

Programmer en Python pour les SHS ?

Quoi ? Comment ? Pourquoi ?

Émilien Schultz

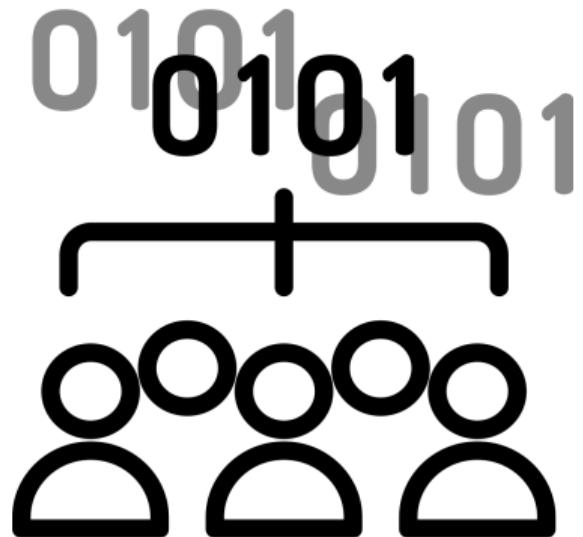
emilien.schultz@sciencespo.fr

médialab - SESSTIM

Pourquoi vous êtes là ?

Avant tout : répondre à 3 questions

1. Pourquoi programmer (en recherche) ?
2. Pourquoi Python ?
3. Pourquoi penser les usages spécifiques aux SHS ?



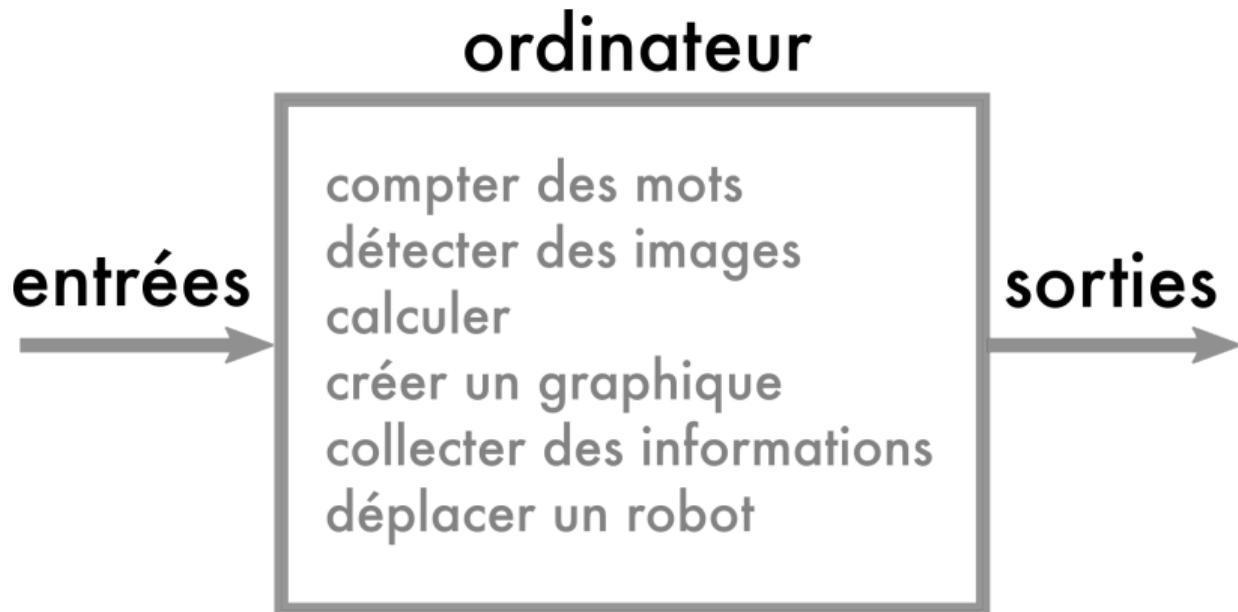
Pourquoi programmer ?

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.



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

Par rapport à un logiciel, programmer :

- ▶ formaliser des manipulations pour les partager
- ▶ adapter à des tâches non prévues par les logiciels
- ▶ interconnecter des opérations sinon séparées



JANE-CLARK.TUMBLR

Programmer ≠ Construire un logiciel

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

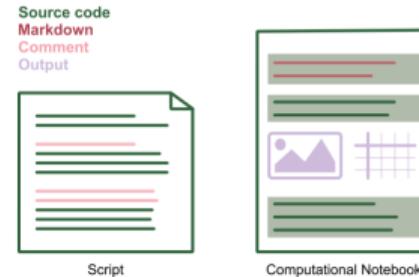
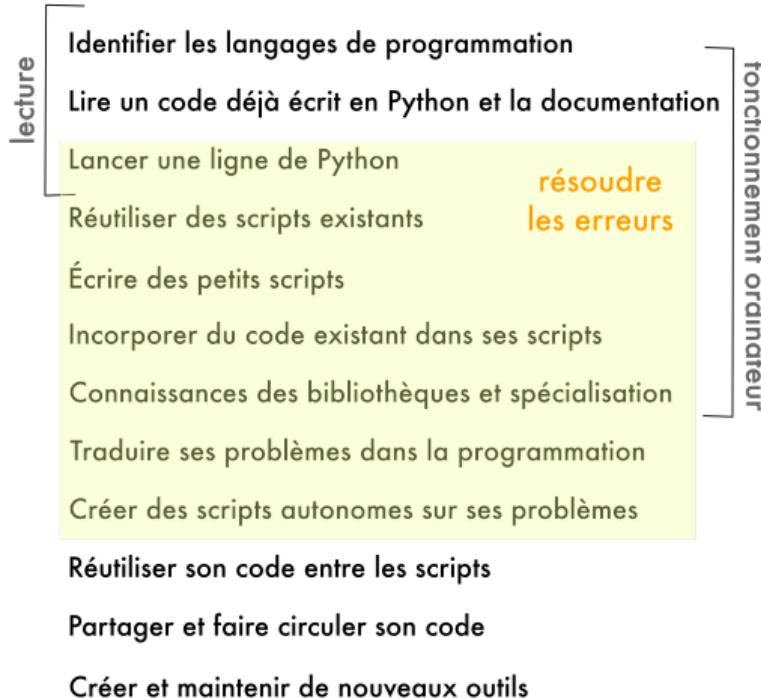


Fig. 1. Current scripting language IDEs support writing and executing code via two programming modalities: scripts (left) and computational notebooks (right). In this paper we investigate how these modalities are used in data

En pratique : une diversité de niveaux de compétences

Découvre la programmation



Contributeur • rice Open Source accompli • e

Aussi : programmer comme point d'entrée

Un effet **oignon** :

- ▶ Penser la structures des données et leurs diversité
 - ▶ Format de fichier : csv ou xls ? Passage vers du relationnel ?
- ▶ Penser la matérialité de nos pratiques
 - ▶ Stockage mémoire vive, cloud ou disque dur ?
- ▶ Possibilité d'échanger avec les collaborateurs ressources
 - ▶ Une langue commune entre spécialités

Un exemple : découvrir qu'une image est en fait un tableau de points, chaque point décrit par trois valeurs (rouge, vert, bleu), et qu'on peut manip

2. Pourquoi Python ?

Un monde de langages

Liste de langages de programmation

À 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]

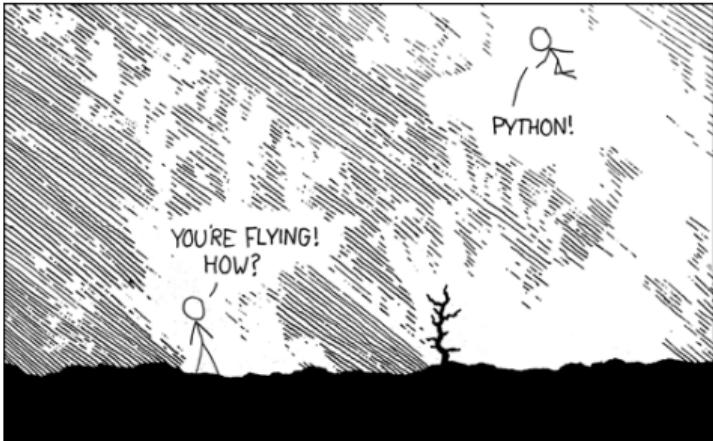
- A+ • ABSYS
- A++ • ALI
- A# .NET • Abundance
- A# (Axiom) (en)
- A-0 System
- ABAL
- ABAL++
- ABAP
- ABC
- ABCI/I
- ABCI/C+
- ABCI/R
- ABCI/R2
- Abel
- ABSET (en)
- ABSYS
- ALI
- Abundance
- ACC (programming language) (en)
- Accent
- ActForex
- Ace DASL
- ACT-III
- Ada
- Adenine
- Afnix
- Agora (programming 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
- Aribertion
- Aribase (langage)
- Assembleur
- ASP.NET
- ATS
- AUPL
- AutoHotkey
- AutoIt
- Averest
- awk
- axe parser
- Axum (programming language) (en)
- APL

B [modifier | modifier le code]

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

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

Tout est possible avec Python (sur un ordinateur)



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

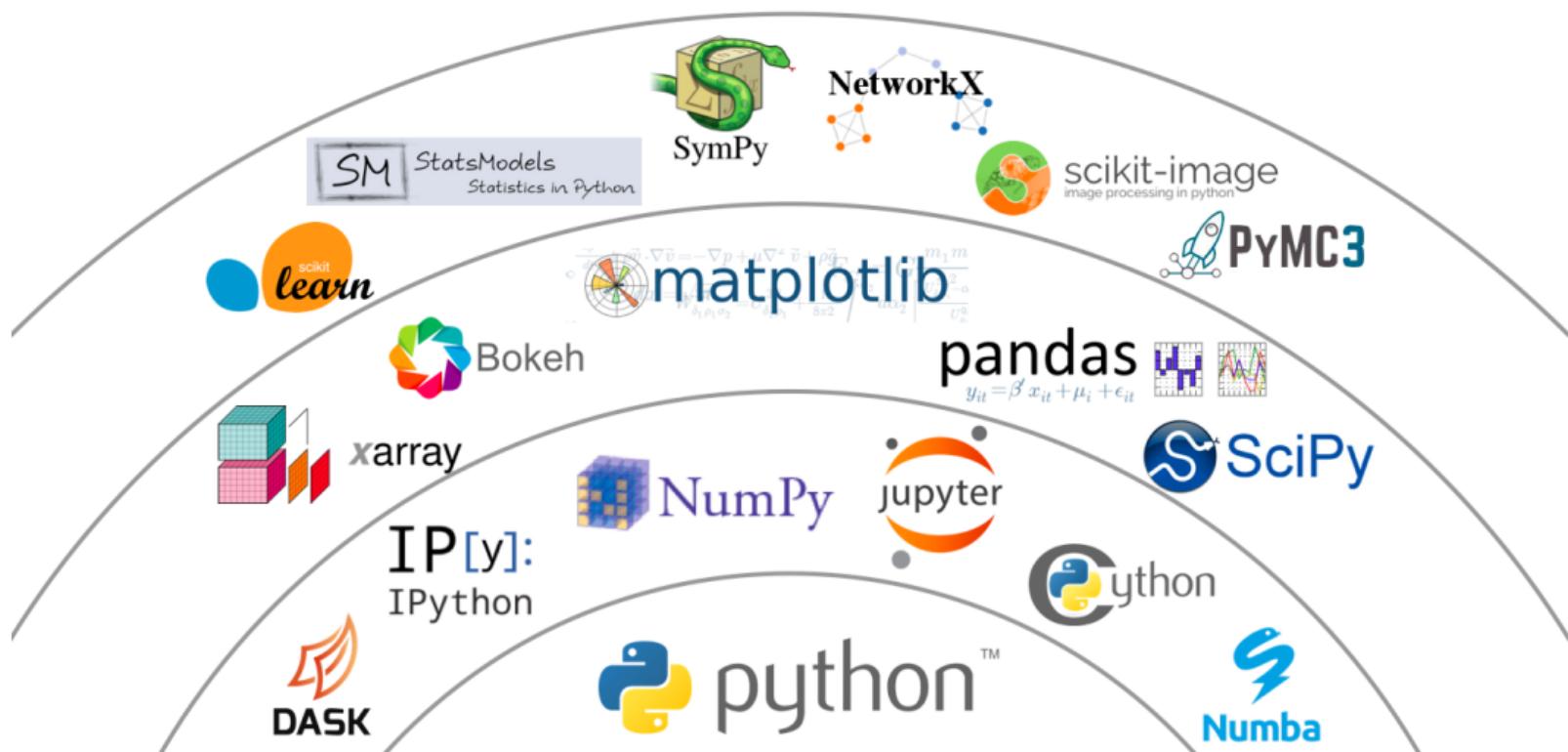
Propriétés de Python

- ▶ 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

Plus qu'un langage : un univers d'outils

Python's Scientific Stack

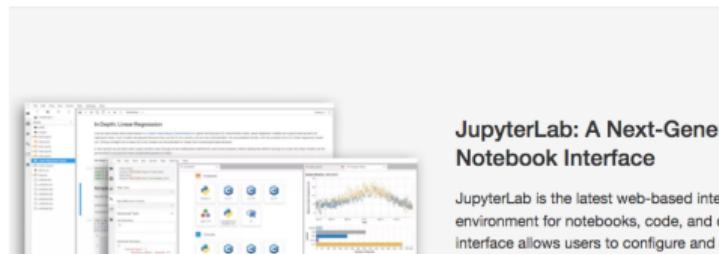
Jake Vanderplas PyCon 2017 Keynote



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

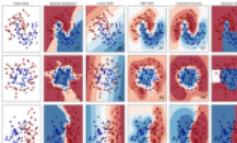
Getting Started Release Highlights for 1.0 GitHub

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

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

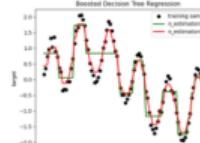


Regression

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

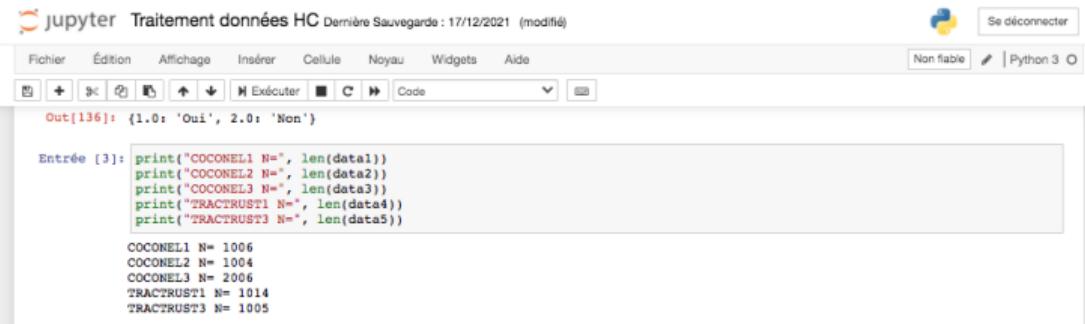
Industrial-Strength Natural Language Processing

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.

Permettant de construire des workflows complets



The screenshot shows a Jupyter Notebook interface. At the top, there's a toolbar with File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a Python 3 tab. Below the toolbar, there's a menu bar with Fichier, Edition, Affichage, Insérer, Cellule, Noyau, Widgets, Aide. On the right side, there are buttons for Non fiable, Se déconnecter, and Python 3. The main area shows code execution results:

```
Out[136]: {1.0: 'Oui', 2.0: 'Non'}
```

```
Entrée [3]: print("COCONEL1 N=", len(data1))
print("COCONEL2 N=", len(data2))
print("COCONEL3 N=", len(data3))
print("TRACTRUST1 N=", len(data4))
print("TRACTRUST3 N=", len(data5))

COCONEL1 N= 1006
COCONEL2 N= 1004
COCONEL3 N= 2006
TRACTRUST1 N= 1014
TRACTRUST3 N= 1005
```

[FIGURE 1] Evolution de l'attitude en France



```
Entrée [9]: # Tableau par enquête
d = {'04-07-2020': pyhs.tri_a_plat(data1,"HC_c","RED")["Pourcentage (%)"],
      '04-19-2020': pyhs.tri_a_plat(data1,"HC_c","RED")["Pourcentage (%)"],
      '06-23-2020': pyhs.tri_a_plat(data1,"HC_c","RED")["Pourcentage (%)"],
      '11-03-2020': pyhs.tri_a_plat(data1,"HC_c","RED")["Pourcentage (%)"],
      '06-08-2021': pyhs.tri_a_plat(data1,"HC_c","RED")["Pourcentage (%)"]
t = pd.concat(d, axis=1).drop("Total").T

# Données Google Trends
hc = pd.read_csv("./multiTimeline.csv").replace({'\xa0':0})
hc["chloroquine: (France)"] = hc["chloroquine: (France)"].apply(int)
hc["hydroxychloroquine: (France)"] = hc["hydroxychloroquine: (France)"].apply(int)
hc["Semaine"] = pd.to_datetime(hc["Semaine"])
hc = hc.set_index("Semaine")["chloroquine: (France)"]

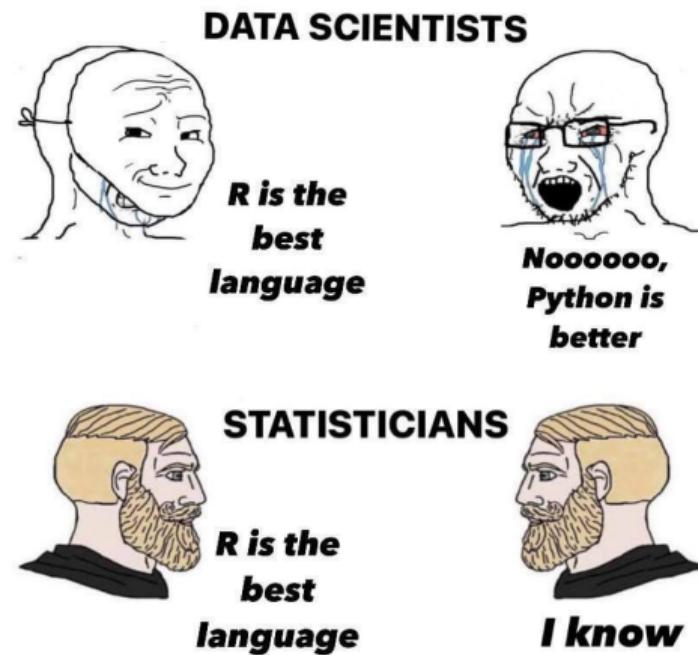
# Graphique
t.index = pd.to_datetime(t.index)
ax = t.plot(color=['r','g','b'], figsize=(10,5),marker='o', linestyle='--')
pd.DataFrame(hc.resample('w').sum()).plot(ax=ax,color="gray")
plt.xlim("2020-02-01","2021-06-20")
plt.xlabel("Date (per week)")
plt.ylabel("Percentage (%)")
plt.legend(["HC is effective","HC is ineffective","Uncertain","Intensity of Google searches using Google Trends"])
plt.title("Figure 1. Evolution of attitudes toward HC in France and media coverage between April 2020 and June 2021")

plt.tight_layout()
plt.savefig("../figures/Figure 1 - evolution.png",dpi=1000)
```



Mais pas le seul choix...

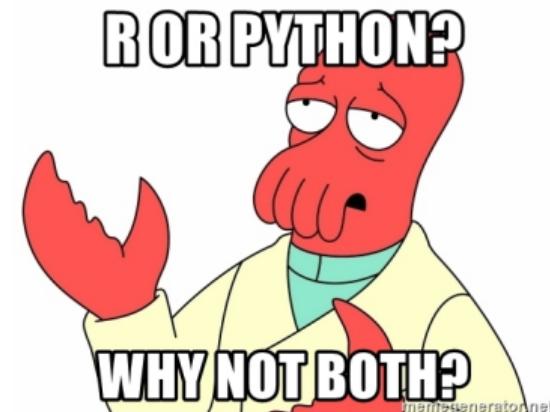
Convergence et divergences avec d'autres langages, R en premier lieu



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

Python ou R? Python et R? 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.

3. Pourquoi réfléchir les usages spécifiques aux SHS ?

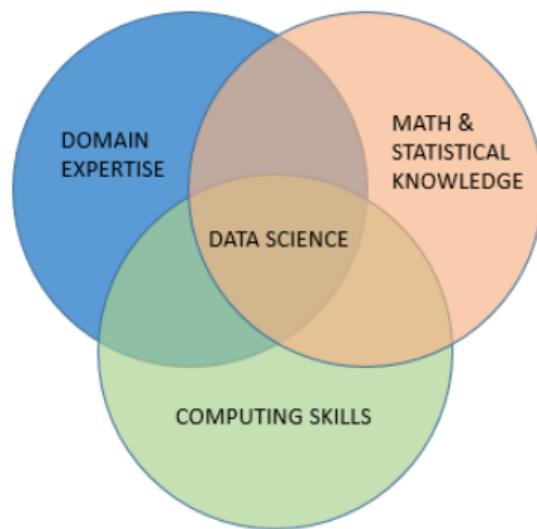
Des identités en transformation autour du numérique

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

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
- ▶ Constat d'une limite des exemples disponibles : que faire ?
- ▶ 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.)
- ▶ Division du travail vs. culture partagée

Un gros potentiel d'interface entre les pratiques. Mais un état des lieux en cours.

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 -**
Raoult : les dessous d'une rencontre surprise Cet psychologue de formation, ses 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

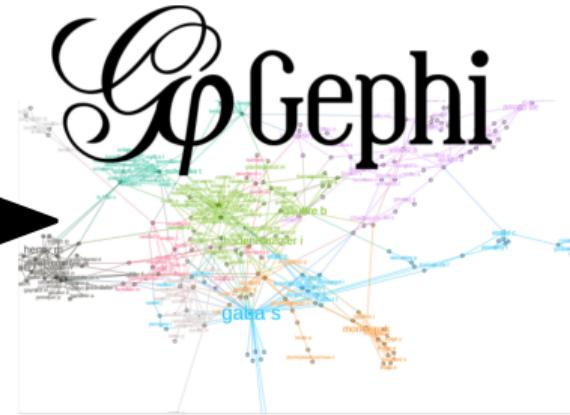
Cas : construire un réseau

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

A	B	C	D	E
ID	ANNEE	AUTEURS	TITRE	JOURNAL
2	35	1996 LEROUX A., BRETAGNOLLE V	Sex ratio variations in broods of Montagu's harrier	Journal of Avian Biology
3	37	1996 A RECORDER	SALAMOLARD M., BRETAGNOLLE V.	
4	44	1998 ARROYO B.E., LEROUX A.B.A.	Egg and clutch	Journal of Reproduction and Development
5	47	1998 de CORNUNIER T., BERNARD R.	Nidification of Redstarts	
6	52	1999 ARROYO B.E., BRETAGNOLLE V.	Breeding biology	Journal of Reproduction and Development
7	55	1999 SALAMOLARD M., MOREAU C.	Habitat selection	Bird Study
8	58	2000 AMAR A., ARROYO B.E., BRETAGNOLLE V.	Post-fledging Ibis	Ibis
9	59	2000 ARROYO B.E., DECORNULIER T.	Sex and age of Condors	Condor
10	62	2000 GUILLEMAIN M., HOUTTE S., FRIJ	Activities and Revue d'Ecologie	Ecology
11	63	2000 JIGUET F., ARROYO B., BRETAGNOLLE V.	Lek mating systems	Behavioural Ecology
12	68	2000 SALAMOLARD M., BUTET A.	Responses of Ecology	Ecology
13	69	2001 ARROYO B., MOUGEOU F., BRETAGNOLLE V.	Colonial birds	Behavioral Ecology
14	70	2001 CLERE E., BRETAGNOLLE V.	Disponibilité	Revue d'Ecologie
15	71	2001 JIGUET F., BRETAGNOLLE V.	Courtship behaviour	Behavioural Ecology

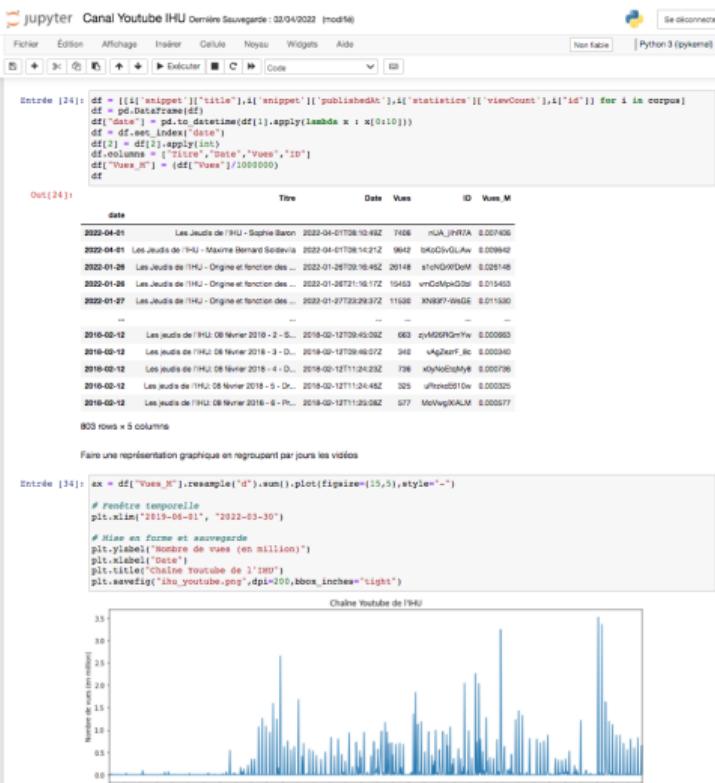


```
<?xml version="1.0" encoding="utf-8"?>
graph TD
    subgraph "http://graphml.graphdrawing.org/mlns"
        min["min<br>http://graphml.graphdrawing.org/mlns"]
        max["max<br>http://graphml.graphdrawing.org/mlns"]
        title["title<br>http://graphml.graphdrawing.org/mlns"]
        id["id<br>http://graphml.graphdrawing.org/mlns"]
        type["type<br>http://graphml.graphdrawing.org/mlns"]
        name["name<br>http://graphml.graphdrawing.org/mlns"]
        cluster["cluster<br>http://graphml.graphdrawing.org/mlns"]
        string["string<br>http://graphml.graphdrawing.org/mlns"]
        long["long<br>http://graphml.graphdrawing.org/mlns"]
        label["label<br>http://graphml.graphdrawing.org/mlns"]
    end
    graph TD
        node0[Sex ratio variations in broods of Montagu's harrier] --- node1[1996]
        node1 --- node2[Article]
        node2 --- node3[5c]
        node3 --- node4[Sex ratio]
        node4 --- node5[Sex ratio]
    
```



Cas : exploration de données de l'API aux statistiques

Exploration d'un tableau de données (ici le nombre de vues par vidéos de la chaîne Youtube de l'IHU)



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

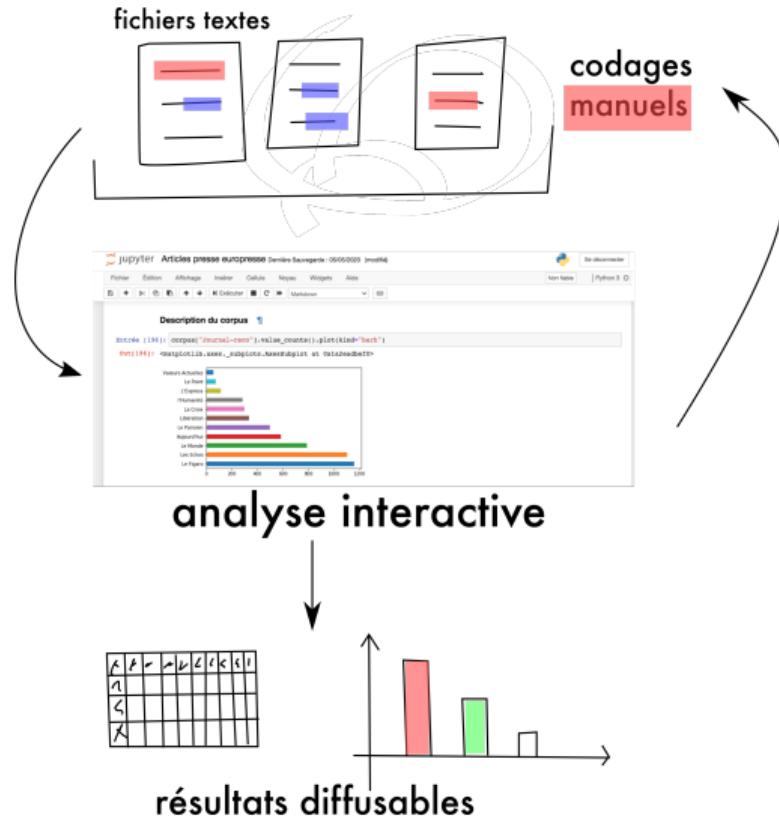
Entrée [3]: rule = gen_rule_payload("ANR lang:fr", results_per_call=50,
                                    from_date="201101210000",
                                    to_date="201102210000")
print(rule)
tweets = collect_results(rule,
                         max_results=1000,
                         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 : toponyme et cartographie

A partir d'un texte, identifier les lieux géographies et produire des cartes (potentiellement interactives)



Cartes interactives

A partir d'outils de cartographies, nous avons obtenu une visualisation des noms géographiques cités dans les romans, ce qui permet de se faire une idée des zones du monde qui faisaient partie de l'univers des enfants français sous la troisième République. Deux titres, Petite-Pierre ou le bon cultivateur, Maurice ou le travail, sont antérieurs à la troisième République mais sont encore présents dans les listes de manuels de la troisième République.

Répartition géographique des lieux cités dans le corpus

Cliquer sur une carte pour afficher la version pleine page interactive

[Carte de chaleur reprenant tous les lieux cités dans l'ensemble du corpus de romans scolaires](#)



<https://baoia.huma-num.fr/contact/tutoriel-complet-de-lextraction-documentaire-a-la-cartographie/>

Cas : figures d'un article faciles à reproduire

Production des statistiques et des figures facile à relancer en cas de révision de l'article.

Open Access Article

French Public Familiarity and Attitudes toward Clinical Research during the COVID-19 Pandemic

by  Émilien Schultz 1,2,*  Jeremy K. Ward 3,4  Laëtitia Atlani-Duault 1,5,6  Seth M. Holmes 2,7,8 and  Julien Mancini 2,9

¹ CEPED (UMR 196), Université de Paris, IRD, 75006 Paris, France
² SESSTIM, Sciences Économiques & Sociales de la Santé & Traitement de l'Information Médicale, CANBIOS Team (Équipe Labelisée LIGUE 2019), Aix-Marseille University, INSERM, IRD, 13009 Marseille, France
³ CERMES3, INSERM, CNRS, EHESS, Université de Paris, 94801 Villejuif, France
⁴ VITROME, Aix-Marseille University, IRD, AP-HM, SSA, 13005 Marseille, France
⁵ Institut COVID-19 Add Memoriam, University of Paris, 75006 Paris, France
⁶ WHO Collaborative Center for Research on Health and Humanitarian Policies and Practices, IRD, Université de Paris, 75006 Paris, France
⁷ Society and Environment, Medical Anthropology, and Public Health, University of Berkeley, Berkeley, CA 94720, USA
⁸ Mediterranean Institute for Advanced Study IMéRA, Institut Paoli Calmettes, Aix-Marseille University, 13004 Marseille, France
⁹ BioSTIC, APHM, Timone, 13005 Marseille, France

* Author to whom correspondence should be addressed.
† Current address: CEPED, 45 Rue des Saints-Pères, 75006 Paris, France.

Academic Editor: Roy McConkey

Int. J. Environ. Res. Public Health **2021**, *18*(5), 2611; <https://doi.org/10.3390/ijerph18052611>

Received: 2 February 2021 / Revised: 2 March 2021 / Accepted: 2 March 2021 / Published: 5 March 2021

(This article belongs to the Section Global Health)

[View Full-Text](#) [Download PDF](#) [Browse Figures](#) [Citation Export](#)

Abstract

The COVID-19 pandemic put clinical research in the media spotlight globally. This article proposes a first measure of familiarity with and attitude toward clinical research in France. Drawing from the “Health Literacy Survey 2019” (HLS19) conducted online between 27 May and 5 June 2020 on a sample of the French adult population ($N = 1003$), we show that a significant proportion of the French population claimed some familiarity with clinical trials (64.8%) and had positive attitudes (72%) toward them. One of the important findings of this study is that positive attitudes toward clinical research exist side by side with a strong distancing from the pharmaceutical industry. While respondents acknowledged that the pharmaceutical industry plays an important role in clinical

Cas : diffuser ses outils à la communauté

The screenshot shows a project page for "pyshs 0.1.12". The top navigation bar includes a search bar, "Help", "Sponsors", "Log in", and "Register" buttons. 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. A release date "Released: Aug 8, 2021" is also shown. The page content includes a "Project description" section with a "Bibliothèque PySHS" heading, a "Statistics" section with links to "Libraries.io" and "Google BigQuery", and a "Contenu" section with a "Traiter des données d'enquête par questionnaire" heading and a bulleted list of features.

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

Autres usages

- ▶ Garder une mémoire de ses traitements.
- ▶ Collaboration autour des données : partager son code, faire relire ses résultats intermédiaires
- ▶ Traitement massif de données : parallélisation, déploiement sur des grandes infrastructures, recours aux outils du machine learning
- ▶ Créer une interface utilisateur pour accéder à vos données.
- ▶ Traitement des images.

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.

Importance d'avoir des exemples

<https://gitlab.huma-num.fr/io>

5. S'y mettre !

Aujourd'hui

- ▶ Base du langage
- ▶ Bibliothèques
- ▶ Focus sur Pandas : statistiques et visualisation
- ▶ Exemple complet de traitement d'une enquête
- ▶ Discussion sur vos usages

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

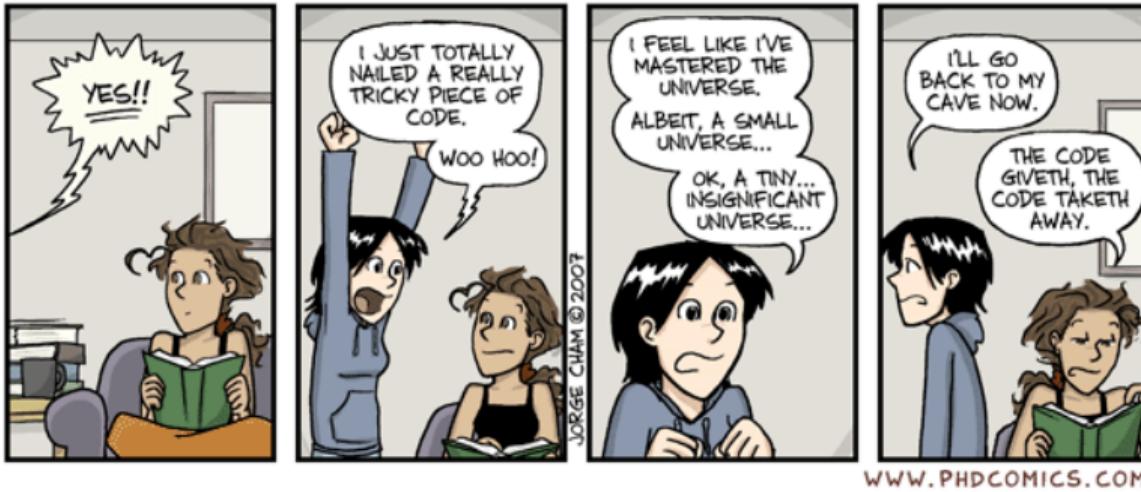
Apprendre à programmer signifie apprendre à potentiellement pouvoir utiliser de nombreux outils développés par des chercheurs.

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

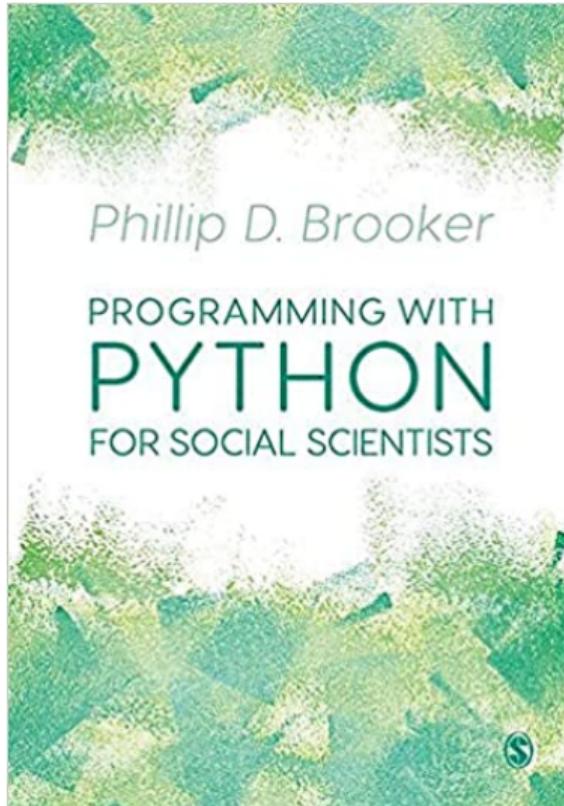
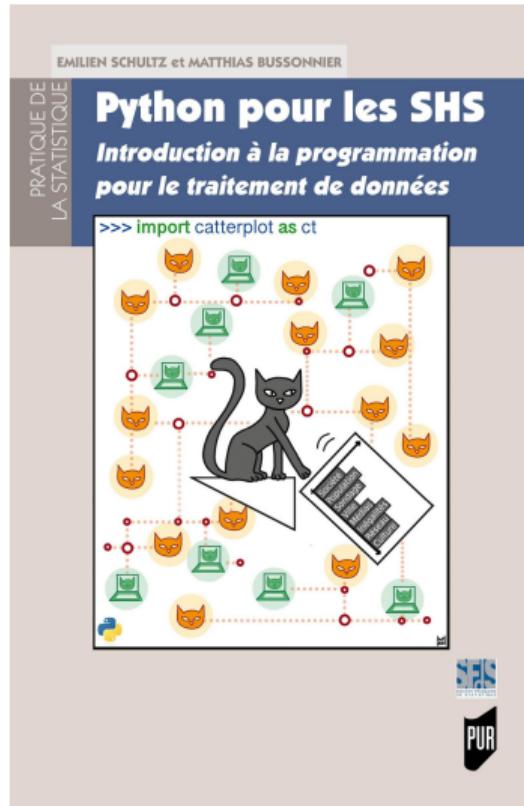
La frontière peut être difficile à tracer.

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

Important de valoriser les petites victoires



Ressources



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

Des espaces collectifs à construire

[https://www.canal-u.tv/chaines/callisto/
les-coulisses-du-code-python-pour-les-shs-cocopyshs/](https://www.canal-u.tv/chaines/callisto/les-coulisses-du-code-python-pour-les-shs-cocopyshs/)



#CocoPySHS

URFIST Lyon

échanger autour de nos pratiques de **programmation en Python** pour les **SHS**

partager nos **expériences**

favoriser la **reproductibilité**

développer de **bonnes pratiques d'ouverture du code**

*Un jeudi par mois, de 13h à 14h30
(en visioconférence)*

17 mars 2022 - Fouille de texte & Ingrédients alimentaires avec Tristan Salord

7 avril 2022 - Données de questionnaire & Statistiques avec Mariannig Le Béchec et Emilien Schultz

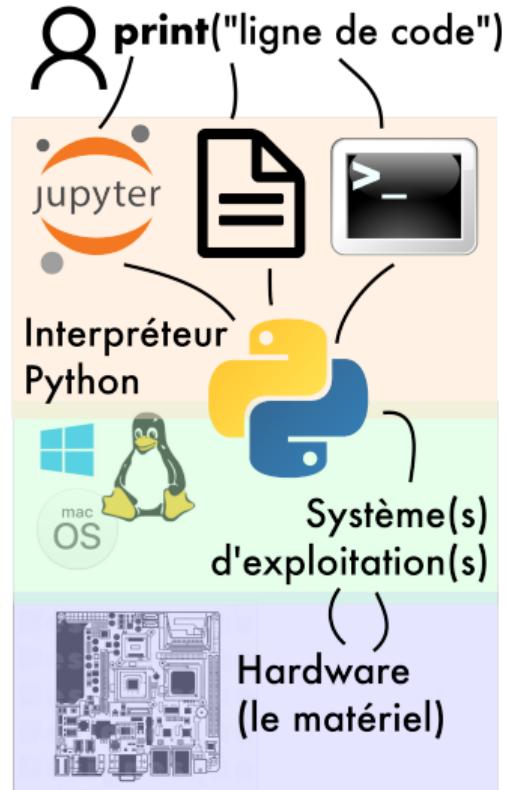
12 mai 2022 - Collecte & Nettoyage de données avec Lucie Loubère

9 juin 2022 - Approches cartographiques et démarche de science ouverte avec Célyna Gruson-Daniel, Maya Anderson-Gonzalez et Camille Moulin

7 juillet 2022 - Collecter des données Twitter & Ethnographies numériques avec Léo Mignot

Les COulisses du COde Python pour les SHS

Trois manières d'exécuter un script



Notre choix



Products

Pricing

Solutions

Resources

Partners

Blog

Company

Contact Sales

Individual Edition is now

ANAconda DISTRIBUTION

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

Anaconda Distribution

Download

For MacOS

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.