

# iPython

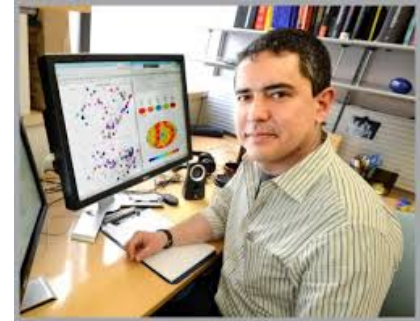
Tool for Data Scientist ;)

# Why Python

- Great libraries for scientific work
  - Numpy
  - Scipy
  - Matplotlib
  - Sympy
  - Pandas
- Popular programming language in the scientific world
- Deployment for production

# What is iPython

- Started by Fernando Perez
- Better interactive python shell
- For scientific usage
- Interactive data visualization
- Embeddable interpreter
- Tools for parallel computing



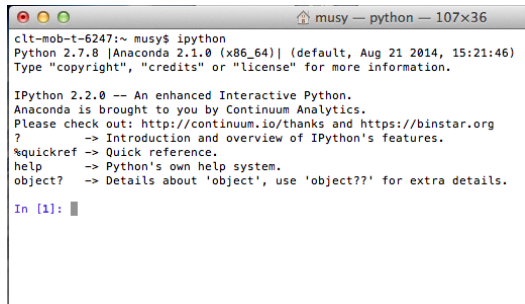
# Lifecycle of a scientific idea

1. Individual exploratory work
2. Collaborative development
3. Production work (HPC, Cloud, parallel)
4. Publication (with reproducible results)
5. Education
6. Goto 1

by Fernando Perez

# Clients and iPython Kernel

## Terminal-Console

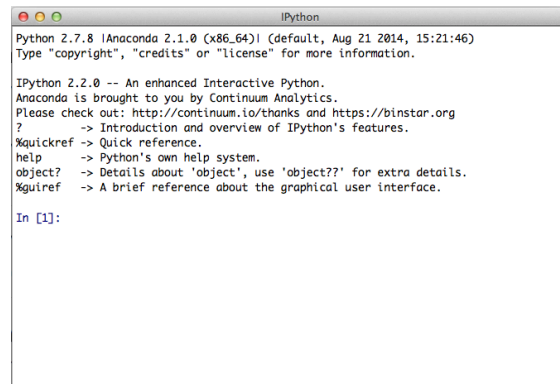


```
cli-mob-t-6247:~ musy$ ipython
Python 2.7.8 [Anaconda 2.1.0 (x86_64)] (default, Aug 21 2014, 15:21:46)
Type "copyright", "credits" or "license" for more information.

IPython 2.2.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
?
-> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

## Qt-Console



```
iPython
Python 2.7.8 [Anaconda 2.1.0 (x86_64)] (default, Aug 21 2014, 15:21:46)
Type "copyright", "credits" or "license" for more information.

IPython 2.2.0 -- An enhanced Interactive Python.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://binstar.org
?
-> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
%quickref -> A brief reference about the graphical user interface.

In [1]:
```

## Web-Client

### IP[y]: Notebook

Notebooks **Running** Clusters

To import a notebook, drag the file onto the listing below or [click here](#).



anaconda

Applications

Applications (Parallels)

code

Desktop

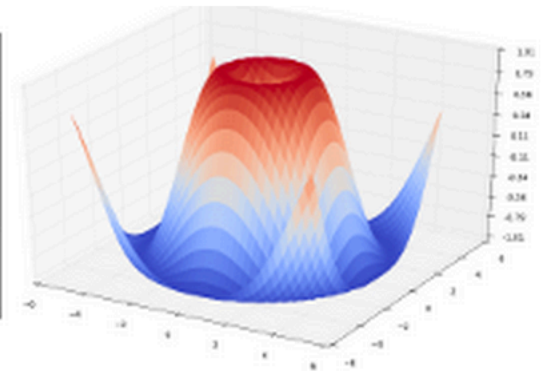
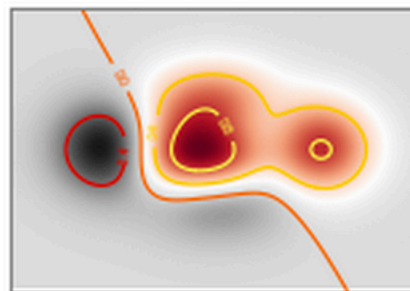
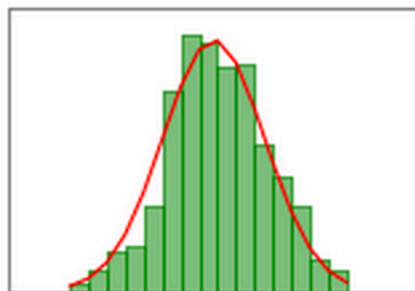
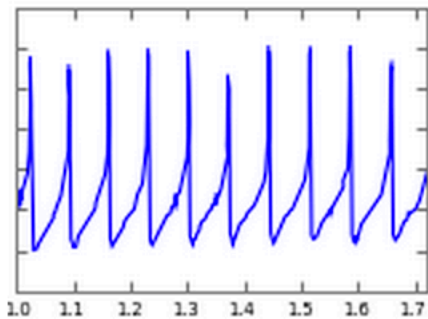
iPython Kernel

# Plot in python

- Lets try: 

```
%pylab  
x = randn(10000)  
hist(x, 100)
```

- It uses matplotlib



# %magic

- %timeit %%timeit
- %%bash
- %%file
- %load to inspect external code

# iPython Notebook

- Web based interface
- The way to go nowadays
- Uses the power of modern web technologies
- Mix code with description (markdown)
- Stores notebook in a JSON file .ipynb



# debug

## Getting started

start pdb from within a script:

```
import pdb;pdb.set_trace()
```

start pdb from the commandline:

```
python -m pdb <file.py>
```

## Basics

**h(elp)** print available commands

**h(elp) *command*** print help about *command*

**q(uit)** quit debugger

## Examine

**p(rint) *expr*** print the value of *expr*

**pp *expr*** pretty-print the value of *expr*

**w(here)** print current position (including stack trace)

**l(ist)** list 11 lines of code around the current line

**l(ist) *first, last*** list from *first* to *last* line number

**a(args)** print the args of the current function

## Movement

**<ENTER>** repeat the last command

**n(ext)** execute the current statement (step over)

**s(tep)** execute and step into function

**r(eturn)** continue execution until the current function returns

**c(ontinue)** continue execution until a breakpoint is encountered

**u(p)** move one level up in the stack trace

**d(own)** move one level down in the stack trace

## Breakpoints

**b(reak)** show all breakpoints

**b(reak) *lineno*** set a breakpoint at *lineno*

**b(reak) *func*** set a breakpoint at the first line of a *func*

## Manipulation

**!*stmt*** treat *stmt* as a Python statement instead of a pdb command

# parallel

- <http://star.mit.edu/cluster/index.html>
- <https://notebookcloud.appspot.com/> (blank)