



南通师范高等专科学校
Nantong Normal College



Matplotlib数据可视化之 基础知识

朱亚林

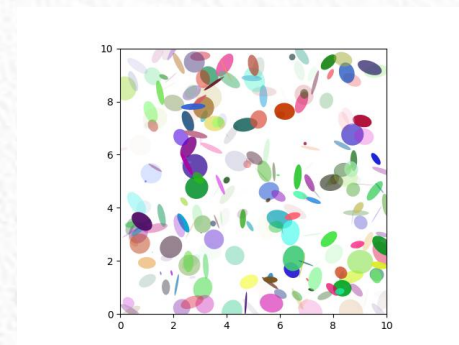
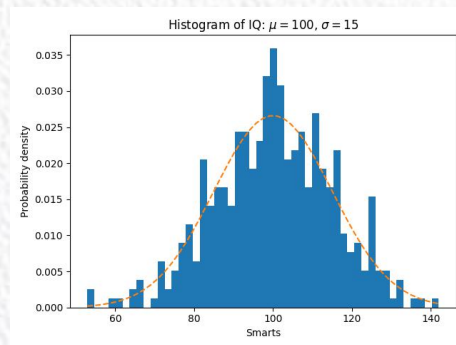
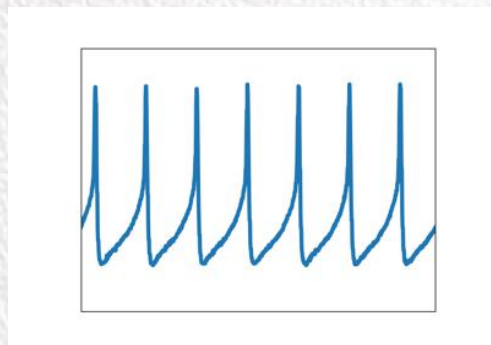




Matplotlib简介

- 丨 Matplotlib 是 Python 的一个扩展程序库
- 丨 提供了大量的绘图组件用于绘制图表,图表可以保存成多种格式
- 丨 包含一个互动的图表控制平台,可以跨平台方便的修改图表样式

matplotlib



一个简单的画图实例

```
>>> import numpy as np

>>> from matplotlib import pyplot as plt

>>> x = np.arange(1,11)

>>> y = 2 * x + 5

>>> plt.title("Matplotlib demo")

>>> plt.xlabel("x axis caption")

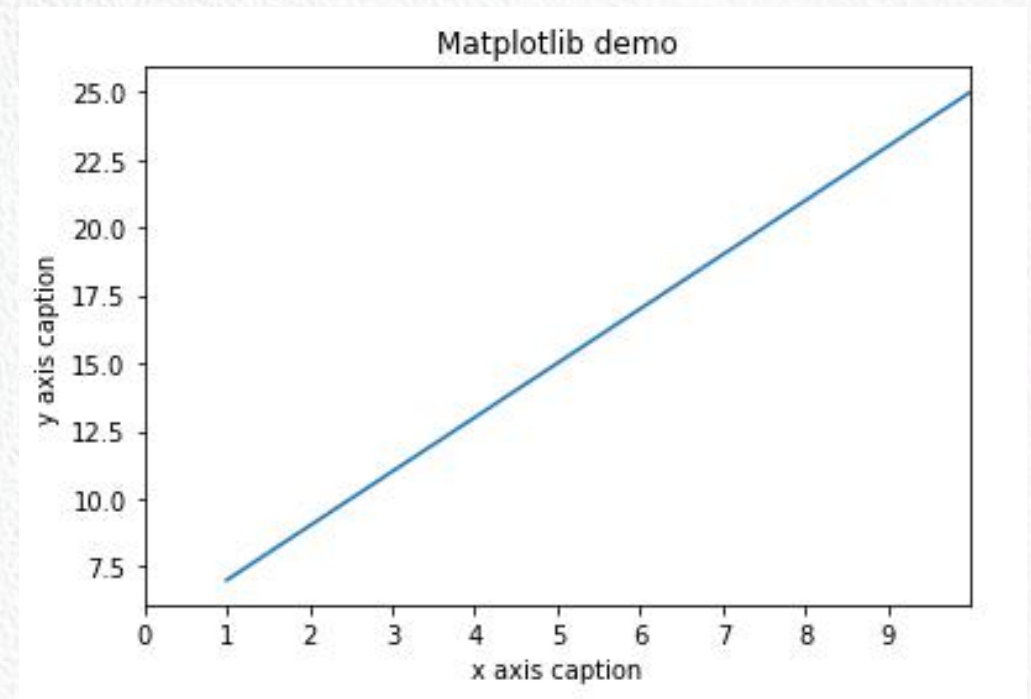
>>> plt.xlim(0,10)

>>> plt.xticks(np.arange(0,10))

>>> plt.ylabel("y axis caption")

>>> plt.plot(x,y)

>>> plt.show()
```



- ▮ plt.title()
- ▮ plt.xlabel()
- ▮ plt.ylabel()
- ▮ plt.xlim()
- ▮ plt.xticks()
- ▮ plt.plot()
- ▮ plt.show()

标题

范围

轴标签

显示

刻度

绘图

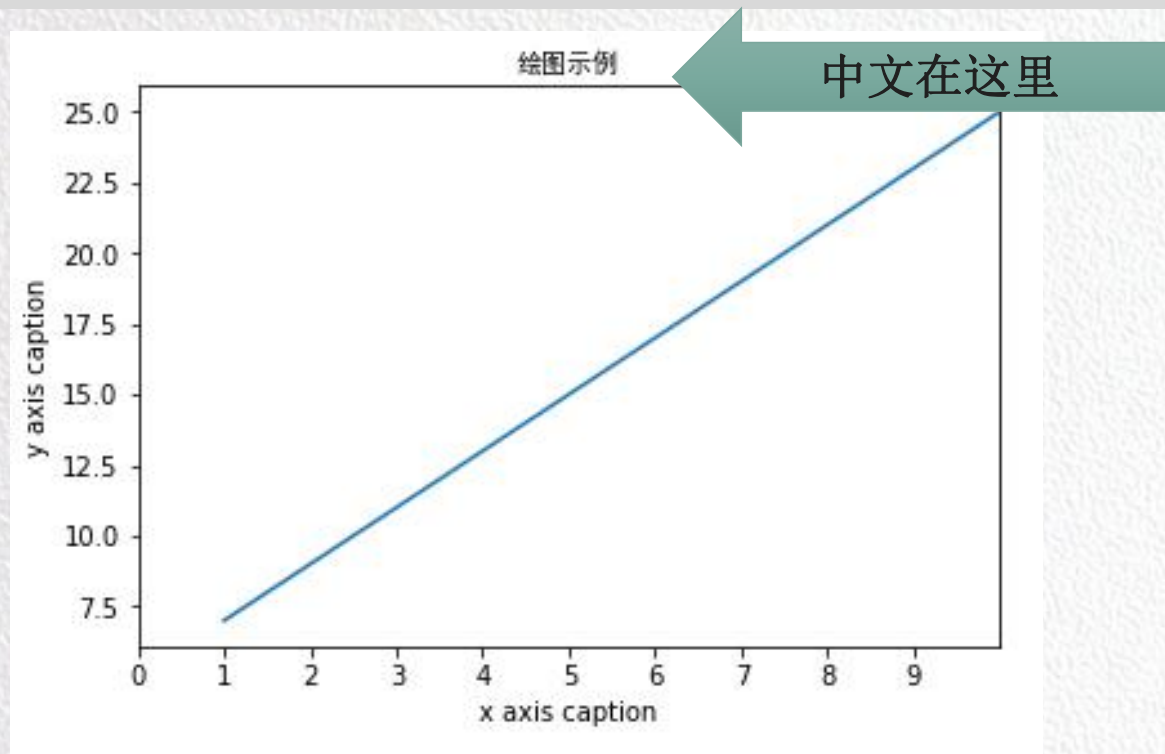
怎样在图表中显示中文？

怎样显示图例？

怎样保存图片？

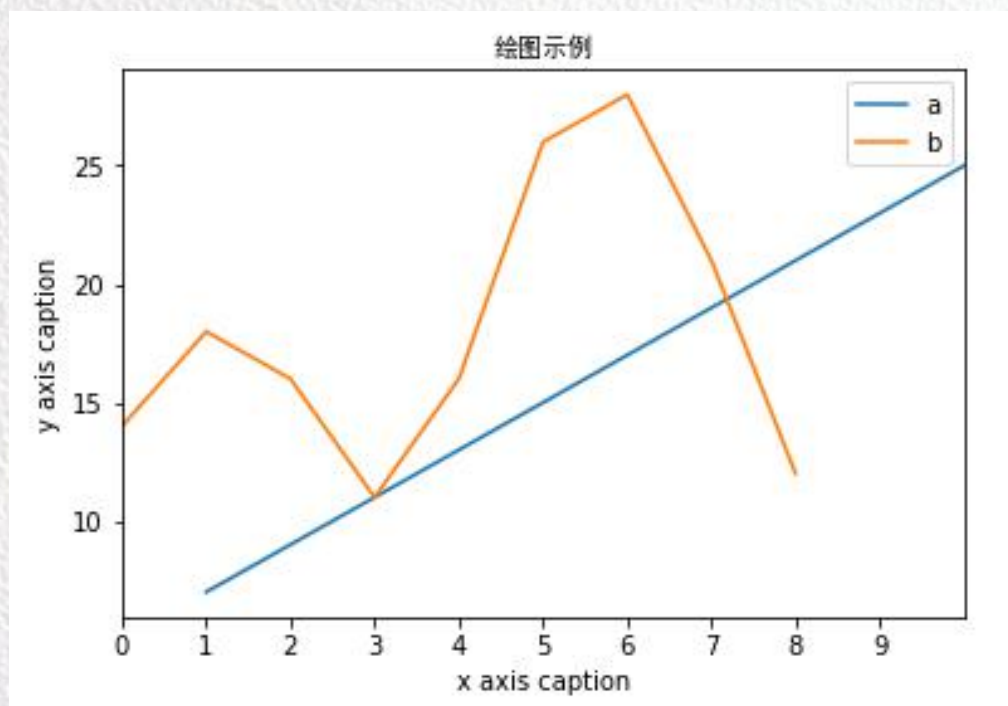
再复杂一点——怎样在图表中显示中文？

- `import matplotlib`
- `zhfont1 = matplotlib.font_manager.FontProperties(fname="simhei.ttf")`
- `plt.title("绘图示例", fontproperties=zhfont1)`



再复杂一点——怎样在图表中设置图例？

- l1,=plt.plot(x,y)
- l2,=plt.plot(a)
- plt.legend(handles = [l1, l2,], labels = ['a', 'b'], loc = 'best')



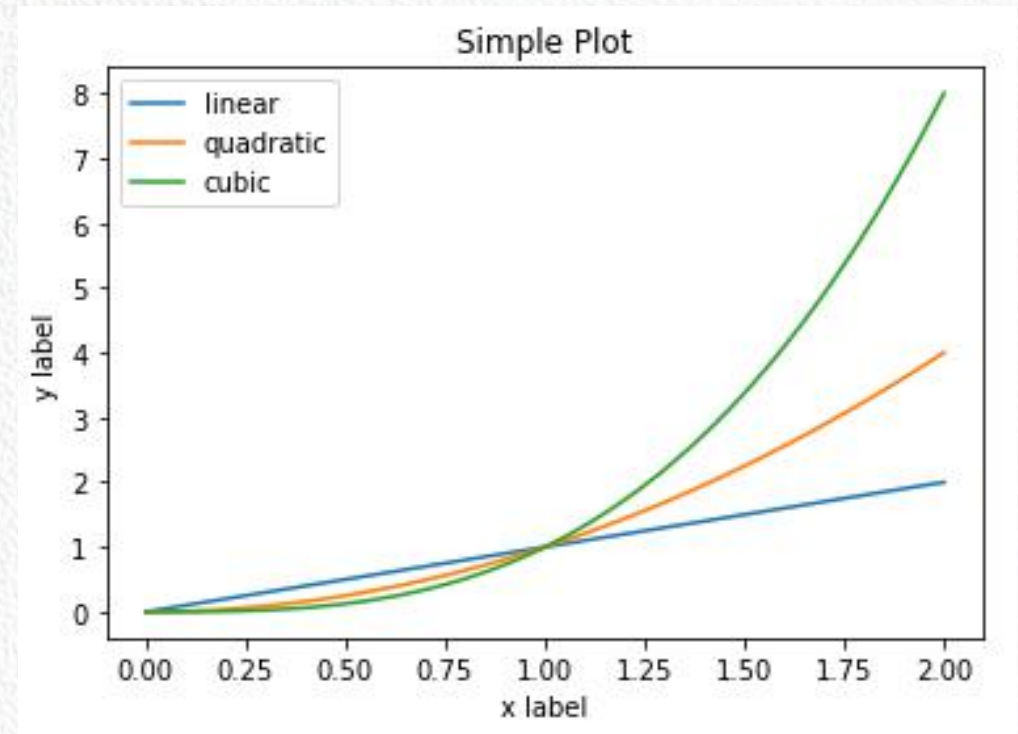
再复杂一点——怎样保存图片？

```
plt.savefig("path-of-file")
```



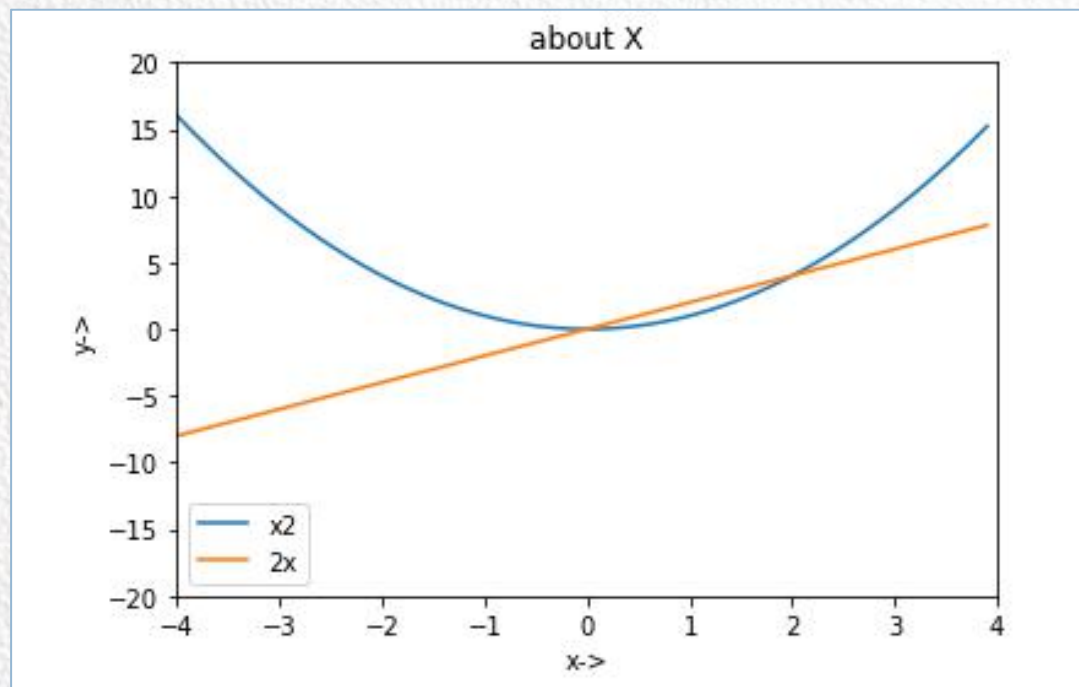
一个简单的综合实例

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 2, 100)
plt.plot(x, x, label='linear')
plt.plot(x, x**2, label='quadratic')
plt.plot(x, x**3, label='cubic')
plt.xlabel('x label')
plt.ylabel('y label')
plt.title("Simple Plot")
plt.legend()
plt.show()
```



一个小练习

- 在一个直角坐标系中显示 x^2 与 $2x$ 的图像。要求：
- 设置标题为about X
- 设置图例
- 设置图形的表值范围：x轴-4~4，y轴-20~20
- 效果如下图所示，最后保存到c盘根目录下



让图像更专业

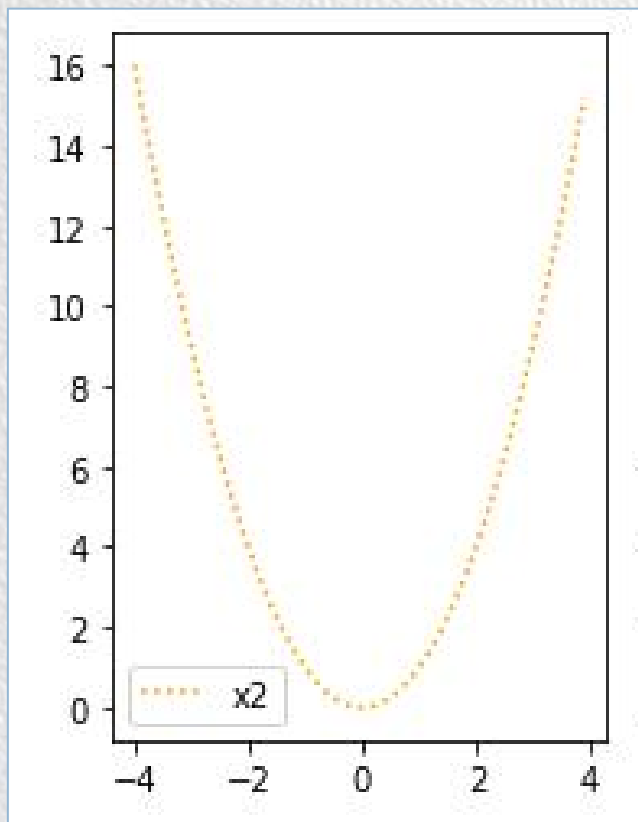
设置线型，让多个线条更有区分度

设置多合一显示

增加网格参照

添加文本

更专业一点——怎样设置线型

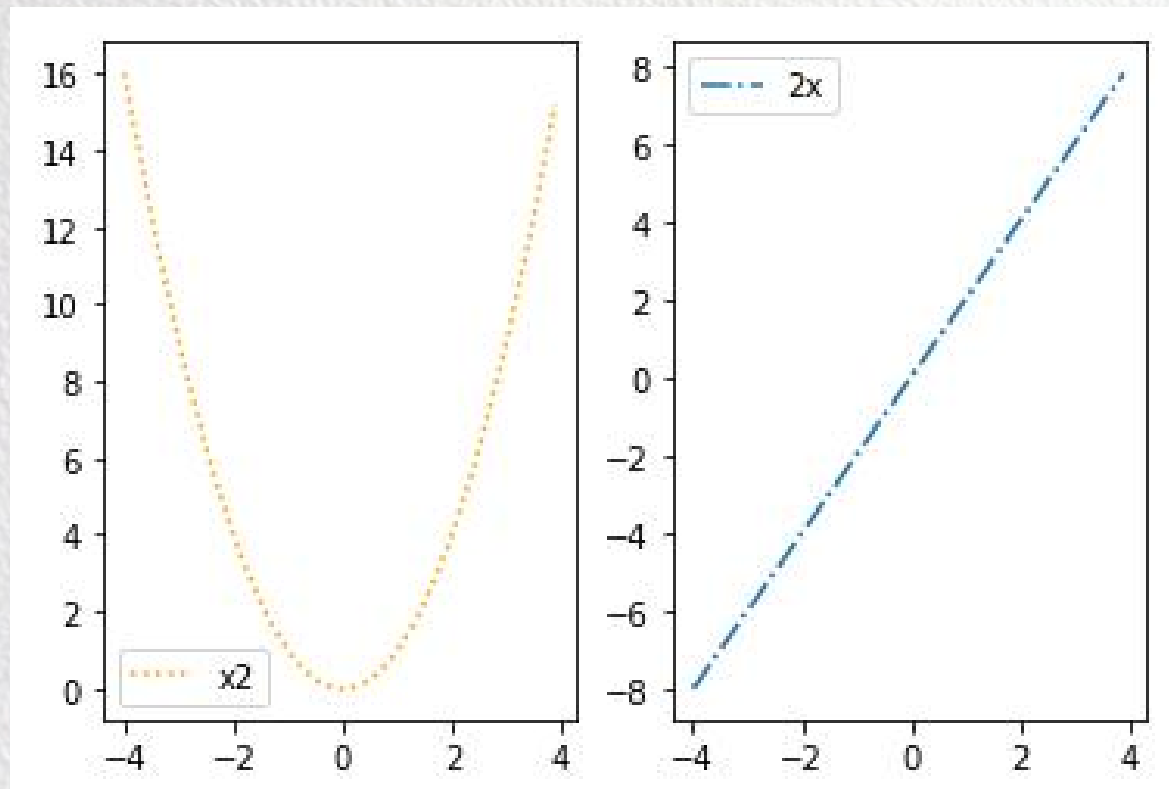


<-- 比如这样

这里是攻略-->



```
plt.plot(x,y1,linestyle=':',label='x2')
```

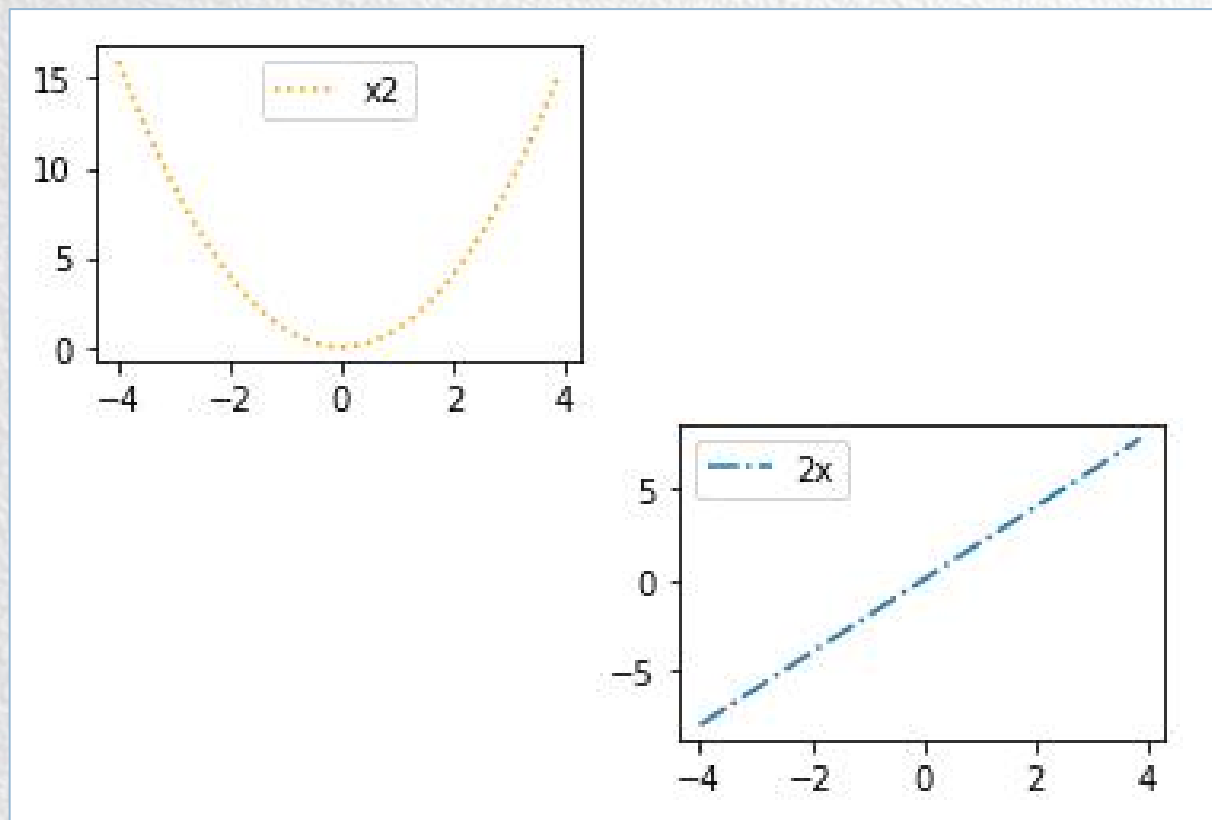


不要怀疑，这就是一张图。

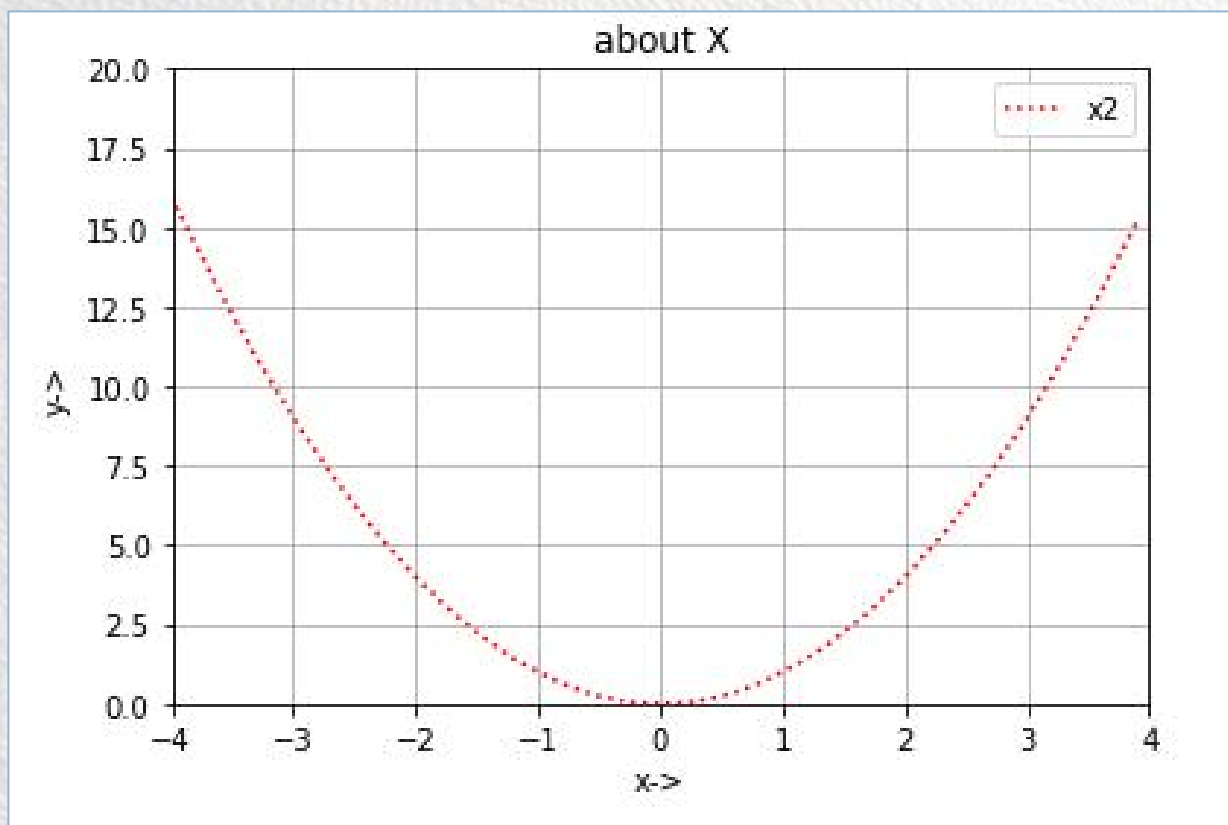
使用subplot工具

`subplot()`函数有三个参数，分别表示这样的含义。第1个参数表示图像分为几行，第2个参数表示图像分为几列，第3个参数表示显示在第几个位置。比如前图：

```
plt.subplot(121)
plt.plot(x,y1,linestyle=':',color='orange',
label='x2')
plt.legend()
plt.subplot(122)
plt.plot(x,y2,linestyle='-.',label='2x')
plt.legend()
```

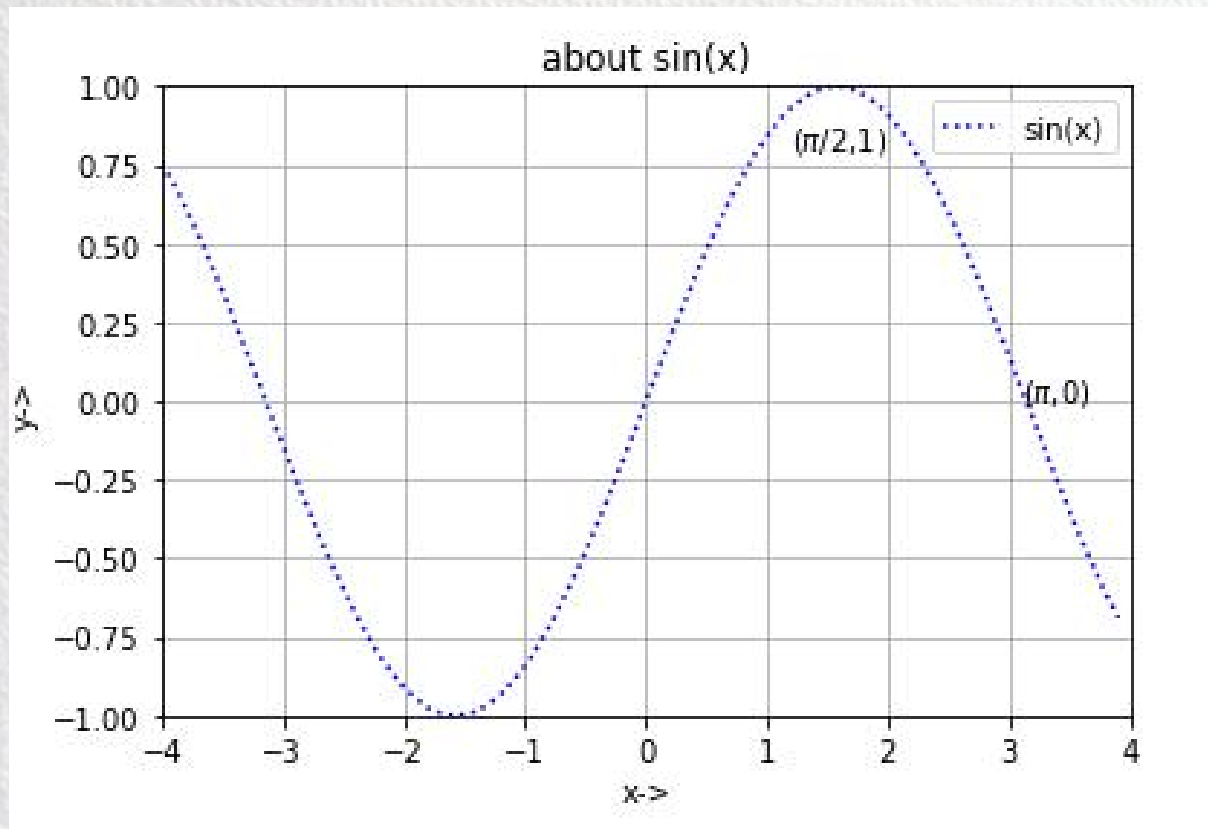


试一试：
这个subplot该怎么设置



使用grid()函数

给图像加上plt.grid()函数，其中设置参数为True，即可增加网格参照。



使用text()函数

给图像加上`plt.text()`函数，就可以在指定位置设置相应的字符标签。

`text()`函数中包含以下几个参数，分别是 (x,y,s) 。 x 和 y 表示`text`在坐标轴内显示的位置， s 表示要显示的内容。

```
plt.text(1.2,0.8,'( $\pi/2,1$ )')
```

```
plt.text(3.1,0,'( $\pi,0$ )')
```