matplotlib

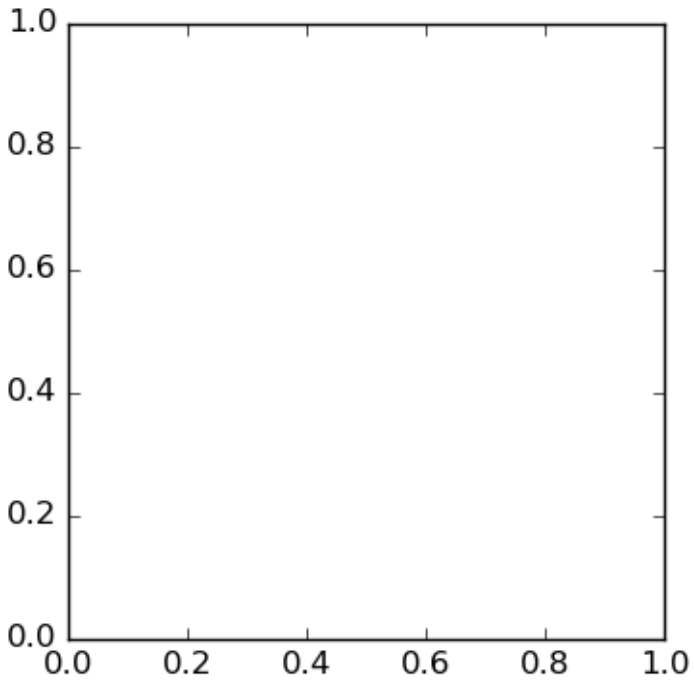
Ben Bolker

23:52 19 March 2015

- matplotlib cheat sheet
- matplotlib gallery

Basic setup

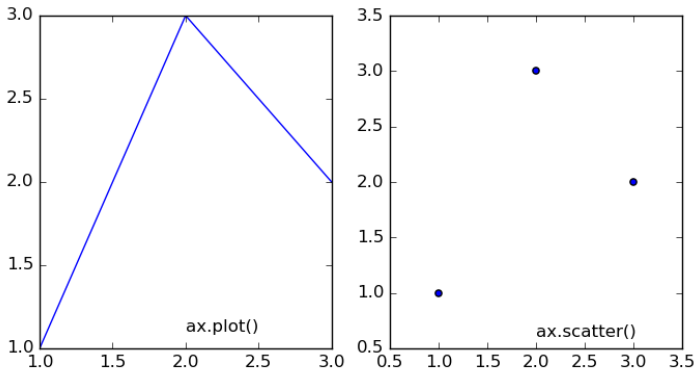
- create a figure: `fig=plt.figure()`
- can include figure size `figsize=(w,h)`, background/edge color, resolution (dpi=dots per inch)
- add a *subplot* (or "axes"): `ax = fig.add_subplot(1,1,1)` (rows,columns,which plot)
- now can show or save the figure: `fig.show()` or `fig.savefig("filename")`
- your operating system probably knows what to do if you click on the saved figure (or you can stick it in a Word document, etc.)



Basic plots

Basic things we can put on the plot: lines, scatter plots

Putting more than one thing on a plot



- You can do more than one `plot()` or `scatter()` on the same set of axes

Distinguish lines: * `color` * `marker` (`+`, `o`, `x`, ...) * `linewidth` * `linestyle` (`-`, `--`, `-.`, `None`, ...)

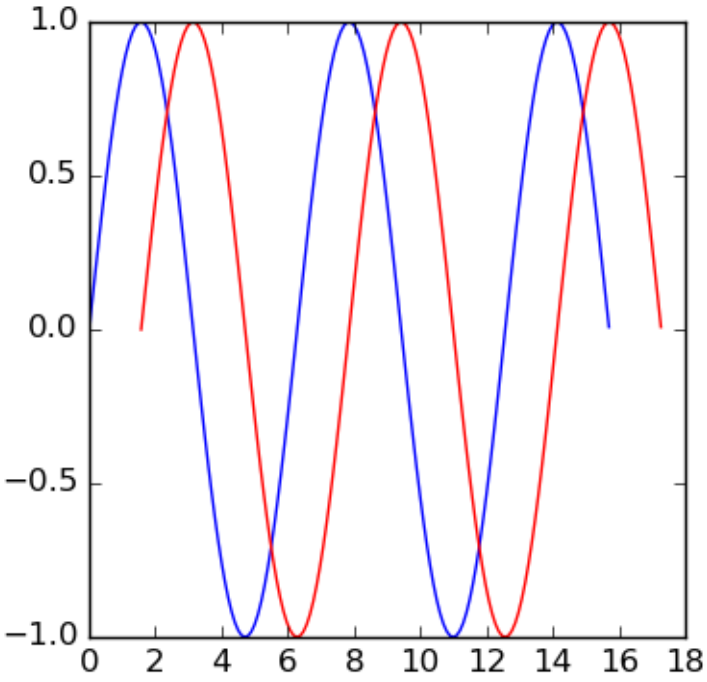
Decorating plots

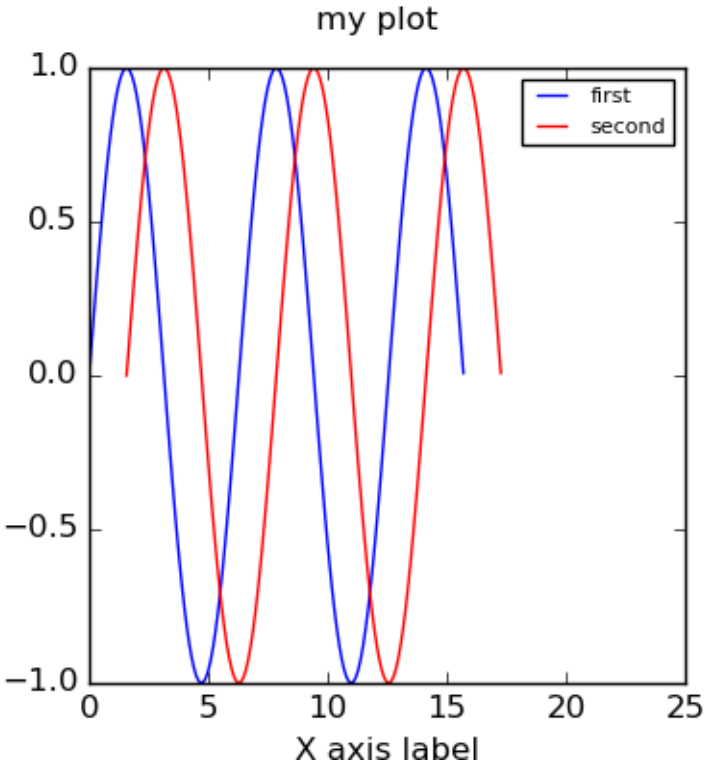
- titles (`ax.set_xlabel()`, `ax.set_ylabel()`)
- change limits
- title: `fig.suptitle()` (refers to figure, not individual axes)
- legend: need to label plotted stuff. e.g.

```
ax1.plot(x,y,label="first")
ax1.plot(x+np.pi/2,y,color="red",label="second")
ax1.set_xlim([0,25])
ax1.legend(fontsize=8)
fig.suptitle("my plot")
```

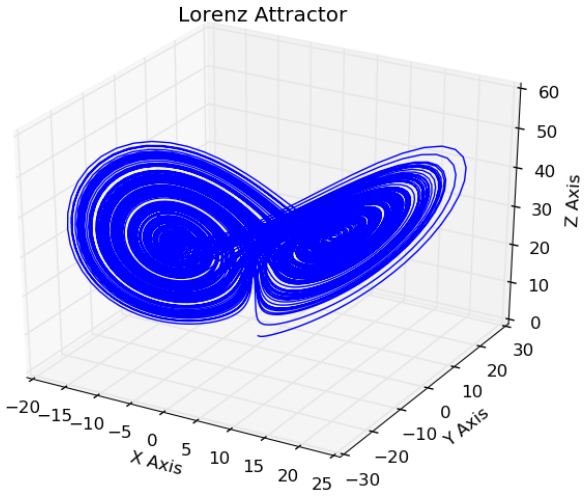
- Lorenz attractor example

```
import odesolve
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def lorenz(time, state, params):
    x, y, z = tuple(state)
    s, r, b = params
```


1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10
12
14
16
18


my plot
first
second
1.0
0.5
0.0
−0.5
−1.0
0
5
10
15
20
25
X axis label

```
    x_dot = s*(y - x)
    y_dot = r*x - y - x*z
    z_dot = x*y - b*z
    return((x_dot, y_dot, z_dot))

tvec = np.arange(0,200,0.01)
lfit = odesolve.solveODE3(lorenz,(0.,1.,1.05),tvec,(10,28,2.667))
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.plot(lfit[:,0], lfit[:,1], lfit[:,2])
ax.set_xlabel("X Axis")
ax.set_ylabel("Y Axis")
ax.set_zlabel("Z Axis")
ax.set_title("Lorenz Attractor")
fig.savefig("pix/lorenz.png")
```



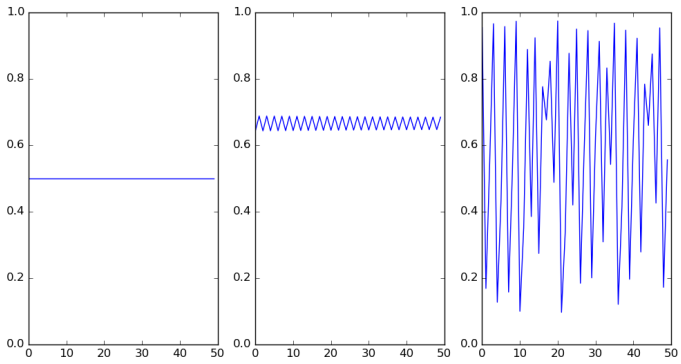
color maps reference

Logistic map

This was homework a while ago. Let's write it again now:

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

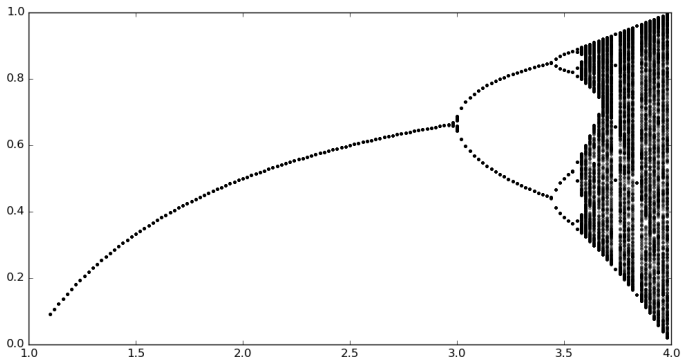
```
def logist_map(x0,r,t_trans,t_save):
    res = np.zeros(t_save)
    x = x0
    for i in range(t_trans):
        x = r*x*(1-x)
    for i in range(t_save):
        res[i] = x
        x = r*x*(1-x)
    return(res)
fig = plt.figure(figsize=(12,6))
ax1 = fig.add_subplot(1,3,1)
ax2 = fig.add_subplot(1,3,2)
ax3 = fig.add_subplot(1,3,3)
y1 = logist_map(0.5,2,100,50)
y2 = logist_map(0.5,3,100,50)
y3 = logist_map(0.5,3.9,100,50)
ax1.set_ylim(0,1)
ax2.set_ylim(0,1)
ax3.set_ylim(0,1)
ax1.plot(y1)
ax2.plot(y2)
ax3.plot(y3)
fig.savefig("pix/lm1.png")
```



Bifurcation diagram

```
import numpy as np
import matplotlib.pyplot as plt
import logist
```

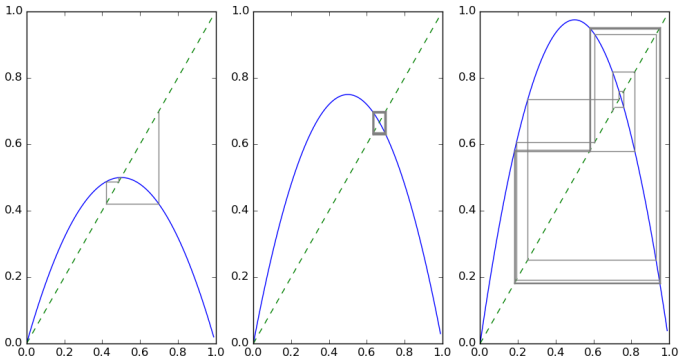
```
rvec = np.arange(1.1,4,0.02)
nr = len(rvec)
nt=600
res = np.zeros((nr,nt))
rvals = np.zeros((nr,nt))
for i in range(nr):
    rvals[i,:] = rvec[i]
    res[i,:] = logist.logist_map(0.5,rvec[i],100,nt)

fig = plt.figure(figsize=(12,6))
ax = fig.add_subplot(1,1,1)
ax.plot(rvals,res,alpha=0.4,color="black",linestyle="None",marker=".")
fig.savefig("pix/lmbif.png")
```



Cobweb diagram

```
import numpy as np
import matplotlib.pyplot as plt
import logist
def cobweb(r,ax,x0=0.7,t_save=20):
    cvec = np.arange(0,1,0.01)
    ax.plot(cvec,r*cvec*(1-cvec))
    ax.plot(cvec,cvec,linestyle="--")
    x = logist.logist_map(x0,r,0,t_save)
    for i in range(1,t_save):
        ax.plot([x[i-1],x[i-1]],[x[i-1],x[i]],color="gray")
        ax.plot([x[i-1],x[i]],[x[i],x[i]],color="gray")
    return(None)
fig = plt.figure(figsize=(12,6))
```

```
ax1 = fig.add_subplot(1,3,1)
ax2 = fig.add_subplot(1,3,2)
ax3 = fig.add_subplot(1,3,3)
cobweb(2.0,ax1)
cobweb(3.0,ax2)
cobweb(3.9,ax3)
fig.savefig("pix/cobweb.png")
```



to do

rearrange default/parameter order for convenience