Rendez-vous de l’info scientifique

Traitement de données avec Pandas & Jupyter notebooks

Pablo Iriarte – pablo.iriarte@unige.ch / DIS

25 septembre 2024

BIBLIOTHÈQUE

UNIVERSITÉ DE GENÈVE

Programme

Introduction

- Historique
- Excel et les erreurs scientifiques
- Reproducibility Crisis & Data deluge

Jupyter Notebooks

- Famille d’outils
- Accès au JupyterHub du cours ou installation via la distribution Anaconda
- Créer, organiser et partager des notebooks

Pandas

- Importer et exporter des données
- Manipuler et analyser les données
- Générer des graphiques

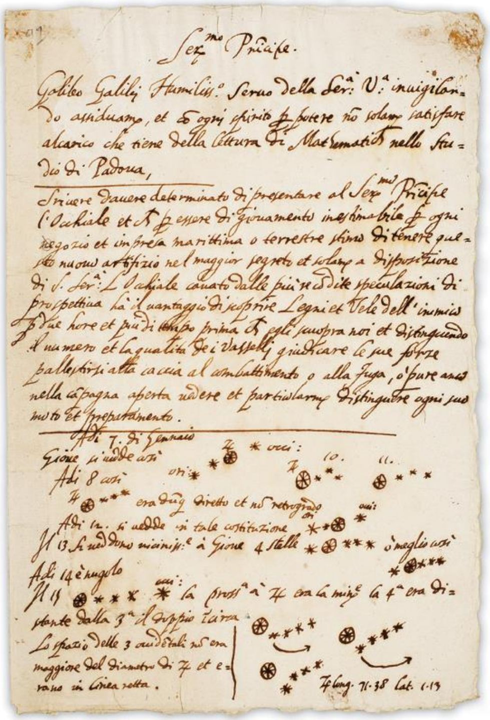
Introduction

Historique

- **iPython** (2001->) https://ipython.org/
- **Jupyter** (2014 ->) https://jupyter.org/
- **Famille d’outils**
  - **Jupyter Hub** : https://jupyterhub.readthedocs.io/
  - **Jupyter Lab** : https://jupyterlab.readthedocs.io/
  - **NB viewer** : https://nbviewer.jupyter.org/
  - **Binder** : https://mybinder.org/

Introduction

Historique : Jupiter



Ser.mo Principe.

Galileo Galilei Humiliss.o Seruo della Ser.tà V.a inuigilando assiduam.te, et con ogni spirito per potere non solam.te satisfare al carico che tiene della lettura di Matematica nello Studio di Padoua,

Scriue d'hauere determinato di presentare al Ser.mo Principe l'Occhiale et di poter essere di giouamento inestimabile per ogni negozio et impresa marittima o terrestre stima di tenere questo nuouo artifizio nel maggior segreto et solam.te a disposizione di V. Ser.tà L'Occhiale cauato dalle più recondite speculazioni di prospettiua ha il uantaggio di scoprire Legni et Vele dell'inimico per due hore et più di tempo prima che egli scuopra noi et distinguendo il numero et la qualità dei Vasselli giudicare le sue forze pallestirsi alla caccia al combattimento o alla fuga, o pure anco nella campagna aperta uedere et particolarm.te distinguere ogni suo moto et preparamento.

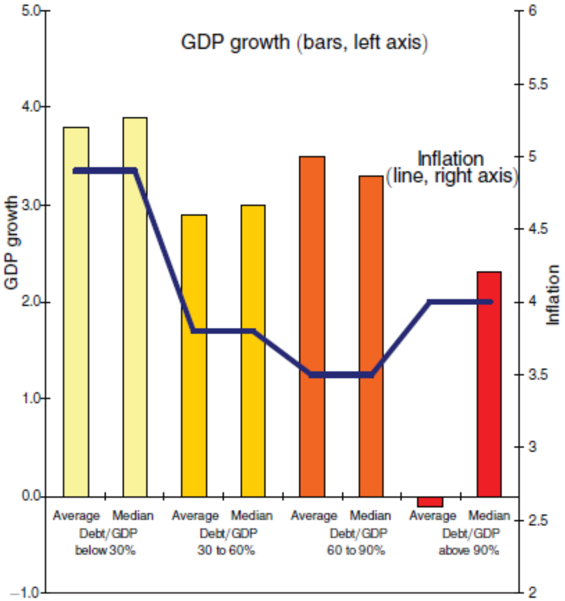
Adi 7. di Gennaio
Giove si uedde così * * ♃ * così:
Adi 8 così ♃ * * * 10. ♃ * * * 11. ⊛ * * *
era dunq. diretto et non retrogrado
Adi 12. si uedde in tale costituzione * * ⊛ * così:
Il 13 si ueddono uicinissime à Giove 4 stelle * ⊛ * * * ò meglio così * ⊛ * * *
Adi 14 è nugolo
Il 15 ⊛ * * * * così: la pross.a à ♃ era la minore la 4.a era distante dalla 3.a il doppio incirca
Lo spatio delle 3 occidentali non era maggiore del diametro di ♃ et erano in linea retta.
♃ long. 71.38 lat. 1.13

https://commons.wikimedia.org/wiki/File:Galileo_manuscript.png

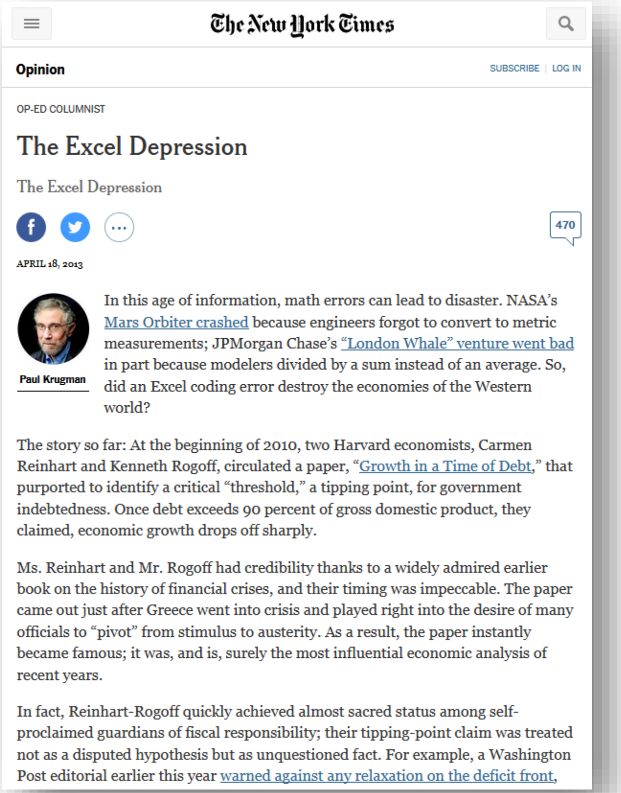
Introduction

Excel et les erreurs scientifiques

L'exemple du «Reinhart-Rogoff error»



Reinhart, Carmen M., and Kenneth S. Rogoff. 2010. DOI:10.1257/aer.100.2.573



OP-ED COLUMNIST

The Excel Depression

The Excel Depression

APRIL 18, 2013

In this age of information, math errors can lead to disaster. NASA's Mars Orbiter crashed because engineers forgot to convert to metric measurements; JPMorgan Chase's "London Whale" venture went bad in part because modelers divided by a sum instead of an average. So, did an Excel coding error destroy the economies of the Western world?

The story so far: At the beginning of 2010, two Harvard economists, Carmen Reinhart and Kenneth Rogoff, circulated a paper, "Growth in a Time of Debt," that purported to identify a critical "threshold," a tipping point, for government indebtedness. Once debt exceeds 90 percent of gross domestic product, they claimed, economic growth drops off sharply.

Ms. Reinhart and Mr. Rogoff had credibility thanks to a widely admired earlier book on the history of financial crises, and their timing was impeccable. The paper came out just after Greece went into crisis and played right into the desire of many officials to "pivot" from stimulus to austerity. As a result, the paper instantly became famous; it was, and is, surely the most influential economic analysis of recent years.

In fact, Reinhart-Rogoff quickly achieved almost sacred status among self-proclaimed guardians of fiscal responsibility; their tipping-point claim was treated not as a disputed hypothesis but as unquestioned fact. For example, a Washington Post editorial earlier this year warned against any relaxation on the deficit front,

https://www.nytimes.com/2013/04/19/opinion/krugman-the-excel-depression.html

Introduction

Excel et les erreurs scientifiques

L’exemple de la conversion des données

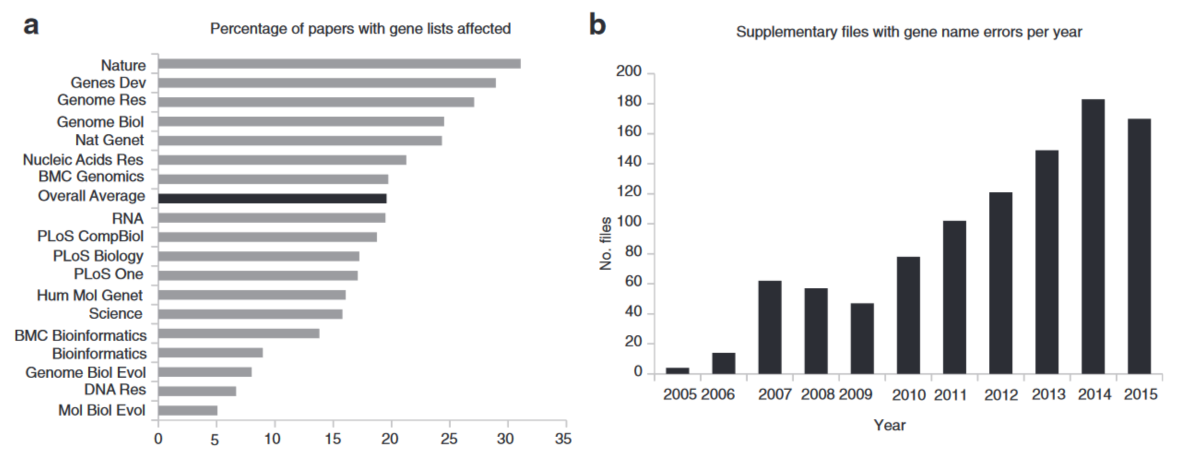
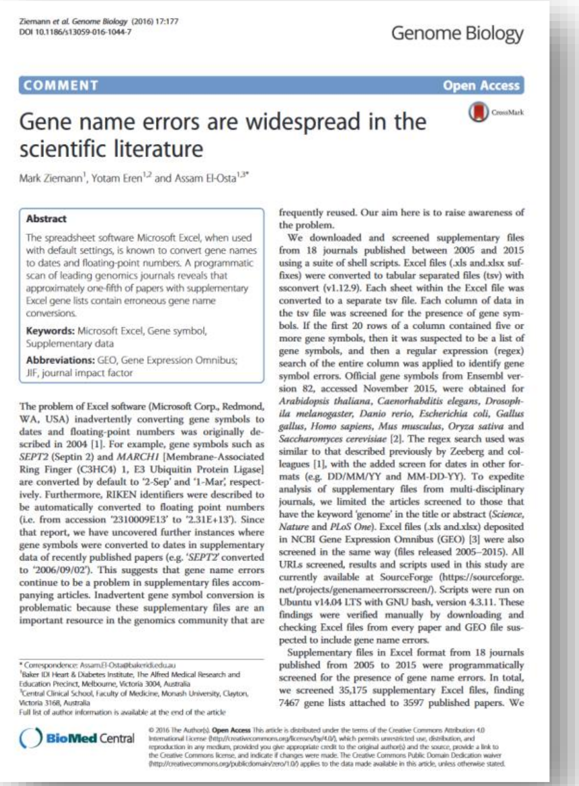


Fig. 1 Prevalence of gene name errors in supplementary Excel files. **a** Percentage of published papers with supplementary gene lists in Excel files affected by gene name errors. **b** Increase in gene name errors by year



Ziemann *et al. Genome Biology* (2016) 17:177
DOI 10.1186/s13059-016-1044-7

Genome Biology

COMMENT — Open Access

CrossMark

Gene name errors are widespread in the scientific literature

Mark Ziemann[1], Yotam Eren[1,2] and Assam El-Osta[1,3*]

Abstract

The spreadsheet software Microsoft Excel, when used with default settings, is known to convert gene names to dates and floating-point numbers. A programmatic scan of leading genomics journals reveals that approximately one-fifth of papers with supplementary Excel gene lists contain erroneous gene name conversions.

Keywords: Microsoft Excel, Gene symbol, Supplementary data

Abbreviations: GEO, Gene Expression Omnibus; JIF, journal impact factor

The problem of Excel software (Microsoft Corp., Redmond, WA, USA) inadvertently converting gene symbols to dates and floating-point numbers was originally described in 2004 [1]. For example, gene symbols such as *SEPT2* (Septin 2) and *MARCH1* [Membrane-Associated Ring Finger (C3HC4) 1, E3 Ubiquitin Protein Ligase] are converted by default to ‘2-Sep’ and ‘1-Mar’, respectively. Furthermore, RIKEN identifiers were described to be automatically converted to floating point numbers (i.e. from accession ‘2310009E13’ to ‘2.31E+13’). Since that report, we have uncovered further instances where gene symbols were converted to dates in supplementary data of recently published papers (e.g. ‘*SEPT2*’ converted to ‘2006/09/02’). This suggests that gene name errors continue to be a problem in supplementary files accompanying articles. Inadvertent gene symbol conversion is problematic because these supplementary files are an important resource in the genomics community that are frequently reused. Our aim here is to raise awareness of the problem.

We downloaded and screened supplementary files from 18 journals published between 2005 and 2015 using a suite of shell scripts. Excel files (.xls and.xlsx suffixes) were converted to tabular separated files (tsv) with ssconvert (v1.12.9). Each sheet within the Excel file was converted to a separate tsv file. Each column of data in the tsv file was screened for the presence of gene symbols. If the first 20 rows of a column contained five or more gene symbols, then it was suspected to be a list of gene symbols, and then a regular expression (regex) search of the entire column was applied to identify gene symbol errors. Official gene symbols from Ensembl version 82, accessed November 2015, were obtained for *Arabidopsis thaliana*, *Caenorhabditis elegans*, *Drosophila melanogaster*, *Danio rerio*, *Escherichia coli*, *Gallus gallus*, *Homo sapiens*, *Mus musculus*, *Oryza sativa* and *Saccharomyces cerevisiae* [2]. The regex search used was similar to that described previously by Zeeberg and colleagues [1], with the added screen for dates in other formats (e.g. DD/MM/YY and MM-DD-YY). To expedite analysis of supplementary files from multi-disciplinary journals, we limited the articles screened to those that have the keyword ‘genome’ in the title or abstract (*Science*, *Nature* and *PLoS One*). Excel files (.xls and.xlsx) deposited in NCBI Gene Expression Omnibus (GEO) [3] were also screened in the same way (files released 2005–2015). All URLs screened, results and scripts used in this study are currently available at SourceForge (https://sourceforge.net/projects/genenameerrorsscreen/). Scripts were run on Ubuntu v14.04 LTS with GNU bash, version 4.3.11. These findings were verified manually by downloading and checking Excel files from every paper and GEO file suspected to include gene name errors.

Supplementary files in Excel format from 18 journals published from 2005 to 2015 were programmatically screened for the presence of gene name errors. In total, we screened 35,175 supplementary Excel files, finding 7467 gene lists attached to 3597 published papers. We

* Correspondence: Assam.El-Osta@bakeridi.edu.au
[1] Baker IDI Heart & Diabetes Institute, The Alfred Medical Research and Education Precinct, Melbourne, Victoria 3004, Australia
[3] Central Clinical School, Faculty of Medicine, Monash University, Clayton, Victoria 3168, Australia
Full list of author information is available at the end of the article

BioMed Central

Ziemann et al. 2016. [*DOI:10.1186/s13059-016-1044-7*](https://doi.org/10.1186/s13059-016-1044-7)

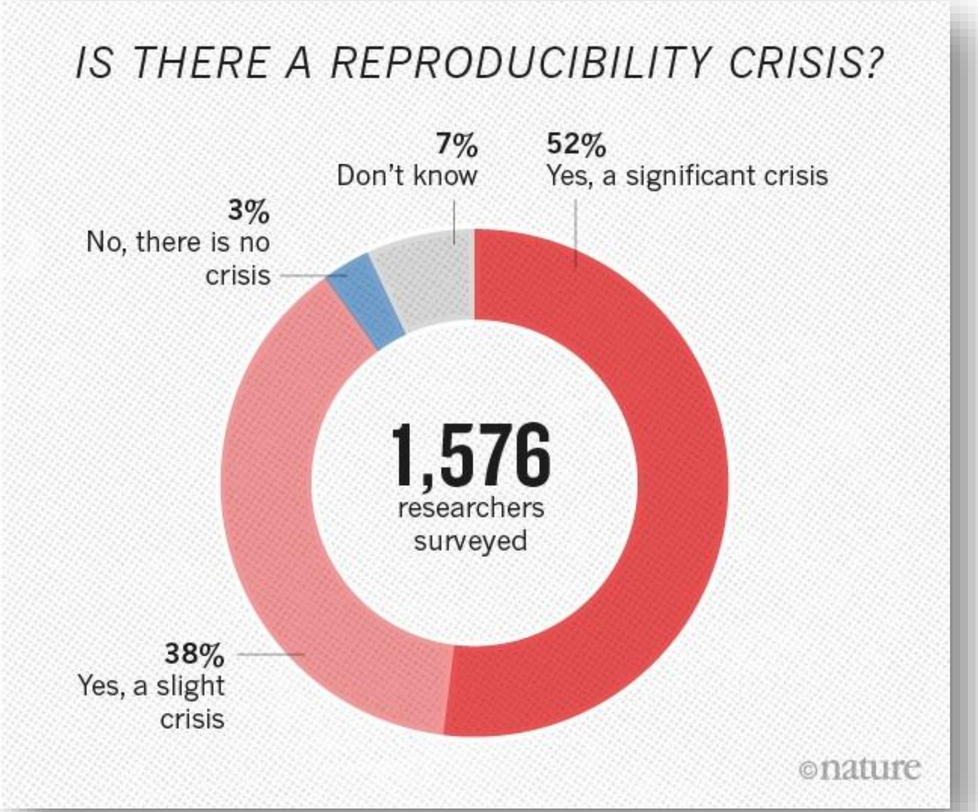
Introduction

Reproductibilité et Open Science

La science en crise?

1,500 scientists lift the lid on reproducibility

Baker 2016, Nature 533
https://doi.org/10.1038/533452a



Introduction

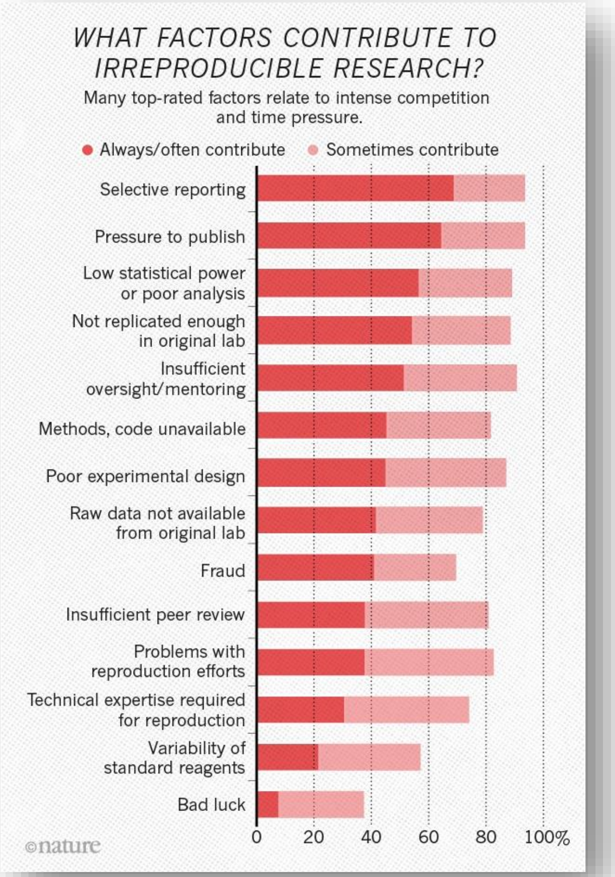
Reproductibilité et Open Science

La science en crise?

1,500 scientists lift the lid on reproducibility

Baker 2016, Nature 533

https://doi.org/10.1038/533452a



Introduction

Reproductibilité et Open Science

Wired

https://www.wired.com/2017/04/want-fix-sciences-replication-crisis-replicate/

MEGAN MEYER OPINION 04.19.17 07:30 AM

WANT TO FIX SCIENCE'S REPLICATION CRISIS? THEN REPLICATE

Introduction

Big Data et Open Data

Quantifying the Data Deluge and the Data Drought

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2984851

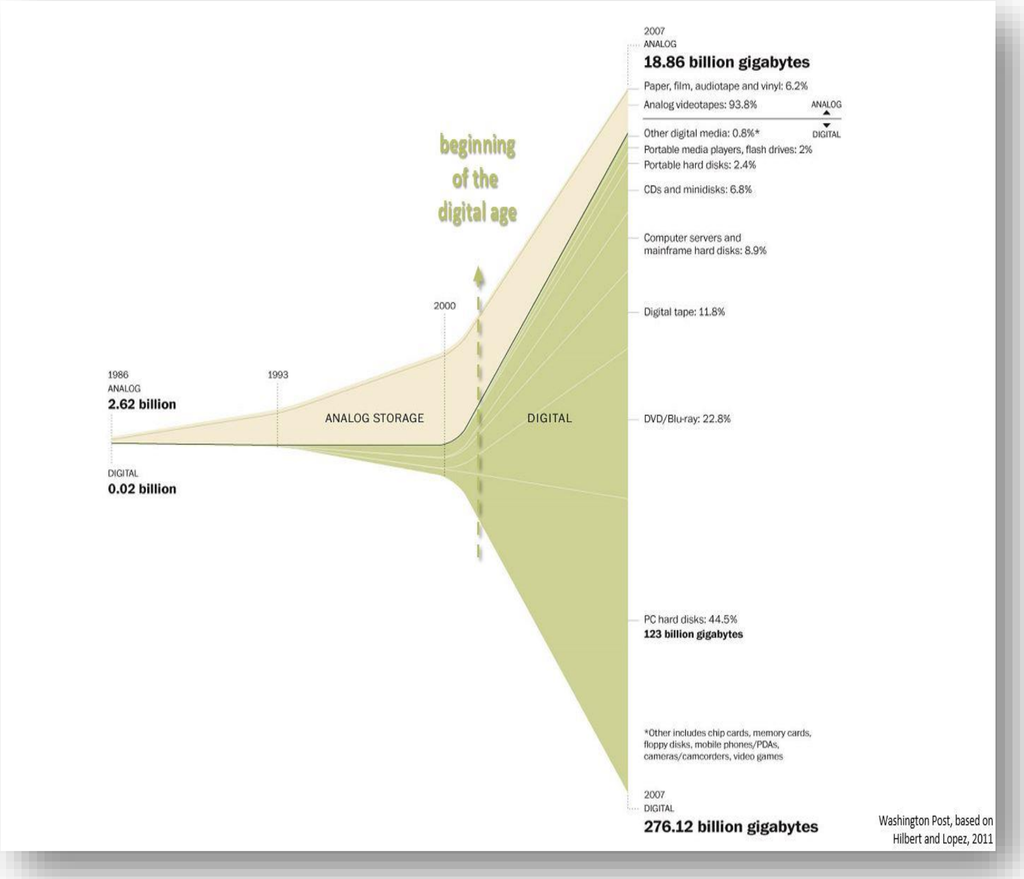
Nombreux réservoirs ouverts

Kaggle : https://www.kaggle.com

Github: http://github.com

WikiData : https://www.wikidata.org

Statistique historique : https://hsso.ch



*Other includes chip cards, memory cards, floppy disks, mobile phones/PDAs, cameras/camcorders, video games

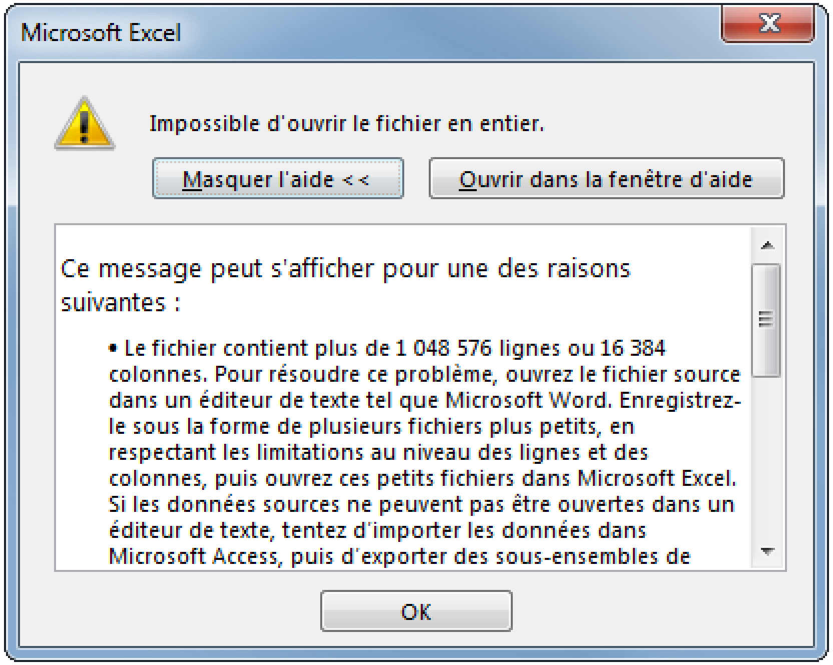
Washington Post, based on Hilbert and Lopez, 2011

Introduction

Excel : limitations

Liste complète :

https://support.office.com/en-us/article/excel-specifications-and-limits-1672b34d-7043-467e-8e27-269d656771c3

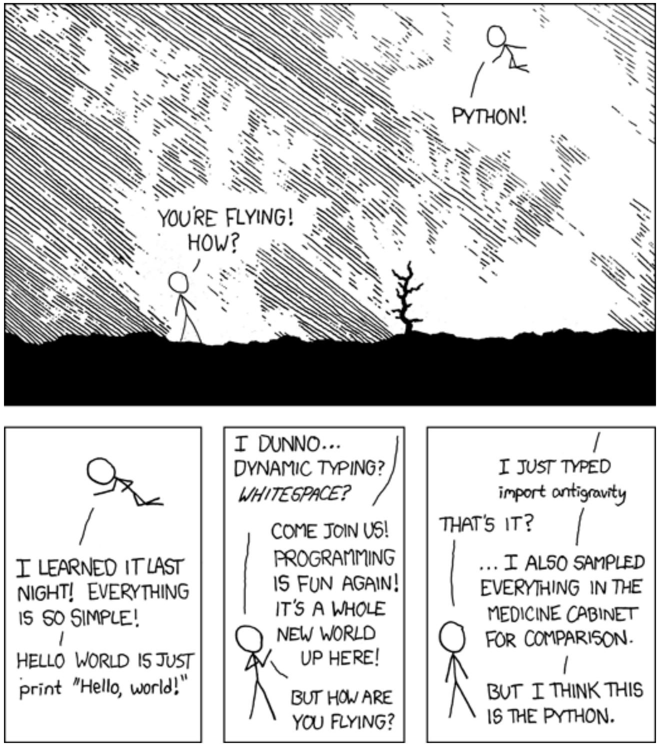


Impossible d'ouvrir le fichier en entier.

Ce message peut s'afficher pour une des raisons suivantes :

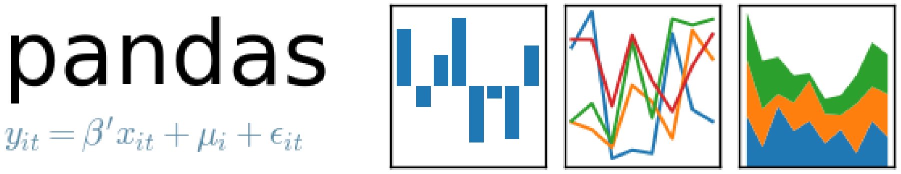
- Le fichier contient plus de 1 048 576 lignes ou 16 384 colonnes. Pour résoudre ce problème, ouvrez le fichier source dans un éditeur de texte tel que Microsoft Word. Enregistrez-le sous la forme de plusieurs fichiers plus petits, en respectant les limitations au niveau des lignes et des colonnes, puis ouvrez ces petits fichiers dans Microsoft Excel. Si les données sources ne peuvent pas être ouvertes dans un éditeur de texte, tentez d'importer les données dans Microsoft Access, puis d'exporter des sous-ensembles de

Introduction


PYTHON!
YOU'RE FLYING! HOW?
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.

https://www.xkcd.com/353/

Introduction



https://pandas.pydata.org

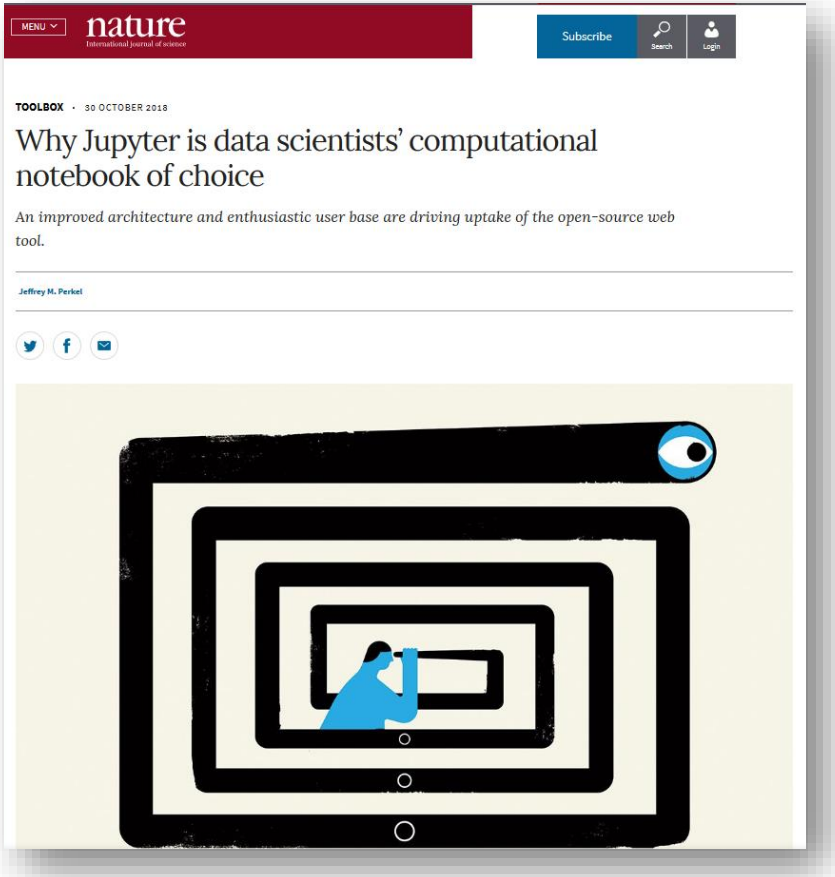
https://www.pinterest.ch/pin/155303887164507907/

Introduction

Reproductibilité et Open Science

Nature

https://www.nature.com/articles/d41586-018-07196-1



TOOLBOX · 30 OCTOBER 2018

Why Jupyter is data scientists' computational notebook of choice

An improved architecture and enthusiastic user base are driving uptake of the open-source web tool.

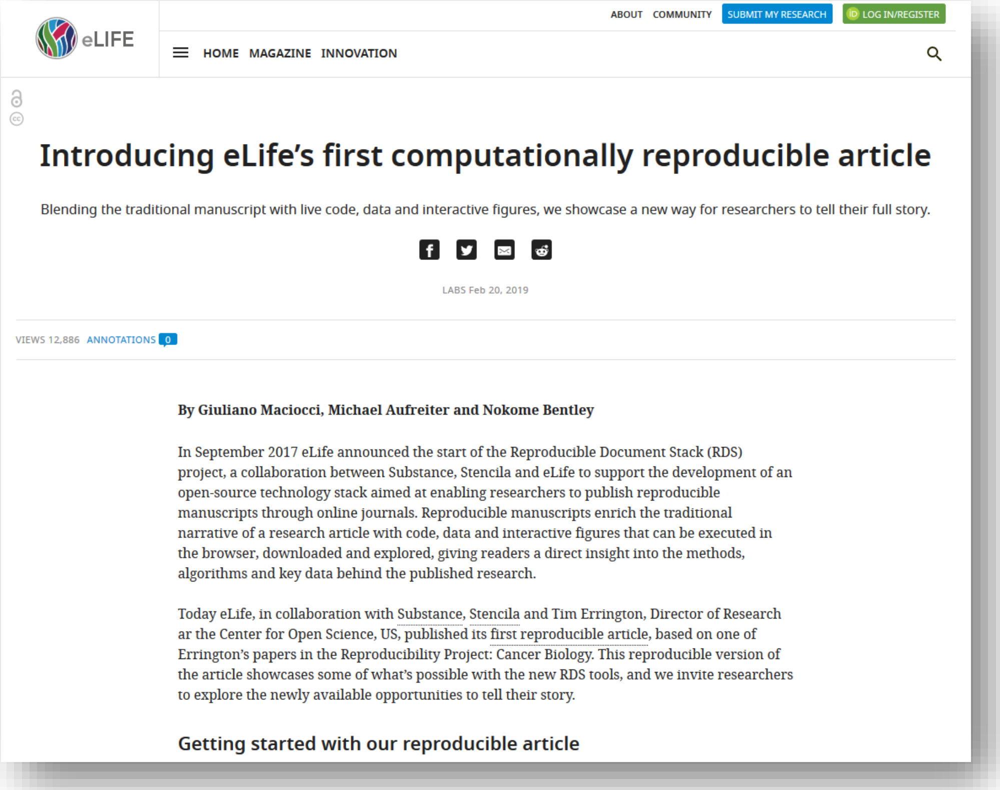
Jeffrey M. Perkel

Introduction

Reproductibilité et Open Science

eLife

https://elifesciences.org/labs/ad58f08d/introducing-elife-s-first-computationally-reproducible-article



Introducing eLife's first computationally reproducible article

Blending the traditional manuscript with live code, data and interactive figures, we showcase a new way for researchers to tell their full story.

LABS Feb 20, 2019

VIEWS 12,886 ANNOTATIONS 0

By Giuliano Maciocci, Michael Aufreiter and Nokome Bentley

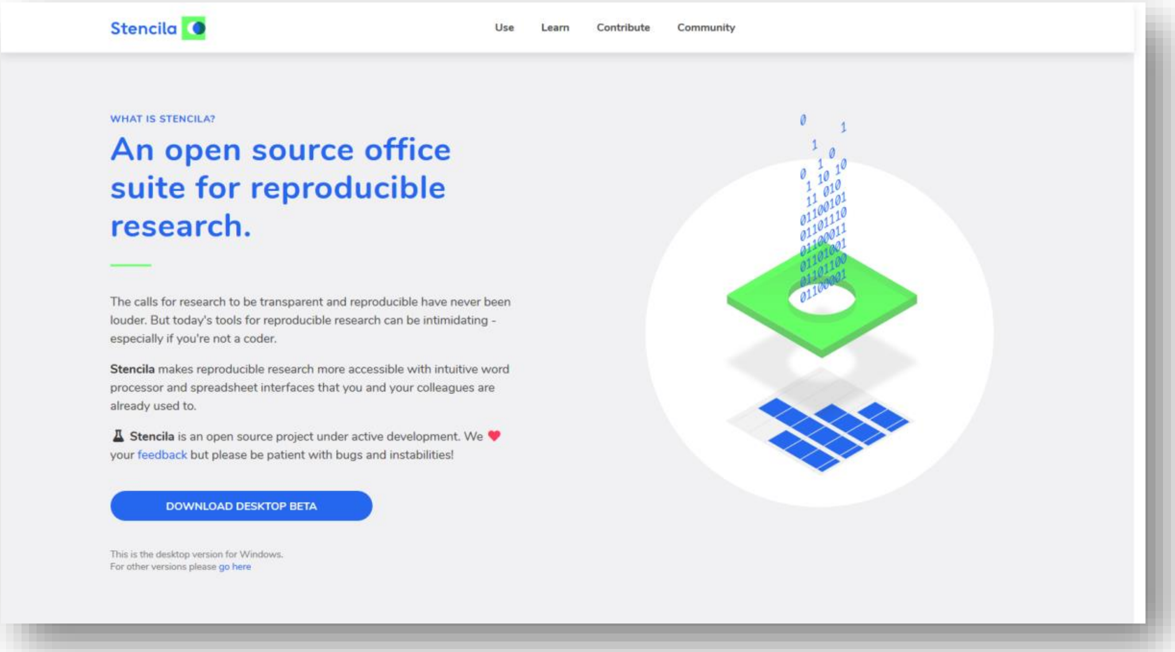
In September 2017 eLife announced the start of the Reproducible Document Stack (RDS) project, a collaboration between Substance, Stencila and eLife to support the development of an open-source technology stack aimed at enabling researchers to publish reproducible manuscripts through online journals. Reproducible manuscripts enrich the traditional narrative of a research article with code, data and interactive figures that can be executed in the browser, downloaded and explored, giving readers a direct insight into the methods, algorithms and key data behind the published research.

Today eLife, in collaboration with Substance, Stencila and Tim Errington, Director of Research ar the Center for Open Science, US, published its first reproducible article, based on one of Errington's papers in the Reproducibility Project: Cancer Biology. This reproducible version of the article showcases some of what's possible with the new RDS tools, and we invite researchers to explore the newly available opportunities to tell their story.

Getting started with our reproducible article

Introduction

Reproductibilité et Open Science



WHAT IS STENCILA?

An open source office suite for reproducible research.

The calls for research to be transparent and reproducible have never been louder. But today's tools for reproducible research can be intimidating - especially if you're not a coder.

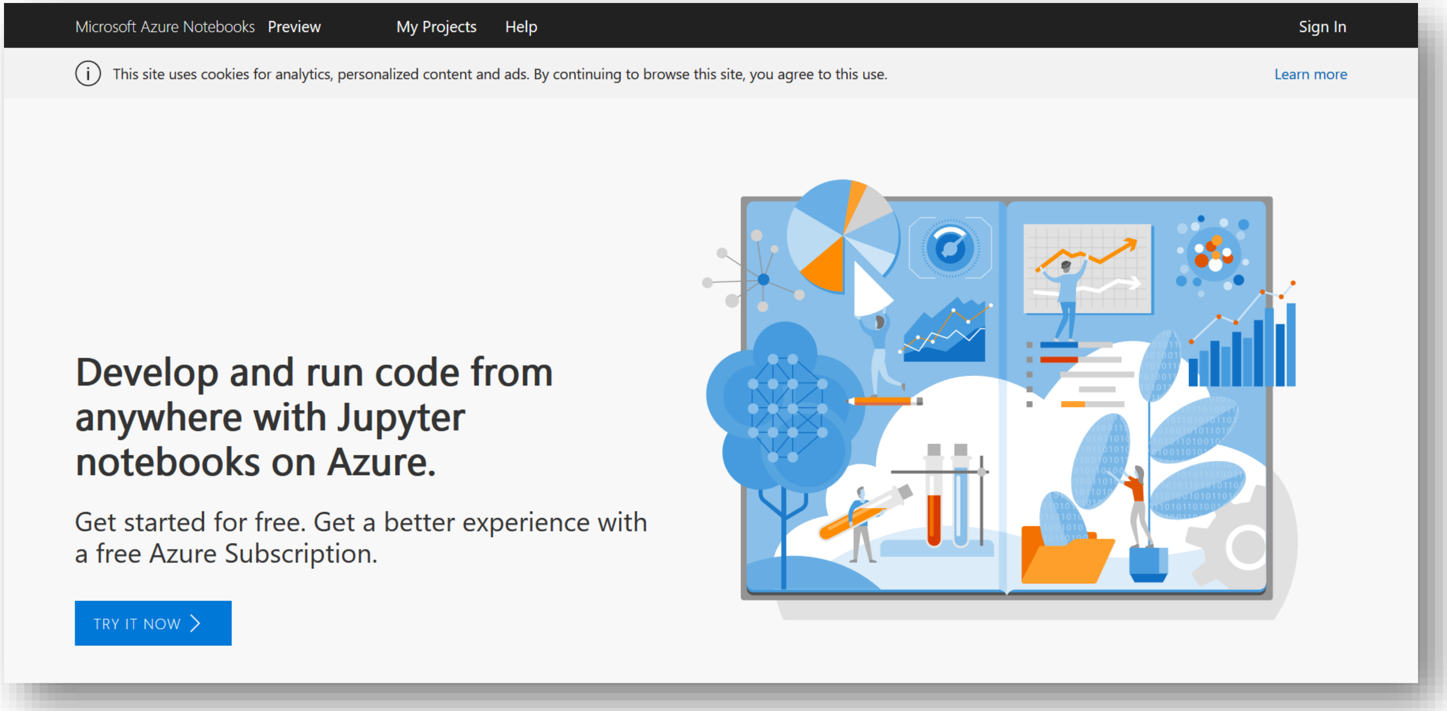
Stencila makes reproducible research more accessible with intuitive word processor and spreadsheet interfaces that you and your colleagues are already used to.

🧪 **Stencila** is an open source project under active development. We ❤ your feedback but please be patient with bugs and instabilities!

This is the desktop version for Windows.
For other versions please go here

https://stenci.la

Introduction



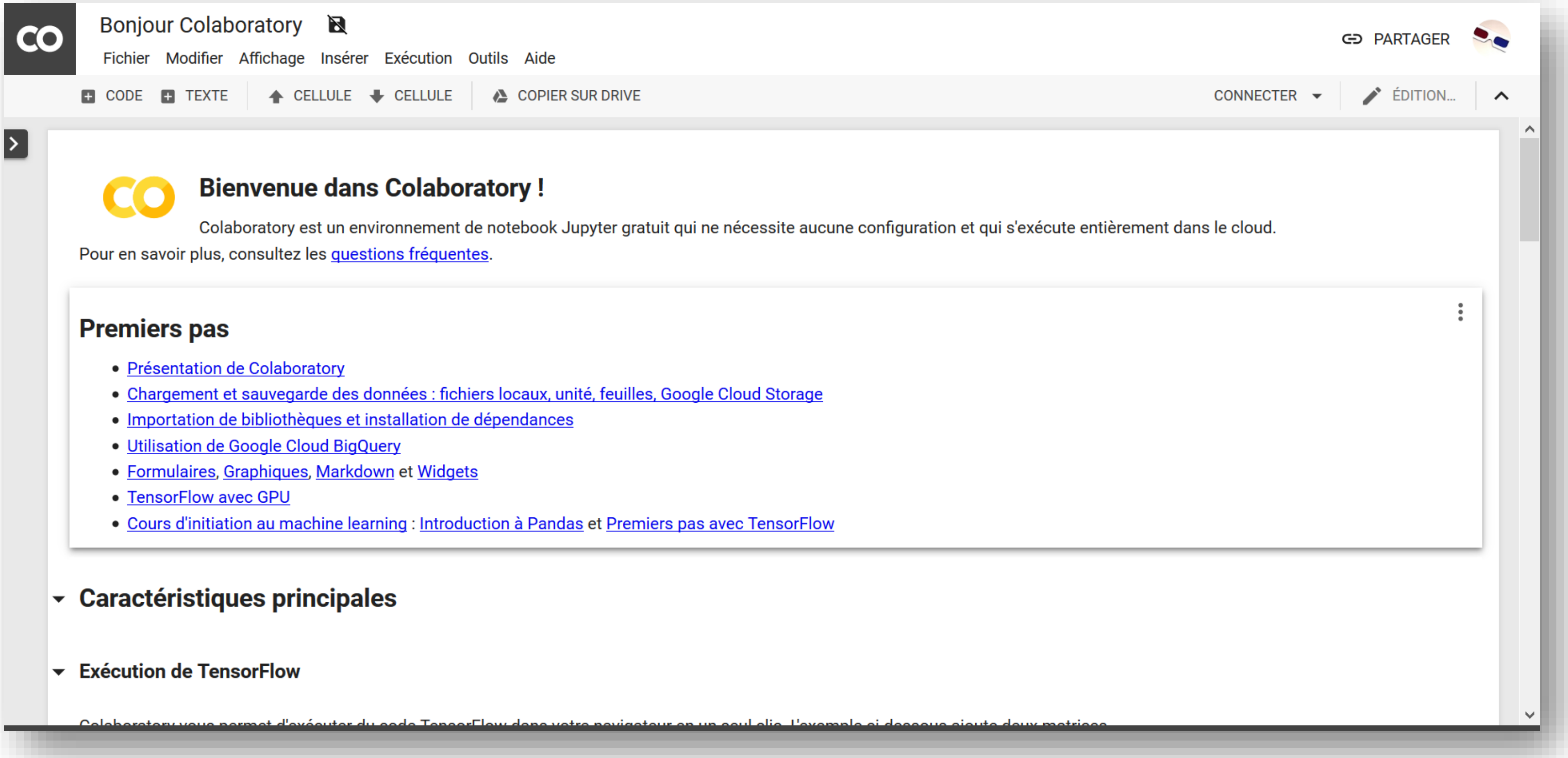
ⓘ This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use. Learn more

Develop and run code from anywhere with Jupyter notebooks on Azure.

Get started for free. Get a better experience with a free Azure Subscription.

https://notebooks.azure.com/

Introduction



Bienvenue dans Colaboratory !

Colaboratory est un environnement de notebook Jupyter gratuit qui ne nécessite aucune configuration et qui s'exécute entièrement dans le cloud. Pour en savoir plus, consultez les questions fréquentes.

Premiers pas

- Présentation de Colaboratory
- Chargement et sauvegarde des données : fichiers locaux, unité, feuilles, Google Cloud Storage
- Importation de bibliothèques et installation de dépendances
- Utilisation de Google Cloud BigQuery
- Formulaires, Graphiques, Markdown et Widgets
- TensorFlow avec GPU
- Cours d'initiation au machine learning : Introduction à Pandas et Premiers pas avec TensorFlow

Caractéristiques principales

Exécution de TensorFlow

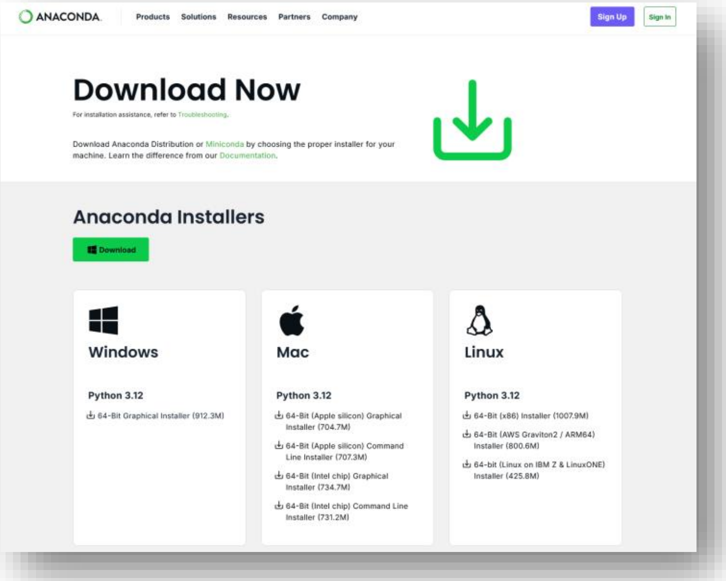
Colaboratory vous permet d'exécuter du code TensorFlow dans votre navigateur en un seul clic. L'exemple ci-dessous ajoute deux matrices

https://colab.research.google.com

Jupyter Notebooks

Installer Jupyter Notebooks et Pandas sur son poste personnel avec la distribution « Anaconda» :

https://www.anaconda.com/download/



Download Now

For installation assistance, refer to Troubleshooting.

Download Anaconda Distribution or Miniconda by choosing the proper installer for your machine. Learn the difference from our Documentation.

Anaconda Installers

Windows

Python 3.12

- 64-Bit Graphical Installer (912.3M)

Mac

Python 3.12

- 64-Bit (Apple silicon) Graphical Installer (704.7M)
- 64-Bit (Apple silicon) Command Line Installer (707.3M)
- 64-Bit (Intel chip) Graphical Installer (734.7M)
- 64-Bit (Intel chip) Command Line Installer (731.2M)

Linux

Python 3.12

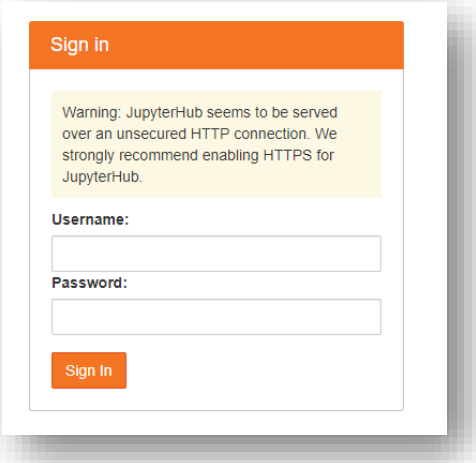
- 64-Bit (x86) Installer (1007.9M)
- 64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)
- 64-bit (Linux on IBM Z & LinuxONE) Installer (425.8M)

Jupyter Notebooks

Travail sur le JupyterHub du cours

Se connecter sur cette adresse avec le login/pwd fourni pendant le cours :

https://learning.jupyterhub.unige.ch



Sign in

Warning: JupyterHub seems to be served over an unsecured HTTP connection. We strongly recommend enabling HTTPS for JupyterHub.

Username:

Password:

Jupyter Notebooks

Packages compris dans l’installation :

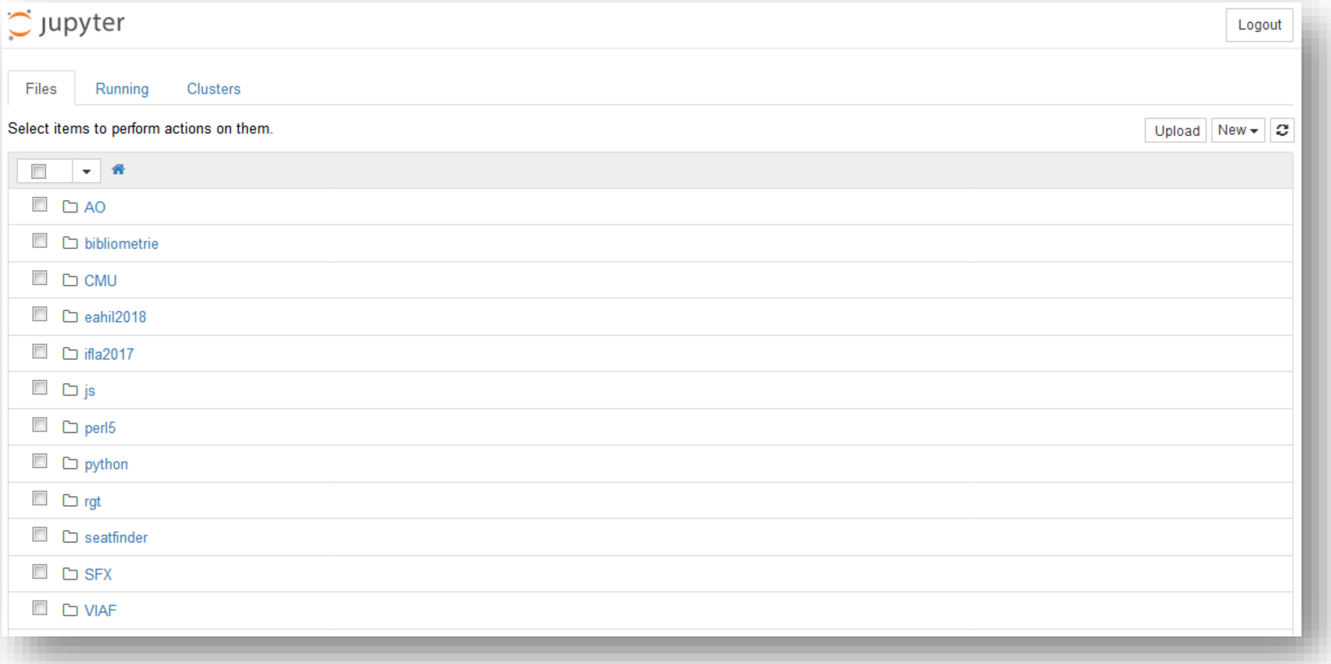
- Notebook (jupyter)
- Pandas
- NumPy
- Matplotlib
- NLTK
- …

Liste complète : https://docs.anaconda.com/anaconda/pkg-docs/

Jupyter Notebooks

Créer, organiser et partager des notebooks

Lancer Anaconda -> Jupyter Notebook



Select items to perform actions on them.

- □ AO
- □ bibliometrie
- □ CMU
- □ eahil2018
- □ ifla2017
- □ js
- □ perl5
- □ python
- □ rgt
- □ seatfinder
- □ SFX
- □ VIAF

Jupyter Notebooks

Si besoin : créer un lien symbolique entre le « home » et le dossier avec les notebooks

1. Avec le shell se positionner sur le «home»
2. Créer le lien avec la commande :

```
mklink /D Nom-du-lien Dossier-de-destination
```

Aide : https://www.howtogeek.com/howto/16226/complete-guide-to-symbolic-links-symlinks-on-windows-or-linux/

Jupyter Notebooks

Se familiariser avec les notebooks

Exercices

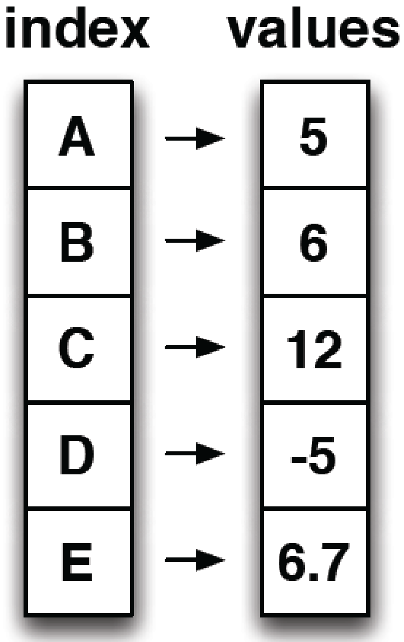
1. Ouvrir un notebook d’exemple (sur le dossier du cours)
2. Créer un nouveau notebook et le renommer
3. Ajouter une cellule de texte (markdown)
4. Ajouter une cellule de code python (calcul simple)
5. L’exporter en format HTML

Aide markdown : https://guides.github.com/features/mastering-markdown/

Aide python : https://www.stavros.io/tutorials/python/

Pandas

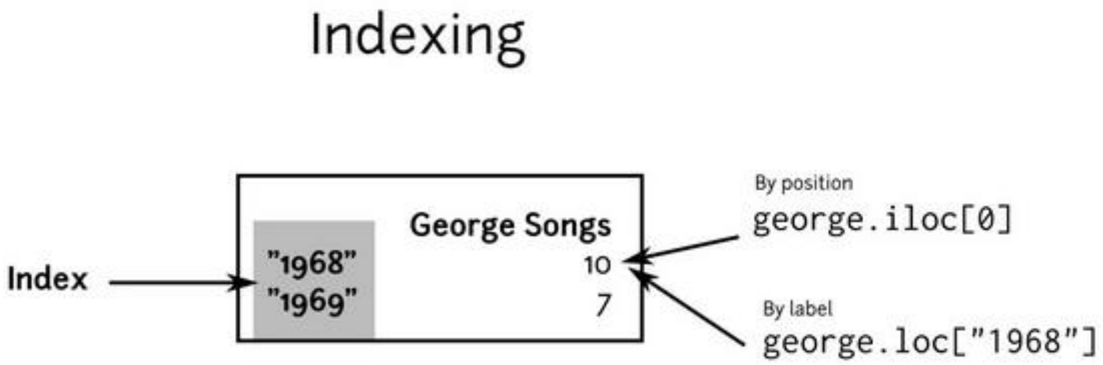
Series : 1 dimension



index	values
A	5
B	6
C	12
D	-5
E	6.7

Pandas

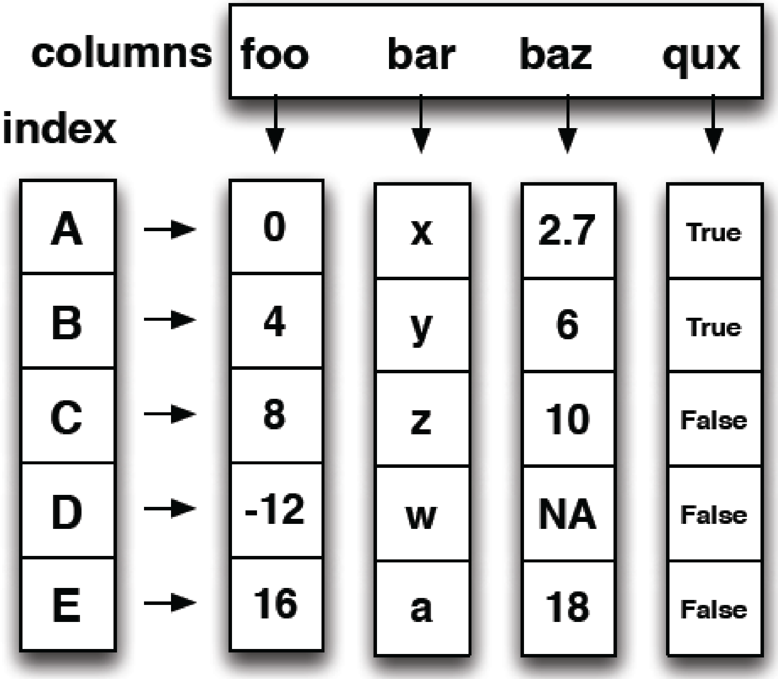
Index : afficher des données par la position ou le nom de l’index



```
george.iloc[0]
george.loc["1968"]
```

Pandas

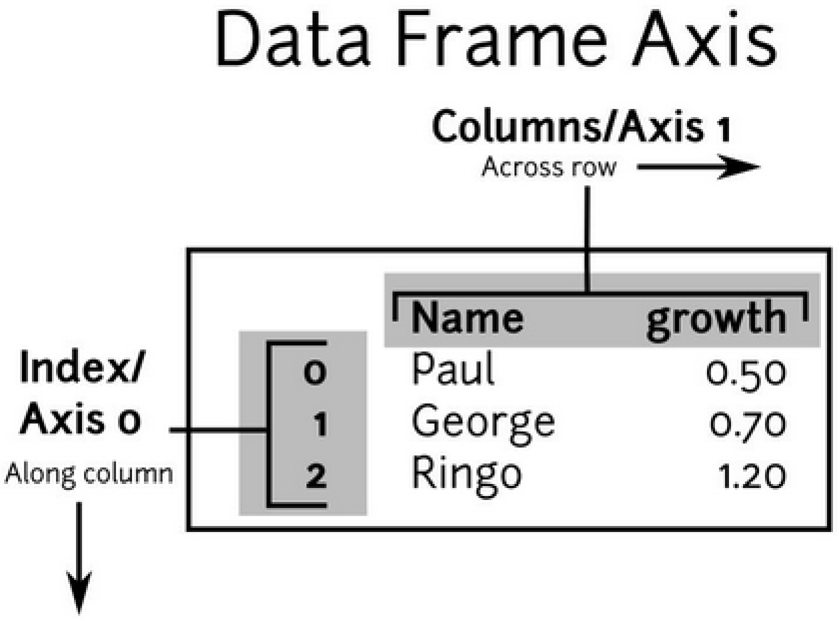
DataFrame : 2 dimensions



index	foo	bar	baz	qux
A	0	x	2.7	True
B	4	y	6	True
C	8	z	10	False
D	-12	w	NA	False
E	16	a	18	False

Pandas

DataFrame : axes

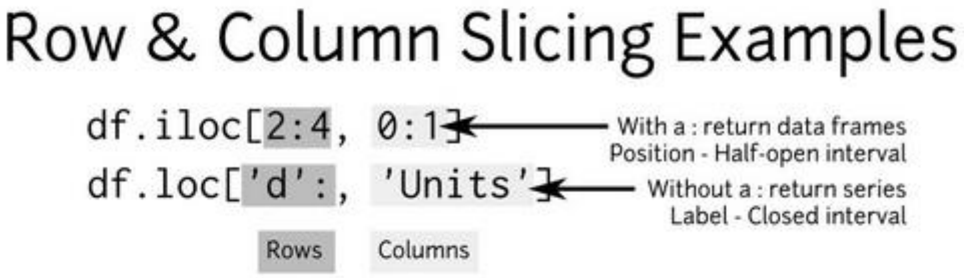


Data Frame Axis

	Name	growth
0	Paul	0.50
1	George	0.70
2	Ringo	1.20

Pandas

DataFrame : slices



Row & Column Slicing Examples

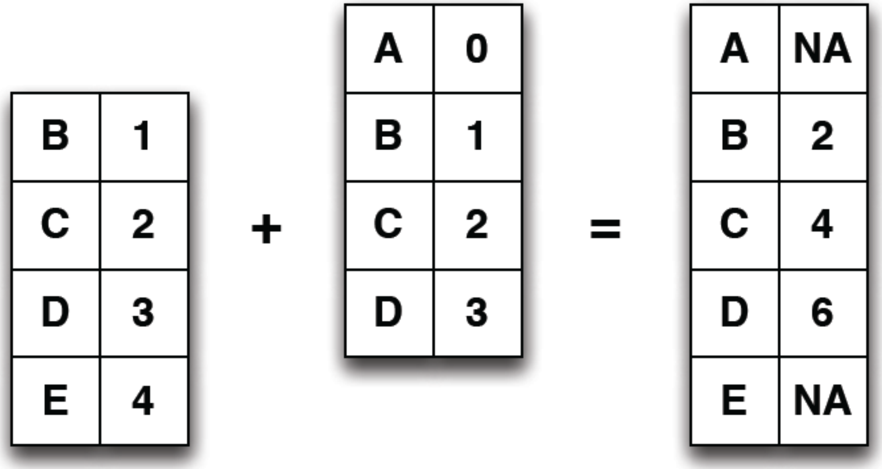
```
df.iloc[2:4, 0:1]
df.loc['d':, 'Units']
```

With a : return data frames
Position - Half-open interval

Without a : return series
Label - Closed interval

Pandas

Opérations facilitées par les index : jointures automatiques



B	1
C	2
D	3
E	4

+

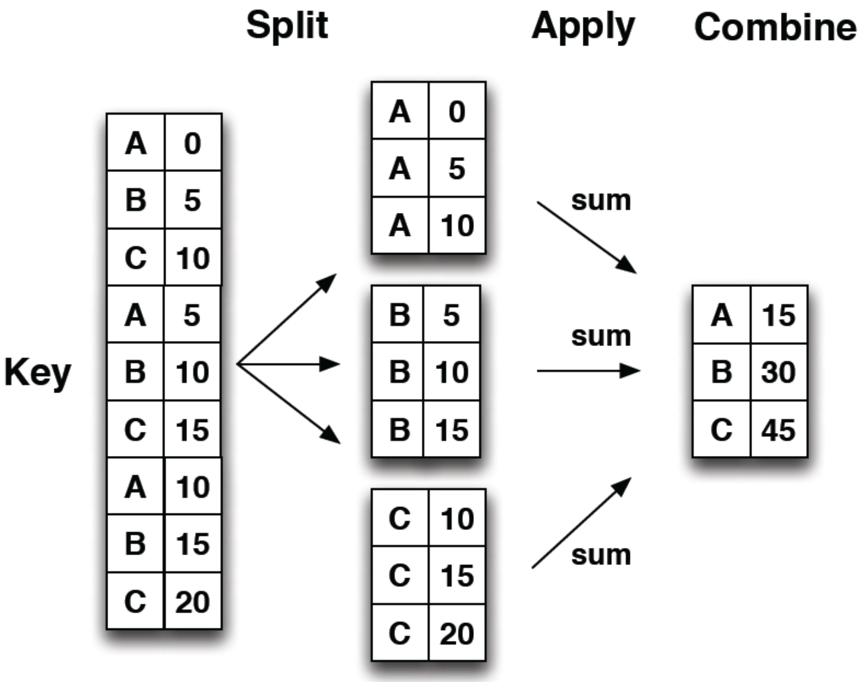
A	0
B	1
C	2
D	3

=

A	NA
B	2
C	4
D	6
E	NA

Pandas

Opérations : GroupBy



Split

Apply

Combine

Key

A	0
B	5
C	10
A	5
B	10
C	15
A	10
B	15
C	20

A	0
A	5
A	10

B	5
B	10
B	15

C	10
C	15
C	20

sum

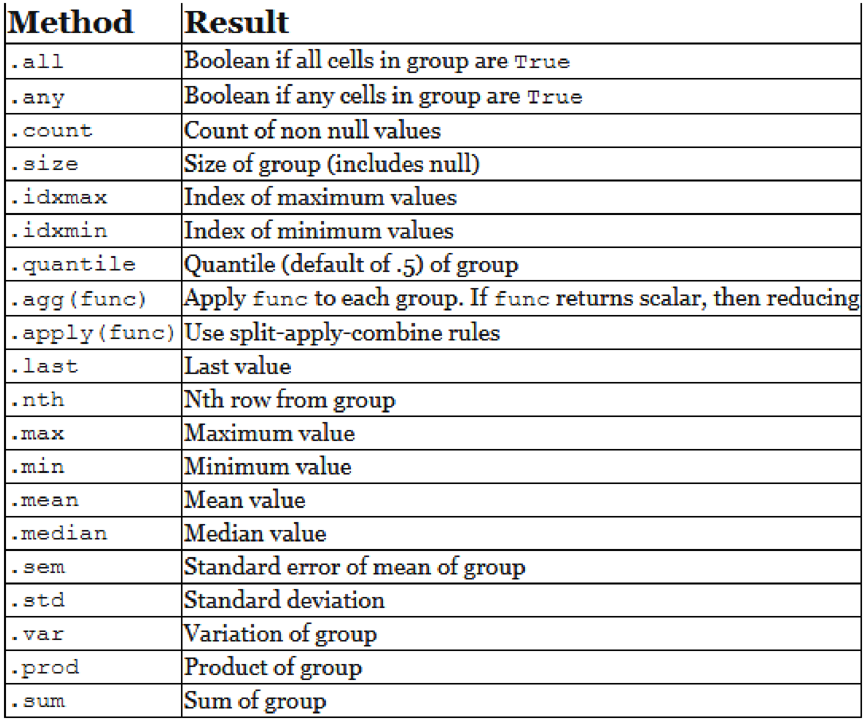
sum

sum

A	15
B	30
C	45

Pandas

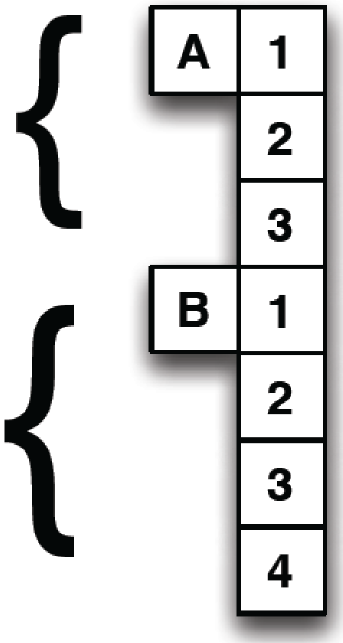
Opérations : GroupBy



Method	Result
`.all`	Boolean if all cells in group are `True`
`.any`	Boolean if any cells in group are `True`
`.count`	Count of non null values
`.size`	Size of group (includes null)
`.idxmax`	Index of maximum values
`.idxmin`	Index of minimum values
`.quantile`	Quantile (default of .5) of group
`.agg(func)`	Apply `func` to each group. If `func` returns scalar, then reducing
`.apply(func)`	Use split-apply-combine rules
`.last`	Last value
`.nth`	Nth row from group
`.max`	Maximum value
`.min`	Minimum value
`.mean`	Mean value
`.median`	Median value
`.sem`	Standard error of mean of group
`.std`	Standard deviation
`.var`	Variation of group
`.prod`	Product of group
`.sum`	Sum of group

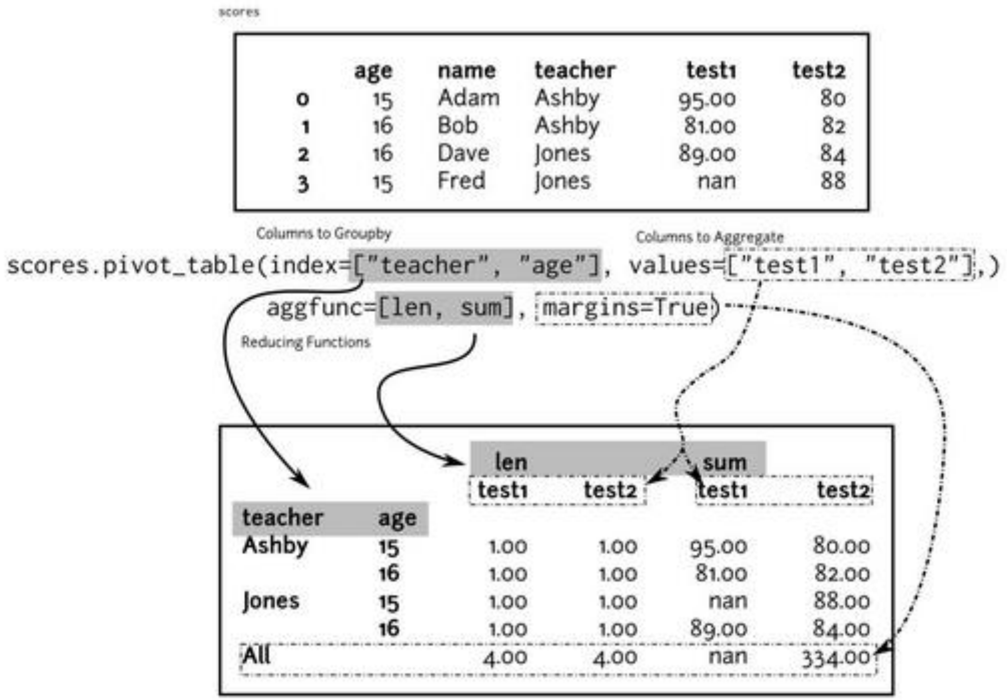
Pandas

Index multidimensionnels



Pandas

Pivoter les tables



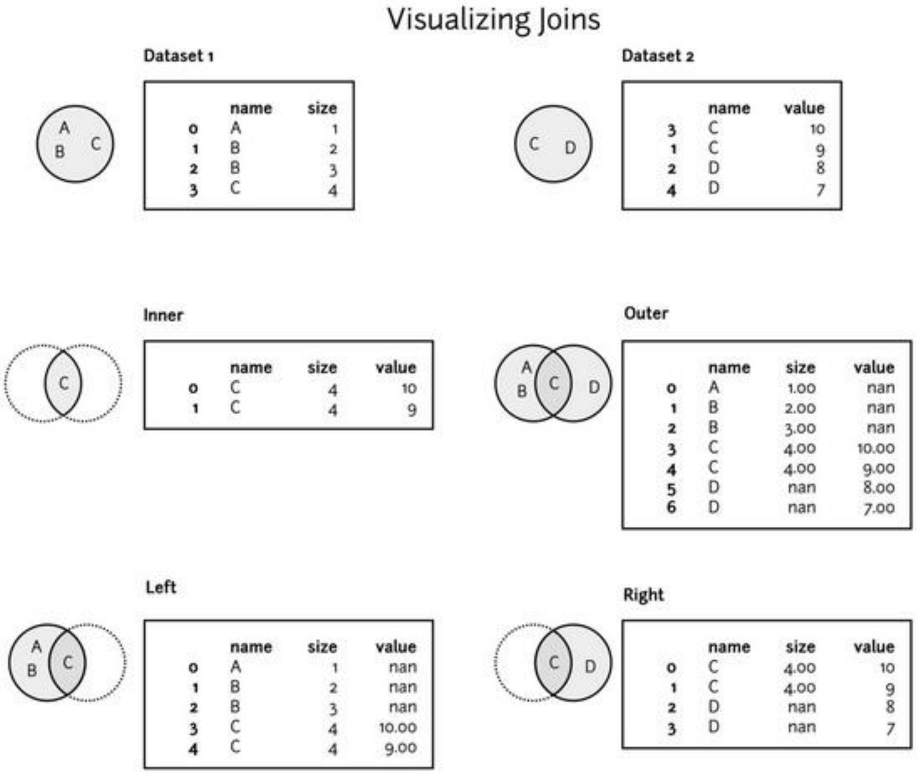
	age	name	teacher	test1	test2
0	15	Adam	Ashby	95.00	80
1	16	Bob	Ashby	81.00	82
2	16	Dave	Jones	89.00	84
3	15	Fred	Jones	nan	88

```
scores.pivot_table(index=["teacher", "age"], values=["test1", "test2"],)
    aggfunc=[len, sum], margins=True)
```

		len		sum	
		test1	test2	test1	test2
teacher	age				
Ashby	15	1.00	1.00	95.00	80.00
	16	1.00	1.00	81.00	82.00
Jones	15	1.00	1.00	nan	88.00
	16	1.00	1.00	89.00	84.00
All		4.00	4.00	nan	334.00

Pandas

Jointures (merge)



Visualizing Joins

Dataset 1

	name	size
0	A	1
1	B	2
2	B	3
3	C	4

Dataset 2

	name	value
3	C	10
1	C	9
2	D	8
4	D	7

Inner

	name	size	value
0	C	4	10
1	C	4	9

Outer

	name	size	value
0	A	1.00	nan
1	B	2.00	nan
2	B	3.00	nan
3	C	4.00	10.00
4	C	4.00	9.00
5	D	nan	8.00
6	D	nan	7.00

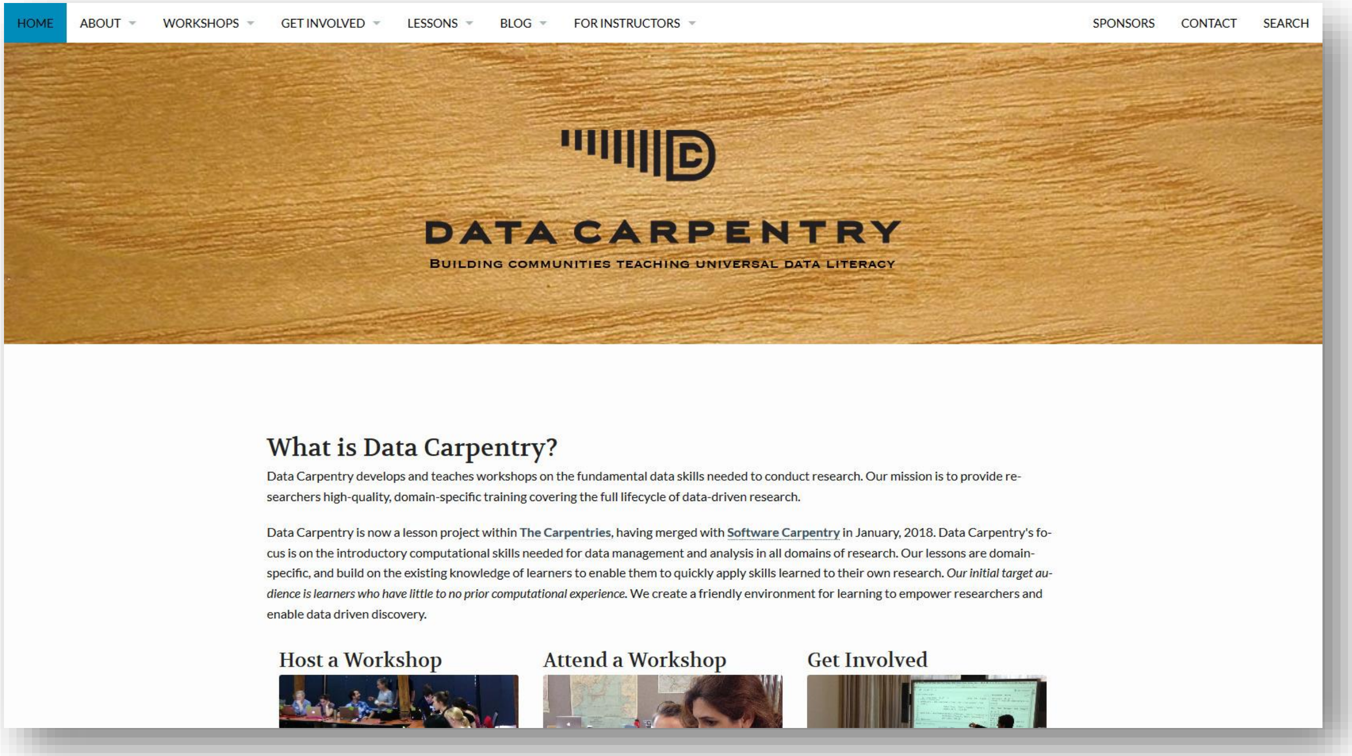
Left

	name	size	value
0	A	1	nan
1	B	2	nan
2	B	3	nan
3	C	4	10.00
4	C	4	9.00

Right

	name	size	value
0	C	4.00	10
1	C	4.00	9
2	D	nan	8
3	D	nan	7

Pour avancer « pas à pas »



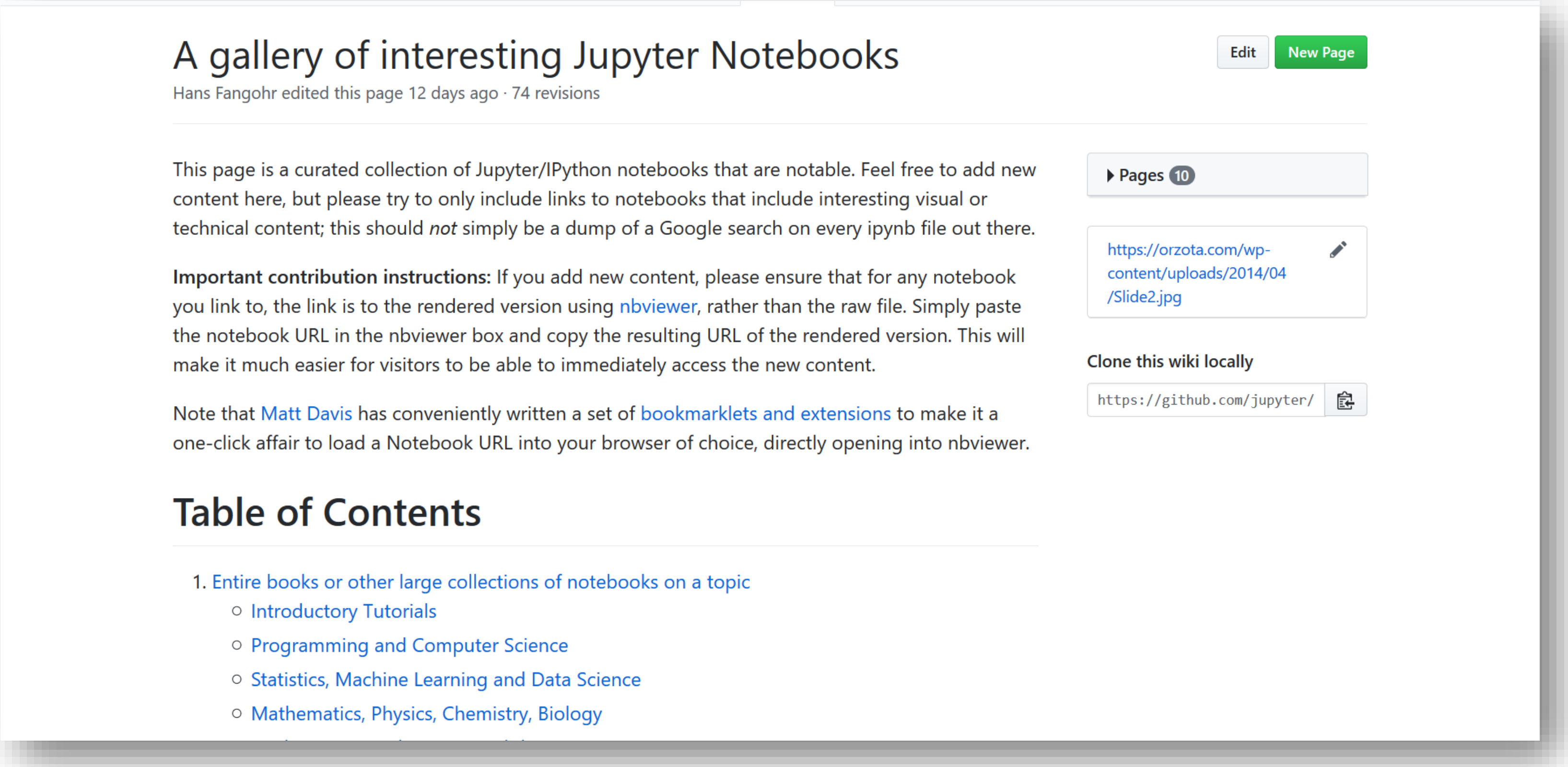
What is Data Carpentry?

Data Carpentry develops and teaches workshops on the fundamental data skills needed to conduct research. Our mission is to provide researchers high-quality, domain-specific training covering the full lifecycle of data-driven research.

Data Carpentry is now a lesson project within The Carpentries, having merged with Software Carpentry in January, 2018. Data Carpentry's focus is on the introductory computational skills needed for data management and analysis in all domains of research. Our lessons are domain-specific, and build on the existing knowledge of learners to enable them to quickly apply skills learned to their own research. *Our initial target audience is learners who have little to no prior computational experience*. We create a friendly environment for learning to empower researchers and enable data driven discovery.

https://datacarpentry.org/python-socialsci/

Pour aller plus loin



A gallery of interesting Jupyter Notebooks

Hans Fangohr edited this page 12 days ago · 74 revisions

This page is a curated collection of Jupyter/IPython notebooks that are notable. Feel free to add new content here, but please try to only include links to notebooks that include interesting visual or technical content; this should *not* simply be a dump of a Google search on every ipynb file out there.

Important contribution instructions: If you add new content, please ensure that for any notebook you link to, the link is to the rendered version using nbviewer, rather than the raw file. Simply paste the notebook URL in the nbviewer box and copy the resulting URL of the rendered version. This will make it much easier for visitors to be able to immediately access the new content.

Note that Matt Davis has conveniently written a set of bookmarklets and extensions to make it a one-click affair to load a Notebook URL into your browser of choice, directly opening into nbviewer.

https://orzota.com/wp-content/uploads/2014/04/Slide2.jpg

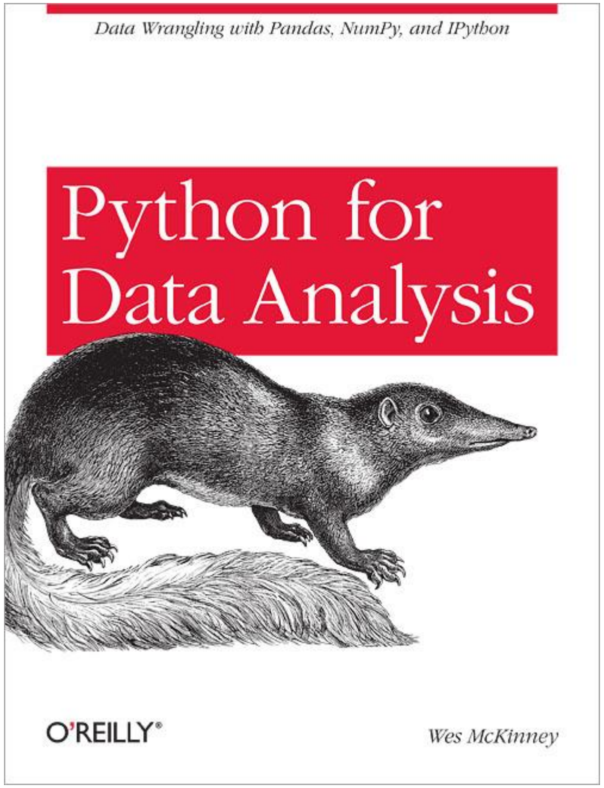
Clone this wiki locally

https://github.com/jupyter/

Table of Contents

https://github.com/jupyter/jupyter/wiki#a-gallery-of-interesting-jupyter-notebooks

Pour aller plus loin

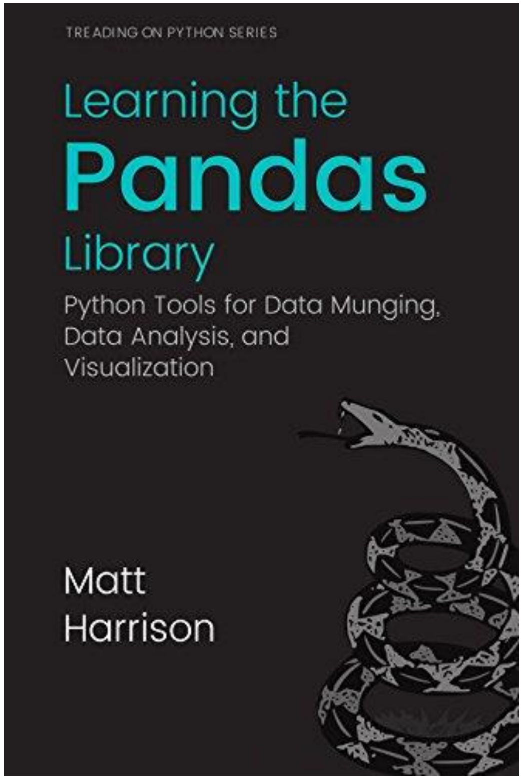


Data Wrangling with Pandas, NumPy, and IPython

Python for Data Analysis

O'REILLY®

Wes McKinney



TREADING ON PYTHON SERIES

Learning the Pandas Library

Python Tools for Data Munging, Data Analysis, and Visualization

Matt Harrison

Pandas

Exercices pratiques disponibles ici :
https://github.com/dis-unige/formations

1. Importer des données
2. Analyser des données
3. Travailler avec différents types de données et des données manquantes
4. Exporter des données
5. Créer des graphiques simples

Aide Pandas : https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html

Sources

Cheat Sheets distribués dans le cours :

- Jupyter notebook :

  https://www.datacamp.com/cheat-sheet/jupyter-notebook-cheat-sheet

- Markdown :

  http://geog.uoregon.edu/bartlein/courses/geog607/Rmd/MDquick-refcard.pdf

- Pandas :

  https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf