

# matplotlib绘图笔记

课程: [Matplotlib教程系列-用Python绘制图形](#)

## 绘制基本图表

[matplotlib绘图入门详解](#)

函数本身都有详细文档

散点图: scatter, 可选参数见 [matplotlib颜色, 线条, mark点](#)

直方图: hist, 参照 [matplotlib.pyplot中的hist函数](#)

堆积图: stackplot, 直接见notebook

K线图: [使用matplotlib绘制K线图以及和成交量的组合图](#)

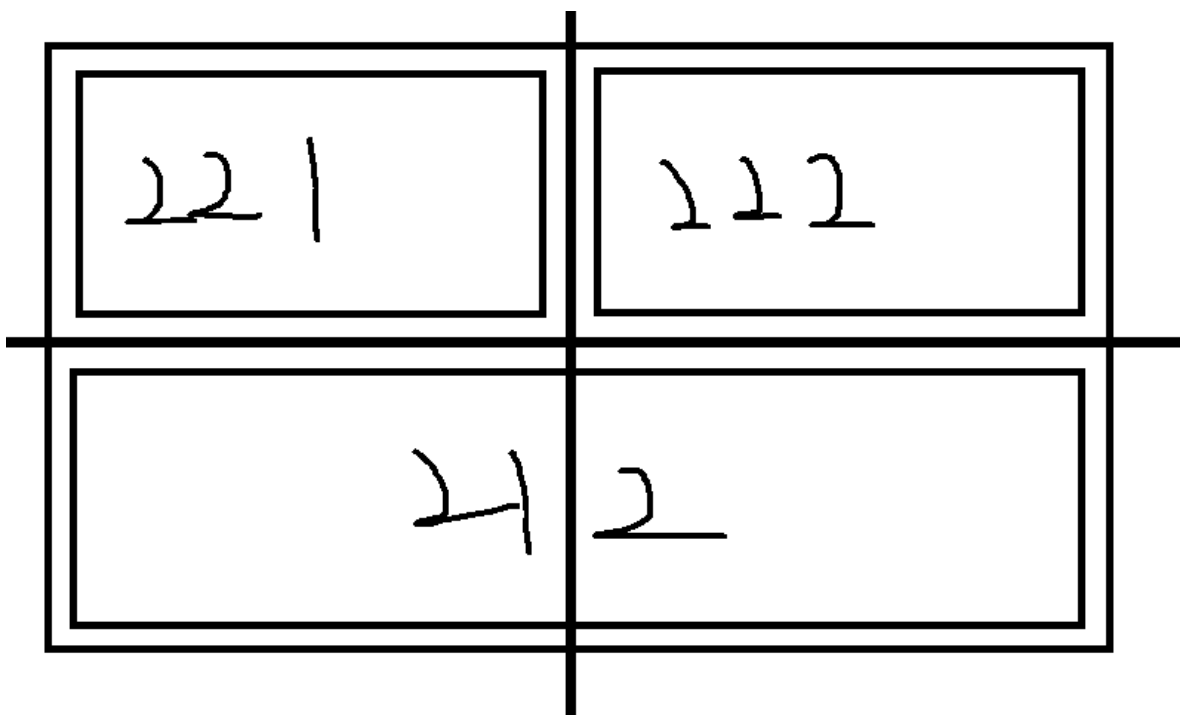
## 绘制子图

[subplot, subplot2grid, gridspec, subplots分图、分格展示](#)

subplot2grid的优势在于其支持不均匀分图, 而subplot必须是均匀的

```
fig=plt.figure()
# 创建子图, 图共包含1行1列, 取【0, 0】位置的图并命名为ax
# rowspan表示该图占多少行, colspan表示该图占多少列
# 通过共享坐标轴可以实现多图同步缩放
ax=plt.subplot2grid((1,1),(0,0),rowspan=1,colspan=1, sharx=ax1)
# 或采用如下方式包含1*1的子图, ax为第一张子图
ax=fig.add_subplot(1,1,1)
```

注意三个数值只是表示位置, 并非意味着一定划分成四份, 如下图所示 (subplot)



# 绘图技巧

## 更改坐标轴标签的颜色

```
# 注意ax没有xlabel属性，但是plt没法改label颜色
plt.xlabel('whatever')
ax.xaxis.label.set_color('c')
```

## 设置坐标轴刻度

```
# 将纵轴刻度设置为如下值
ax.set_yticks([0,25,50,75])
```

## 填充曲线1与曲线2之间的面积/数据高亮

[Matplotlib中的fill\\_between总结](#)

```
# 填充line1与line2之间的面积
# where是判断条件，例如只填充line1>line2的情况，写为where=(line1>line2 )
ax.fill_between(x,line1,line2,where,facecolor,alpha)

# 给面积添加label，课程中给红色以及盈利分别添加了loss和gain的标签，即画空图
ax.plot([],[],linewidth=5,label='loss',color='r',alpha=0.5)
```

## 处理时间

[Formatting date ticks using ConciseDateFormatter](#)

```
import date as dt

# date的格式形如20200419
dateconv=np.vectorize(dt.datetime.fromtimestamp)
data=dateconv(date)
```

## 设置网格线

```
ax.grid(True,color='g',linestyle='-',linewidth=2)
```

## 调节图像距框边距

在不使用notebook的情况下可以点击config来调节各边离边框的距离

```
#ax针对的是子图，而plt针对的是全局，数值可以理解成padding
plt.subplots_adjust(left=0.1,bottom=0.18,right=0.94,top=0.9,wspace=0.2,hspace=0)
```

## 使图像更紧凑

`plt.tight_layout()`：自动调整子图参数，使之填充整个图像区域。

## 坐标轴和标签

## [matplotlib.ticker](#)

```
# 还可以设置为right top bottom
# 设置左边线为青色
ax.spines['left'].set_color('c')
# 设置宽度
ax.spines['left'].set_linewidth(5)
# 设置右边线不可见
ax.spines['right'].set_visible(False)

# 设置横轴标签的颜色
ax.tick_params(axis='x',color='#f06215')
# 将横轴标签旋转45度
for label in ax.xaxis.get_ticklabels():
    label.set_rotation(45)

# set_major_locator设置参数的位置
# MaxNLocator表示最多有5个标签，当其重叠时，去掉数值较低的标签
ax.yaxis.set_major_locator(mticker.MaxNLocator(nbins=5,prune='lower'))
```

## 标记线

```
# 水平标记线
ax.axhline(val,color='c',linewidth=5)
```

## 设置样式

### [Customizing Matplotlib with style sheets and rcParams](#)

```
from matplotlib import style

# 查看所有样式
print(plt.style.available)
# 使用特定样式
style.use('ggplot')
```

通过 `print(plt.__file__)` 找到matplotlib的文件位置，样式文件保存在 `mpl-data\stylelib` 目录下，可以修改已有样式，也可以新增自定义样式。

## 动态图表

### [matplotlib.animation](#)

```
import matplotlib.animation as animation
```

notebook无法显示动态图，notebook中动态图代码显示的是通过不断更新起始点实现的，教程（P16）中的代码是通过不断读取文件并重新绘图实现的（注意先执行 `ax.clear()` 再plot）

## 注释

[matplotlib.pyplot.text\(\)结构及用法](#) | [参数详解](#)

[Matplotlib中的annotate（注解）的用法](#)

☆ [bbox参数, 箭头形状等](#)

```
# 在x[0],y[0]处添加注释
fontdict={'family':'serif','color':'darkred','size':15}
ax.text(x[0],y[0],"text",fontdict=fontdict)

# 文本, 被注释点, 注释点, 注释点的参考系, 箭头样式, 外壳样式
ax.annotate("annotation", (x[0],y[0]),
            xytext=(0.8,0.9),textcoords="axes fraction",
            arrowprops=dict(facecolor='grey',color='grey'),
            bbox=dict(boxstyle='round',fc='w',ec='k',lw=1))
```

## 设置artist属性

[matplotlib Artist 结构详解](#)

[matplotlib.pyplot.setp](#)

```
# 设置横轴标签不可见
plt.setp(ax.get_xticklabels(),visible=False)
```

## 多y轴图

```
# 共用x轴
ax2v=ax2.twinx()
ax2v.fill_between(...)
# 如果[...]为空, 则表示取消该轴标签
ax2v.axes.yaxis.set_ticklabels([])
ax2v.grid(False)
```

## 调整图例

[matplotlib.axes.Axes.legend](#)

```
leg=ax.legend(loc=9,ncol=2,prop={'size':1})
```

## 问题处理

保存图片时颜色出错

```
fig=plt.figure(facecolor='...')
...
fig.savefig('test.png',facecolor=fig.get_facecolor())
```

## 3D

```

from mpl_toolkits.mplot3d import axes3d

fig=plt.figure()
ax=fig.add_subplot(111,projection='3d')
...
# 3维线框,也就是网面
ax.plot_wireframe(x,y,z, rstride=3,cstride=3)
# 3维点图
ax.scatter(x,y,z,c='r',marker='*')
# 3维柱状图
ax.bar3d(x,y,z,dx,dy,dz)

```

## 绘制地图

[basemap document](#)

[cartopy document](#)

cartopy将取代basemap (但basemap还是很酷炫)

```

from mpl_toolkits.basemap import Basemap

# 投影方式决定地图样式, 而后四个参数表示显示的区域 (两个点的经纬度)
# resolution表示画质, 其中'h'表示高品质,'l'表示低品质,'f'表示完全
m=Basemap(projection='moll',
           llcrnrlat=-40,llcrnrlon=-40,
           urcrnrlat=50,urcrnrlon=75,
           resolution='l')

# 画出海岸线轮廓
m.drawcoastlines()
# 给大洲上色
m.fillcontinents()
# 显示国家
m.drawcountries(linewidth=2)
# 给美国各州上色
m.drawcounties(color='darkred')
# 地理图
m.etopo()
# 太空视角下的地球
m.bluemarble()

# 经度对应x值, 纬度对应y值
lat, lon = ..., ...
xpt,ypt=m(lon,lat)
m.plot(xpt,ypt)
m.drawgreatcir(lon1,lat1,lon2,lat2)

```