

# 沙湖王

知识只有分享才能延续生命

## matplotlib绘图练习

博客最近一个月都很少更新了，导致访问量一直没有增加。本来想着到毕业要达到5万人的访问量，现在越看越悬了。matplotlib之前学完了一遍，但是现在估计又忘记很多了，很多东西就是要一边学一边练才能加深印象的。不过，我觉得，把学习的过程写成教程，也能加深印象。把Matplotlib for Python Developers这本书的例子都在我博客上讲解一遍之后，我下一步打算把R Graph Cookbook里面的例子用matplotlib重新实现一遍。希望7月份能把这份工作完成。

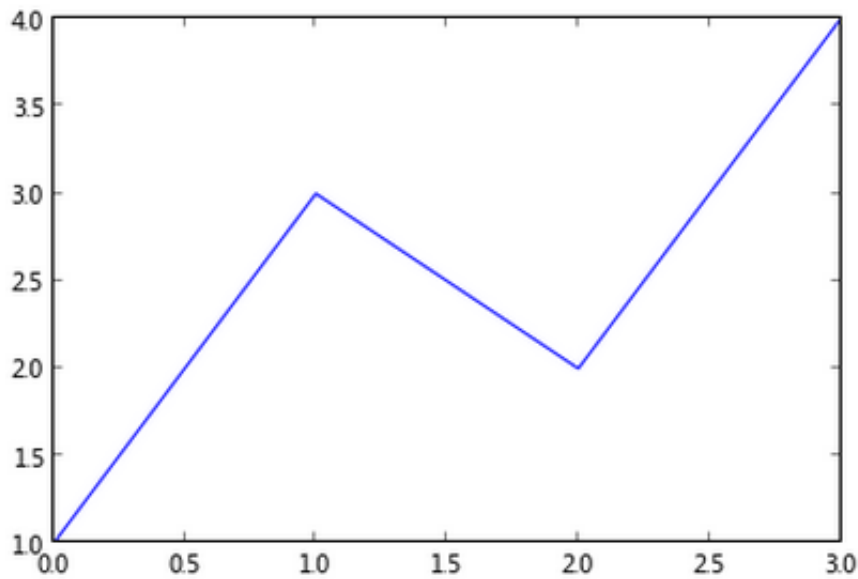
好了，现在开始进入matplotlib的世界。

首先说明一点，我下面的例子都是使用ipython notebook运行的，启动命令为ipython notebook --pylab inline.如果你不会安装notebook，可以参考我前面的博客。如果还是不会，可以直接只用ipython --pylab启动ipython。不过，强烈建议你用ipython notebook.

实际上，我上面的启动命令已经把numpy， scipy， matplotlib的命名空间都导入进来了，只是书上用的还是完整的导入，下面就照着书上的来吧，首先，画一个最简单的折线图，代码如下：

```
1 | import matplotlib.pyplot as plt
2 | plt.plot([1,3,2,4])
3 | plt.show()
```





如果想在一幅图中绘制多条线，该怎么办呢？matplotlib里怎么设置线的颜色呢？代码如下：

```
1 import matplotlib.pyplot as plt
2 x = range(1,5)
3 plt.plot(x,[xi*1.5 for xi in x])
4 plt.plot(x,[xi*3.0 for xi in x])
5 plt.plot(x,[xi/3.0 for xi in x])#matplotlib会自动给三条线
6 plt.show()#这里貌似可以用面向对象中的model来解释
7 #只需关系画上去，不用去关系它们是怎么组合的
```

注意到什么没？matplotlib会自动给三条线不同的颜色的哦。当然，也可以自己设置，不过那到后面再说。

如果你觉得上面的代码还稍显麻烦，可以用下面的代码作出一样的图来：

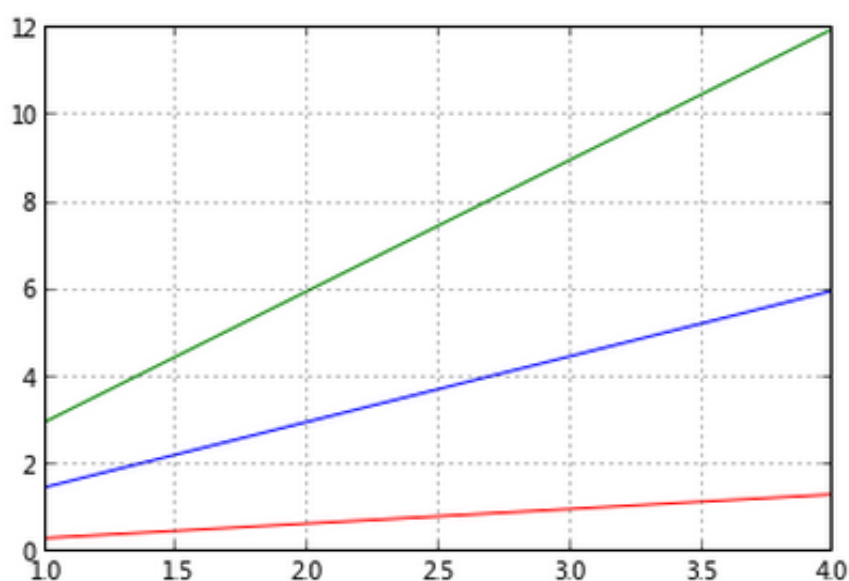
```
1 import matplotlib.pyplot as plt
2 x = range(1,5)
3 #这里和前面绘制出来的图是一样的，但是只是在plot参数里面，用了
4 #从而绘制出来多条线
5 plt.plot(x,[xi*1.5 for xi in x],x,[xi*3.0 for xi in x],x,
6          [xi/3.0 for xi in x])
7 plt.show()
```

图还是和上面那幅是一样的。

有时候，希望绘制出的图像，背景中能有格子，如何画呢？代码如下：

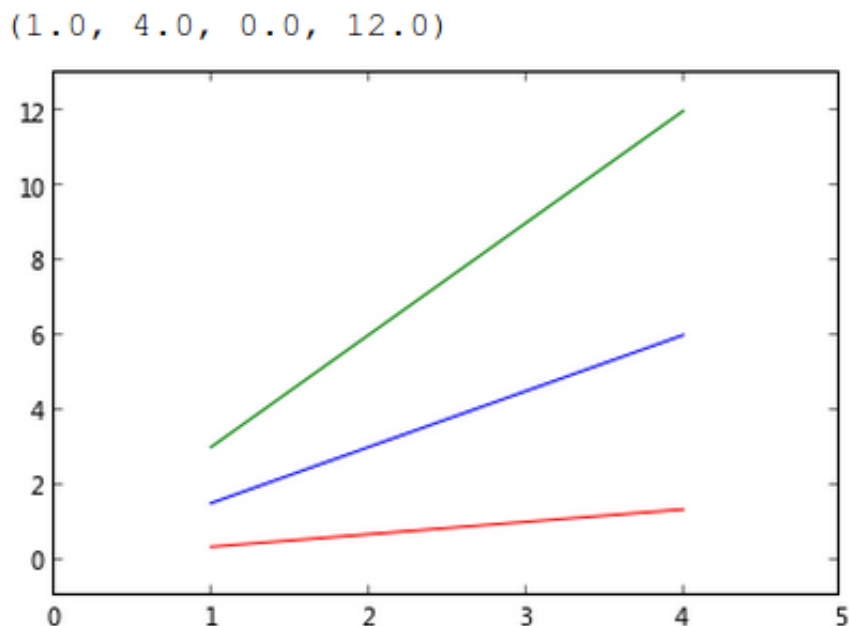
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(1,5)
4 plt.plot(x,x*1.5,x,x*3.0,x,x/3.0)
5 plt.grid(True)#显示格子
6 plt.show()
```

绘制出的图像如下图：



有时候需要设置图像绘制的区间，该怎么弄呢？代码如下：

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(1,5)
4 plt.plot(x,x*1.5,x,x*3.0,x,x/3.0)
5 print plt.axis()#这里是显示当前的坐标轴的取值范围的，由于我
6 #所以要加一个print才能显示出来，如果在shell模式下使用ipython
7 plt.axis([0,5,-1,13])#设置x轴区间(0,5),y轴区间(-1,13)
8 plt.show()
```



绘制出来的图像，就如上面那样了。不过，设置坐标轴区间的方式可不止上面一种，还有很多种，如下面的代码：

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(1,5)
4 plt.plot(x,x*1.5,x,x*3.0,x,x/3.0)
5 plt.axis(xmim=0,ymim = 0,ymax = 10)#这里通过名字来指定参数
6 plt.show()
```

上面的是一种，下面展示另一种：

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(1,5)
4 plt.plot(x,x*1.5,x,x*3.0,x,x/3.0)
5 plt.xlim(1,6)#xmin是设置单个的，而xlim则是设置x轴的区间的
6 plt.ylim(2,10)#设置y轴的区间，学习过R语言的话，就会很熟悉这
7 plt.show()
```

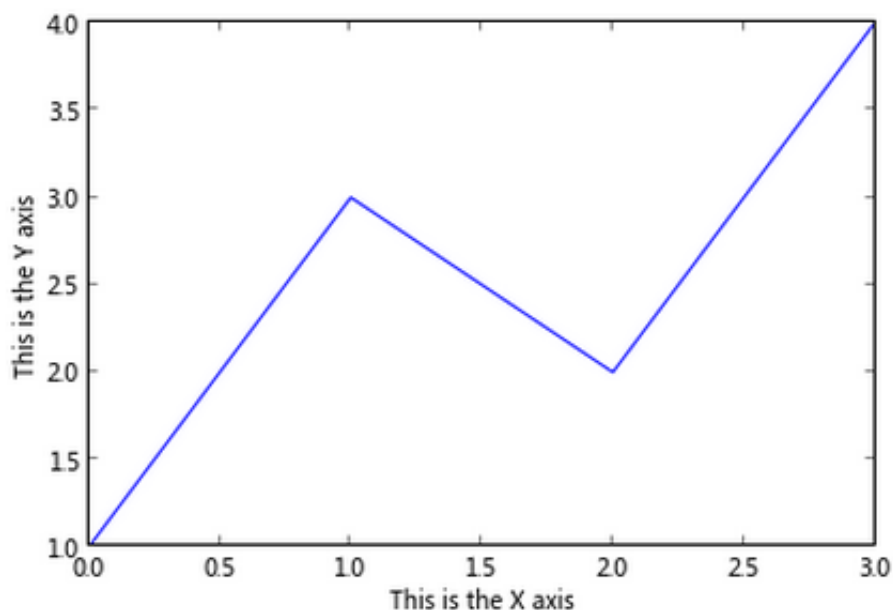
其实，这几种设置方式，很好的展示了Python多范式的特点，面向过程，面向对象都没有问题。

平时绘图，都会有图例的，说明这个x轴代表什么意思，y轴代表什么意思，

matplotlib怎么做到呢？代码如下：

```
1 import matplotlib.pyplot as plt
2 plt.plot([1,3,2,4])
3 plt.xlabel('This is the X axis')#设置x轴的说明
4 plt.ylabel('This is the Y axis')#设置y轴的说明
5 plt.show()
```

绘制出来的图像如下：

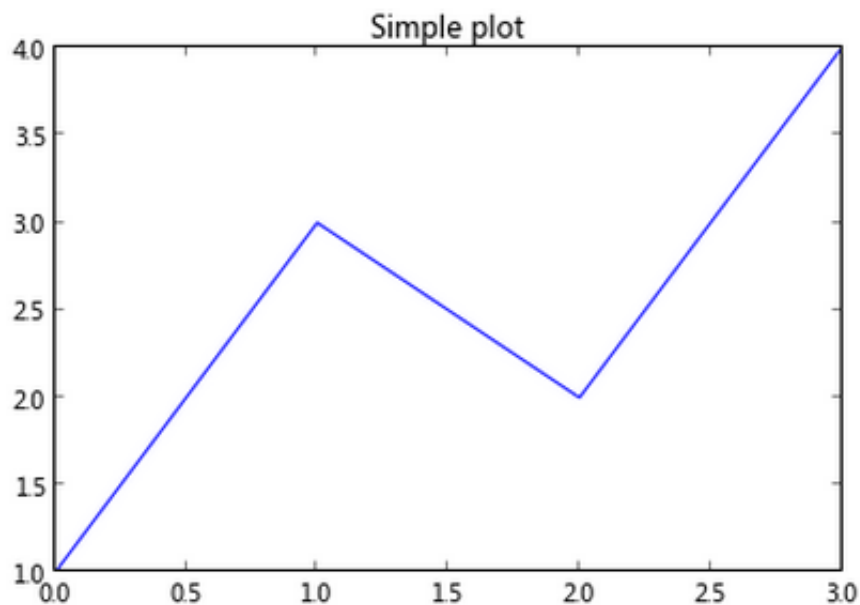


如果没有修改设置一下，matplotlib是不支持中文的，如果要支持中文，也可以找我前面写过的文章，按照我设置的方式，就能使用中文了。

下面展示如何设置图像的标题,代码如下：

```
1 import matplotlib.pyplot as plt
2 plt.plot([1,3,2,4])
3 plt.title('Simple plot')#设置标题，但是中文的话比较麻烦，，
4 #matplotlib不能支持中文显示，要进行修改才可以，可以参见我的博客
5 plt.show()
```

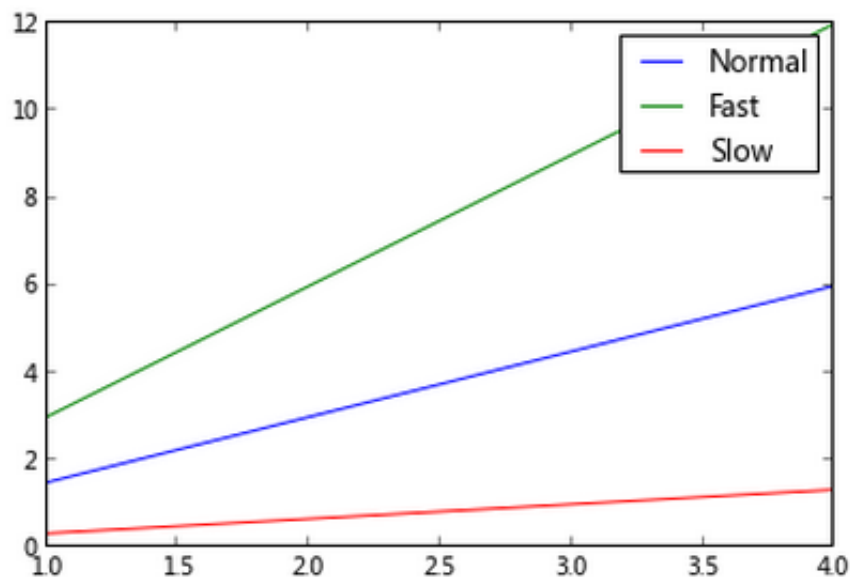
生成的图像如下图：



可以看到，顶部的标题已经设置好了。

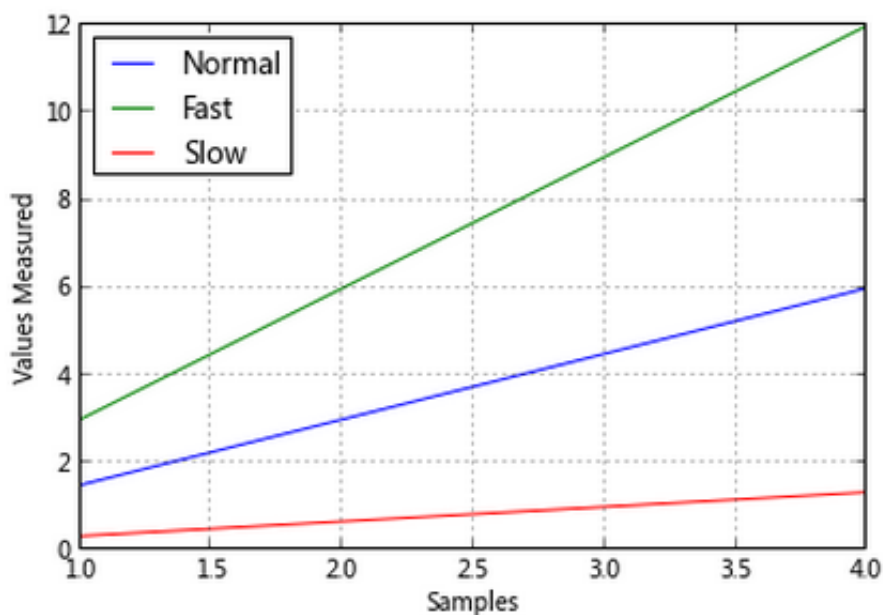
现在要说一下怎么设置图例了，所谓图例，就是那条颜色的线代表哪组数据的图例，代码如下：

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(1,5)
4 plt.hold(True)
5 plt.plot(x,x*1.5,label='Normal')#设置图例
6 plt.plot(x,x*3.0,label='Fast')
7 plt.plot(x,x/3.0,label='Slow')
8 plt.legend()#显示标签，或者说图例
9 plt.show()
```



如果要改变图例的位置，该怎么办呢？代码如下：

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 x = np.arange(1,5)
4 plt.plot(x,x*1.5,label="Normal")
5 plt.plot(x,x*3.0,label='Fast')
6 plt.plot(x,x/3.0,label='Slow')
7 plt.grid(True)
8 plt.xlabel('Samples')
9 plt.ylabel('Values Measured')
10 plt.legend(loc='upper left')#图例位置的设置，详情查看legend
11 plt.show()
```



要保存图片，怎么弄呢？很简单的，如下面的代码：

```
1 import matplotlib.pyplot as plt
2 plt.plot([1,2,3])
3 plt.savefig('plot123.png')#保存图片
```

这样就把图片保存在了当前工作目录下了。当然，还有更多的参数可以设置，看文档就可以了。

前面也说了，如果不自己设置颜色的话，matplotlib是会自动帮你设置颜色的。那要自己设置颜色，该怎么办呢？如下面的代码：

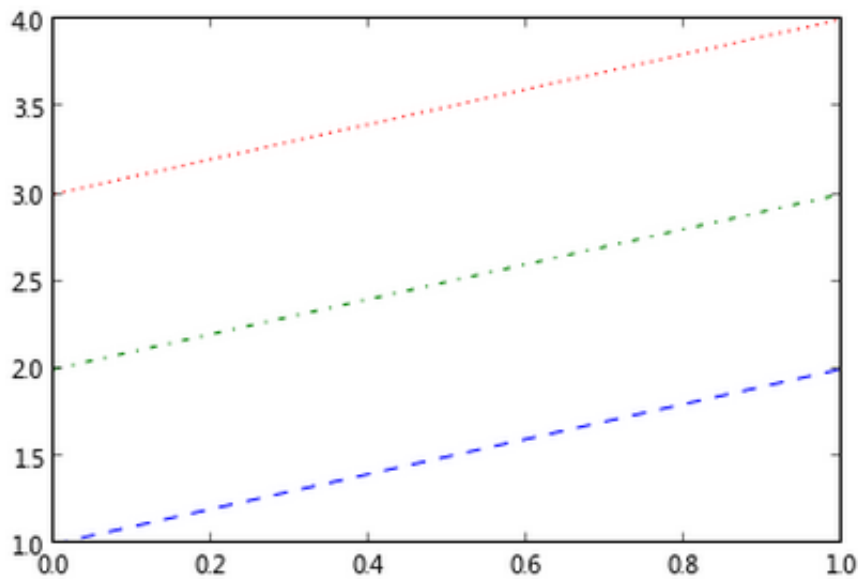
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 y = np.arange(1,3)
4 plt.plot(y,'y')#这些都是颜色的简写，但是我实在记不住，查看文档
5 #还是用调色器选取数值更方便一些
6 plt.plot(y+1,'m')
7 plt.plot(y+2,'c')
8 plt.show()
```

其中，'m'，'c'这些都是一些常用颜色的简写。

不仅可以设置颜色，还可以设置图形的形状，不过这里好像比较复杂，我也就了解了个大概，代码如下：

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 y = np.arange(1,3)
4 plt.plot(y,'--',y+1,'-.',y+2,':')#分别设置了线的形状，详情
5 #非常建议用ipython 的notebook，查看文档非常方便，http://www
6 plt.show()
```





这里还有几份代码是展示如何设置图线的形状的，图像我就不上传了，代码如下：

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 y = np.arange(1,3,0.2)
4 plt.plot(y, 'x', y+0.5, 'o', y+1, 'D', y+1.5, '^', y+2, 's')#设置
5 plt.show()
```

第二份如下：

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 y = np.arange(1,3,0.3)
4 plt.plot(y, 'cx--', y+1, 'mo:', y+2, 'kp-.')#设置点的形状和线,
5 plt.show()
```

第三份如下：

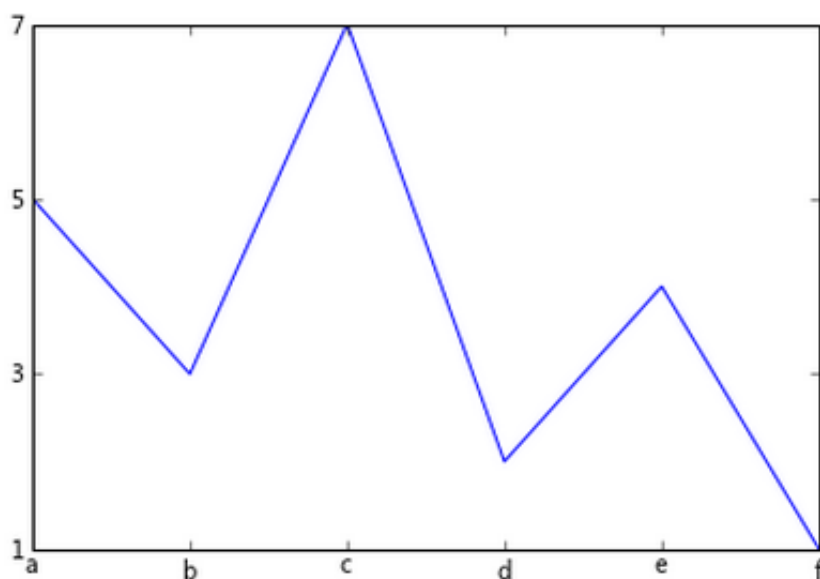
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 y = np.arange(1,3,0.3)
4 plt.plot(y, color = 'blue', linestyle='dashdot', linewidth=
5         #linewidth是设置线的宽度
6         marker='o', markerfacecolor='red', markeredgecolor=
7         #markeredgecolor设置点周围的颜色
8         markeredgewidth=3, markersize=12)#设置边缘的宽度,
```

```
9 | plt.show()
```

现在我们要设置坐标轴的刻度怎么办？代码如下：

```
1 | import matplotlib.pyplot as plt
2 | x = [5,3,7,2,4,1]
3 | plt.plot(x)
4 | plt.xticks(range(len(x)),['a','b','c','d','e','f'])#通过
5 | plt.yticks(range(1,8,2))
6 | plt.show()
```

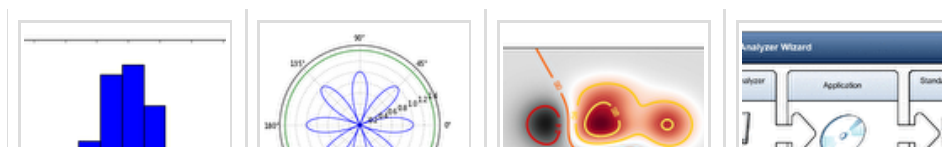
就是用了xticks和yticks这两个函数，具体参数的设置可以看文档,获得的图像如下：

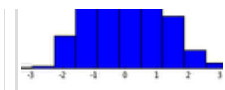


x轴坐标现在刻度是英文的了。

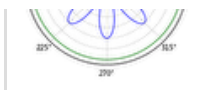
文章一长，写起来就麻烦了，就先写到这吧。后面会继续努力的。

您可能也喜欢：





matplotlib绘图2



matplotlib绘图3



matplotlib简介（译自matplotlib文档）



matplotlib的用户向导翻译：介绍篇

无关联推荐[?]

0

## Related Posts:

1. [matplotlib绘图2](#)
2. [matplotlib绘图3](#)
3. [继续学习Sage](#)
4. [matplotlib简介（译自matplotlib文档）](#)

标签: [ipython](#), [matplotlib](#), [notebook](#), [Python](#), [绘图](#)

分类: [Python](#), [科学计算](#)

Posted on 2012 年 6 月 4 日 by rickey

---

[Decode](#) by Scott Smith