

Lecture5

红色标注的语句，为重点。

蓝色下划线标注的语句，说明给出了参考阅读链接，可依兴趣阅读。

Scipy入门——Matplotlib篇

Matplotlib是一个支持numpy数据结构的，功能全面的绘图模块，它也是Python生态环境中最主要的绘图工具，和jupyter notebook等主流运行环境密切结合。

Matplotlib的主要特点如下：1.默认绘图精美，相较于Excel、R等绘图工具，Matplotlib的默认配色、主流绘图模版更为精致耐看。2.可定制化程度高，几乎图中的所有元素都可以由代码自定义，很好地契合了学术绘图的需求。3.可扩展性强，以Matplotlib为基础，有一大批构建其上的第三方模块，如Seabron、Geoplot等，它们有些针对绘图过程的简化，有些专攻某种特殊绘图（如地理图像绘制），形成了一整套科学绘图的解决方案。4.完全免费开源，这对很多学术研究者是一大福音。

在本节内容中，我们主要讨论Matplotlib本体，不涉及第三方扩展模块。

对扩展模块感兴趣的朋友可以参照：<https://matplotlib.org/thirdpartypackages/index.html>。

1. Matplotlib的基本概念

——安装与引入

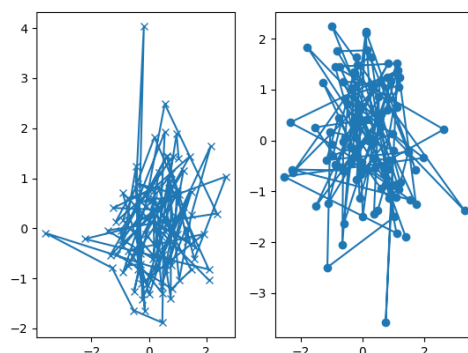
作为Scipy生态圈的一部分，Matplotlib的配置与Numpy、Pandas类似。没有安装的朋友可以使用pip工具安装（见Lecture1），如果已经配置好Anaconda环境的朋友依旧可以跳过。

我们通常使用以下语句声明对Matplotlib的调用：

```
In [6]: import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
```

如图，一般来说，先行导入Numpy，方便画图过程中的数据存储。Matplotlib一般的缩写为mpl。此外，我们还需要导入matplotlib.pyplot，约定俗成记为plt。事实上，**plt正是我们绘图时操作的主要对象**。

——Figure与Axes



Lecture5

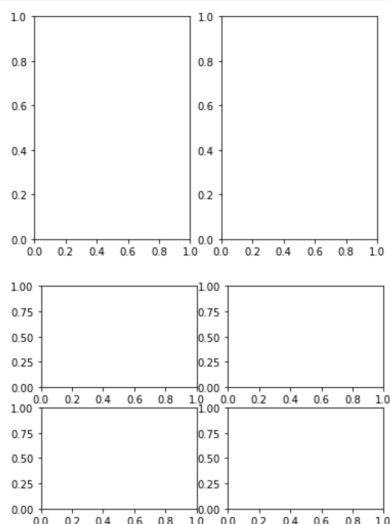
在Matplotlib中，一张图片被放在一个Figure中，一个Figure可能有包含多个Axes，每个Axes存储一个子图。

上图中，整张图片为一个Figure，包含两个子图，左边的图片为Axes1，右边的为Axes2。

创建Figure和Axes的方法如下：

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

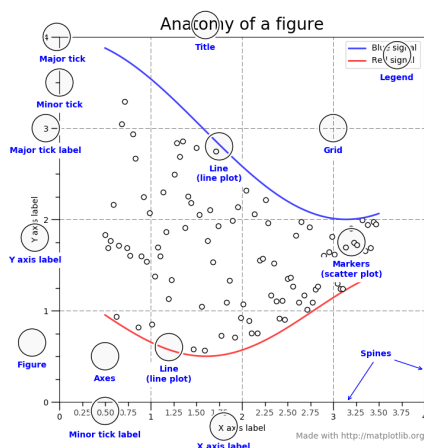
#fig, ax = plt.subplots()
fig, (ax1,ax2) = plt.subplots(1,2)
fig, ((ax1,ax2),(ax3,ax4)) = plt.subplots(2,2)
```



可以发现，我们能够使用plt.subplots()方法建立图和子图，使用固定的fig变量承接整张图，使用ax变量去承接每一张子图。

plt.subplots()方法的文档见：https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots.html#matplotlib.pyplot.subplots

Figure的几乎所有部分都可以自定义，这也符合前面提到的Matplotlib的特点，可定义的元素如下：



Lecture5

我们主要关注点、线、坐标轴、图例、标题、网格这几大主要元素。

— — Matplotlib的两种写法

为了方便帮助习惯的用户快速适应，Matplotlib支持两种风格的写法，一种以对象为核心，一种以前文提到的plt为核心。风格分别如下（1为对象核心，2为plt核心）：

```
In [12]: x = np.linspace(0, 2, 100)

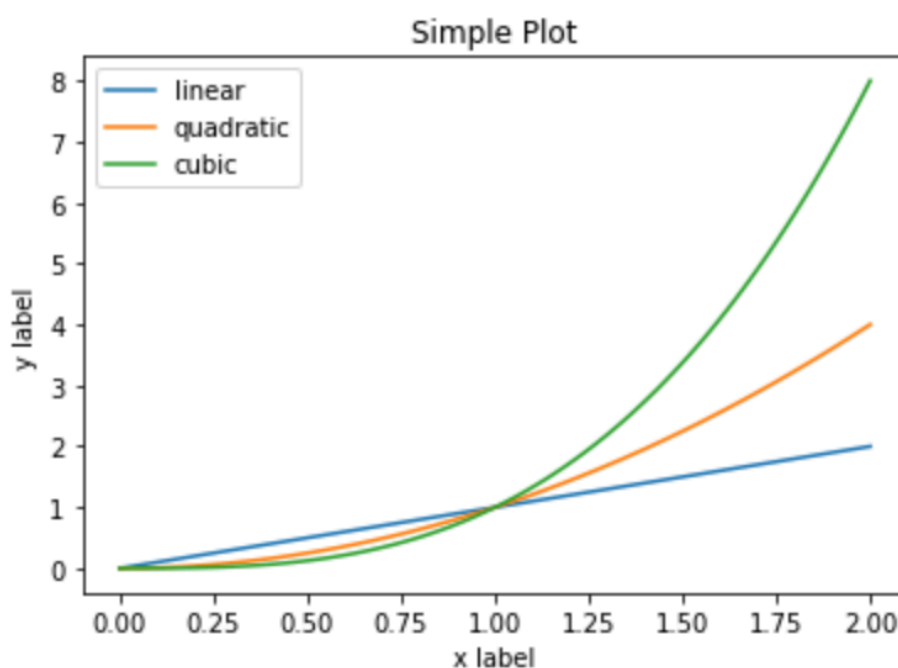
# Note that even in the OO-style, we use `.pyplot.figure` to create the figure.
fig, ax = plt.subplots() # Create a figure and an axes.
ax.plot(x, x, label='linear') # Plot some data on the axes.
ax.plot(x, x**2, label='quadratic') # Plot more data on the axes...
ax.plot(x, x**3, label='cubic') # ... and some more.
ax.set_xlabel('x label') # Add an x-label to the axes.
ax.set_ylabel('y label') # Add a y-label to the axes.
ax.set_title("Simple Plot") # Add a title to the axes.
ax.legend() # Add a legend.
```

```
In [14]: x = np.linspace(0, 2, 100)

plt.plot(x, x, label='linear') # Plot some data on the (implicit) axes.
plt.plot(x, x**2, label='quadratic') # etc.
plt.plot(x, x**3, label='cubic')
plt.xlabel('x label')
plt.ylabel('y label')
plt.title("Simple Plot")
plt.legend()
```

互联网上查询到的样例，往往两种风格混杂，非常不利于初学者的理解。在这里，我们建议大家使用第二种，即以plt为核心的写法，本节所有展示代码也会以第二种风格为准。

— — 用Matplotlib绘制第一张图



Lecture5

上图即为前面两段代码绘制出的图片，建议大家自己在Jupyter中输入上面代码的内容，体会Python绘图的代码风格。

这里我们结合图片和代码，做一个简单的解读：可以看到，我们使用`np.linspace`函数（可见Lecture3）建立了一个0-100（包含100），间距为2的数组，该数组为数据中的x。后面的三条`plt.plot()`指令为最常用的绘制线的指令，需要我们给出一系列的x, y坐标，显然，y分别为x, x平方, x的三次方。在这里，由于给出的x, y坐标足够稠密，绘制出的线为曲线，如果过于稀疏的话，绘制出来为折线，因此实际操作中，想要画出曲线，我们往往要对数据进行插值。

在绘制线时，我们可以指定label标签，使用`plt.legend()`方法可以自动根据label生成图例。

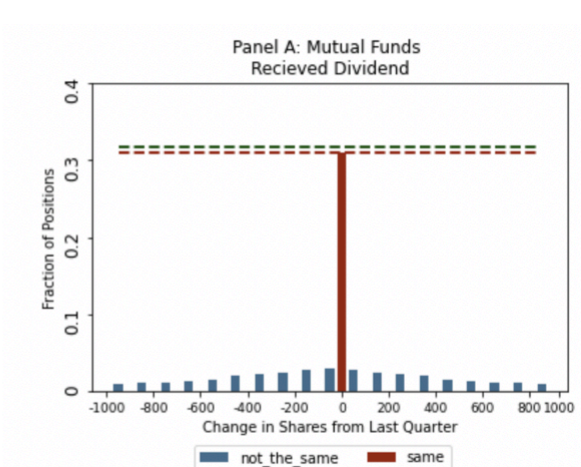
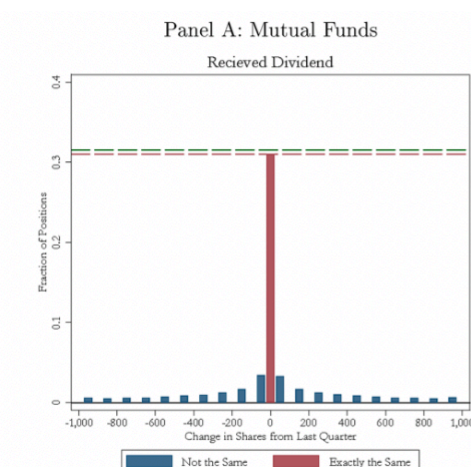
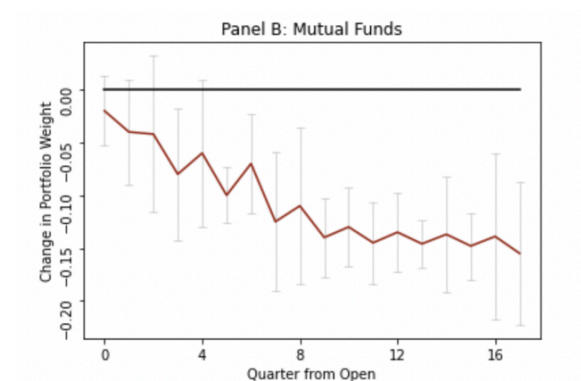
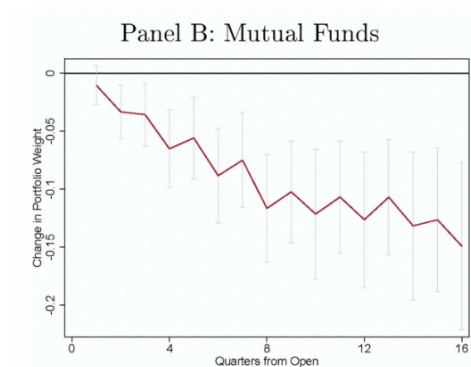
此外，使用`plt.xlabel()`/`plt.ylabel()`方法，让label标签被指定到x轴和y轴上。使用`plt.title()`方法指定标题。

想要导出图片，使用`plt.savefig('test.png',dpi=326)`方法即可，其会在指定路径下生成图片`test.png`（不指定的话默认为当前工作目录），dpi参数一般选择300以上即可满足清晰度要求。

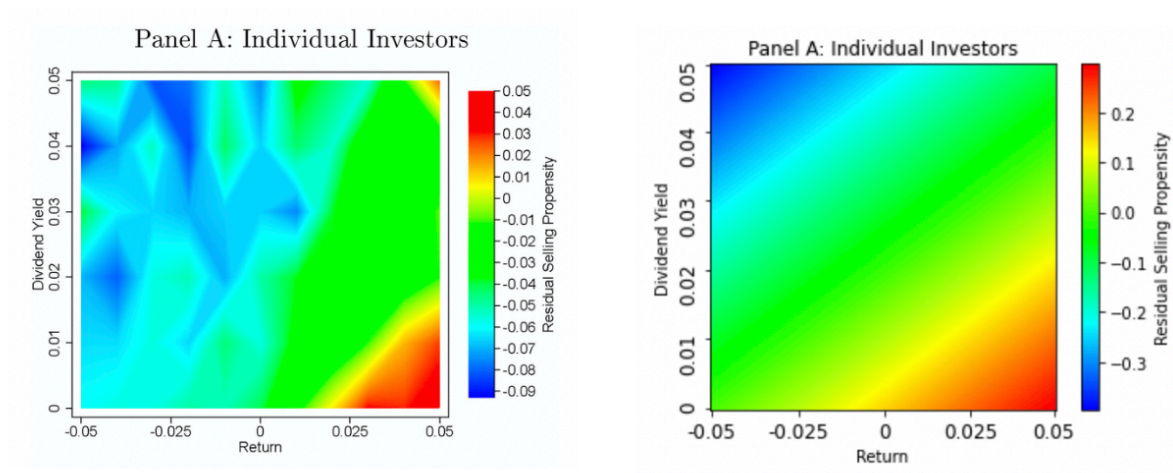
2. Matplotlib实战

对于Matplotlib这样完全实践导向的模块，最快的学习方法莫过于自己动手实践。为了贴合大家在学术研究中的实际需求，我选取了原计划在本周组会上分享的，发表于Journal of Finance上的2019年度最佳论文：The Dividend Disconnect中的三张插图，使用Matplotlib进行复制，大家可以在动手中快速掌握较复杂图形的绘制方法。

Paper原图（左）和我们的复制品（右）如下：



Lecture5



(由于数据存在差异，绘图效果可能有一定差别)

对于每张图片，我会在pdf中提供未经注释的代码截图，希望有时间的朋友可以自己动手敲一敲，理解和猜测每行代码可能的意思。

对于每张图片，我也会提供理解代码可能需要的参考材料，供大家探索学习。

我会在下周三（10.28）上传注释过的代码。

— Task1:带误差线的折线图

```
In [18]: import matplotlib.pyplot as plt
import numpy as np

x_1 = np.arange(0, 18)
y_1 = np.array([-0.02,-0.04,-0.042,-0.08,-0.06,-0.1,
               -0.07,-0.125,-0.11,-0.14,-0.13,-0.145,
               -0.135,-0.146,-0.137,-0.148,-0.139,-0.155])
y_2 = np.array([0]*18)
err_1 = 2* np.random.choice(np.arange(0.01, 0.04, 0.0001), 18)

plt.plot(x_1,y_1,color="#902B19")
plt.plot(x_1,y_2,color="black")

plt.errorbar(x=x_1, y=y_1, yerr=err_1, color="grey", capsize=3, alpha = 0.3, linewidth=1,
             linestyle="None",mfc="black", mec="black")

x_ticks = (0, 4, 8, 12, 16)
x_labels = ("0", "4", "8", "12", "16")
plt.xticks(x_ticks, x_labels, rotation=0)

plt.yticks(rotation=90)

plt.xlabel("Quarter from Open")
plt.ylabel("Change in Portfolio Weight")

plt.title('Panel B: Mutual Funds')
plt.show()
```

Lecture5

参考链接：

[涉及Numpy的基本操作：Lecture3](#)

[如何画误差线：https://www.cnblogs.com/HuZihu/p/9418903.html](https://www.cnblogs.com/HuZihu/p/9418903.html)

[plt.errorbar的官方文档：https://matplotlib.org/api/_as_gen/matplotlib.pyplot.errorbar.html](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.errorbar.html)

[如何重新标注刻度线和刻度：https://www.cnblogs.com/nju2014/p/5633768.html](https://www.cnblogs.com/nju2014/p/5633768.html)（看网址似乎是一位NJU校友的撰写的博客，该页面中两种代码风格混杂，请注意理解）

——Task2：两组不同数据的柱状图

```
In [24]: import numpy as np
import matplotlib.pyplot as plt

not_the_same = np.array([0.01, 0.011, 0.012, 0.013, 0.015,
                        0.02, 0.022, 0.025, 0.028, 0.03,
                        0.028, 0.025, 0.022, 0.02, 0.015,
                        0.013, 0.012, 0.011, 0.01])
same = np.array([0.31])

a = plt.bar(list(np.arange(len(not_the_same))*4), not_the_same, width = 1.5,
            color='#486C8B', label='not_the_same', align='center')
b = plt.bar(38, same, width = 1.5, color='#902B19', label='same', align='center')

plt.xlabel('Change in Shares from Last Quarter')
plt.xticks([-2,6,14,22,30,38,46,54,62,70,75], [str(ele) for ele in np.arange(-1000,1100,200)],
           rotation=0, fontsize=9)

plt.ylabel('Fraction of Positions')
plt.yticks([0,0.1,0.2,0.3,0.4], [0,0.1,0.2,0.3,0.4], rotation=90, fontsize=12)

plt.legend(ncol=3, loc=0, bbox_to_anchor=(0.77,-0.15))

x = np.linspace(0,71,72)
y1 = np.array([0.31]*72)
y2 = np.array([0.317]*72)

plt.ylim(0,0.4)
plt.plot(x, y1, '--', color='#902B19', linewidth=2)
plt.plot(x, y2, '--', color='#25551A', linewidth=2)

plt.title('Panel A: Mutual Funds \nRecieved Dividend')
plt.show()
```

参考资料：

[使用列表生成式方便地生成List：https://www.liaoxuefeng.com/wiki/1016959663602400/1017317609699776](https://www.liaoxuefeng.com/wiki/1016959663602400/1017317609699776)

[如何调整图例的默认位置：https://www.cnblogs.com/lvyWong/p/9916791.html](https://www.cnblogs.com/lvyWong/p/9916791.html)

[如何绘制柱状图：https://blog.csdn.net/qq_29721419/article/details/71638912](https://blog.csdn.net/qq_29721419/article/details/71638912)

Lecture5

— Task3:平滑的热力图

```
In [26]: import numpy as np
import numpy.random

import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib.colors import LinearSegmentedColormap

x = np.random.randn(100)
y = np.random.randn(100)

heatmap = np.random.rand(100,100)
for i in range(100):
    for j in range(100):
        heatmap[i,j] = 0.001*(3*i-4*j)

cdict = {'red': ((0.0, 0.0, 0.0), # no red at 0.0
                 (0.25, 0.0, 0.0), # no red at 0.25
                 (0.5, 0.0, 0.0), # no red at 0.5
                 (0.75, 1.0, 1.0), # red at 0.75
                 (1.0, 1.0, 1.0)), # red at 1.0
         'green': ((0.0, 0.0, 0.0),
                  (0.25, 1.0, 1.0),
                  (0.5, 1.0, 1.0),
                  (0.75, 1.0, 1.0),
                  (1.0, 0.0, 0.0)),
         'blue': ((0.0, 1.0, 1.0),
                  (0.25, 1.0, 1.0),
                  (0.5, 0.0, 0.0),
                  (0.75, 0.0, 0.0),
                  (1.0, 0.0, 0.0))}

plt.xlabel('Return')
plt.xticks([0, 25, 50, 75, 99], ['-0.05', '-0.025', '0', '0.025', '0.05'], rotation=0, fontsize=12)

plt.ylabel('Dividend Yield')
plt.yticks([0, 20, 40, 60, 80, 99], ['0', '0.01', '0.02', '0.03', '0.04', '0.05'], rotation=90, fontsize=12)

cmap = LinearSegmentedColormap('Rd_Bl_Rd', cdict, 256)

plt.title("Panel A: Individual Investors")
plt.imshow(heatmap.T,origin='lower',cmap=cmap)
plt.colorbar().set_label('Residual Selling Propensity')
plt.show()
```

参考资料：

[plt.imshow\(\)函数：https://www.jb51.net/article/193182.htm](https://www.jb51.net/article/193182.htm)

[plt.imshow\(\)函数的官方文档：](#)

https://matplotlib.org/3.1.1/api/as_gen/matplotlib.pyplot.imshow.html

[设置colorbar：https://www.jianshu.com/p/d97c1d2e274f](https://www.jianshu.com/p/d97c1d2e274f)

[使用LinearSegmentedColormap自定义colorbar等颜色：](#)

https://matplotlib.org/3.1.1/api/as_gen/matplotlib.colors.LinearSegmentedColormap.html?highlight=linearsegmentedcolormap