PYTHON

INTRODUCCIÓN

Historia

- Creado a finales de los 80's, por Guido van Rossum.
- 1991: Liberado como Software de Código abierto.
- 2000: Python 2.0.
- 2008: Python 3.0. Cambios importantes!

Guido van Rossum

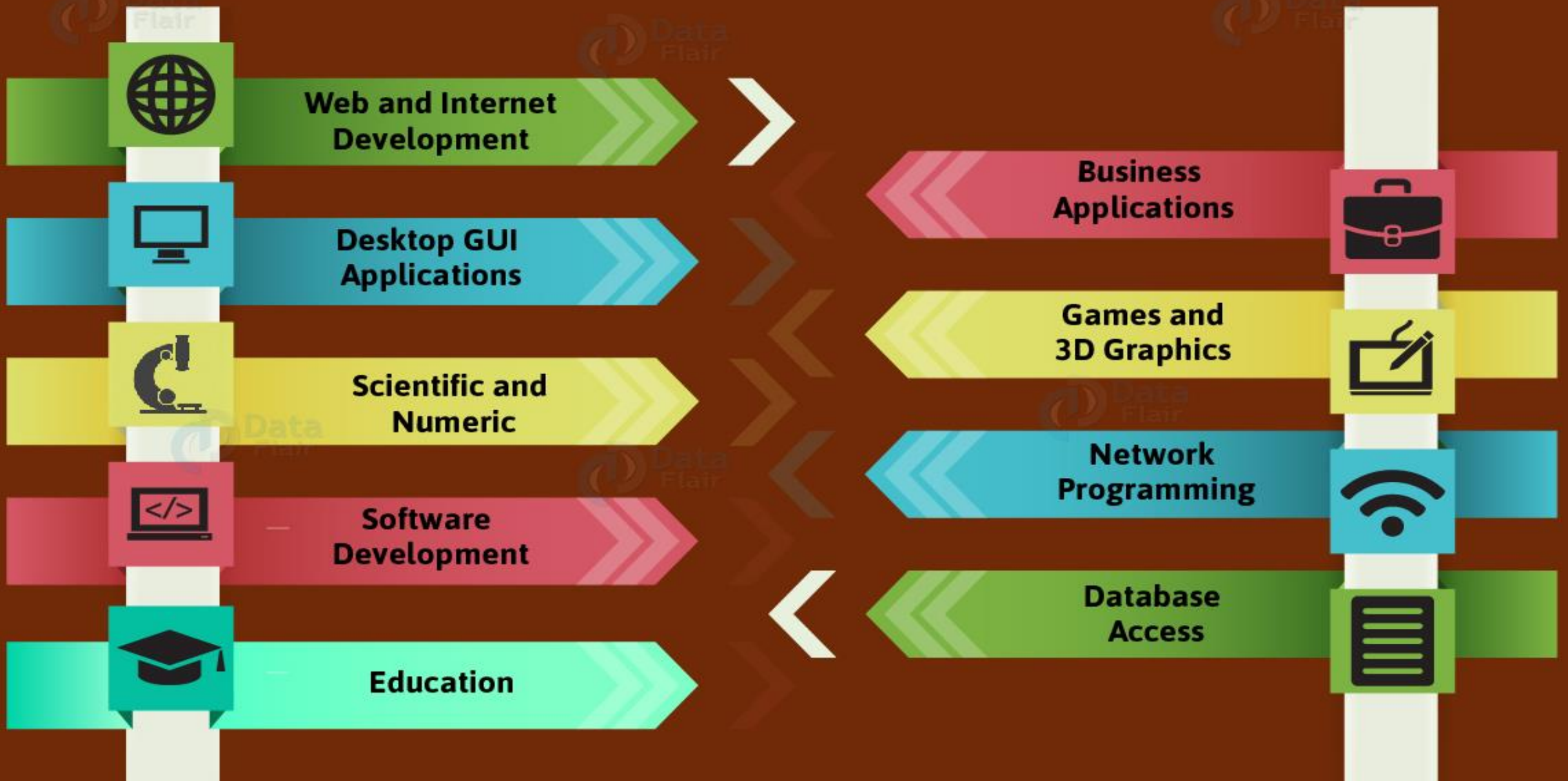
Monty Python's Flying Circus

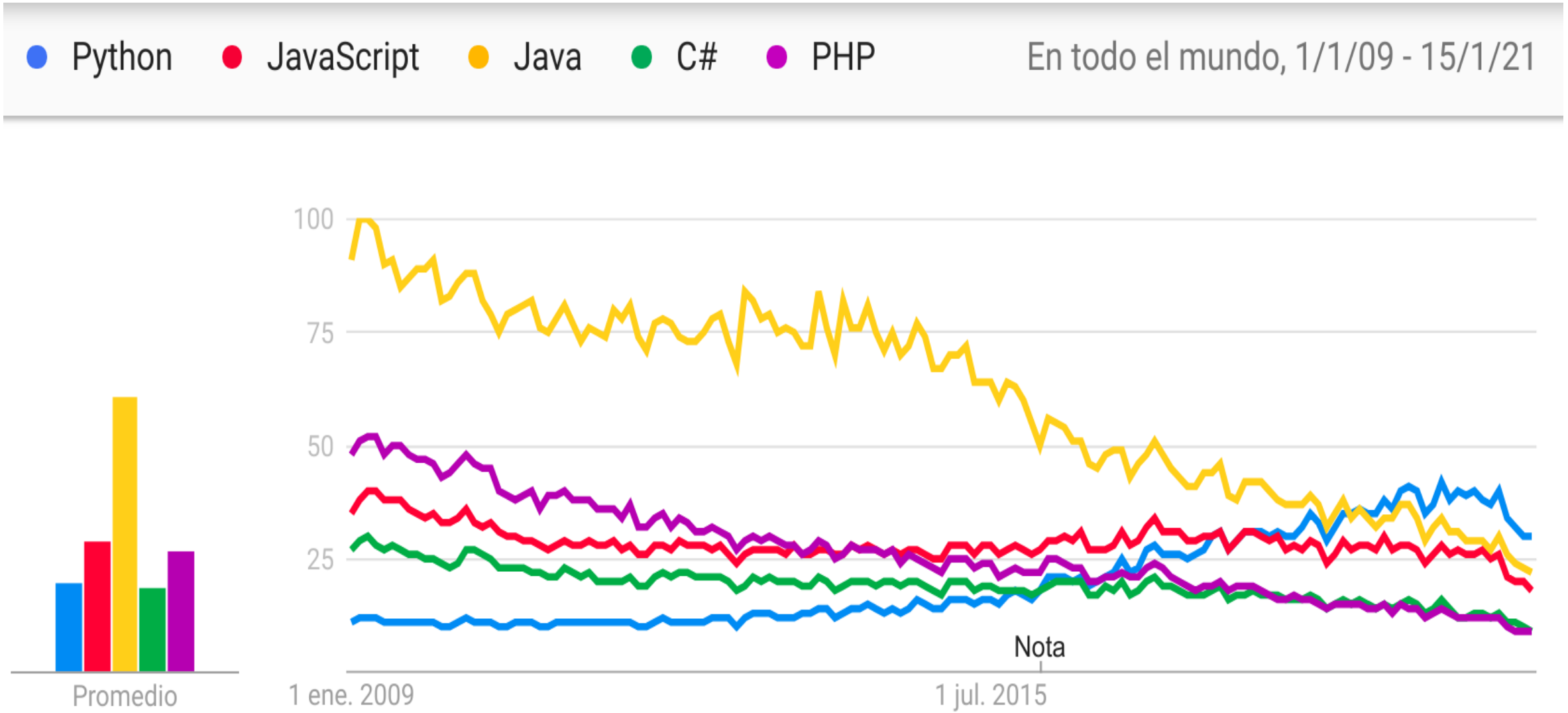
ALGUNAS CARACTERÍSTICAS DE PYTHON

- Sintaxis clara y fácil de aprender (lenguaje de alto nivel).
- De propósito general = muy versátil.
- Lenguaje interpretado (no compilado).
- Requiere un tiempo de desarrollo **reducido** y un tiempo de ejecución **razonable**.
- De tipado dinámico.
- Multiparadigma: programación orientada a objetos, funcional, procedural.
- Extensible y modular, gran cantidad de bibliotecas o módulos.

```python
from mpl_toolkits.mplot3d import axes3d
from matplotlib.pyplot import *
from matplotlib import cm, rc
from numpy import *
from scipy.integrate import quad, dblquad
from scipy.special import jn,jn_zeros
from scipy import real, imag

R = 1.
v = 1.

#Se define n-ésimo cero de la función de Bessel $J_{m}$
def alpha(m,n):
    return jn_zeros(m,n)[-1]

from matplotlib.animation import FuncAnimation

def ani(m,n):
    def psimn(rho,phi,t):
        return jn(m,(alpha(m,n)/R)*rho)*cos((v/R)*alpha(m,n)*t)*cos(m*phi)

    n_ang = 120 # divisiones de la coordenada angular
    n_radio = 50 # divisiones de la coordenada radial
    radio = linspace(0, R, n_radio)
    ang = linspace(0, 2*pi, n_ang)
    RHO, PHI = meshgrid(radio,ang)
    X = RHO*cos(PHI) #Definiendo dominio en x
    Y = RHO*sin(PHI) #Definiendo dominio en y

    fig = figure(figsize=(11,10))
    ax = fig.gca(projection='3d')

    def animate(i):
        t = (5/100.)*i
        animate.p.remove()
        animate.p = ax.plot_surface(X, Y, psimn(RHO,PHI,t), rstride=1, cstride=1, color='C0', antialiased=False)
        return animate.p,

    animate.p = ax.plot_surface(X, Y, psimn(RHO,PHI,0), rstride=1, cstride=1, color='C0', antialiased=False)
    ax.set_zlim(-1.1,1.1)
    ax.set_xlabel(r'$x$',fontsize=15)
    ax.set_ylabel(r'$y$',fontsize=15)
    ax.set_title(r'$\Psi^{\rm est}_{%d,%d}(\rho,\varphi,t)$'%(m,n),fontsize=15)
    ax.set_xticks(arange(-1,1.1,0.5))
    ax.set_yticks(arange(-1,1.1,0.5))
    ax.set_zticks(arange(-1,1.1,0.5))
    anim = FuncAnimation(fig, animate, interval=50, frames=200, repeat=True, blit=True)
    anim.save('psimn-m-'+str(m)+'-n-'+str(n)+'-estacionarios.gif', writer='imagemagick')

for m in range(5):
    for n in range(1,5):
        ani(m,n)
```

Python Applications





Google Trends: https://trends.google.com/trends/explore?date=2009-01-01%202020-01-05&q=%2Fm%2F05z1_,%2Fm%2F02p97,%2Fm%2F07sbkfb,%2Fm%2F07657k,%2Fm%2F060kv

PYPL PopularitY of Programming Language

pypl.github.io

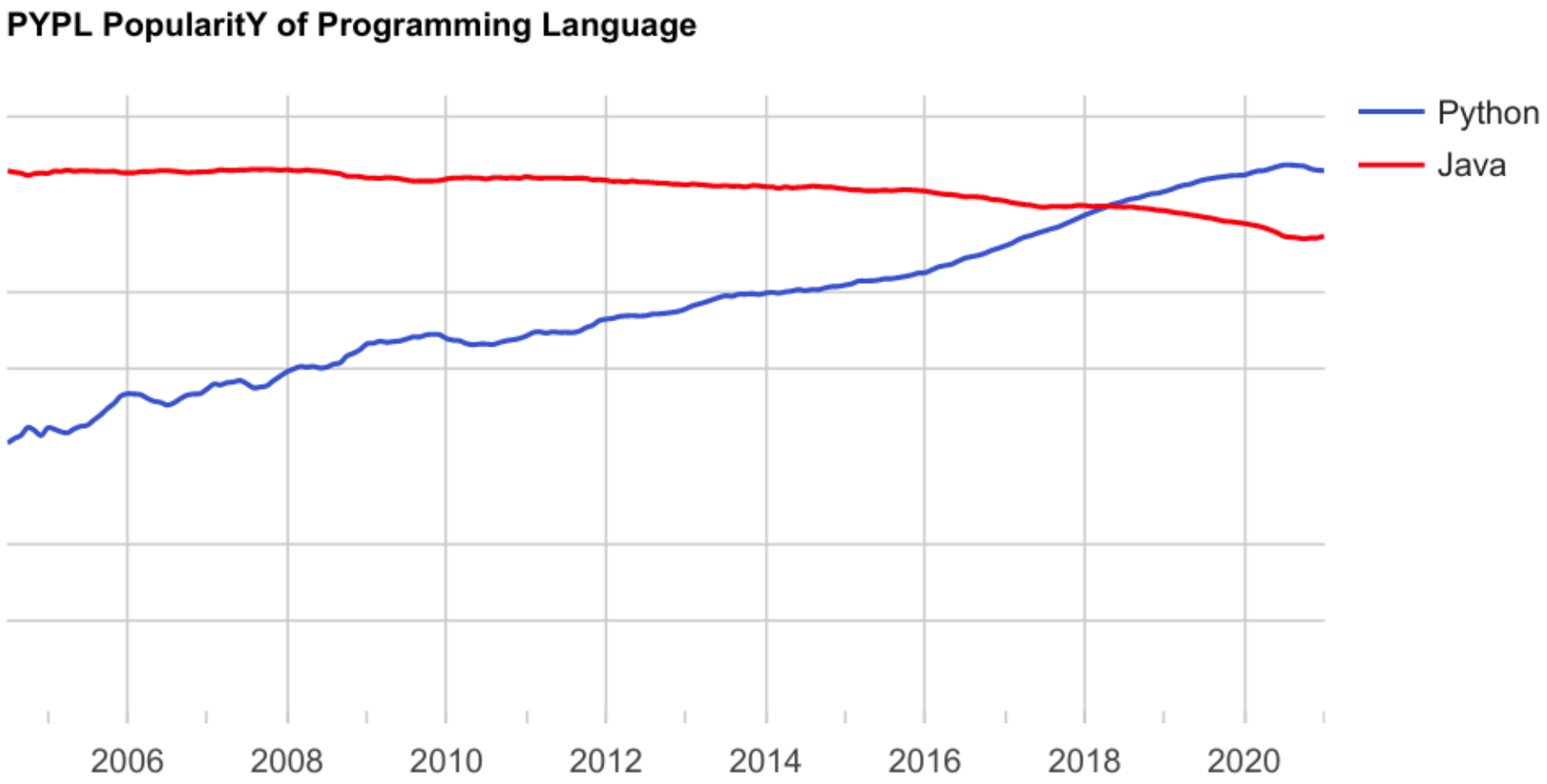
The PYPL PopularitY of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

Worldwide, Jan 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.44 %	+1.2 %
2		Java	16.76 %	-2.0 %
3		JavaScript	8.44 %	+0.3 %
4		C#	6.53 %	-0.7 %
5	↑	C/C++	6.33 %	+0.3 %
6	↓	PHP	6.05 %	-0.2 %
7		R	3.87 %	+0.1 %
8		Objective-C	3.71 %	+1.2 %
9		Swift	2.14 %	-0.3 %

Worldwide, Python is the most popular language, Python grew the most in the last 5 years (18.5%) and Java lost the most (-8.5%)

pypl.github.io

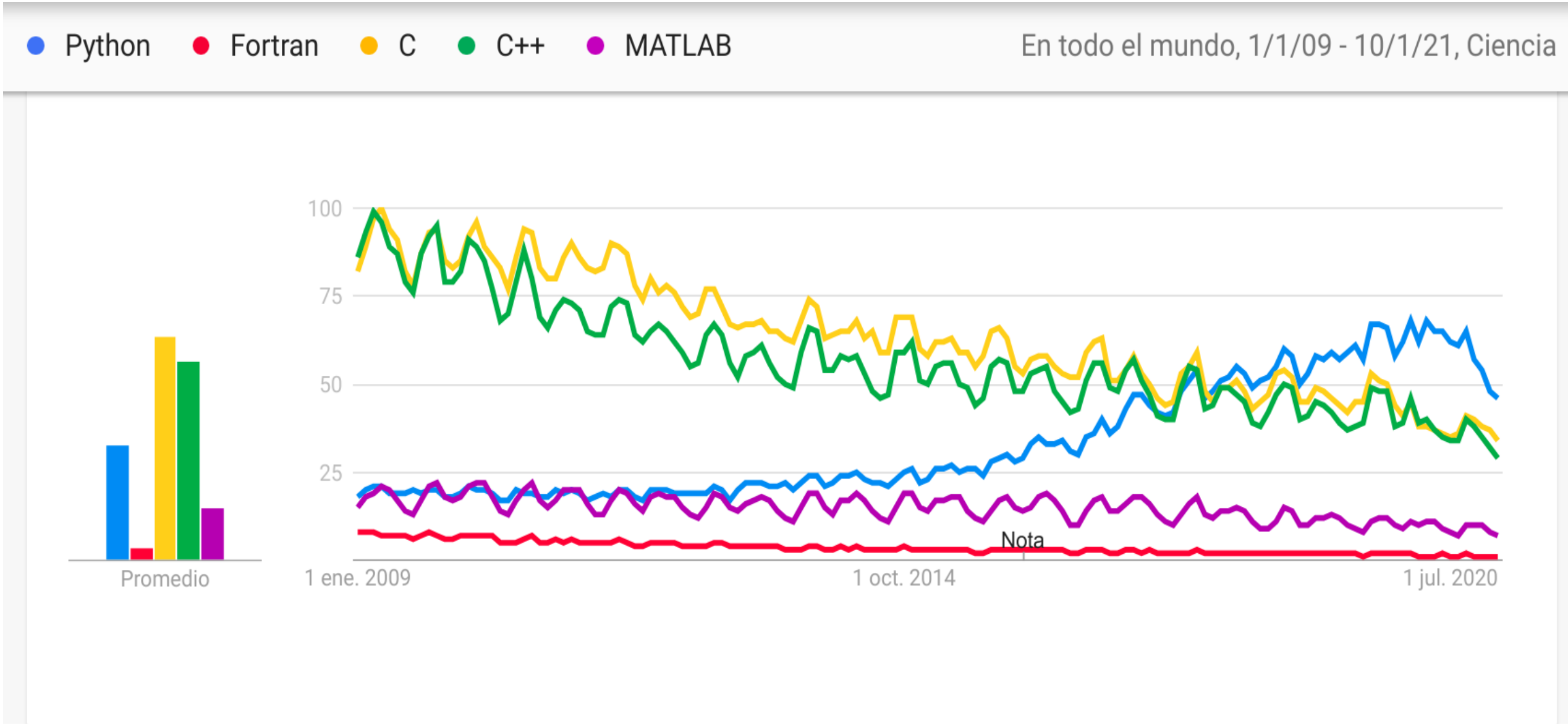


Algunas (grandes) empresas que usan Python

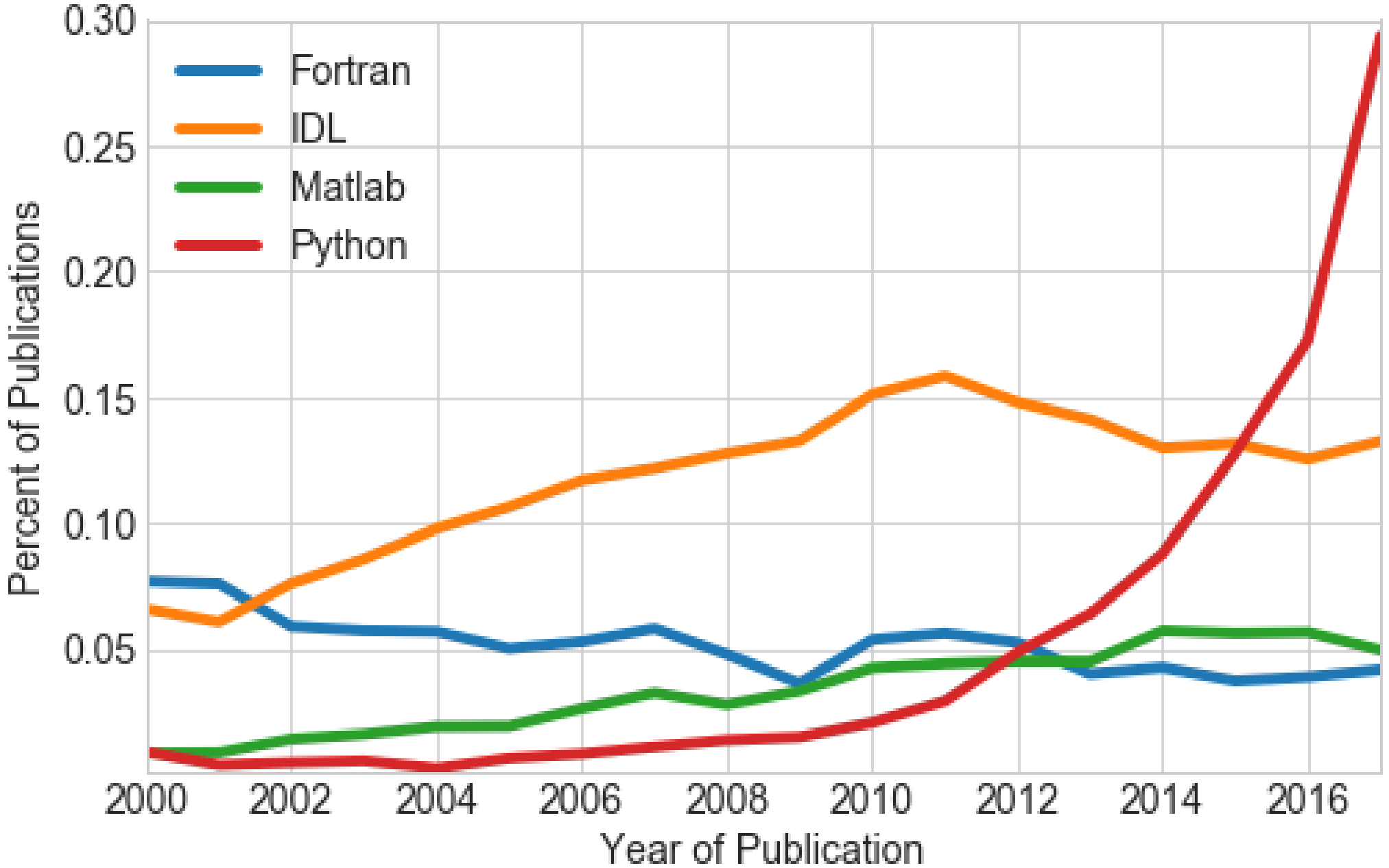
Python en Ciencias

- Gratis y de Código abierto
- Puede manejar gran cantidad de datos
- Operaciones con arreglos (vectores, matrices, tensores, etc.) con Numpy.
- Gráficos con calidad "de paper" con Matplotlib.
- Muchas librerías/módulos para Ciencias e Ingeniería

Créditos: Lorena Barba, https://youtu.be/vq2KiQmIkRc

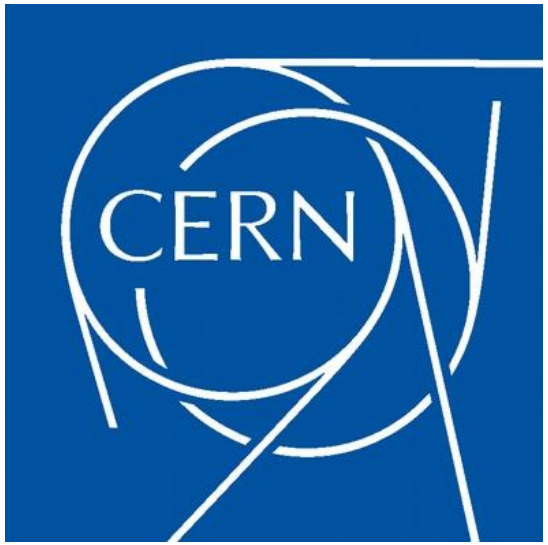


Google Trends: https://trends.google.com/trends/explore?cat=174&date=2009-01-01%202021-01-10&q=%2Fm%2F05z1_,%2Fm%2F02_94,%2Fm%2F01t6b,%2Fm%2F0jgqg,%2Fm%2F053_x



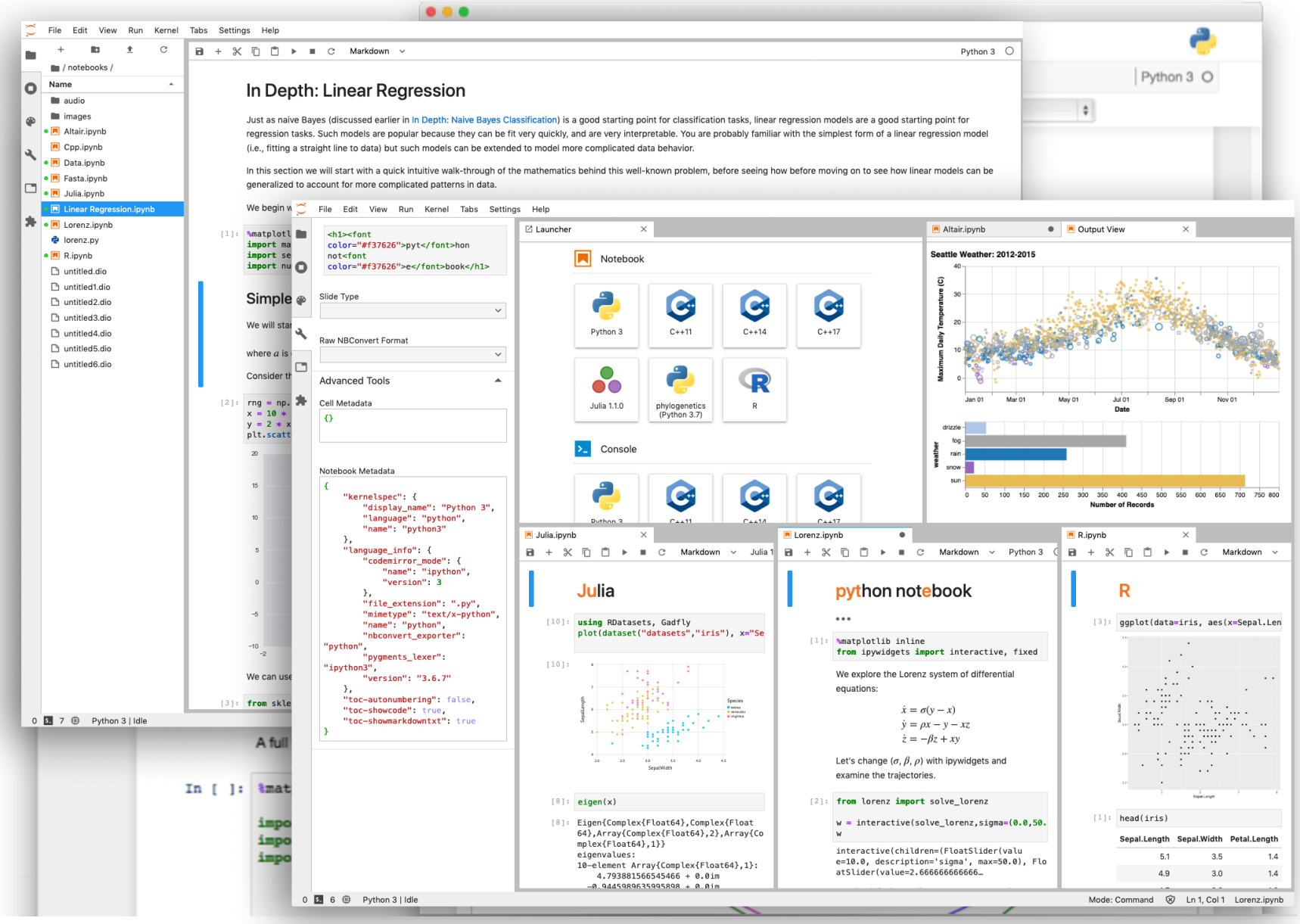
Fuente: Jake VanderPlas: https://gist.github.com/jakevdp/f75c09e43320290ffbedbca43f9fd917

Algunas (grandes) instituciones científicas que usan Python

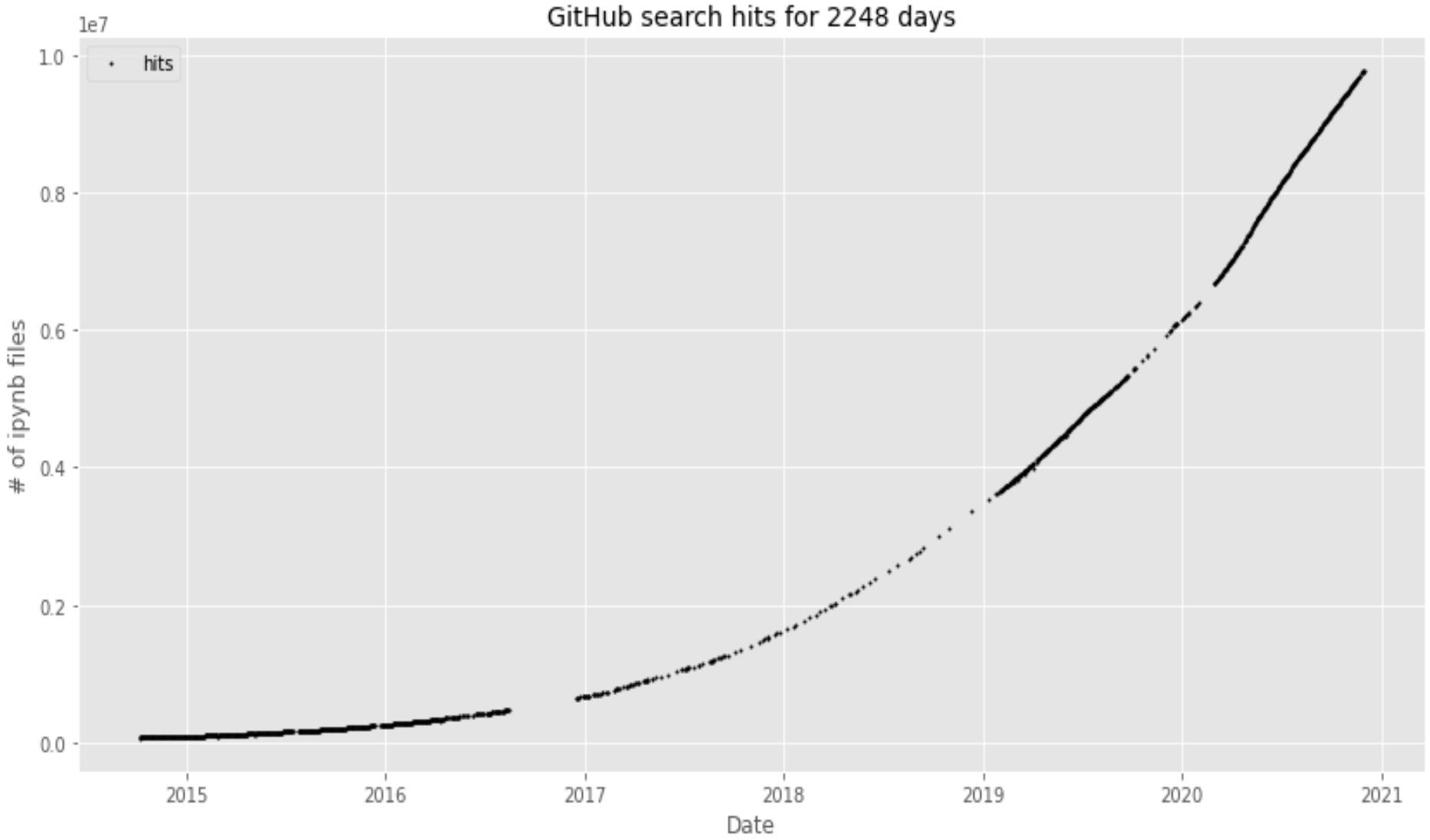


Jupyter Notebooks

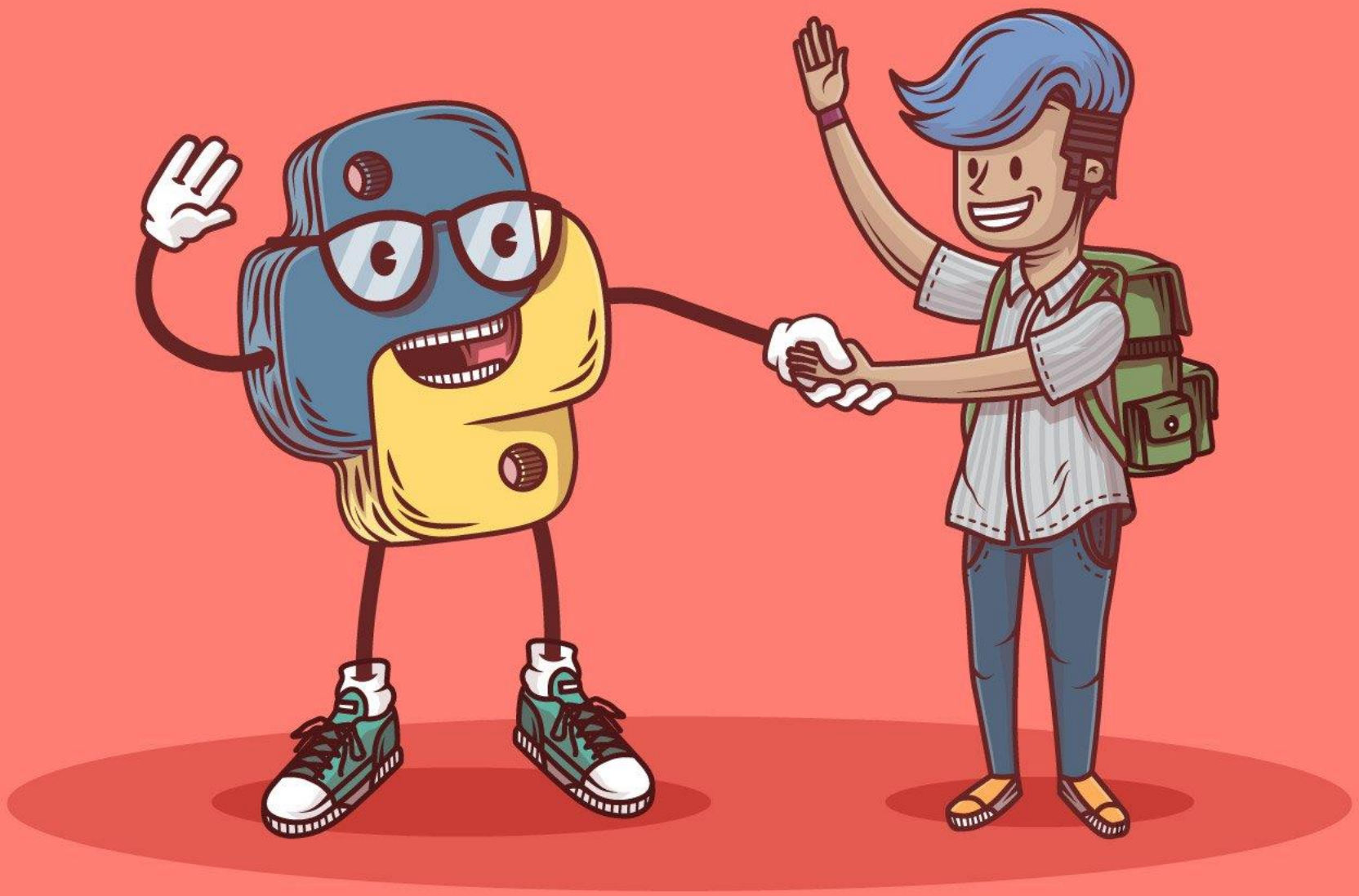
jupyter.org



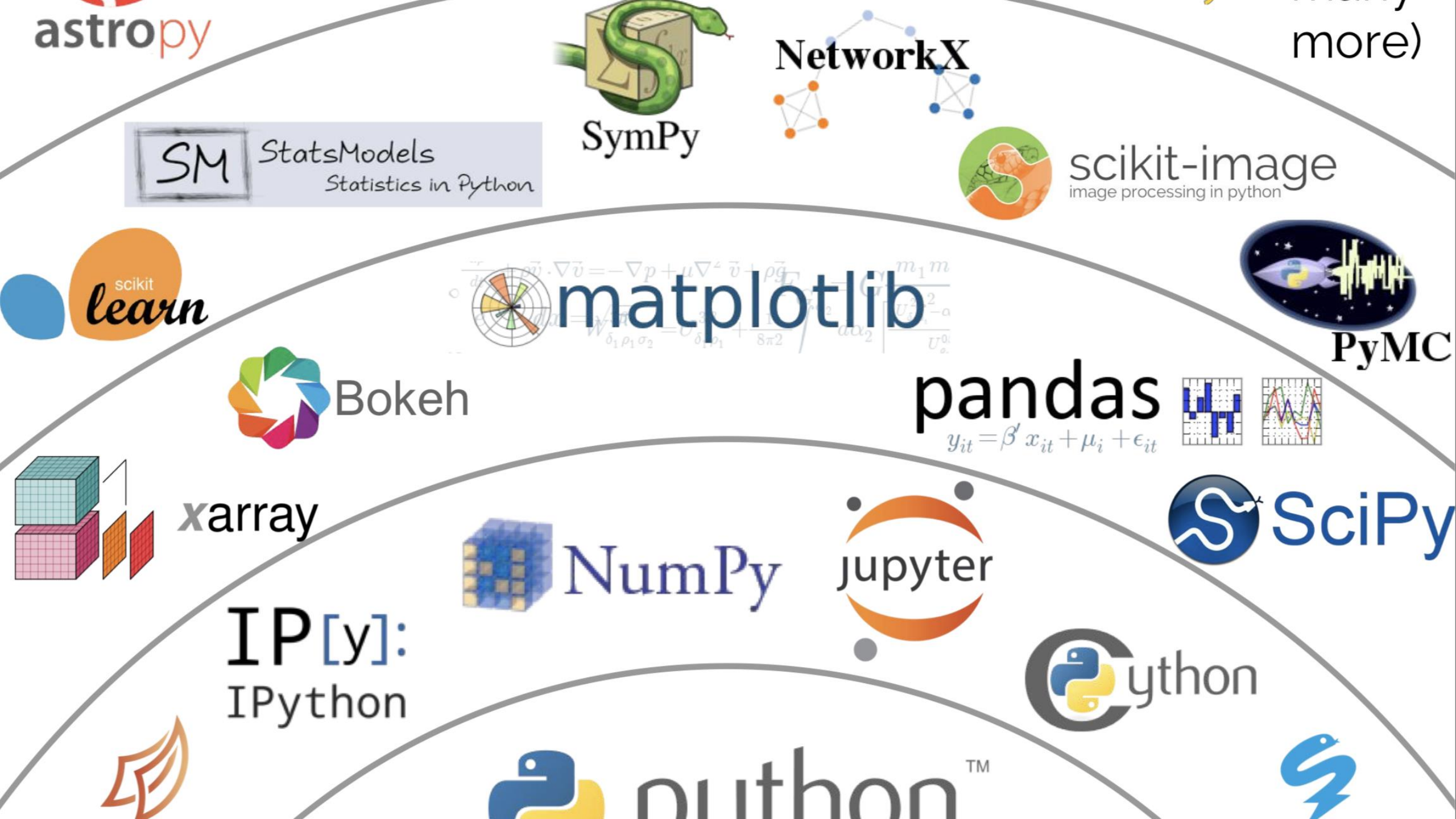
Jupyter Notebooks



Fuente: Peter Parente, https://github.com/parente/nbestimate



Real Python


astropy
more)
SymPy
NetworkX
SM StatsModels
Statistics in Python
scikit-image
image processing in python
scikit learn
matplotlib
PyMC
Bokeh
pandas
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$
xarray
SciPy
NumPy
jupyter
IP[y]:
IPython
ython
python™