

## matplotlib: 先搞明白plt. /ax./ fig再画



姚太多啊

4,310 人赞同了该文章

我这个傻子的matplotlib学习过程很神奇:

刚开始, 找几个教程, 呕吼, 简单, `plt.***` 一顿写。画完了。遇到坐标啊之类的细节不会调?

打开谷歌, 要啥搜啥。

我最开始觉得这样蛮好的, 这是一个很容易学的lib嘛。

然后问题出现了, matplotlib最烦人的地方就是, 同样一个目标, 你可以有N种不同的方法去实现他。比如说哦, 加标题:

`plt.set_title()` 可以吧。

`ax.title.set_text('title')` 也可以吧。

`ax.set_title()`也可以啊。

所以到底要用哪个??

当你在网上搜索答案的时候, 你会遇到各种实现的方法, 如果你只是单纯的把找到的答案coppypaste, 那总有一天你会和我一样崩溃。感觉自己就是个没有谷歌就不会画图的傻子。。。

所以, 如果你:

- 学习matplotlib的时候是从标题类似于"五分钟教你matplotlib画图" 这种教程开始的
- 起码知道这一句 `'import matplotlib.pyplot as plt'`
- 遇到稍微复杂的细节就要一顿查, 遇到修改xy坐标, 名称, 等等就头大
- 打开matplotlib的官方指南, 不止一次, 但是一脸懵逼
- 在这几个词 `figure`, `axes`, `axis` 在matplotlib出现时就一脸完了的表情

那么你不一个人。我一直和你在一起。

所以这份指南是用来(尝试) 解决这个问题。我们花短短的时间, 来从根本了解一下matplotlib的架构, 各种名词是什么意思, 一个正常的画图程序是什么。



# 知乎

我认为所有不先讲清楚plt.\*\*\*和ax.\*\*\* 两种画图方式的区别的教程都是耍流氓。一上来就告诉你，plt.figure(), plt.plot(), plt.show(), 这么画就对了的，都是不负责任的表现！

- 在matplotlib中，有两种画图方式：
  - plt.figure(): plt.\*\*\*系列。通过plt.xxx来画图，其实是取了一个捷径。这是通过matplotlib提供的一个api，这个plt提供了很多基本的function可以让你很快的画出图来，但是如果你想更细致的精调，就要使用另外一种方法。

```
plt.figure(1)
plt.subplot(211)
plt.plot(A,B)
plt.show()
```

- fig, ax = plt.subplots(): 这个就是正统的稍微复杂一点的画图方法了。指定figure和axes，然后对axes单独操作。等下就讲figure和axes都神马意思。

```
fig, ax = plt.subplots()
ax.plot(A,B)
```

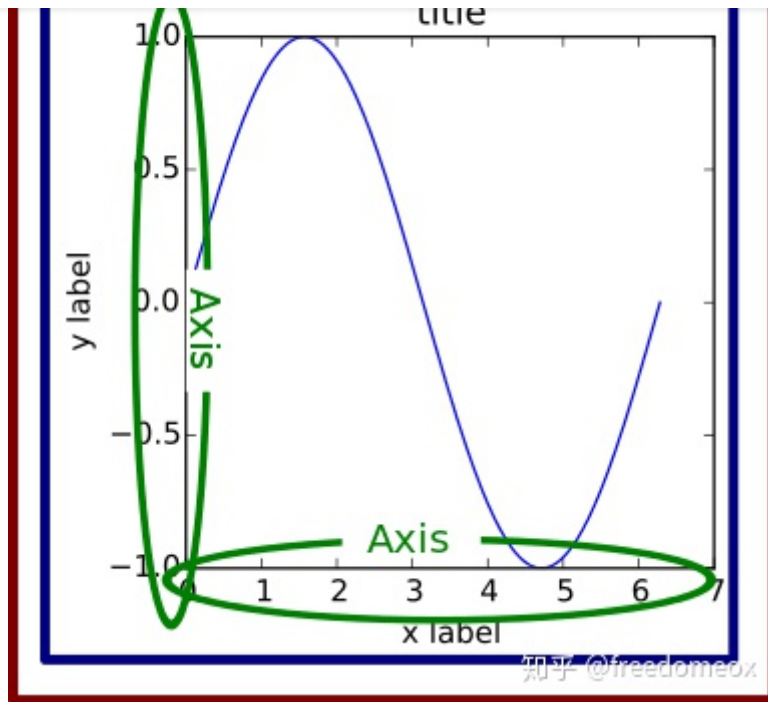
- 大部分的新手教程，都是以plt.xxx开始的。我个人认为这**不是**一个很好的方式。的确，plt.xxx简单易懂上手快，但是plt把matplotlib的结构看成一个blackbox，新手在不知道这个lib到底是如何运作的情况下以为自己上手了，可是当你需要细调你的图的时候，你就发现你完全不明白网上的人为什么要用ax.xxx，为什么还有其他那么多方式来加题目。这样的学习过程是非常不利于长期发展的。
- 因此，从我这个傻子的经验，我强烈建议在初学matplotlib的时候，**尽量避免使用plt.xxx系列**。当你明白figure/axes/axis都是控制什么的时候，如果你想要简单的制作一个quick and dirty的图，用plt.xxx才是OK。

## 名词解释 in matplotlib

matplotlib的名词定于对于非英语母语的人来说实在是太不友好了。尤其是axes。仰天长啸。

从官方借了个图





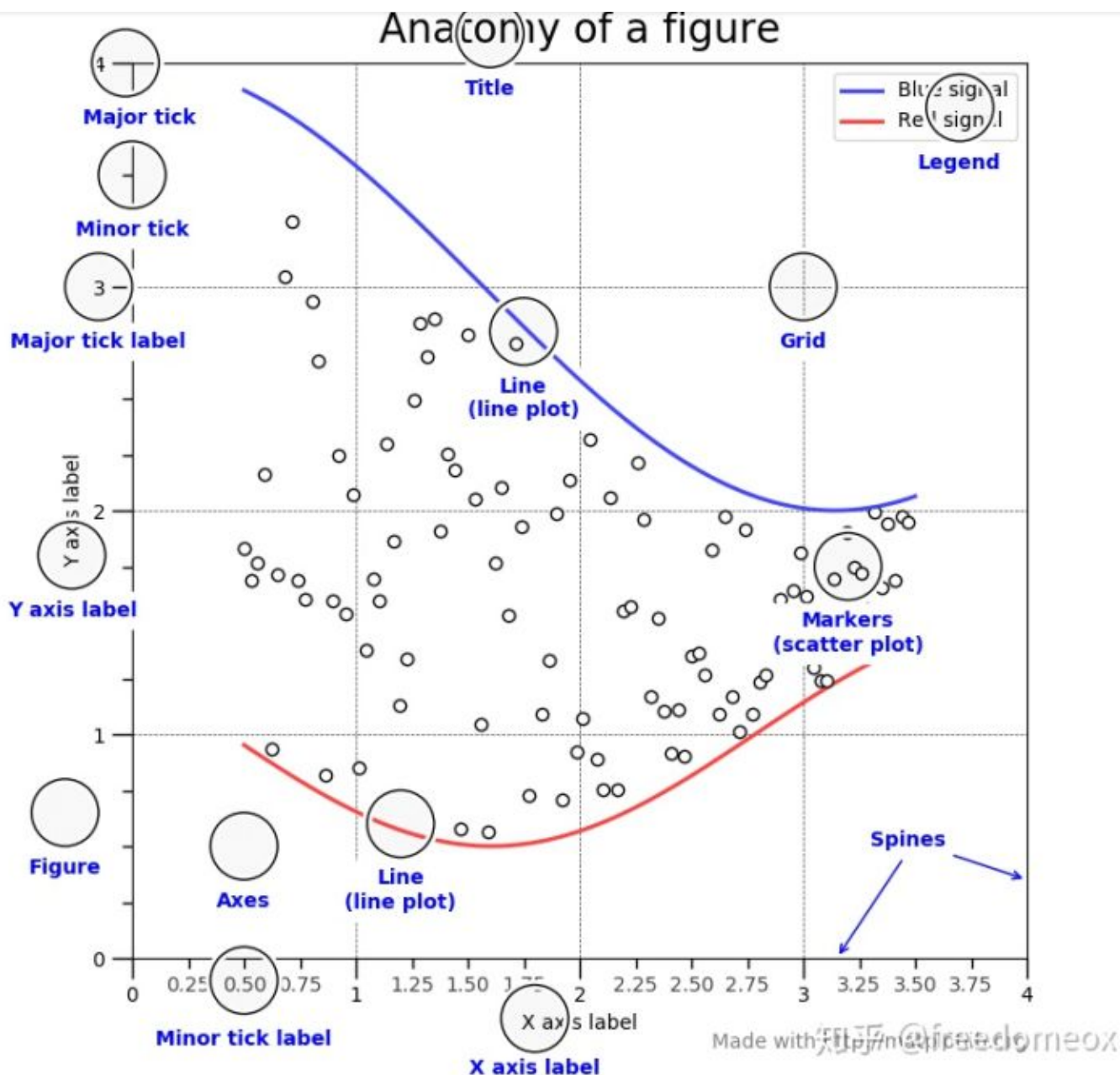
- `Figure fig = plt.figure()` : 可以解释为画布。
  - 画图的第一件事, 就是创建一个画布figure, 然后在这个画布上加各种元素。
- `Axes ax = fig.add_subplot(1,1,1)` : 不想定义, 没法定义, 就叫他axes!
  - 首先, 这个不是你画图的xy坐标轴!
  - 希望当初写这个lib的时候他们用一个更好的名字。。。
  - 可以把axes理解为你要放到画布上的各个物体。比如你要画一个太阳, 一个房子, 一个车在画布上, 那么太阳是一个axes, 房子是一个axes, etc。
  - 如果你的figure只有一张图, 那么你只有一个axes。如果你的figure有subplot, 那么每一个subplot就是一个axes
  - axes是matplotlib的宇宙中心! axes下可以修改编辑的变量非常多, 基本上能包含你的所有需求。
- `Axis ax.xaxis/ax.yaxis` : 对, 这才是你的xy坐标轴。
  - 每个坐标轴实际上也是由竖线和数字组成的, 每一个竖线其实也是一个axis的subplot, 因此ax.xaxis也存在axes 这个对象。对这个axes进行编辑就会修改xaxis图像上的表现。

## 图像的各个部位名称

再从使用指南 [User Guide](#) 借个图。每个部分的名称指南, 这样当你想修改一个部位的时候, 起码知道关键字啊。



知乎



## 一步一步来，用传统方法画个图

下面就是实战。用调取ax的方式来画个图。不要用plt！！

## Import library and data for plotting

```
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
A = np.arange(1,5)
```



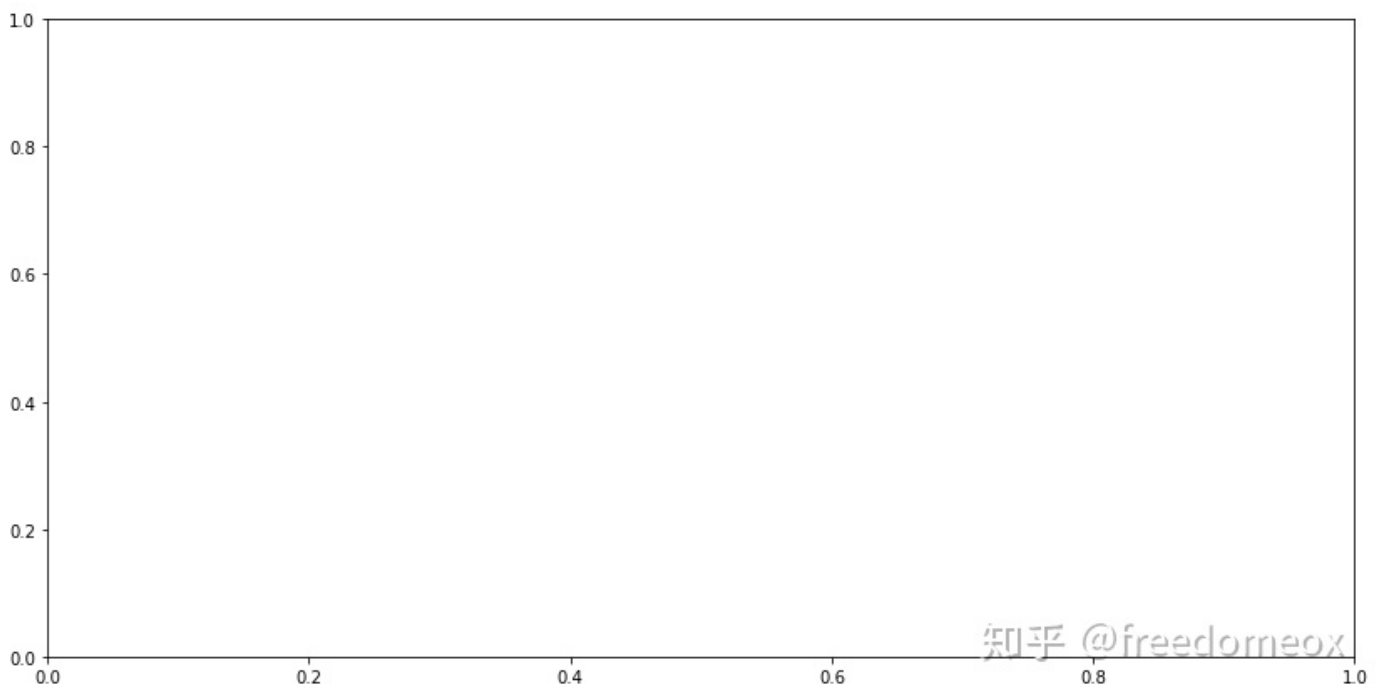
## 首先，搞个画布

我喜欢用这个命令来开始画图。哪怕你没有subplot，也可以用这个subplots来创建一个画布。

这个function创建了一个大小为（14，7）的画布，把这个画布赋值给变量fig，同时在这个画布上创建了一个axes，把这个axes赋值给ax。这样，所有未来的fig.xxx都是对这个画布的操作，所有ax.xxx都是对这个axes的操作。

如果你有两个图，那么ax是一个有两个元素ax[0],ax[1] 的list。ax[0]就对应第一个subplot的ax。

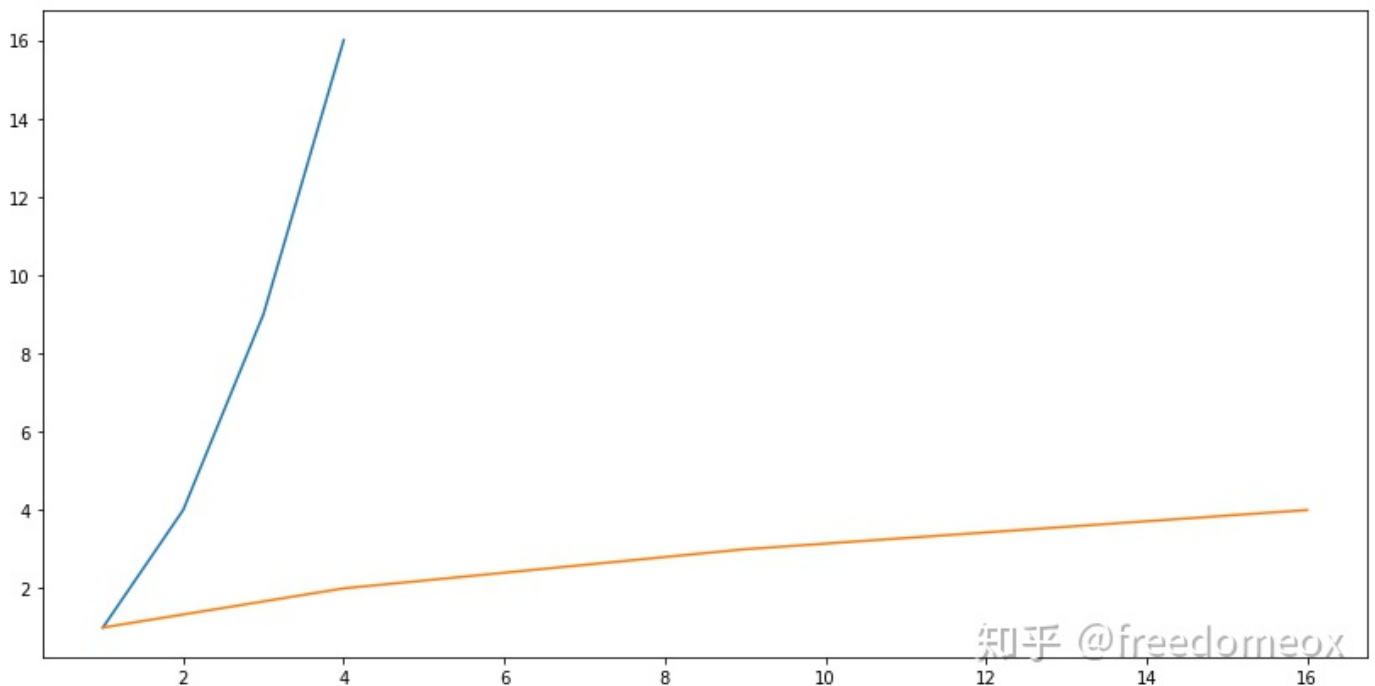
```
fig, ax = plt.subplots(figsize=(14,7))
# fig, ax = plt.subplots(2,1,figsize(14,7))
# ax[0].***
# ax[1].***
```



## 好了画布搞好了，画数据。

注意，我们这里依然不使用plt！因为我们要在这个axes上画数据，因此就用ax.plot()来画。画完第一个再call一次，再画第二个。





## 下面开始细节的处理

数据画好了就可以各种细调坐标轴啊, tick啊之类的。

首先把标题和xy坐标轴的标题搞定。Again, 不用plt。直接在axes上进行设定。

```
ax.set_title('Title', fontsize=18)
ax.set_xlabel('xlabel', fontsize=18, fontfamily = 'sans-serif', fontstyle='italic')
ax.set_ylabel('ylabel', fontsize='x-large', fontstyle='oblique')
ax.legend()
```

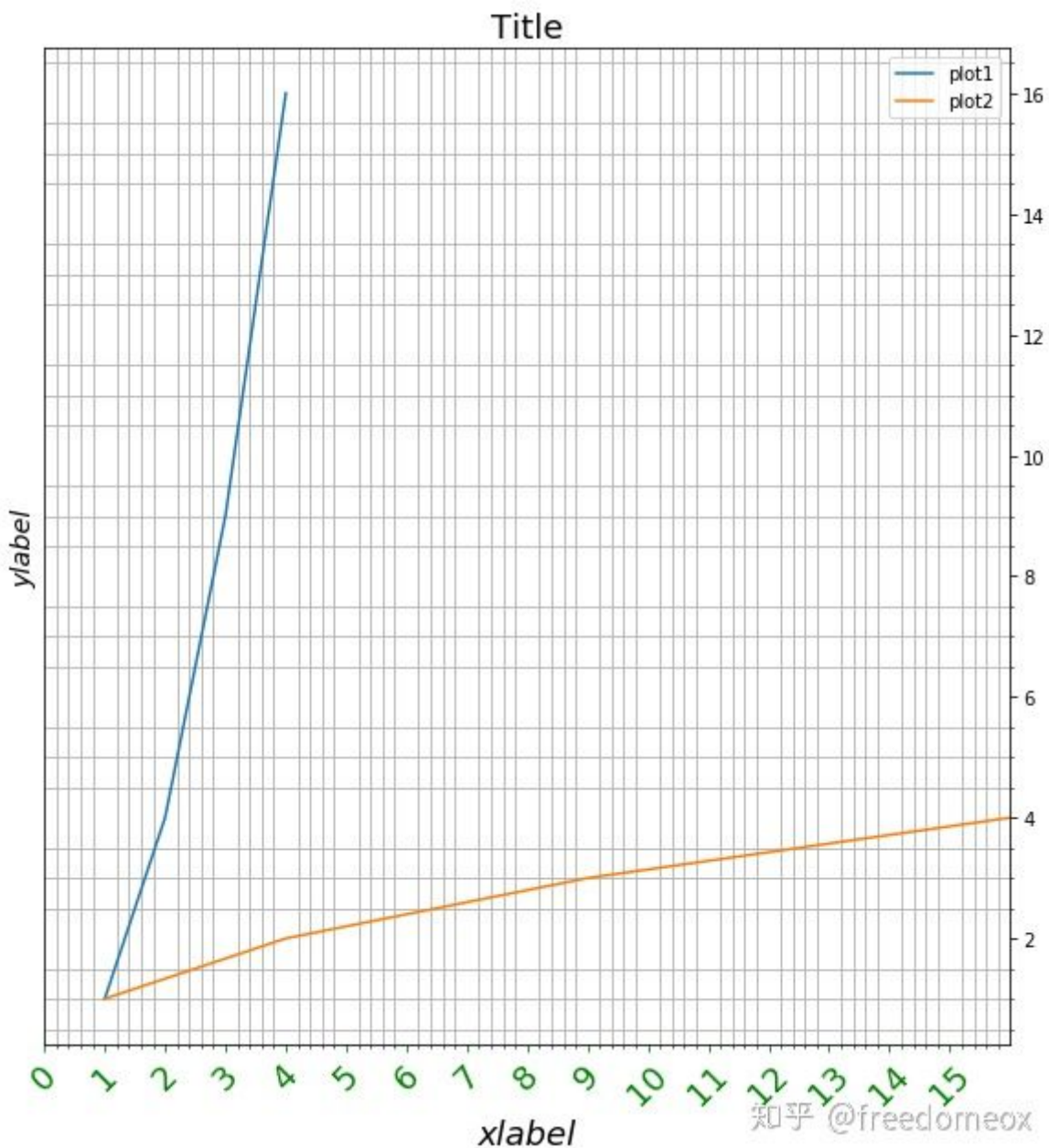
然后是xy坐标轴的一些属性设定, 也是在axes level上完成的

```
ax.set_aspect('equal')
ax.minorticks_on()
ax.set_xlim(0, 16)
ax.grid(which='minor', axis='both')
```

最后是坐标轴tick和细节, 这个在axes.xaxis or axes.yaxis上完成。



```
ax.xaxis.set_ticks(np.arange(start, end,1))  
ax.yaxis.tick_right()
```



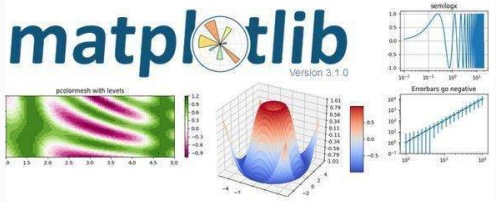
这样一个丑陋的基本图的绘画和编辑就完成了。如果有一些其他的细节调整，在搜索的时候，尽量选择不用plt的答案。原则上来说，plt和ax画图两者是可以互相转换的，然而转换过程让你的代码更复杂，有时还会产生难以理解的bug。因此画图的时候，请坚持使用一种格式。

搞定，吃饭去。

编辑于 2019-11-25



推荐阅读



**python如何使用Matplotlib画图（基础篇）**

雪里挑灯      发表于数据分析

148 条评论

⇌ 切换为时间排序

写下你的评论...



知乎用户

2019-11-24

再讲讲colorbar吧

👍 25



知乎用户

2019-11-24

"from matplotlib.pyplot import plt"? ? ? 是import matplotlib.pyplot as plt 吧[飙泪笑]

👍 33



姚太多啊 (作者) 回复 知乎用户

2019-11-25

还是兄dei你学的扎实啊！改了改了！

👍 11



ze alex 回复 知乎用户

2019-12-29

这两好像都可以的

👍 赞





兄弟写的不错，希望能再多写点东西哦！

👍 赞



白鱼道

2019-12-01

兄弟，🐼皮

👍 赞



程乐齐

2019-12-03

有用！

👍 赞



左不群

2019-12-22

醍醐灌顶啊，感谢🙏

👍 赞



JianJuly

2019-12-23

兄弟，🐼皮

👍 赞



大毛毛虫

2019-12-26

为啥不用asymptote?

👍 赞



Hsin

2019-12-26

是的，花了很多时间泡在精细化画图上之后，我也觉得subplots()才是开局的最佳实践，而且画图的时候官方文档也不能停，随时得查，能想得到的元素微调基本都写在axes和figure的页面下了

👍 5



Zombie110year

2019-12-27

plt 开头的绘图函数会操作最后一个 axes

👍 11



Monstarrrr

2019-12-27

棒！为啥我跟着画画不出x轴数据倾斜的效果

👍 2



金勇夜 回复 Monstarrrr

2020-04-





Issac 回复 Monstarrrr

2020-08-12

换个颜色

赞



拾荒娃

2019-12-27

明白了

赞



傲胖

2019-12-27

这么一讲，感觉疏通了许多。

赞



知乎用户

2019-12-27

通俗易懂，赞！

赞



知乎用户

2019-12-28

其实超像matlab

1



Neo Lee 回复 知乎用户

2019-12-30

本来就是照着 matlab 交互式作图指令做的 [飙泪笑]

1



ucag

2019-12-28

我修正一下对于 axes 命名的问题。如果想要知道 axes 是什么东西，就需要先知道 axis 是什么。axis 指的就是 x 坐标轴，y 坐标轴等，代表的是一根坐标轴。而 axes 在英文里是 axis 的复数形式，也就是说，axes 代表的其实是 figure 当中的一套坐标轴。之所以说一套而不是两个坐标轴，是因为如果你画三维的图，axes 就代表 3 根坐标轴了。所以，在一个 figure 当中，每添加一次 subplot，其实就是添加了一套坐标轴，也就是添加了一个 axes，放在二维坐标里就是你添加了两根坐标轴，分别是 x 轴和 y 轴。所以当你只画一个图的时候，plt.xxx 与 ax.xxx 其实都是作用在相同的图上的。

132



姚太多啊 (作者) 回复 ucag

2019-12-28

我最开始学的时候,也觉得指的是复数坐标轴. 但是后来学到了figure和axes的嵌套关系就觉得用“坐标轴们”来解释有一定的误导性。这里引用一下官方文档：

The Figure is the final image that may contain 1 or more Axes



## 知乎

我的理解是,figure包含axes, axes包含大部分图像元素, 比如坐标轴, tick, 注释文

字, line, 等等。所以我认为axes的命名有一定误导性, 如果简单的理解为坐标轴们是不准确的。

👍 10



知乎用户 回复 姚太多啊 (作者)

2019-12-29

叫坐标系会不会好点?

👍 7

[查看全部 7 条回复](#)



lioneagle

2019-12-28

axes可能指的是坐标系

👍 4



小梁梁 回复 lioneagle

2019-12-28

指轴域

👍 5



知乎用户

2019-12-28

以前用python, 最近疯狂写matlab GUI, 发现其实他们在画图上极为一致。figure, axes的概念, 完全一样。figure上可以有各种东西, 比如panel, groupbutton, 各种uicontrol, 当然也有 axes。它们各有各的属性, 明白了这些, 其实就简单了, 需要的时候, 去帮助里看一下都有哪些属性就行了

👍 1



Vanadium 回复 知乎用户

2019-12-29

matplotlib好像就是仿照matlab的绘图方式做的库

👍 2



知乎用户 回复 Vanadium

2019-12-29

应该是的, 名字都那么像

👍 赞



知乎用户

2019-12-28

👍

👍 赞



为什么没让我早点看到这篇又早。。。

👍 1



Airlet 回复 甲乙丙丁

2019-12-30

朝闻道夕死可矣 🐼

👍 4



甲乙丙丁 回复 Airlet

2019-12-30

不不不，我还要用它画图不能死

👍 2

展开其他 1 条回复

