

# IPython e *notebooks* Jupyter

Thiago Teixeira Santos  
<thiago.santos@embrapa.br>

17 de novembro de 2017

## 1 IPython

IPython é um ambiente de **computação interativa** que enriquece os recursos originais do console Python padrão. Na computação interativa, fragmentos de código podem ser inseridos pelo pesquisador, executados imediatamente e seus resultados se tornam prontamente disponíveis. O sistema mantém seu estado em memória, isto é, seu conjunto de variáveis em sua configuração atual é acessível ao usuário. [Perez e Granger \(2007\)](#) argumentam sobre as vantagens de um ambiente interativo ao pesquisador:

"Este estilo flexível casa bem com o espírito da computação em um contexto científico, no qual a determinação de quais computações devem ser realizadas em seguida geralmente requer um esforço significativo [por parte do pesquisador]. Um ambiente interativo permite aos cientistas examinar os dados, testar novas ideias, combinar algoritmos e avaliar os resultados diretamente."

IPython provê ainda recursos para paralelizar a execução do código, com pouco esforço adicional, permitindo ao pesquisador tirar proveito de processadores multi-core e *clusters* de computação<sup>1</sup>.

<sup>1</sup> O leitor interessado pode visitar [Using IPython for parallel computing](#) para maiores detalhes sobre computação paralela com IPython.

- IPython é um sistema para computação científica **interativa**
- Estende a funcionalidade básica do interpretador Python
- Possui recursos para **visualização de dados**
- Facilita o emprego de **computação paralela e distribuída**

### 1.1 Console básico

O console básico pode ser iniciado executando-se o comando `ipython`:

```
$ ipython
Python 2.7.5+ (default, Feb 27 2014, 19:37:08)
Type "copyright", "credits" or "license" for more information.

IPython 0.13.1 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
```

## TOOLBOX

# INTERACTIVE NOTEBOOKS: SHARING THE CODE

*The free IPython notebook makes data analysis easier to record, understand and reproduce.*



BY HELEN SHEN

researchers to keep a detailed lab notebook for their computational work.

used increasingly by scientists who want to keep detailed records of their work. devise teaching

Interactive notebooks: Sharing the code, Nature 515:7525

```
%quickref -> Quick reference.  
help      -> Python's own help system.  
object?   -> Details about 'object', use 'object??' for extra details.
```

```
In [1]:
```

## 1.2 Qtconsole

Qtconsole é um terminal IPython que utiliza uma interface gráfica baseada na biblioteca Qt. Ele pode ser inicializado através do comando:

```
$ ipython qtconsole
```

## 1.3 A opção pylab

A opção pylab faz com que o interpretador IPython importe para o ambiente os pacotes essenciais para arrays da NumPy e as rotinas de *plotting* da Matplotlib. Alternativamente, pode-se executar o comando "mágico" (*magic command*) %pylab dentro de um console IPython em execução.

```
$ ipython qtconsole --pylab
```

## 1.4 Exercício 1

Explore os *containers* da linguagem Python utilizando um console IPython. Utilize os recursos de auto-completar do console IPython para explorar a funcionalidade dessas estruturas de dados.

1. Crie uma lista. Visualize seu conteúdo. Adicione e remova elementos.
2. Crie um dicionário. Acesse os valores através de suas chaves. Adicione e remova novos pares chave/valor.
3. Crie e visualize tuplas.

## 1.5 Exercício 2

Compute o valor de  $\pi$  utilizando a fórmula de Wallis:

$$\pi = 2 \prod_{i=1}^{\infty} \frac{4i^2}{4i^2-1}$$

### 1.5.1 Solução

```
In [1]: a = range(1,10000,1)  
        pi = 2.  
        for i in a:  
            pi = pi * 4*i**2/(4*i**2-1)  
        pi
```

```
Out[1]: 3.141514110828136
```

## 1.6 Um caderno de laboratório executável: Jupyter Notebook

Um **Jupyter Notebook** é uma aplicação web que permite a mesma computação interativa dos consoles, mas oferece funcionalidade adicional. Utilizando um navegador web, o pesquisador pode organizar trechos de anotações e código de maneira flexível. Texto e código são organizados em **células** que podem ser inseridas, apagadas, reorganizadas e executadas conforme necessário.

Uma notebook Jupyter pode apresentar gráficos, fórmulas matemáticas e resultados de execução de código (**Julia**, **Python** ou **R**), tudo organizado em um único documento. Notebooks vêm sendo utilizados em anotações de pesquisa, redação de artigos científicos, análises de dados e na produção de livros ([DAVIDSON-PILON, 2014](#)).

- IPython notebooks são uma forma elegante de manter **anotações e código**
- Possibilitam a criação de "**artigos executáveis**" (*executable papers*)

Um servidor de notebooks iniciado com o comando:

```
$ jupyter notebook
```

- Jupyter notebooks estão impactando a divulgação de material científico.
  - Livros inteiros têm sido escritos utilizando-se notebooks
  - Notebooks podem ser publicados na Web, em sítios como [nbviewer.ipython.org](http://nbviewer.ipython.org)

### 1.6.1 Células

Notebooks são compostos por células de dois tipo: células de texto e células de código. Células de código contém trechos de código Python a ser executado. Já as células de texto podem conter anotações formatadas utilizando as convenções [Markdown](#).

- O conteúdo dos notebooks é dividido em **células**
- Há dois tipos de célula: *código* e *texto*
- Células de código contém *micro-programas* em Python
- Células de texto contém anotações, que podem ser facilmente formatadas utilizando-se a marcação [Markdown](#)

**Matemática** Células de texto suportam também **expressões matemáticas**, utilizando a sintaxe do LaTeX. Por exemplo, a expressão:

$$y = 3x^2 + \epsilon$$

produz a fórmula  $y = 3x^2 + \epsilon$

A seguir estão alguns exemplos de fórmulas matemáticas produzidas através da notação LaTeX<sup>1</sup>.

$\lim_{x \rightarrow \infty} \exp(-x) = 0$  produzido por `\lim_{x \to \infty} \exp(-x) = 0`

$\sqrt[n]{1+x+x^2+x^3+\dots}$  produzido por `\sqrt[n]{1+x+x^2+x^3+\ldots}`

$\int_0^\infty e^{-x} dx$  produzido por `\int_0^\infty \mathrm{e}^{-x} \mathrm{d}x`

$\sum_{i=1}^{10} t_i$  produzido por `\sum_{i=1}^{10} t_i`

<sup>1</sup> Para mais detalhes sobre como escrever fórmulas matemáticas na sintaxe do LaTeX, o leitor pode considerar [este guia no Wikibooks](#).

### 1.6.2 Como notebooks são armazenados

Notebooks são armazenados como arquivos JSON que guardam o conteúdo das células, incluindo os resultados das computações realizadas. Tais arquivos possuem a extensão `.ipynb` e podem ser convertidos em programas Python regulares e em documentos HTML ou PDF. O que faz dos notebooks uma excelente ferramenta para pesquisa reprodutível é que eles são **documentos executáveis** que armazenam não só descrições textuais e matemáticas de um procedimento, mas também o código necessário para replicar as computações.

- Os notebooks Python são armazenados como arquivos *JSON*
- Esses arquivos podem ser *convertidos* para outros formatos:
  - um script Python
  - um documento em reStructuredText
  - um documento em HTML
  - um documento LaTeX
  - um documento em Markdown
  - slides

- Para mais detalhes, execute:

```
$ ipython nbconvert --help
```

## 1.7 Visualização científica

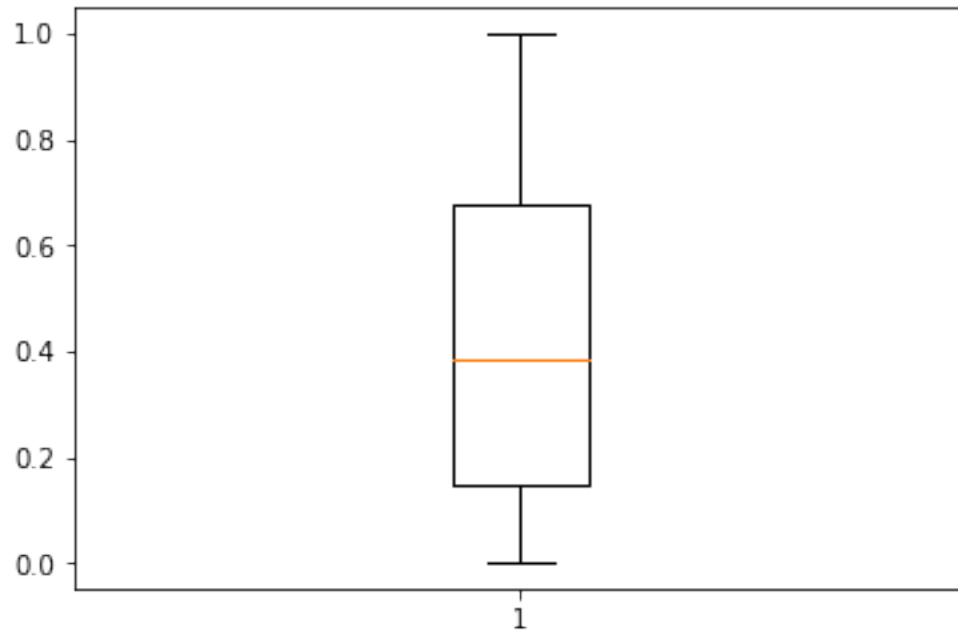
IPython (console ou na forma de notebooks Jupyter) integra-se facilmente à Matplotlib, permitindo a exibição de gráficos diretamente no navegador Web ou no console Qt. A criação de gráficos será introduzida posteriormente, na Seção *Matplotlib*.

```
In [2]: %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

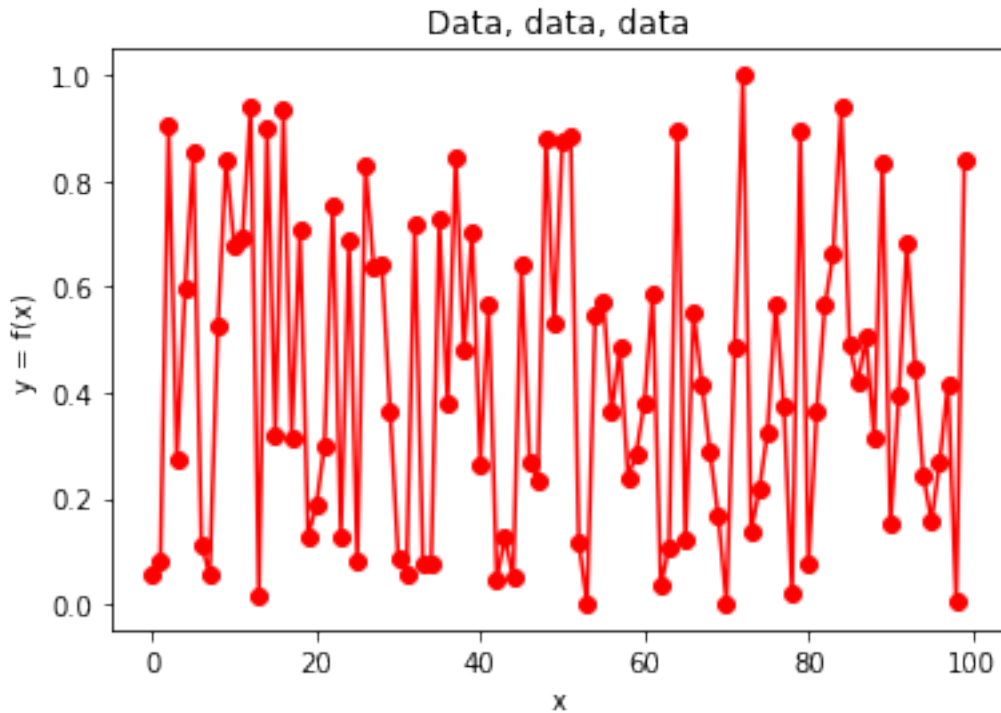
```
/usr/local/lib/python2.7/dist-packages/IPython/core/magics/pylab.py:161: UserWarning: pylab impo
`%matplotlib` prevents importing * from pylab and numpy
"\n`%matplotlib` prevents importing * from pylab and numpy"
```

```
In [3]: data = random.rand(100)
        bp = boxplot(data)
```



```
In [4]: plot(data, 'r-')  
        plot(data, 'ro')  
        title('Data, data, data')  
        xlabel('x')  
        ylabel('y = f(x)')
```

```
Out[4]: Text(0,0.5,u'y = f(x)')
```



## 1.8 IPython como um console de sistema

Podemos executar comandos do sistema diretamente do console IPython. Por exemplo, para listar arquivos no diretório corrente, podemos utilizar o comando de sistema `ls` (em sistemas baseados em Unix como Linux e Mac OS):

```
In [5]: ls
```

```
00._Aquecimento.ipynb
00._Aquecimento.pdf
01._Introducao.ipynb
01._Introducao.pdf
01-Introducao-slides.html
01._Introducao.slides.html
02._A_Linguagem_Python.ipynb
02._A_Linguagem_Python.pdf
02-A_Linguagem_Python-slides.html
02._A_Linguagem_Python.slides.html
03._IPython.ipynb
03._IPython.pdf
03-IPython-slides.html
03._IPython.slides.html
03._IPython.tex
04._NumPy.ipynb
```

```

04._NumPy.pdf
04-NumPy-slides.html
04._NumPy.slides.html
05._Matplotlib.ipynb
05._Matplotlib.pdf
05-Matplotlib-slides.html
05._Matplotlib.slides.html
05._Visualizacao_e_graficos.html
05._Visualizacao_e_graficos.ipynb
05._Visualizacao_e_graficos.slides.html
06._Scipy.ipynb
06._Scipy.pdf
06-Scipy-slides.html
06._Scipy.slides.html
07._GDAL.ipynb
07-GDAL-slides.html
07._GDAL.slides.html
08._Clustering_Hierarquico_Microarray.ipynb
08._Clustering_Hierarquico_Microarray.slides.html
10._Fenotipagem_-_Plantas_em_vaso.ipynb
11._Fenotipagem_-_Alinhamento_de_imagens_obtidas_por_VANTs.ipynb
data/
Doc131.pdf
figs/
LICENSE
MT803.ipynb
README.md
reveal.js/
trig.pdf
trig.tif
Untitled.ipynb

```

Utilizando ! antes do comando, podemos armazenar o resultado em uma variável Python:

```

In [6]: l = !ls figs

In [7]: l

Out[7]: ['fig.png',
         'fitting.png',
         'ipython-nature.png',
         'john-hunter.jpg',
         'python-nature.png',
         'python.png']

In [8]: figs = [f for f in l if f.endswith('.png')]

In [9]: figs

```



```
Out[9]: ['fig.png',  
        'fitting.png',  
        'ipython-nature.png',  
        'python-nature.png',  
        'python.png']
```

Comandos do sistema podem ser executados utilizando parâmetros armazenados em variáveis Python. O exemplo abaixo copia arquivos entre diretórios utilizando o comando de sistema `cp`. Os arquivos de interesse são lidos da lista `figs` e passados como argumento para o comando:

```
In [10]: for f in figs:  
        !cp figs/$f /tmp/
```

```
In [11]: ls /tmp/*.png
```

```
/tmp/fig.png      /tmp/ipython-nature.png  /tmp/python.png  
/tmp/fitting.png /tmp/python-nature.png   /tmp/qt-trayicon-Fs3892.png
```

## 1.9 Exercício 3

Explore o uso das células de texto (*Markdown*).

- Visite <http://www.markitdown.net/markdown> e teste várias formatações de texto (**negrito**, *itálico*, code, etc.).
- Visite <http://en.wikibooks.org/wiki/LaTeX/Mathematics> e teste várias fórmulas matemáticas em seu notebook.