IPython

A multi-language architecture for interactive computing

ipython.org (http://ipython.org)

Fernando Pérez

fperez.org (http://fperez.org), @fperez_org (http://twitter.com/fperez_org)

U.C. Berkeley

In [1]: %pylab inline
%run talktools

Populating the interactive namespace from numpy and matplotlib

Why IPython?

"The purpose of computing is insight, not numbers"

Hamming '62
The Lifecycle of a Scientific Idea (schematically)

1. Individual exploratory work
2. Collaborative development
3. Parallel production runs (HPC, cloud, ...)
4. Publication (with reproducible results!)
5. Education

From a better shell...

2001, instead of a Physics dissertation
... to a fancier shell (and a protocol!)...

2010, E Patterson, Enthought (key support!)

... 6 notebook attempts over 6 years...

T. Matev, T. Alatalo, R. Kern, Min RK, J. Gao, B. Granger
... up to today's IPython Notebook

Everyday exploratory computing, with annotations

Let’s plot the Bessel functions $J_i$ for $i = 0, 3, 6, 9$:

In [2]:
```python
import scipy.special as spec
x = linspace(0, 20, 200)
for n in range(0, 13, 3):
    plot(x, spec.jn(n, x), label=r'$J_{%i}(x)$' % n)
grid()
legend()
title('Bessel Functions are neat');
```

![Graph showing Bessel Functions](image)

Explore parameters interactively

In [4]:
```python
@interactive(x=(1, 10))
def f(x):
    print 'X is:', x
```

X is: 5
In [5]:
from sklearn import datasets
digits = datasets.load_digits()
n = len(digits.images)

#interactive(i=(0,n-1))
def view_image(i):
    print 'Classif. Label:', digits.target[i]
    plt.matshow(digits.images[i], cmap=cm.gray_r)
    plt.show()

Classif. Label: 8

Displaying results: a rich protocol

In [6]: Image('fig/logo.png')
Out[6]:

IP[y]: IPython
Interactive Computing

In [7]: Video('fig/animation.m4v')  # Credit: Chris Kees, Army ERDC; created with Proteus.
Out[7]:

Browser multimedia + scientific visualization
In [8]: plot_audio('voice.wav')
Audio('voice.wav')

Out[8]:

LaTeX and symbolic mathematics support

In [9]: Math(r'F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k} dx')

Out[9]:

In [10]: from sympy import symbols, Eq, factor, init_printing, expand
init_printing(use_latex=True)
x, y = symbols("x y")

In [11]: eq = ((x+y)**3 * (x+1))
display(Eq(eq, expand(eq)))

\[(x+1)(x+y)^3 = x^4 + 3x^3y + x^3 + 3x^2y^2 + 3x^2y + xy^3 + 3xy^2 + y^3\]

In [12]: @interactive(n=(1,10))
def _(n):
    eq = ((x+y)**n * (x+1))
    display(Eq(eq, expand(eq)))

\[(x+1)(x+y)^5 = x^6 + 5x^5y + x^5 + 10x^4y^2 + 5x^4y + 10x^3y^3 + 10x^3y^2 + 5x^2y^4 + 10x^2y^3 + xy^5\]

Dynamic graph visualizations with D3

Talk to other languages: R (or Octave)
In [13]:
X = np.array([0,1,2,3,4])
Y = np.array([3,5,4,6,7])
# Now, load R support
%load_ext rmagic

In [14]:
%%R -i X,Y -o XYcoef
XYlm = lm(Y~X)
XYcoef = coef(XYlm)
print(summary(XYlm))
par(mfrow=c(2,2))
plot(XYlm)

Call:
  lm(formula = Y ~ X)

Residuals:
       1       2       3       4       5
  -0.2     0.9 -1.0     0.1     0.2

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.2000     0.6164   5.191   0.0139 *
X             0.9000     0.2517   3.576   0.0374 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7958 on 3 degrees of freedom
Multiple R-squared:  0.81, Adjusted R-squared:  0.7467

F-statistic: 12.79 on 1 and 3 DF,  p-value: 0.03739

Julia: seamless 2-way communication
In [15]: %load_ext julia.magic
%%julia
@pyimport matplotlib.pyplot as plt
@pyimport numpy as np

Initializing Julia interpreter. This may take some time...

In [16]:

%%julia
# Note how we mix numpy and julia:
x = linspace(0, 2*pi, 1000); # use the julia linspace
y = sin(3*x + 4*np.cos(2*x)); # use the numpy cosine and julia sine
plt.plot(x, y, color="red", linewidth=2.0, linestyle="--")

Out[16]: [Line2D(_line0)]

Julia and Python

The call stack as a layer cake

In [17]:

jfib = %%julia jfib(n, fib) = n < 2 ? n : fib(n-1, jfib) + fib(n-2, jfib )
def pyfib(n, fib):
    if n < 2:
        return n
    return fib(n-1, pyfib) + fib(n-2, pyfib)
pyfib(20, jfib)  # Credit: Steven Johnson, MIT.

Out[17]: 6765

Fortran integration

In [18]: %load_ext fortranmagic

In [22]:

%%fortran
subroutine f1(x, y, n)
    real, intent(in), dimension(n) :: x
    real, intent(out), dimension(n) :: y
    intent(hide) :: n
    y = sin(x**2)
end subroutine f1
In [23]:

```python
x = np.linspace(0, 2*pi, 300)
plot(x, f1(x));
```

### nbconvert: notebook manipulation tools

- Library for format conversions
- HTML, LaTeX, PDF output.
- Basic command-line usage.
- Rich customization possibilities.

### NBViewer: Share notebooks online

**Matthias Bussonnier, 2012**
IPython Notebook Viewer

A simple way to share IPython Notebooks

Reproducible research...

Min RK, Brian, Greg Caporaso, Rob Knight (group), Justin Riley
Notebooks and AMI for reproducibility:
Companion Website: (http://qiime.org/home_static/nih-cloud-apr2012)

**Instructions and supporting data for the QIIME/IPython/StarCluster demo at the 2012 NIH Cloud Computing the Microbiome workshop and our corresponding paper in the ISME Journal.**

The analysis made use of the IPython Notebook, QIIME, StarCluster, PyCogent and PrimerProspector. All of these tools are pre-installed in the ami-9f69c1f6 public Amazon EC2 instance, which was used in this study.

**Supporting Files**

The IPython notebooks supporting this study can be viewed [here](http://qiime.org/home_static/nih-cloud-apr2012) and are available here in PDF format:

- NIH Cloud Demo (Complete)

**Notebook-based technical blogs**

Jake VanderPlas (astronomy/ML at UW)
In [27]:

Out[27]:

Pytonic Perambulations
(\http://jakevdp.github.io/blog/2013/12/05
/static-­‐interactive-­‐widgets\)

Pytonic Perambulations
(\http://jakevdp.github.io/\)

Musings and ramblings through the world of Python and beyond

- Atom (/atom.xml)

Search

» Atom

- Archives (/archives.html)
- Home Page (\http://www.astro.washington.edu/users/vanderplas\)

Static Interactive Widgets for IPython Notebooks

Training IBM Watson

Using Notebook and IPython.parallel

- From: 8000 lines of Java, JS and HTML5, 2min/query
- To: 200 lines of Python, 2sec/query
Books: "Literate Computing"

Python for Signal Processing, By Jose Unpingco (https://github.com/unpingco/Python-for-Signal-Processing)

- Each chapter is an IPython Notebook
- Posted as a blog entry (http://python-for-signal-processing.blogspot.com/)
- And all available as a Github repo (https://github.com/unpingco/Python-for-Signal-Processing)

... "Probabilistic Programming and Bayesian Methods for Hackers"

By Cam Davidson Pilon (http://www.camdp.com).
Probabilistic Programming & Bayesian Methods for Hackers

An intro to Bayesian methods and probabilistic programming from a computation/understanding-first, mathematics-second point of view.

1. Prologue
2. Contents

... "Mining the Social Web"


- O'Reilly (http://shop.oreilly.com/product/0636920010203.do) book
- Companion Notebook collection

CS 109: Data Science at Harvard
CS109 Data Science

- Home
- Piazza
- Syllabus
- Schedule
- Homework
- Readings
- Projects
- Resources

A growing (and amazing) body of work

For much more: (http://ipython.org/gallery)

A simple and generic architecture
A protocol that enables kernels in other languages

- IJulia (http://nbviewer.ipython.org/url/jdj.mit.edu/~stevenj/IJulia%20Preview.ipynb)
- IHaskell (http://nbviewer.ipython.org/github/gibiansky/IHaskell/blob/master/demo/IHaskell.ipynb)
- IFSharp (http://nbviewer.ipython.org/github/BayardRock/IfSharp/blob/master/Feature%20Notebook.ipynb)
- IRuby (http://nbviewer.ipython.org/github/minad/iruby/blob/master/IRuby-Example.ipynb)
- IGo (https://github.com/takluyver/igo)
- IScala (https://github.com/mattpap/IScala)
- In planning stages: R and Matlab.

The interactive widget architecture is also 100% language-agnostic.
A healthy ecosystem, OSS and commercial

Emacs IPython Notebook Client

Takafumi Arakaki
Vim client

Paul Ivanov @Berkeley

Enthought Canopy
Microsoft Python Tools for Visual Studio and Azure

Shahrokh Mortazavi, Dino Viehland, Wenming Ye, Dennis Gannon. Thanks!!

Microsoft Azure (S. Mortazavi, W. Ye)

StarCluster

Justin Riley @ MIT
Continuum Analytics: Wakari

Cloud.sagemath.com: UW startup, W. Stein

Plot.ly: interactive JS plots in Python
In [31]: website('http://nbviewer.ipython.org/gist/jackparmer/7729584', 'Rosling Countries', width=900, height=600)  

Out[31]: Rosling Countries (http://nbviewer.ipython.org/gist/jackparmer/7729584)

**Bubble Charts & Hover Text with Plotly**

**I'm Jack Parmer**

Plotly is like graphing crack. It standardizes the graphing interface across scientific computing languages (Python, R, MATLAB, etc) while giving rich interactivity and web shareability that has been possible before with matplotlib, ggplot, MATLAB, etc. On the Plotly website, you can style your graphs with a GUI, so you don’t have to spend hours writing code that simply changes legend opacity.

Plotly does this all while backing up your graphs on the cloud, so that years later, you can find that may have otherwise been on a hard drive in a landfill. If you make your data public, other people can also find your graphs and data. The best practice that we have today for saving and sharing research data is to entomb it as a thesis in the engineering library basement. All that is changing.

Like d3.js? Like interactive, NYT graphics? So do we. Now, with the Plotly APIs (https://plot.ly, you can make them yourself without being an expert web programmer. If you are an expert programmer, now you have scientific languages and tools like R, Python, Pandas, and MATLAB instead of javascript to wrangle your data and create beautiful data visualizations. Science meets the world-wide-web. Engineering meets design. Let’s do this.

I’m going to show you this brave new world below, starting with bubble charts. Bubble charts are sweet because they take advantage of the innate interactivity of Plotly graphs. When you hover bubble chart point, you want to see what its size represents, you want to zoom-in to points that are clustered, and you want to pan around once you’re zoomed-in. You become a Bubble Chart Explorer. Plotly lets you do all this, all while upping the game for scientific, publication-quality graphics.

**The Core IPython team (2013)**

![Profiles of the Core IPython team members](https://plot.ly/assets/ipython-team-profiles.png)

Plus Jonathan Frederic (https://github.com/jdfreder) and Zach Sailer (https://github.com/Zsailer)!
An incomplete cast of characters

- **Brian Granger** - Physics, Cal State San Luis Obispo
- **Min Ragan-Kelley** - Nuclear Engineering, UC Berkeley
- **Matthias Bussonnier** - Physics, Institut Curie, Paris
- **Jonathan March** - Enthought
- **Thomas Kluyver** - Biology, U. Sheffield
- **Paul Ivanov** - Neuroscience, UC Berkeley.
- **Robert Kern** - Enthought
- **Evan Patterson** - Physics, Caltech/Enthought
- **Brad Froehle** - Mathematics, UC Berkeley
- **Stefan van der Walt** - UC Berkeley
- **John Hunter** - TradeLink Securities, Chicago.
- **Satra Ghosh** - MIT Neuroscience
- **Gaël Varoquaux** - Neurospin (Orsay, France)
- **Ville Vainio** - CS, Tampere University of Technology, Finland
- **Barry Wark** - Neuroscience, U. Washington.
- **Ondrej Certik** - Physics, LANL
- **Darren Dale** - Cornell
- **Justin Riley** - MIT
- **Mark Voorhies** - UC San Francisco
- **Nicholas Rougier** - INRIA Nancy Grand Est
- **Thomas Spura** - Fedora project

Many more! (~220 commit authors)

Public "Lab meetings on air"
In [32]: YouTubeVideo('UUjTAq8cCcs', width=600, height=500)

Out[32]:

Current IPython funding

![Alfred P. Sloan Foundation](logo.png)
![Microsoft](logo.png)
![Simons Foundation](logo.png)

Note: We're hiring! Machine learning web pipelines for time-series analysis (astro, geo, neuro). With Josh Bloom (UCB Astro), NSF funding. Talk to me!
Prior IPython support, thanks!!

- **Enthought**, Austin, TX: Lots!
- **Microsoft**: WinHPC support, Visual Studio integration, Azure
- **DoD/DRC Inc**: 2011/12 (thanks to Jose Unpingco and Chris Keees).
- Indirect: NIH via NiPy grant, NSF via Sage grant.
- Tech-X Corp., Boulder, CO: Parallel/notebook (previous versions)

**A new $37.8M initiative in Data Science**

- Moore/Sloan Foundations, 5 year support for UC Berkeley, U. Washington, NYU.
- Open source scientific computing tools will be central to this effort.
- We're hiring: Executive Director (now) ([http://vcresearch.berkeley.edu/datascience/career-opportunities](http://vcresearch.berkeley.edu/datascience/career-opportunities)), Data Science Fellows (soon).

Join us for the launch of the Berkeley Institute for Data Science
December 12, 2013, 1:00 - 3:00 pm
Banatao Auditorium, Sutardja Dai Hall
- My talk from today is a slightly updated version of this one from NIPS last December:


- The interactive widgets and other new features are documented in this notebook collection:

http://nbviewer.ipython.org/urls/raw.githubusercontent.com/ipython/ipython/master/examples/Index.ipynb

(that's the link I tried to open on nbviewer which failed, we just moved the URL yesterday).

- A gallery of interesting IPython Notebooks:


- Some videos in case folks are interested:

http://ipython.org/videos.html