Les carnets numériques interactifs en Sciences humaines et sociales : l'exemple des carnets Jupyter

Maxime Popineau, Émilien Schultz, Marie-Laure Massot, Agnès Tricoche

Les carnets numériques interactifs en Sciences humaines et sociales : l'exemple des carnets Jupyter

Maxime POPINEAU

Rapport réalisé en stage de master 2,
co-encadré par Marie-Laure Massot, Emilien Schultz et Agnès Tricoche
dans le cadre de l'initiative Digit_Hum, d'avril-juillet 2023[1]

Mots clés : programmation, carnet numériques interactifs, Jupyter Notebooks, Sciences humaines et sociales (SHS)

Introduction

En 2014, le chercheur en physique Fernando Pérez annonce un projet communautaire issu de son environnement interactif iPython facilitant la programmation pour les scientifiques : le projet Jupyter. Ce dernier se développe autour d'un écosystème de standards, de logiciels libres et de services pour le calcul interactif : Jupyter notebook, JupyterLab, JupyterHub, etc. Cette synthèse porte sur les carnets numériques en Sciences humaines et sociales (SHS) en prenant l'exemple des carnets Jupyter[2]. Il s'agit en effet d'un outil en pleine expansion à destination des acteurs de l'Enseignement supérieur et de la recherche.
Les carnets numériques permettent de combiner à la fois du code et du texte lu par les utilisateurs[3]. L'idée de document entremêlant narration et code pour s'adresser simultanément aux êtres humains et à l'ordinateur n'est pas nouvelle puisqu'elle a été explorée dès la fin des années 1970. Dans son ouvrage *The TeXBook*, publié en 1984, Donald Knuth propose une solution : la programmation lettrée. Le principe est de traiter le code comme un texte qui doit être compréhensible pour les lecteurs. La programmation lettrée prend la forme d'un texte en prose au sein duquel s'imbriquent des macros (une forme de code abrégé). La programmation lettrée a inspiré le calcul lettré qui est au cœur des carnets numériques.
Les carnets Jupyter ne sont pas les premiers carnets numériques[4]. Dès les années 1980, des logiciels comme Maple, Matlab et Wolfram Mathematica disposaient d'interfaces de ce type. Dans les années

[1] Ce stage a été réalisé dans le cadre de l'École universitaire de recherche Translitteræ (programme Investissements d'avenir ANR-10-IDEX-0001-02 PSL* et ANR-17-EURE-0025), avec le soutien financier du CAPHÉS (UMS 3610, CNRS-ENS) et d'AOROC (UMR 8546, CNRS-ENS-EPHE). Je tiens à remercier les personnes qui m'ont aidé à mener à bien ce rapport : mon tuteur de stage, Emilien Schultz, ainsi que Mathilde Fichen, Marie Jouble, Mathilde Nguyen et Nicolas M. Thiéry pour leurs relectures attentives et leurs suggestions d'amélioration. Merci également à Marie-Laure Massot et Agnès Tricoche pour avoir co-encadré ce stage dans le cadre de l'initiative Digit_Hum (https://digithum.huma-num.fr).
[2] https://fr.wikipedia.org/wiki/Jupyter.
[3] https://programminghistorian.org/fr/lecons/introduction-aux-carnets-jupyter-notebooks.
[4] https://tutoriels-jupyter.pages.in2p3.fr/introduction.html.

1990, les carnets numériques Mathematica apparaissent. En 2001, IPython est lancé et possède depuis le 21 avril 2012 une interface web : le carnet numérique interactif. En 2007, Sagemath carnet numérique est mis en place, suivi en 2013 par la création des « carnets Voyant » par Stéfan Sinclair et Geoffrey Rockwell. Vient ensuite le projet Jupyter en 2014 issu d'un découpage du projet Ipython en deux. Grâce à son interopérabilité et sa communauté, Jupyter s'est imposé comme solution principale de carnet numérique, sans être non plus exclusive (RStudio…). Les carnets Jupyter ne remplacent néanmoins pas la programmation lettrée ainsi que les logiciels cités précédemment.
Le logiciel Matlab et le consortium de la TEI (qui est une forme de programmation lettrée) sont en effet encore très utilisés aujourd'hui. Ainsi, de nos jours, beaucoup de carnets numériques co-existent : certains sont des services basés sur les carnets Jupyter comme Google Collab, tandis que d'autres sont des concurrents comme BeakerX, Apache Zeppelin et DeepNote.

1. Présentation, intérêts et spécificités des carnets Jupyter

Un carnet Jupyter[5] est un document manipulable dans le navigateur internet composé de plusieurs cellules, combinant à la fois du code, du texte et des visualisations graphiques, le code pouvant être modifié et exécuté interactivement. Ce type de document permet de faire de la narration, du calcul, de la visualisation, de l'interaction et de la programmation.
Les carnets Jupyter permettent d'interagir avec une centaine de langages et systèmes, dont Python, Julia et R (d'où le nom de Jupyter). Les carnets Jupyter et notamment le langage informatique Python[6] sont utilisés massivement dans de nombreux domaines scientifiques, dans le monde académique comme dans l'industrie, et commencent à l'être en SHS. Ce support est en mesure de faciliter la pratique computationnelle des chercheurs en SHS ; en favorisant un standard permettant d'homogénéiser les réflexions intégrant du code et de partager facilement leur recherche. Les carnets Jupyter peuvent avoir de nombreux avantages comme celui de faciliter une approche interdisciplinaire car il est facile de réutiliser un carnet numérique pour le modifier selon ses besoins. Un autre avantage des carnets numériques est leur aspect versatile : ils peuvent à ce titre être utilisés pour la recherche comme l'enseignement.
Bien qu'encore peu exploités dans le domaine des SHS, les carnets numériques ont gagné en popularité dans le contexte de l'éducation. En France, c'est actuellement l'un des outils les plus utilisés par les enseignants en informatique en raison de sa facilité pédagogique[7]. Les carnets Jupyter sont un outil d'apprentissage du code qui permet aux élèves de progresser étape par étape. Les cellules s'exécutent au fur et à mesure au sein d'une narration cohérente.

Les carnets Jupyter sont utilisés par exemple dans le projet Capytale[8], service web utilisant les carnets numériques pour créer et partager des activités pédagogiques de codage. Le site enregistre 500000 utilisateurs dont 100000 qui utilisent le site web de manière hebdomadaire. Les carnets Jupyter sont utilisés aussi par les plus grandes entreprises mondiales comme Google, Amazon et Netflix. Ce dernier est d'ailleurs l'un des principaux sponsors du colloque mondial autour des carnets Jupyter, la JupyterCon[9] dont la dernière édition s'est tenue à Paris en mai 2023. Dès lors, comment expliquer l'intérêt grandissant pour ce type de carnet numérique par rapport aux autres ? C'est à cette question que nous allons essayer de répondre en donnant des éléments de réponses au fil de cette synthèse.

[5] https://fr.wikipedia.org/wiki/Jupyter.
[6] Python est le langage de programmation le plus utilisé et le plus simple dans les carnets numériques. Python est utilisé dans l'enseignement pour permettre une initiation aux concepts de base de la programmation mais aussi dans le cadre de la recherche pour obtenir des réponses à des questionnements scientifiques.
[7] Christophe Casseau, Jean-Rémy Falleri, Xavier Blanc, Thomas Degueule. Immediate Feedback for Students to Solve Notebook Reproducibility Problems in the Classroom. 2021 IEEE Symposium on Visual Languages and HumanCentric Computing (VL/HCC), Oct 2021, Saint Louis, Missouri, United States. ⟨10.1109/VL/HCC51201.2021.9576363⟩. ⟨hal-03378094⟩.
[8] https://www.ac-paris.fr/capytale-un-service-web-pour-creer-et-partager-des-activites-pedagogiques-de-codage-121816.
[9] https://www.jupytercon.com/about.

Les carnets Jupyter sont disponibles dans un simple navigateur internet et ont des fonctionnalités spécifiques qui les rendent très intéressants dans un contexte de recherche et d'enseignement, telles que :

- **L'interactivité**. Les utilisateurs peuvent exécuter le code directement dans le carnet numérique ; ils peuvent voir le résultat rapidement et le modifier si besoin.

- **L'interopérabilité**. Ce type de carnet numérique supporte plus de 40 langages de programmation et le document n'est pas juste limité au code : il est possible de mettre des images, des visualisations, des formules mathématiques, des explications textuelles et même des vidéos.

- **La facilité du partage, de l'appropriation et de la réutilisation**. Les carnets Jupyter peuvent être exportés au format HTML, PDF ou Markdown. Cela est très pratique dans le cadre de la recherche collaborative mais aussi dans un contexte d'enseignement.

- **La richesse de l'écosystème Jupyter**. Les carnets Jupyter permettent d'interagir avec de nombreuses bibliothèques et la communauté est très active. Il est donc facile de trouver des ressources et de se faire aider pour résoudre des tâches spécifiques.

Voici également quelques exemples d'utilisations concrètes et variées de ce type de carnets en SHS :

- **En linguistique :** Les carnets numériques sont utiles pour la linguistique computationnelle. Le carnet réalisé par le projet « Traduction » du laboratoire de littérature à Stanford[10] propose une analyse sur la traduction d'un texte en différentes langues. L'analyse se porte sur le nombre de mots par traduction et leur divergence au niveau statistique.

- **En géographie :** Les carnets numériques améliorent le travail collaboratif en donnant plus de souplesse au traitement de données géolocalisées (par exemple en combinant GeoPandas et CartoPy dans une interface interactive reproductible).

- **En sociologie :** Les carnets numériques permettent d'explorer les données statistiques largement disponibles auprès des institutions ou des organismes de recherche participant à la science ouverte. Un carnet Jupyter[11] a été réalisé par Emilien SCHULTZ et Mathieu MOREY sur la mobilité professionnelle en partant des données de l'INSEE par exemple.

Dans le domaine des Sciences Humaines et Sociales (SHS), les enseignants, les étudiants et les chercheurs ont recours à une variété d'outils pour leurs besoins en calcul et en code. Parmi les logiciels couramment utilisés, on trouve SPSS, RStudio ainsi que des scripts utilisant des langages informatiques tels que Python, R et Javascript et des logiciels tels que Matlab. Les carnets Jupyter peuvent être utilisés en complément de ces outils existants pour faciliter la documentation, la reproductibilité des analyses et la collaboration entre chercheurs. Ils offrent une approche interactive qui améliore l'efficacité des chercheurs en SHS et rend leur travail plus reproductible. Les carnets Juputer ont été conçus pour être compatibles avec de nombreux langages de programmation utilisés en SHS, notamment Python, R et Julia. Ils permettent d'exécuter facilement du code dans ces langages et d'interagir avec les bibliothèques utilisées.

Dans ce contexte d'engouement général, nous proposons ci-dessous un panorama des potentiels d'utilisation de cet outil, dans le cadre de la pratique de recherche en SHS impliquant à divers degrés du traitement de données avec un langage de programmation. En contrepoint, il sera aussi question des difficultés de prise en main liées à d'éventuels obstacles techniques : face à la diversité des usages se pose en effet la question de la stabilisation des bonnes pratiques et de l'adoption de cet outil par de nouveaux arrivants.

[10] https://github.com/quinnanya/litlab_translations/blob/master/litlab_translations_2019-04-15_jupyter_notebook.ipynb.
[11] https://gitlab.huma-num.fr/io/mobilites-professionnelles-final.

2. Atouts et potentiels des carnets Jupyter

Création d'un standard	- Utilisé dans toutes les branches des sciences sociales et dans des domaines différents. - 100% libre et interopérable. - Plus de 100 langages de programmation. - Utilisable depuis tout navigateur et équipement (ordinateur, tablette,) - Homogénéise les données créées. - Facilite le partage des données et leur traitement.
Un support unique pour penser et raconter, avec du code et des données	- Permet de combiner narration, calcul mathématique, visualisation graphique et interactivité dans un même document. - Met en place des cellules pour clarifier les différentes étapes. Favorise : - L'autonomie et la personnalisation grâce à la structure narrative, à la micro-scénarisation, aux retours immédiats. - L'engagement des étudiants grâce à l'interaction et à la liberté d'exploration (possibilité de faire des exercices interactifs). - La flexibilité : n'importe où, n'importe quand sur n'importe quel terminal. - L'exploration notamment par des interactions riches grâces aux ipywidgets[12] - La transmission à des chercheurs ou à des étudiants.
Travail réutilisable	- Facilite l'accès à son travail et son partage. - Peut être converti dans d'autres formats. - S'intègre naturellement avec des forges comme GitHub[13] ou GitLab[14] pour faciliter la traçabilité, la robustesse, le partage, la publication et l'archivage. - Permet à d'autres chercheurs de modifier le code pour obtenir d'autres résultats. - Permet aux étudiants d'envoyer leurs travaux en programmation à leur professeur.

[12] Les ipywidgets sont des éléments conçus pour les carnets Jupyter. Ils permettent d'ajouter des composants interactifs tels que des boutons, des curseurs, des listes déroulantes, des graphiques aux carnets Jupyter. Cew widgets facilitent l'interaction utilisateur en offrant des fonctionnalités interactives et en permettant la manipulation des données en temps réel. Ils sont très utilisés dans le cadre de l'enseignement et ils permettent aussi à des personnes qui ne savent pas programmer d'interagir avec le carnet numérique.

[13] Github est une plateforme de collaboration qui permet d'héberger au format open-source des projets informatiques comportant du code. Les utilisateurs ont accès à des millions et des millions de fichiers python ou des Jupyter Notebooks sur des sujets très variés et dans des disciplines différentes. Ce service propose aussi des fonctionnalités recherchées par les programmeurs : l'intégration continue, la gestion de versions…

[14] GitLab est une plateforme web de gestion de code source et de collaboration pour les développeurs. Cette plateforme fournit un système de contrôle de version basé sur Git, ainsi que des fonctionnalités complètes pour la gestion de projets comme le déploiement d'une application.

	- Utile pour la publication d'articles.
Facilite les échanges entre chercheurs en SHS	- Permet à la fois de partager du texte et du code dans le même document. - Utile pour la publication d'articles. - Pas de but fixé par le logiciel : l'utilisateur réfléchit aux résultats qu'il veut obtenir. - Permet d'expliquer clairement son domaine de recherche à d'autres chercheurs lors d'une conférence. - Permet le travail collaboratif à partir d'un même outil.
Mise en place d'un écosystème d'extensions et d'évènements	- Peut être traduit possible dans d'autres formats : livres, slides. - Peut être partagé et visualisé via Nbviewer[15] et Binder[16]. - Utilisé par une communauté d'utilisateurs qui construit des extensions, des applications et qui aide les nouveaux utilisateurs comme JupyterLite[17] qui permet d'avoir un carnet Jupyter qui s'exécute entièrement dans un navigateur web sans avoir à installer Anaconda ou Python. Un autre exemple est myST[18] qui est une extension permettant d'écrire des documents avec des fonctionnalités avancées de formatage et de structuration de texte. Le but de MyST est d'automatiser la création d'articles scientifiques à partir d'un carnet Jupyter en créant à partir d'un modèle choisi par l'utilisateur en sortie une page HTML. - Suscite l'organisation d'évènements communs (Jupyter Community Workshop[19], JupyterDays[20] et JupyterCon[21]).

[15] Nbviewer est disponible à cette adresse : https://nbviewer.org/. Cette solution est utilisée pour partager plus facilement les carnets Jupyter, sans passer par l'installation de logiciels tiers. Le carnet Jupyter est disponible sous forme de page HTML statique. Il est possible d'utiliser le carnet Jupyter au format HTML dans un site web en copiant/collant l'URL dans le code. Un tutoriel est disponible pour apprendre à utiliser nbviewer ici : https://www.tutorialspoint.com/jupyter/sharing_jupyter_notebook_using_github_and_nbviewer.htm.

[16] Binder est une extension permettant de réexécuter le code et de supprimer ou d'ajouter des cellules. Elle rend le carnet Jupyter interactif au contraire de nbviewer qui donne une version du carnet Jupyter statique.

[17] https://jupyter.org/try-jupyter/lab/index.html.

[18] https://jupyterbook.org/en/stable/content/myst.html.

[19] La communauté Jupyter propose beaucoup d'ateliers sur des thématiques précises. L'un d'eux s'est par exemple tenu en décembre 2022 à Paris sur le projet JupyterLite qui permet d'avoir un carnet Jupyter qui tourne dans le navigateur web sans avoir à faire d'autres installations.

[20] Les JupyterDays sont des conférences d'un jour sur le sujet des carnets Jupyter.

[21] La JupyterCon est le colloque qui regroupe tous les utilisateurs des carnets Jupyter. L'événement dure plusieurs jours et comporte des conférences, des tables rondes et des cours. La JupyterCon permet de renforcer la communauté et d'avoir des interactions en personne pour des utilisateurs qui collaborent à distance durant le reste de l'année.

3. Limites et difficultés d'utilisation des carnets Jupyter

Nécessite des connaissances de base	- Fortement conseillé de connaître un langage informatique (Python ou autre) - Propose une philosophie qui n'est pas nécessairement intuitive dans l'exécution du code. - La force des carnets est de permettre d'entremêler narration, calcul, programmation. Lorsqu'une seule de ses activités est en jeu, les utilisateurs doivent être conscients que les carnets n'ont pas vocation à remplacer les outils dédiés comme les Environnements de Développement Intégrés (IDE)[22] pour la programmation. - Face à cet outil très personnalisable : difficulté de savoir quels plugins ou extensions exploiter.
Outil qui nécessite un apprentissage	- Compliqué de passer de Visual Studio Code à un carnet Jupyter pour un débutant - Faible pratique des tests[23] : il est compliqué de faire des tests pour un carnet Jupyter mais des solutions existent comme : testbook qui est un framework qui aide à tester le code dans le notebook. D'autres solutions sont : Nbval, pytest-notebook qui permettent de garantir la reproductibilité du notebook. - Favorise des pratiques discutables[24] dans le nom des fichiers et dans la version et la modularisation du code. - Pas de règles homogènes sur l'écriture des carnets Jupyter. Par exemple : il y a moins de commentaires à la fin des carnets Jupyter : la plupart des carnets Jupyter ont une cellule d'introduction (55%) mais presque aucun n'a de cellules de conclusions[25]. - Le code peut être exécuté de manière non linéaire, provoquant des « hidden states[26] ».
Possibles problèmes de reproductibilité	- Compliqué à reproduire pour les débutants car ils nécessitent de réécrire le code (sur un million de carnets numériques sur

[22] Un IDE est un ensemble d'outils permettant d'augmenter la productivité des programmeurs qui créent des logiciels. Un IDE comporte un éditeur de texte pour faire de la programmation. Les exemples d'IDE sont nombreux : Visual Studio Code, CodeLite, NetBeans, PytCharm…

[23] Un test est un morceau de code qui permet de vérifier que le programme ne produit pas de bugs.

[24] Les carnets Jupyter ont souvent des noms de fichiers générés automatiquement basés sur des informations par défaut, tels que « Untilted.ipynb » ou des noms qui correspondent à l'ordre dans lequel ils ont été créés. Cela peut rendre l'organisation et la recherche des carnets Jupyter compliqué.

[25] Rule, A., Tabard, A., & Hollan, J. D. (2018). *Exploration and Explanation in Computational Notebooks*. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18. doi:10.1145/3173574.3173606.

[26] Il est possible d'exécuter le code de manière non linéaire et donc de provoquer des « hidden states ». Ces derniers peuvent entraîner la création de variables ou de résultats intermédiaires qui ne sont pas immédiatement visibles dans le carnet Jupyter. Lorsque le code est exécuté de manière linéaire, chaque instruction est exécutée dans un ordre séquentiel mais les carnets Jupyter donnent la possibilité d'exécuté le code de manière non linéaire. Il est primordial de prendre cela en compte lors de l'écriture d'un carnet Jupyter car cela peut rendre la gestion des dépendances entre les cellules plus complexe et peut entraîner des changements dans les résultats si les cellules sont exécutées dans un ordre différent.

	Github, seulement 24% ont pu être exécutés sans erreurs et 4% sont utilisables en état)[27]. - Impossible à reproduire le carnet Jupyter dans certains cas : le code peut être obsolète. Il faut aussi retrouver la bonne version des bibliothèques utilisées. - Le lien entre les données et le carnet numérique n'est pas évident : l'accès aux fichiers est l'une des causes courantes pour lesquelles les carnets numériques ne sont pas reproductibles. Il faut changer le chemin du fichier directement dans le code du carnet numérique.
Un écosystème complexe	- Importance des utilisateurs experts qui vont explorer l'écosystème et construire des solutions et bonnes pratiques en fonction des besoins, en combinant les outils existants : importance des communautés, des réseaux et de la formation pour partager ces solutions. - Les différents IDE (PyCharm, Spyder, Visual Studio Code) permettent de contrôler le code avec des linters[28], de faire du contrôle de version et du debugging[29]. - Les outils « cloud based »[30] comme Google Collab et Azure Notebook permettent de collaborer avec d'autres utilisateurs et de contrôler la version[31]. - Diversité de formats similaires et difficulté pour choisir : des carnets numériques comme Apache Zeppelln[32] ou BeakerX[33]. - Deepnote[34] permet de collaborer dans le même carnet numérique alors que les carnets Jupyter sont généralement stockés en local dans un ordinateur. Deepnote prend en charge le contrôle des versions et permet de mettre en place des permissions différentes en fonction des utilisateurs du carnet numérique.

4. Solutions aux obstacles techniques et bonnes pratiques

La plupart des problèmes exposés ci-dessus peuvent cependant être résolus grâce à des extensions, des logiciels à utiliser en complément des carnets Jupyter. Par exemple, une des critiques adressées est le fait qu'il n'y ait pas d'auto-complétion du code, ce qui peut poser problème à des utilisateurs

[27] Manon Marchand, Stefania Amodeo, Mark Allen. *Jupyter notebooks maintenance tips and tricks*. 2022. ⟨hal03970692⟩
[28] Un Linter est un outil d'analyse de code qui permet de détecter les erreurs et les problèmes de syntaxe.
[29] Le « debugging » est une pratique qui consiste à trouver et à corriger les erreurs dans le code. Les programmeurs étudient le code pour déterminer la raison des erreurs. Le but est d'exécuter le code et de le vérifier étape par étape pour résoudre le problème.
[30] « Cloud based » signifie que les données ne sont pas sur un serveur local ou sur un ordinateur personnel mais sur des serveurs informatiques à distance puis ensuite hébergés sur internet.
[31] Le « versioning » permet d'avoir la version des librairies utilisées ce qui est utile pour pouvoir reproduire le carnet numérique. Certaines fonctionnalités ne sont disponibles qu'avec une version spécifique de la librairie.
[32] Apache Zepellin est un projet similaire aux carnets Jupyter. Apache Zepellin est un projet permettant d'analyser et de mettre en forme, de manière visuelle et interactive, de gros volumes de données traités via le framework de calcul distribué Spark. (source : https://fr.wikipedia.org/wiki/Apache_Zeppelin).
[33] BeakerX est une extension open source du projet Jupyter visant à améliorer la visualisation graphique et l'interactivité des carnets Jupyter.
[34] DeepNote est un type de carnet numérique qui facilite la collaboration entre plusieurs utilisateurs.

débutants. Toutefois l'extension Hinterland permet d'activer l'auto-complétion du code dans les carnets Jupyter[35].

Une autre critique est la difficulté pour les utilisateurs débutants à utiliser les carnets Jupyter. Mais le "Snippets" menu permet d'obtenir du code prêt à être utilisé avec différentes librairies et pour différents usages (figure 1)[36]. Cela permet ainsi à des utilisateurs débutants de ne pas avoir à regarder la documentation pour avoir la syntaxe du code.

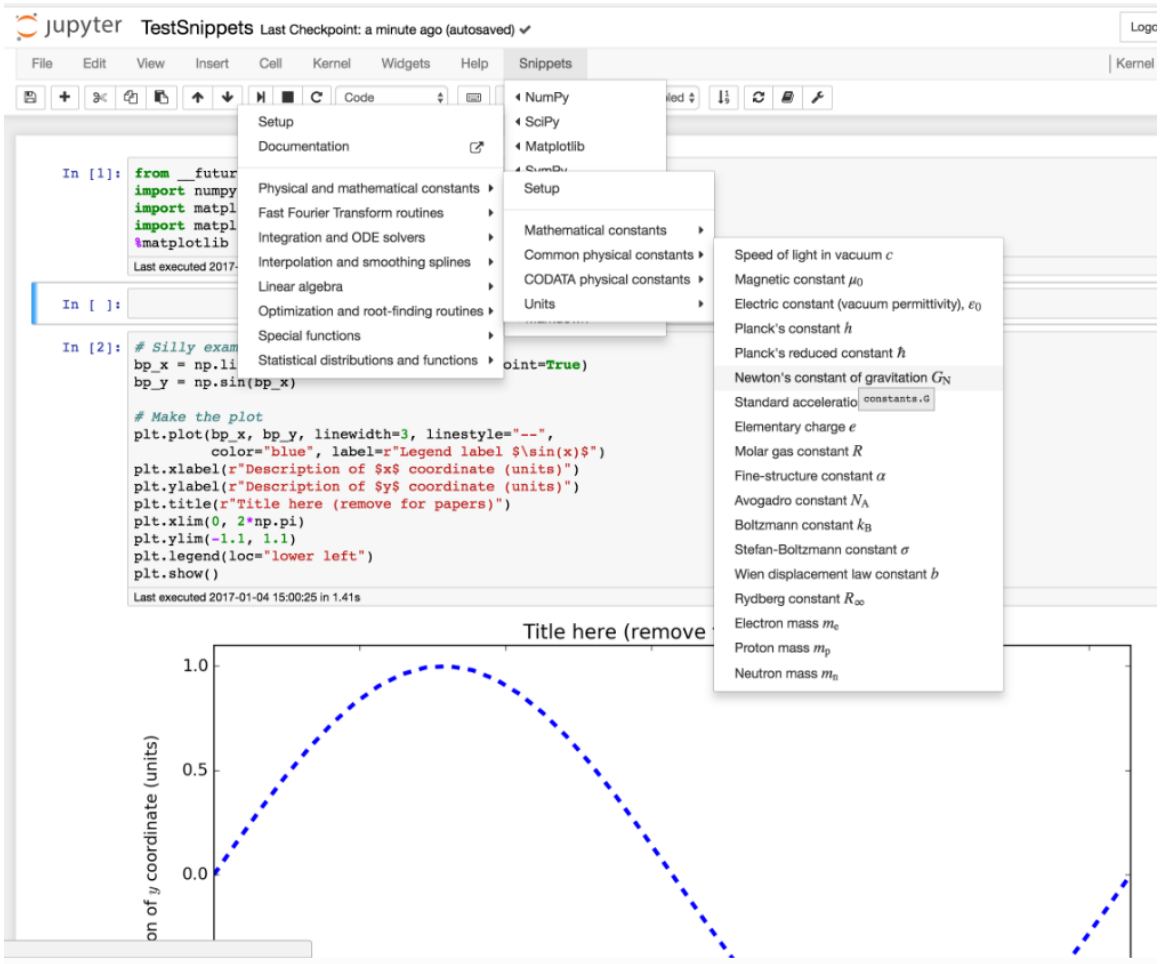

Figure 1 : Exemple d'utilisation des Snippets

Bien qu'il y ait beaucoup d'extensions facilitant l'usage des carnets Jupyter, il existe des pré-requis pour les utiliser, comme avoir des connaissances en codage Python sur un IDE (comme Visual Studio Code) préférablement avant de passer à un carnet Jupyter, savoir changer les chemins des fichiers pour pouvoir réutiliser le code, ou installer Anaconda sur son ordinateur pour pouvoir utiliser les carnets Jupyter et d'autres outils.
L'atout majeur des carnets numériques est donc de permettre d'entremêler narration, calcul et programmation. Les carnets n'ont pas vocation à remplacer les outils dédiés, comme les

[35] Pour un tutoriel d'installation d'Hinterland, ainsi que d'autres extensions, voir : https://www.endtoend.ai/blog/jupyter-notebook-extensions-to-enhance-your-efficiency/.
[36] Pour un tutoriel d'installation de Snippets, voir : https://jupyter-contrib-nbextensions.readthedocs.io/en/latest/nbextensions/snippets_menu/readme.html

environnements de développement intégrés pour la programmation, mais ouvrent un dialogue entre différents modes de traitements et d'exploration des données. La littérature permet d'identifier quelques bonnes pratiques pour construire et partager un carnet numérique[37] dans le but de rendre ce médium lisible et réutilisable :

- **Raconter une histoire**. Cela nécessite une explication des différentes étapes pour essayer de résoudre la question de la recherche.

- **Documenter tout le processus** et pas seulement les résultats.

- **Structurer la narration** en la découpant en cellules (quelques paragraphes maximums par cellule).

- **Rendre le carnet numérique le plus lisible possible.**

- **Utiliser la modularité**. Il est important d'éviter de dupliquer le code. Dans les carnets numériques, il est facile de copier/coller du code et de changer quelques lignes. Mais cela rend le carnet numérique très compliqué à lire et impossible à maintenir. Le concept important à garder en tête est le DRY (Don't Repeat Yourself). Il est préférable de mettre le code dans une fonction et d'appeler cette fonction dans différentes cellules.

- **Bien séparer programmation et narration**. Un code non trivial ne doit apparaître dans un carnet que s'il fait explicitement partie de l'histoire, s'il est important que l'auditeur le lise. Tout le reste du code a vocation à être dans des fichiers annexes. Cela favorise aussi la réutilisabilité.

- **Choisir une licence libre pour permettre la réutilisation** (https://choosealicense.com/)
- **Utiliser le contrôle de version**. Du fait de l'aspect interactif du carnet numérique il est facile de changer accidentellement du code. Un papier existe pour utiliser Git[38] pour faire du contrôle de version[39].

- **Publier sur une forge publique** comme Gitlab ou bien Github avec un fichier READme[40] et un fichier requirement avec les différentes versions des librairies.

- **Bien organiser l'architecture du dossier du projet**, notamment les données et les documents intermédiaires.

- **Créer un modèle :** Le but est de faire un carnet numérique qui soit réutilisable en changeant les données et les paramètres voir papermil[41] pour plus d'informations.

- **Possibilité de décrire l'environnement d'exécution requis :** Ce qui ouvre la possibilité d'utilisation en ligne (voir Zenodo[42] ou Binder[43]).

[37] Rule, A., Birmingham, A., Zuniga, C., Altintas, I., Huang, S.-C., Knight, R., … Rose, P. W. (2019). Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks. PLOS Computational Biology, 15(7), e1007007. doi: https://doi.org/10.1371/journal.pcbi.1007007.

[38] Git est un système de gestion de versions décentralisé qui facilite la gestion et le suivi des modifications apportées à un contenu. Il est largement utilisé et il est à la base de la fameuse plateforme Github. Chaque fois qu'un utilisateur effectue une mise à jour de son dépôt GitHub, Git est utilisé pour enregistrer et gérer ces modifications, ce qu'on appelle un « git-push ». Git permet de conserver un historique complet des modifications apportées au contenu, ce qui facilite la collaboration avec la gestion des différentes versions d'un projet.

[39] Perez-Riverol, Y., Gatto, L., Wang, R., Sachsenberg, T., Uszkoreit, J., Leprevost, F. da V., … Vizcaíno, J. A. (2016). Ten Simple Rules for Taking Advantage of Git and GitHub. PLOS Computational Biology, 12(7), e1004947. doi:10.1371/journal.pcbi.1004947 .

[40] Un fichier readme est un fichier contenant des informations sur les autres fichiers du même répertoire. Il est utilisé pour expliquer la démarche derrière un projet publié sur Github par exemple.

[41] https://github.com/nteract/papermill.

[42] https://zenodo.org/record/4421040.

[43] Binder est une extension permettant de réexécuter le code et de supprimer ou d'ajouter des cellules. Elle rend le carnet Jupyter interactif au contraire de nbviewer qui donne une version du carnet Jupyter statique.

Exploring A Medical History of British India

Created in July-September 2020 for the National Library of Scotland's Data Foundry by Lucy Havens, Digital Library Research Intern

About the *A Medical History of British India* Dataset

The dataset consists of 468 official publications from British India, mainly from 1850-1950, that report on public health, disease mapping, vaccination efforts veterinary experiments, and other medical topics. The publications are a subset of a larger collection of 40,000 volumes that report on the administration of British India. The Wellcome Trust funded the digitisation of the medical history volumes in this dataset.

- Data format: digitised text
- Data creation process: Optical Character Recognition (OCR) and manual cleaning
- Data source: https://data.nls.uk/data/digitised-collections/a-medical-history-of-british-india/

Table of Contents

Citations

- Alex, Beatrice and Llewellyn, Clare. (2020) *Library Carpentry: Text & Data Mining*. Centre for Data, Culture & Society, University of Edinburgh. http://librarycarpentry.org/lc-tdm/.
- Bird, Steven and Klein, Ewan and Loper, Edward. (2019) *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolki* O'Reilly Media. 978-0-596-51649-9. https://www.nltk.org/book/.

0. Preparation

Import libraries to use for cleaning, summarising and exploring the data:

In [27]:
```
# To prevent SSL certificate failure
import os, ssl
if (not os.environ.get('PYTHONHTTPSVERIFY', '') and
    getattr(ssl, '_create_unverified_context', None)):
    ssl._create_default_https_context = ssl._create_unverified_context

# Libraries for data loading
import pandas as pd
import numpy as np
import string
import re

# Libraries for visualization
import altair as alt
import matplotlib.pyplot as plt

# Libraries for text analysis
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
nltk.download('punkt')
from nltk.corpus import PlaintextCorpusReader
nltk.download('wordnet')
from nltk.corpus import wordnet
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.text import Text
from nltk.stem.porter import PorterStemmer
from nltk.probability import FreqDist
nltk.download('averaged_perceptron_tagger')
from nltk.tag import pos_tag
nltk.download('tagsets')  # part of speech tags
```

Figure 2 : Exemple d'un carnet Jupyter de qualité[44]

[44] Voir https://data.nls.uk/wp-content/uploads/2020/10/Exploring_Medical_History_of_British_India.html.

Le carnet Jupyter de la figure 2 est de très bonne qualité. Sont présents le titre, la date de création et les auteurs. Il contient une présentation des données utilisées ainsi qu'une table des matières. L'article qui discute de ce code est référencé. La démarche est très claire : un titre est présent, le code a des commentaires et les cellules au format texte permettent de suivre la réflexion. Le carnet Jupyter présenté est disponible au format Web ce qui est utile pour l'utilisateur n'ayant pas installé Anaconda sur son ordinateur.

Conclusion

Le calcul et le traitement de données en science, notamment avec Python, et les nouveaux outils comme les carnets Jupyter sont importants pour les sciences humaines et sociales. Une partie des critiques adressées aux carnets Jupyter ne sont pas dues au « médium » mais plutôt à la manière dont les personnes s'en servent. La plupart des problèmes exposés dans cet article ont des solutions : les carnets Jupyter sont très modifiables selon les besoins de ses utilisateurs.

Voici quelques types d'usage concret de carnet Jupyter :

- Dans le cadre d'une conférence pour montrer et expliquer les résultats de recherche produits. Le carnet numérique facilite l'interaction : Il permet de faire une démonstration très facilement, ce qui en dynamise la présentation.
- Dans le cadre d'un travail collaboratif de recherche entre des chercheurs connaissant l'outil et Python. Il est plus facile de travailler sur un outil en commun plutôt qu'avec des fichiers utilisant des logiciels différents.
- Dans le cadre d'un enseignement sur l'informatique. Les carnets numériques permettent de mettre en place des exercices qui peuvent être ensuite fait par les étudiants. Les carnets numériques sont très facilement modifiables : les étudiants peuvent comprendre la modification dans le code et voir comment les résultats évoluent. Dans le domaine de l'enseignement, les retours des étudiants sont très positifs[45] :

    - « L'interface des carnets Jupyter permet une scénarisation du TP mêlant réflexion scientifique et utilisation d'un langage informatique sans trop de difficultés ».
    - « J'ai beaucoup aimé les carnets Jupyter car ils permettent de bien organiser le travail et de revenir dessus quand on veut ».
    - « Très pratique, facilite l'accès à notre travail partout, vraiment utile ».

Les carnets Jupyter ne mettent pas la technique au cœur du processus de recherche mais bien l'humain : l'environnement favorise l'interaction et l'exploration. Grâce à leur interactivité, les carnets Jupyter offrent aux utilisateurs un environnement intégré ou ils peuvent exécuter, modifier et document leurs travaux dans un même endroit. Cela en fait un outil qui peut être très apprécié dans le domaine des sciences humaines et sociales.

La communauté des carnets Jupyter est largement ancrée dans la communauté scientifique : le projet Jupyter a été créé par des chercheurs, pour des chercheurs. Ce faisant, des valeurs similaires sont partagées autour du livre et des communs numériques, de l'aide aux nouveaux arrivants, de l'exploration des nouvelles solutions et de la reproductibilité. Les carnets Jupyter ne sont pas seulement un outil, mais aussi et principalement une communauté dynamique qui crée de nouvelles solutions en permanence adaptées à la recherche. La diversité des conférences présentées à La JupyterCon 2023 qui s'est tenue à Paris témoigne de cette vitalité et de la capacité d'adaptation et d'innovation de ses

[45] Jérémy Tuloup, Claire Vandiedonck, Sandrine Caburet, Pierre Poulain. Plasma : plateforme d'e-learning pour l'analyse interactive de données. Journées Réseaux de l'Enseignement et de la Recherche (JRES), May 2022, Marseille, France. ⟨hal-03563658v2⟩

membres. Le projet Jupyter est donc avant tout un projet communautaire : les acteurs sont au centre du processus et ils créent de nombreux outils. Avec le développement des nouvelles technologies dans le champ de la recherche, l'évolution de la recherche et de l'enseignement en SHS, les carnets numériques ont un rôle à jouer pour démocratiser le calcul et le traitement de données et contribuer à la science ouverte dans les SHS.

Ressources utiles

- Téléchargement de Python sur le site officiel : https://www.python.org/downloads/.

- Téléchargement de l'éditeur de code Visual Studio Code : https://code.visualstudio.com/.

- Ressources gratuites pour apprendre à coder, depuis le site FreeCodeCamp : https://www.freecodecamp.org/.

- Vidéo sur Python (de très bonne qualité) sur la chaîneYouTube de FreeCodeCamp : https://youtu.be/rfscVS0vtbw.

- Leçons en ligne sur le site Programming Historian, en anglais, espagnol et français, pour aider les chercheurs en SHS à acquérir des compétences en programmation : https://programminghistorian.org/fr/.

- Téléchargement d'Anaconda, pour ouvrir et créer des carnets Jupyter : https://www.anaconda.com/download. Anaconda propose de nombreux services, comme des IDE (DataSpell, Spyder) et d'autres logiciels comme JupyterLab, Jupyter Notebook, RStudio.

- Site officiel de Jupyter, où trouver de nombreuses extensions pour pouvoir partager son code et modifier celui d'autres travaux : https://jupyter.org/.

- Forum Stack Overflow, dédié à la communauté des développeurs : https://stackoverflow.com/.

- Un exemple de Jupyter notebook réalisé par l'auteur de cet article (visant à présenter les bonnes pratiques à mettre en place pour constituer un Notebook) : https://maximepopineau-project.curve.space/markdown-notebooks.