

Nociones básicas sobre el uso de computadores con sistema operativo tipo Linux y de programación aplicada a ciencias farmacéuticas.





¿Cuales son las ventajas de Linux frente a Windows?, ¿en qué se diferencian?

La instalación:

- En Linux a pesar de todos los esfuerzos la instalación no resulta sencilla siempre, pero te permite personalizar totalmente los paquetes que quieras instalar.
- En Windows la instalación es mínimamente configurarle aunque es muy sencilla.

La compatibilidad: Ninguno de los dos sistemas operativos son totalmente compatibles con el Hardware, a pesar de que Windows se acerca más, los dos están cerca de conseguirlo.

- Aunque Linux no esta detrás de ninguna casa comercial gracias a su elevada popularidad ofrece una alta compatibilidad ofreciendo, además, actualizaciones frecuentes.
- Windows al ser parte de Microsoft intenta ofrecer una gran cantidad de drivers ya que su gran poder económico hace que las empresas mismas de hardware creen sus propios drivers.

Software:

- Linux al tener menos software en algunos campos sufre una menor aceptación por parte de las empresas, aunque gracias a los apoyos de empresas como Sun Microsystems o IBM se ha logrado muchos avances.
- Windows al ser el más fácil de usar en las empresas, posee una gran cantidad de software.

Robustez:

- Linux se ha caracterizado siempre por la robustez de su sistema ya que pueden pasar meses e incluso años sin la necesidad de apagar o reiniciar el equipo, también si una aplicación falla simplemente no bloquea totalmente al equipo.
- En Windows siempre hay que reiniciar cuando se cambia la configuración del sistema, se bloquea fácilmente cuando ejecuta operaciones aparentemente simples por lo que hay que reiniciar el equipo.



Razones para cambiar:

- Es software libre, lo que quiere decir que no hay que pagar nada por el sistema en sí.
- Es un sistema operativo muy fiable ya que hereda la robustez de UNIX.
- Ideal para las redes ya que fue diseñado en Internet y para Internet
- No es cierto que tenga pocos programas, solo en algún campo muy específico.
- Es 100% configurarle.
- Es el sistema más seguro, ya que al disponer del código fuente cualquiera puede darse cuenta de algún fallo, se puede decir que decenas de miles de personas velan por tu seguridad.
- Existe muchísima documentación, también en español gracias a los proyectos como LUCAS.
- Cuenta con el soporte de muchas grandes empresas como IBM, Corel, Lotus, Siemens, Motorola, Sun, etc.
- Puedes encontrar ayuda en millones de sitios en Internet como los foros.
- Es muy portable, si tienes un Mac un Alpha o un Sparc puedes usar Linux sin problemas.



MacTM OS



Windows 7
Windows 8

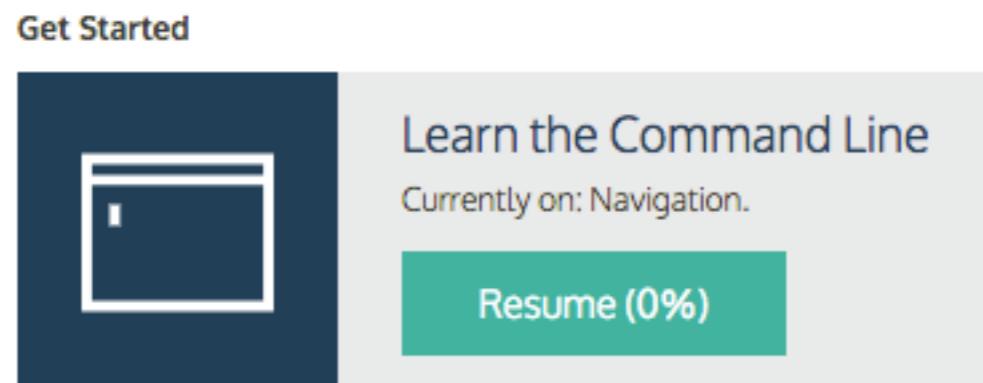


Tutorial - Linux

1. Ir a sitio web: www.codecademy.com
2. Iniciar sesión y realizar curso: **Learn the Command Line**



Get Started



A screenshot of a Codecademy course card. It features a dark blue 'Get Started' button on the left. To its right is a light gray box containing a white icon of a terminal window with a single character inside. Next to the icon, the text 'Learn the Command Line' is written in blue, and below it, 'Currently on: Navigation.' is written in smaller gray text. At the bottom of the light gray box is a teal button with the text 'Resume (0%)' in white.

Tutorial - Linux

Conectarse mediante protocolo SSH a servidor *chernobyl* @152.74.100.143

1.1 Abrir PuTTy, MovaXterm o CMD

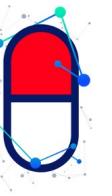
Conectarse al servidor invitado01@152.74.100.143 mediante comando:

```
$ ssh invitado1@152.74.100.143 → Passwd: ramirezlab2022
```

2. Abrir documento: 04_Commandos_de_consola.pdf

3. Crear directorio personal en \$ Documents/

4. Entrar a Directorio personal: \$ cd Documents/MyFolder

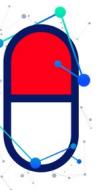


Linux Shell Scripting tutorial



el **shell** o intérprete de órdenes o intérprete de comandos es el **programa informático** que provee una **interfaz de usuario** para acceder a los servicios del **sistema operativo**. (Wikipedia)

- 1. Usar cualquier editor de texto como vi para escribir el script your_script_name**
- 2. Despues de escribir el script, darle permisos de ejecución, por ejemplo**
 - 1. chmod permission your_script_name**
 - 1. \$ chmod -x your_script_name**
 - 2. \$ chmod 755 your_script_name**
 - 3. \$ chmod 777 your_script_name**
- 4. Ejecutar el script con las siguientes ordenes:**
 - 1. \$ bash your_script_name**
 - 2. \$ sh your_script_name**
 - 3. \$./your_script_name**



Linux Shell Scripting tutorial

Common vi editor command list

For this Purpose	Use this vi Command Syntax
To insert new text	esc + i (You have to press 'escape' key then 'i')
To save file	esc + : + w (Press 'escape' key then 'colon' and finally 'w')
To save file with file name (save as)	esc + : + w "filename"
To quit the vi editor	esc + : + q
To quit without saving	esc + : + q!
To save and quit vi editor	esc + : + wq
To search for specified word in forward direction	esc + /word (Press 'escape' key, type /word-to-find, for e.g. to find word 'shri', type as /shri)
To continue with search	n
To search for specified word in backward direction	esc + ?word (Press 'escape' key, type word-to-find)
To copy the line where cursor is located	esc + yy
To paste the text just deleted or copied at the cursor	esc + p
To delete entire line where cursor is located	esc + dd
To delete word from cursor position	esc + dw
To Find all occurrence of given word and Replace then globally without confirmation	esc + :\$s/word-to-find/word-to-replace/g For. e.g. :\$s/mumbai/pune/g Here word "mumbai" is replace with "pune"
To Find all occurrence of given word and Replace then globally with confirmation	esc + :\$s/word-to-find/word-to-replace/cg
To run shell command like ls, cp or date etc within vi	esc + !shell-command For e.g. :!pwd

Linux Shell Scripting tutorial

1. Escribiendo el script.

```
$ vi first
#
# My first shell script
#
clear
echo "Curso farmacoinformatica - 2023"
```

2. Después de guardar el script, se puede ejecutar así:

```
$ ./first
```

3. Esto no ejecutará el script porque le hacen falta permisos, así que:

```
$ chmod 777 first
$ ./first
```

Script Command(s)	Meaning
\$ vi first	Start vi editor
# # My first shell script #	# followed by any text is considered as comment. Comment gives more information about script, logical explanation about shell script. <i>Syntax:</i> # comment-text
clear	clear the screen
echo "Knowledge is Power"	To print message or value of variables on screen, we use echo command, general form of echo command is as follows <i>syntax:</i> echo "Message"

Linux Shell Scripting tutorial

Ejercicio: Escribir el siguiente script, guardarla y ejecutarlo (después de darle permisos de ejecución):

```
$ vi scrit-1
#
#
# Script to print user information who currently login, current date & time
#
clear
echo "Hello $USER"
echo "Today is \c ";date
echo "Number of user login : \c" ; who | wc -l
echo "Calendar"
cal
exit 0
```

Linux Shell Scripting tutorial

System Variable	Meaning
BASH=/bin/bash	Our shell name
BASH_VERSION=1.14.7(1)	Our shell version name
COLUMNS=80	No. of columns for our screen
HOME=/home/vivek	Our home directory
LINES=25	No. of rows for our screen
LOGNAME=students	students Our logging name
OSTYPE=Linux	Our Os type
PATH=/usr/bin:/sbin:/bin:/usr/sbin	Our path settings
PS1=[\u@\h \W]\\$	Our prompt settings
PWD=/home/students/Common	Our current working directory
SHELL=/bin/bash	Our shell name
USERNAME=vivek	User name who is currently login to this PC



Python Environment To Works!



Python

□ Interactive

Default prompt >>>

```
>>> print("Hello world")
Hello world
>>>
```

□ IPython shell

```
In [1]: print("Hello world")
Hello world
In [2]: "Hello world"
Out[2]: Hello world
```

□ From a file

File hello.py

```
print("Hello world")
```

```
C:\>python hello.py
Hello world
```

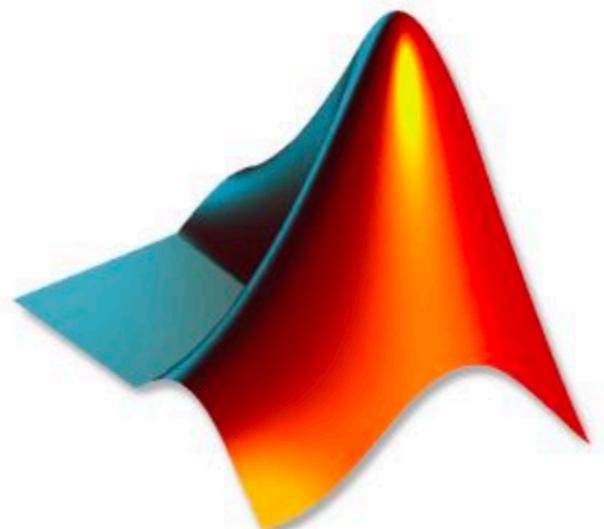
Ipython (rojo) v/s Jupyter Notebook (Abril 2017)



Tendencias

Baja de uso

Matlab



MATLAB
The Language of Technical Computing

Tendencias

Python



R



Editores, Medioambientes e IDEs

- Editores
 - Sublime-text
 - Atom
- IDES
 - PyCharm
 - Spyder
 - Visual Code Studio
- Ipython Shell
- Jupyter



```
athletelist.py
```

```
1  class AthleteList(list):
2      def __init__(self, a_name, a_dob=None, a_times=[]):
3          list.__init__([])
4          self.name = a_name
5          self.dob = a_dob
6          self.extend(a_times)
7
8      @staticmethod
9      def sanitize(time_string):
10         if '-' in time_string:
11             splitter = '-'
12         elif ':' in time_string:
13             splitter = ':'
14         else:
15             return(time_string)
16         (mins, secs) = time_string.split(splitter)
17         return(mins + '.' + secs)
18
19     @property
20     def to_dict(self):
21         data = {'Name': self.name, 'DOB': self.dob, 'Top3': self.top3}
22         return(data)
23
24     @property
25     def clean_data(self):
26         return(sorted(set([self.sanitize(t) for t in self])))
```

IDEs: Entorno de desarrollo integrado

Medioambientes

Deprecated

Virtual env

Easy_install

Setup Tools

Otros lenguajes

Rvm <https://rvm.io>



Jenv <http://www.jenv.be>



Python

 **ANACONDA®**

<https://www.continuum.io>

 **pip**

<https://pypi.python.org/pypi/pip>

Miniconda

	 Windows	 Mac OS X	 Linux
Python 3.6	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)
Python 2.7	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)

Características de Jupyter

- Es una forma de combinar texto y fórmulas en conjunto con código (ejecutable) en un documento que es accesible desde un navegador.
- Los cuadernos son almacenados en un archivo de texto JSON con la extensión .ipynb
- Jupyter puede ejecutar sobre 40 lenguajes diferentes, originalmente fue concebido para Julia, Python y R (Ju-Pyt-eR).

Características de Jupyter



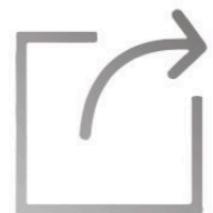
Language of choice

The Notebook has support for over 40 programming languages, including those popular in Data Science such as Python, R, Julia and Scala.



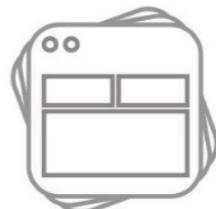
Big data integration

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, dplyr, etc.



Share notebooks

Notebooks can be shared with others using email, Dropbox, GitHub and the [Jupyter Notebook Viewer](#).



Interactive widgets

Code can produce rich output such as images, videos, LaTeX, and JavaScript. Interactive widgets can be used to manipulate and visualize data in realtime.

Ventajas de Jupyter

Proveer tu código en un cuaderno favorece la reproducibilidad de los resultados.

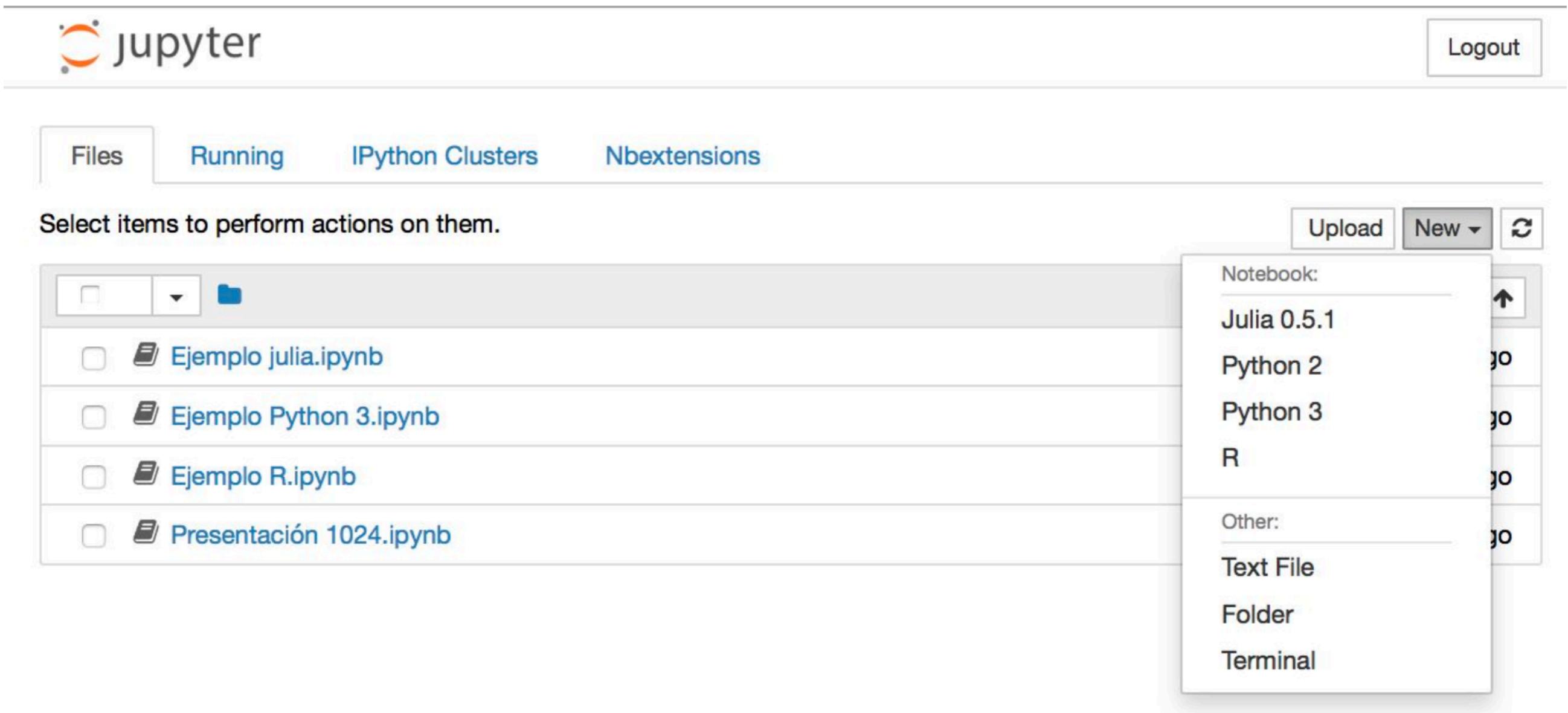
Para responder todas las preguntas no es suficiente sólo el código.

Discute tus resultados obtenidos.

Comparte tus cuadernos en varios formatos.

Cuaderno en un navegador

<http://localhost:8888/tree#>



The screenshot shows a Jupyter Notebook interface within a web browser. At the top, there's a header with the Jupyter logo on the left and a "Logout" button on the right. Below the header is a navigation bar with four tabs: "Files" (selected), "Running", "IPython Clusters", and "Nbextensions". A message "Select items to perform actions on them." is displayed above the file list. The main area contains a list of files:

- Ejemplo julia.ipynb
- Ejemplo Python 3.ipynb
- Ejemplo R.ipynb
- Presentación 1024.ipynb

To the right of the file list is a sidebar with several options:

- Upload
- New ▾
- ↻
- Notebook:
 - Julia 0.5.1
 - Python 2
 - Python 3
 - R
- Other:
 - Text File
 - Folder
 - Terminal



Cuaderno de Jupyter

jupyter Presentación 1024 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Logout Python 3

Code

Ejemplo de un Cuaderno

* Alexis Salas

In [1]: lista = [2, 3, 5, 6]

In []: lista.
lista.append
lista.clear
lista.copy
lista.count
lista.extend
lista.index
lista.insert
lista.pop
lista.remove
lista.reverse

✓ Code
Markdown
Raw NBConvert
Heading
-

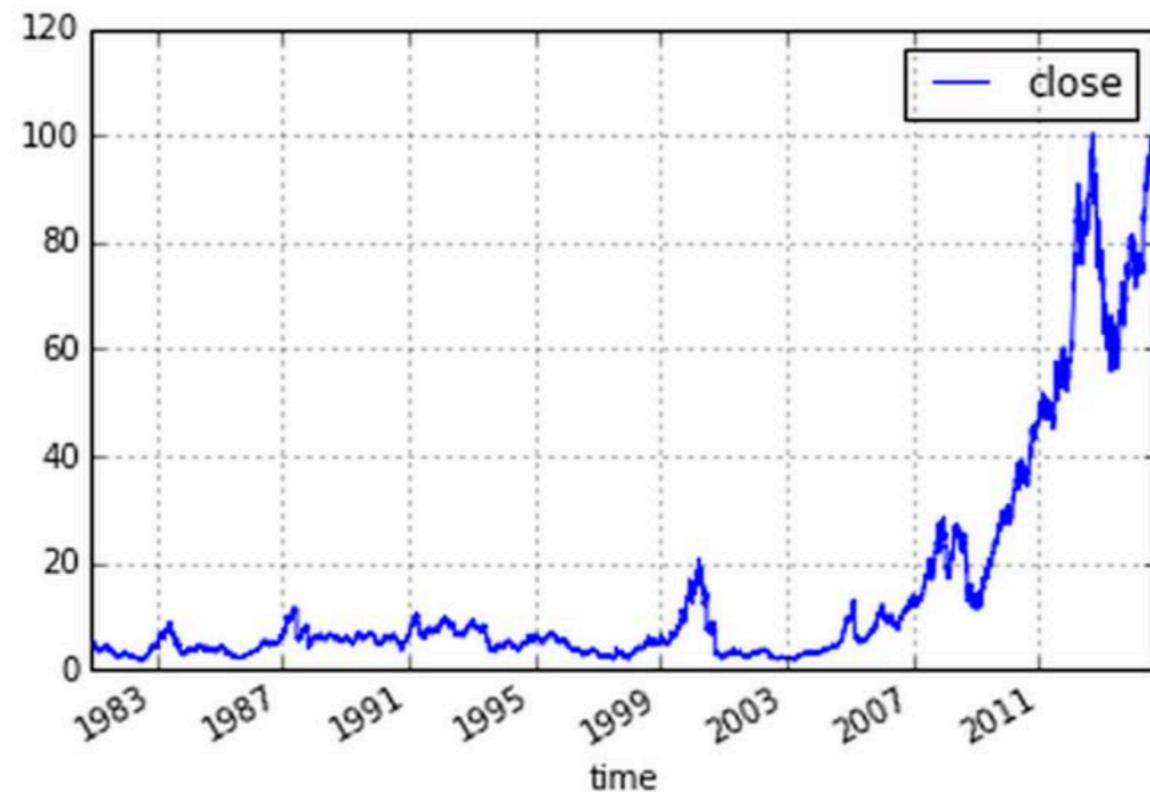
File>>Export>>

Notebook (.ipynb)
Python (.py)
HTML (.html)
Markdown (.md)
reST (.rst)
LaTeX (.tex)
PDF via LaTeX (.pdf)

Matplotlib <http://matplotlib.org>

```
In [8]: df = td.read_td_query(''  
    select time, close from nasdaq where symbol='AAPL'  
    ''', engine, index_col='time', parse_dates={'time': 's'})  
  
df.plot()
```

```
Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x107557438>
```



Seaborn <http://seaborn.pydata.org>

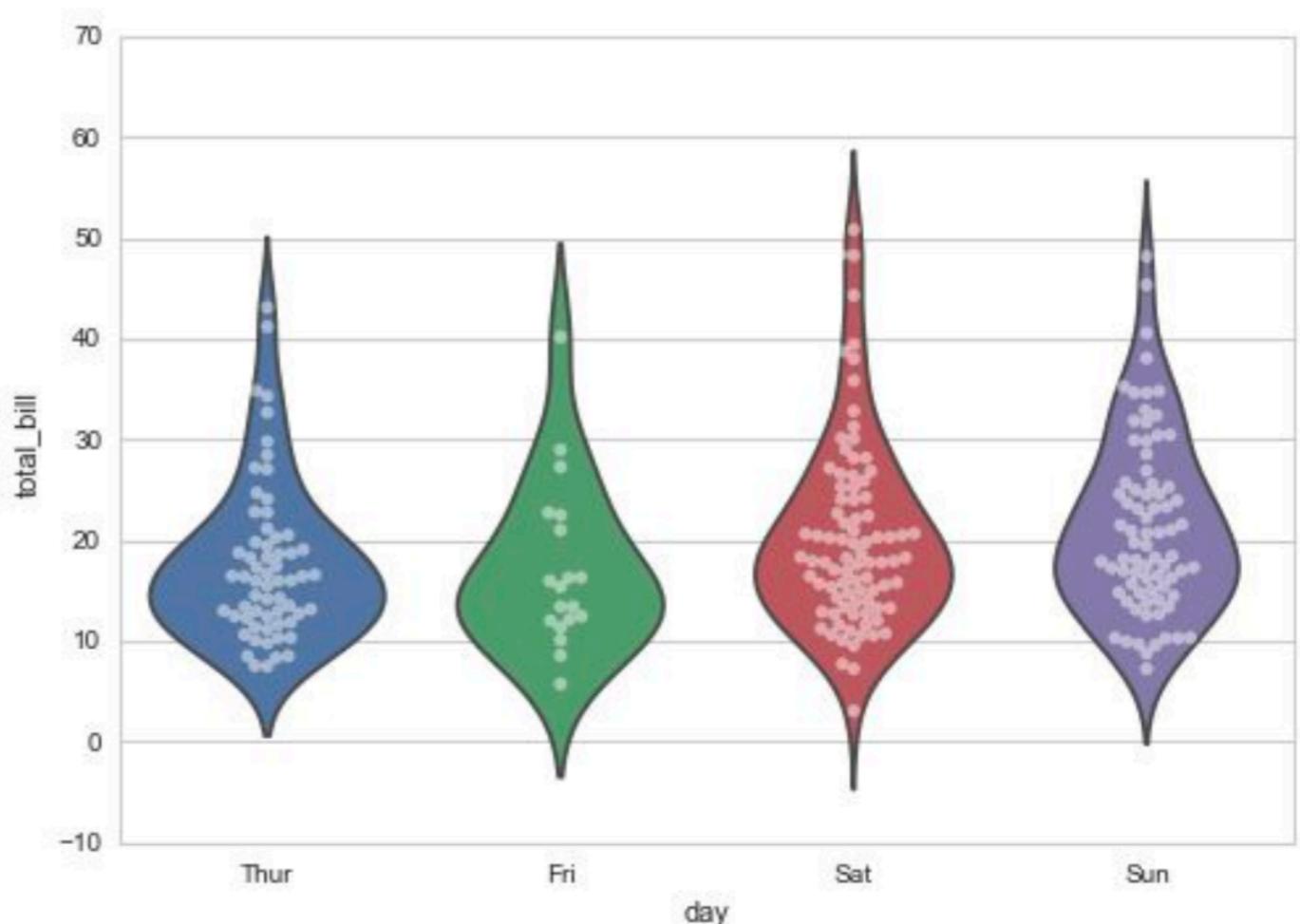
```
%matplotlib inline
```

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
import seaborn as sns
sns.set(style="whitegrid", color_codes=True)
```

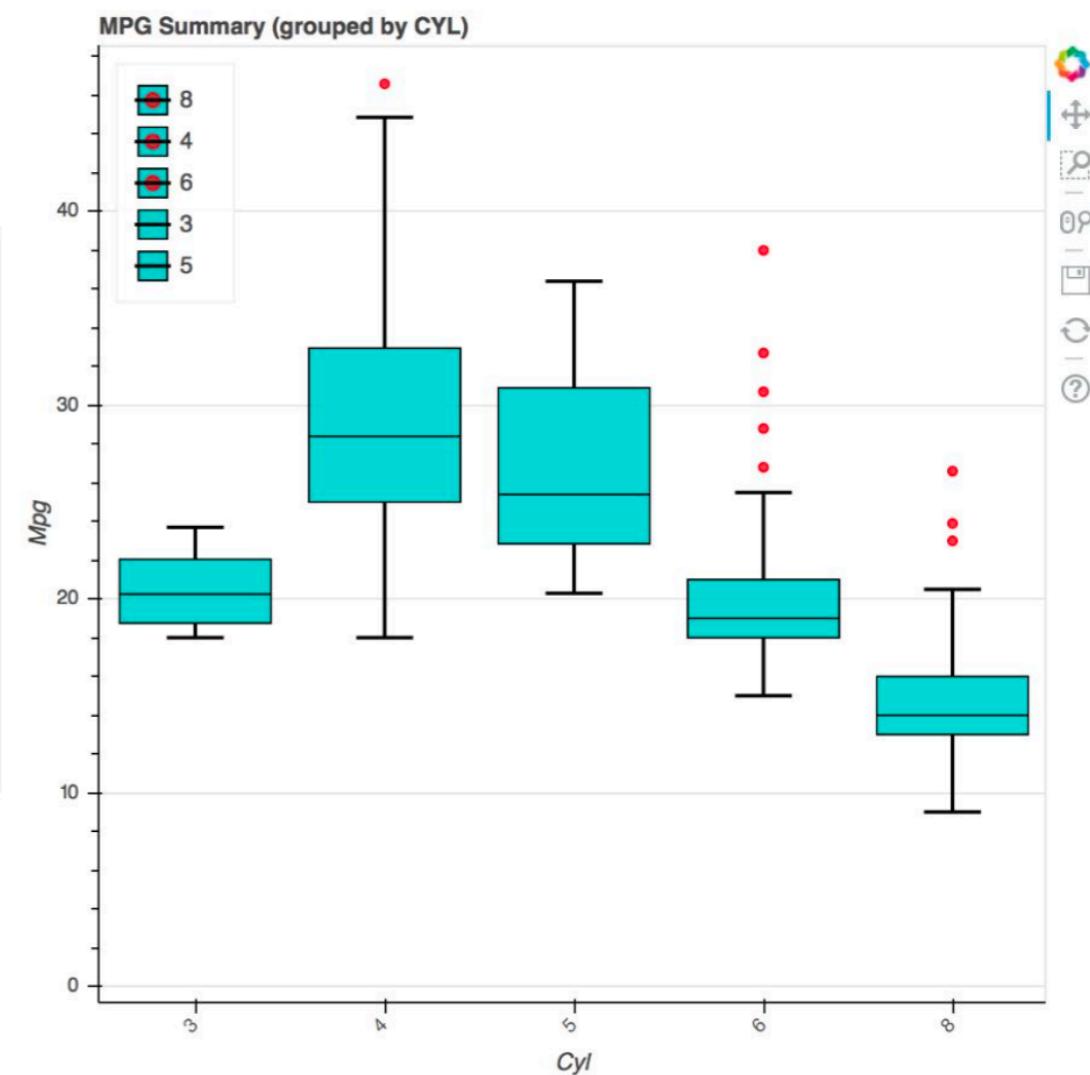
```
np.random.seed(sum(map(ord, "categorical")))
```

```
titanic = sns.load_dataset("titanic")
tips = sns.load_dataset("tips")
iris = sns.load_dataset("iris")
```

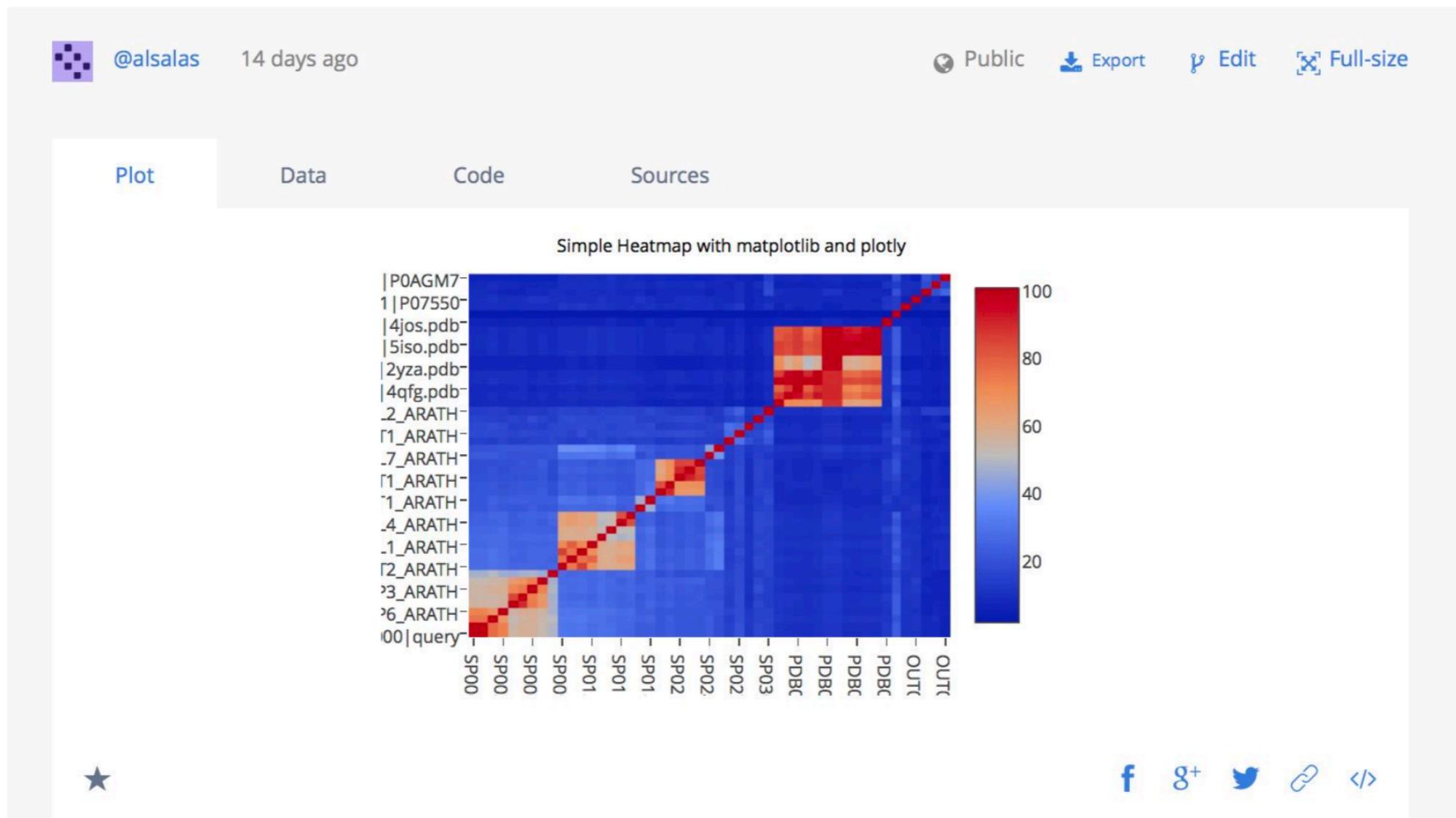


Bokeh <http://bokeh.pydata.org>

```
from bokeh.charts import BoxPlot, output_file , show  
from bokeh.sampledata.autompg import autompg as df  
  
p = BoxPlot(df, values='mpg', label='cyl', color='#00cccc',  
             title="MPG Summary (grouped by CYL)")  
  
output_file ("boxplot.html")  
  
show(p)
```



Plot.ly <https://plot.ly>



Jupyter Widgets



Notebook Widgets

<http://jupyter.org/widgets>

Instalación:

```
pip install ipywidgets  
jupyter nbextension enable --py  
--sys-prefix widgetsnbextension
```

```
pip install ipyleaflet
```

```
jupyter nbextension enable --py --sys-prefix ipyleaflet
```

```
pip install bqplot
```

```
jupyter nbextension enable --py --sys-prefix bqplot
```

```
pip install pythreejs
```

```
jupyter nbextension enable --py --sys-prefix pythreejs
```

```
pip install ipyvolume
```

```
jupyter nbextension enable --py --sys-prefix ipyvolume
```

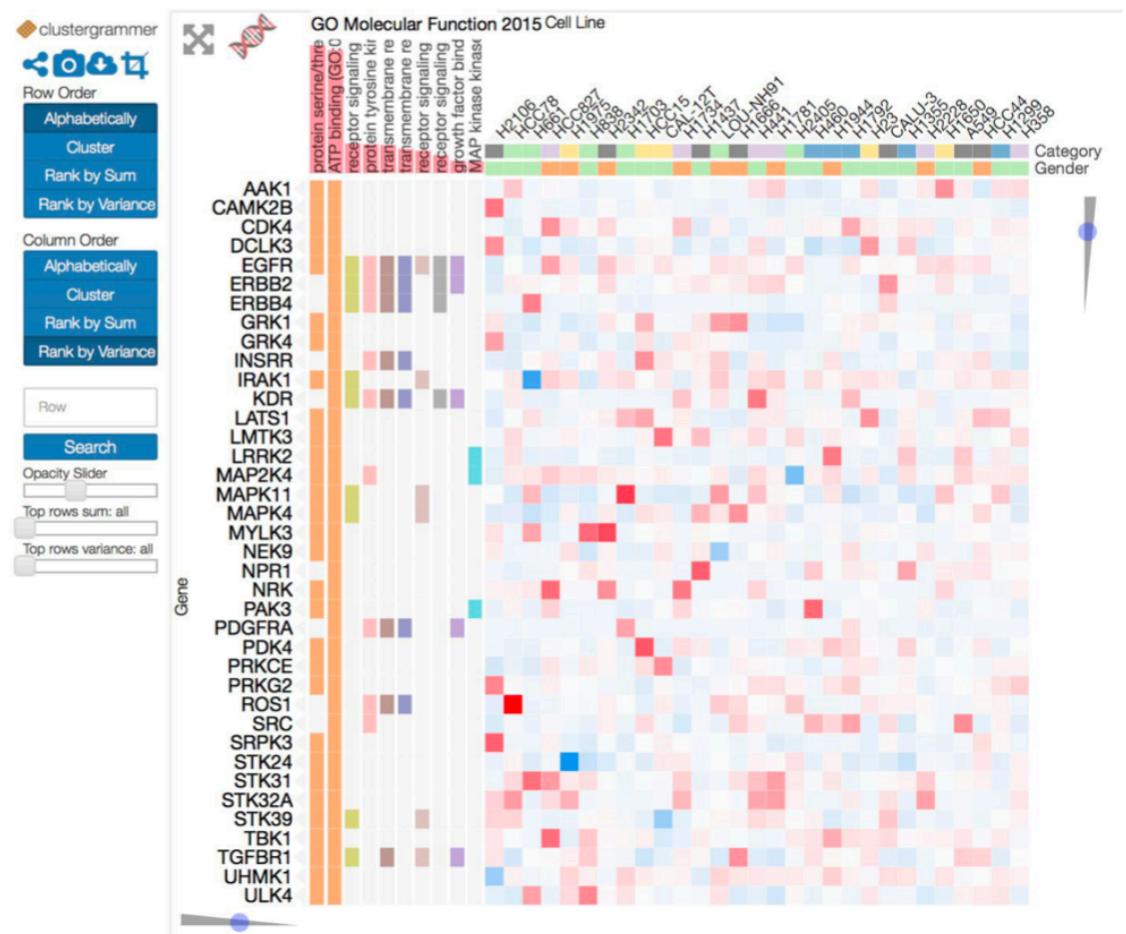
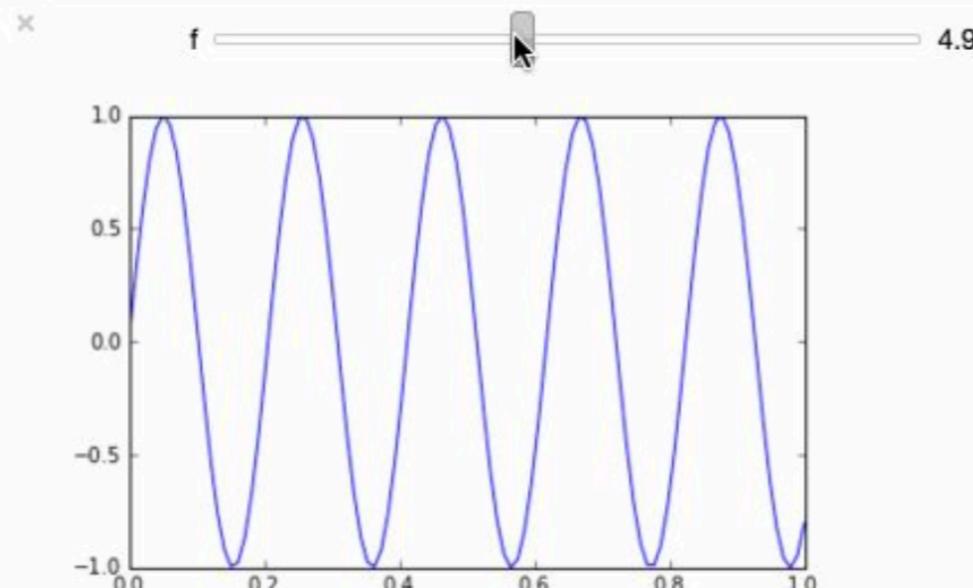


Jupyter Widgets

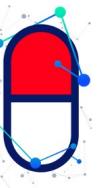
```
In [22]: from IPython.html.widgets import
          widgets
          t = arange(0.0, 1.0, 0.01)

          def pltsin(f):
              plt.plot(x,sin(2*pi*t*f))
              plt.show()

          interact(pltsin, f=(1,10,0.1))
```



http://nbviewer.jupyter.org/github/MaayanLab/clustergrammer-widget/blob/master/Running_clustergrammer_widget.ipynb



JupyterLab: La nueva generación de Jupyter notebook

The screenshot displays the JupyterLab interface. On the left, a file browser shows a directory structure for 'jupyterlab'. In the center, a code editor window titled 'Untitled.ipynb' contains Python code for generating a polar plot, which is displayed below as a circular chart with radial bars. To the right, a 'Launcher' panel shows a list of available kernels, including 'Python 3 (1)' which is currently selected. Below the launcher, a terminal window titled 'Terminal 1' shows system monitoring output, including CPU usage bars and a table of processes.

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

N = 20
theta = np.linspace(0.0, 2 * np.pi, N, endpoint=False)
radii = 10 * np.random.rand(N)
width = np.pi / 4 * np.random.rand(N)
ax = plt.subplot(111, projection='polar')
bars = ax.bar(theta, radii, width=width, bottom=0.0)
for r, bar in zip(radii, bars):
    bar.set_facecolor(plt.cm.jet(r / 10.))
    bar.set_alpha(0.5)
```

```
In [1]: %matplotlib inline
from numpy.random import beta
import matplotlib.pyplot as plt
plt.style.use('bmh')

def plot_beta_hist(a, b):
    plt.hist(beta(a, b, size=10000), histtype="stepfilled",
            bins=25, alpha=0.8, normed=True)
    return

plot_beta_hist(10, 10)
plot_beta_hist(4, 12)
plot_beta_hist(50, 12)
plot_beta_hist(6, 55)
```

```
In [ ]:
```

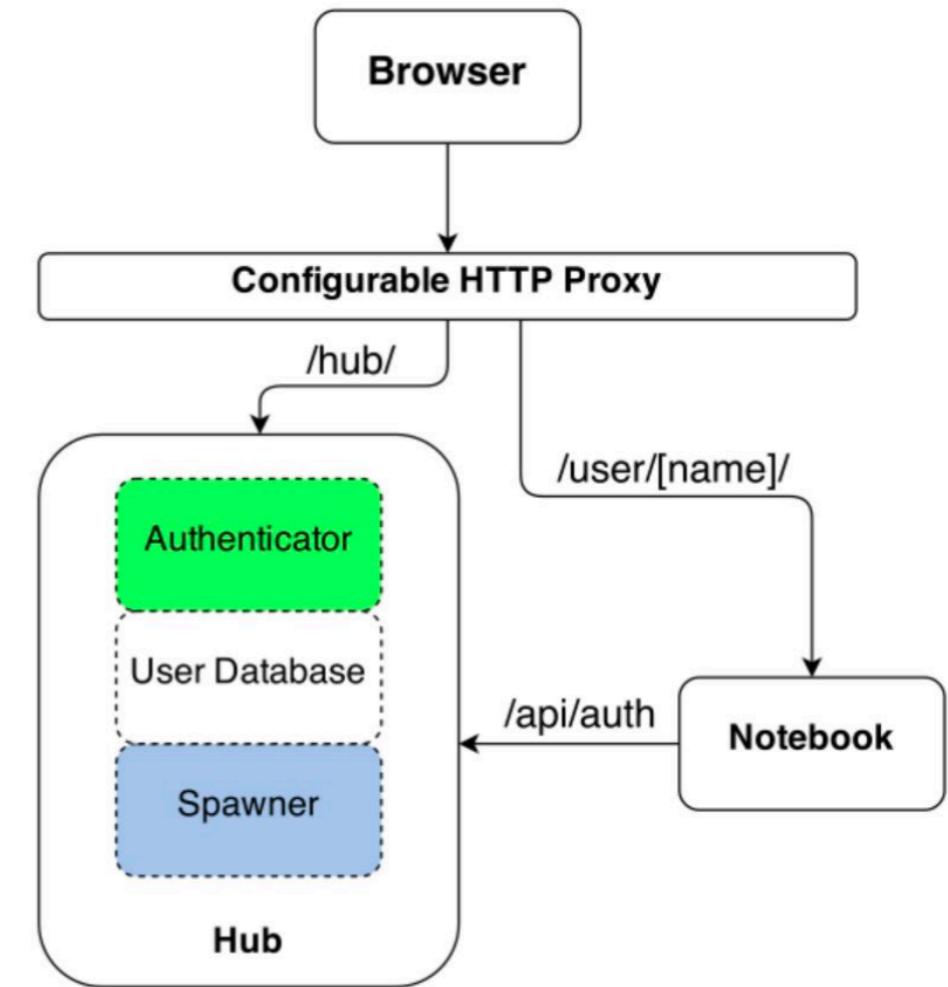
```
1  [ 18.1%  Tasks: 305 total, 1 running
2  [ 5.0%  Load average: 2.29 2.07 2.09
3  [ 15.6%  Uptime: 4 days, 21:59:11
4  [ 5.0%
Mem[ 5987/8192MB
Swp[ 2487/3072MB]
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
82374	fpererez	31	0	2389M	2048	0	R	0.0	0.0	0:00.00	htop
1	root	0	0	0	0	0	0.0	0.0	0.0	0:00.00	(launchd)
46	root	0	0	0	0	0	0.0	0.0	0.0	0:00.00	(syslogd)
47	root	0	0	0	0	0	0.0	0.0	0.0	0:00.00	(UserEventAgent)

<https://github.com/jupyterlab/jupyterlab>



A multi-user version of the notebook designed for companies,
classrooms and research labs



```
docker run -d --name jupyterhub jupyterhub/jupyterhub jupyterhub
```

<https://github.com/jupyterhub/jupyterhub>

jupyterhub

- El ingreso de usuarios es administrado por el Hub
- Los usuarios se autentifican vía un formulario
- Hub soporta proxies
- Redirige al usuario a /user/[name]
- Configuración centralizada
- Seguimiento de cuadernos



Linux Shell Scripting tutorial

HomeWork

1. Realizar el tutorial de programación en Shell:

<https://iie.fing.edu.uy/~vagonbar/unixbas/shprog.htm>,

Realizar todos los scripts ejemplo que se enuncian allí.

2. Revisar páginas de interés.

- 1: <http://aprendeonline.udea.edu.co/lms/ova/course/view.php?id=6>
2. <http://www.freeos.com/guides/lsst/ch02sec01.html>
3. <http://www.freeos.com/guides/lsst/>