Librerías científicas en Python:

MatplotLib básico

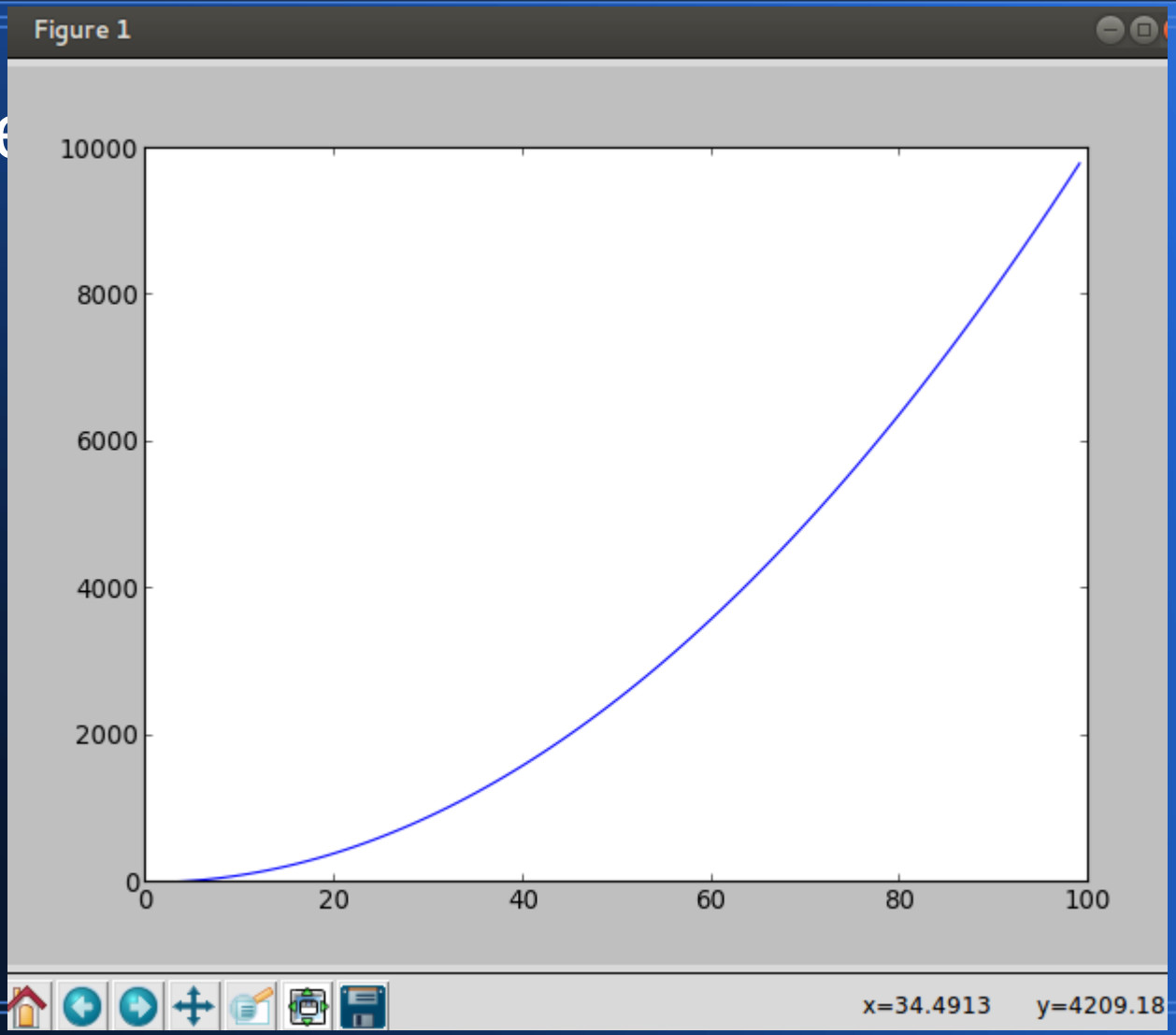
Ejemplos de hoy: git clone https://github.com/jeudy/sci-libs.git

Características

- from matplotlib import pyplot
- Librería de python para producir gráficos 2D con calidad de publicación.
- pyplot: interface de comandos similar a Matlab.
- Pylab = pyplot + numpy
- ipython --pylab
- Instalación:
  - sudo apt-get install python-matplotlib

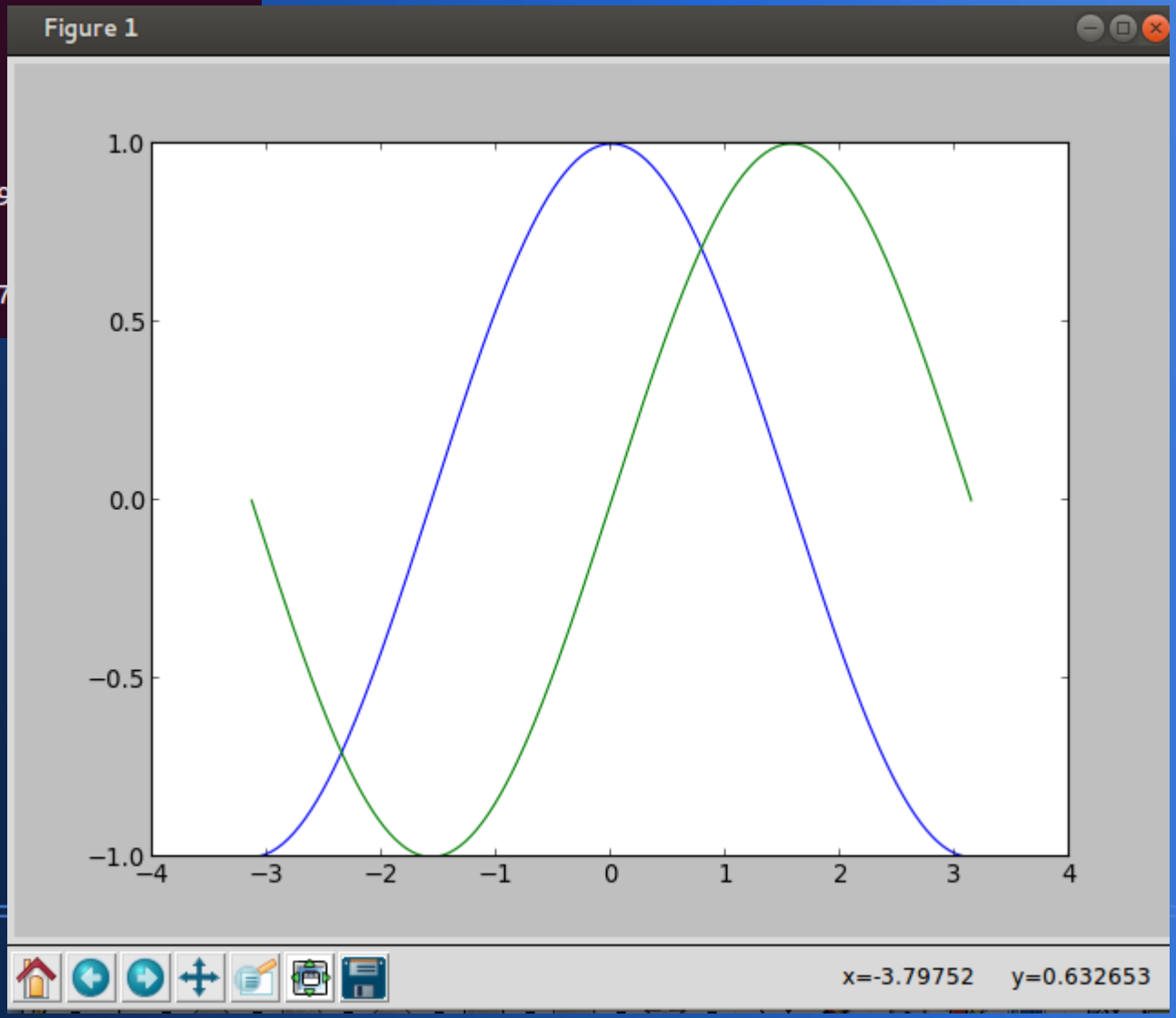
Ejemplo

- Plot de



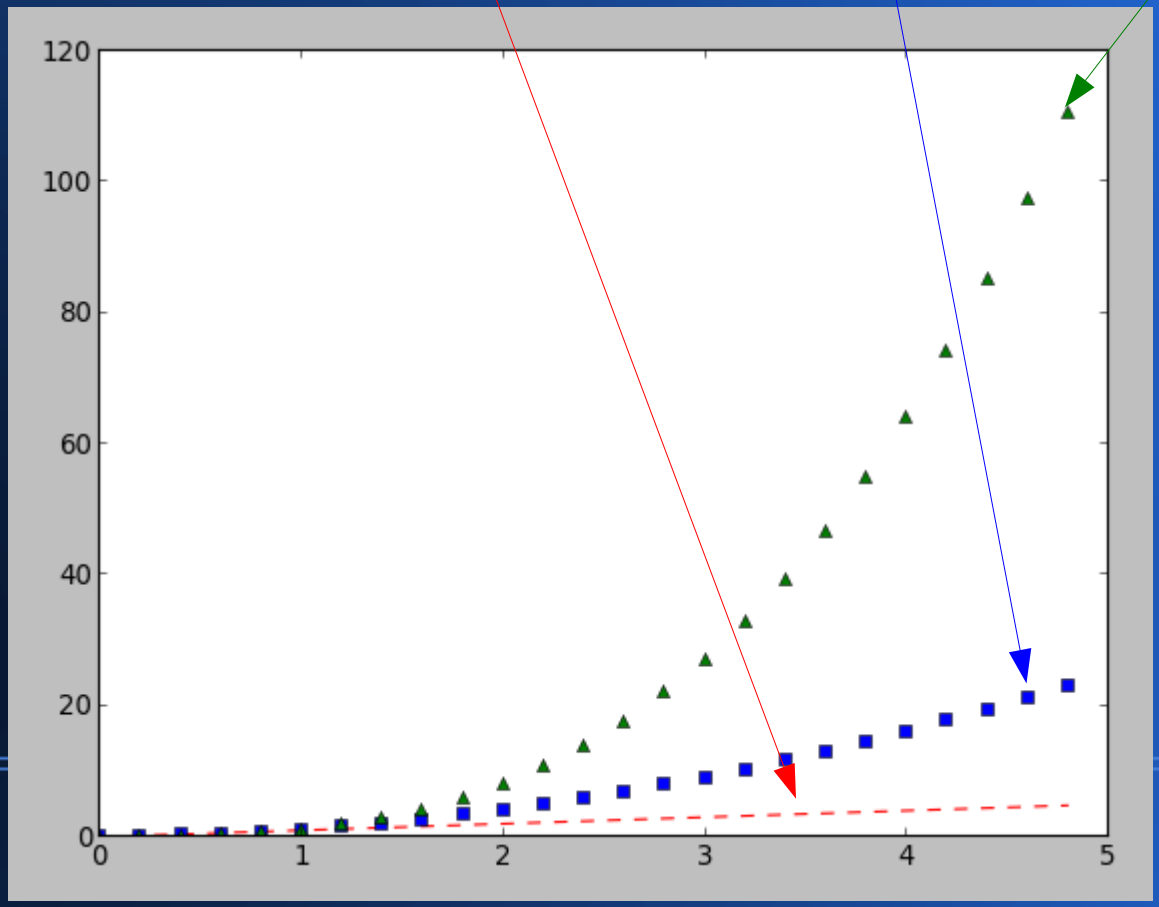
Otro ejemplo: seno y coseno

```
In [77]: X = linspace(-np.pi, np.pi, 256, endpoint=True)

In [78]: C = cos(X)

In [79]: S = sin(X)

In [80]: plot(X, C)
Out[80]: [<matplotlib.lines.Line2D at 0x609

In [81]: plot(X, S)
Out[81]: [<matplotlib.lines.Line2D at 0x607
```



Otro ejemplo: 3 en 1

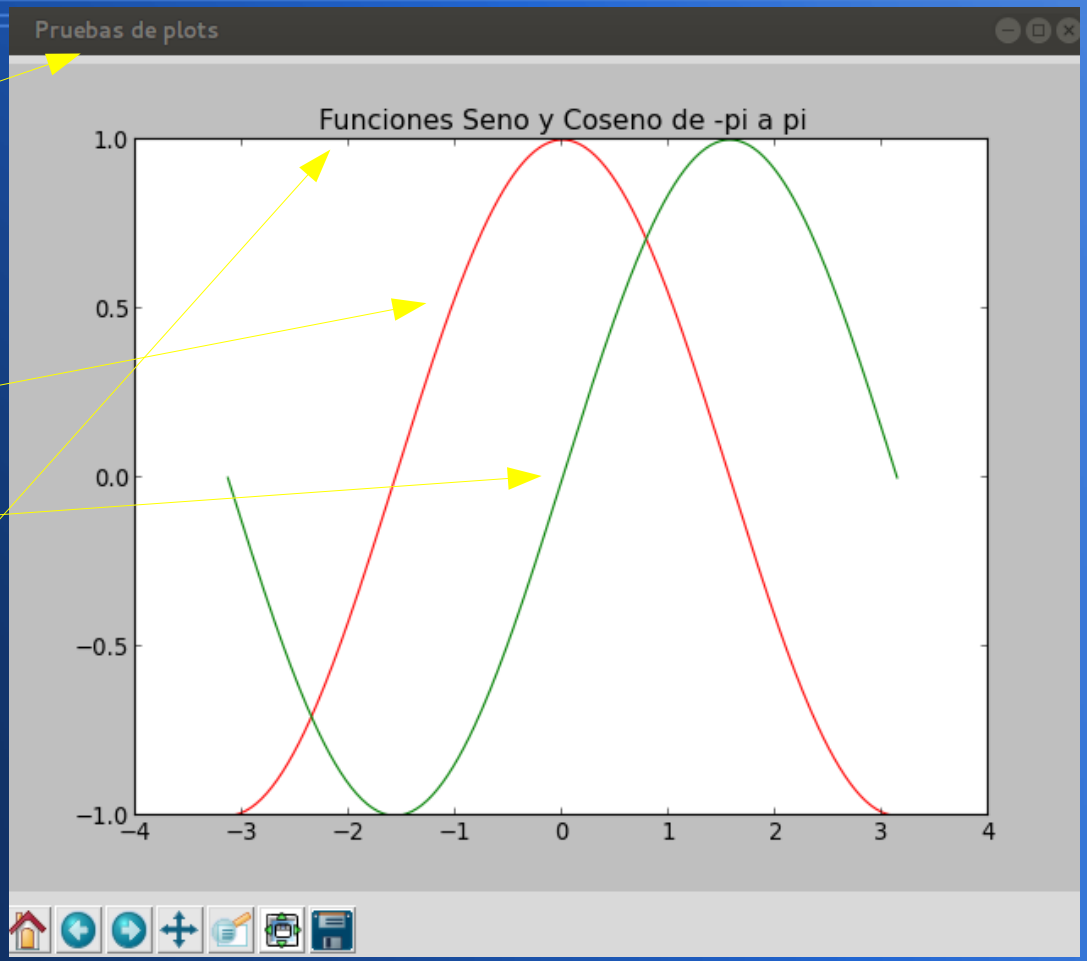
```
In [64]: x = arange(0., 5., 0.2)

In [65]: plt.plot(x, x, 'r--', x, x**2, 'bs', x, x**3, 'g^')
 # red dashes, blue squares and green triangles
```



Ejemplo desde script

custom_plot.py

```
# Ejemplo basado en del libro PythonScientific, capitulo 4

import pylab as pl
import numpy as np

# Create a figure of size 8x6 points, 80 dots per inch
pl.figure("Pruebas de plots", figsize=(8, 6), dpi=80)

# Create a new subplot from a grid of 1x1
pl.subplot(1, 1, 1)
X = np.linspace(-np.pi, np.pi, 256, endpoint=True)
C, S = np.cos(X), np.sin(X)
# Plot cosine with a blue continuous line of width 1 (pixels)
pl.plot(X, C, color="red", linewidth=1.0, linestyle="-")
# Plot sine with a green continuous line of width 1 (pixels)
pl.plot(X, S, color="green", linewidth=1.0, linestyle="-")
# Set x limits
pl.xlim(-4.0, 4.0)
# Set x ticks
pl.xticks(np.linspace(-4, 4, 9, endpoint=True))
# Set y limits
pl.ylim(-1.0, 1.0)
# Set y ticks
pl.yticks(np.linspace(-1, 1, 5, endpoint=True))
# Save figure using 72 dots per inch
# Show result on screen
pl.title('Funciones Seno y Coseno de -pi a pi')
pl.show()
```



Figuras y Subplots

- Figure: una ventata entera que contiene elementos.
- Subplot: varios plots puestos en un grid. Se especifica el número de filas, columas y la posición del plot.

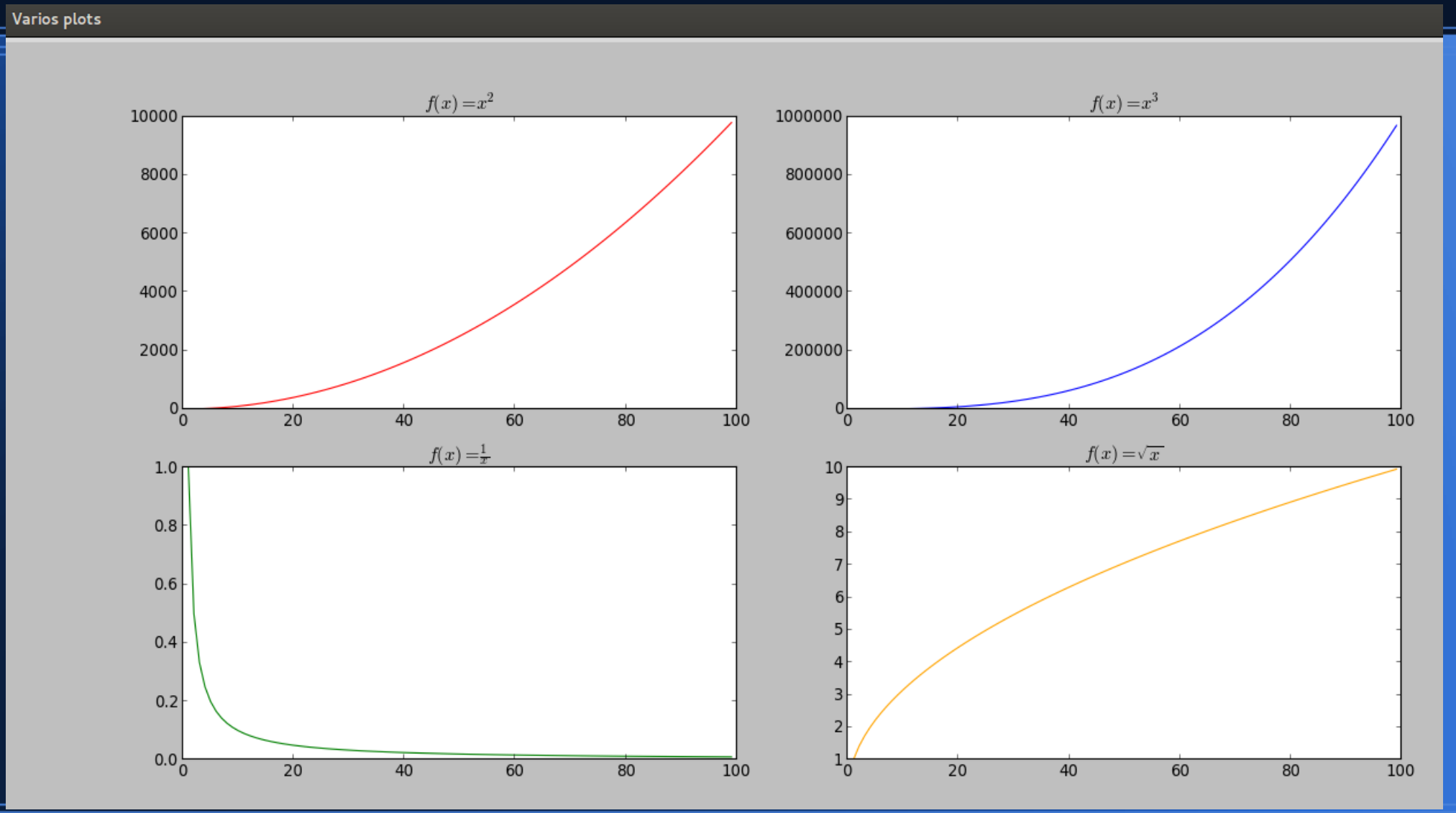
Ejemplo de subplots (1)

- 4 subplots en un grid 2x2
- Abrir varios_subplots.py de repositorio sci-libs

```python
# -*- coding: utf-8 -*-

import pylab as pl

# Definición de funciones auxiliares
def f1(x):
    return x**2

def f2(x):
    return x**3

def f3(x):
    return 1./x

def f4(x):
    return pl.sqrt(x)
```

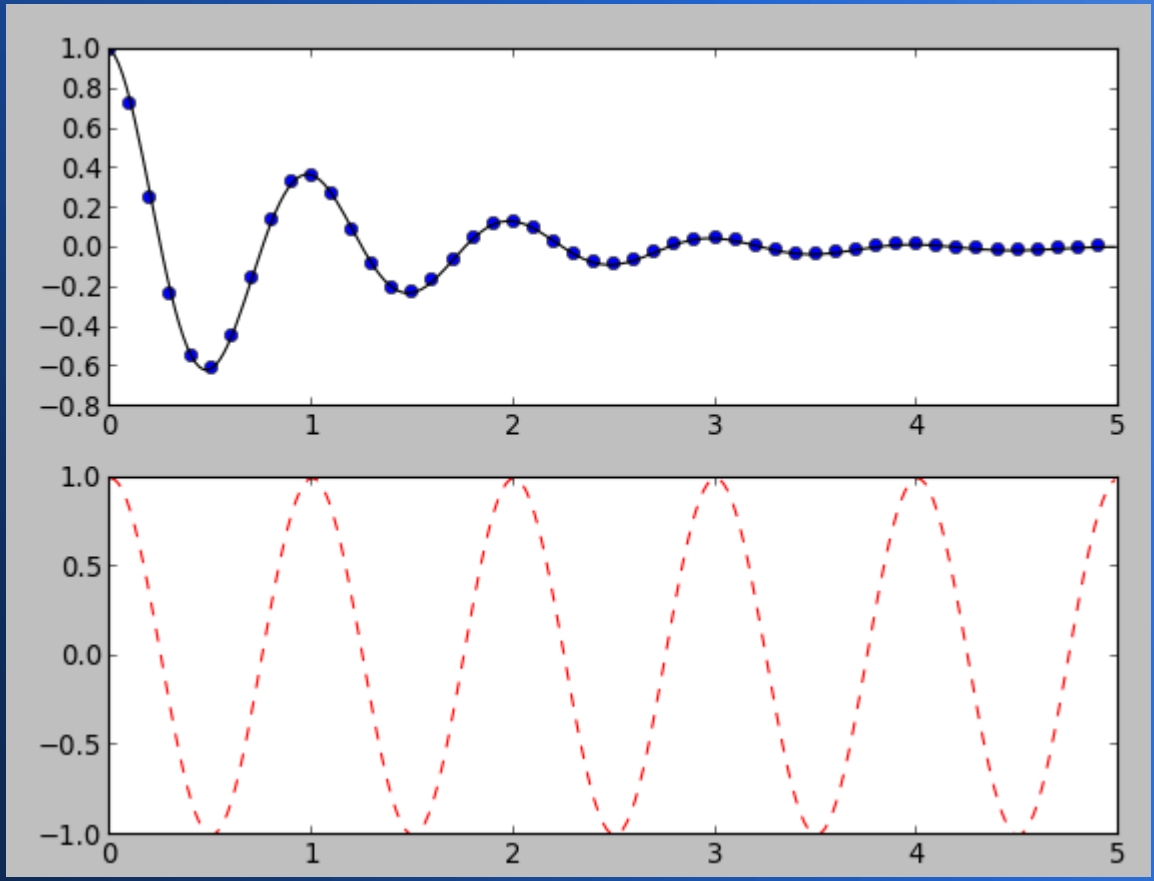
```python
X = pl.arange(1,100)
fig = pl.figure("Varios plots")
# Vamos a crear un grid de 2x2 con 4 subplots
sp1 = fig.add_subplot(221, title='$f(x) = x^2$')    # Ojo al titulo con expresión matemática
sp2 = fig.add_subplot(222, title='$f(x) = x^3$')
sp3 = fig.add_subplot(223, title= '$f(x) = \\frac{1}{x}$')
sp4 = fig.add_subplot(224, title = '$f(x) = \sqrt{x}$')

Y1 = f1(X)
Y2 = f2(X)
Y3 = f3(X)
Y4 = f4(X)

sp1.plot(X, Y1, color="red")
sp2.plot(X, Y2, color="blue")
sp3.plot(X, Y3, color="green")
sp4.plot(X, Y4, color="orange")

pl.show()
```

Ejemplo de subplots (1)



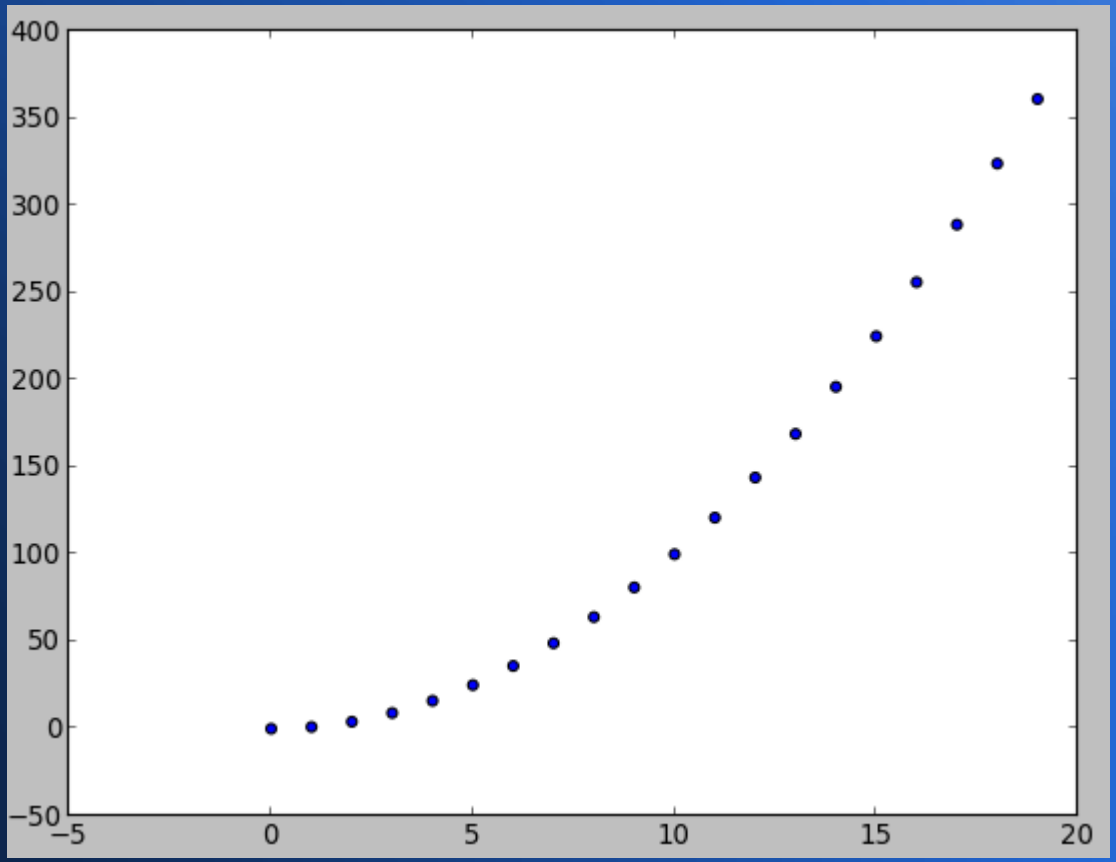
Ejemplo de subplots (2)

```
:def f(t):
:    return exp(-t) * cos(2*pi*t)
:
:t1 = arange(0.0, 5.0, 0.1)
:t2 = arange(0.0, 5.0, 0.02)
:
:subplot(211)
:plot(t1, f(t1), 'bo', t2, f(t2), 'k')
:
:subplot(212)
:plot(t2, cos(2*pi*t2), 'r--')
```



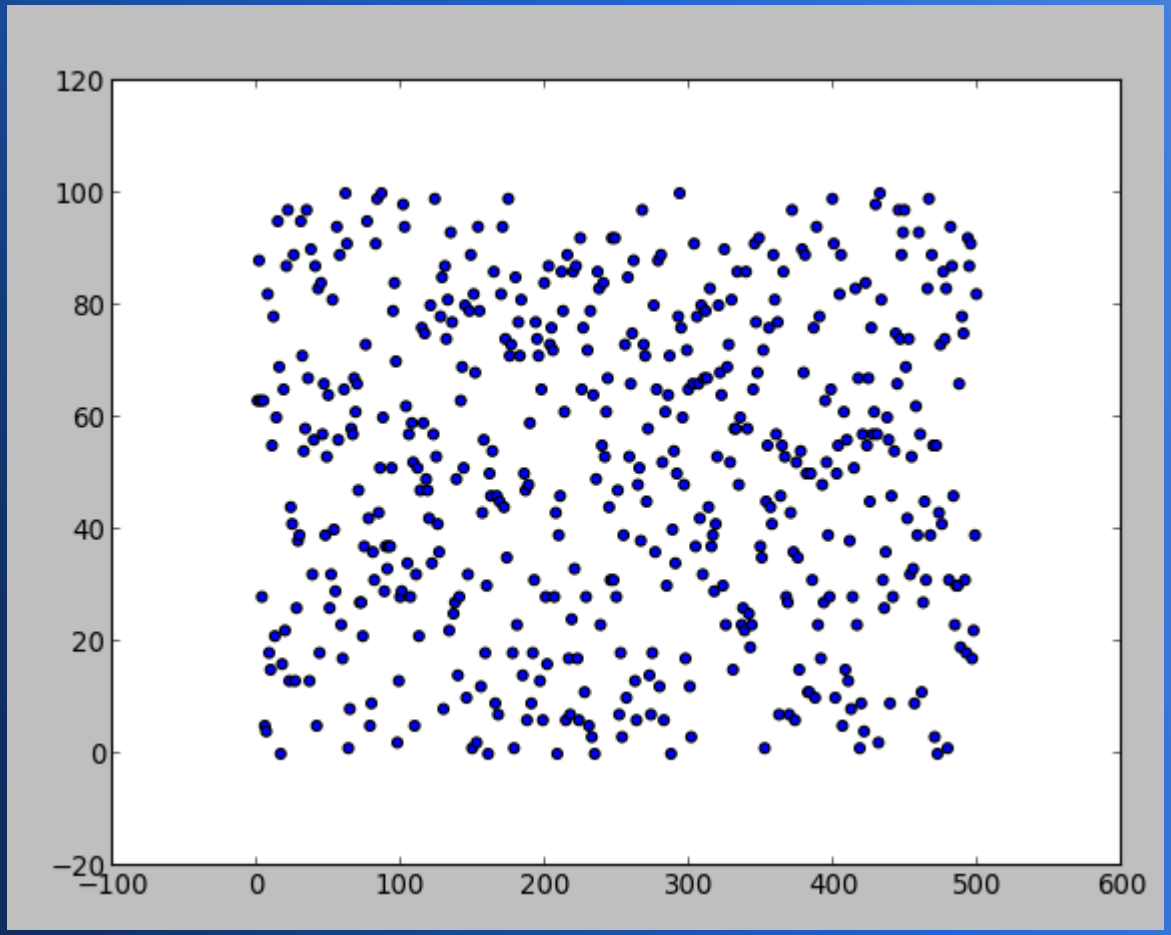
Ejemplo: plots de dispersión (1)

```
In [26]: def f(x):
   ....:     return x**2
   ....:

In [27]: X = arange(1,20)

In [28]: Y = f(X)

In [29]: scatter(X,Y)
```



Ejemplo: plots de dispersión (2)

```
In [56]: X = arange(0,500)

In [57]: Y = random.random_integers(0,100, 500)

In [58]: scatter(X,Y)
```

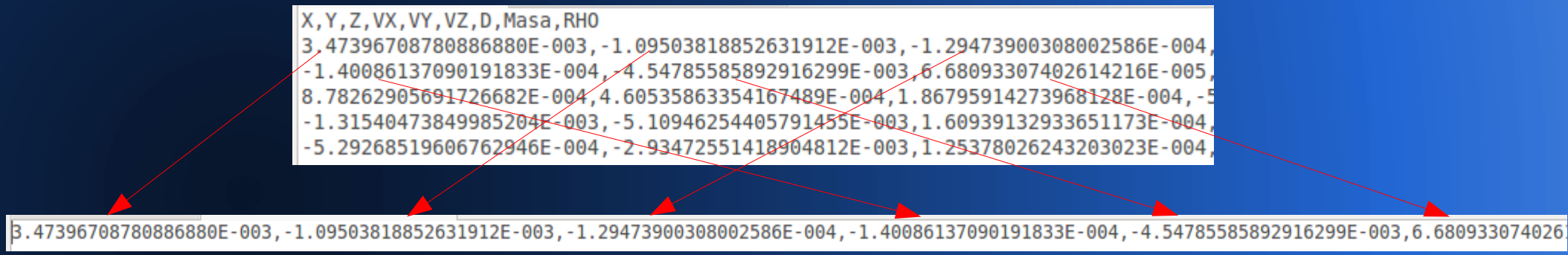


Ejercicio

- Clone el siguiente repositorio:
- https://github.com/matplotlib/matplotlib/tree/master/examples/pylab_examples
- Ejecute 10 ejemplos.

Proyecto en clase

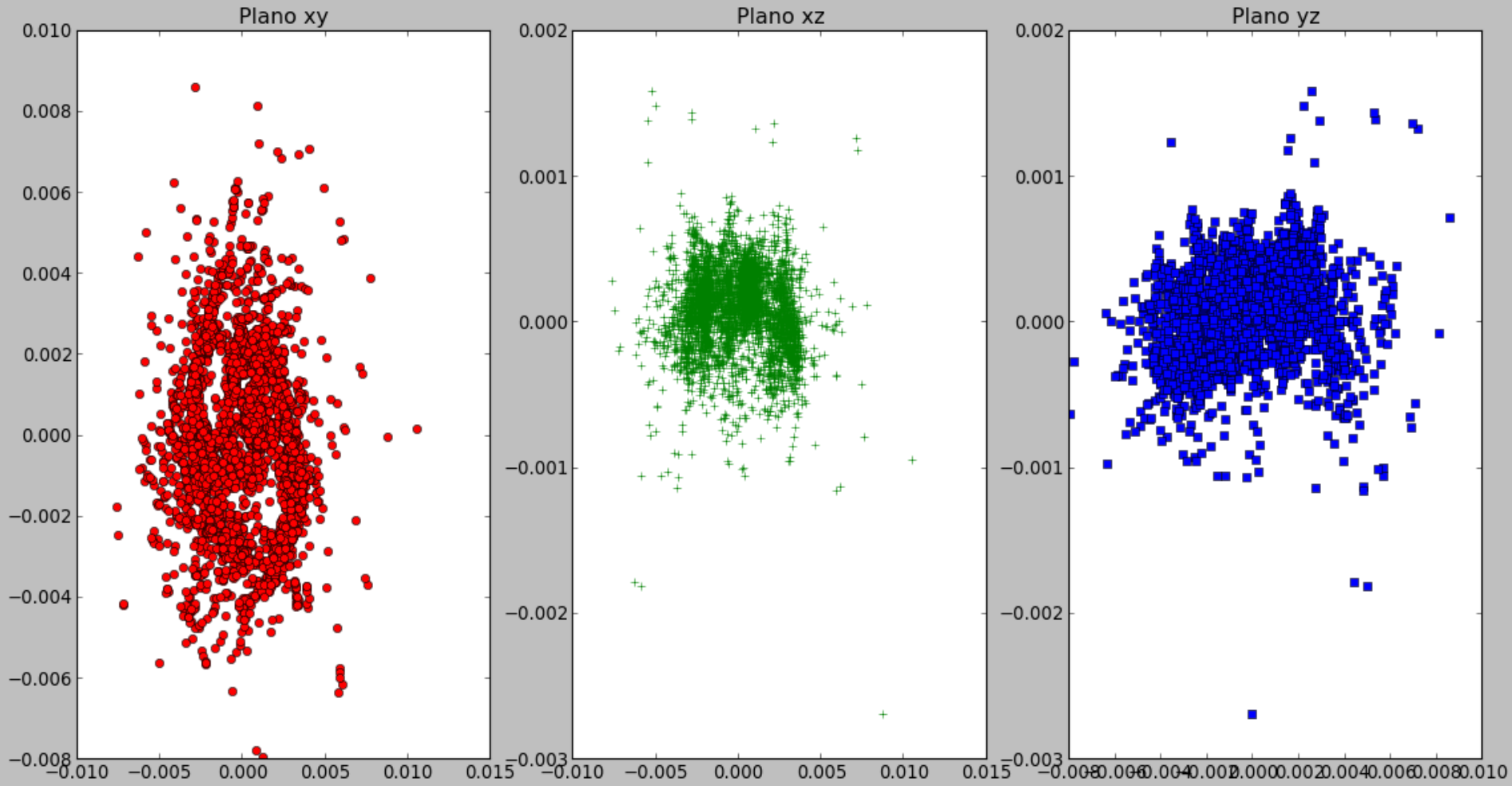
- Existe un archivo posiciones.csv en sci-libs/ejercicios/data con datos que incluyen las coordenadas x,y,z de 5000 partículas.
- Lea ese archivo con python y escriba otro posiciones_numpy.csv solo con las posiciones x,y,z, de esta forma:

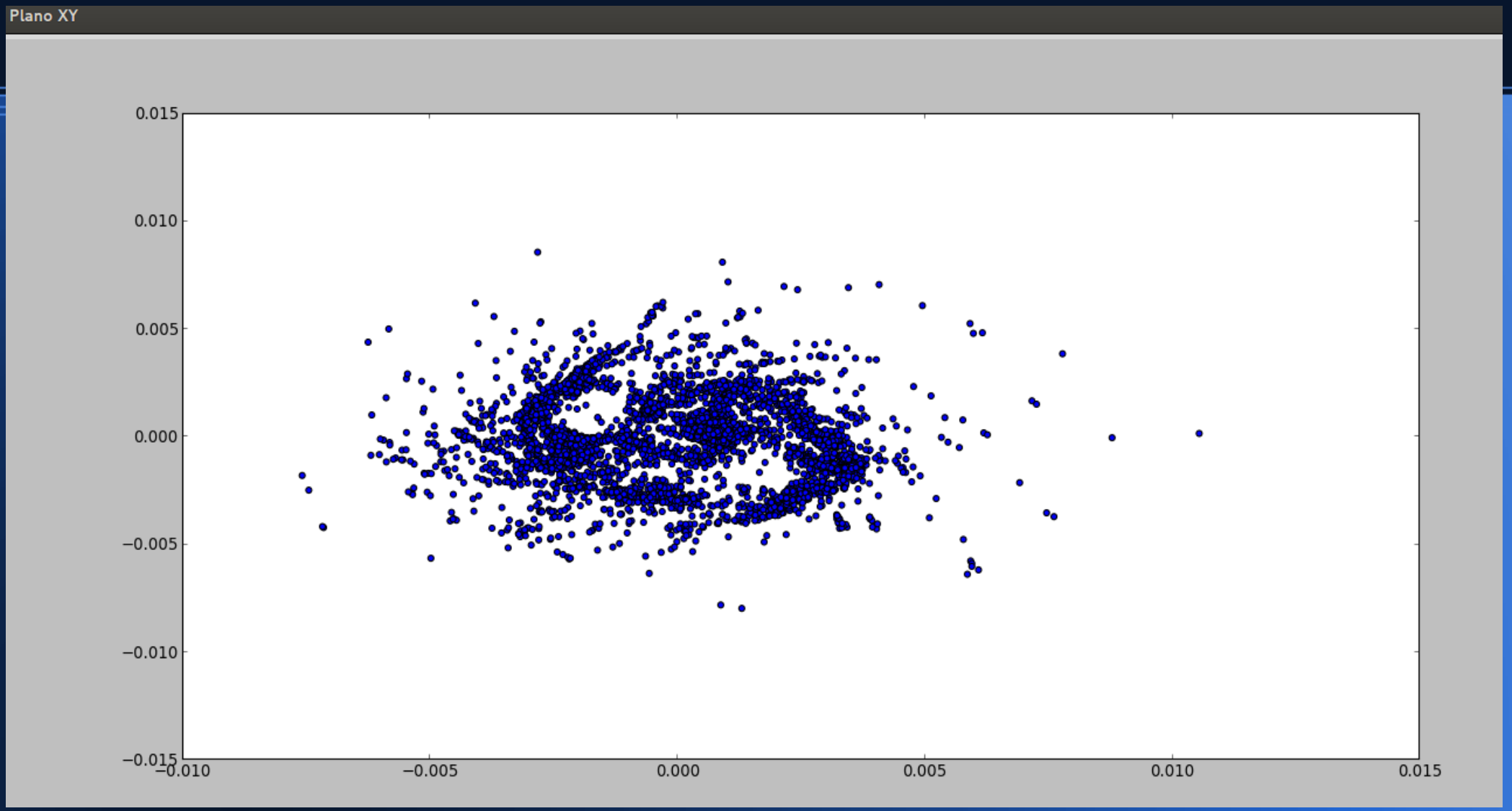


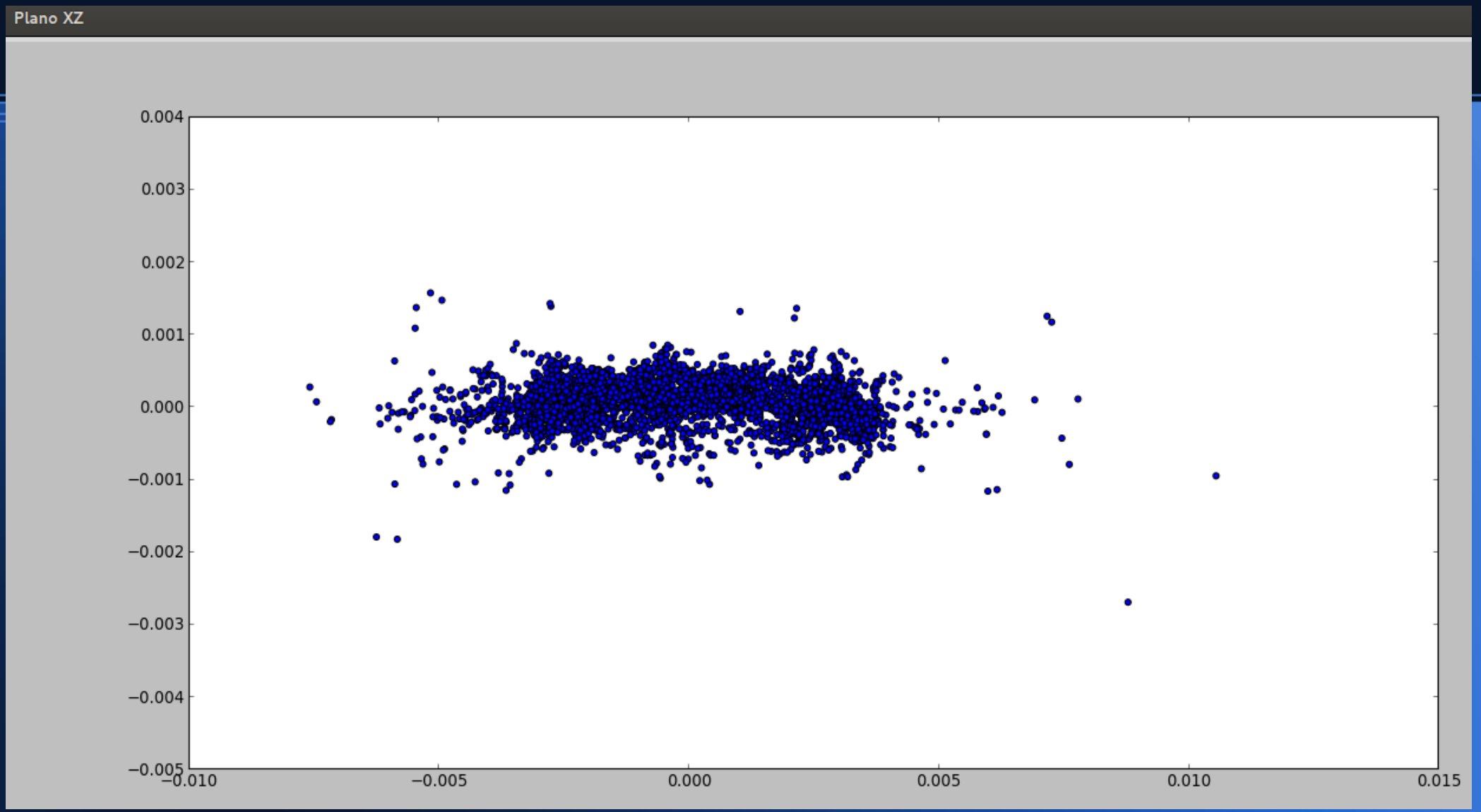
Proyecto en clase

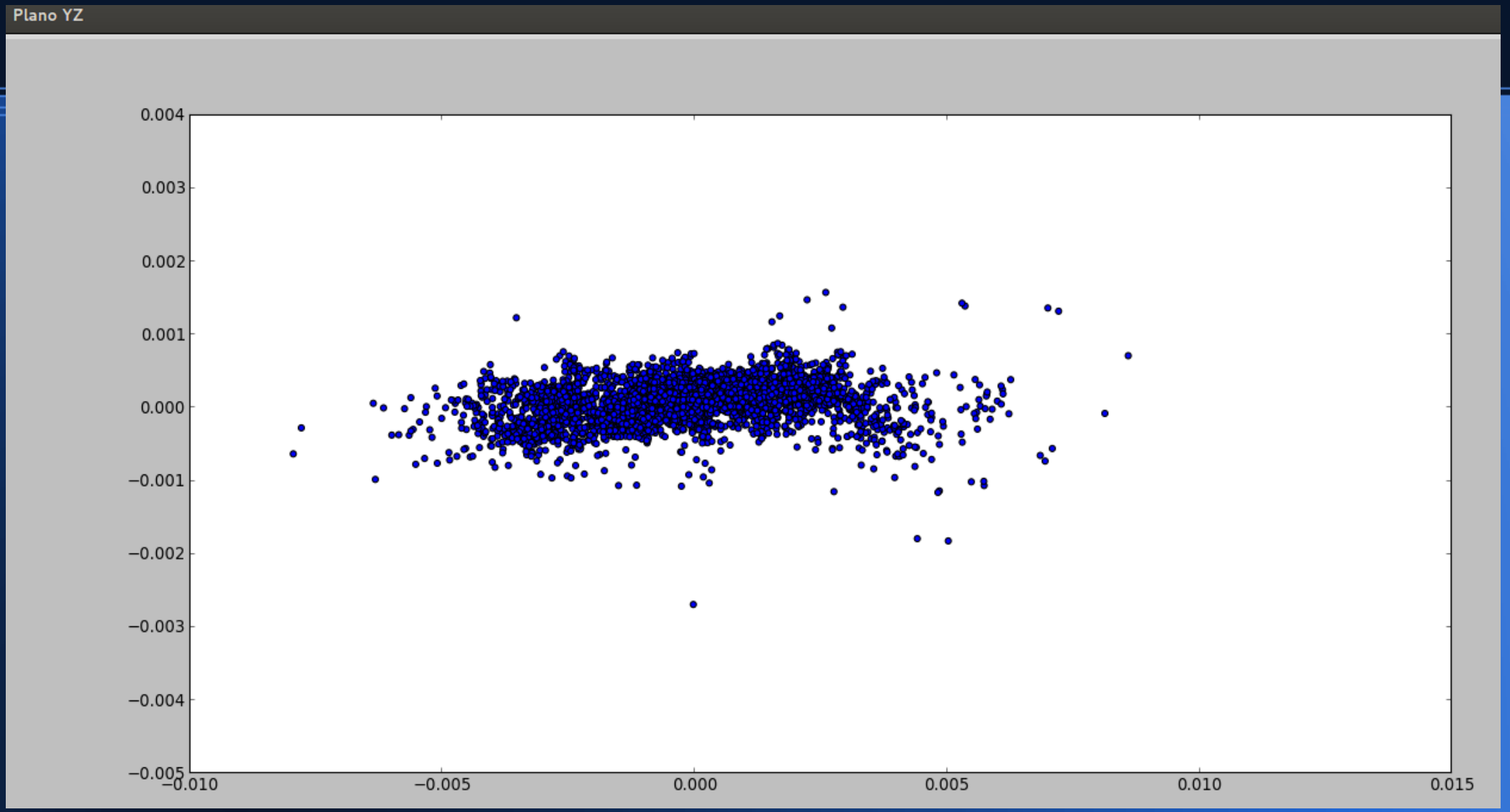
- Utilice la función fromfile para cargar los datos a un numpy array con un shape de 5000x3
- Extraiga los arreglos x, y, z de forma separada
- Plotee un gráfico de 3 filas por 1 columna, con la proyección de los planos xy, xz, yz
- Decore los plots a su gusto.
- Plotee los 3 planos en ventanas separadas.

Cumulo estelar




Plano XY
0.015
0.010
0.005
0.000
−0.005
−0.010
−0.015
−0.010
−0.005
0.000
0.005
0.010
0.015


Plano XZ
0.004
0.003
0.002
0.001
0.000
−0.001
−0.002
−0.003
−0.004
−0.005
−0.010
−0.005
0.000
0.005
0.010
0.015


Plano YZ
0.004
0.003
0.002
0.001
0.000
−0.001
−0.002
−0.003
−0.004
−0.005
−0.010
−0.005
0.000
0.005
0.010
0.015

Mas ejemplos y detalles

- http://matplotlib.org/users/pyplot_tutorial.html