

Python Analysis & Visualization

Peter Wang, *Continuum Analytics*
`pwang@continuum.io`

Joseph Cottam, *Indiana University*
`jcottam@indiana.edu`

Draft Outline

- Intro to Python
 - IDE & Dev Tools
 - Running scripts & programs
 - Language basics
- Numpy: Numerical & array library
- Pandas: Dataframes for Python
- Matplotlib
- Chaco
- Bokeh
- Overview of other Python viz tools

Overview of Tools & Intro to Python

Intro to Numpy

Intro to Matplotlib

More Advanced Matplotlib

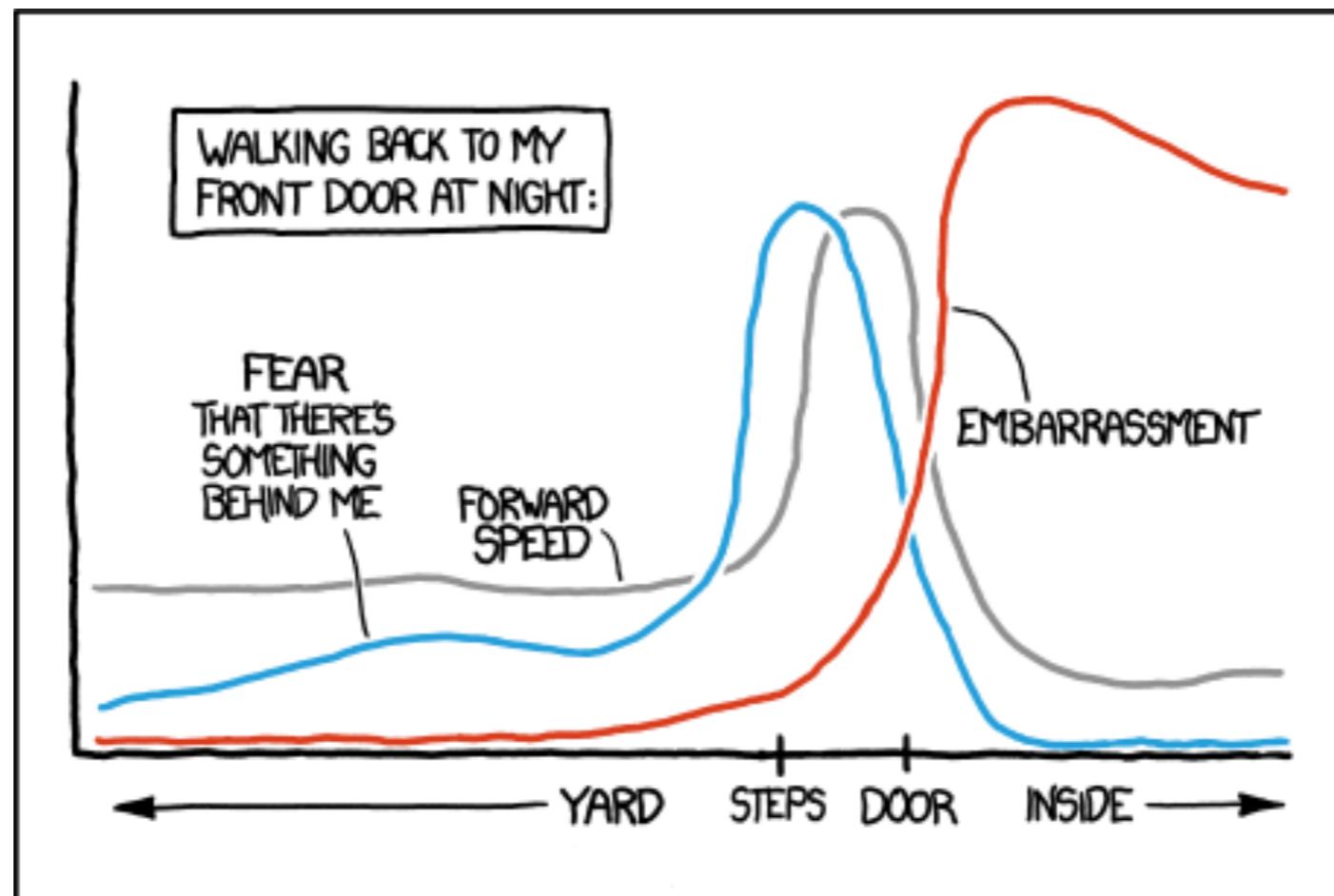
- LaTeX labels / titles
`plt.title(r'$\sigma_i=15$')`
- Basemap
- Underlying Artist interface
- mplh5canvas

Matplotlib hack: Super Mario

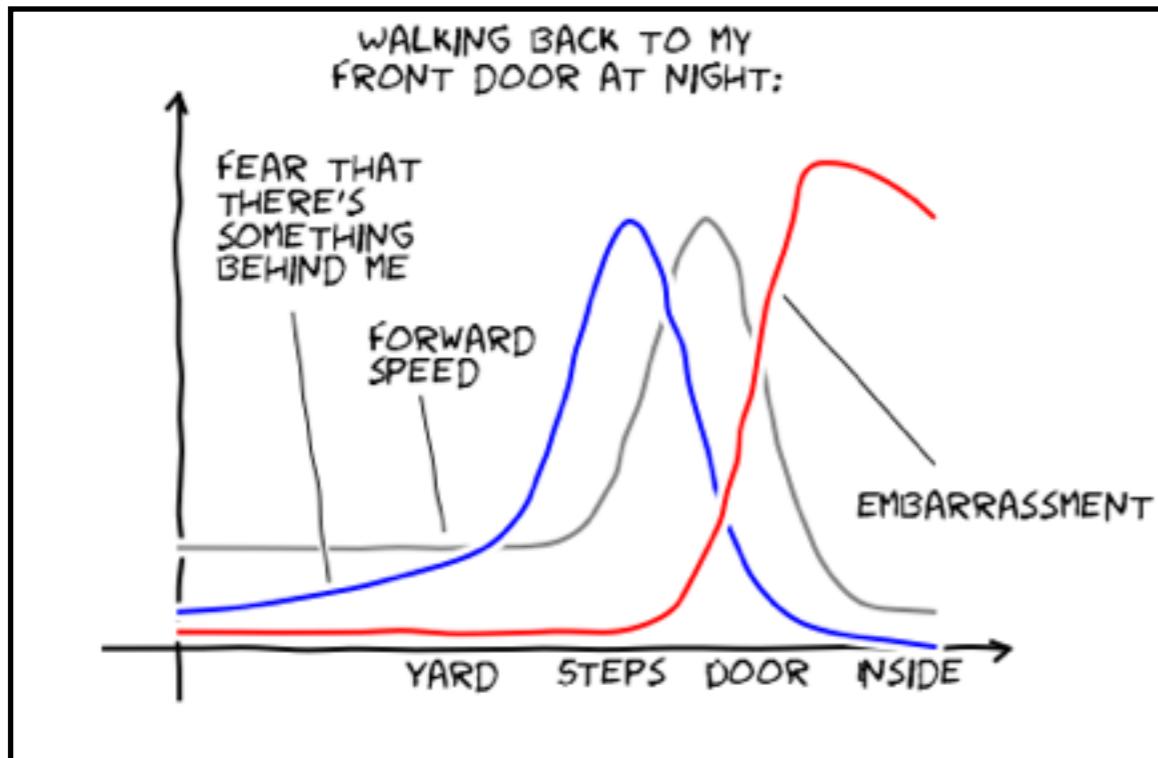


<http://jakevdp.github.com/blog/2013/01/13/hacking-super-mario-bros-with-python/>

Matplotlib hack: XKCD



Matplotlib hack: XKCD



```
def norm(x, x0, sigma):
    return np.exp(-0.5 * (x - x0) ** 2 / sigma ** 2)

def sigmoid(x, x0, alpha):
    return 1. / (1. + np.exp(-(x - x0) / alpha))

# define the curves
x = np.linspace(0, 1, 100)
y1 = np.sqrt(norm(x, 0.7, 0.05)) + 0.2 * (1.5 - sigmoid(x, 0.8, 0.05))
y2 = 0.2 * norm(x, 0.5, 0.2) + np.sqrt(norm(x, 0.6, 0.05)) + 0.1 * \
    (1 - sigmoid(x, 0.75, 0.05))
y3 = 0.05 + 1.4 * norm(x, 0.85, 0.08)
y3[x > 0.85] = 0.05 + 1.4 * norm(x[x > 0.85], 0.85, 0.3)

# draw the curves
ax = pl.axes()
ax.plot(x, y1, c='gray')
ax.plot(x, y2, c='blue')
ax.plot(x, y3, c='red')
ax.text(0.3, -0.1, "Yard")
ax.text(0.5, -0.1, "Steps")
ax.text(0.7, -0.1, "Door")
ax.text(0.9, -0.1, "Inside")

ax.text(0.05, 1.1, "fear that\nthere's\nsomething\nbehind me")
ax.plot([0.15, 0.2], [1.0, 0.2], '-k', lw=0.5)

ax.text(0.25, 0.8, "forward\nspeed")
ax.plot([0.32, 0.35], [0.75, 0.35], '-k', lw=0.5)

ax.text(0.9, 0.4, "embarrassment")
ax.plot([1.0, 0.8], [0.55, 1.05], '-k', lw=0.5)

ax.set_title("Walking back to my\nfront door at night:")

ax.set_xlim(0, 1)
ax.set_ylim(0, 1.5)

# modify all the axes elements in-place
XKCDify(ax, expand_axes=True)
```

<http://jakevdp.github.com/blog/2012/10/07/xkcd-style-plots-in-matplotlib/>

Intro to Pandas

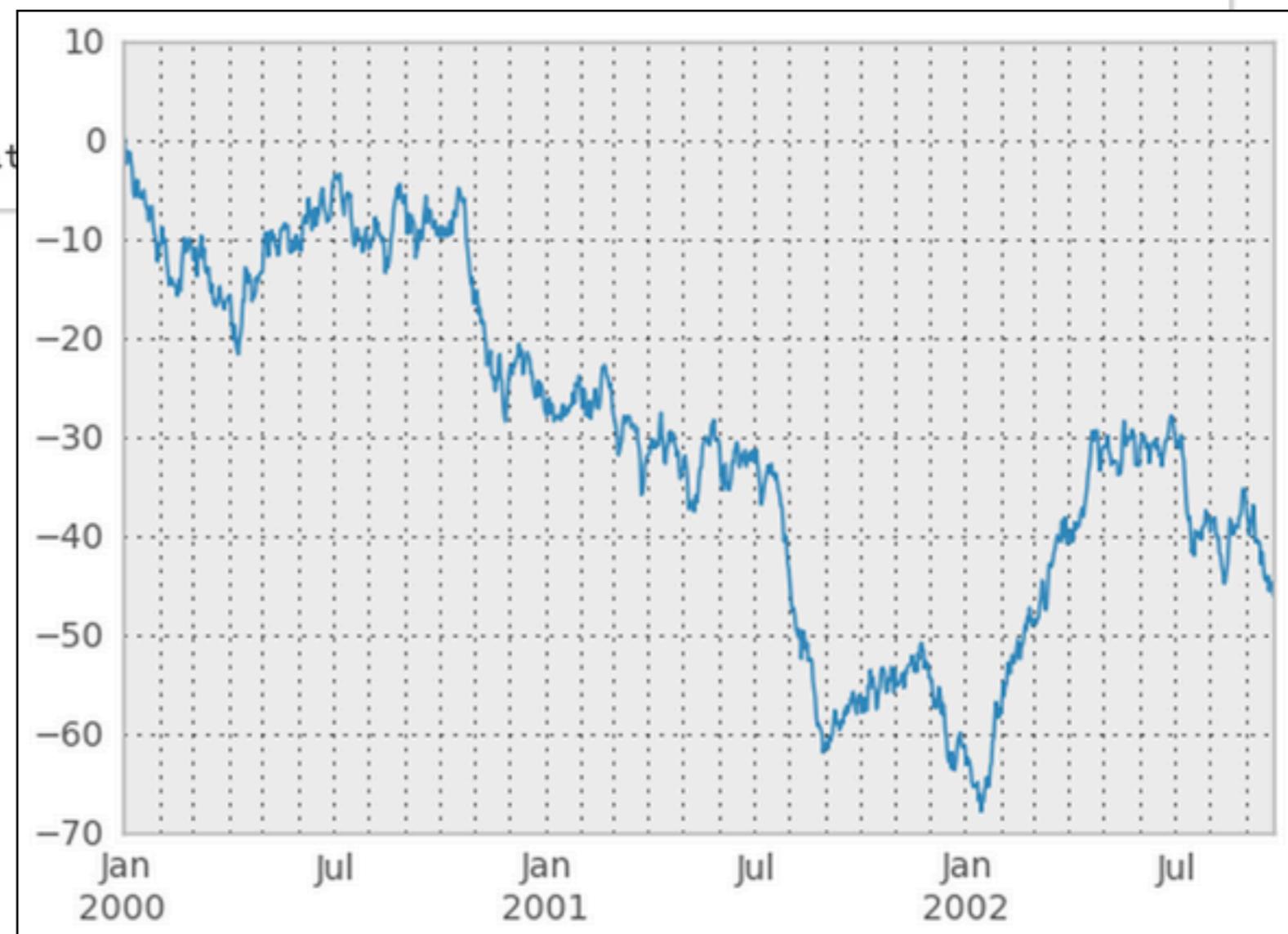
Pandas Plotting

```
In [2]: ts = Series(randn(1000), index=date_range('1/1/2000', periods=1000))
```

```
In [3]: ts = ts.cumsum()
```

```
In [4]: ts.plot()
```

```
<matplotlib.axes.AxesSubplot at 0x10c1a50>
```

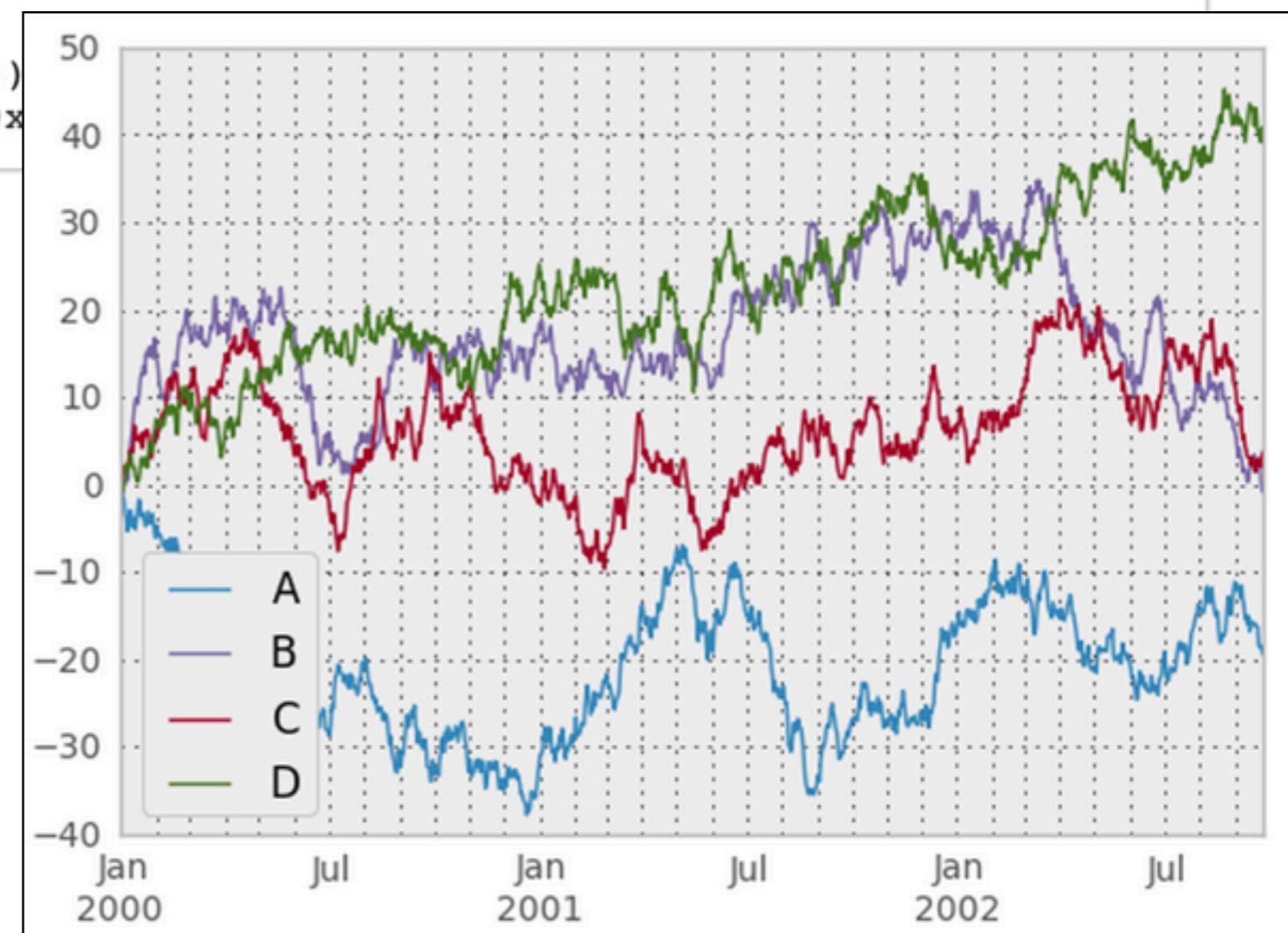


Pandas Plotting

```
In [6]: df = DataFrame(randn(1000, 4), index=ts.index, columns=list('ABCD'))
```

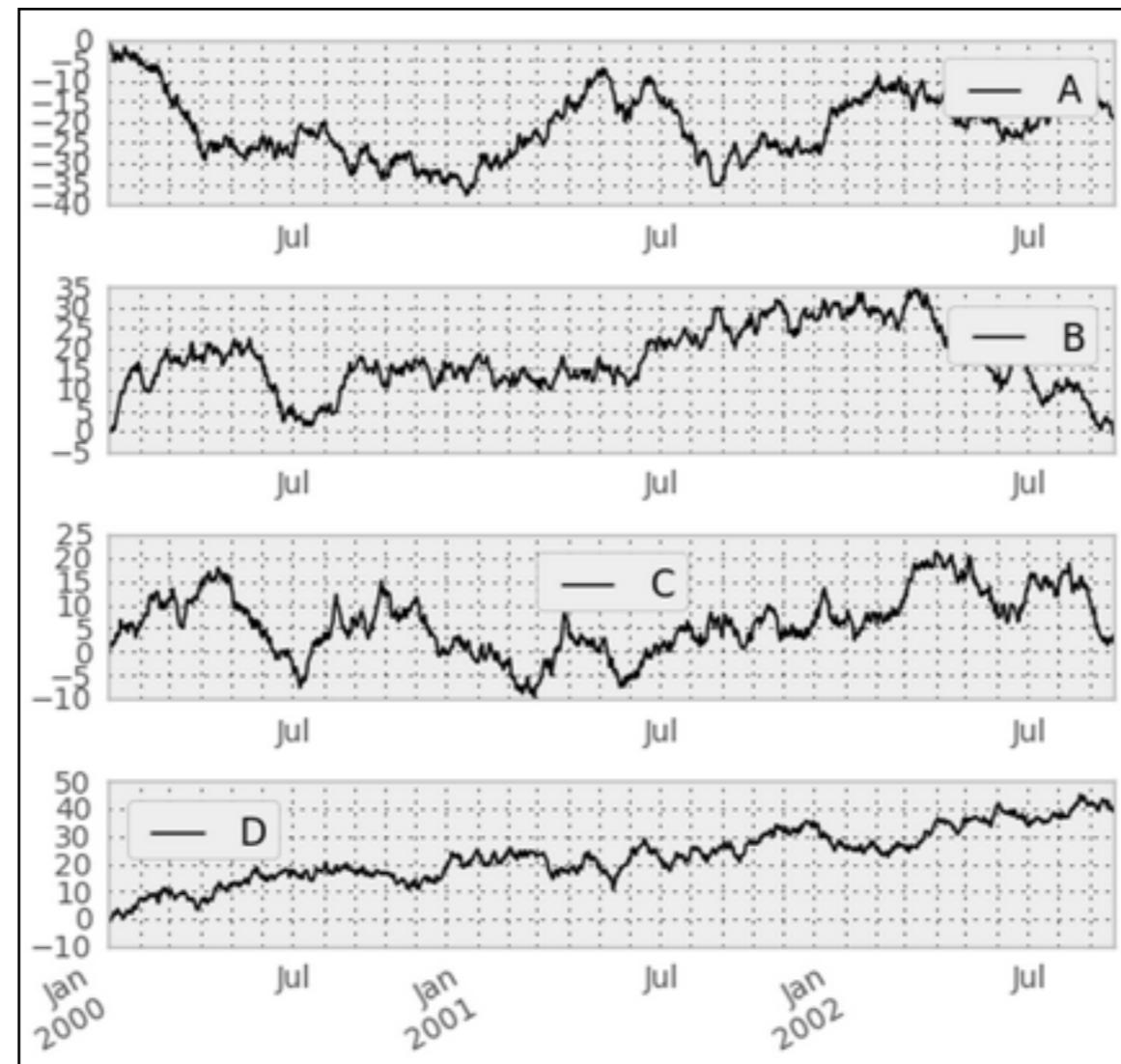
```
In [7]: df = df.cumsum()
```

```
In [8]: plt.figure(); df.plot()  
<matplotlib.legend.Legend at 0x
```



Subfigures

```
In [10]: df.plot(subplots=True, figsize=(6, 6)); plt.legend(loc='best')  
<matplotlib.legend.Legend at 0xeeba650>
```



More Tutorial...

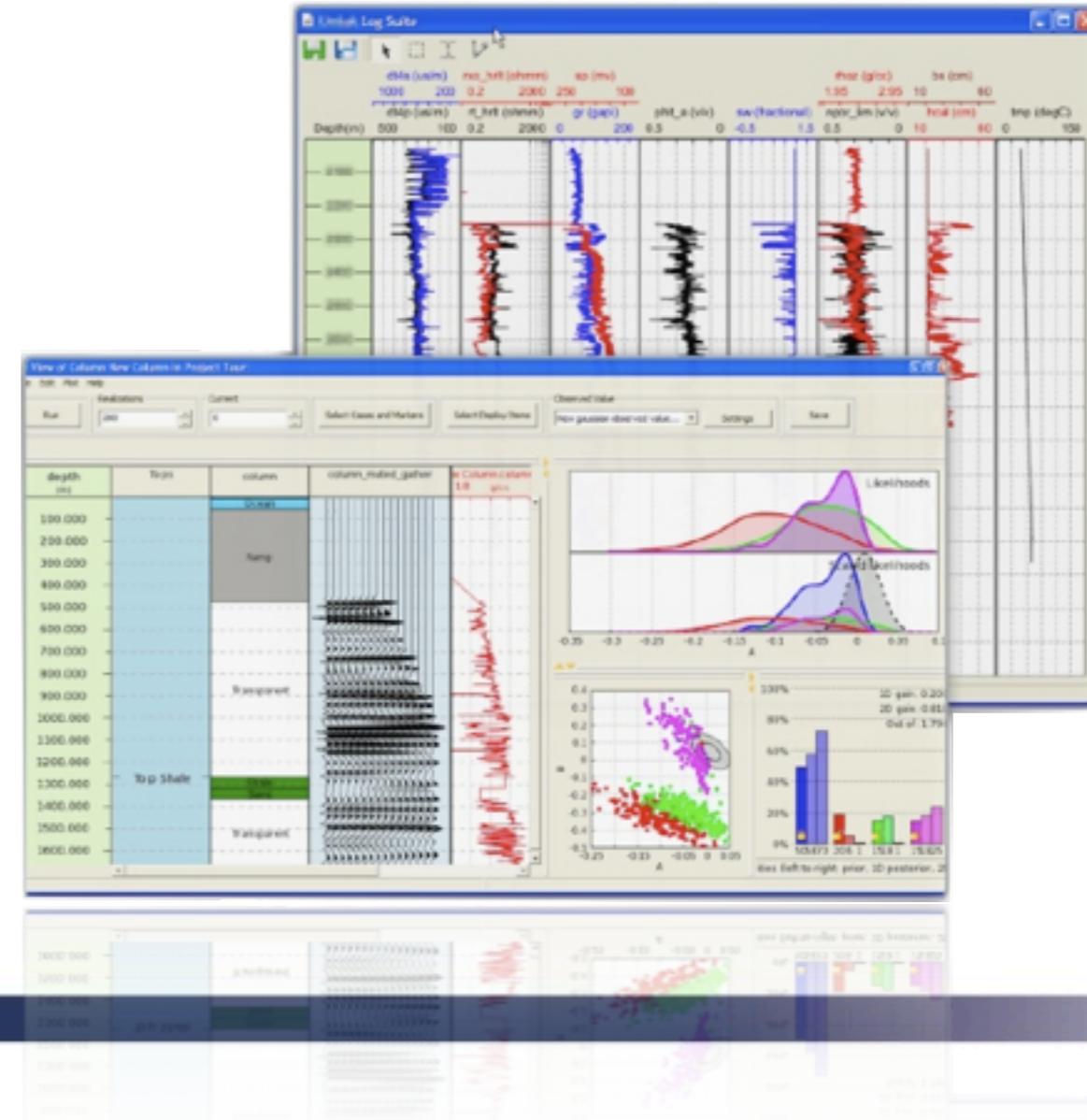
[http://pandas.pydata.org/pandas-docs/stable/
visualization.html](http://pandas.pydata.org/pandas-docs/stable/visualization.html)

Chaco Overview

Chaco

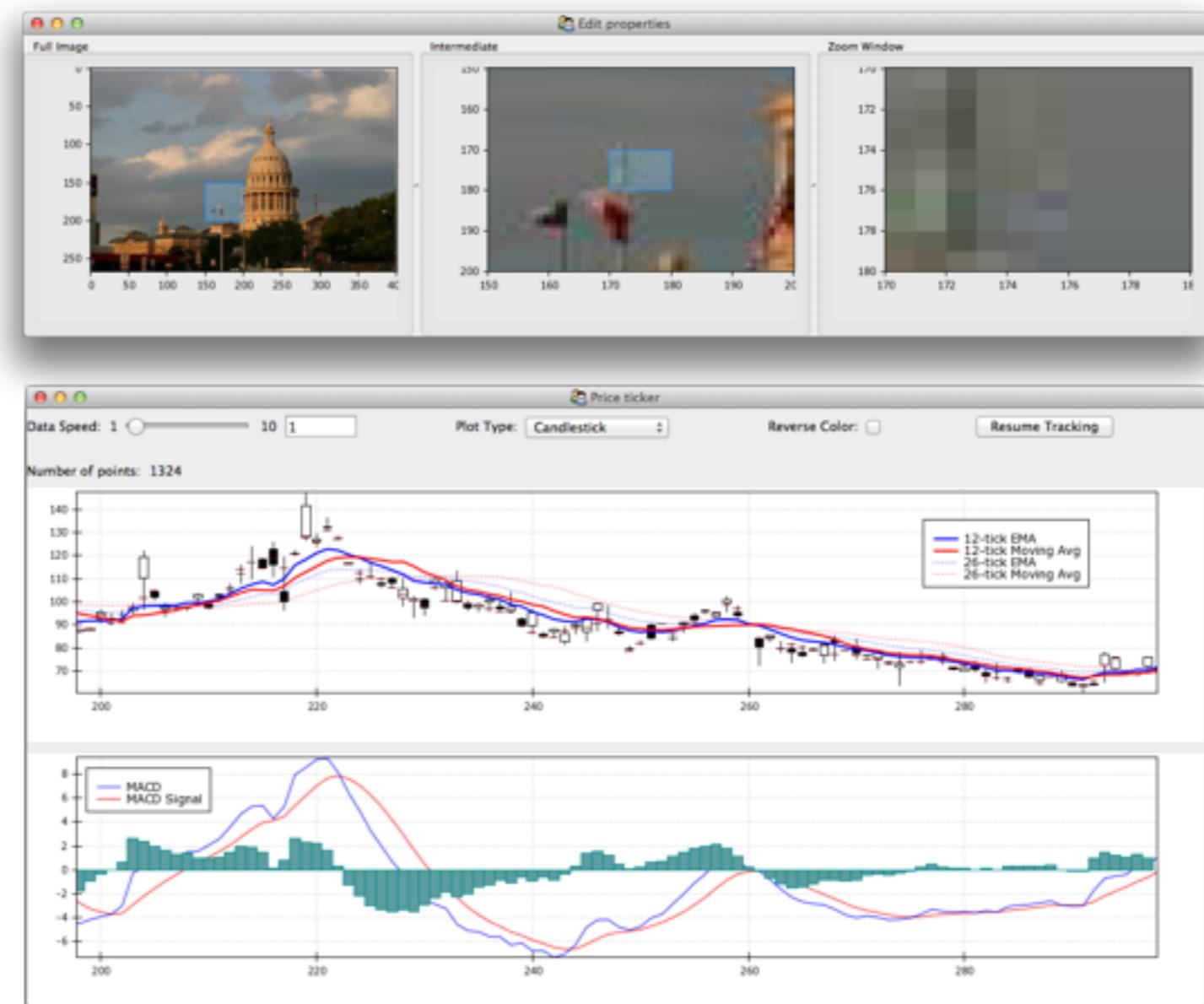
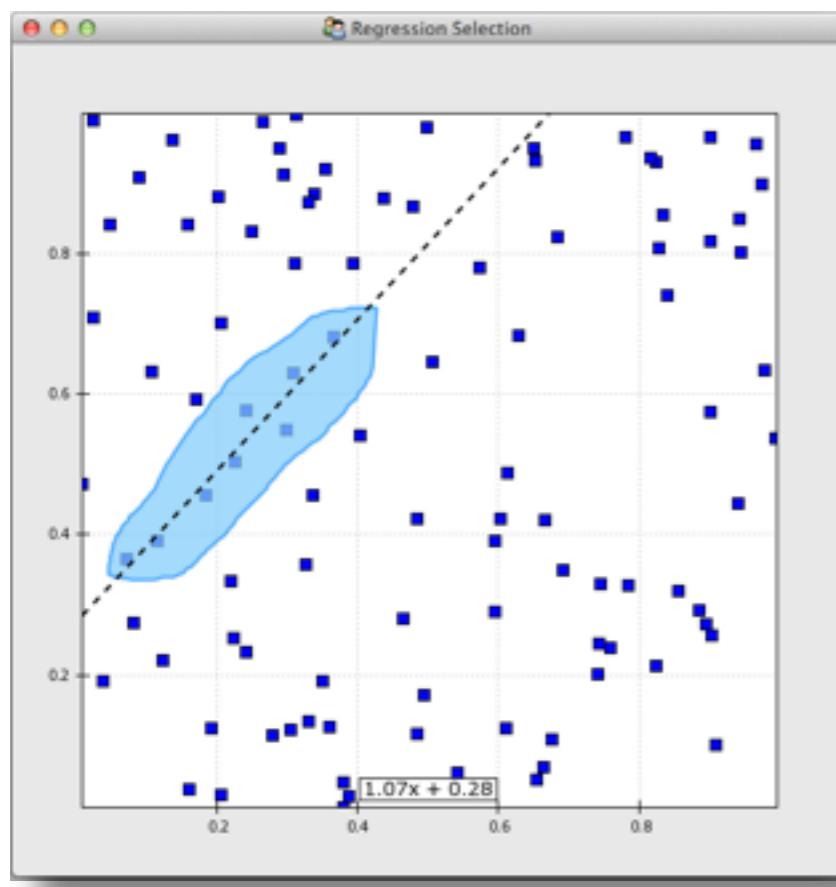
Primary design goals:

- plotting widgets for GUI applications
- modular, extensible architecture
- interactive
- realtime



Chaco

Can also be used for interactive data exploration



Bokeh: Python Plotting for the Web

Bokeh

Inspirations:

- Chaco: interactive, viz pipeline for large data
- Protovis & Stencil :
 Binding visual Glyphs to data and expressions
- ggplot2: faceting, statistical overlays

Design goal:

Accessible, extensible, interactive plotting for the web...

... for non-Javascript programmers

Bokeh

v0.2 Just released: <http://continuum.io/blog/bokeh02>

- <http://bokeh.pydata.org>
- <http://github.com/ContinuumIO/bokeh>

Interactive IPython notebook:

- <https://www.wakari.io/sharing/bundle/pwang/cars>

Bokeh - HL Grammar (coming)

Stencil

```
stream flowers(sepalL, petalW, sepalW,  
petalL, species, obs)
```

```
layer FlowerPlot  
guide
```

```
legend from FILL_COLOR
```

```
axis from Y
```

```
axis from X
```

```
from flowers
```

```
ID: obs
```

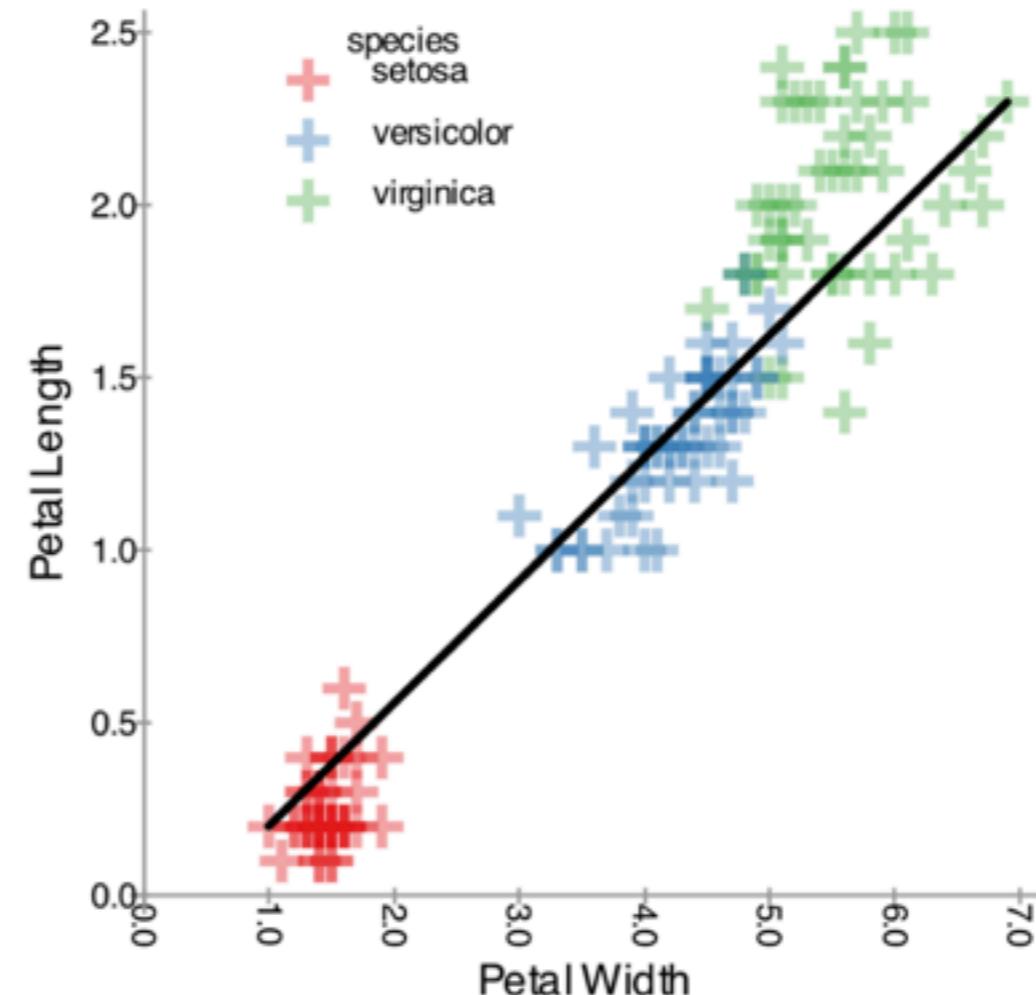
```
X:* Scale(0,100, petalW)
```

```
Y:* Scale(0,100, petalL)
```

```
FILL_COLOR: BrewerColors(species)
```

```
REGISTRATION: "CENTER"
```

```
SHAPE : "CROSS"
```



Bokeh - Abstract Rendering

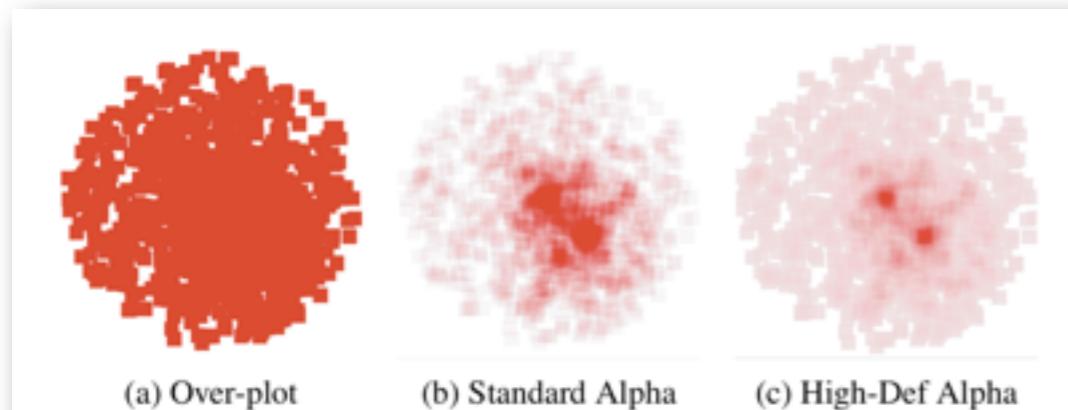
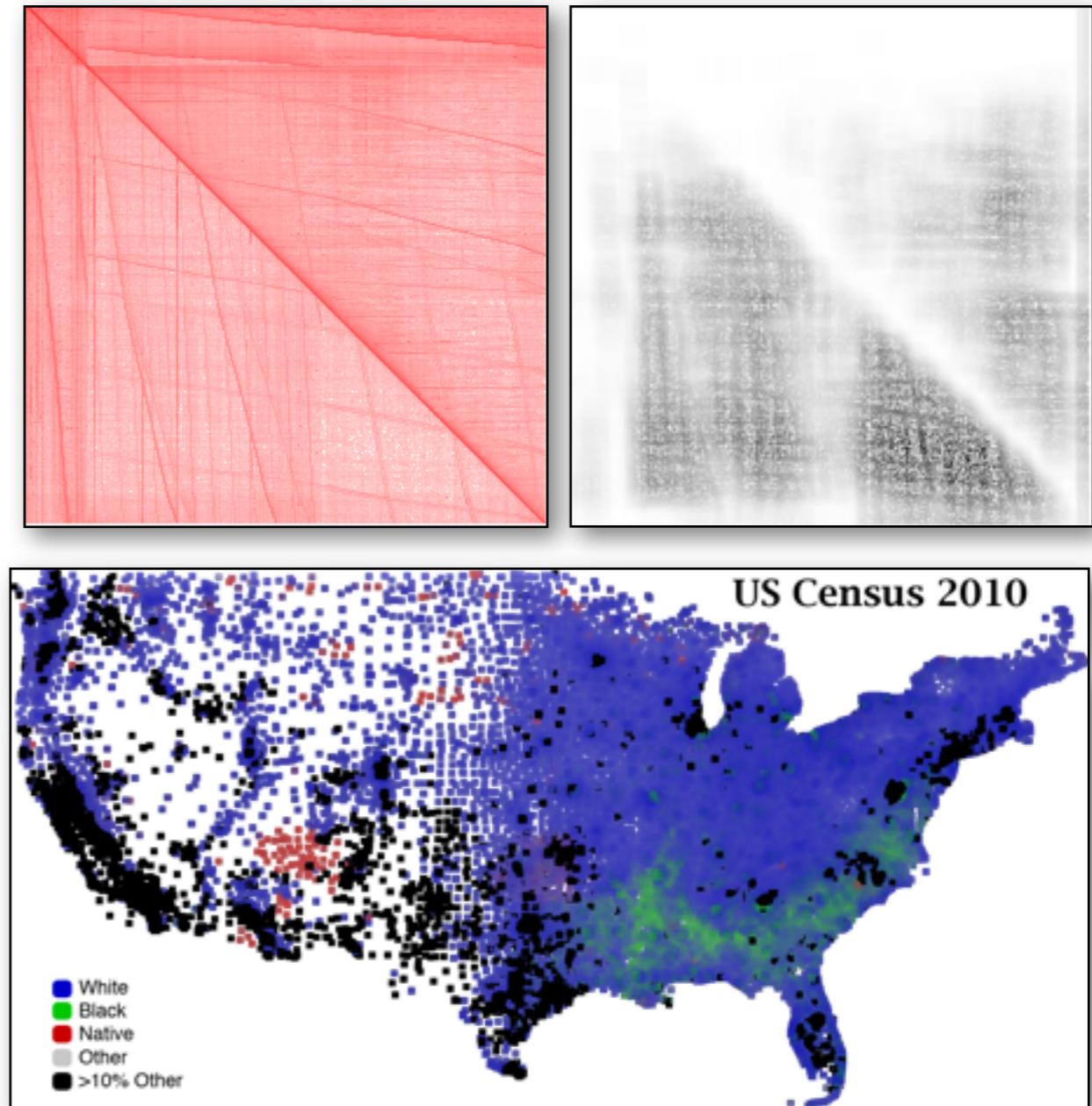
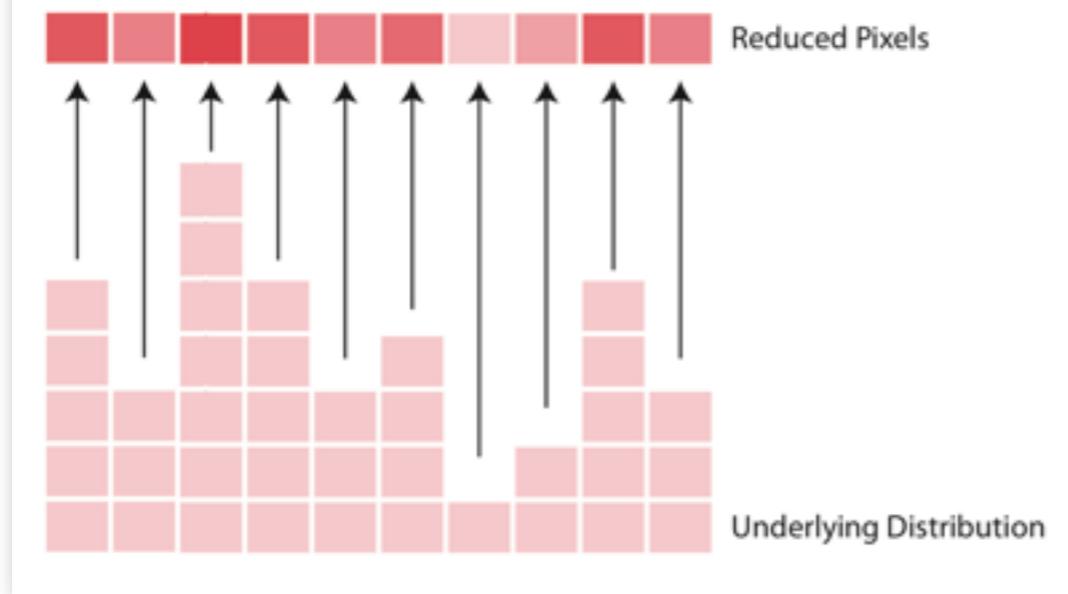
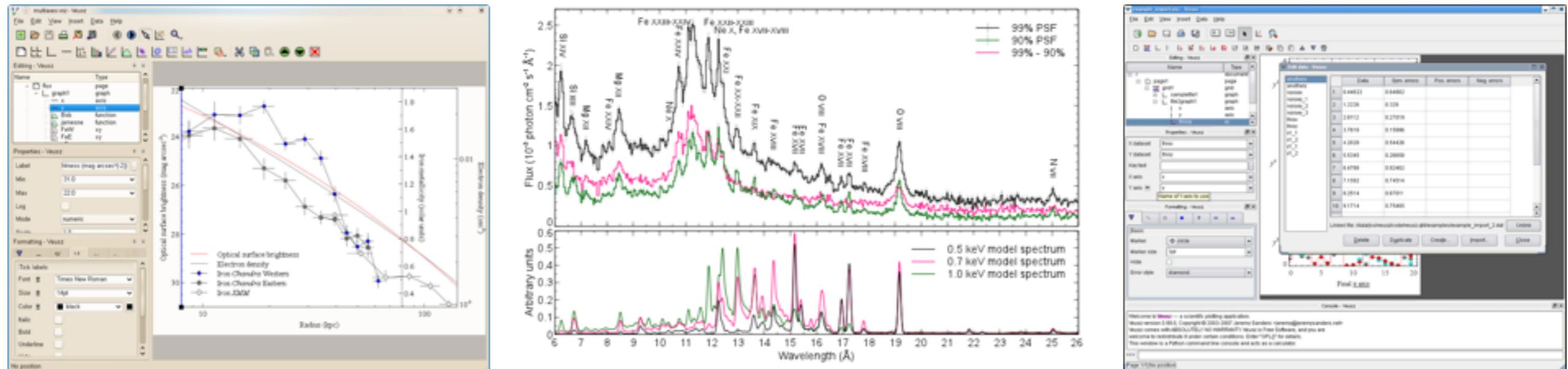


Fig. 3: A scatter plot of synthetic data under various treatments.



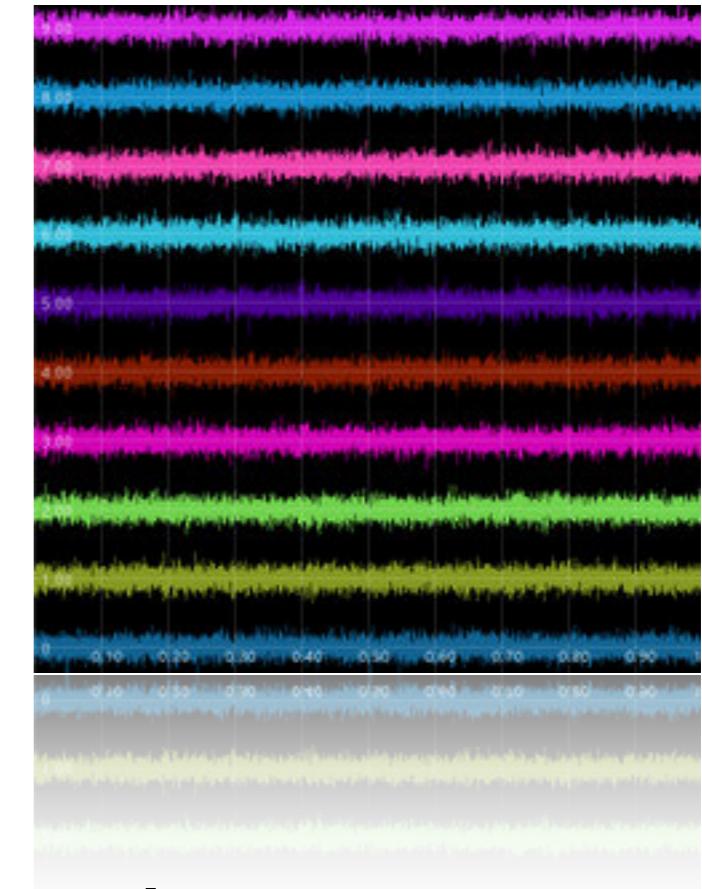
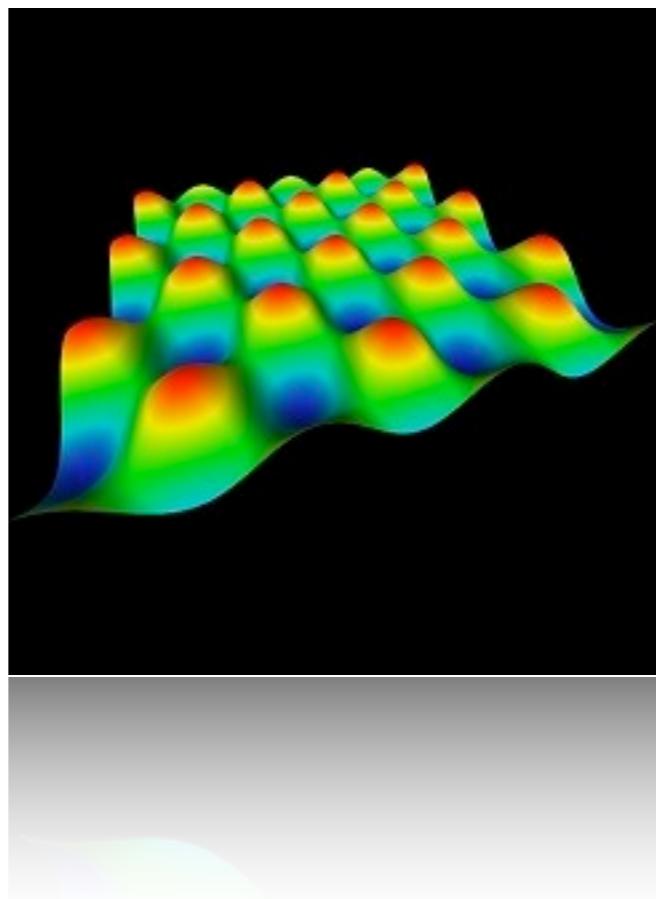
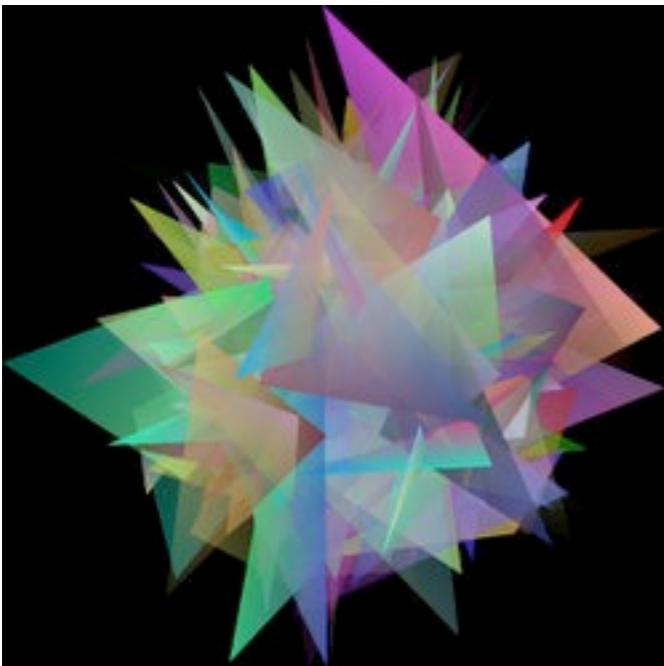
Overview of Other Tools

Veusz



<http://home.gna.org/veusz/>

Galry

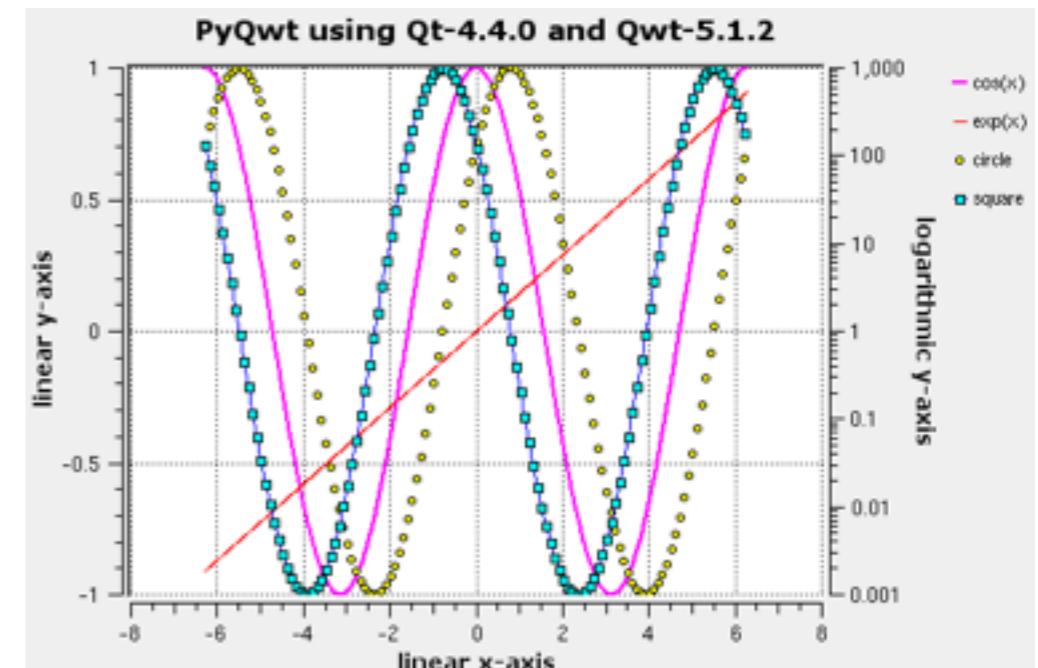


<http://rossant.github.com/galry/>

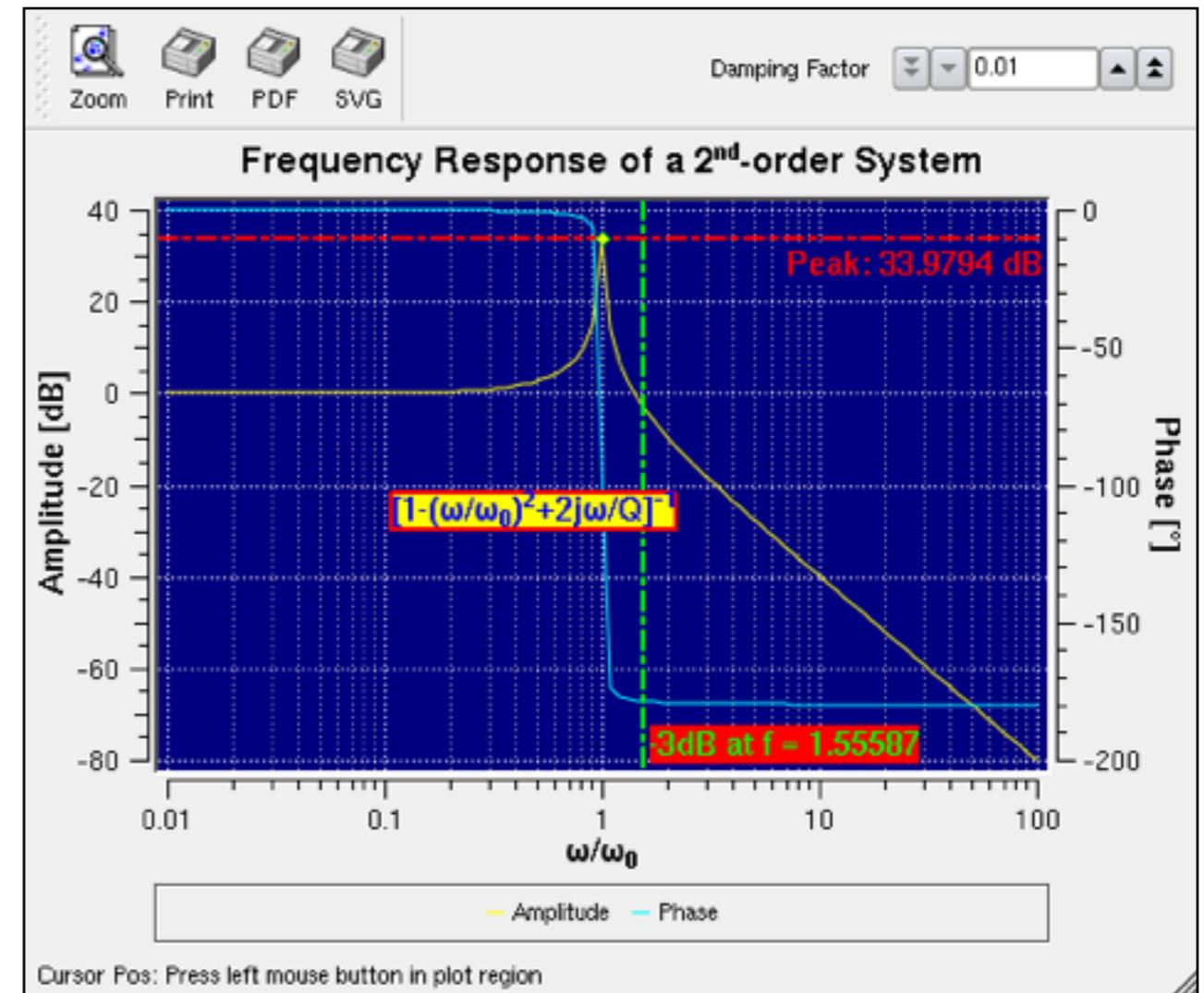
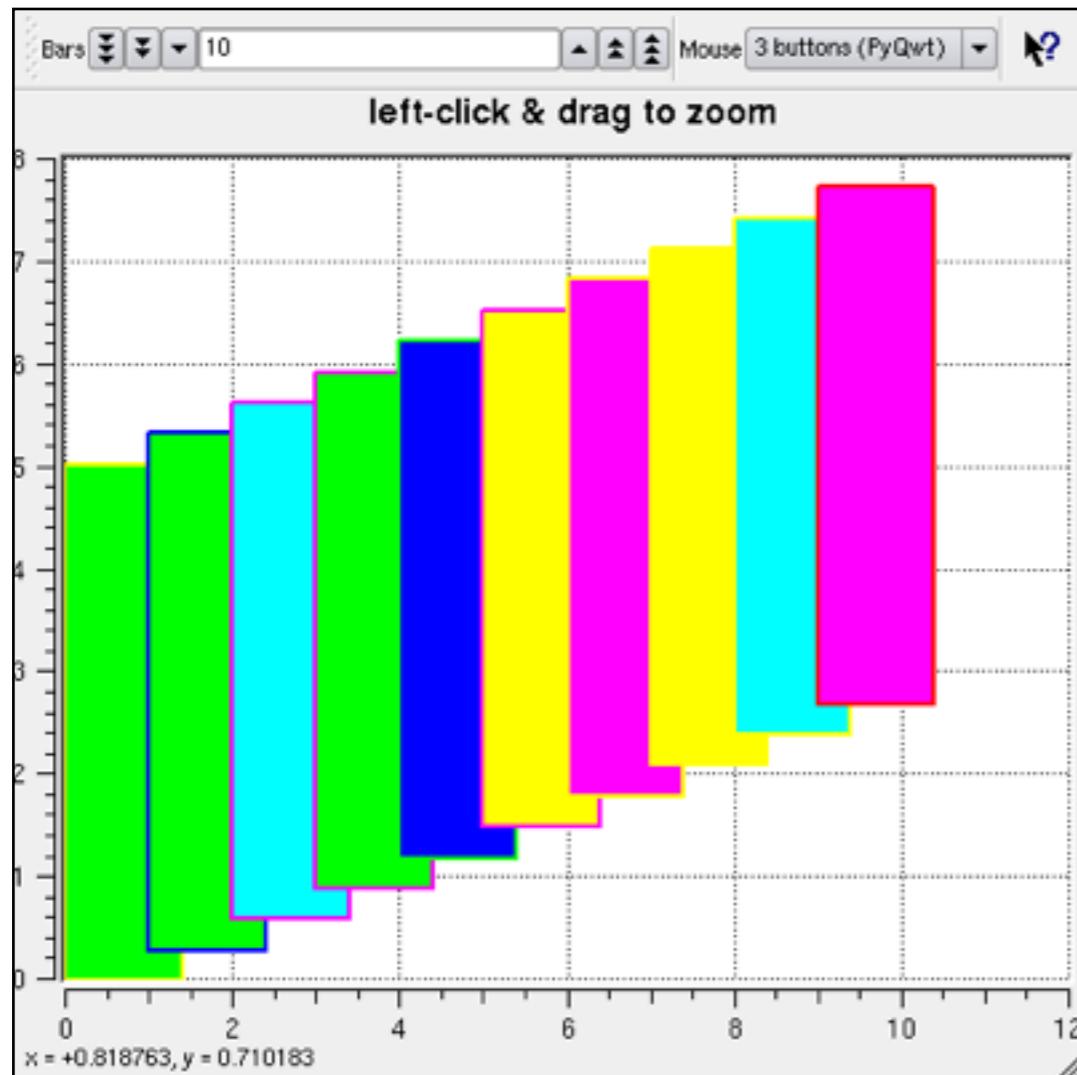
PyQwt

- Wrapper around the Qwt plotting widgets for Qt
- Can be used from the command line via iqt

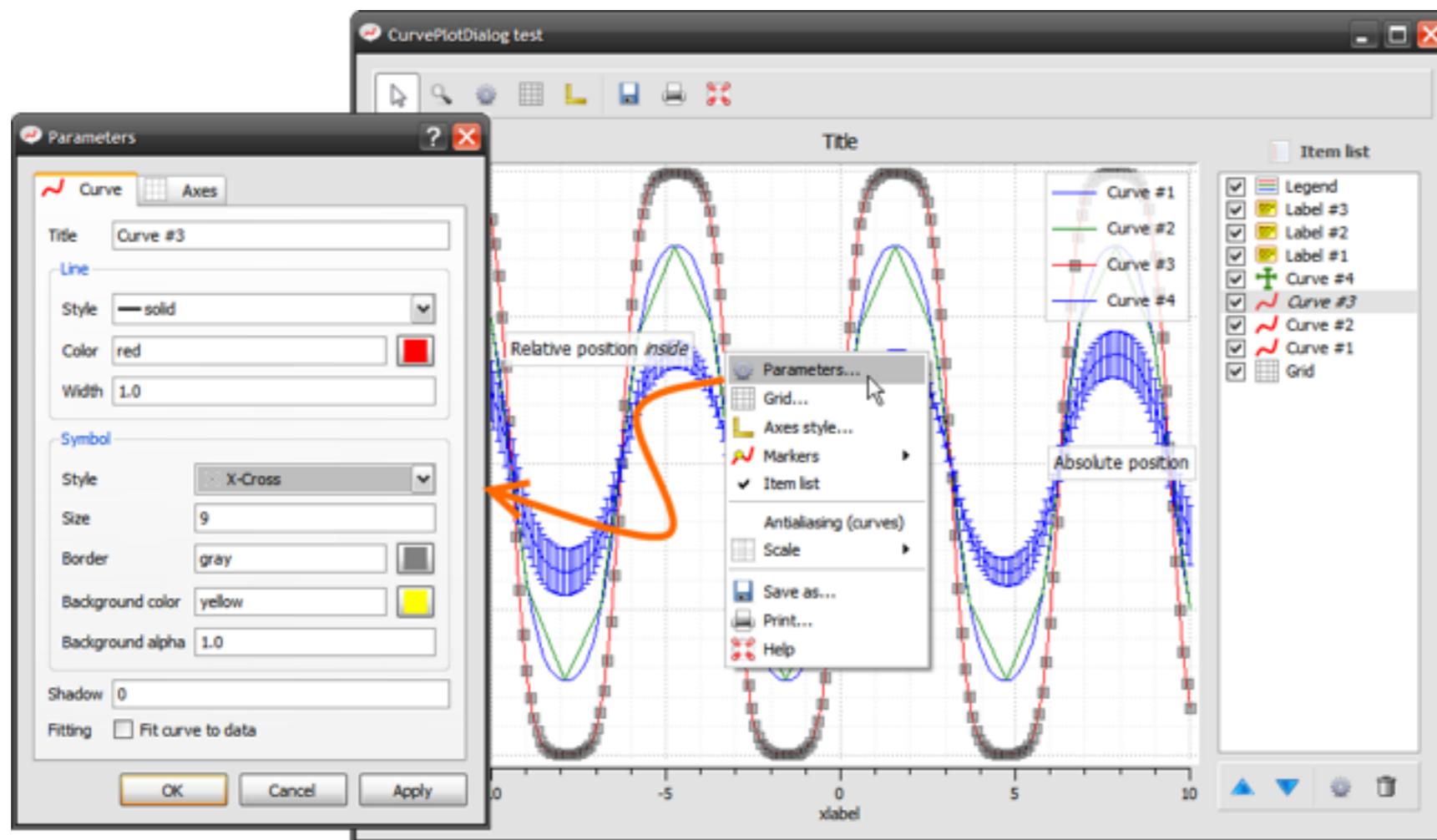
```
>>> import numpy as np
>>> import PyQt4.Qwt5.iqt
>>> from PyQt4.Qwt5.qplt import *
>>> x = np.arange(-2*np.pi, 2*np.pi, 0.01)
>>> p = Plot(Curve(x, np.cos(x),
...                 Pen(Magenta, 2), "cos(x)"),
...             Curve(x, np.exp(x),
...                 Pen(Red), "exp(x)", Y2),
...             Axis(Right, Log),
...             "PyQwt using Qwt-%s -- http://qwt.sf.net" % QWT_VERSION_STR)
```



PyQwt

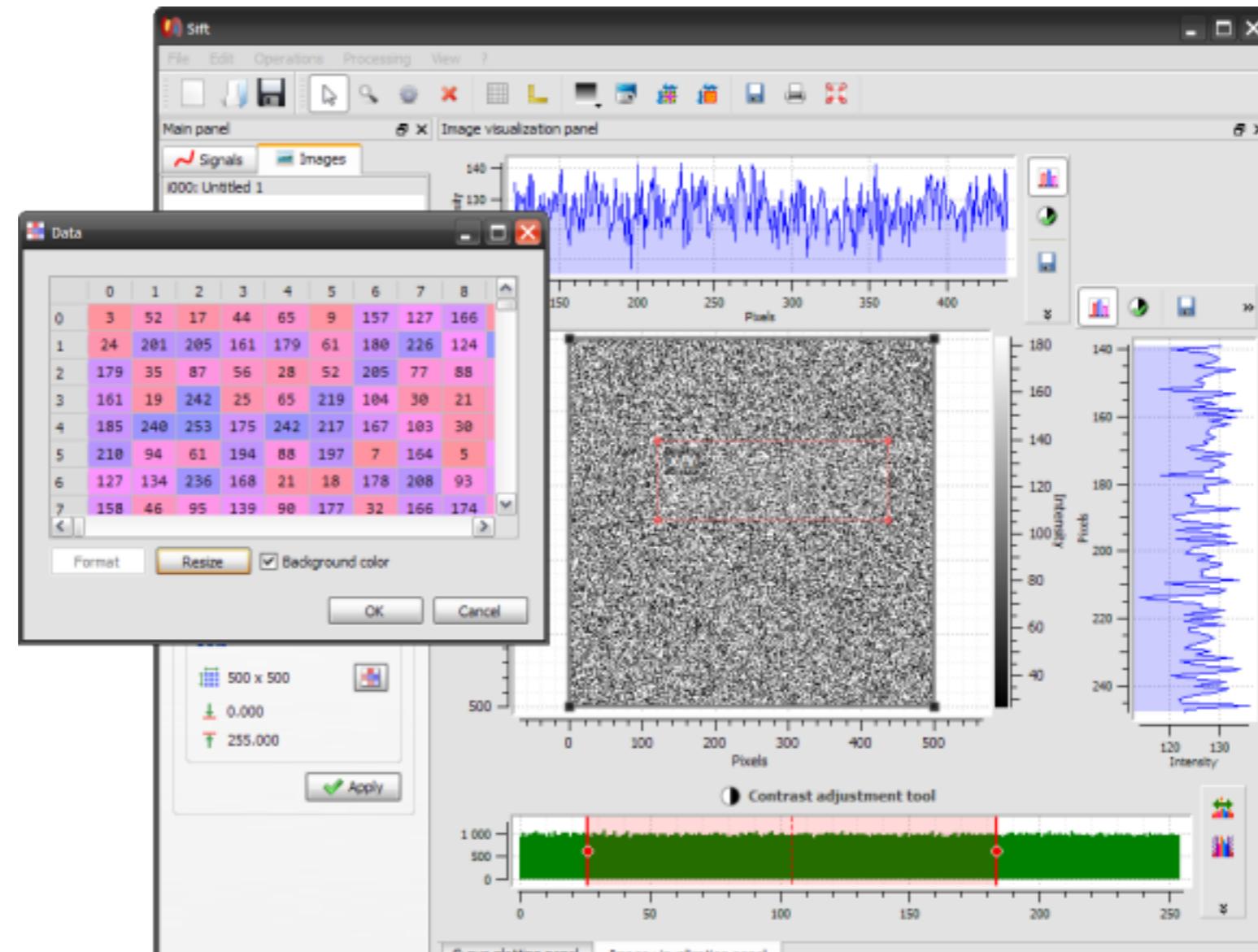


guiqwt

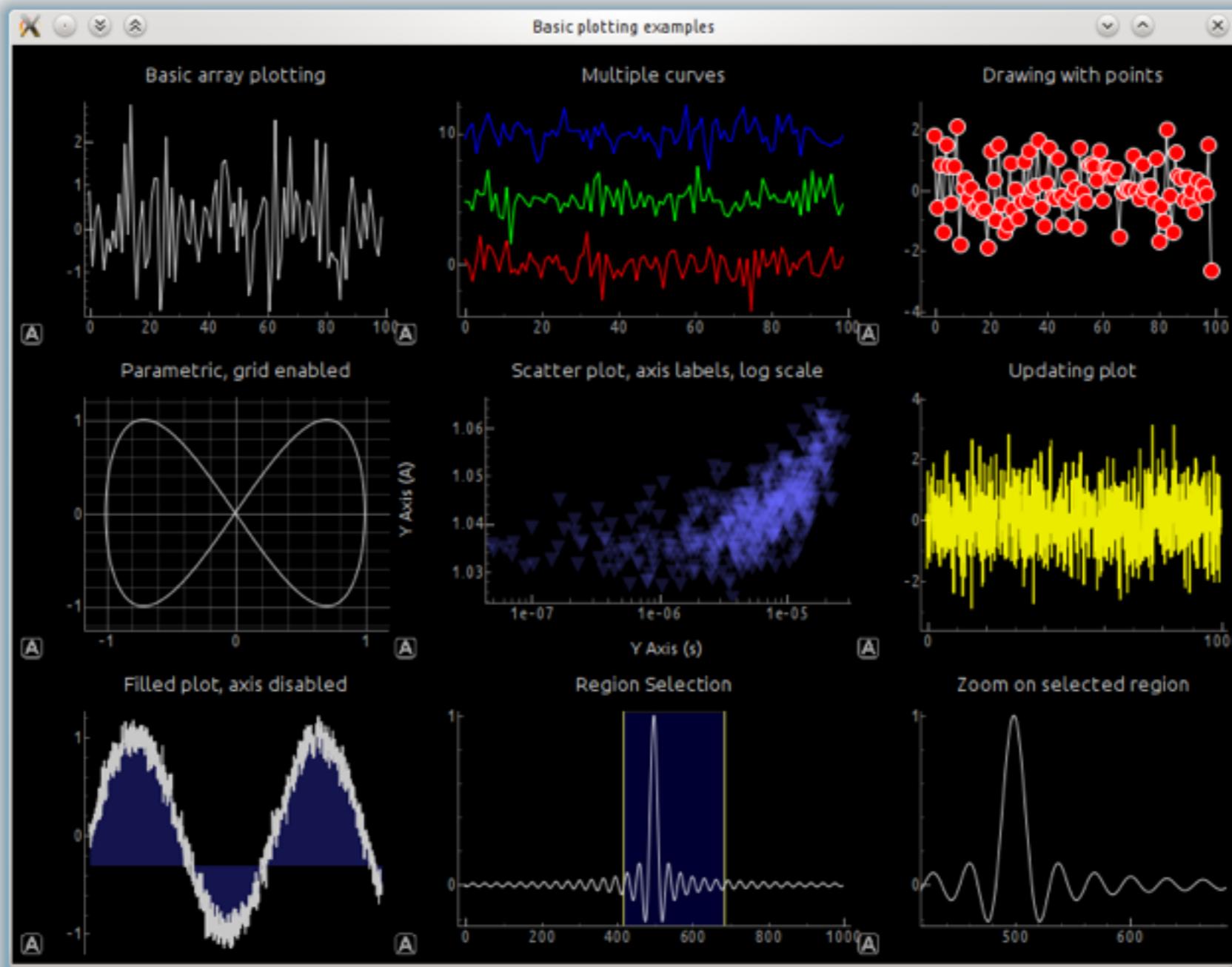


<https://code.google.com/p/guiqwt/>

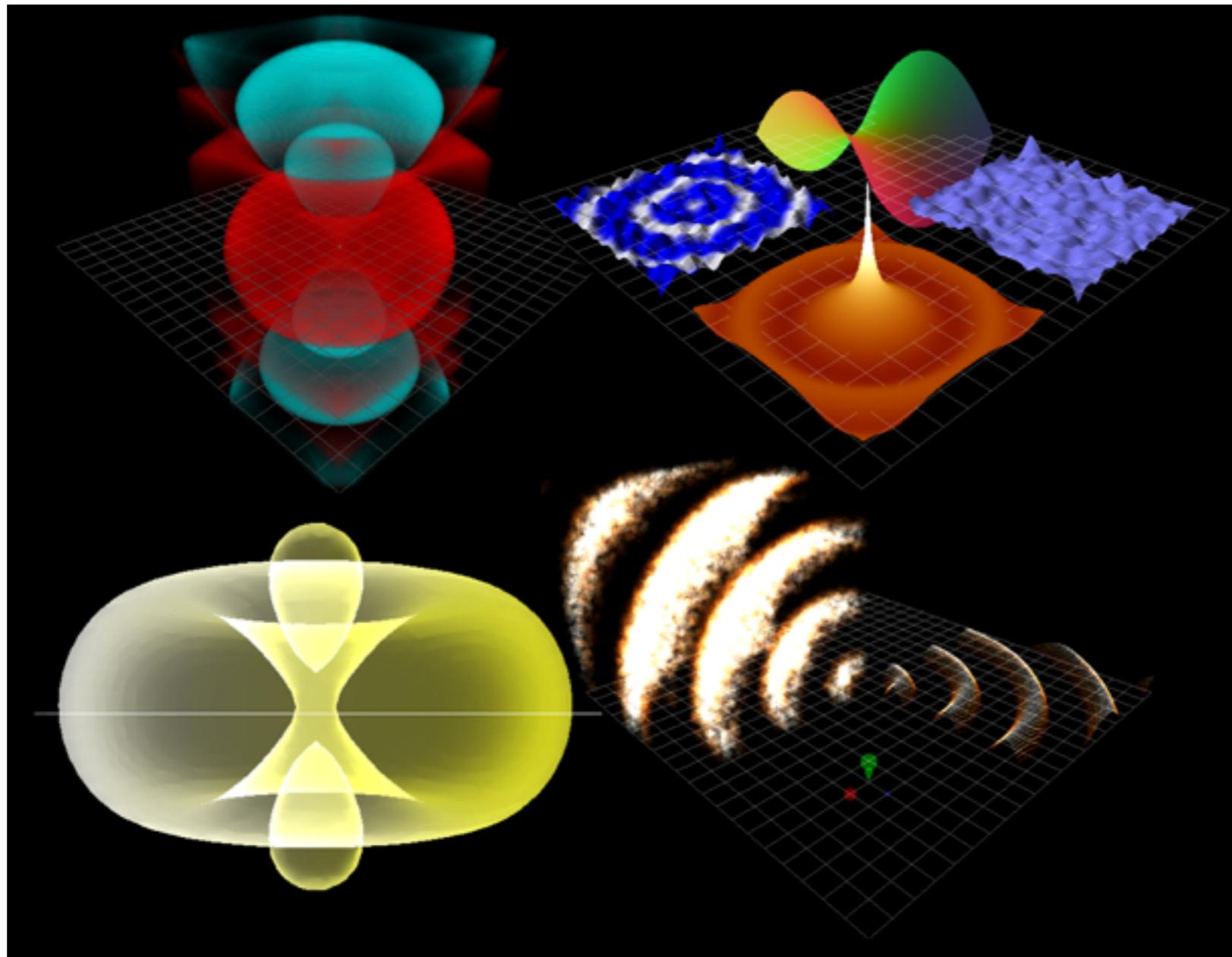
guiqwt



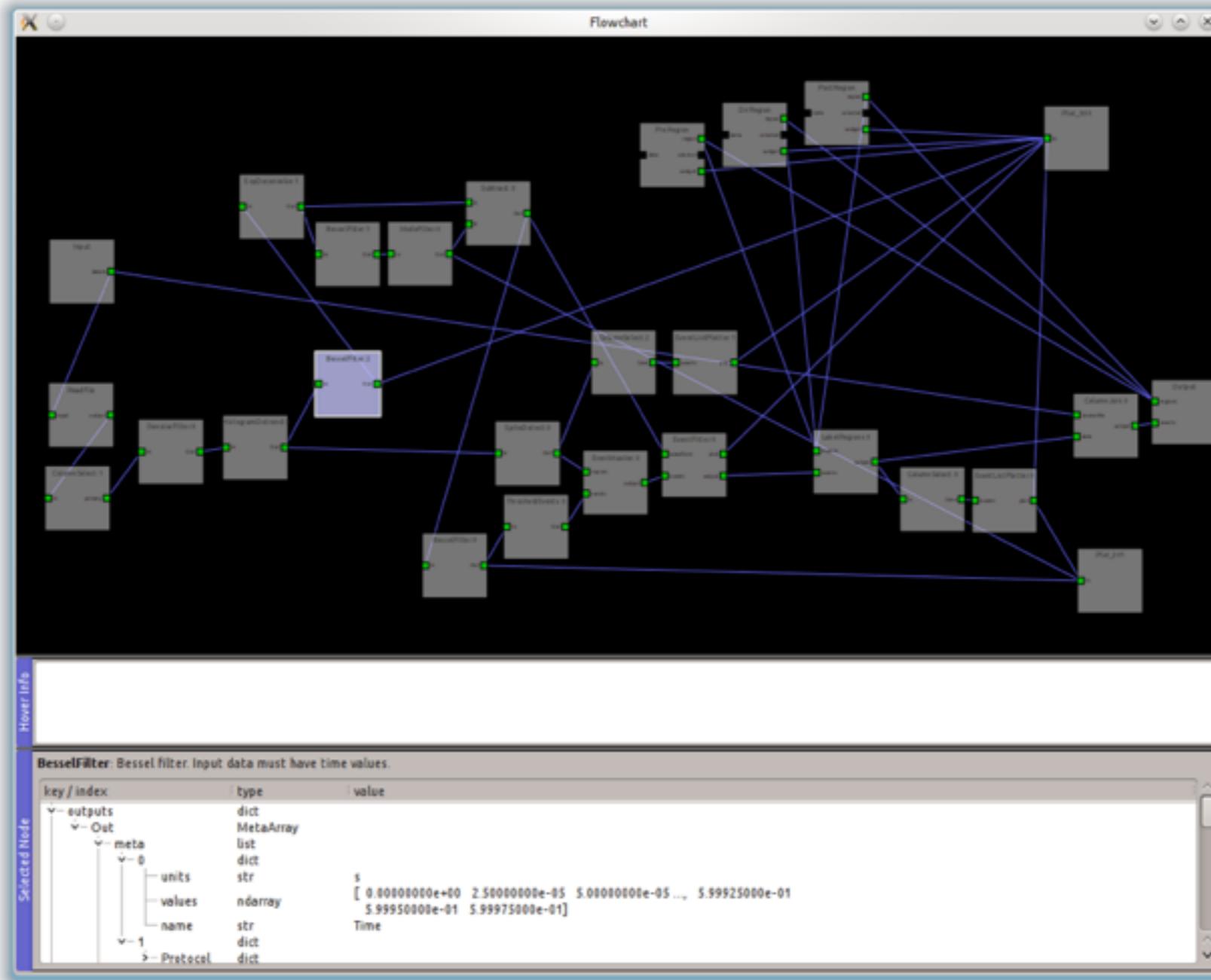
PyQtGraph



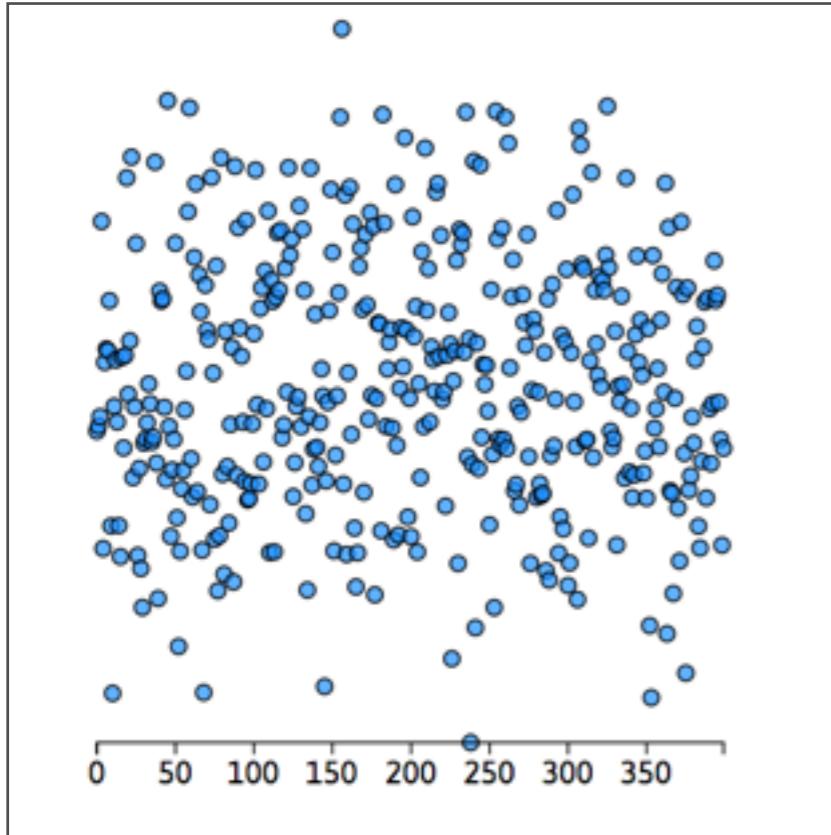
PyQtGraph



PyQtGraph



d3py



```
import d3py
import pandas
import numpy as np

# some test data
T = 100
# this is a data frame with three columns
df = pandas.DataFrame({
    "time" : range(T),
    "pressure": np.random.rand(T),
    "temp" : np.random.rand(T)
})
## build up a figure, ggplot2 style
# instantiate the figure object
fig = d3py.PandasFigure(df, name="basic_example",
                        width=300, height=300)
# add some red points
fig += d3py.geoms.Point(x="pressure", y="temp",
fill="red")
# writes 3 files, starts up a server, then draws
# some beautiful points in Chrome
fig.show()
```

d3py

Figures can have the following added to them:

- Geoms: point, line, bar, area
- Axes: xaxis, yaxis
- Force-directed graph

<https://github.com/mikedewar/d3py>

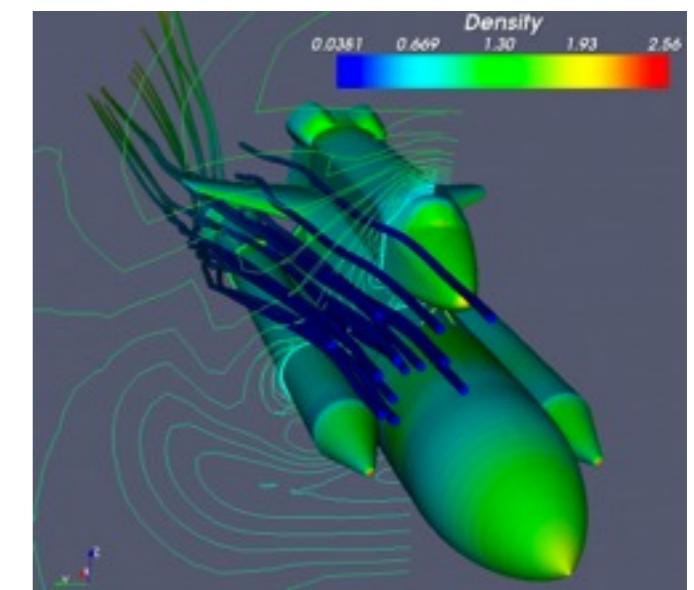
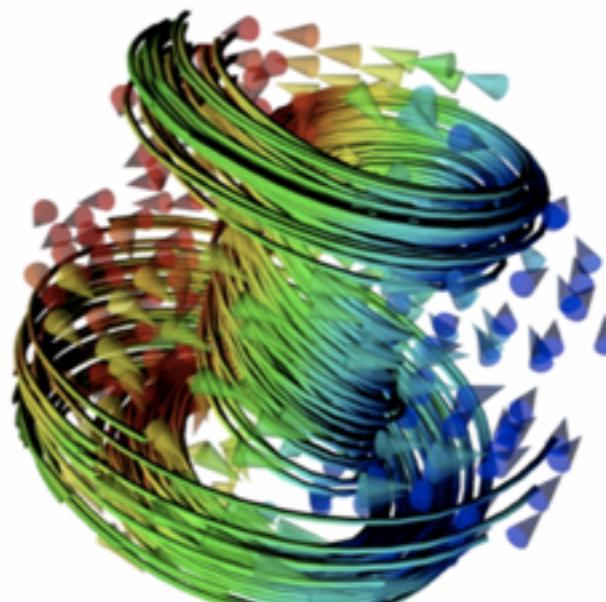
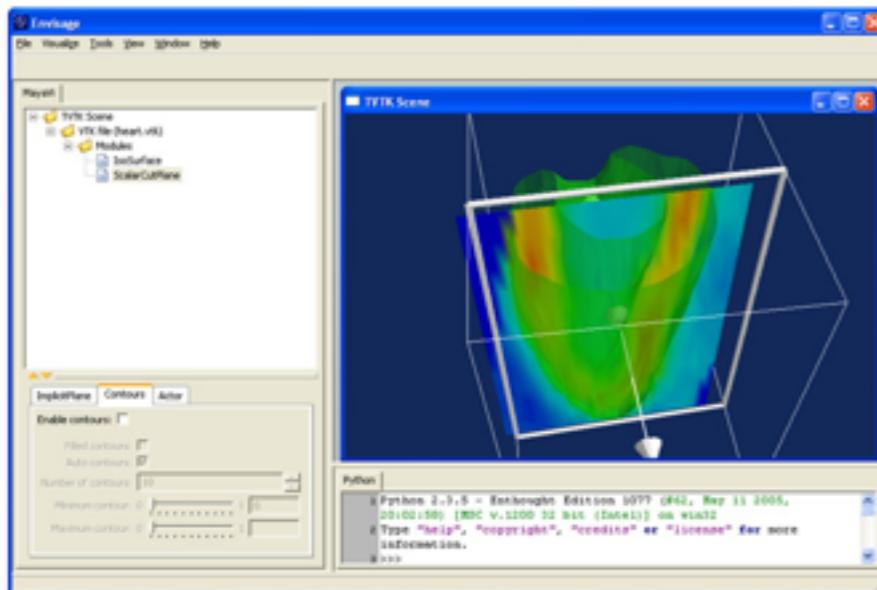
d3py

Top-level PandasFigure and NetworkXFigure.
Each has .show() method that creates:

- HTML code (from template)
- CSS for each Geom
- JSON of data
- Javascript source file (containing d3.js)

(Also, starts a webserver if necessary.)

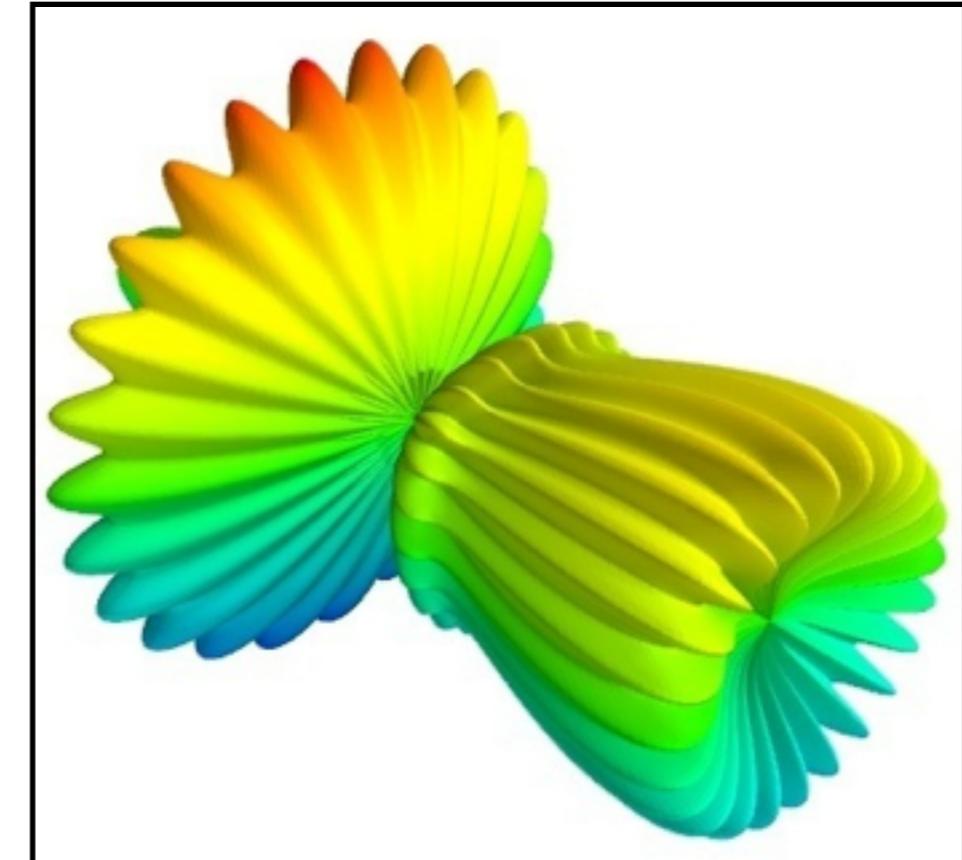
MayaVi / VTK



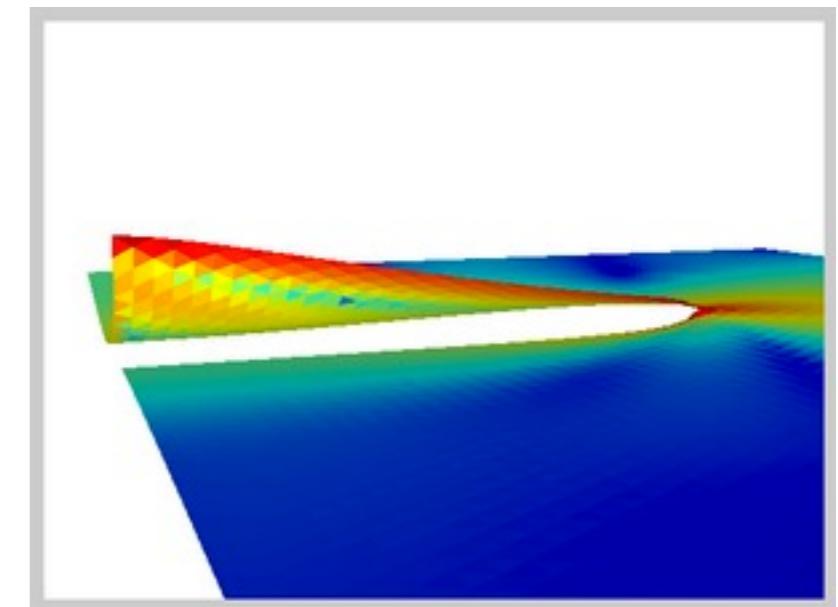
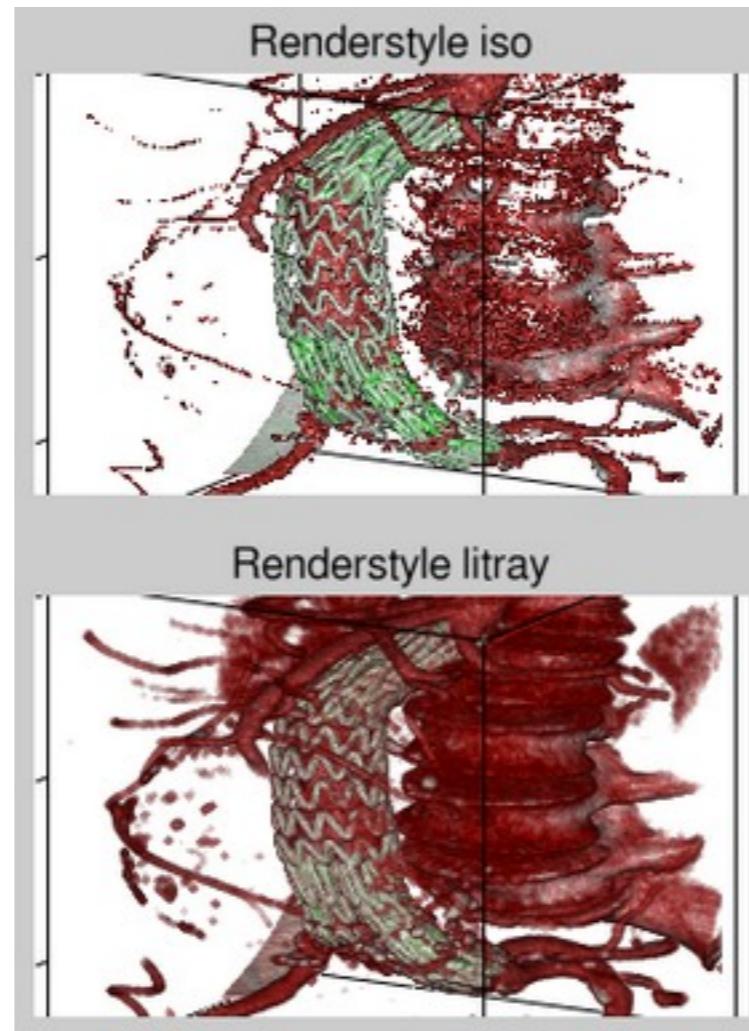
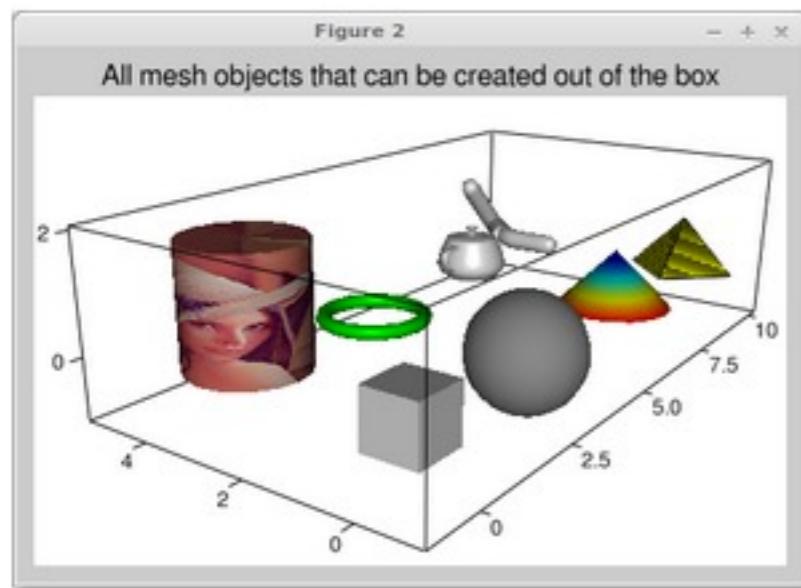
<http://code.enthought.com/projects/mayavi/>

mlab

```
# Create the data.  
from numpy import pi, sin, cos, mgrid  
dphi, dtheta = pi/250.0, pi/250.0  
[phi,theta] = mgrid[0:pi+dphi*1.5:dphi,0:2*pi  
+dtheta*1.5:dtheta]  
m0 = 4; m1 = 3; m2 = 2; m3 = 3; m4 = 6; m5 = 2; m6  
= 6; m7 = 4;  
r = sin(m0*phi)**m1 + cos(m2*phi)**m3 +  
sin(m4*theta)**m5 + cos(m6*theta)**m7  
x = r*sin(phi)*cos(theta)  
y = r*cos(phi)  
z = r*sin(phi)*sin(theta)  
  
# View it.  
from mayavi import mlab  
s = mlab.mesh(x, y, z)  
mlab.show()
```



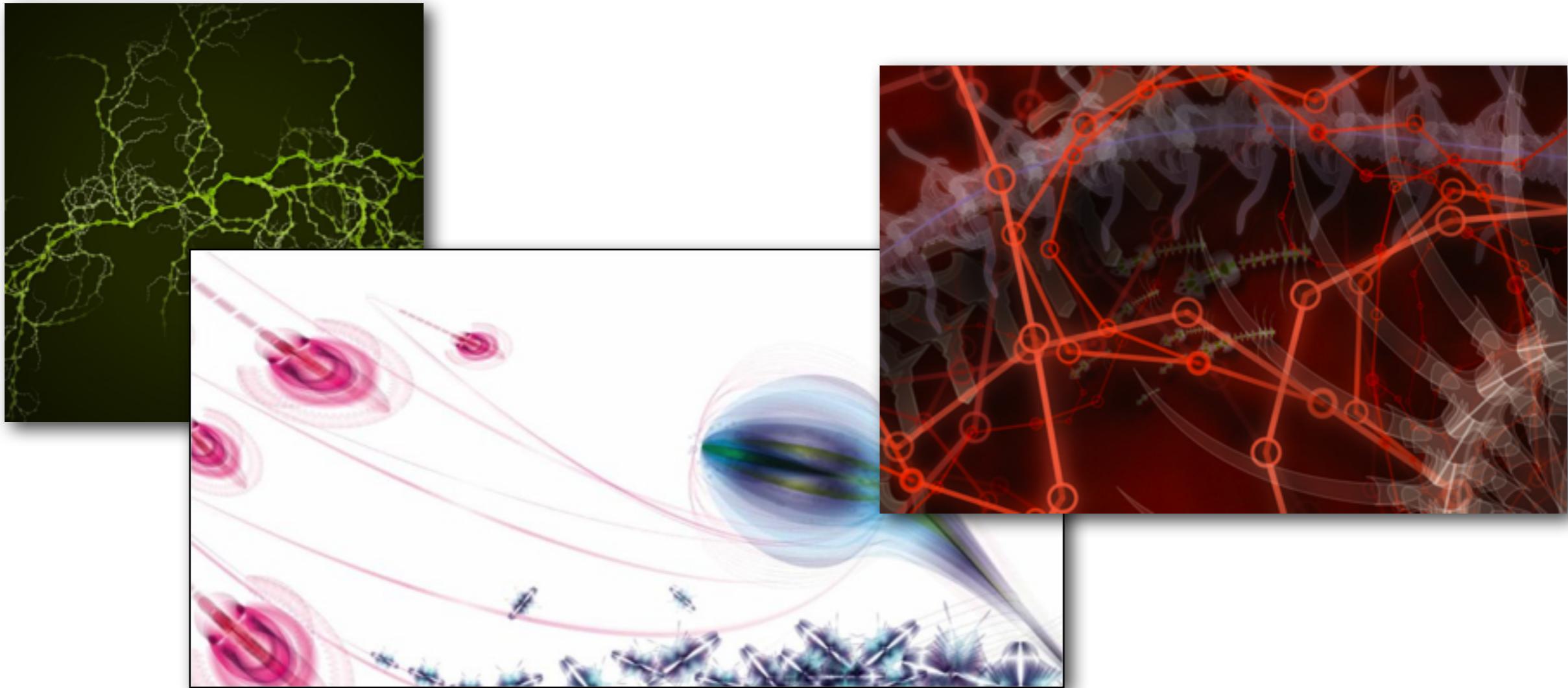
visvis



<https://code.google.com/p/visvis/>
<http://bit.ly/visvis-slides>

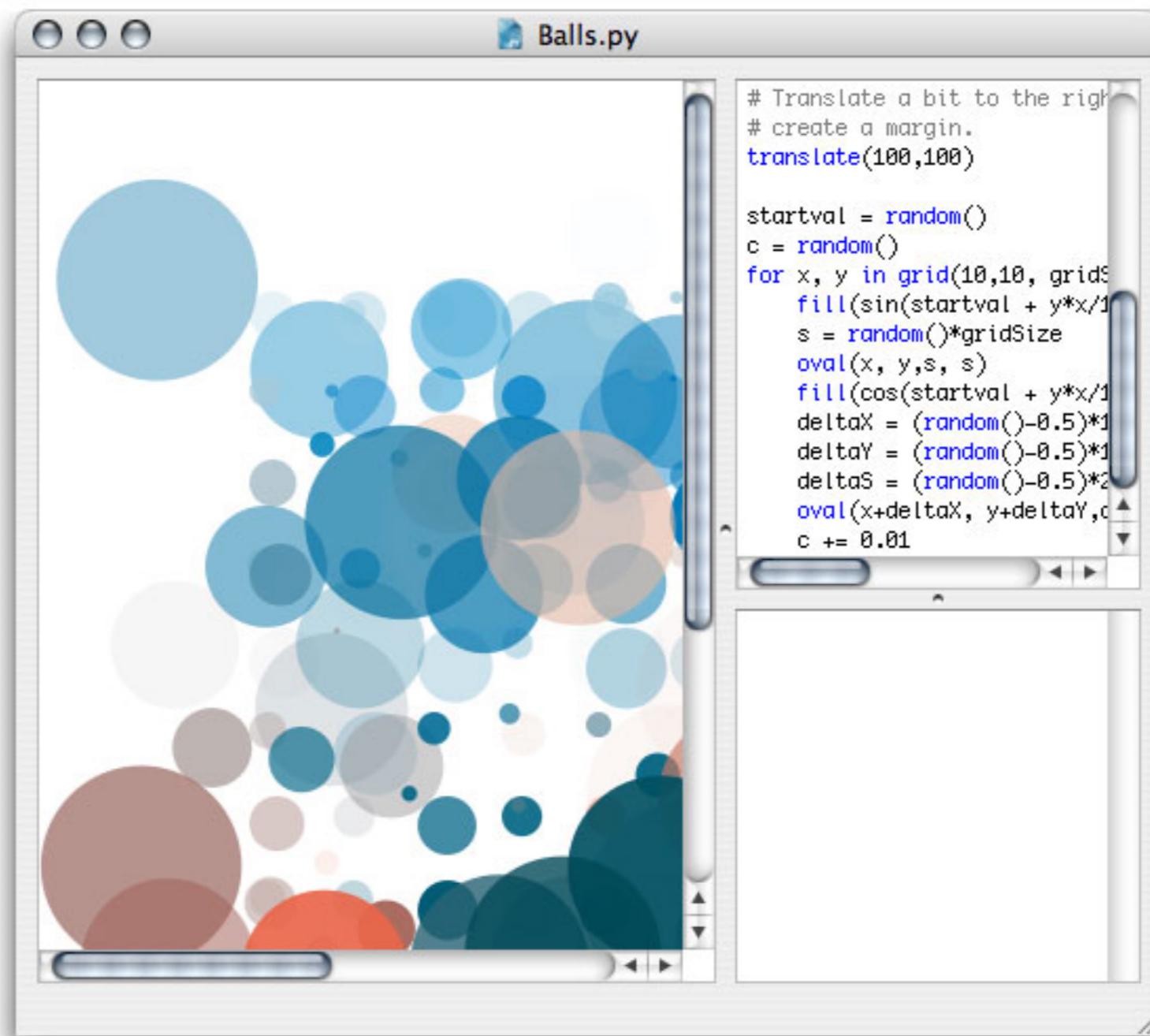
Showcase

Teaching / Education: Nodebox

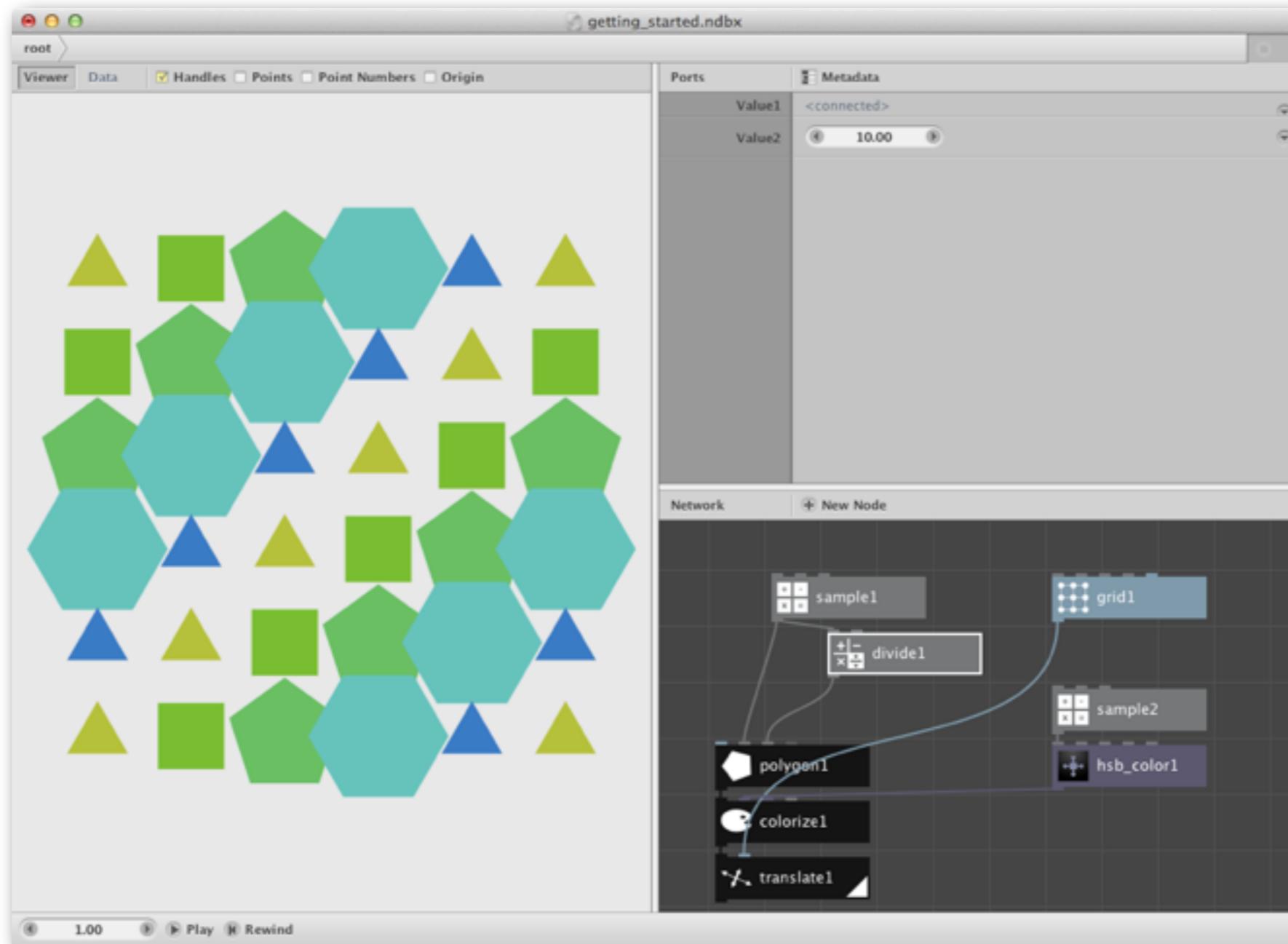


<http://nodebox.net/code/index.php/Home>

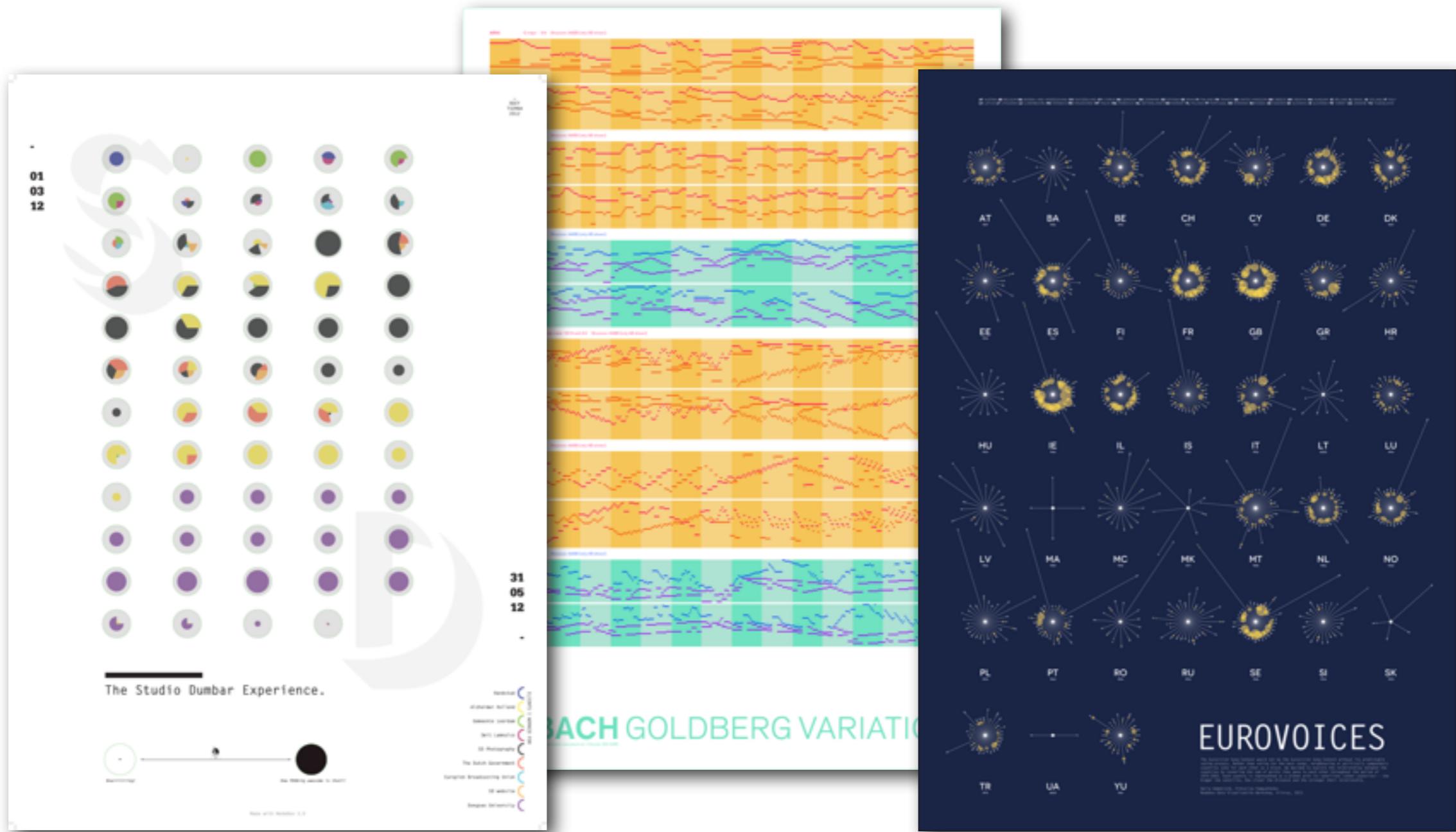
Teaching / Education: Nodebox



Teaching / Education: Nodebox 3



Teaching / Education: Nodebox 3



yt - Gorgeous Astrophysics

