Matplotlib

```

import pylab, numpy
```

Setting rcParams

Changes to the basic pylab settings. This can be done in the pylab.rcParams dict.

```
print pylab.rcParams['figure.figsize']
```

```
(8, 6)
```

```
print pylab.rcParams['text.usetex']
```

```
False
```

```
print pylab.rcParams['figure.dpi']
```

```
80
```

```
print pylab.rcParams['savefig.dpi']
```

```
100
```

```
print
```

Simple plotting

New figures

```
fig1 = pylab.figure()
```

```
fig2 = pylab.figure()
```

Activate a figure

```
pylab.figure(fig1.number)
```

Show and draw

```
# pylab.show()
# pylab.draw()
```

Clear a figure

```
pylab.clf()
```

Close figures

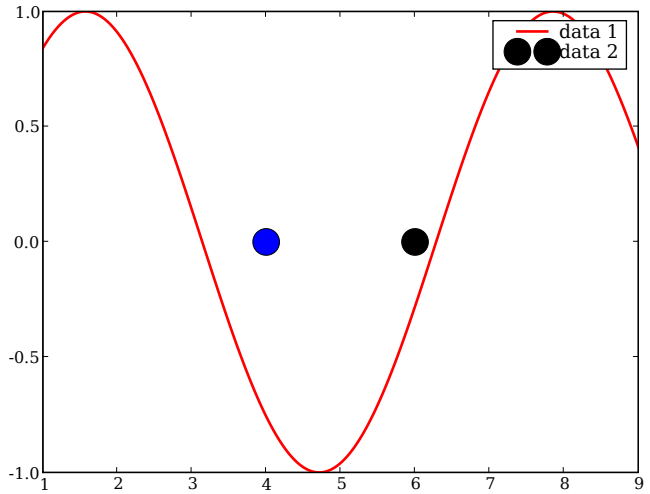
```
pylab.close()
pylab.close('all')
```

Interactive use

```
pylab.ion()
pylab.ioff()
```

Plotting functions

```
x = numpy.arange(0.0,10.,0.001)
y = numpy.sin(x)
pylab.plot(x,y, color='red', lw=2,label='data 1')
pylab.plot([4],[0],'ob',ms=20.,label='_nolegend_')
pylab.plot([6],[0],'ok',ms=20.,label='data 2')
pylab.xlim(1,9)
pylab.ylim(-1,1)
pylab.legend()
pylab.show()
```

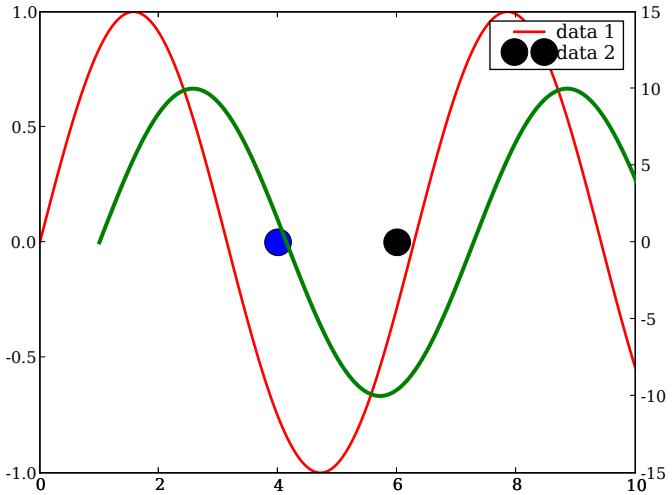



```
print
```

Second y-axis

```
59 pylab.twinx()
```

```
60 pylab.plot(x+1,y*10., color='green',lw=3,label='data 3')
61 pylab.ylim(-15,15.)
62 pylab.show()
```

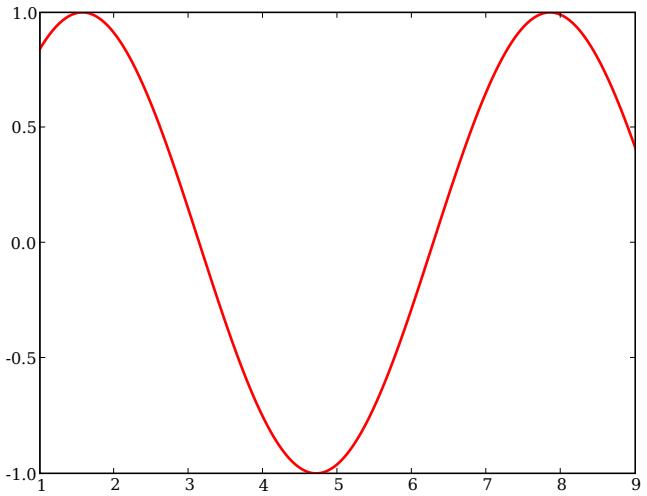



```
64 print
```

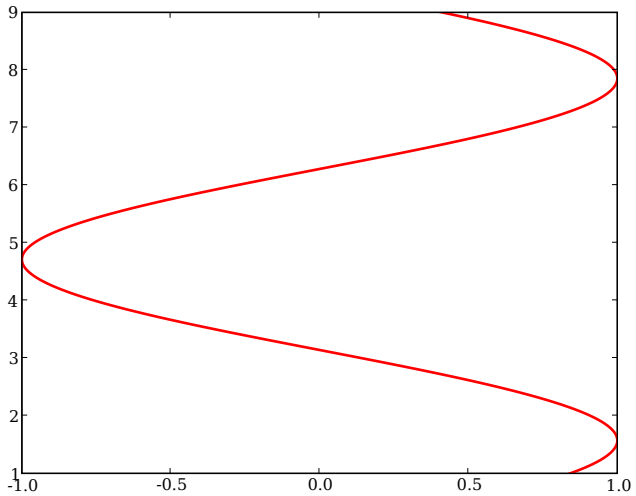
Exchange the data without creating a new figure

```
67 pylab.figure()
```

```
68 h, = pylab.plot(x,y, color='red', lw=2,label='data 1')
69 pylab.xlim(1,9)
70 pylab.ylim(-1,1)
71 pylab.show()
```




```
72 h.set_data(y,x)
73 pylab.xlim(-1,1)
74 pylab.ylim(1,9)
75 pylab.show()
```




```
77 print
```

Save figures to files

```
80 # Many formats are suported: png, pdf, ps, svg...
81 pylab.savefig('filename.png')
```

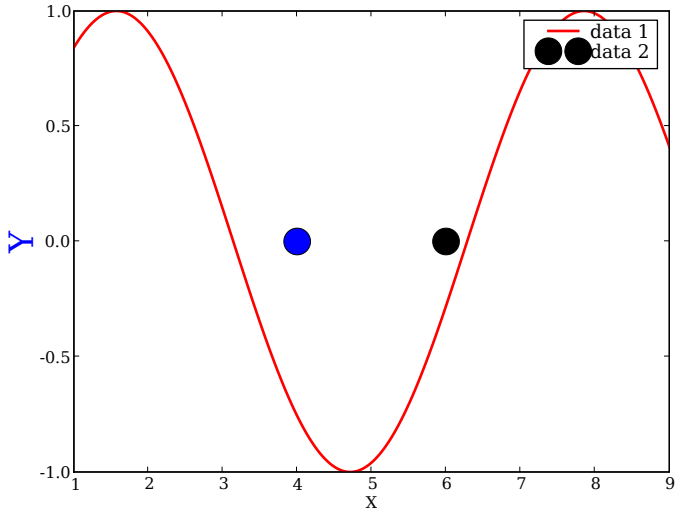
```
84 print
```

Changing labels, ticks, titles etc...

Labels

```
90 pylab.figure()
91 pylab.plot(x,y, color='red', lw=2,label='data 1')
92 pylab.plot([4],[0],'ob',ms=20.,label='_nolegend_')
93 pylab.plot([6],[0],'ok',ms=20.,label='data 2')
94 pylab.xlim(1,9)
95 pylab.ylim(-1,1)
96 pylab.legend()
97 pylab.xlabel('X',fontsize=12)
```

```
98 pylab.ylabel('Y',color='blue',fontsize=25)
99 pylab.show()
```

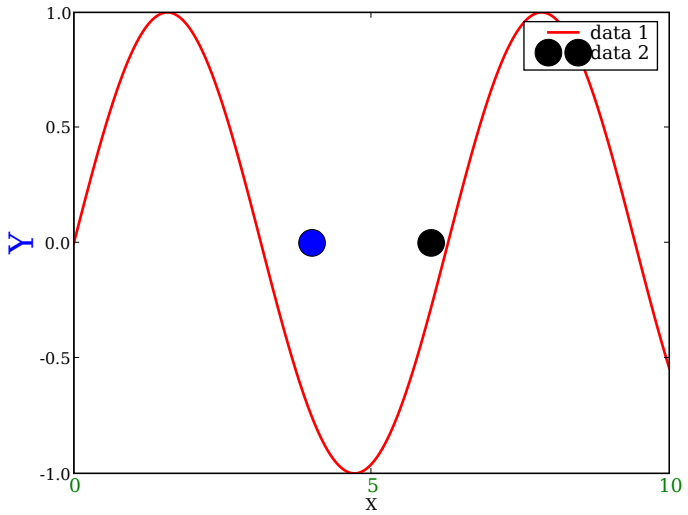



```
101 print
```

Ticks, tickslabels

```
104 xticks = numpy.linspace(0,10,3)
```

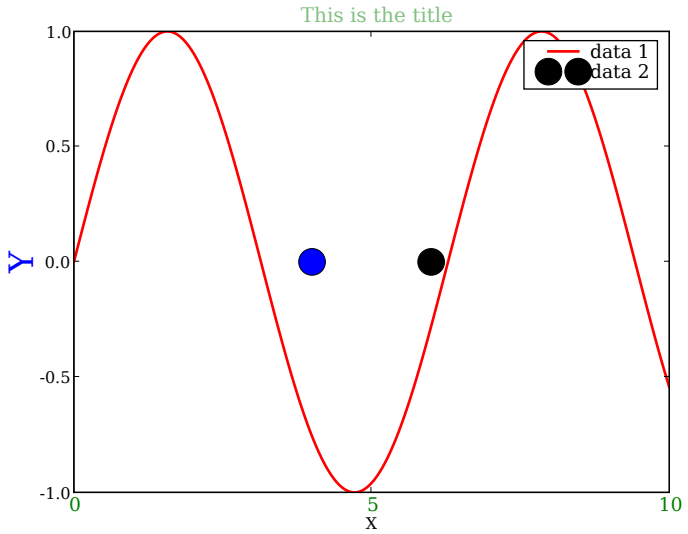
```
105 xticks_labels = ['%g'%i for i in xticks]
106 pylab.xticks(xticks,xticks_labels,fontsize=14,color='green')
107 pylab.show()
```




```
109 print
```

Title

```
112 pylab.title('This is the title',fontsize=15,color='green',alpha=0.5)
113 pylab.show()
```



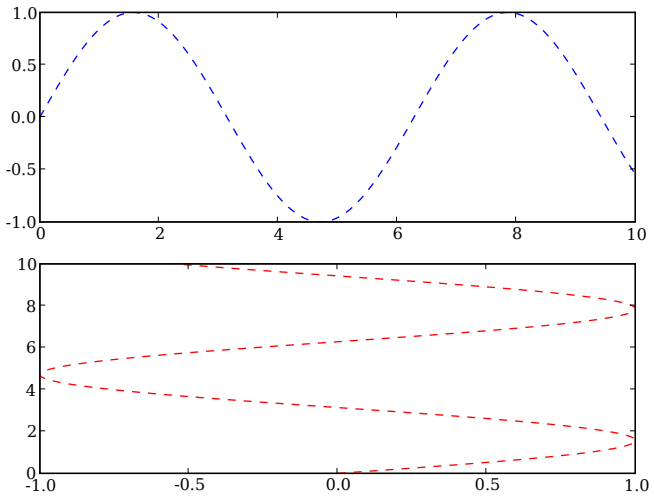

```
115 print
```

Multiple sub-plots

Using subplot

```
120 # pylab.subplots_adjust: Tune the subplot layout
121 pylab.figure()
```

```
123 ax = pylab.subplot(2,1,1)
124 pylab.plot(x,y,'b--')
125 ax = pylab.subplot(2,1,2)
126 pylab.plot(y,x,'r--')
127 pylab.show()
```




```
129 print
```

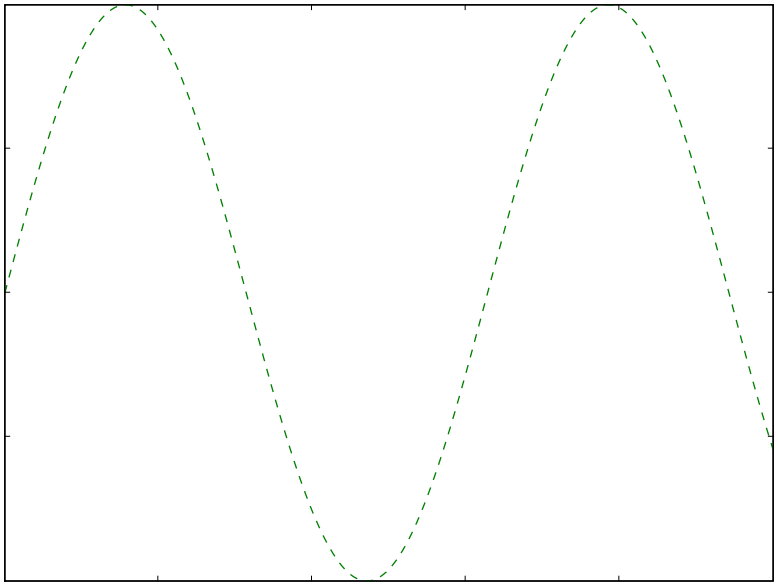
Using axes

```
133 pylab.figure()
```

Whole figure axes

```
136 ax_1 = pylab.axes([0,0,1,1])
```

```
137 pylab.plot(x,y,'g--')
138 pylab.show()
```

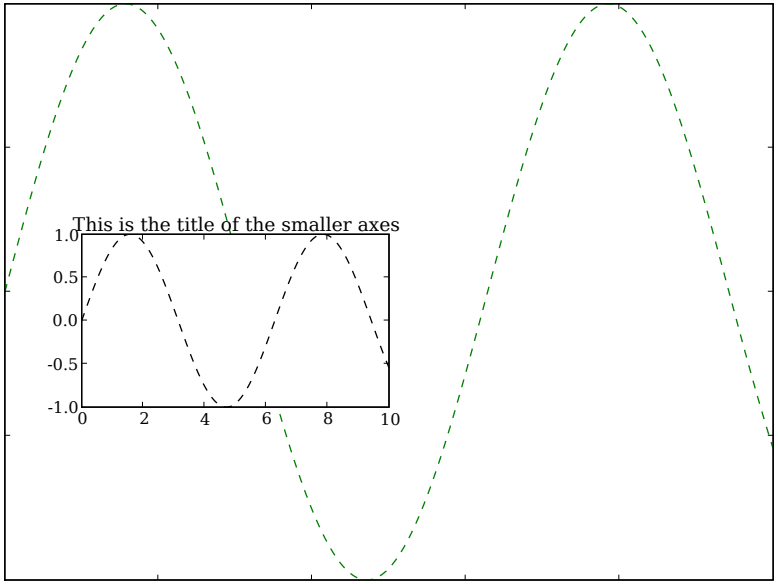


```
140 print
```

Smaller axes

```
143 l = 0.1;b = 0.3; w = 0.4; h = 0.3
```

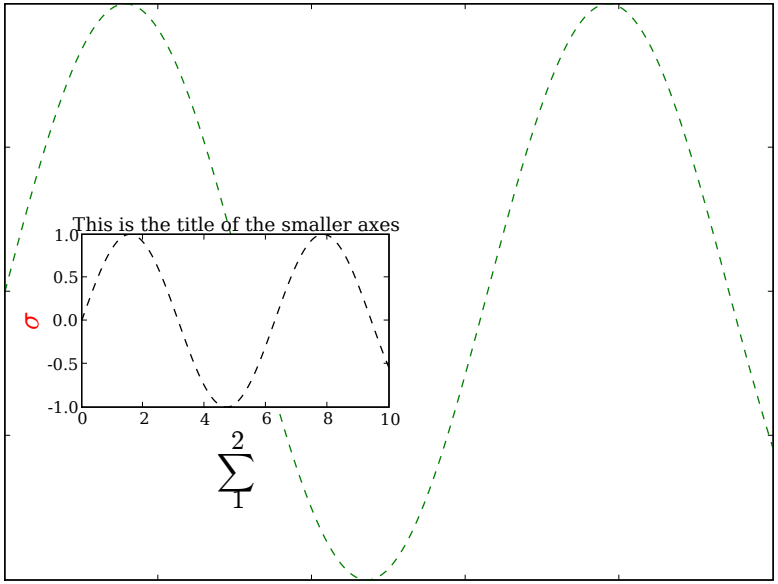
```
144 ax = pylab.axes([l,b,w,h])
145 pylab.plot(x,y,'k--')
146 pylab.title('This is the title of the smaller axes')
147 pylab.show()
```




```
149 print
```

Latex

```
152 pylab.xlabel(r"$\sum_1^2$",fontsize=20)
153 pylab.ylabel(r'$\sigma$',fontsize=23,color='red')
154
155 pylab.show()
```

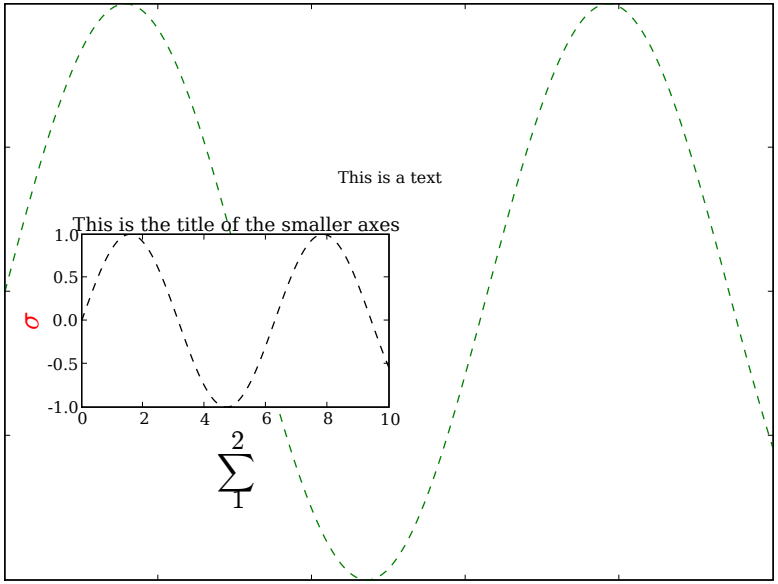



```
157 print
```

Text

```
164 pylab.text(0.5, 0.7,'This is a text',
165              horizontalalignment='center',
166              verticalalignment='center',
167              transform = ax_1.transAxes,
168          )
```

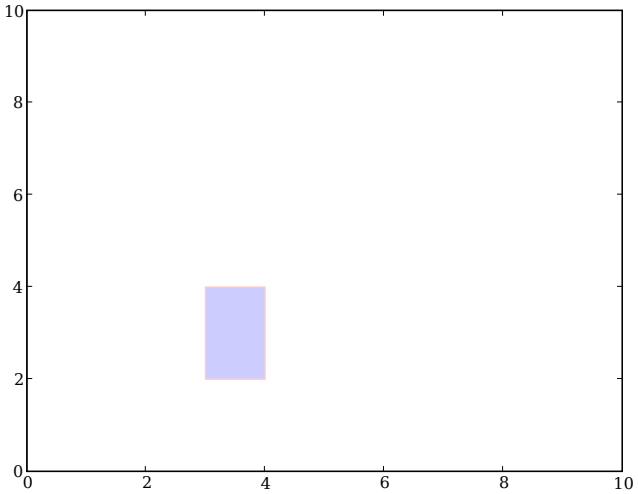
```
165 pylab.show()
```




```
167 print
```

Shaded Regions

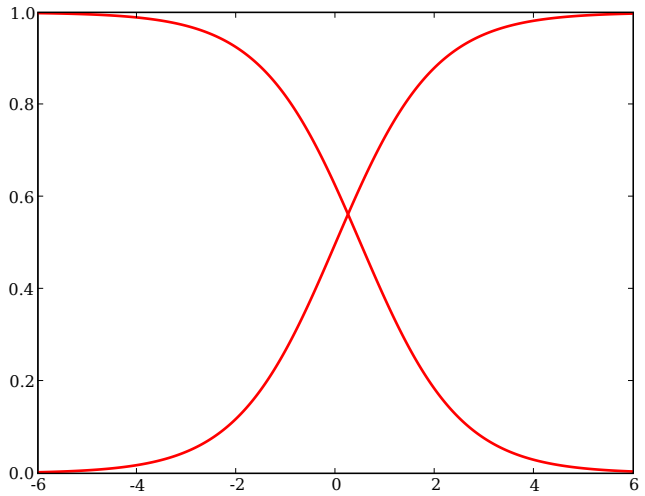
```
172 pylab.close('all')
173
174 # Make a blue box that is somewhat see-through
175 # and has a red border.
176 pylab.fill([3,4,4,3], [2,2,4,4], 'b', alpha=0.2, edgecolor='r')
177 pylab.xlim(0,10)
178 pylab.ylim(0,10)
179 pylab.show()
```




```
180 print
```

Fill below intersection

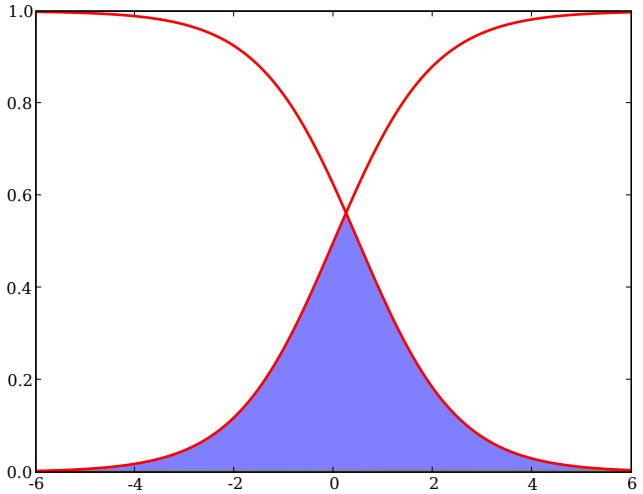
```
190
191 def boltzman(x, xmid, tau):
192     """
193     evaluate the boltzman function with midpoint xmid and time constant tau
194     over x
195     """
196     return 1. / (1. + numpy.exp(-(x-xmid)/tau))
197
198 pylab.clf()
199 x = numpy.arange(-6, 6, .01)
200 S = boltzman(x, 0, 1)
201 Z = 1-boltzman(x, 0.5, 1)
202 pylab.plot(x, S, x, Z, color='red', lw=2)
203 pylab.show()
```




```
198 print
```

```
200
201 pylab.clf()
202 def fill_below_intersection(x, S, Z):
203     """
204     fill the region below the intersection of S and Z
205     """
206     #find the intersection point
207     ind = numpy.nonzero( numpy.absolute(S-Z)==min(numpy.absolute(S-Z)))[0]
208     # compute a new curve which we will fill below
209     Y = numpy.zeros(S.shape, dtype=numpy.float)
210     Y[:ind] = S[:ind]  # Y is S up to the intersection
211     Y[ind:] = Z[ind:]  # and Z beyond it
212     pylab.fill(x, Y, facecolor='blue', alpha=0.5)
213
214 x = numpy.arange(-6, 6, .01)
215 S = boltzman(x, 0, 1)
216 Z = 1-boltzman(x, 0.5, 1)
217 pylab.plot(x, S, x, Z, color='red', lw=2)
218 fill_below_intersection(x, S, Z)
```

```
219 pylab.show()
```



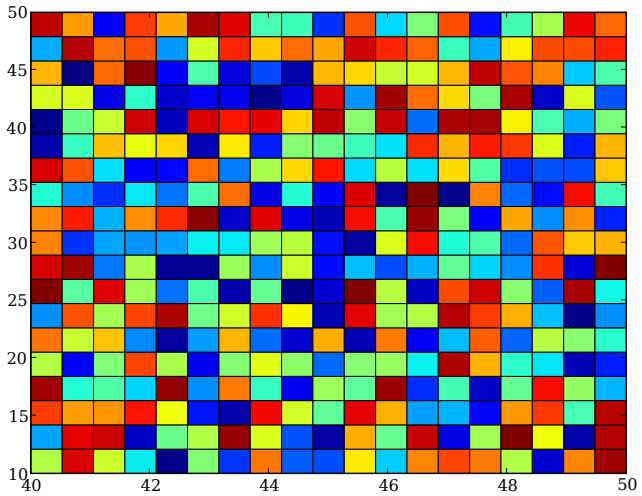

```
221 print
```

Colormaps

```
225 pylab.close('all')
226 data = numpy.random.uniform(size=(20,20))
```

With evenly spaced axes

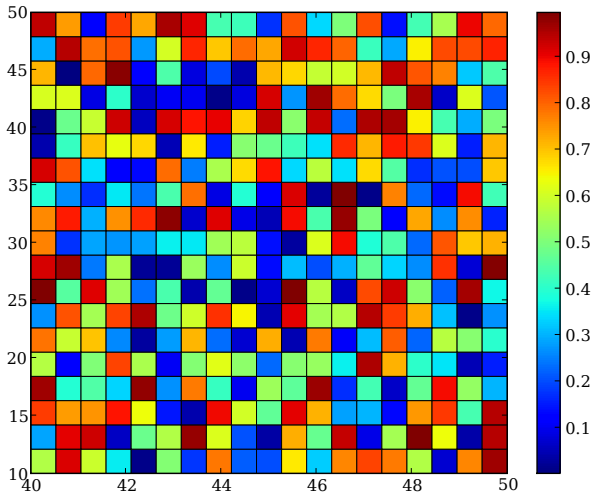
```
230 x = numpy.linspace(40,50,20)
231 y = numpy.linspace(10,50,20)
232 pylab.pcolor(x,y,data)
233 pylab.show()
```




```
235 print
```

Colorbar

```
237 pylab.colorbar()
238 pylab.show()
```

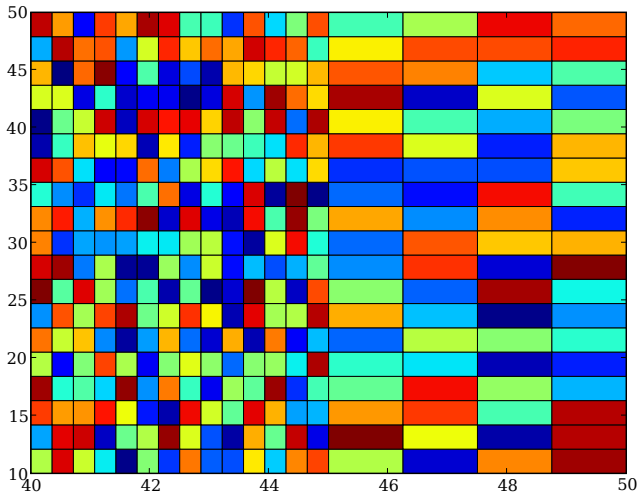



```
239 print
```

```
242
243 pylab.clf()
```

Variable axes

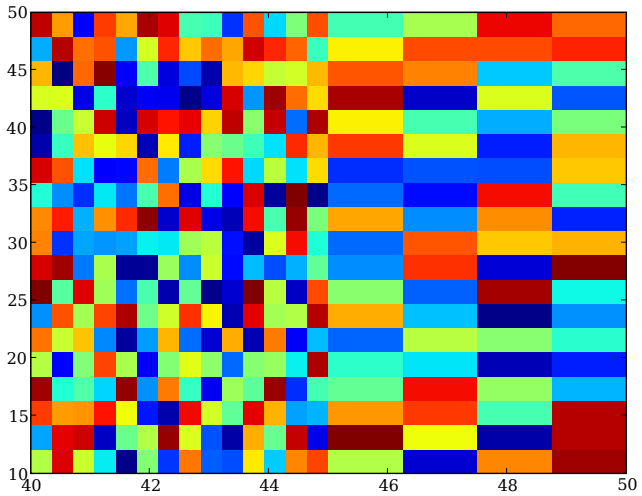
```
244 x = numpy.append(numpy.linspace(40,45,15),numpy.linspace(45,50,5))
245 pylab.pcolor(x,y,data)
246 pylab.show()
```




```
248 print
```

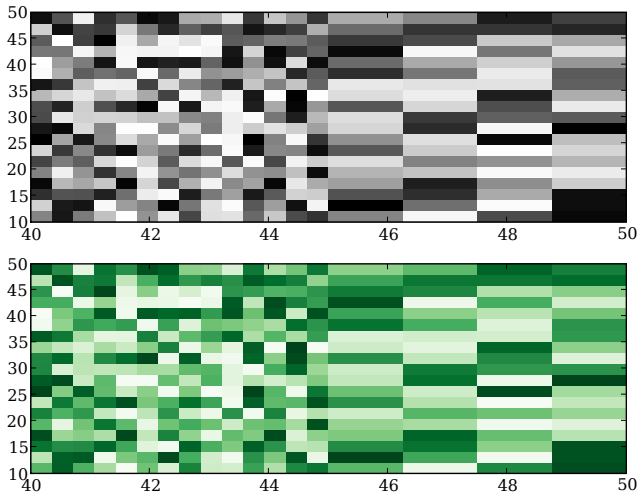
Flat shading

```
250 pylab.clf()
251 pylab.pcolor(x,y,data,shading='flat')
252 pylab.show()
```



Different colormaps in the same figure

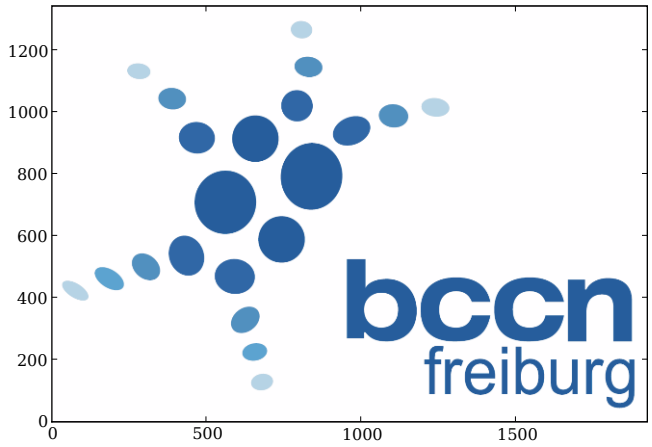
```
256 pylab.clf()
257 pylab.subplot(2,1,1)
258 pylab.pcolor(x,y,data,shading='flat',cmap=pylab.cm.Greys)
259
260 pylab.subplot(2,1,2)
261 pylab.pcolor(x,y,data,shading='flat',cmap=pylab.cm.Greens)
262
263 pylab.show()
```




```
265 print
```

Images

```
269 img = pylab.imread('bccn.png')
270 pylab.figure()
271 pylab.imshow(img)
272 pylab.show()
```




```
273 print
```

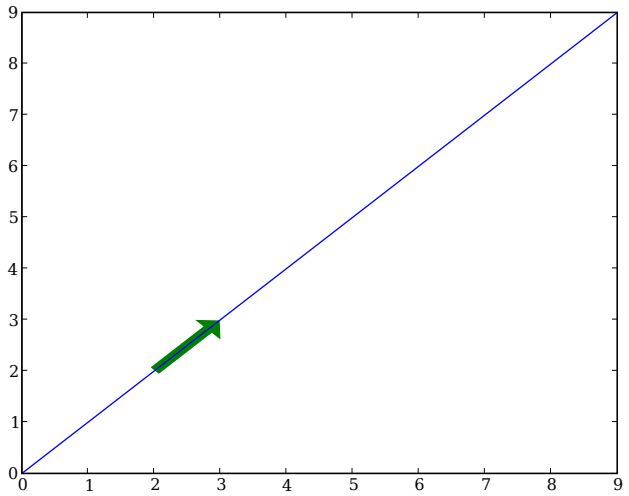
```
274 pylab.axis('off')
275 pylab.show()
```


```
277 print
```

Arrows

```
281 pylab.close('all')
282
283
284 x = numpy.arange(10)
285 y = x
286
287 # Plot line
288 pylab.plot(x, y)
289
290 # Now lets make an arrow object
291 arr = pylab.Arrow(2, 2, 1, 1, edgecolor='white')
292
293 # Get the subplot that we are currently working on
294 ax = pylab.gca()
295
296 # Now add the arrow
```

```
297 ax.add_patch(arr)
298
299 # We should be able to make modifications to the arrow.
300 # Lets make it green.
301 arr.set_facecolor('g')
302
303 pylab.show()
```

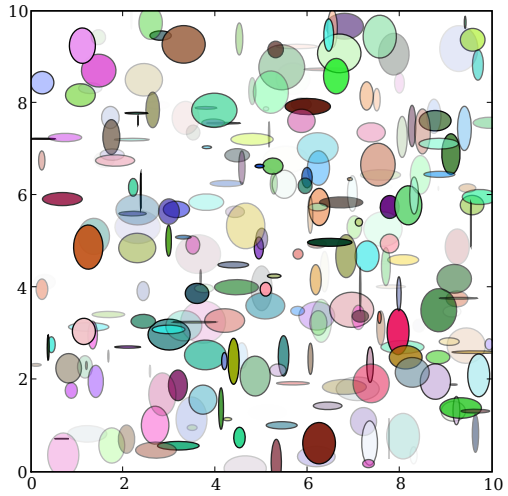



```
305 print
```

Elipses

```
310
311 pylab.close('all')
312 from matplotlib.patches import Ellipse
313
314 NUM = 250
315
316 ells = [Ellipse(xy=pylab.rand(2)*10, width=pylab.rand(), height=pylab.rand(), angle=pylab.rand()*360)
317         for i in xrange(NUM)]
```

```
318
319 fig = pylab.figure()
320 ax = fig.add_subplot(111, aspect='equal')
321 for e in ells:
322     ax.add_artist(e)
323     e.set_clip_box(ax.bbox)
324     e.set_alpha(pylab.rand())
325     e.set_facecolor(pylab.rand(3))
326
327 ax.set_xlim(0, 10)
328 ax.set_ylim(0, 10)
329
330 pylab.show()
```

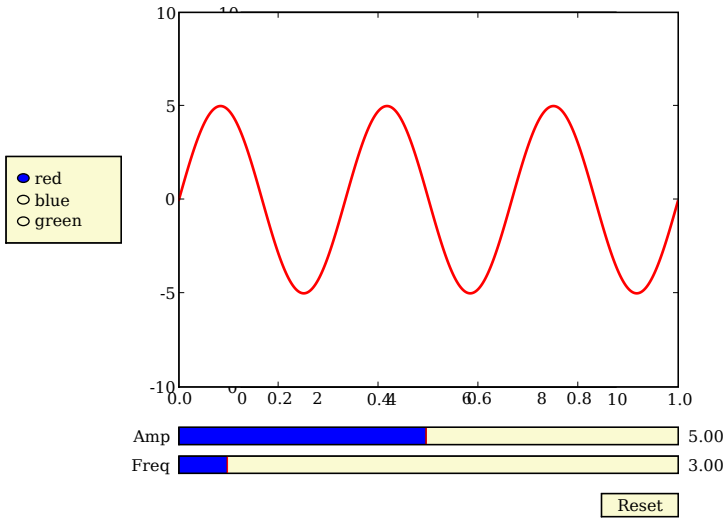



```
334 print
```

Interactive - Sliders

```
338 from matplotlib.widgets import Slider, Button, RadioButtons
339
```

```
340 ax = pylab.subplot(111)
341 pylab.subplots_adjust(left=0.25, bottom=0.25)
342 t = numpy.arange(0.0, 1.0, 0.001)
343 a0 = 5
344 f0 = 3
345 s = a0*numpy.sin(2*numpy.pi*f0*t)
346 l, = pylab.plot(t,s, lw=2, color='red')
347 pylab.axis([0, 1, -10, 10])
348
349 axcolor = 'lightgoldenrodyellow'
350 axfreq = pylab.axes([0.25, 0.1, 0.65, 0.03], axisbg=axcolor)
351 axamp  = pylab.axes([0.25, 0.15, 0.65, 0.03], axisbg=axcolor)
352
353 sfreq = Slider(axfreq, 'Freq', 0.1, 30.0, valinit=f0)
354 samp = Slider(axamp, 'Amp', 0.1, 10.0, valinit=a0)
355
356 def update(val):
357     amp = samp.val
358     freq = sfreq.val
359     l.set_ydata(amp*sin(2*pi*freq*t))
360     draw()
361 sfreq.on_changed(update)
362 samp.on_changed(update)
363
364 resetax = pylab.axes([0.8, 0.025, 0.1, 0.04])
365 button = Button(resetax, 'Reset', color=axcolor, hovercolor='0.975')
366 def reset(event):
367     sfreq.reset()
368     samp.reset()
369 button.on_clicked(reset)
370
371 rax = pylab.axes([0.025, 0.5, 0.15, 0.15], axisbg=axcolor)
372 radio = RadioButtons(rax, ('red', 'blue', 'green'), active=0)
373 def colorfunc(label):
374     l.set_color(label)
375     pylab.draw()
376 radio.on_clicked(colorfunc)
377
378 pylab.show()
```


red
blue
green
10
5
0
-5
-10
0.0
0.2
0.4
0.6
0.8
1.0
Amp
5.00
Freq
3.00
Reset