

IPyNBtut

November 19, 2014

1 IPython Notebook: más que una consola en el navegador

1.0.1 Milton Labanda <- Loja

1.0.2 TIC_ec 2014 <- Universidad de Cuenca

Requerimientos para el tutorial:

- Librerías Sistema:
- pandoc
- linuxdoc-tools-latex
- texlive-full (para generación correcta de pdfs)
- Librerías Python:
- ipython 2.0+
- pandas
- openpyxl
- matplotlib
- rpy2

1.1 Introducción a IPython

- Más que intérprete de comandos mejorado para python
- Sistema de ayuda incluido
- Exploración de objetos (?)
- Autocompletado de código con la tecla Tab
- Funciones mágicas (%)
- Historial de comandos y salidas
- Llamadas a comandos del sistema (!)
- Se puede ejecutar código de otros lenguajes: *R, Octave, Cython, Bash, Perl, Ruby, etc.*

1.2 Que es el Notebook de IPython

- “El paper ejecutable para los científicos”
- Un intérprete interactivo avanzado que ha revolucionando la manera en que se utiliza Python en ámbitos *científicos, conferencias y tutoriales*
- Mucho ahorro de tiempo en escritura de código y pruebas
- Permite escribir en formato *Markdown, HTML, o texto puro (raw)*

- Permite escribir en **LateX**
- Permite incluir imágenes, links a archivos locales, videos, etc
- Exporta hacia diferentes formatos tales como: PDF, RST, Latex, reveal.js, etc
- Trabaja iniciando **servidor web** local y luego sirviendo páginas desde tal servidor: **Conozcámoslo entonces !!! ...**

1.3 Trabajar con el teclado en IPython Notebook

(Conjunto compatible de atajos de teclado) - Shift-Enter : run cell - Ctrl-Enter : run cell in-place - Alt-Enter : run cell, insert below - Ctrl-m x : cut cell - Ctrl-m c : copy cell - Ctrl-m v : paste cell - Ctrl-m d : delete cell - Ctrl-m z : undo last cell deletion - Ctrl-m : : split cell - Ctrl-m a : insert cell above - Ctrl-m b : insert cell below - Ctrl-m o : toggle - Ctrl-m O : toggle output scroll - Ctrl-m l : toggle line numbers - Ctrl-m s : save notebook - Ctrl-m j : move cell down - Ctrl-m k : move cell up - Ctrl-m y : code cell - Ctrl-m m : markdown cell - Ctrl-m t : raw cell - Ctrl-m 1-6 : heading 1-6 cell - Ctrl-m p : select previous - Ctrl-m n : select next - Ctrl-m i : interrupt kernel - Ctrl-m . : restart kernel - Ctrl-m h : show keyboard shortcuts

1.4 Formateando el texto en IPyNB

Texto puro

1.5 Marckdown

2 Título 1

2.1 Título 2

2.1.1 Título 3

Lista: - uno - dos - tres

Característica	Valor
Sistema Operativo	Linux Mint
Ram	4MG
Disco	100GB
Lenguaje de Programación	Python

Tablas en Markdown

Formateando código fuente en markdown Javascript

```
console.log('Hello world')
```

Java

```
public static void main(String [] args){
    System.out.println("hola mundo");
}
```

Python

```
def hello_ipython_notebook():
    print "Hello IPython Notebook"
```

R

```
df <- data.frame(x=c('a','e','i','o','u'), y=1:5)
```

2.2 HTML

negrita cursiva subrayado

3 LATEX

$$\text{polinomio} \rightarrow x^4 - 11x^3 + 9x^2 + 11x - 10$$

$$\pi r^2$$

$$D_{KL}(P||Q) = \sum_i \ln\left(\frac{P(i)}{Q(i)}\right)P(i)$$

3.1 Comandos mágicos de IPython Notebook

```
In [2]: %lsmagic
```

```
Out[2]: Available line magics:
```

```
%alias %alias_magic %autocall %automagic %autosave %bookmark %cat %cd %clear %colors %
```

```
Available cell magics:
```

```
%! %%HTML %%SVG %%bash %%capture %%debug %%file %%html %%javascript %%latex %%perl %
```

```
Automagic is ON, % prefix IS NOT needed for line magics.
```

```
In [15]: %%file /tmp/boo.txt
         esto es una prueba
         desde un ipynote
```

```
Writing /tmp/boo.txt
```

```
In [36]: %who_ls
```

```
Out[36]: ['math']
```

```
In [38]: %matplotlib inline
```

```
In []: %config InlineBackend.figure_format = 'png'
```

3.2 Interacción con el Sistema Operativo

```
In [24]: %ls data
```

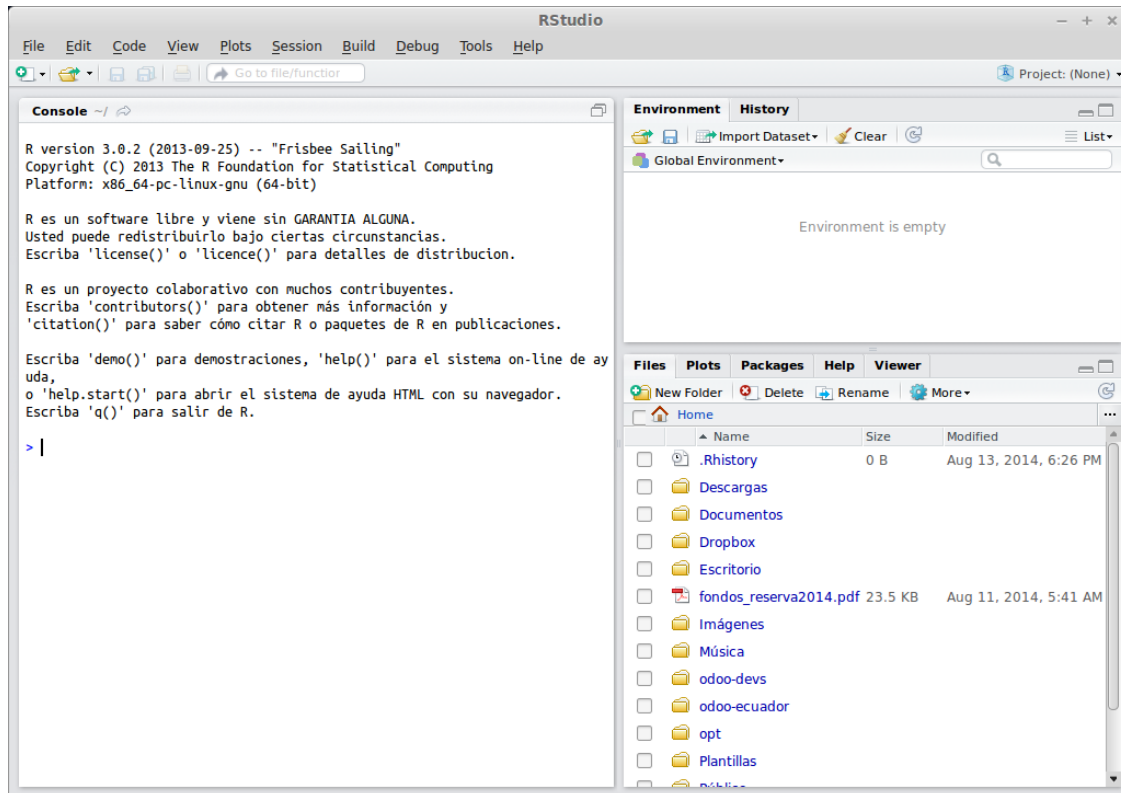
```
housing.csv sgaunl.csv sgaunl.xlsx
```

```
In [37]: %pwd
```

```
Out[37]: u'/home/miltonlab/Documentos/pyresearch/ticec2014'
```

3.3 Contenido hypermedia en IPython Notebook

```
In [60]: #Mostrar imágenes
from IPython.display import Image
from IPython.display import display
display(Image('images/rstudio.png', width=700, height=450))
```



```
In [5]: #Videos embebidos de Youtube
from IPython.display import YouTubeVideo
YouTubeVideo('nRHBdkxVn48')
```

```
Out[5]: <IPython.lib.display.YouTubeVideo at 0x7f2519f82710>
```

```
In [3]: #Incrustar páginas HTML
from IPython.display import HTML
HTML('<iframe src="http://pycon.com.ar/pyconar2014/" width=500 height=600>')
```

```
Out[3]: <IPython.core.display.HTML at 0x7f202c156410>
```

3.4 Introducción al trabajo científico con Python

```
In [49]: import numpy
import matplotlib
```

```
In [43]: a = numpy.array([[5,2,9,4],[1,8,8,9],[6,6,4,7],[3,5,6,2]])
a
# Seleccionar columna 2
```

```

a[:,2]
# Seleccionar fila 2
a[2,:]
# Selección de un subarreglo
a[0:2,1:3]
# Selección condicional
numpy.where(a%2==0,a,0)
a>5
a*3
# Determinante de un arreglo
numpy.linalg.det(a)
b = numpy.array([[5, 4, 2, 0], [9, 9, 6, 1], [3, 0, 5, 3], [8, 0, 6, 1]])
b
c = a * b
c

```

```

Out[43]: array([[25,  8, 18,  0],
               [ 9, 72, 48,  9],
               [18,  0, 20, 21],
               [24,  0, 36,  2]])

```

```

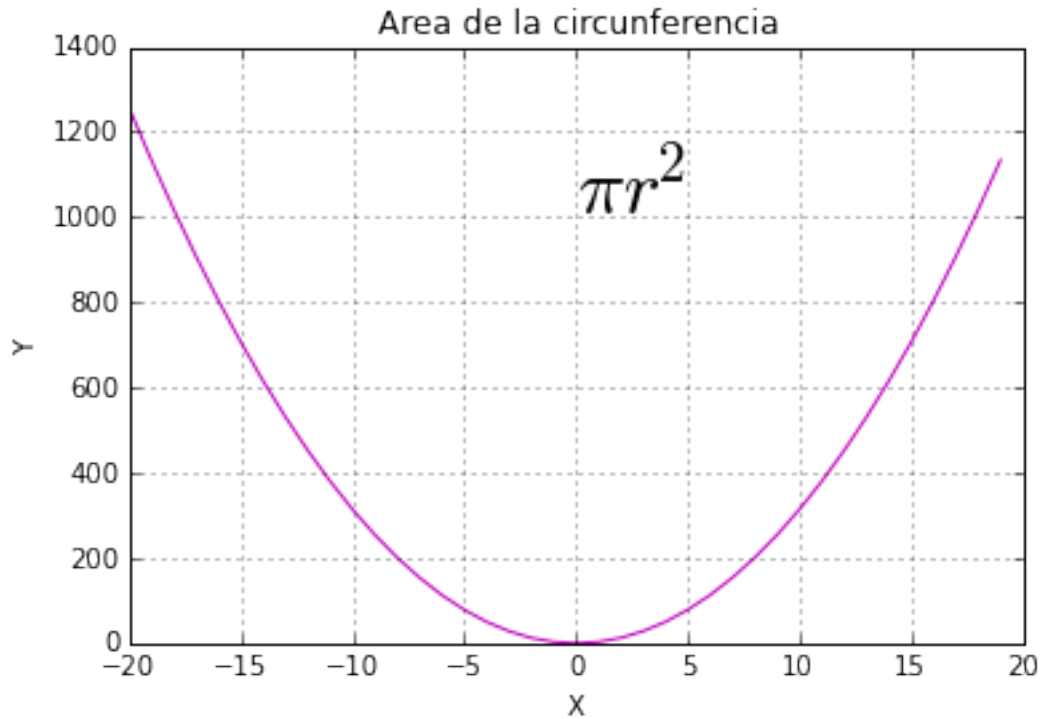
In [47]: x=numpy.arange(-20,20)
matplotlib.pyplot.title('Area de la circunferencia')
matplotlib.pyplot.xlabel('X')
matplotlib.pyplot.ylabel('Y')
matplotlib.pyplot.grid(True)
# Latex
matplotlib.pyplot.text(0,1000, r'$\pi r^2$', fontsize=30)
matplotlib.pyplot.plot(x, x*x*numpy.pi, 'm')
#pyplot.show()

```

```

Out[47]: [<matplotlib.lines.Line2D at 0x7fecfbf116d0>]

```

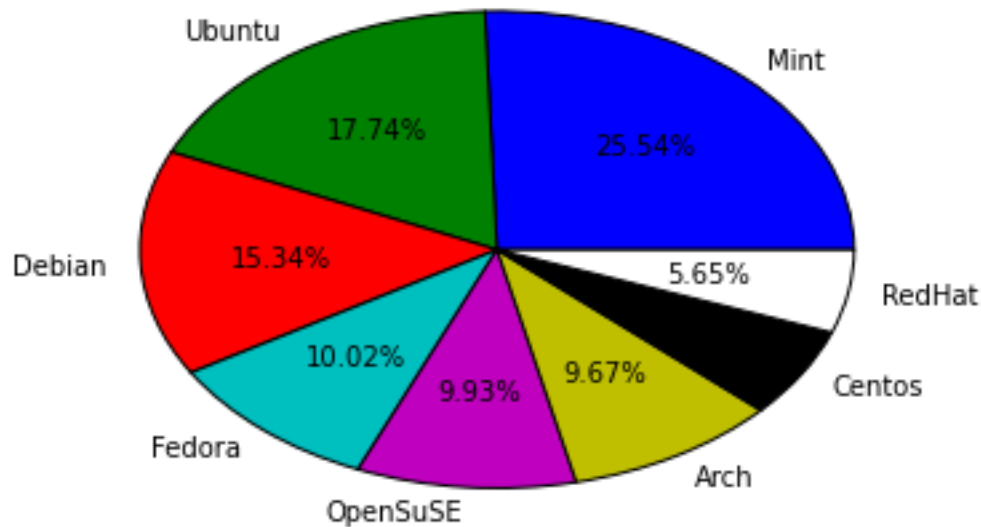


```
In [9]: # Graficos de Pastel
distros = ['Mint', 'Ubuntu', 'Debian', 'Fedora', 'OpenSuSE', 'Arch',
           'Centos', 'RedHat']
# Datos según http://distrowatch.com la tercera semana de Mayo del 2014
ranking = [3105, 2157, 1865, 1218, 1207, 1176, 743, 687]
```

```
matplotlib.pyplot.pie(ranking, labels=distros, autopct='%.2f%%')
```

```
Out[9]: ([<matplotlib.patches.Wedge at 0x7fdedac4e210>,
<matplotlib.patches.Wedge at 0x7fdedac5b090>,
<matplotlib.patches.Wedge at 0x7fdedac5be50>,
<matplotlib.patches.Wedge at 0x7fdedac68c50>,
<matplotlib.patches.Wedge at 0x7fdedac73a50>,
<matplotlib.patches.Wedge at 0x7fdedac00850>,
<matplotlib.patches.Wedge at 0x7fdedac0a650>,
<matplotlib.patches.Wedge at 0x7fdedac18410>],
[<matplotlib.text.Text at 0x7fdedac4e7d0>,
<matplotlib.text.Text at 0x7fdedac5b650>,
<matplotlib.text.Text at 0x7fdedac68450>,
<matplotlib.text.Text at 0x7fdedac73250>,
<matplotlib.text.Text at 0x7fdedac73f50>,
<matplotlib.text.Text at 0x7fdedac00e10>,
<matplotlib.text.Text at 0x7fdedac0abd0>,
<matplotlib.text.Text at 0x7fdedac189d0>],
[<matplotlib.text.Text at 0x7fdedac4ec10>,
<matplotlib.text.Text at 0x7fdedac5ba10>,
<matplotlib.text.Text at 0x7fdedac68810>],
```

```
<matplotlib.text.Text at 0x7fddedac73610>,
<matplotlib.text.Text at 0x7fddedac00410>,
<matplotlib.text.Text at 0x7fddedac0a210>,
<matplotlib.text.Text at 0x7fddedac0af90>,
<matplotlib.text.Text at 0x7fddedac18d90>])
```



3.5 Análisis de Datos con Python desde IPyNotebook

```
In [50]: import pandas
# Datos de ingresos de extranjeros al ecuador primer cuatrimestre según:
# http://servicios.turismo.gob.ec/images/estadisticas/Boletin-abril-2014-comprimido.pdf
turismo = pandas.DataFrame({'Anio': [2010, 2011, 2012, 2013, 2014],
                             'enero' : [96109, 105548, 127116, 130842, 152576],
                             'febrero' : [89924, 86421, 99521, 103768, 124584],
                             'marzo' : [82452, 87495, 96948, 113361, 114007],
                             'abril' : [70540, 87507, 92627, 89669, 118614]
                             })
```

```
In [22]: turismo
```

```
Out[22]:
```

	Anio	abril	enero	febrero	marzo
0	2010	70540	96109	89924	82452
1	2011	87507	105548	86421	87495
2	2012	92627	127116	99521	96948
3	2013	89669	130842	103768	113361
4	2014	118614	152576	124584	114007

```
In [23]: turismo.columns
```

```
Out[23]: Index([u'Anio', u'abril', u'enero', u'febrero', u'marzo'], dtype='object')
```

```
In [24]: turismo.describe()
```

```
Out[24]:
```

	Anio	abril	enero	febrero	marzo
count	5.000000	5.000000	5.000000	5.000000	5.000000
mean	2012.000000	91791.400000	122438.200000	100843.600000	98852.600000
std	1.581139	17281.729986	22244.652643	15009.638846	14506.479321
min	2010.000000	70540.000000	96109.000000	86421.000000	82452.000000
25%	2011.000000	87507.000000	105548.000000	89924.000000	87495.000000
50%	2012.000000	89669.000000	127116.000000	99521.000000	96948.000000
75%	2013.000000	92627.000000	130842.000000	103768.000000	113361.000000
max	2014.000000	118614.000000	152576.000000	124584.000000	114007.000000

```
In [25]: turismo['enero']
```

```
Out[25]: 0    96109
         1   105548
         2   127116
         3   130842
         4   152576
         Name: enero, dtype: int64
```

```
In [26]: turismo[['enero', 'febrero']]
```

```
Out[26]:
```

	enero	febrero
0	96109	89924
1	105548	86421
2	127116	99521
3	130842	103768
4	152576	124584

```
In [27]: # Aquellos años que en enero tienen valores sobre la media
         turismo[turismo.enero > turismo.enero.mean()] [['Anio', 'enero']]
```

```
Out[27]:
```

	Anio	enero
2	2012	127116
3	2013	130842
4	2014	152576

```
In [53]: unl = pandas.read_csv('./data/sgaunl.csv')
```

```
In [14]: unl
```

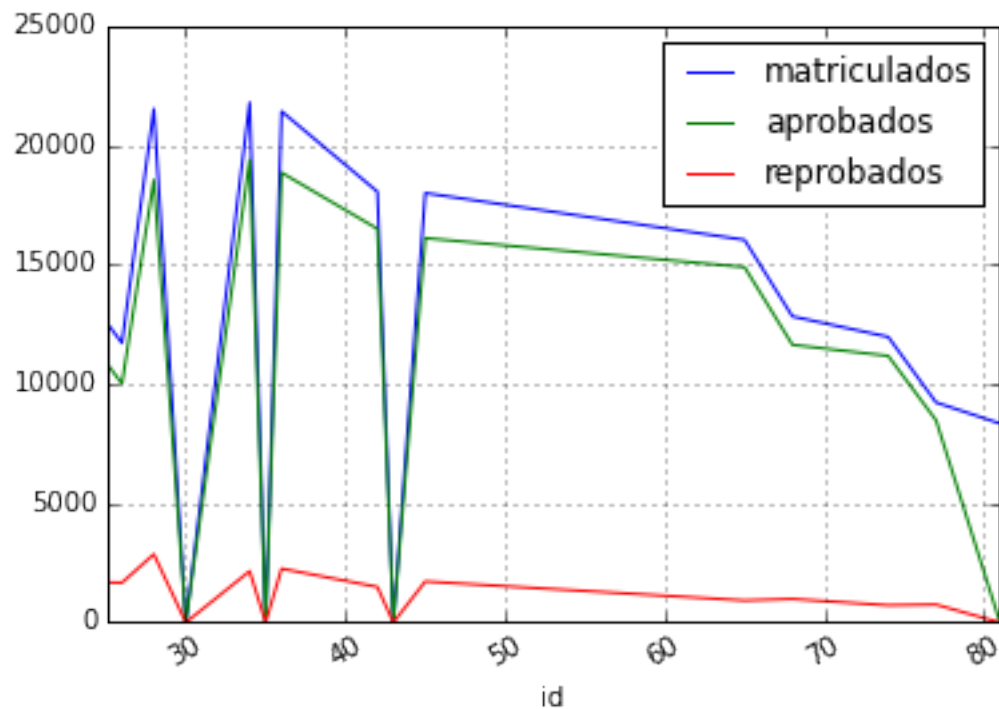
```
Out[14]:
```

	Unnamed: 0	id	nombre	matriculados
0	8	25	Pregrado Septiembre 2008 - Febrero 2009	12616
1	9	26	Pregrado Marzo 2009 - Julio 2009	11723
2	10	28	Pregrado Septiembre 2009 - Febrero 2010	21545
3	4	30	Admisiones Pregrado	0
4	13	34	Pregrado Marzo 2010 - Julio 2010	21810
5	11	35	Admisiones Pregrado 2010	1
6	12	36	Pregrado Septiembre 2010 Febrero 2011	21439
7	18	42	Pregrado Marzo 2011 - Julio 2011	18071
8	16	43	Admisiones Pregrado 2011	0
9	35	45	Pregrado Septiembre 2011 Febrero 2012	18011
10	34	65	Pregrado Marzo 2012 Julio 2012	16049
11	40	68	Pregrado Septiembre 2012 Febrero 2013	12837
12	45	74	Pregrado Marzo 2013 Julio 2013	11980
13	43	77	Pregrado Septiembre 2013 Febrero 2014	9227
14	50	81	Pregrado Marzo-Julio 2014	8347

	aprobados	reprobados
0	10917	1680
1	10012	1668
2	18563	2863
3	0	0
4	19413	2152
5	1	0
6	18868	2258
7	16515	1492
8	0	0
9	16121	1721
10	14907	923
11	11647	984
12	11182	722
13	8478	749
14	8	0

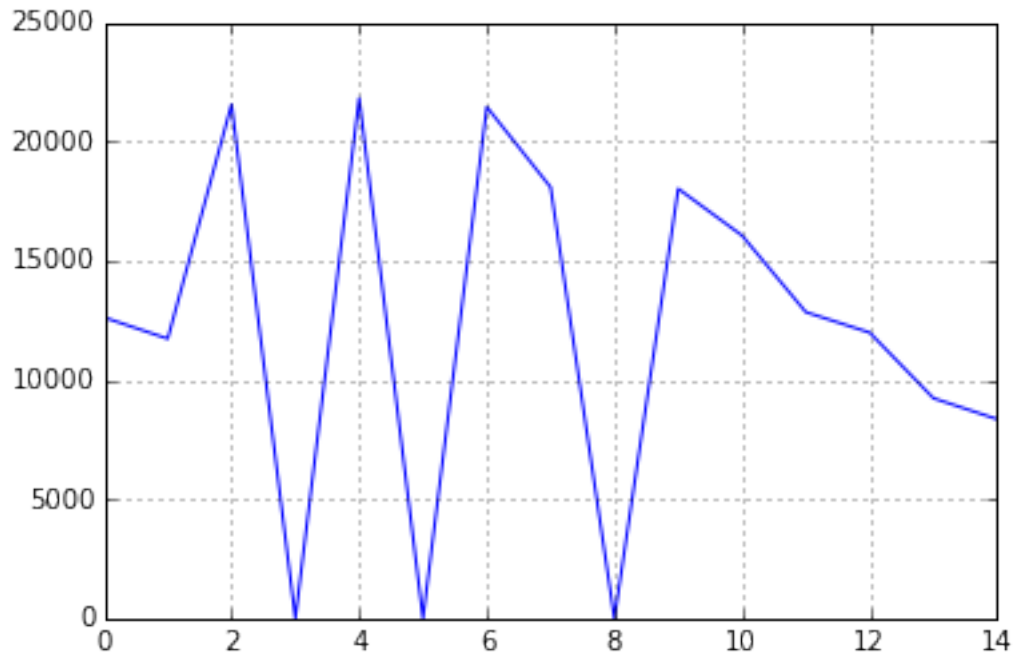
```
In [54]: seccion = unl[['matriculados', 'aprobados', 'reprobados']]
seccion.plot(x=unl.id, rot=30)
```

```
Out[54]: <matplotlib.axes._subplots.AxesSubplot at 0x7fed04077f50>
```



```
In [55]: unl.matriculados.plot(x_compat=True)
```

```
Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x7fecfbd91990>
```



```
In [58]: unl.to_excel('./data/sgaunl.xlsx')
```

```
In [33]: # Consumo de datos en formato json desde el portal http://datosabiertos.ec/
from urllib2 import urlopen
urlagua = urlopen('http://www.datosabiertos.ec/storage/f/2013-10-25T06%3A45%3A20Z/ecuador-')
dataagua = pandas.read_json((urlagua))
dataagua.head(4)
```

```
Out[33]:
```

	created_at	id	maximoAmazonia	maximoConsumo	maximoCosta	maximoSierra	mediaAmazonia	mediaConsumo	mediaCosta	mediaSierra	valorMediaAmazonia	valorMediaCosta	valorMediaSierra	valorMinimo	valorMinimoAmazonia	valorMinimoCosta	valorMinimoSierra	valorTotalAmazonia	valorTotalCosta	valorTotalSierra
0	2013-10-25 06:45:20	1	999	999	999	999	100.42	34.15	32.61	35.65	10.71	11.25	11.38	0.89	1	0.99	0.89	879.733	11.907	17.438

```
[1 rows x 34 columns]
```

```
In [34]: len (dataagua.columns)
```

```
Out[34]: 34
```

3.6 Programando en “R” desde IPython Notebook

```
In [4]: %load_ext rmagic

In [6]: %%R
        setwd('.')
        fileUrl <- 'https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06hid.csv'
        download.file(fileUrl, destfile='./data/housing.csv', method='curl')

In [8]: %R list.files('./data')

Out[8]: array(['housing.csv'],
              dtype='|S11')

In [44]: %%R
        house <- read.csv('./data/housing.csv')
        ## For the state of Idaho how many properties are worth £1,000,000 or more?
        expensives <- subset(house, ST==16 & VAL==24)
        print (nrow((expensives)))
        print (ncol(expensives))

[1] 53
[1] 188
```

3.7 Widgets interactivos (sobresaliente en IPython 2.0+)

```
In [1]: from IPython.html import widgets
        from IPython.display import display
        sld = widgets.FloatSliderWidget(
            value=30, min=2, max=32, step=1, description='Test Float Widget'
        )
        chkb = widgets.CheckboxWidget(
            value=True, description='Test Checkbox'
        )
        drpd = widgets.DropdownWidget(
            values=[1,2,3,4,5,6], value=2, description='Test DropDown'
        )

        display(sld)
        display(chkb)
        display(drpdp)

In [75]: def conectar(val):
        print val
        return val
        w = widgets.interactive(conectar, val=widgets.FloatSliderWidget(
            value=128.0, min=0, max=255.0, step=1.0))
        display(w)

128.0
```

4 Links interesantes

- <http://pyvideo.org/video/2798/ipython-revealjs-attacks-again-but-now-it-is-alive> [IPyNb with Reveal.js alive]

- <http://nbviewer.ipython.org/gist/ellisonbg/3837783/TwitterNetworkX.ipynb> [Analysis of twitter stream data]
- <http://nbviewer.ipython.org/github/NelisW/ComputationalRadiometry/blob/master/01-IPythonHintsAndTips.ipynb>