

Traitement et analyse de données en Python

Francis Wolinski
Master 2 TIDE – octobre 2021

<https://github.com/fran6w/TIDE>

Agenda

1. Introduction à Python
2. Univers des modules Python
3. Distributions de Python
4. IPython : Interactive Python
5. Gestion des librairies Python
6. Écosystème Python pour la Data Science



1. Introduction à Python

- ▶ Python est un langage informatique né aux Pays-Bas au début des années 90 et a été conçu par Guido van Rossum
- ▶ La version actuelle, Python 3, est éditée par la Python Software Foundation aux États-Unis
- ▶ *“The mission of the Python Software Foundation is to promote, protect, and advance the Python programming language, and to support and facilitate the growth of a diverse and international community of Python programmers.”*
- ▶ La PSF est sponsorisée par une quarantaine de sociétés (USA, Pays-Bas, Royaume-Uni, Belgique, Australie, Singapour...) dont : AWS, Bloomberg, Facebook, Google, Huawei, JP Morgan, LinkedIn, Microsoft, Netflix, O'Reilly, RedHat, Salesforce, Slack...
- ▶ Le langage est distribué avec une licence open source compatible avec la GPL (GNU General Public License)

1. Qualités du langage Python

▶ Interprété

- Les instructions sont traduites en langage machine au fur et à mesure de leur exécution.

▶ Interactif

- Il existe des modes « console » où l'utilisateur peut exécuter des instructions et voir immédiatement le résultat.

▶ Orienté Objets

- Tout objet du langage est instance d'une classe qui définit la structure des objets (variables d'instance), leurs comportements (méthodes) et leur réutilisation (héritage).

▶ Riche

- Nombreuses classes de base (bool, int, float, chaînes, bytes...), nombreuses structures de données (tuple, list, dict, range, set...) et nombreuses bibliothèques.

▶ Syntaxe concise et lisible

```
>>> [i * i for i in range(10)]  
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

1. Qualités du langage Python

- ▶ **Typage dynamique**
 - Chaque objet est implicitement typé et n'a pas à être déclaré au préalable.
- ▶ **Gestion dynamique des ressources**
 - Python dispose de mécanismes gérant automatiquement la mémoire (ramasse miettes, garbage collector), les descripteurs de fichiers...
- ▶ **Extensible et ouvert**
 - Python bénéficie de très nombreuses librairies. Il s'intègre également aux objets COM, .NET, CORBA et aux langages Fortran, C/C++ et Java.
- ▶ **Portable**
 - Linux et autres UNIX
 - Mac OS
 - Windows
 - Autres OS : AS/400, z/OS, iOS, Android, etc.
- ▶ **Open source**
 - Licence compatible GPL
 - Autres licences libres pour certaines librairies !



1. La Licence Python

Terms and conditions for accessing or otherwise using Python

PSF LICENSE AGREEMENT FOR PYTHON 3.8.0

1. This LICENSE AGREEMENT is between the Python Software Foundation ("PSF"), and the Individual or Organization ("Licensee") accessing and otherwise using Python 3.8.0 software in source or binary form and its associated documentation.
2. Subject to the terms and conditions of this License Agreement, PSF hereby grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce, analyze, test, perform and/or display publicly, prepare derivative works, distribute, and otherwise use Python 3.8.0 alone or in any derivative version, provided, however, that PSF's License Agreement and PSF's notice of copyright, i.e., "Copyright © 2001-2019 Python Software Foundation; All Rights Reserved" are retained in Python 3.8.0 alone or in any derivative version prepared by Licensee.
3. In the event Licensee prepares a derivative work that is based on or incorporates Python 3.8.0 or any part thereof, and wants to make the derivative work available to others as provided herein, then Licensee hereby agrees to include in any such work a brief summary of the changes made to Python 3.8.0.
4. PSF is making Python 3.8.0 available to Licensee on an "AS IS" basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 3.8.0 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.
5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON 3.8.0 FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON 3.8.0, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.
6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.
7. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between PSF and Licensee. This License Agreement does not grant permission to use PSF trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.
8. By copying, installing or otherwise using Python 3.8.0, Licensee agrees to be bound by the terms and conditions of this License Agreement.

Le langage est distribué sous une licence open source compatible avec la licence GPL (GNU General Public License)

D'autres licences open source pour certains modules (par ex., MIT, BSD, Apache)

L'open source fournit une certaine liberté mais introduit des contraintes juridiques et techniques

1. Petite histoire de Python

- ▶ Les années 1990 : Python 1
 - Langage de scripts
 - Alternative à *bash*
- ▶ Les années 2000 : Python 2 (2.7* arrêt en avril 2020)
 - Calcul scientifique
 - Alternative à *MATLAB*
- ▶ Les années 2010 : Python 3 (3.10)
 - Data science
 - Alternative à *R*

Source: Jake VanderPlas (University of Washington)

1. Les deux versions de Python

▶ Python 2

- Ancienne version
- Dernière version Python 2.7

▶ Python 3

- Unicode
- Fonction : print()
- Divisions réelles avec des entiers : $3/2$ vs $3//2$
- Divers changements : exceptions, méthodes des dictionnaires, range, input, types...
- Refonte des librairies
- La plupart des librairies disponibles en Python 2 ont été portées en Python 3
- Python 3.8 sorti en octobre 2019
- Python 3.9 sorti en novembre 2020
- Python 3.10 sorti en octobre 2021
- ⇒ **Attention au code trouvé sur internet !**
 - Python
 - Versions des librairies

2. Univers des modules Python

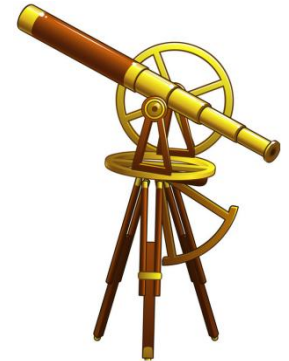
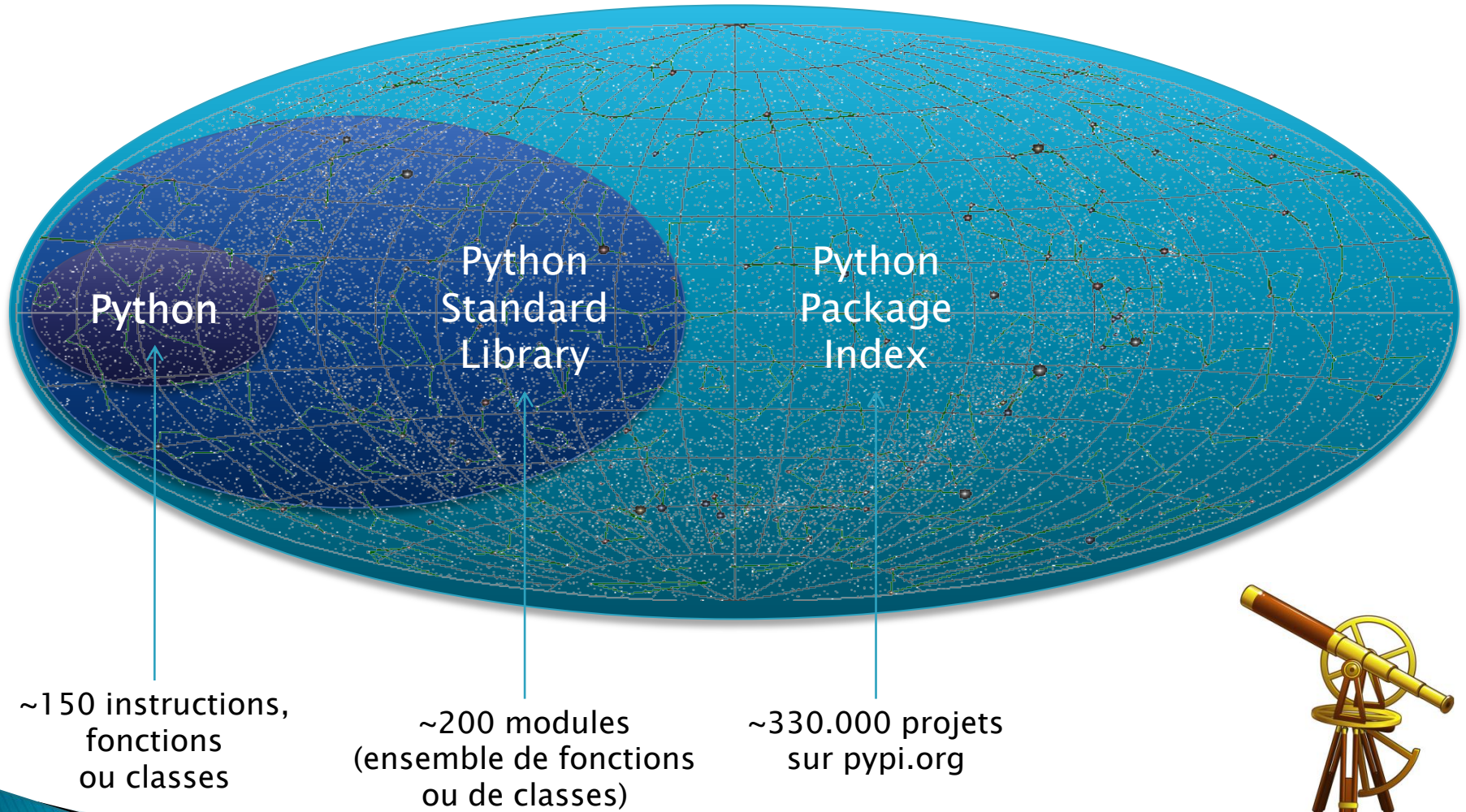
▶ Qu'est-ce qu'un module?

- Un module est un fichier qui contient des définitions et des instructions Python.
- Un module peut contenir des instructions exécutables, des définitions de fonctions ou de classes.
- Les modules peuvent importer d'autres modules.

▶ Exemple: *math* (Python Standard Library)

- Ce module est toujours disponible. Il fournit un accès aux fonctions mathématiques définies par le standard C.
 - Fonctions standard: e.g. *math.factorial(x)*
 - Puissances et logarithmes: e.g. *math.exp(x)*, *math.pow(x, y)*, *math.sqrt(x)*
 - Fonctions trigonométriques : e.g. *math.cos(x)*, *math.sin(x)*, *math.tan(x)*
 - Conversion des angles, fonctions hyperboliques, fonctions spéciales
 - Constantes: par ex. *math.pi*, *math.e*

2. Univers des modules Python



3. Distributions de Python



Différentes distributions de Python ont sélectionné et packagé des modules pour offrir un environnement de calcul scientifique et de data science performant.



Distribution open source originale : www.python.org
Accès à PyPI avec l'utilitaire *pip*



Accelerate your data science and software development with a complete, secure and supported Python distribution.



The Most Popular Python Data Science Distribution.



Canopy is tailor-made for the workflows of scientists and engineers, combining a streamlined integrated analysis environment with over 450 proven scientific and analytic Python packages.



Python(x,y) is a free scientific and engineering development software for numerical computations, data analysis and data visualization based on Python programming language, Qt graphical user interfaces and Spyder interactive scientific development environment.

3. Distribution Anaconda

- ▶ Distribution commerciale
 - Version gratuite pour les individus et les petites entreprises
 - Version 3.8 : 639 modules, 328 modules préinstallés
- ▶ Site web
 - <https://anaconda.com/> (ex. Continuum Analytics)

3. Distribution Anaconda

- ▶ Multiplateformes
 - Windows 32-bit et 64-bit
 - Mac OS 64-bit
 - Linux 32-bit et 64-bit
- ▶ Intégration système aisée
 - Installation dans un répertoire unique
 - Ne nécessite pas de droits d'administration
 - Compatible avec d'autres installations de Python
- ▶ Utilitaire *conda* Python Package Manager
 - Installation et mise à jour de modules

3. Succès actuel de Python

IEEE Spectrum: Python is the top programming language of 2020



By Ryan Daws | 27th July 2020 | TechForge Media

Editor at TechForge Media. Often sighted at global tech conferences with a coffee in one hand and laptop in the other. If it's geeky, I'm probably into it.



The latest IEEE Spectrum shows Python retaining its lead to be the top programming language of 2020.

First created 30 years ago in 1990, Python's popularity has surged in recent years thanks to it being ideally suited for artificial intelligence and web development. Many schools also teach Python because it's considered a relatively easy language for beginners.

Rank	Language	Type	Score
1	Python	Web, Mobile, Embedded	100.0
2	Java	Web, Mobile	95.3
3	C	Mobile, Embedded	94.6
4	C++	Mobile, Embedded	87.0
5	JavaScript	Web	79.5
6	R	Web	78.6
7	Arduino	Embedded	73.2
8	Go	Web, Mobile	73.1
9	Swift	Mobile	70.5
10	Matlab	Web	68.4
11	Ruby	Web, Mobile	66.8

Source:

developer-tech.com/news/2020/jul/27/ieee-spectrum-python-top-programming-language-2020/

Language Types

Web	Enterprise
Mobile	Embedded

3. Succès actuel de Python



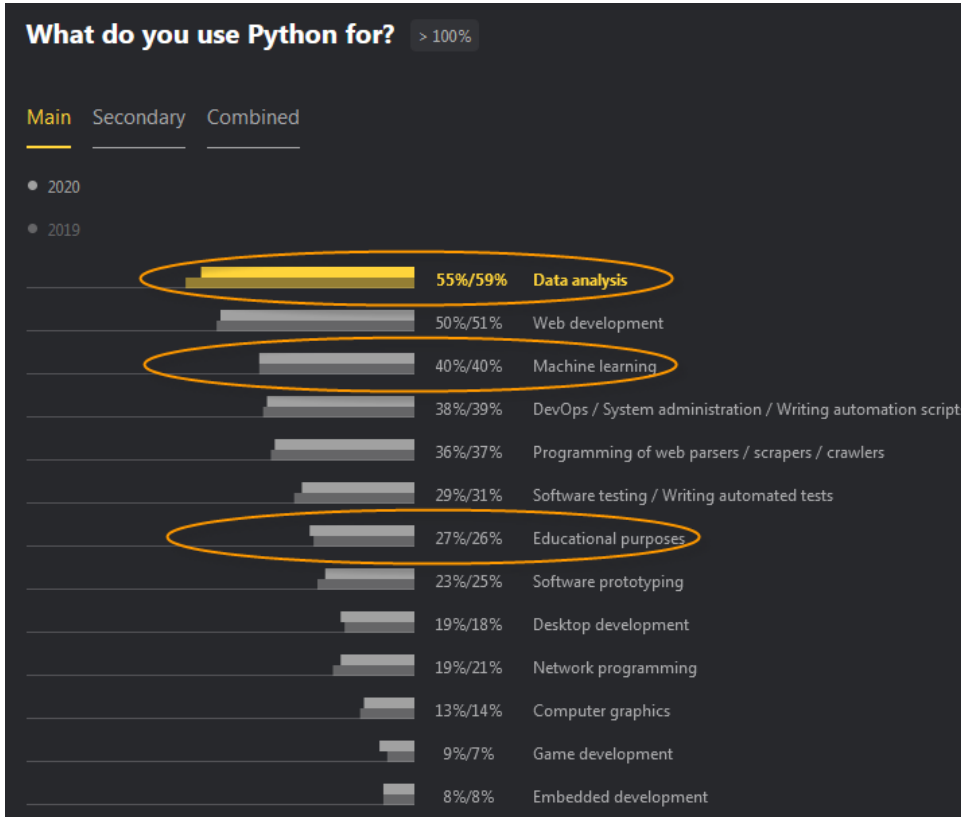
[Download this article as a PDF \(free registration required\).](#)

Source:

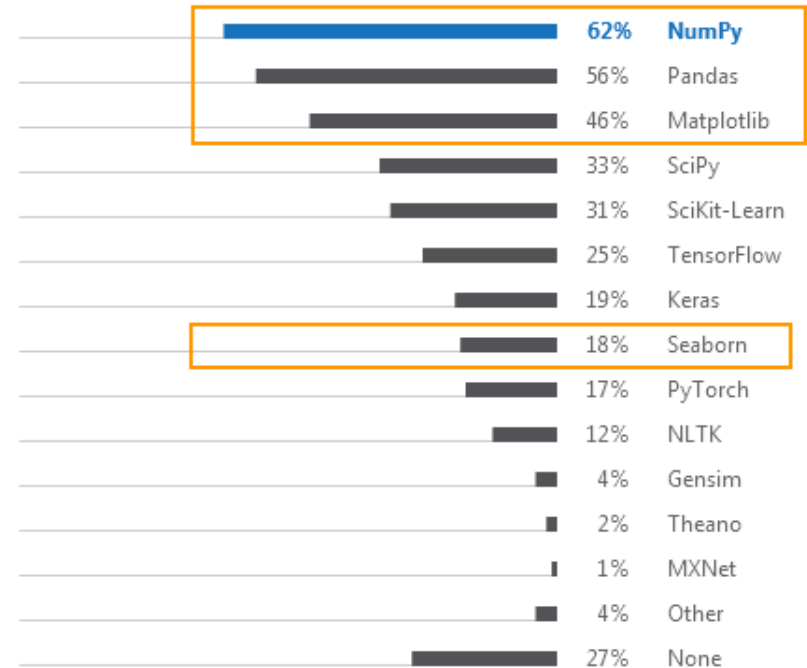
www.zdnet.com/google-amp/article/python-is-eating-the-world-how-one-developers-side-project-became-the-hottest-programming-language-on-the-planet/

"I can remember one person in particular who said, 'You cannot tell anybody that I'm here because our use of Python is a competitive advantage.' It was their secret weapon, right?"

3. Succès actuel de Python



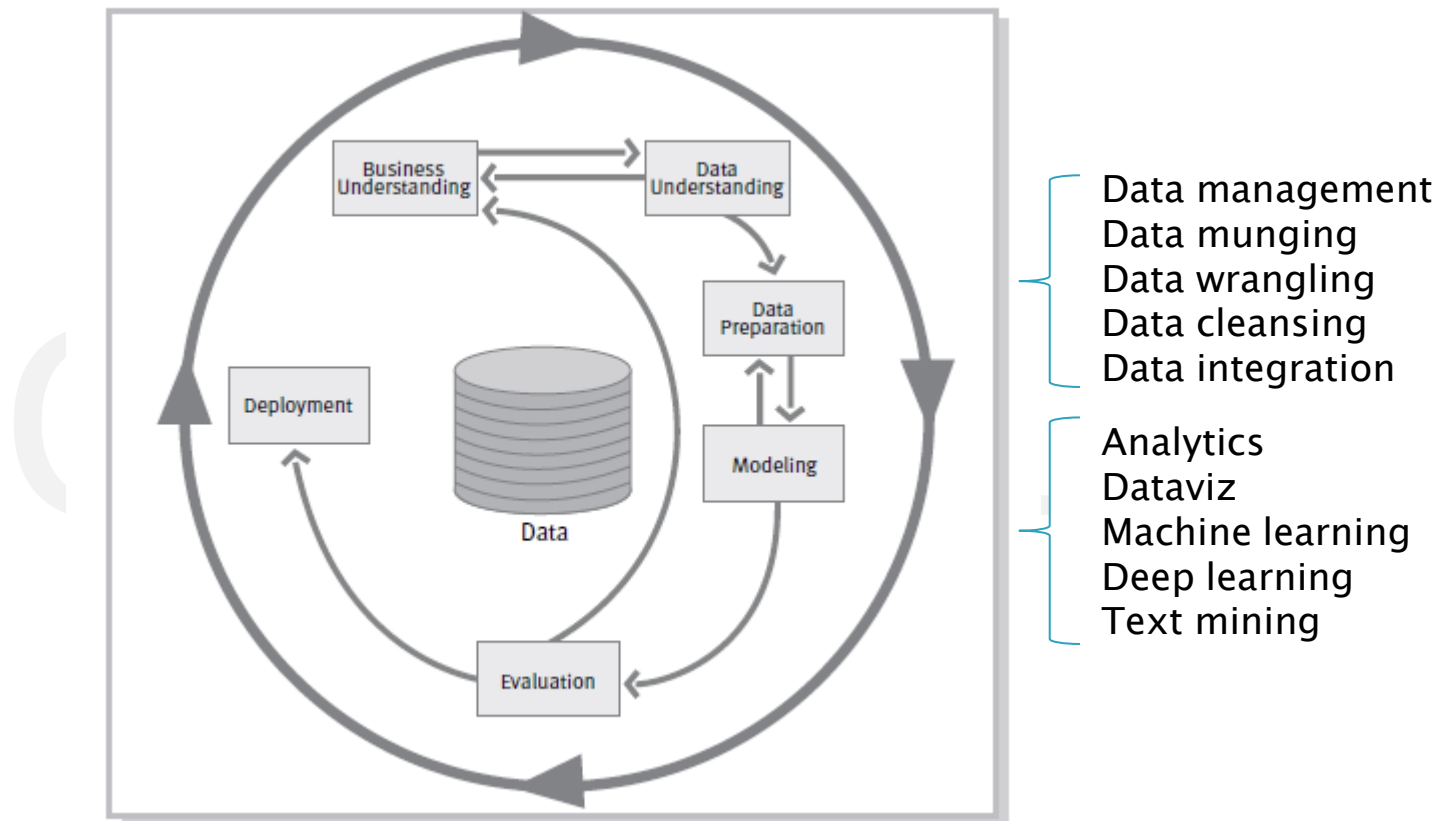
Data science frameworks and libraries > 100%



Source:
<https://www.jetbrains.com/lp/python-developers-survey-2020/>

4. Méthodologie CRISP-DM

Cross Industry Standard Process for Data Mining



Source: CRISP-DM 1.0 SPSS – *Statistical Package for the Social Sciences* – 2000
(acquis par IBM en 2009)

4. Méthodologie CRISP-DM

Cross Industry Standard Process for Data Mining

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives <i>Background</i> <i>Business Objectives</i> <i>Business Success Criteria</i>	Collect Initial Data <i>Initial Data Collection Report</i>	Select Data <i>Rationale for Inclusion/Exclusion</i>	Select Modeling Techniques <i>Modeling Technique</i> <i>Modeling Assumptions</i>	Evaluate Results <i>Assessment of Data Mining Results w.r.t. Business Success Criteria</i> <i>Approved Models</i>	Plan Deployment <i>Deployment Plan</i>
Assess Situation <i>Inventory of Resources</i> <i>Requirements, Assumptions, and Constraints</i> <i>Risks and Contingencies</i> <i>Terminology</i> <i>Costs and Benefits</i>	Describe Data <i>Data Description Report</i>	Clean Data <i>Data Cleaning Report</i>	Generate Test Design <i>Test Design</i>	Review Process <i>Review of Process</i>	Plan Monitoring and Maintenance <i>Monitoring and Maintenance Plan</i>
Determine Data Mining Goals <i>Data Mining Goals</i> <i>Data Mining Success Criteria</i>	Explore Data <i>Data Exploration Report</i>	Construct Data <i>Derived Attributes</i> <i>Generated Records</i>	Build Model <i>Parameter Settings</i> <i>Models</i> <i>Model Descriptions</i>	Determine Next Steps <i>List of Possible Actions</i> <i>Decision</i>	Produce Final Report <i>Final Report</i> <i>Final Presentation</i>
Produce Project Plan <i>Project Plan</i> <i>Initial Assessment of Tools and Techniques</i>	Verify Data Quality <i>Data Quality Report</i>	Integrate Data <i>Merged Data</i>	Assess Model <i>Model Assessment</i> <i>Revised Parameter Settings</i>		Review Project <i>Experience</i> <i>Documentation</i>
		Format Data <i>Reformatted Data</i>			
		<i>Dataset</i> <i>Dataset Description</i>			

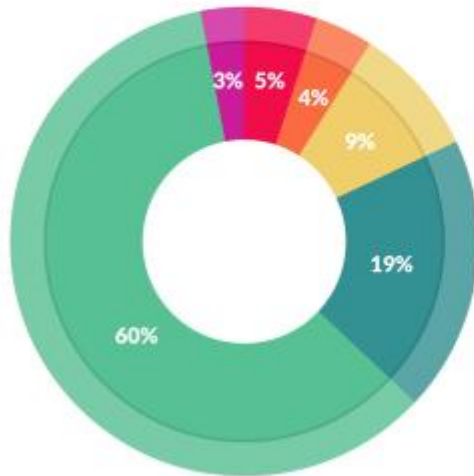
Figure 3: Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model

Source: CRISP-DM 1.0 SPSS – *Statistical Package for the Social Sciences* – 2000
(acquis par IBM en 2009)

4. Que font les Data Scientists ?

Data Scientists Spend Most of Their Time Cleaning Data

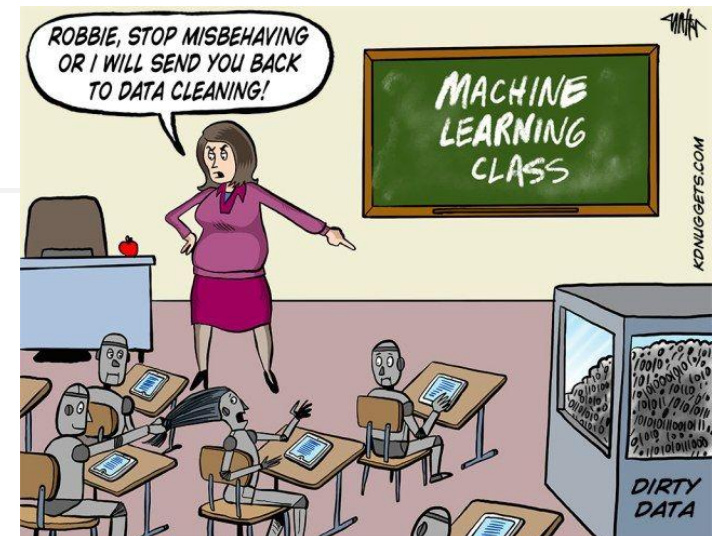
Posted on May 1, 2016



What data scientists spend the most time doing

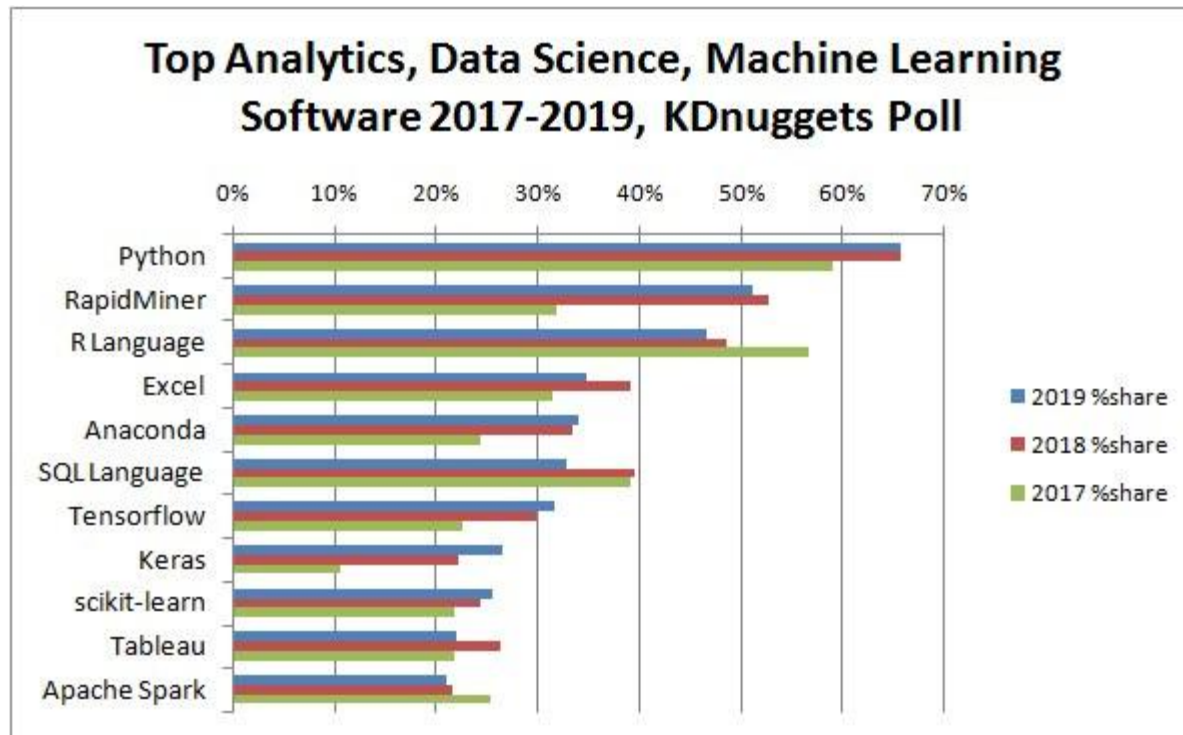
- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

Source: whatsthebigdata.com/2016/05/01/data-scientists-spend-most-of-their-time-cleaning-data/



Source: www.kdnuggets.com/2017/09/cartoon-machine-learning-class.html

4. Python et la Data Science

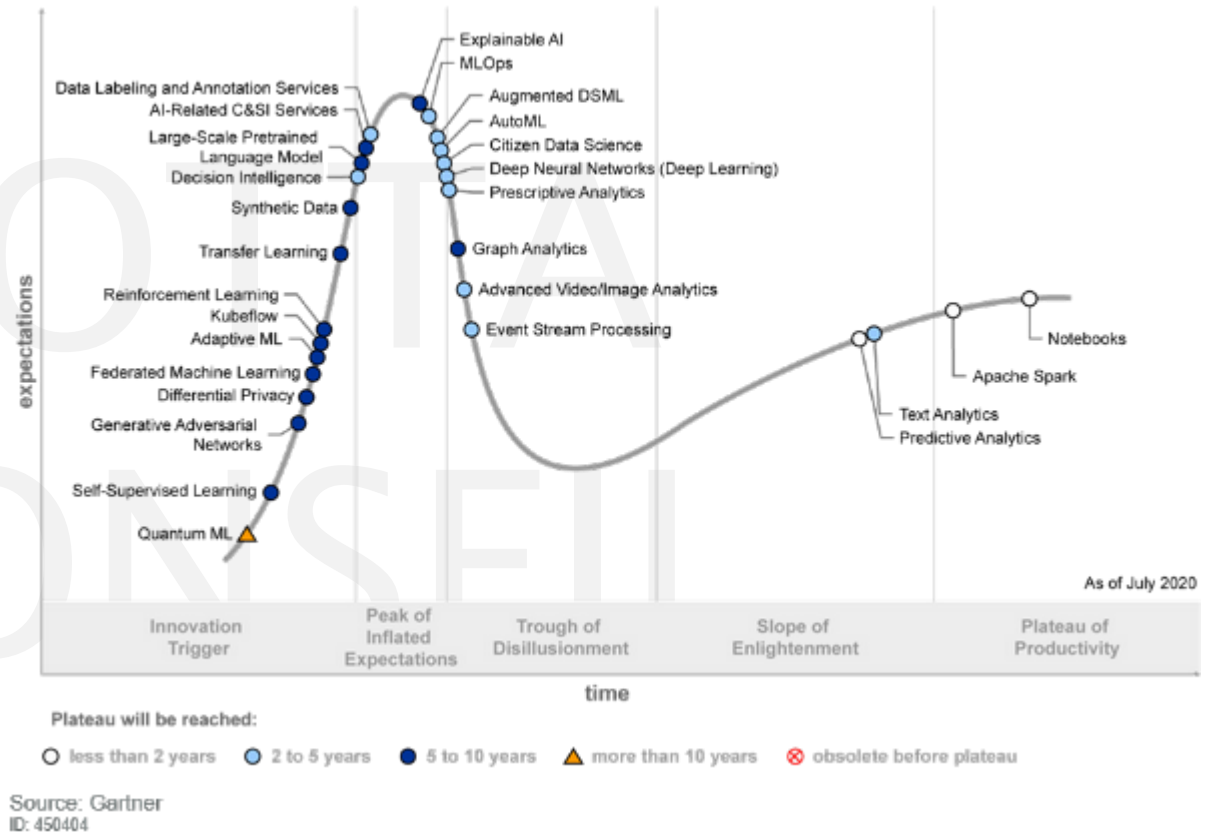
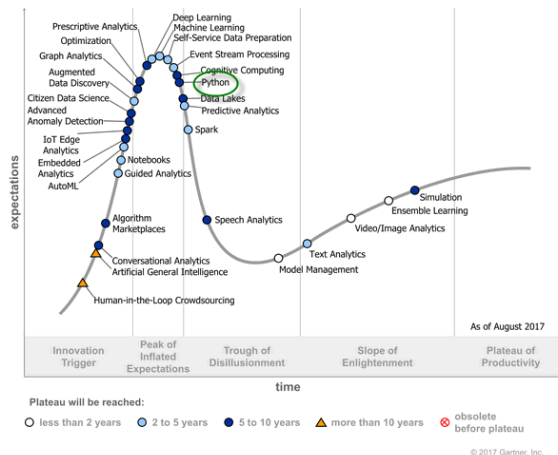


Source:

www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html

4. Hype Cycle du Gartner

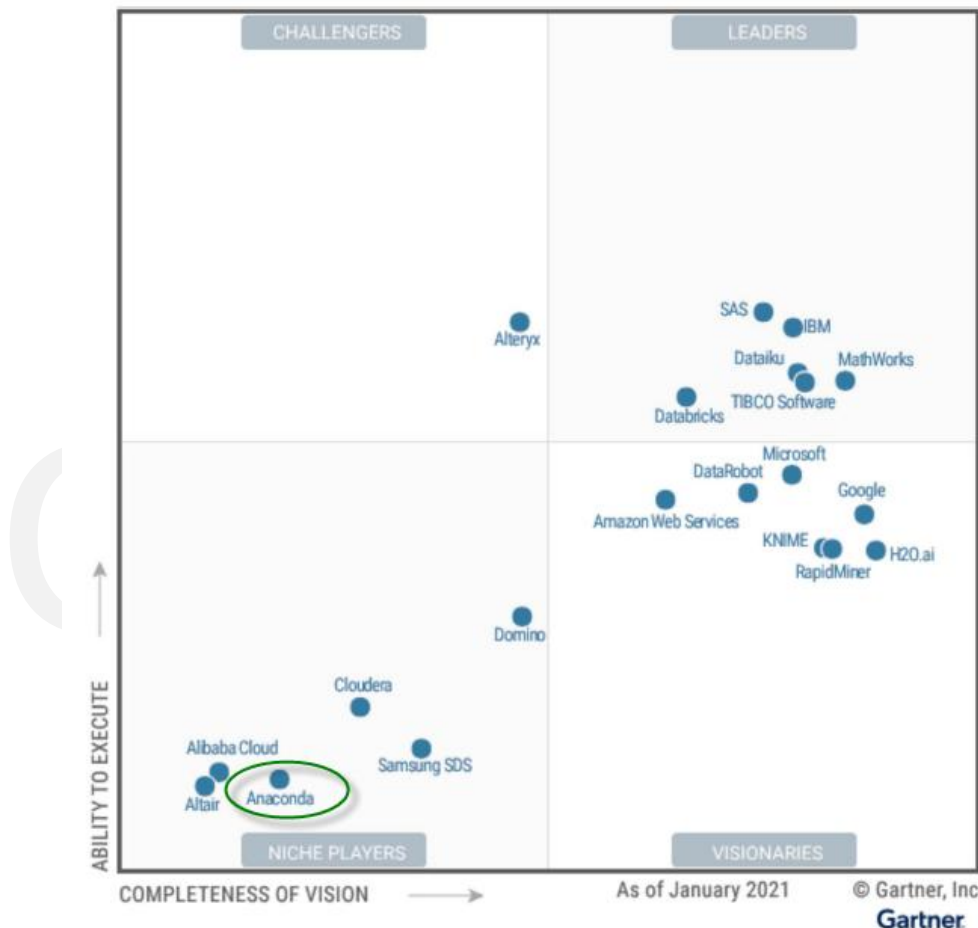
Hype Cycle for Data Science and Machine Learning, 2020



Source:
datafloq.com/read/confidence-from-gartner-data-analytics-dead-5-year/4956

4. Magic Quadrant de Gartner

pour les plates-formes de Data Science et de Machine Learning



Source:

<https://www.alteryx.com/third-party-content/gartner-magic-quadrant-data-science-machine-learning>

4. Ecosystème pour la Data Science



 SciPy.org  Sponsored by
ENTHOUGHT

 Install  Getting Started  Documentation  Report Bugs  Blogs

SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages:

 NumPy 2006 Base N-dimensional array package	 SciPy library 2001 Fundamental library for scientific computing
 Matplotlib 2003 Comprehensive 2D Plotting	 IPython 2007 Enhanced Interactive Console
 SymPy 2007 Symbolic mathematics	 pandas 2008 Data structures & analysis

More information...

[About SciPy](#)
[Install](#)
[Getting Started](#)
[Documentation](#)
[Bug Reports](#)
[Topical Software](#)
[Citing](#)
[Cookbook](#) 
[SciPy Conferences](#)

[Blogs](#) 
[NumFOCUS](#) 

CORE PACKAGES:
[Numpy](#) 
[SciPy library](#) 
[Matplotlib](#) 
[IPython](#) 
[SymPy](#) 
[Pandas](#) 

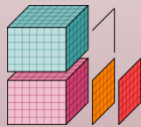
4. Ecosystème pour la Data Science

Data Management

 pandas

 SQLAlchemy

 NetworkX
Network Analysis in Python



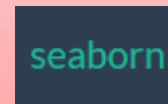
xarray

 DASK

Data Visualization

 matplotlib
Version 3.3.1

 plotly

 seaborn

 boken

Data Processing

 SciPy

 scikit-learn

 statsmodels

 scikit-image
image processing in python

 Keras

 NLTK

 python™
 NumPy

Python Environment



 jupyter

 ANACONDA

 NUMFOCUS
OPEN CODE = BETTER SCIENCE

 PyData

 pyParis

 pythonanywhere

© Yotta Conseil 2021

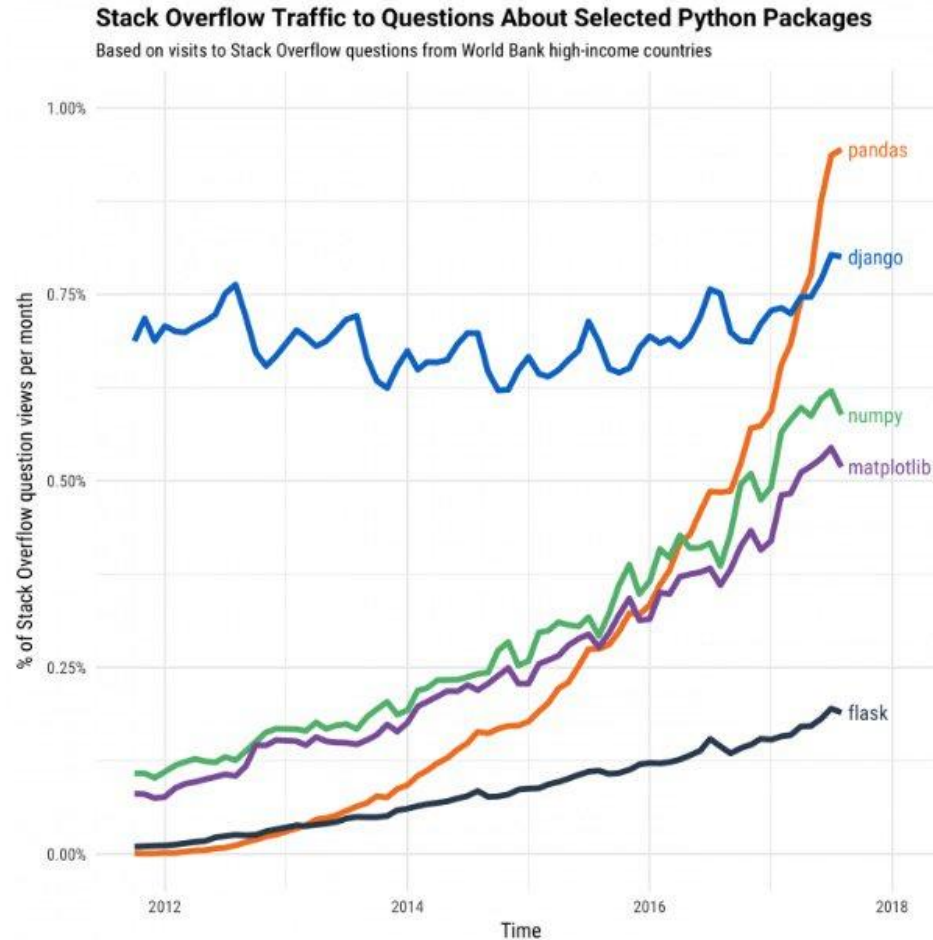
4. Ecosystème pour la Data Science

Module	Summary	License
NumPy	NumPy is the fundamental package for scientific computing with Python	BSD 3-clause
SQLAlchemy	SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.	MIT
Pandas	Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language	BSD 3-clause
NetworkX	NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.	BSD 3-clause
Dask	Parallel PyData with Task Scheduling	BSD 3-clause
SciPy	The SciPy library provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization.	BSD 3-clause
Scikit-learn	A set of python modules for machine learning and data mining	BSD 3-clause
Keras	Keras is a high-level neural networks API, written in Python. It was developed with a focus on enabling fast experimentation.	MIT
StatsModels	StatsModels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.	BSD 3-clause
NLTK	<i>NLTK</i> is a leading platform for building Python programs to work with human language data.	Apache 2.0

4. Ecosystème pour la Data Science

Module	Summary	License
Matplotlib	Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.	PSF-like
Seaborn	Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.	BSD 3-clause
Bokeh	Bokeh is an interactive visualization library that targets modern web browsers for presentation.	BSD 3-clause
Plotly	Plotly's Python graphing library makes interactive, publication-quality graphs online.	MIT
Spyder	The Scientific Python Development Environment	MIT
IPython/Jupyter	IPython provides a rich architecture for interactive computing. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.	BSD 3-clause

4. Succès de quelques modules



Source: David Robison – Why is Python Growing So Quickly?
stackoverflow.blog/2017/09/14/python-growing-quickly/

4. De la programmation à la mise en oeuvre d'API : un changement de paradigme

- ▶ De la programmation...
 - Implémente des algorithmes
 - Modélisation des données
 - Comprendre ou implémenter des algorithmes
 - Implémenter et tester des algorithmes
- ▶ ... à la mise en œuvre d'API
 - Trouver des modules
 - Comprendre les modules : concepts, algorithmes and fonctions
 - Utiliser les modules : syntaxe, arguments
 - Implémenter et tester des modèles
- ▶ Ne réinventez pas la roue. Trouvez le module qui résout votre problème et mettez le en œuvre !



4. Environnement Spyder

Editeur
de code

Outils de
débugage

Inspecteur
d'objets

Interpréteur(s)
Python et/ou
IPython

The screenshot displays the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.8)". The menu bar includes "Fichier", "Édition", "Recherche", "Source", "Exécution", "Débuguer", "Console", "Projets", "Outils", "Affichage", and "Aide". The toolbar contains icons for file operations, running, and debugging. The code editor on the left shows a file named "temp.py" with the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Éditeur de Spyder
4
5 Ceci est un script temporaire.
6 """
7
8 from numpy import linspace, pi
9 from scipy.special import jn
10 import matplotlib.pyplot as plt
11 x = linspace(0, 4*pi)
12 for i in range(5, -1, -1):
13     plt.plot(x, jn(i, x))
14
15 plt.show()
```

The variable explorer on the right shows a table of variables:

Noi	Type	Taille	Valeur
i	int	1	0
pi	float	1	3.141592653589793
x	Array of float64	(50,)	[0. 0.25645654 0.512...

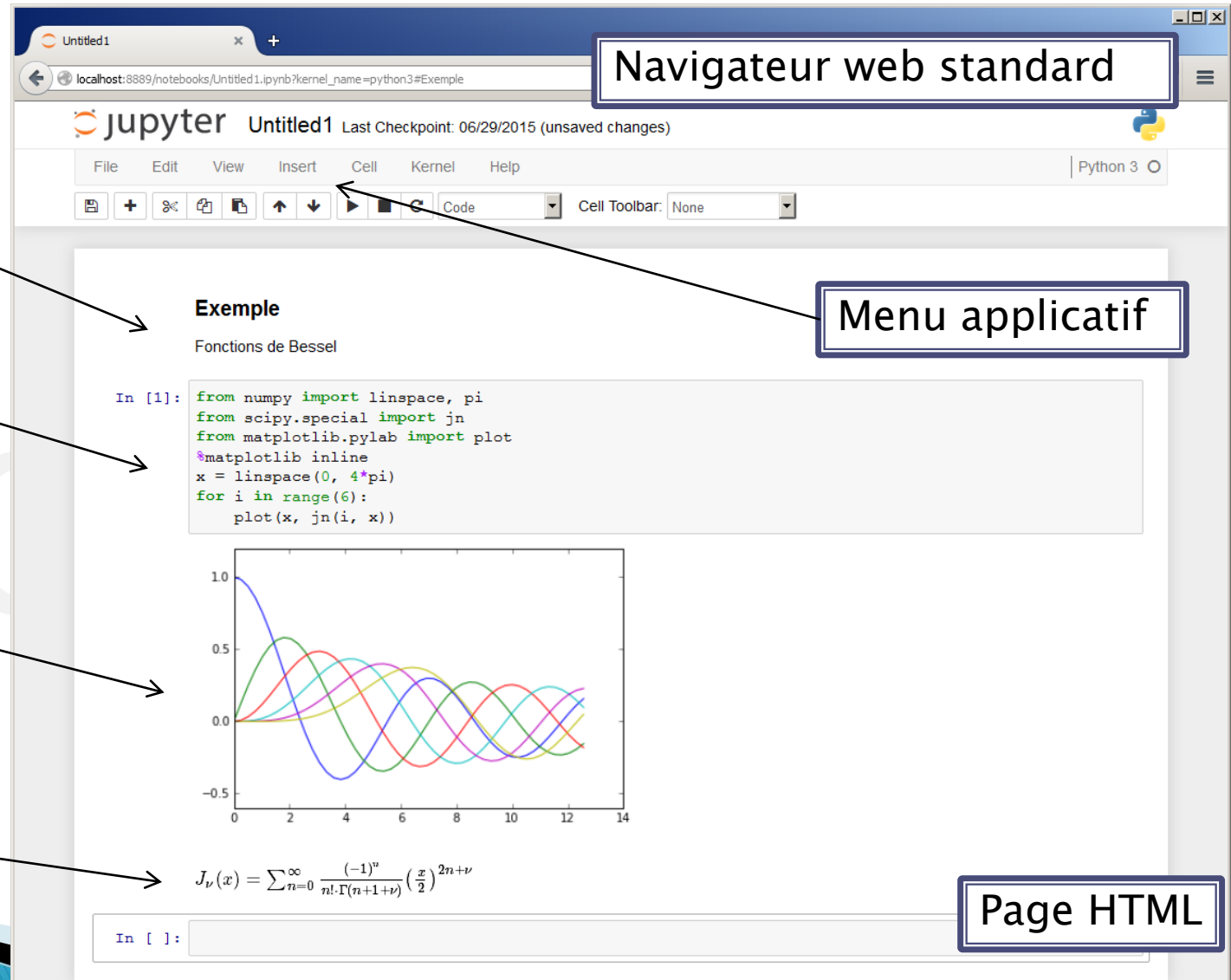
The console at the bottom shows the IPython prompt and the output of the script:

```
Python 3.8.10 (default, May 19 2021, 13:12:57) [MSC v. 1916 64 bit (AMD64)]
Type "copyright", "credits" or "license()" for more
information.

IPython 7.24.1 -- An enhanced Interactive Python.
>>> runfile('C:/Users/Francis/.spyder-py3/temp.py', wdir='C:/Users/Francis/.spyder-py3')
In [2]:
```

A separate window titled "Figure 1" shows a 3D surface plot of a function.

4. IPython Notebook alias Jupyter



The screenshot shows a Jupyter Notebook interface in a web browser. The browser's address bar is labeled "Navigateur web standard". The notebook's menu bar is labeled "Menu applicatif". The notebook contains three cells: a text cell, a code cell, and a result cell. The code cell is labeled "Cellule de code Python" and contains Python code for plotting Bessel functions. The result cell is labeled "Cellule de résultats et graphiques" and displays a plot of Bessel functions. The text cell is labeled "Cellule de texte riche" and contains the Bessel function formula. The notebook's status bar is labeled "Page HTML".

Navigateur web standard

Menu applicatif

Cellule de texte riche

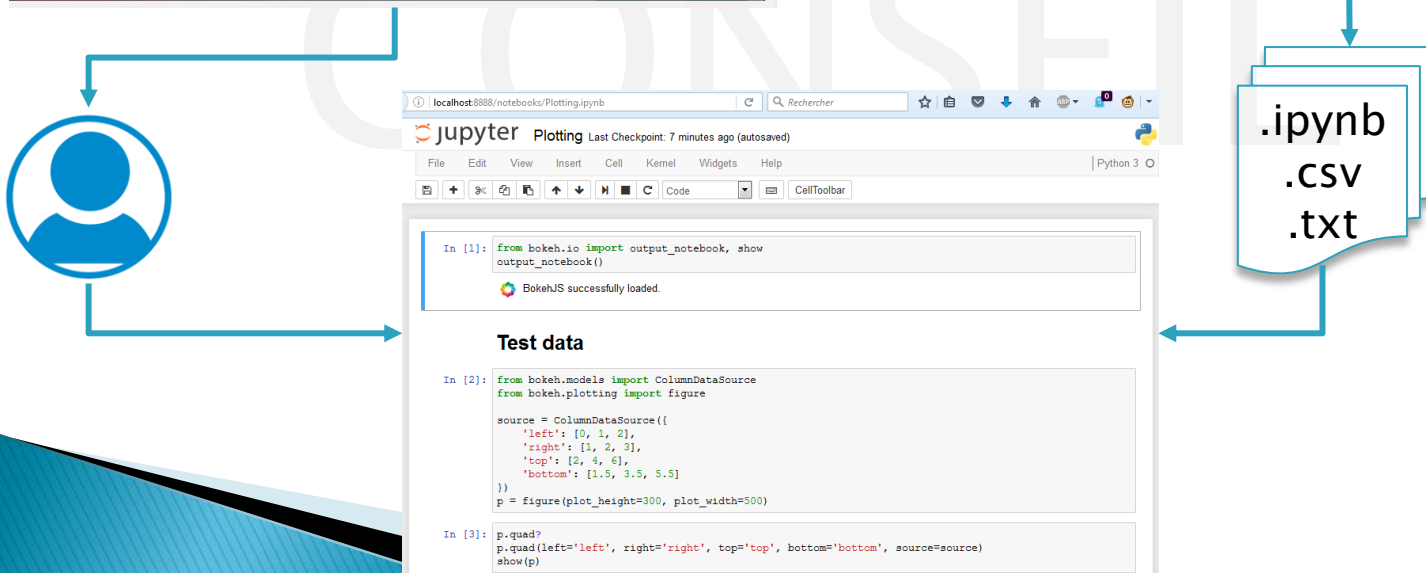
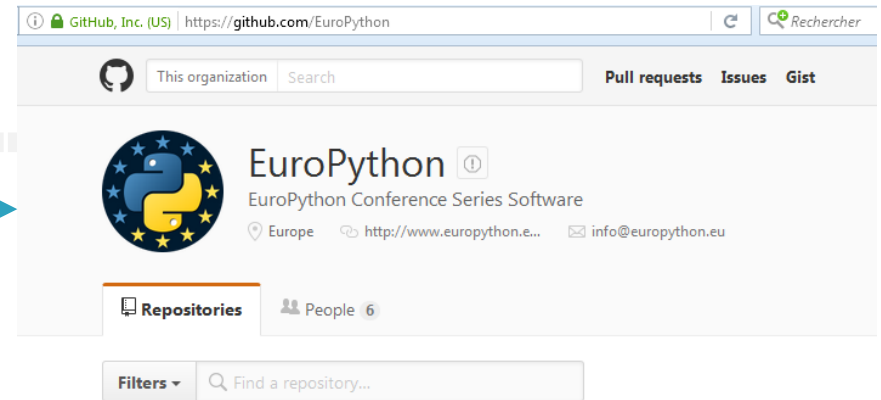
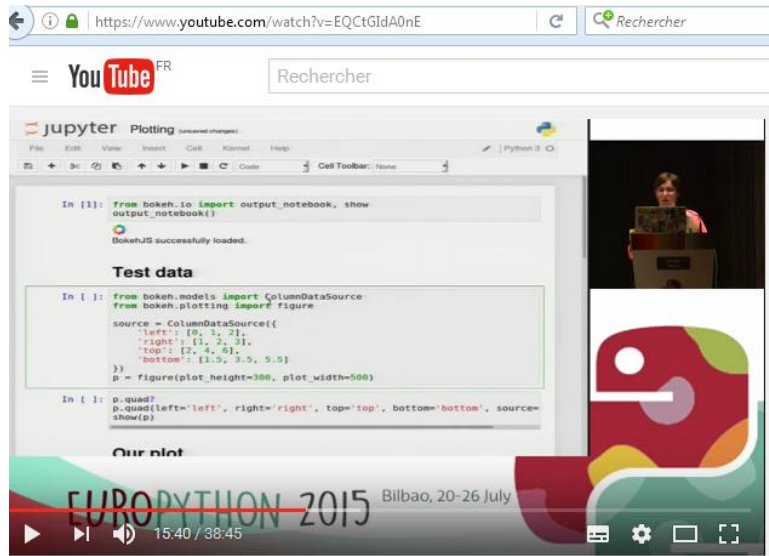
Cellule de code Python

Cellule de résultats et graphiques

Cellule de formules en LaTeX

Page HTML

4. La révolution Jupyter



5. Gestion des librairies

- ▶ Python est fourni d'emblée avec plusieurs dizaines de librairies (extrait)
 - **re** Regular expression operations
 - **datetime** Basic date and time types
 - **calendar** General calendar-related functions
 - **collections** Container datatypes
 - **math** Mathematical functions
 - **itertools** Functions creating iterators for efficient looping
 - **os.path** Common pathname manipulations
 - **os** Miscellaneous operating system interfaces
 - **urllib.request** Extensible library for opening URLs
 - **socket** Low-level networking interface
- ▶ Les autres librairies doivent être installées
 - Exemples : NumPy, SciPy, StatsModels, matplotlib, pandas, NLTK, scikit-learn...

5. Utilitaire pip de Python

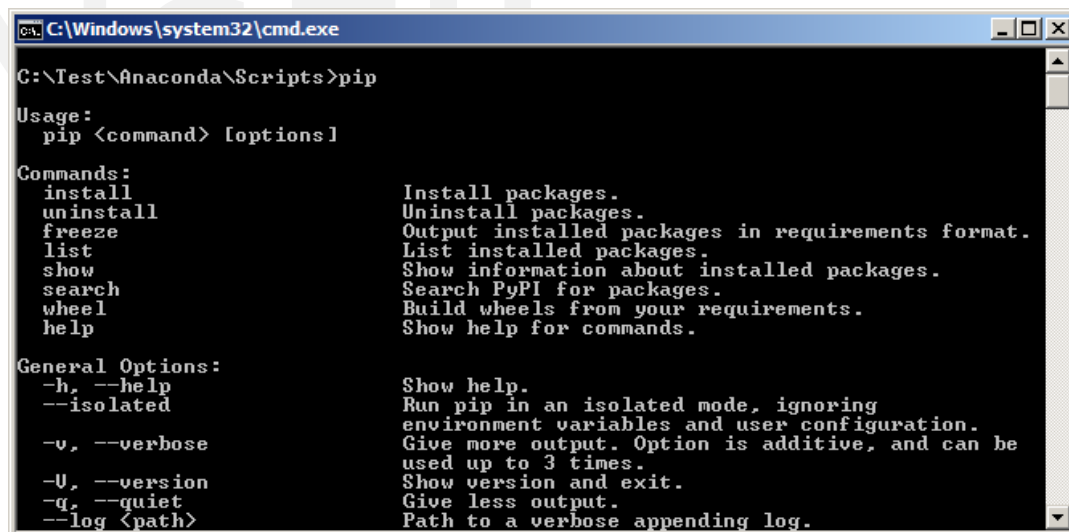
► pip

- Gestionnaire de librairie qui permet, entre autres, d'installer et de mettre à jour des librairies qui ne sont pas incluses nativement dans les distributions Python.
- Les modules proviennent de PyPI (Python Package Index) qui comprend plus de 330.000 modules !
- Mise à jour en ligne

`pip install <module>`

► Limites :

- Modules nécessitant un compilateur C++
- Existence d'installateurs spécialisés



```
C:\Windows\system32\cmd.exe
C:\Test\Anaconda\Scripts>pip

Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list              List installed packages.
  show              Show information about installed packages.
  search            Search PyPI for packages.
  wheel             Build wheels from your requirements.
  help             Show help for commands.

General Options:
  -h, --help        Show help.
  --isolated         Run pip in an isolated mode, ignoring
                    environment variables and user configuration.
  -v, --verbose     Give more output. Option is additive, and can be
                    used up to 3 times.
  -U, --version     Show version and exit.
  -q, --quiet       Give less output.
  --log <path>     Path to a verbose appending log.
```

5. Utilitaire conda de Anaconda

► conda

- Conda is an open source package management system and environment management system for installing multiple versions of software packages and their dependencies and switching easily between them.
- It works on Linux, OS X and Windows, and was created for Python programs but can package and distribute any software.

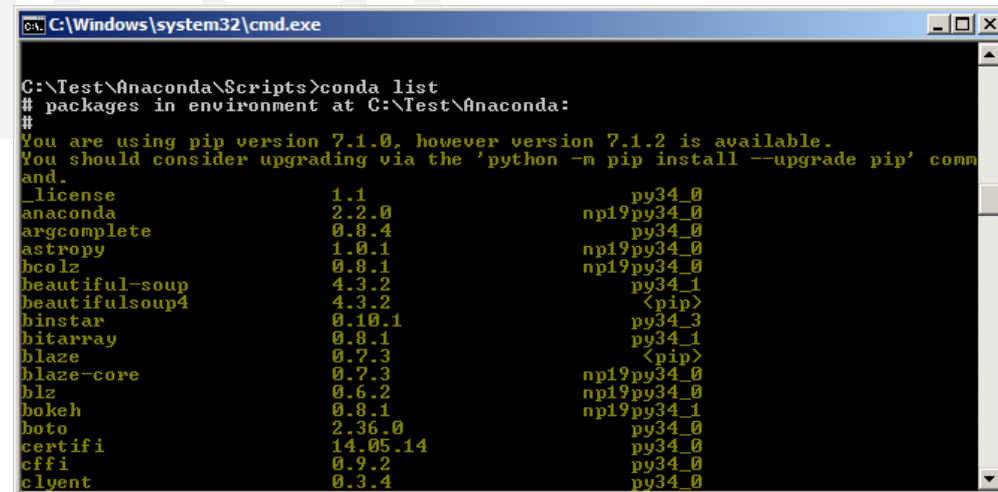
► Utilisation de conda

`conda update conda`

`conda update anaconda`

...

`conda install <module>`



```
C:\Windows\system32\cmd.exe

C:\Test\Anaconda\Scripts>conda list
# packages in environment at C:\Test\Anaconda:
#
You are using pip version 7.1.0, however version 7.1.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
_license                1.1                      python34_0
anaconda                 2.2.0                   np19py34_0
argcomplete              0.8.4                   python34_0
astropy                  1.0.1                   np19py34_0
bcolz                    0.8.1                   np19py34_0
beautiful-soup           4.3.2                   python34_1
beautifulsoup4           4.3.2                   <pip>
binstar                  0.10.1                  python34_3
bitarray                 0.8.1                   python34_1
blaze                    0.7.3                   <pip>
blaze-core               0.7.3                   np19py34_0
blz                      0.6.2                   np19py34_0
bokeh                    0.8.1                   np19py34_1
boto                     2.36.0                  python34_0
certifi                  14.05.14                python34_0
cffi                     0.9.2                   python34_0
clyent                   0.3.4                   python34_0
```