

# Miller's ihome

One number,one world! One heart, one world!

## python中matplotlib绘制图形

发表于七月 11, 2013

**matplotlib** 是python最著名的绘图库，它提供了一整套和matlab相似的命令API，十分适合交互式地进行制图。该包下面有很多对象，比如pylab，pyplot等等，pylab集成了pyplot和numpy两个模块，能够快速绘图。PyLab和pyplot([http://matplotlib.org/api/pyplot\\_api.html](http://matplotlib.org/api/pyplot_api.html))都能够通过对象或者属性对图像进行操作。Pyplot下也有很多对象，如figure，Axes对象等等，对图像进行细节处理。下面通过一些简单和常用的例子进行操作。

1. 通过pyplot画散点图，调用scatter函数，同时该函数也可以画不同大小点的散点图。例子如下：

```
# *.coding : utf-8 *.
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0,20,2)
y = np.linspace(0,20,10)
plt.figure()
plt.scatter(x,y,c='r',marker='*')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('$X*Y$')
plt.show()
```

2. 画曲线图，通过plot函数画曲线图，如下：

```
import numpy as np
import random
from matplotlib import pylab as pl
from matplotlib import pyplot as plt

#画一条曲线图
plt.figure(figsize=(8,6)) #设置图形的尺寸，单位为英尺
x = np.random.randn(10)*10
y = np.pi*np.sin(x) + 8
x1 = np.arange(0,10,1)
#lines = plt.plot(x,y)
plt.plot(x,y,label = '$np.pi*np.sin(x)+8$',color='red',linewidth = 2)plt
plt.title('Y=a*sin(x)+b')
plt.ylabel('Function-Y')
plt.xlabel('Var-X')
plt.tight_layout() #图形的左上角显示图形的标签，和上面的plt.savefig('label=11')有关
```

```
plt.legend() #图形的右上用显示图形的标签,和上面的plt.plot(x,y,label='')有大
#plt.setp(lines,color = 'r',linewidth = 2.5)

plt.show()
```

3. 将图像画在多个轴上面,并在一个figure中。可以通过subplot, 还有ubplot2grid等等方式均可实现。

```
import numpy as np
import matplotlib.pyplot as plt
'''将图形画在一个矩阵制定的表中, 利用subplot, plot()中放有多个自变量和函数,画多坐标图
def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)
t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)
plt.figure(1)
plt.subplot(211) #subplot(numRows, numCols, plotNum),如: subplot(221)
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')
plt.title('pylab first example',fontsize = 16)
plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
plt.title('pylab second example',fontsize = 16)

plt.show()
```

4. 使用text将文本表现在图中,并对图像之间进行设置,坐标设置等。

```
# -*- coding: utf-8 -*-
import matplotlib.pyplot as plt
#通过对对象的形式文本作图
fig = plt.figure()
fig.suptitle('bold figure suprtitle', fontsize=14, fontweight='bold')#fig
ax = fig.add_subplot(111) #figure下的坐标图ax
fig.subplots_adjust(top=0.85)
ax.set_title('axes title') #给ax设置标题
ax.set_xlabel('xlabel') #给ax设置横坐标
ax.set_ylabel('ylabel') #给ax设置纵坐标
ax.text(3, 8, 'boxed italics text in data coords', style='italic',
        bbox={'facecolor':'red', 'alpha':0.5, 'pad':10})
ax.text(2, 6, r'an equation:  $E=mc^2$ ', fontsize=15)
ax.text(3, 2, unicode('unicode: Institut f\374r Festk\366rperphysik', 'l
ax.text(0.95, 0.01, 'colored text in axes coords',
        verticalalignment='bottom', horizontalalignment='right',
        transform=ax.transAxes,
        color='green', fontsize=15)
ax.plot([2], [1], 'o') #ax画图, 以及下面添加注释
ax.annotate('annotate', xy=(2, 1), xytext=(3, 4),
            arrowprops=dict(facecolor='black', shrink=0.05))
ax.axis([0, 10, 0, 10]) #设置横-纵轴的刻度
plt.show()
```

5. 通过hist画直方图

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0,1000,0.5)
y = np.random.rand(2000)*10
n,bins,patches = plt.hist(y,50,normed=1,alpha=0.8)
plt.title('Hist of Y')
plt.xlabel('Smarts')
plt.ylabel('NormProbability')
plt.text(2,0.12,'$\mu=10, \sigma=20$')
plt.grid(True)
plt.show()
```

## 6. 通过bar画条形图

```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0,1000,0.5)
y = np.random.rand(2000)*10
n,bins,patches = plt.hist(y,50,normed=1,alpha=0.8)
plt.title('Hist of Y')
plt.xlabel('Smarts')
plt.ylabel('NormProbability')
plt.text(2,0.12,'$\mu=10, \sigma=20$')
plt.grid(True)
plt.show()
```

## 7. 对图形中的重要点进行注释，采用annotate实现该功能。具体用法可以采用官方文档。如下：

```
# -*- coding : utf-8 -*-
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(-10,10,1)
y = 2*(x**2)-5*x+np.pi
maxy = min(y)
maxyx = x[np.argmin(y)]
plt.figure()
plt.plot(x,y,'-',color='red',linewidth=2.5,label='$2*(x^2)-5*x+np.pi$')
plt.annotate('Max y at points',xy=(maxyx,maxy),xytext=(maxyx+1,maxy+10),
            arrowprops=dict(facecolor='black', shrink=0.01),fontsize=12)
#xy为注释的点, xytext注释点文本的位置
plt.xlabel('X')
plt.ylabel('Y')
plt.xlim(-11,11)
plt.title('plot of y=x^2')

plt.show()
```

matplotlib官方文档中包含可很多画图的方法，如时间序列方面的处理，如：acorr；图像的处理，信号处理，数学公式等等。官方文档（<http://matplotlib.org/py-modindex.html>）

相关

python中最小二乘法  
([scipy.optimize](#))  
八月 7, 2013  
In "Python"

关于Python装饰器的@学习  
十一月 8, 2013  
In "Python"

python笔记之numpy模块  
二月 26, 2013  
In "Python"

此条目由[Miller Wu](#)发表在[Python](#)分类目录，并贴了[matplotlib](#)、[pyplot](#)标签。将[固定链接](#)  
[<http://www.idataskys.com/python%E4%B8%ADmatplotlib%E7%BB%98%E5%88%B6%E5%9B%BE%E5%BD%A2/>]  
加入收藏夹。

《PYTHON中MATPLOTLIB绘制图形》上有2条评论



123

在八月 2, 2013 8:04 上午说道:

123



zzzlll

在四月 25, 2014 4:40 上午说道:

6的代码跟5一样了

[To Top](#)