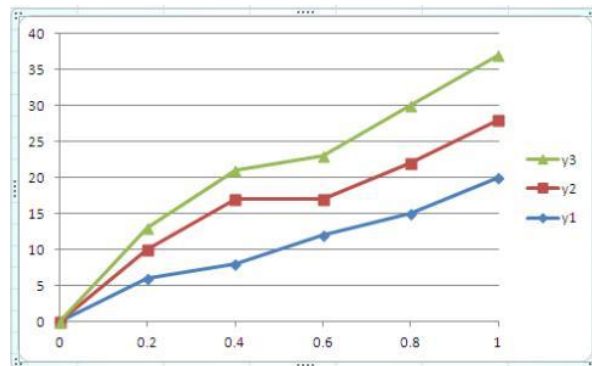
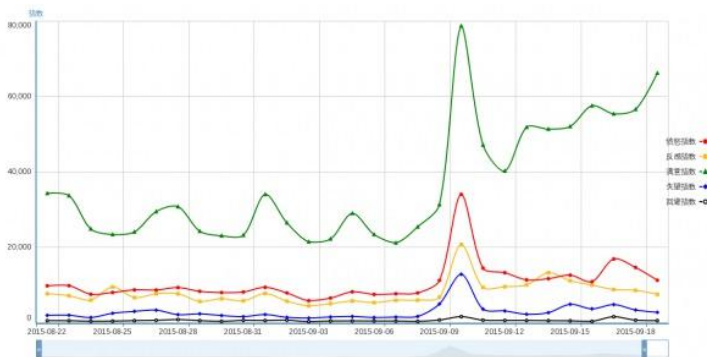




6.3折线图和柱状图

■ **折线图** (Line Chart) : 散点图的基础上, **将相邻的点用线段相连接**

□ 描述变量变化的趋势



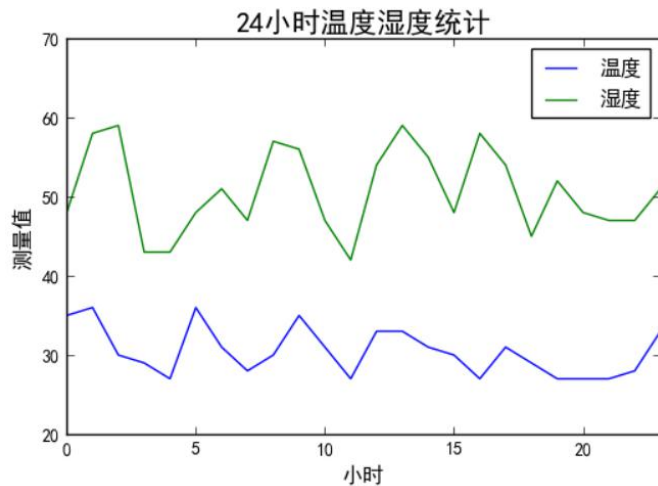
□ plot()函数

```
plot( x, y, color, marker, label, linewidth, markersize )
```

参 数	说 明	默认值
x	数据点的x坐标	0,1,2,3...
y	数据点的y坐标	不可省略
color	数据点的颜色	
marker	数据点的样式	'o' (圆点)
label	图例文字	
linewidth	折线的宽度	
markersize	数据点的大小	



例：绘制温度和湿度数据的折线图



□ 生成随机数列

```
n=24
```

```
y1 = np.random.randint(27,37,n)
```

```
y2 = np.random.randint(40,60,n)
```

□ 绘制折线图

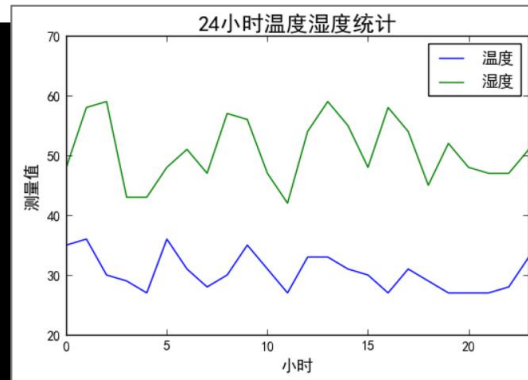
```
plt.plot(y1, label='温度')
```

```
plt.plot(y2, label='湿度')
```

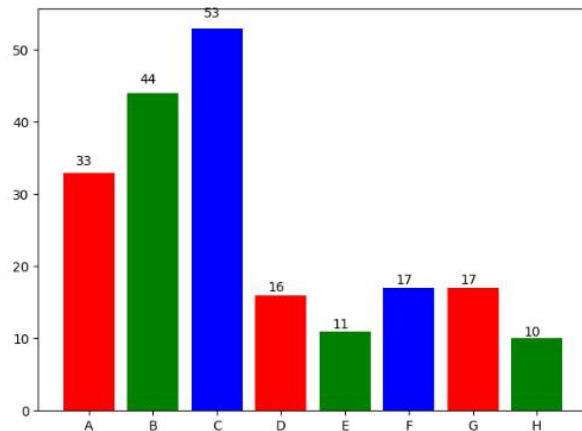
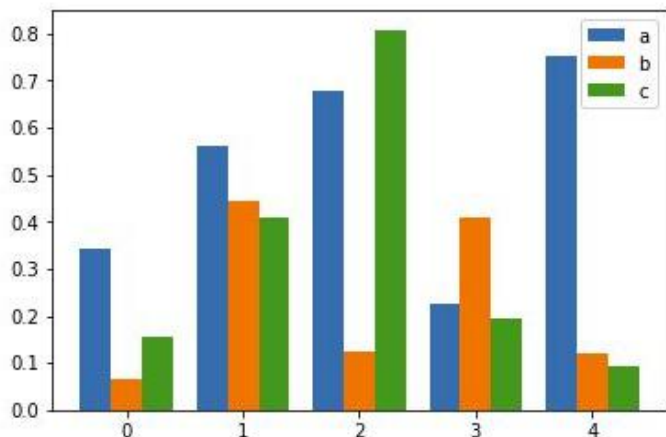


6.3 折线图和柱状图

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 plt.rcParams['font.sans-serif'] = 'SimHei'
5
6 n = 24
7 y1 = np.random.randint(27,37,n)
8 y2 = np.random.randint(40,60,n)
9
10 plt.plot(y1, label='温度')
11 plt.plot(y2, label='湿度')
12
13 plt.xlim(0,23)
14 plt.ylim(20,70)
15 plt.xlabel('小时', fontsize=12)
16 plt.ylabel('测量值', fontsize=12)
17
18 plt.title('24小时温度湿度统计', fontsize=16)
19
20 plt.legend()
21 plt.show()
```

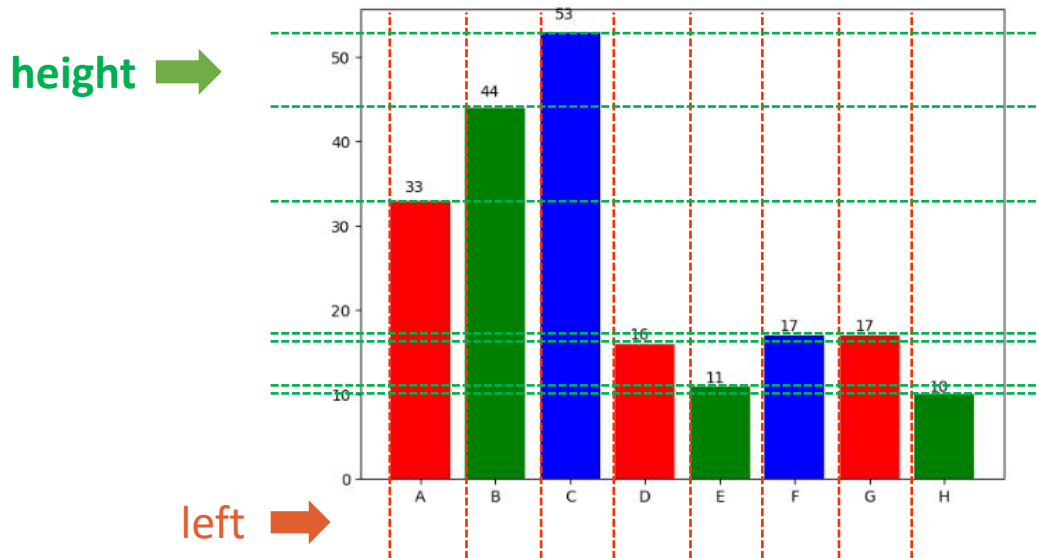


■ **柱形图** (Bar Chart) : 由一系列高度不等的柱形条纹表示数据分布的情况

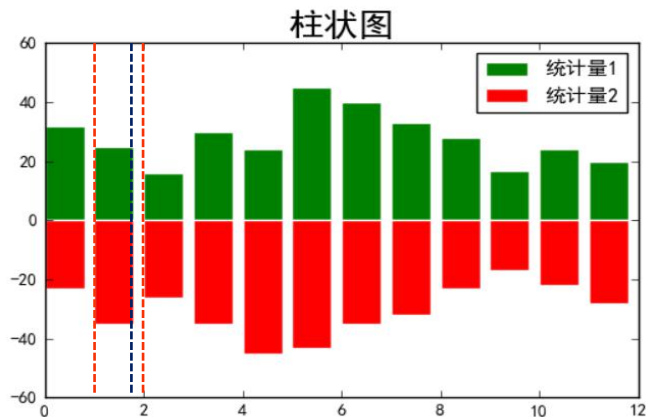


□ bar()函数

```
bar( left, height, width, facecolor, edgecolor, label )
```



例：绘制柱形图



自动生成left坐标序列

```
plt.bar(range(len(y1)), y1,width=0.8,facecolor='green',edgecolor='white',label='统计量1')  
plt.bar(range(len(y2)), y2,width=0.8,facecolor='red',edgecolor='white',label='统计量2')
```

□ 条纹高度

$y1=[32,25,16,30,24,45,40,33,28,17,24,20]$

$y2=[-23,-35,-26,-35,-45,-43,-35,-32,-23,-17,-22,-28]$

□ 条纹left坐标

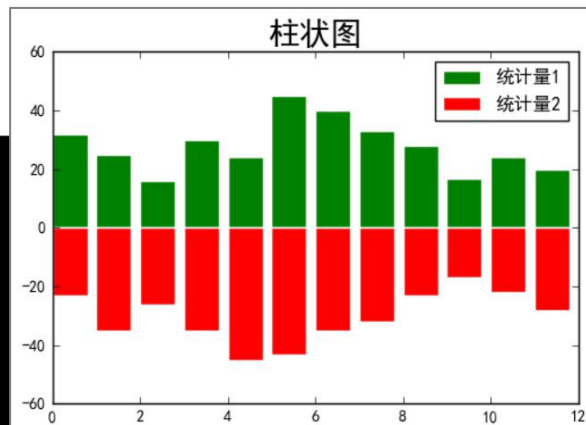
条纹的宽度**0.8**，每隔**1cm**开始画一个条纹



6.3 折线图和柱状图

例：绘制柱形图

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 plt.rcParams['font.sans-serif']="SimHei"
5 plt.rcParams["axes.unicode_minus"] = False
6
7 y1=[32,25,16,30,24,45,40,33,28,17,24,20]
8 y2=[-23,-35,-26,-35,-45,-43,-35,-32,-23,-17,-22,-28]
9
10 plt.bar(range(len(y1)), y1,width=0.8,facecolor='green',edgecolor='white',label='统计量1')
11 plt.bar(range(len(y2)), y2,width=0.8,facecolor='red',edgecolor='white',label='统计量2')
12
13 plt.title("柱状图",fontsize=20)
14
15 plt.legend()
16 plt.show()
```



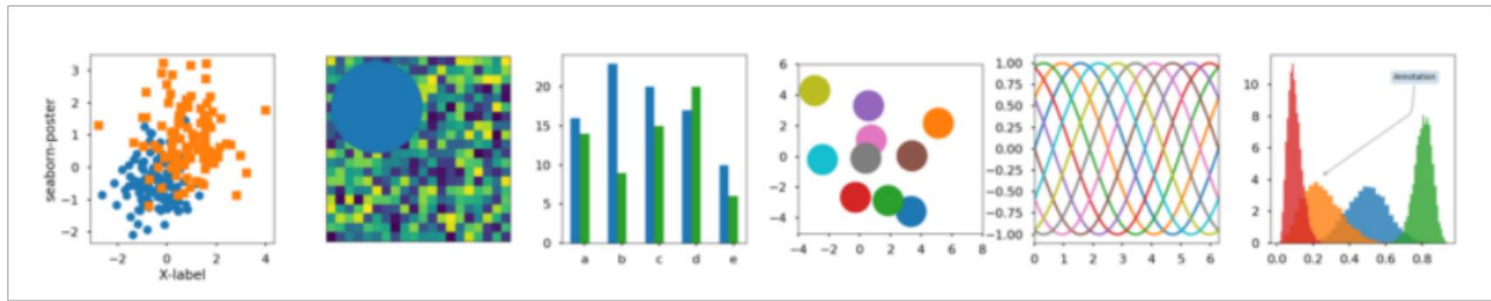
■ Matplotlib官网

[http:// matplotlib.org](http://matplotlib.org)

<https://matplotlib.org/genindex.html>

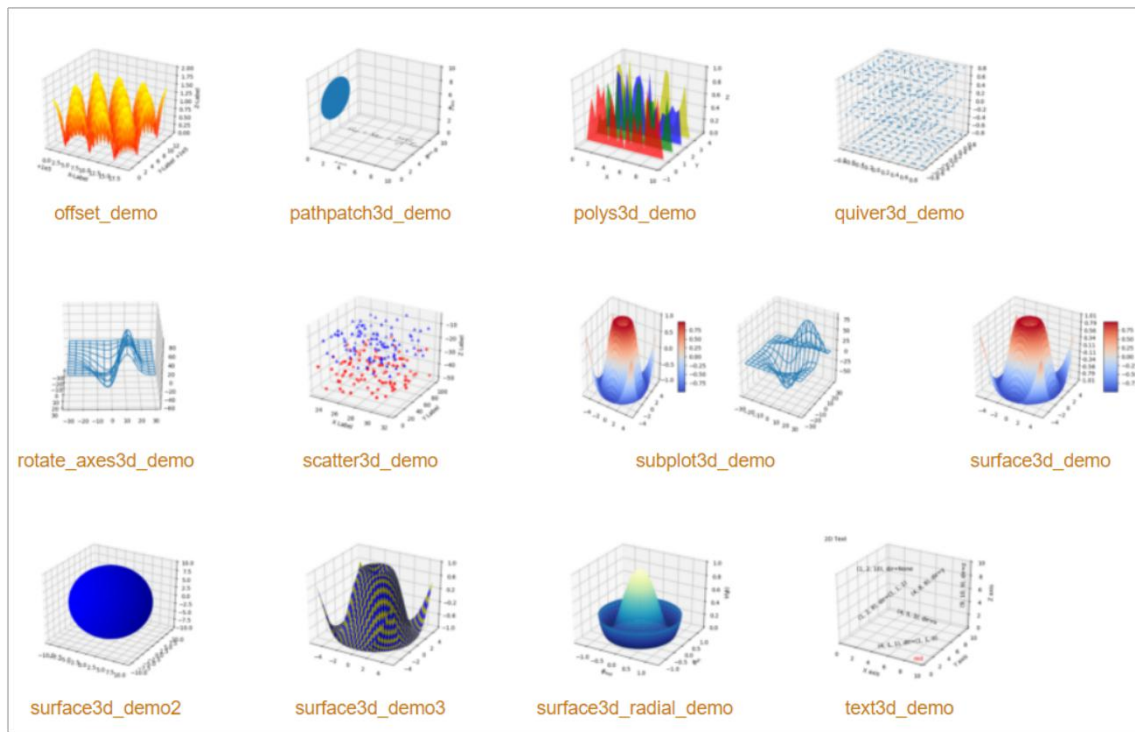
□ Gallery页面

<https://matplotlib.org/gallery.html>



6.3 折线图和柱状图

□ 缩略图

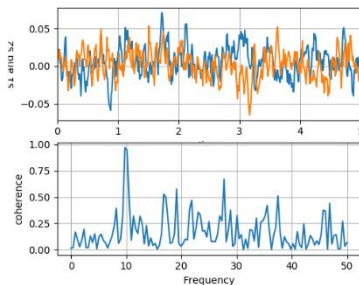


6.3 折线图和柱状图

详细页面 清晰大图&源码

pylab_examples example code: cohere_demo.py

(Source code, png, pdf)



```
"""
Compute the coherence of two signals
"""
import numpy as np
import matplotlib.pyplot as plt

# make a little extra space between the subplots
plt.subplots_adjust(wspace=0.5)

dt = 0.01
t = np.arange(0, 30, dt)
nse1 = np.random.randn(len(t)) # white noise 1
nse2 = np.random.randn(len(t)) # white noise 2
r = np.exp(-t/0.05)

cnsel = np.convolve(nse1, r, mode='same')*dt # colored noise 1
cnsel2 = np.convolve(nse2, r, mode='same')*dt # colored noise 2

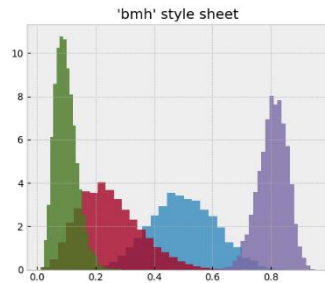
# two signals with a coherent part and a random part
s1 = 0.01*np.sin(2*np.pi*10*t) + cnsel
s2 = 0.01*np.sin(2*np.pi*10*t) + cnsel2

plt.subplot(211)
plt.plot(t, s1, t, s2)
plt.xlim(0, 5)
plt.xlabel('time')
plt.ylabel('s1 and s2')
plt.grid(True)

plt.subplot(212)
cov, f = plt.cohere(s1, s2, 256, 1./dt)
plt.ylabel('coherence')
plt.show()
```

style_sheets example code: plot_bmh.py

(Source code, png, pdf)



```
"""
Bayesian Methods for Hackers style sheet
"""

This example demonstrates the style used in the Bayesian Methods for Hackers
[1], online book.

.. [1] http://camdavidsonpilon.github.io/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/

"""
from numpy.random import beta
import matplotlib.pyplot as plt

plt.style.use('bmh')

def plot_beta_hist(ax, a, b):
    ax.hist(beta(a, b, size=10000), histtype='stepfilled',
            bins=25, alpha=0.8, normed=True)

fig, ax = plt.subplots()
plot_beta_hist(ax, 10, 10)
plot_beta_hist(ax, 4, 12)
plot_beta_hist(ax, 50, 12)
plot_beta_hist(ax, 6, 55)
ax.set_title("'bmh' style sheet")

plt.show()
```

Keywords: python, matplotlib, pylab, example, codex (see [Search examples](#))

