'un certain nombre de lettres