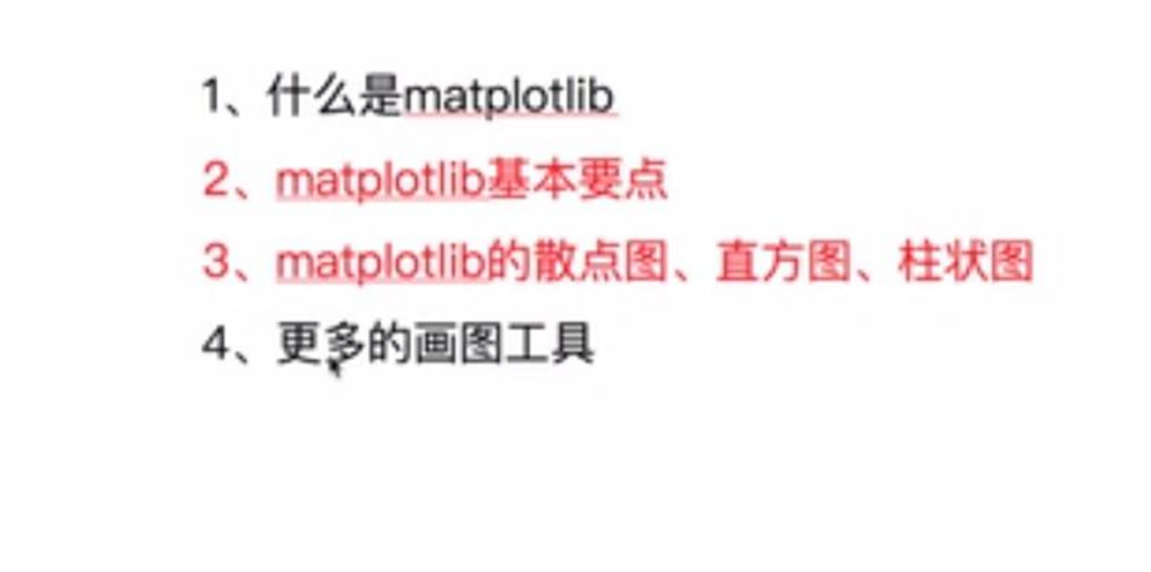
Matplotlib 总结

0. Matplotlib 学习框架



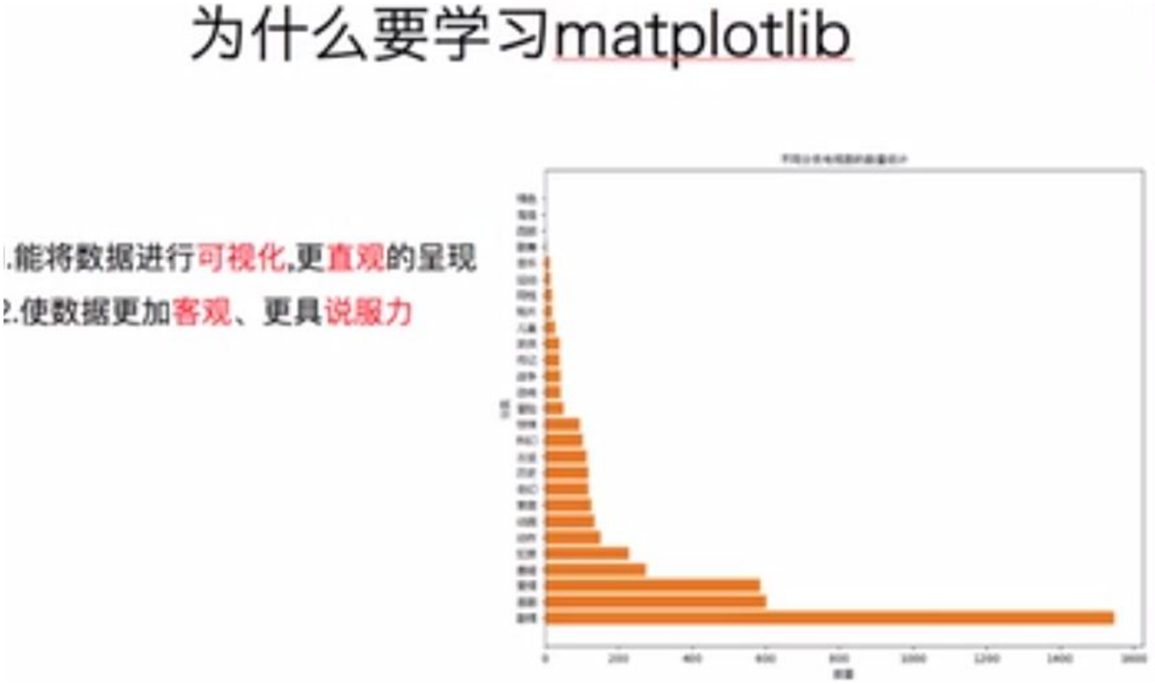
1、什么是matplotlib

2、matplotlib基本要点

3、matplotlib的散点图、直方图、柱状图

4、更多的画图工具

一． 什么是 matplotlib



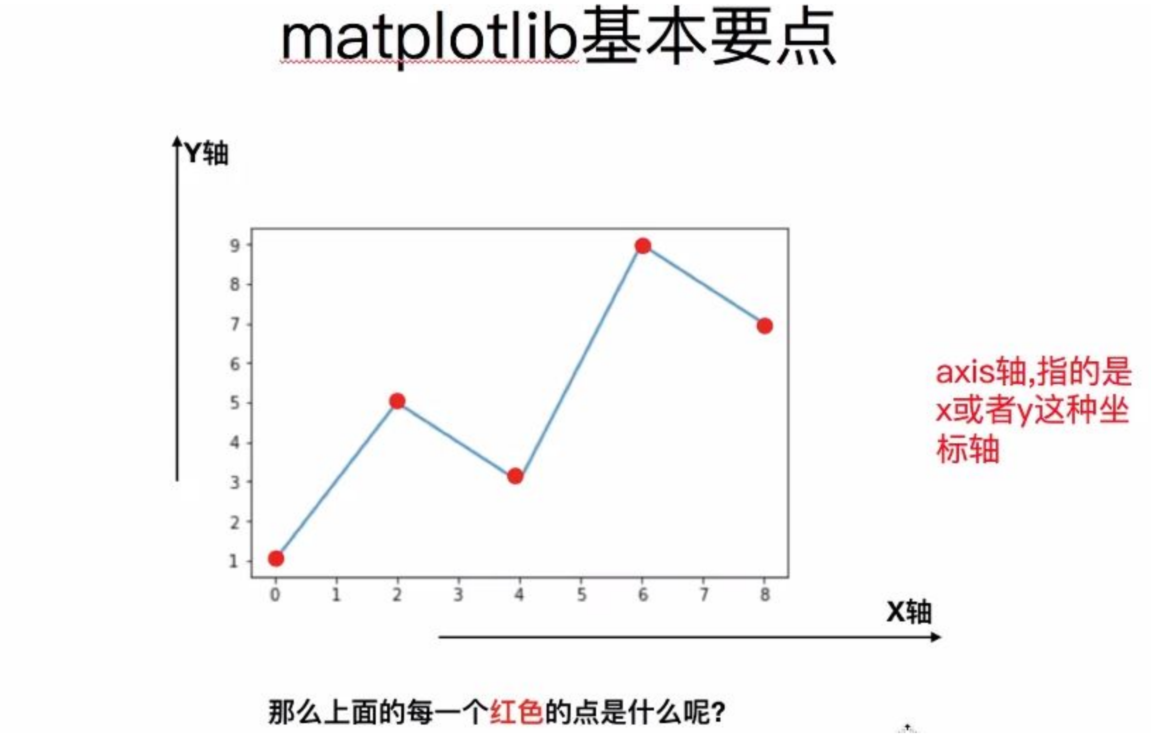
为什么要学习matplotlib

1.能将数据进行可视化,更直观的呈现

2.使数据更加客观、更具说服力

什么是matplotlib

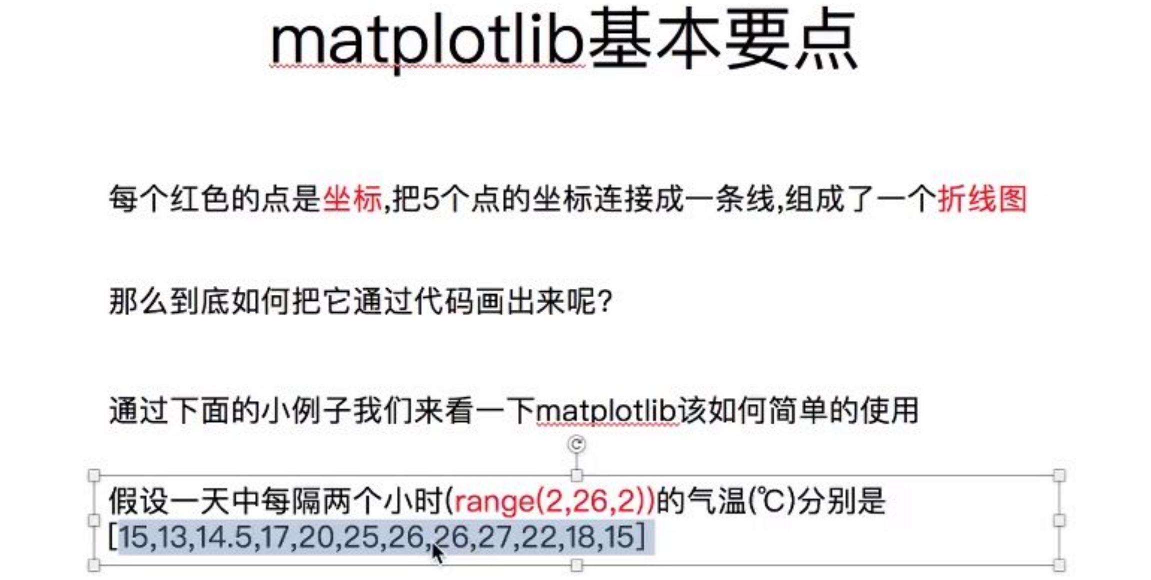
matplotlib: 最流行的Python底层绘图库，主要做数据可视化图表,名字取材于MATLAB，模仿MATLAB构建



matplotlib基本要点

axis轴,指的是x或者y这种坐标轴

那么上面的每一个红色的点是什么呢?



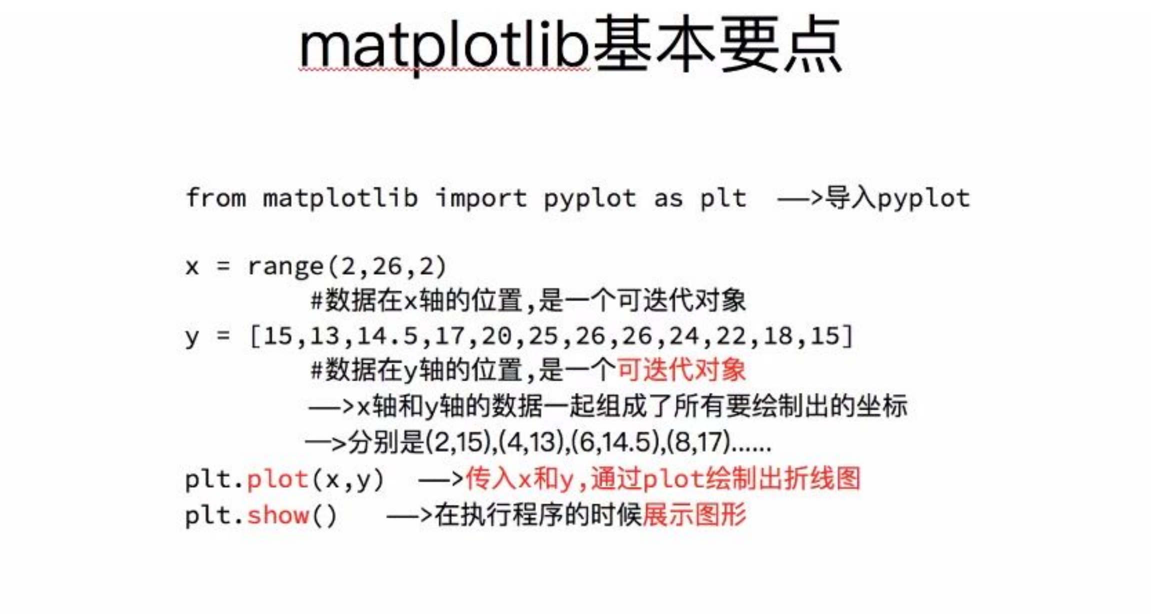
matplotlib基本要点

每个红色的点是坐标,把5个点的坐标连接成一条线,组成了一个折线图

那么到底如何把它通过代码画出来呢?

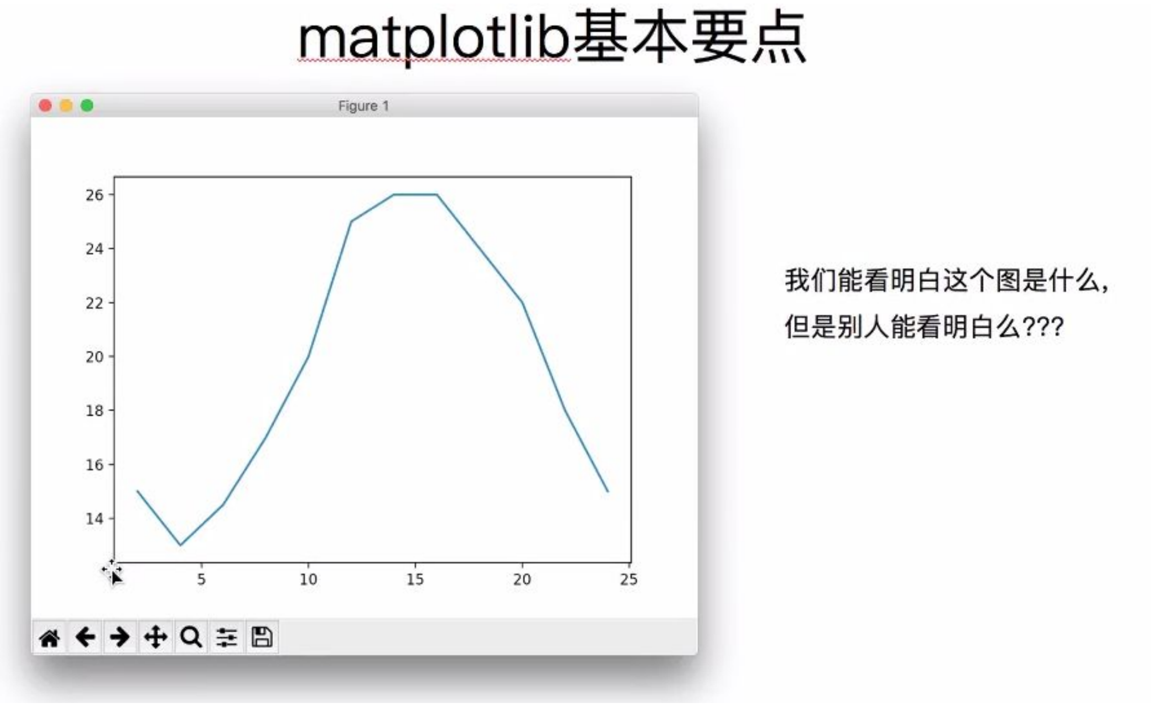
通过下面的小例子我们来看一下matplotlib该如何简单的使用

假设一天中每隔两个小时(range(2,26,2))的气温(℃)分别是
[15,13,14.5,17,20,25,26,26,27,22,18,15]



matplotlib基本要点

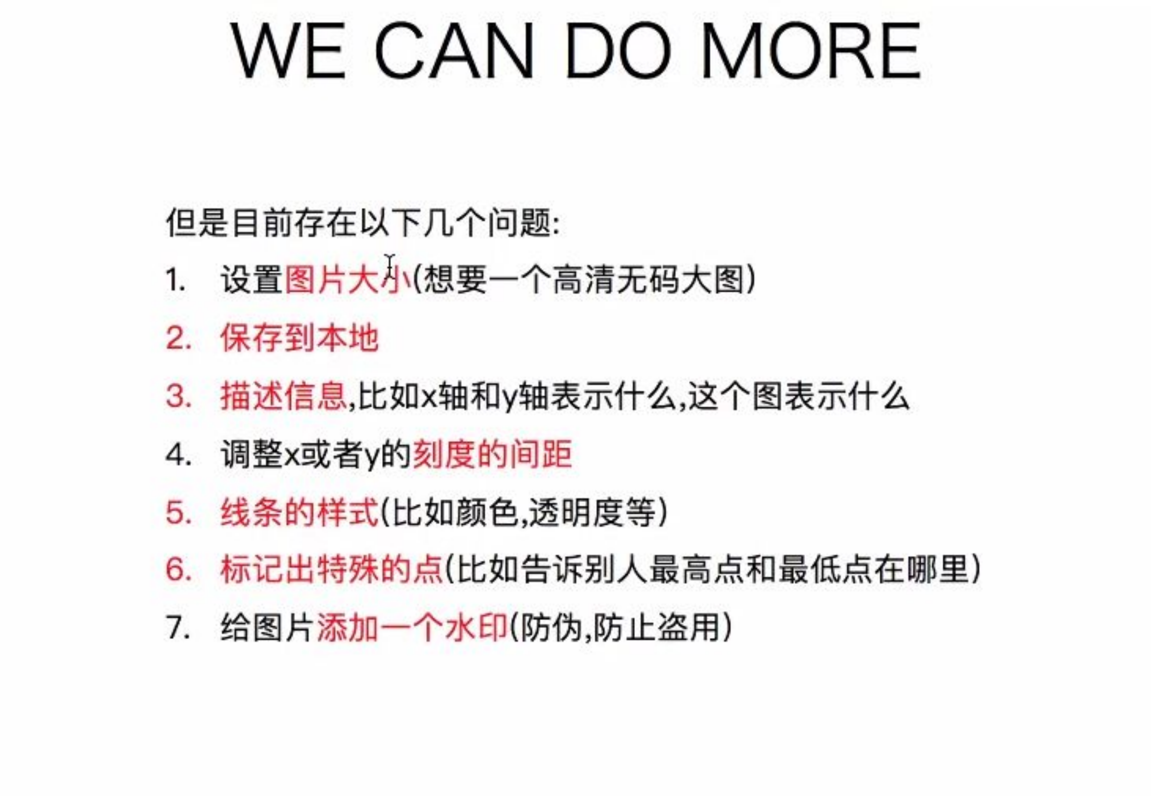
```
from matplotlib import pyplot as plt  —>导入pyplot

x = range(2,26,2)
        #数据在x轴的位置,是一个可迭代对象
y = [15,13,14.5,17,20,25,26,26,24,22,18,15]
        #数据在y轴的位置,是一个可迭代对象
        —>x轴和y轴的数据一起组成了所有要绘制出的坐标
        —>分别是(2,15),(4,13),(6,14.5),(8,17)......
plt.plot(x,y)  —>传入x和y,通过plot绘制出折线图
plt.show()   —>在执行程序的时候展示图形
```



matplotlib基本要点

我们能看明白这个图是什么，
但是别人能看明白么???

二. Matplotlib 常用设置（折线图）

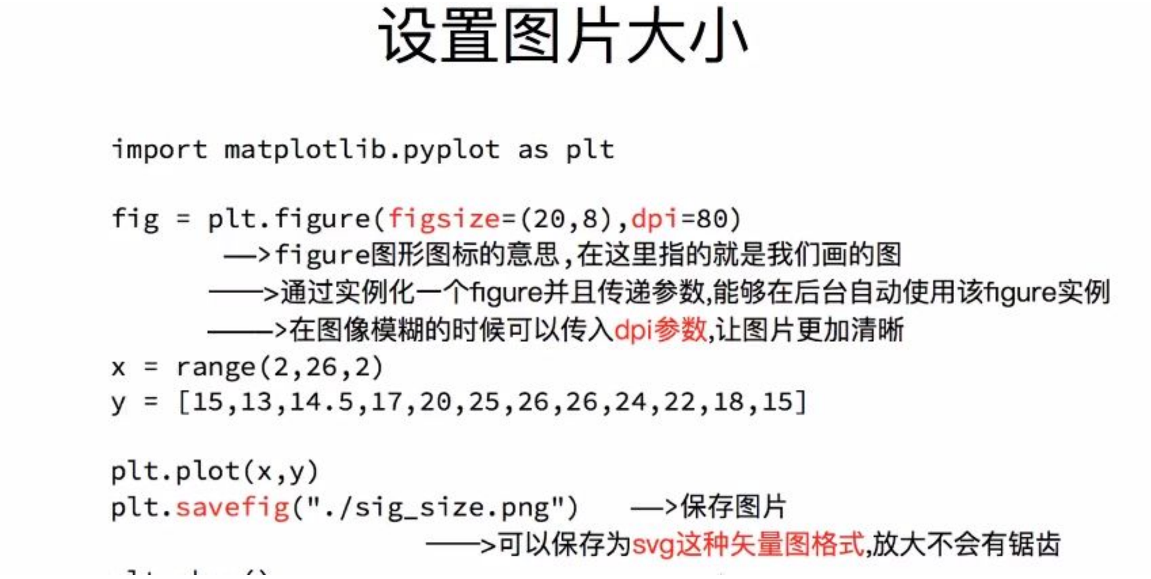


WE CAN DO MORE

但是目前存在以下几个问题:

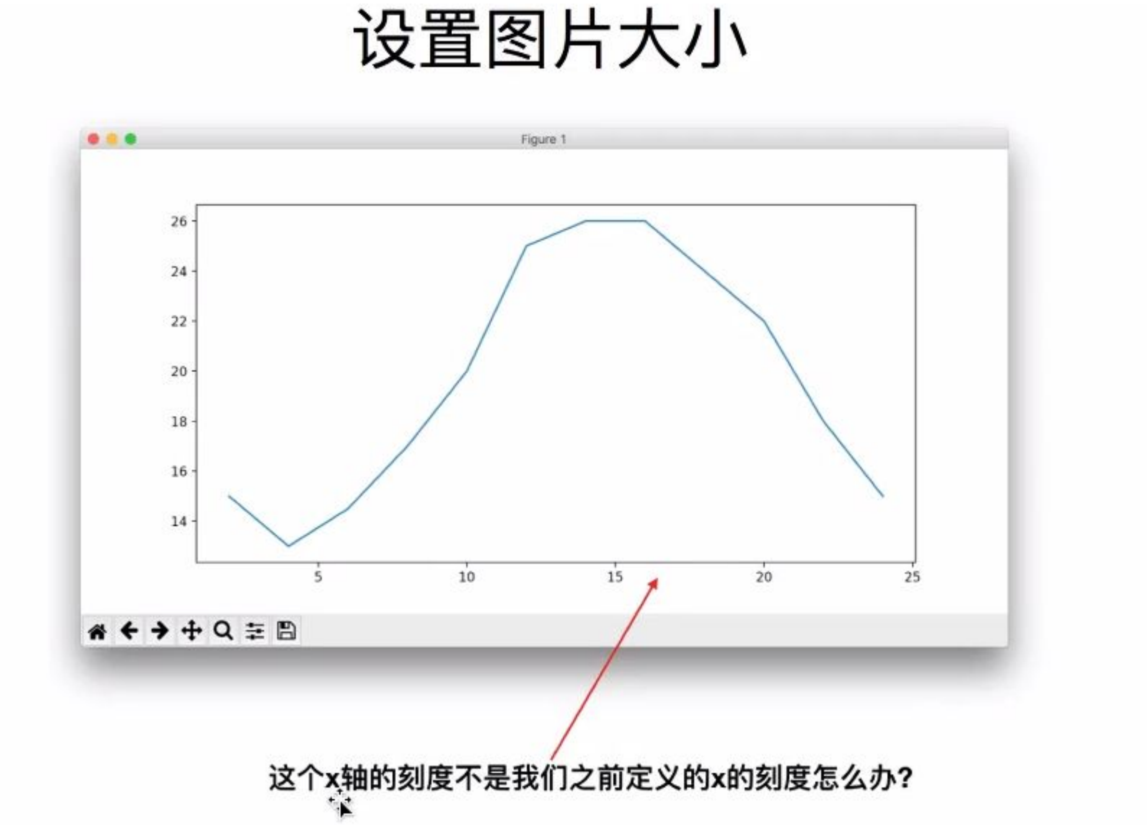
1. 设置图片大小(想要一个高清无码大图)
2. 保存到本地
3. 描述信息,比如x轴和y轴表示什么,这个图表示什么
4. 调整x或者y的刻度的间距
5. 线条的样式(比如颜色,透明度等)
6. 标记出特殊的点(比如告诉别人最高点和最低点在哪里)
7. 给图片添加一个水印(防伪,防止盗用)

2-1. 设置图片大小



设置图片大小

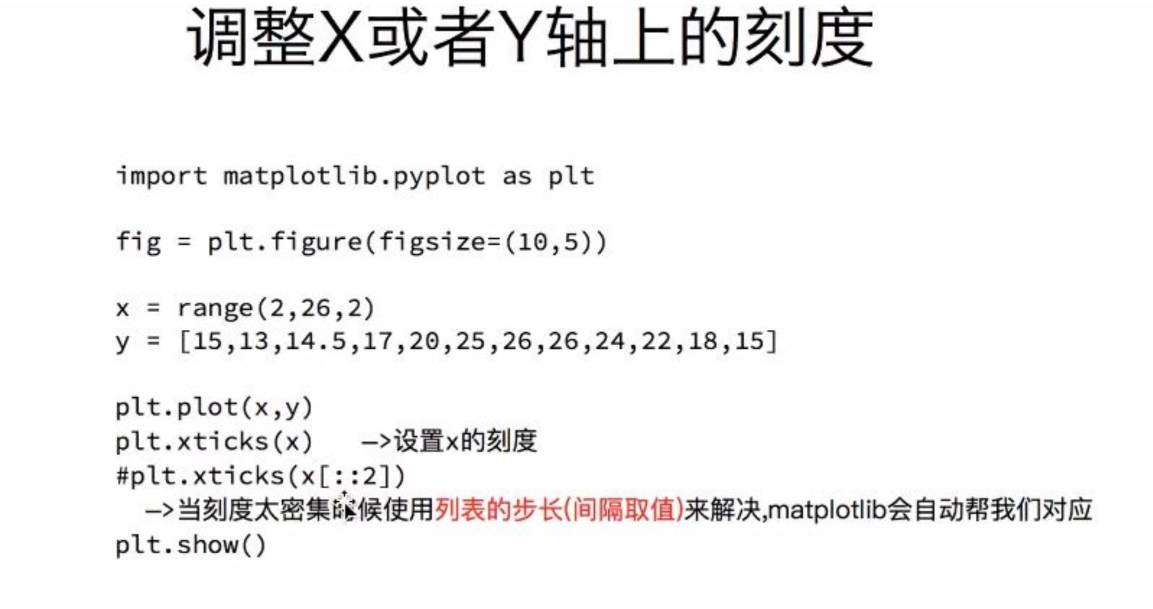
```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(20,8),dpi=80)
         —>figure图形图标的意思,在这里指的就是我们画的图
        ——>通过实例化一个figure并且传递参数,能够在后台自动使用该figure实例
        ———>在图像模糊的时候可以传入dpi参数,让图片更加清晰
x = range(2,26,2)
y = [15,13,14.5,17,20,25,26,26,24,22,18,15]

plt.plot(x,y)
plt.savefig("./sig_size.png")    —>保存图片
                            ——>可以保存为svg这种矢量图格式,放大不会有锯齿
```



设置图片大小

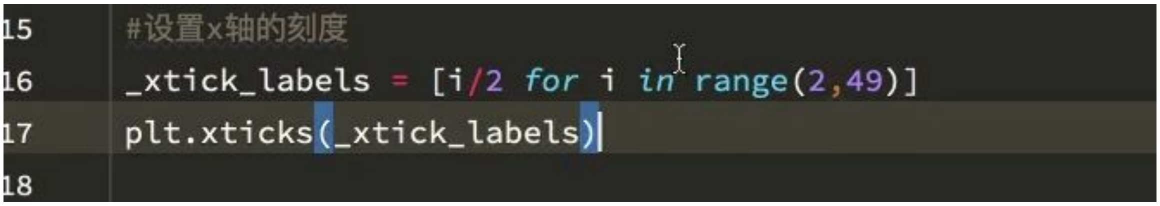
这个x轴的刻度不是我们之前定义的x的刻度怎么办?

2-2．调整x 轴或者y 轴上的刻度

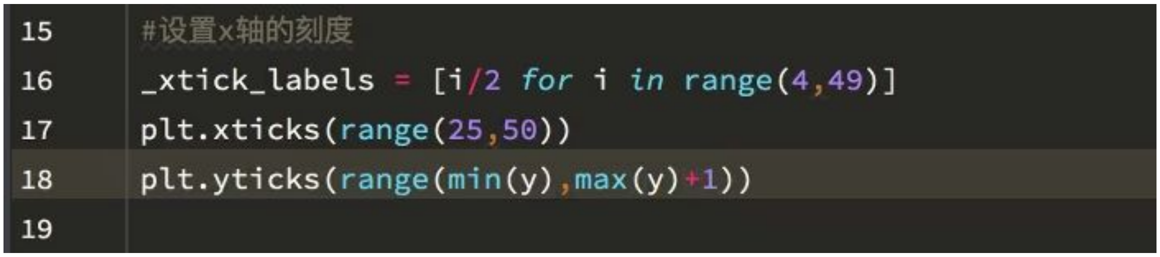


调整X或者Y轴上的刻度

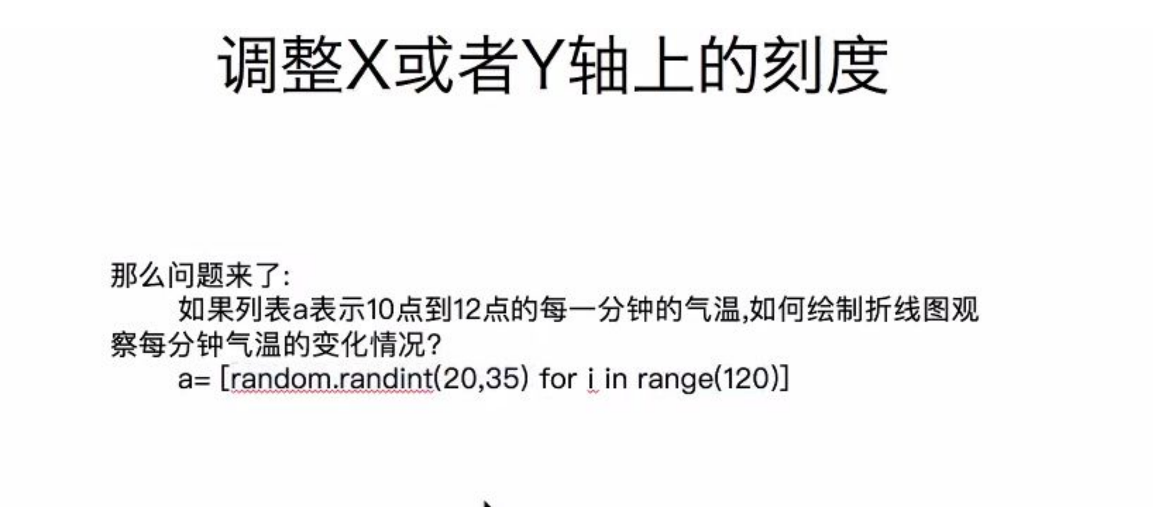
```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(10,5))

x = range(2,26,2)
y = [15,13,14.5,17,20,25,26,26,24,22,18,15]

plt.plot(x,y)
plt.xticks(x)    —>设置x的刻度
#plt.xticks(x[::2])
  —>当刻度太密集时候使用列表的步长(间隔取值)来解决,matplotlib会自动帮我们对应
plt.show()
```



```
#设置x轴的刻度
_xtick_labels = [i/2 for i in range(2,49)]
plt.xticks(_xtick_labels)
```

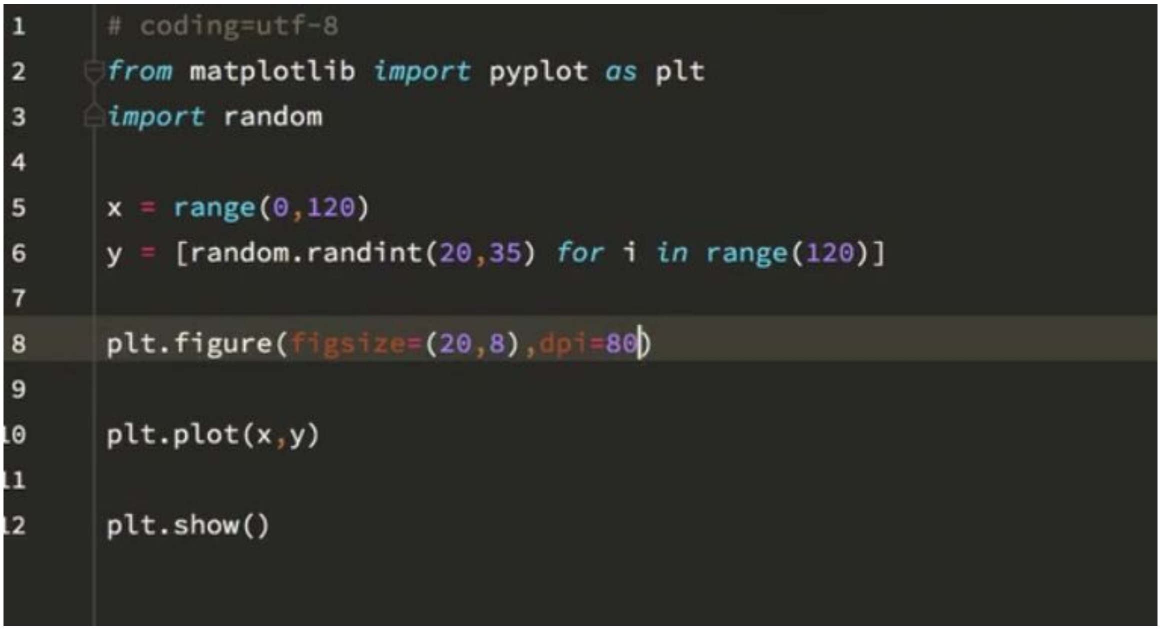


```
#设置x轴的刻度
_xtick_labels = [i/2 for i in range(4,49)]
plt.xticks(range(25,50))
plt.yticks(range(min(y),max(y)+1))
```

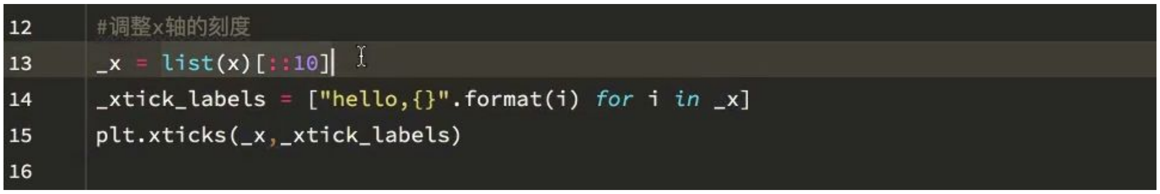


调整X或者Y轴上的刻度

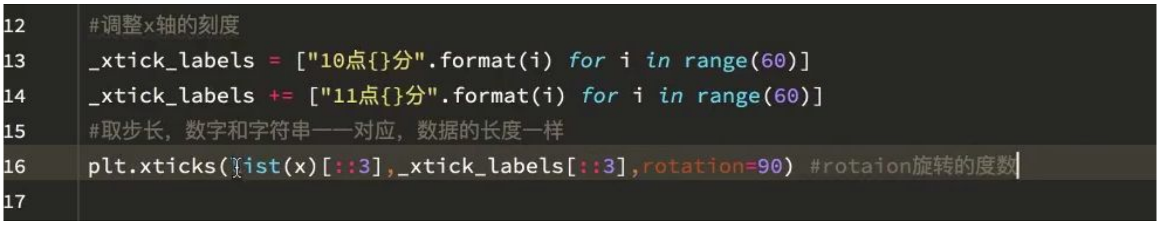
那么问题来了:
  如果列表a表示10点到12点的每一分钟的气温,如何绘制折线图观察每分钟气温的变化情况?
  a= [random.randint(20,35) for i in range(120)]



```python
# coding=utf-8
from matplotlib import pyplot as plt
import random

x = range(0,120)
y = [random.randint(20,35) for i in range(120)]

plt.figure(figsize=(20,8),dpi=80)

plt.plot(x,y)

plt.show()
```

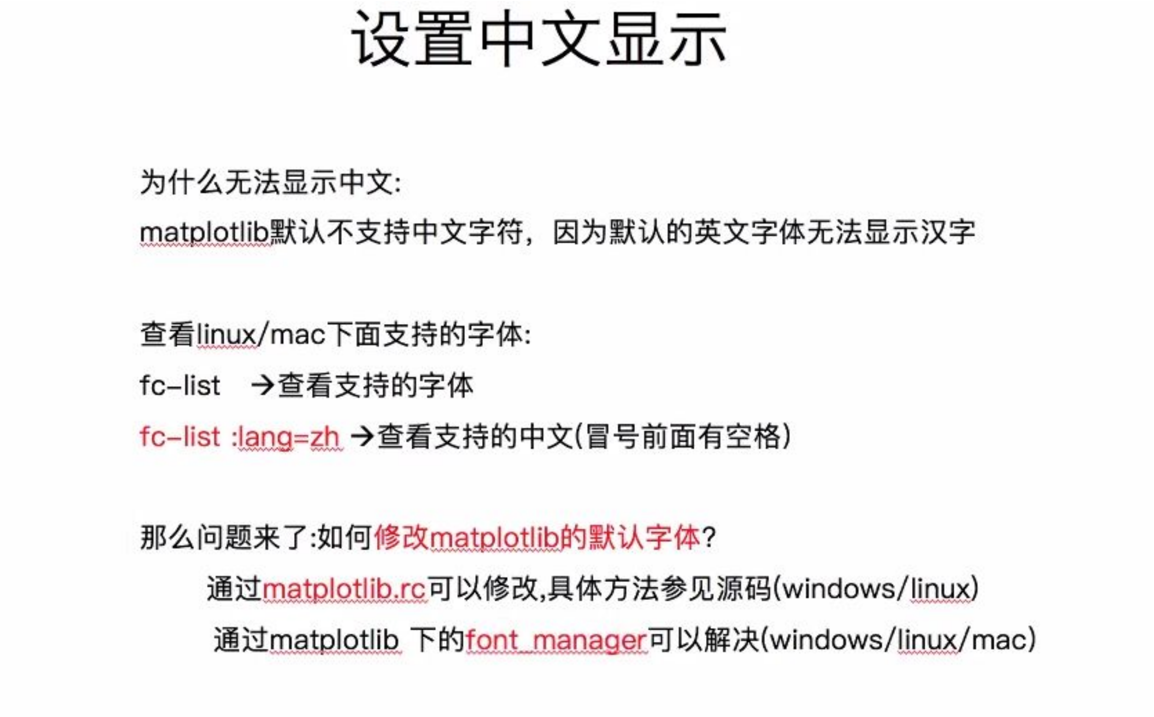


```python
#调整x轴的刻度
_x = list(x)[::10]
_xtick_labels = ["hello,{}".format(i) for i in _x]
plt.xticks(_x,_xtick_labels)

```



```python
#调整x轴的刻度
_xtick_labels = ["10点{}分".format(i) for i in range(60)]
_xtick_labels += ["11点{}分".format(i) for i in range(60)]
#取步长，数字和字符串一一对应，数据的长度一样
plt.xticks(list(x)[::3],_xtick_labels[::3],rotation=90) #rotaion旋转的度数

```

2-3．设置中文显示



设置中文显示

为什么无法显示中文:

matplotlib默认不支持中文字符，因为默认的英文字体无法显示汉字

查看linux/mac下面支持的字体:

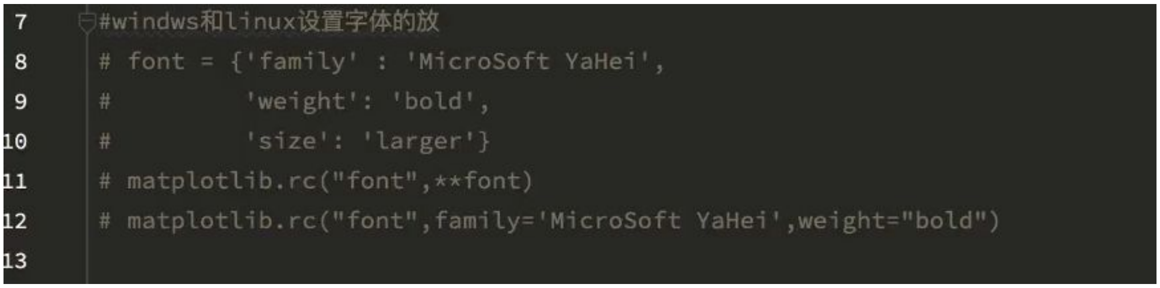
fc-list →查看支持的字体

fc-list :lang=zh →查看支持的中文(冒号前面有空格)

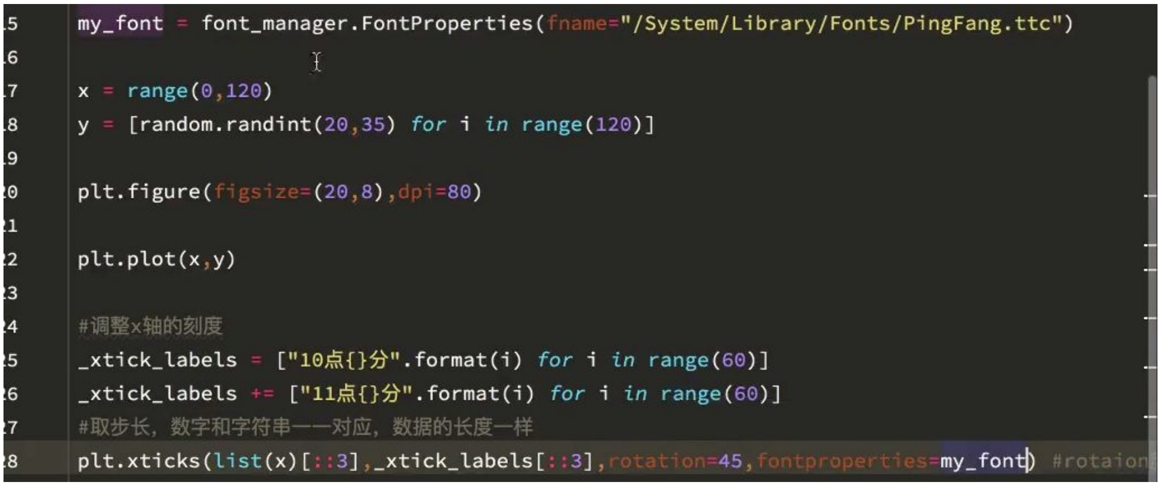
那么问题来了:如何修改matplotlib的默认字体?

通过matplotlib.rc可以修改,具体方法参见源码(windows/linux)

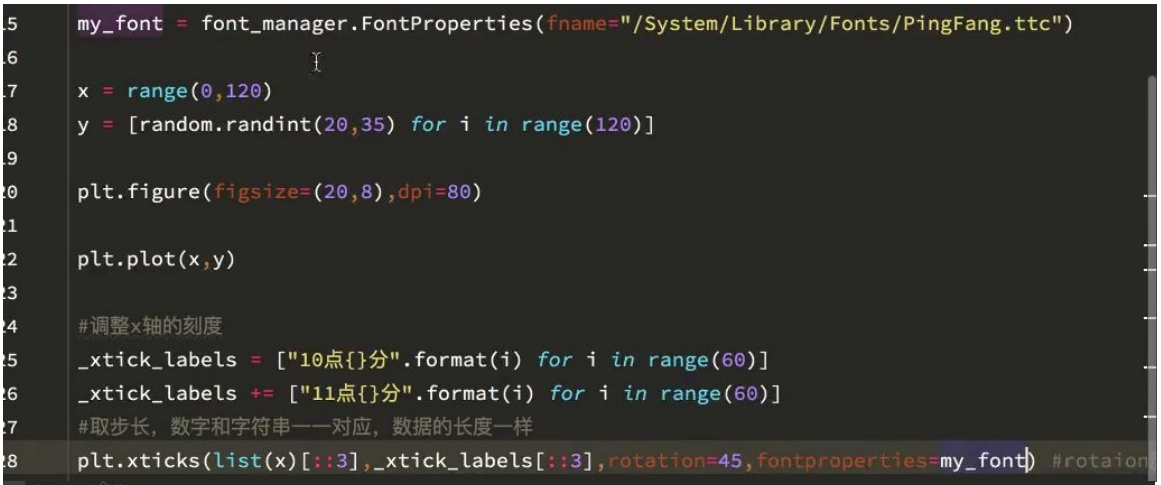
通过matplotlib 下的font_manager可以解决(windows/linux/mac)



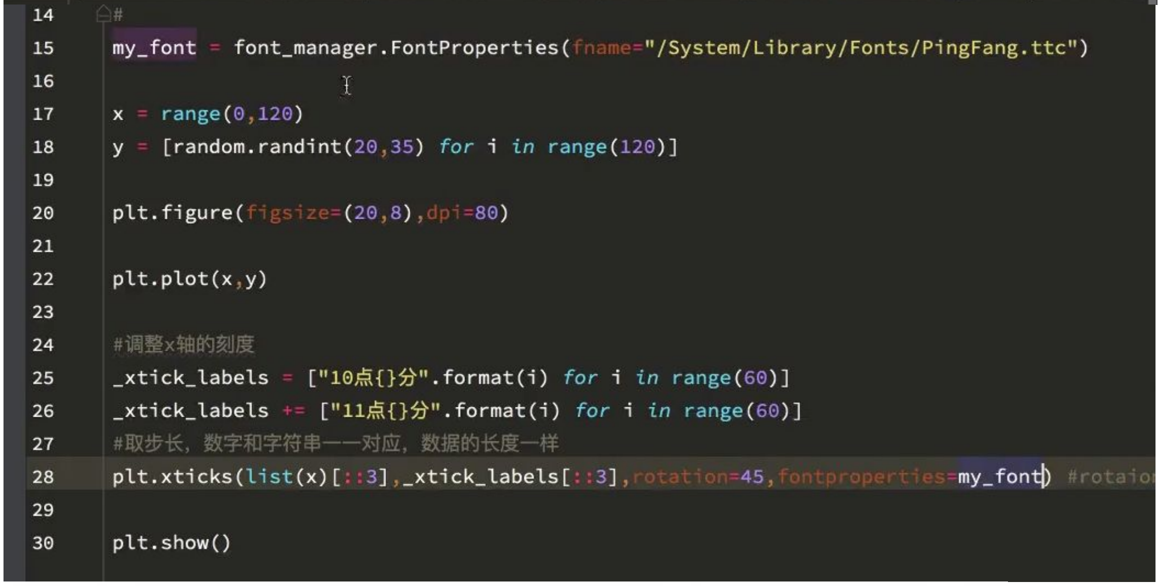
```
#windws和linux设置字体的放
# font = {'family' : 'MicroSoft YaHei',
#         'weight': 'bold',
#         'size': 'larger'}
# matplotlib.rc("font",**font)
# matplotlib.rc("font",family='MicroSoft YaHei',weight="bold")

```



```
my_font = font_manager.FontProperties(fname="/System/Library/Fonts/PingFang.ttc")

x = range(0,120)
y = [random.randint(20,35) for i in range(120)]

plt.figure(figsize=(20,8),dpi=80)

plt.plot(x,y)

#调整x轴的刻度
_xtick_labels = ["10点{}分".format(i) for i in range(60)]
_xtick_labels += ["11点{}分".format(i) for i in range(60)]
#取步长，数字和字符串一一对应，数据的长度一样
plt.xticks(list(x)[::3],_xtick_labels[::3],rotation=45,fontproperties=my_font) #rotaion
```

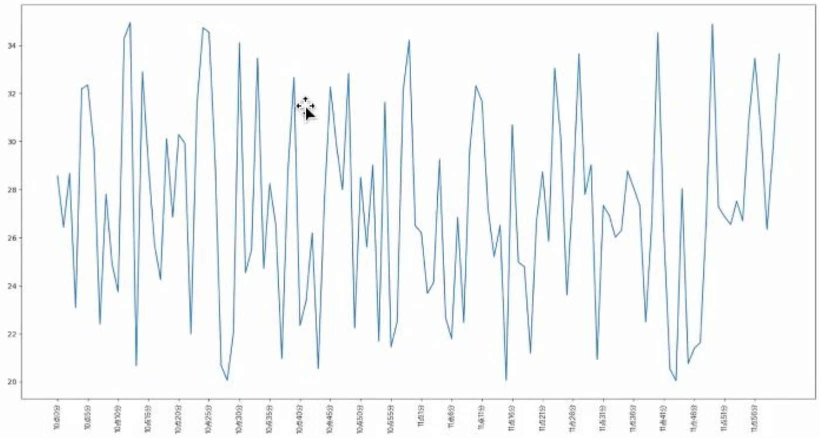


```
my_font = font_manager.FontProperties(fname="/System/Library/Fonts/PingFang.ttc")

x = range(0,120)
y = [random.randint(20,35) for i in range(120)]

plt.figure(figsize=(20,8),dpi=80)

plt.plot(x,y)

#调整x轴的刻度
_xtick_labels = ["10点{}分".format(i) for i in range(60)]
_xtick_labels += ["11点{}分".format(i) for i in range(60)]
#取步长，数字和字符串一一对应，数据的长度一样
plt.xticks(list(x)[::3],_xtick_labels[::3],rotation=45,fontproperties=my_font) #rotaion
```

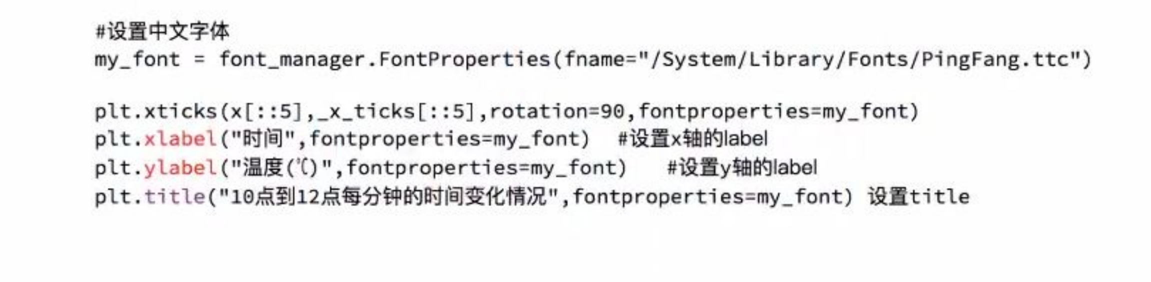


```
#
my_font = font_manager.FontProperties(fname="/System/Library/Fonts/PingFang.ttc")

x = range(0,120)
y = [random.randint(20,35) for i in range(120)]

plt.figure(figsize=(20,8),dpi=80)

plt.plot(x,y)

#调整x轴的刻度
_xtick_labels = ["10点{}分".format(i) for i in range(60)]
_xtick_labels += ["11点{}分".format(i) for i in range(60)]
#取步长，数字和字符串一一对应，数据的长度一样
plt.xticks(list(x)[::3],_xtick_labels[::3],rotation=45,fontproperties=my_font) #rotaio

plt.show()
```

设置中文显示

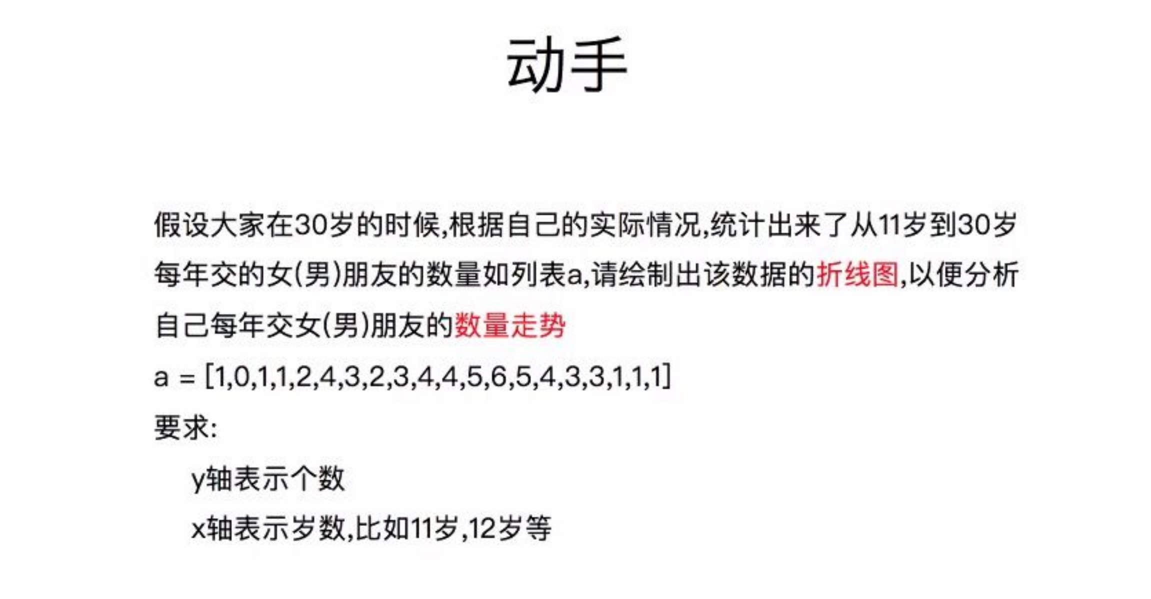


那么x轴y轴和当前图形到底表示什么是不是应该明确一下呢?



```
#设置中文字体
my_font = font_manager.FontProperties(fname="/System/Library/Fonts/PingFang.ttc")

plt.xticks(x[::5],_x_ticks[::5],rotation=90,fontproperties=my_font)
plt.xlabel("时间",fontproperties=my_font)  #设置x轴的label
plt.ylabel("温度(℃)",fontproperties=my_font)   #设置y轴的label
plt.title("10点到12点每分钟的时间变化情况",fontproperties=my_font) 设置title
```

2-4. 练习1：11~30 岁每年交男女数量的走势



动手

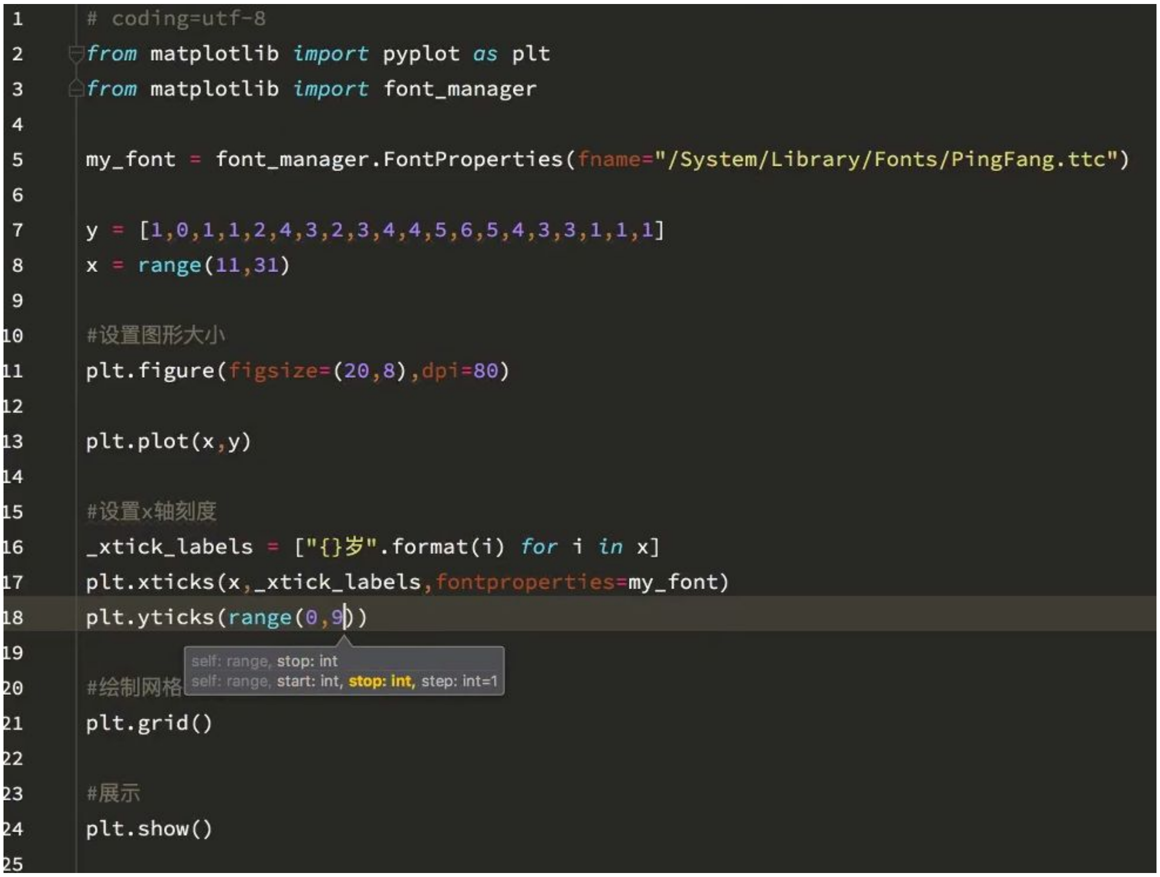
假设大家在30岁的时候,根据自己的实际情况,统计出来了从11岁到30岁每年交的女(男)朋友的数量如列表a,请绘制出该数据的折线图,以便分析自己每年交女(男)朋友的数量走势

a = [1,0,1,1,2,4,3,2,3,4,4,5,6,5,4,3,3,1,1,1]

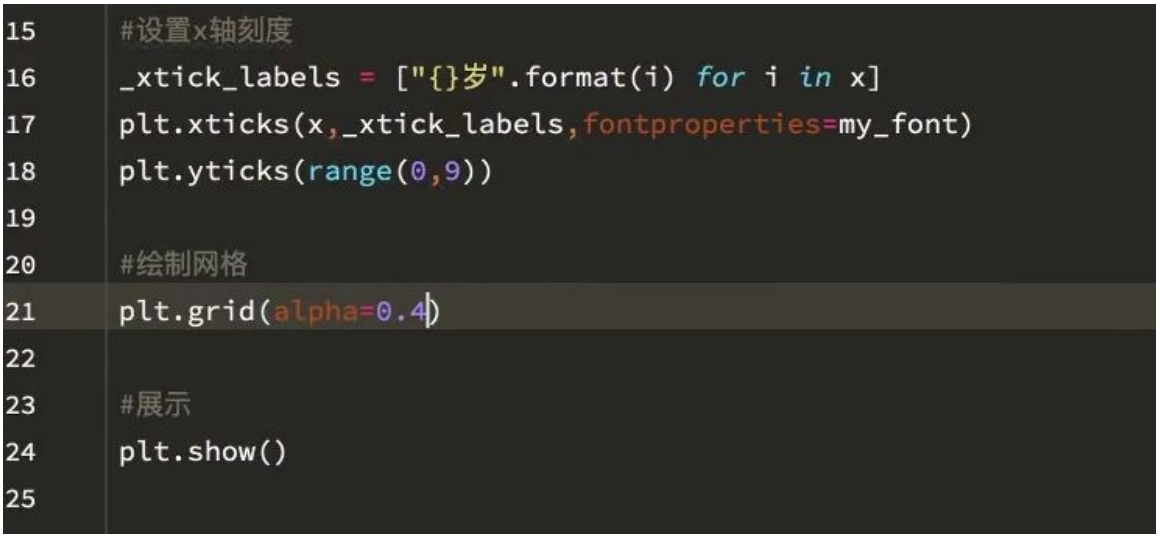
要求:

y轴表示个数

x轴表示岁数,比如11岁,12岁等

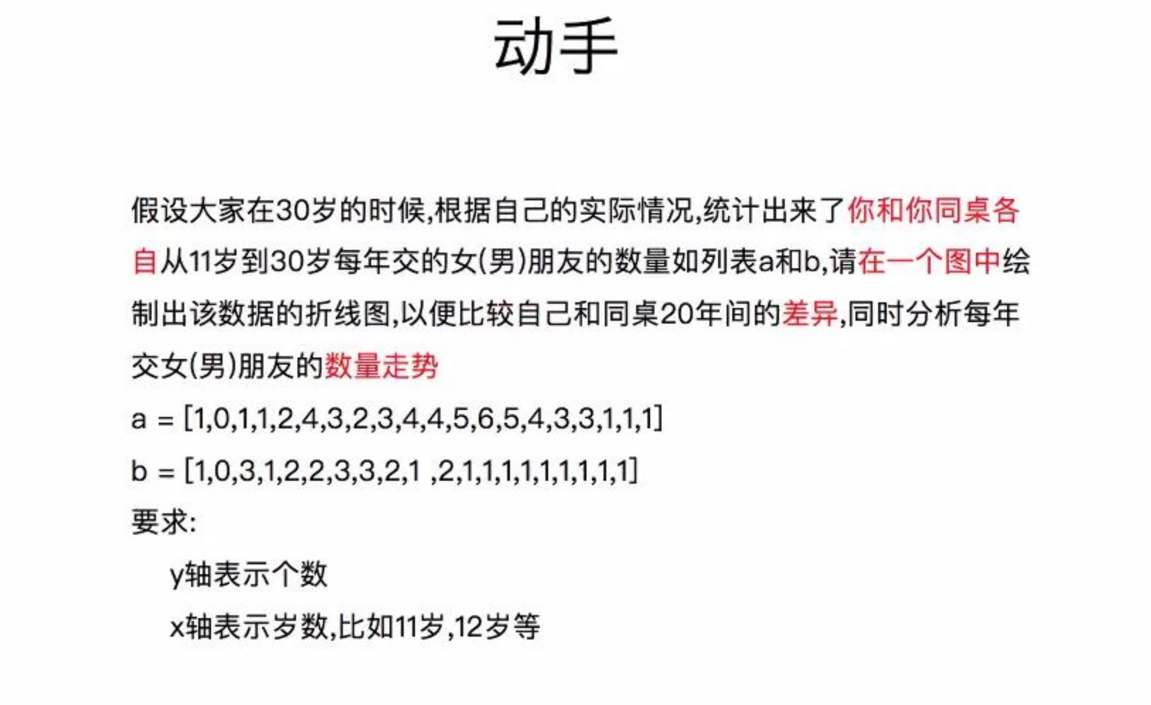


```
# coding=utf-8
from matplotlib import pyplot as plt
from matplotlib import font_manager

my_font = font_manager.FontProperties(fname="/System/Library/Fonts/PingFang.ttc")

y = [1,0,1,1,2,4,3,2,3,4,4,5,6,5,4,3,3,1,1,1]
x = range(11,31)

#设置图形大小
plt.figure(figsize=(20,8),dpi=80)

plt.plot(x,y)

#设置x轴刻度
_xtick_labels = ["{}岁".format(i) for i in x]
plt.xticks(x,_xtick_labels,fontproperties=my_font)
plt.yticks(range(0,9))

#绘制网格
plt.grid()

#展示
plt.show()

```



```
#设置x轴刻度
_xtick_labels = ["{}岁".format(i) for i in x]
plt.xticks(x,_xtick_labels,fontproperties=my_font)
plt.yticks(range(0,9))

#绘制网格
plt.grid(alpha=0.4)

#展示
plt.show()

```

2-5．练习2：11~30 岁你和同桌每年交男女数量的走势



动手

假设大家在30岁的时候,根据自己的实际情况,统计出来了你和你同桌各自从11岁到30岁每年交的女(男)朋友的数量如列表a和b,请在一个图中绘制出该数据的折线图,以便比较自己和同桌20年间的差异,同时分析每年交女(男)朋友的数量走势

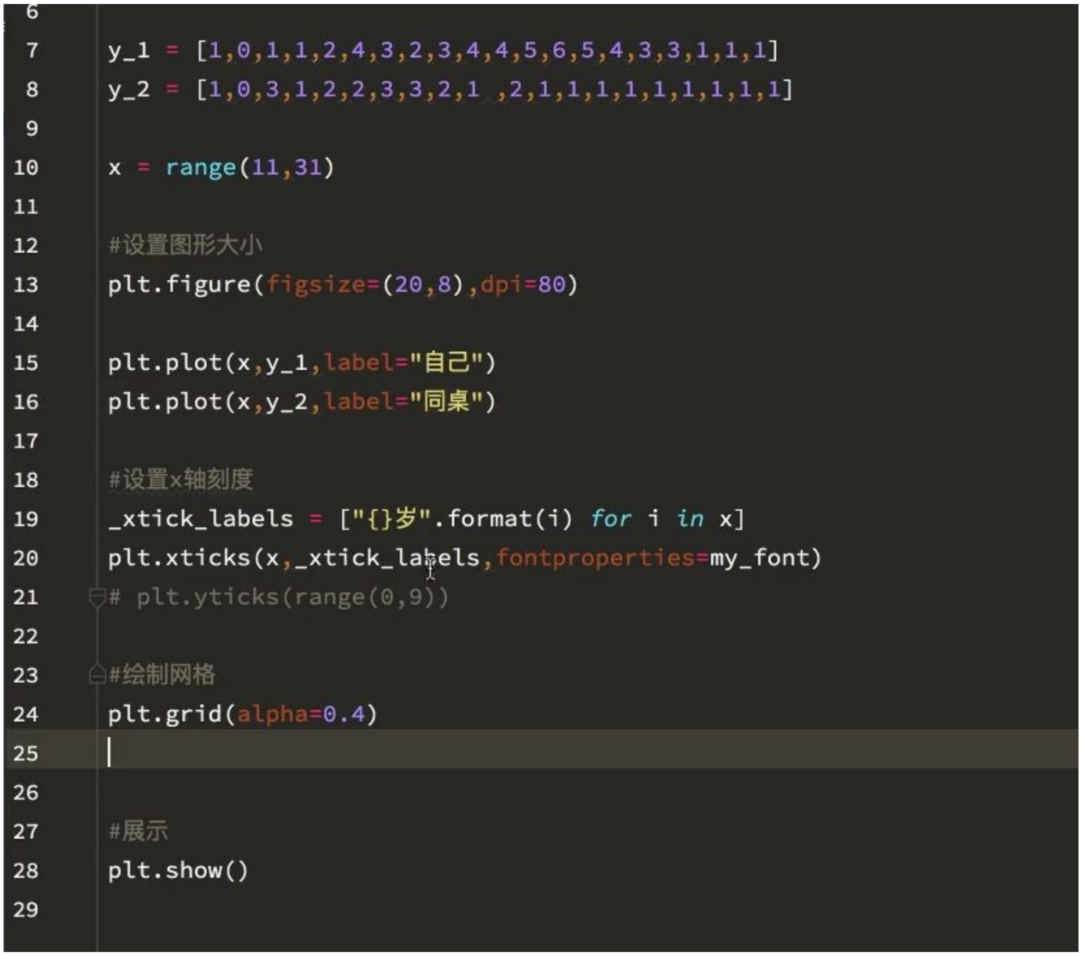
a = [1,0,1,1,2,4,3,2,3,4,4,5,6,5,4,3,3,1,1,1]

b = [1,0,3,1,2,2,3,3,2,1 ,2,1,1,1,1,1,1,1,1,1]

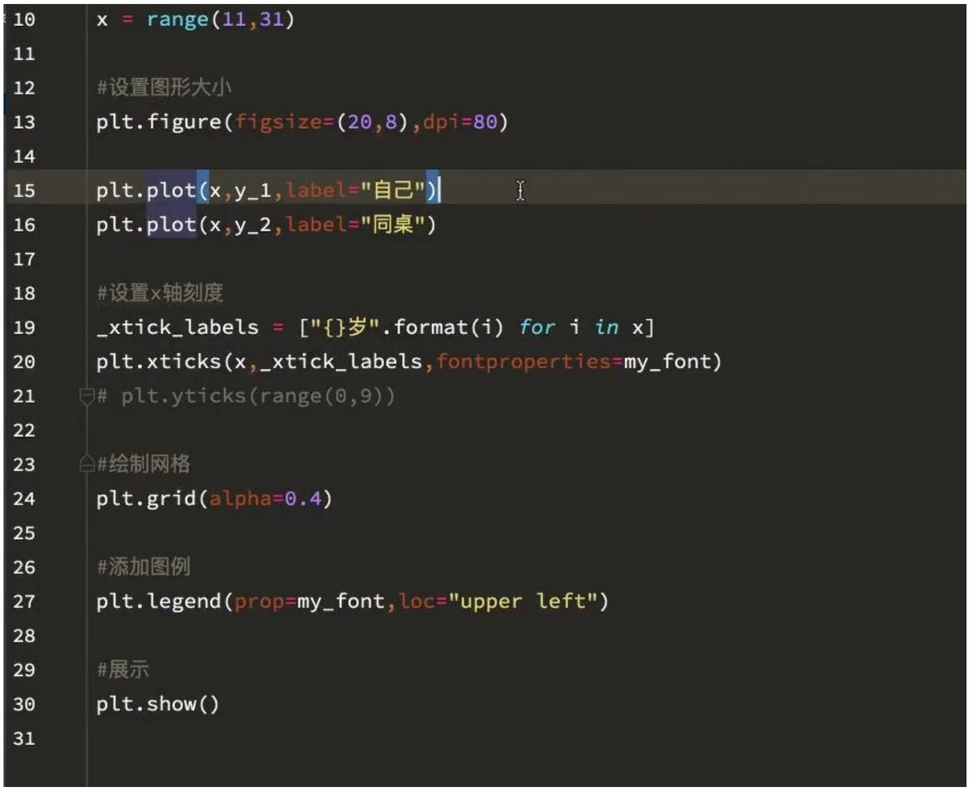
要求:

y轴表示个数

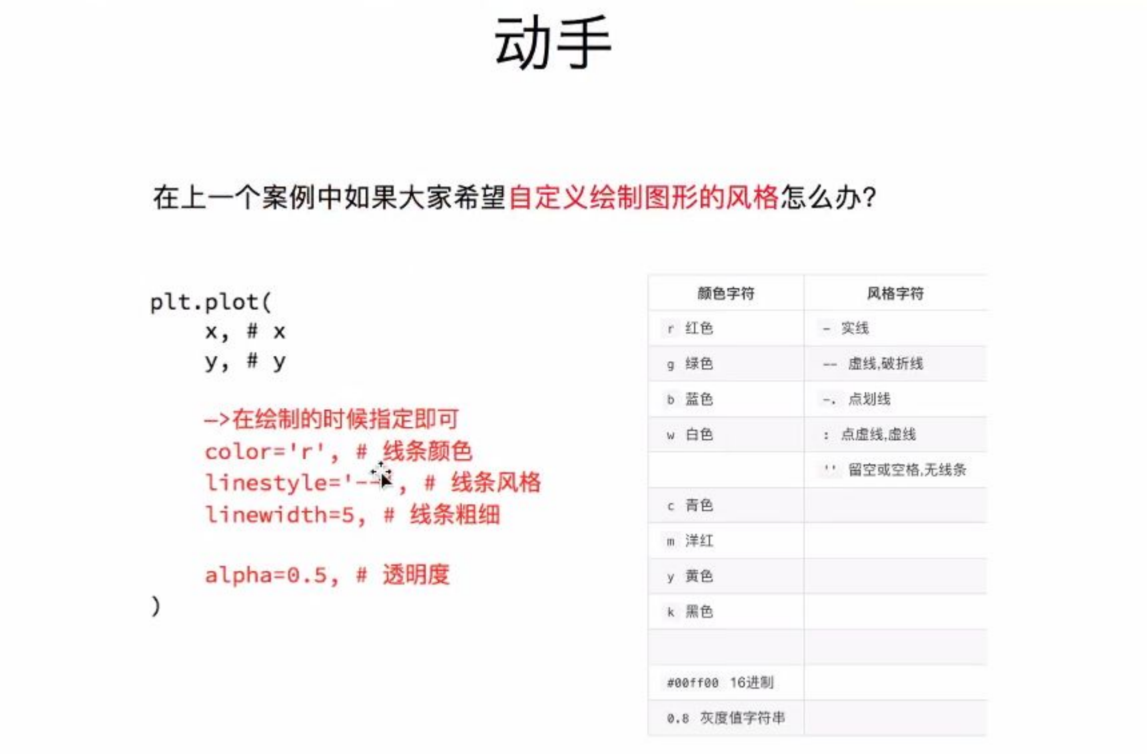
x轴表示岁数,比如11岁,12岁等



```python

y_1 = [1,0,1,1,2,4,3,2,3,4,4,5,6,5,4,3,3,1,1,1]
y_2 = [1,0,3,1,2,2,3,3,2,1 ,2,1,1,1,1,1,1,1,1,1]

x = range(11,31)

#设置图形大小
plt.figure(figsize=(20,8),dpi=80)

plt.plot(x,y_1,label="自己")
plt.plot(x,y_2,label="同桌")

#设置x轴刻度
_xtick_labels = ["{}岁".format(i) for i in x]
plt.xticks(x,_xtick_labels,fontproperties=my_font)
# plt.yticks(range(0,9))

#绘制网格
plt.grid(alpha=0.4)



#展示
plt.show()

```



```python
x = range(11,31)

#设置图形大小
plt.figure(figsize=(20,8),dpi=80)

plt.plot(x,y_1,label="自己")
plt.plot(x,y_2,label="同桌")

#设置x轴刻度
_xtick_labels = ["{}岁".format(i) for i in x]
plt.xticks(x,_xtick_labels,fontproperties=my_font)
# plt.yticks(range(0,9))

#绘制网格
plt.grid(alpha=0.4)

#添加图例
plt.legend(prop=my_font,loc="upper left")

#展示
plt.show()

```

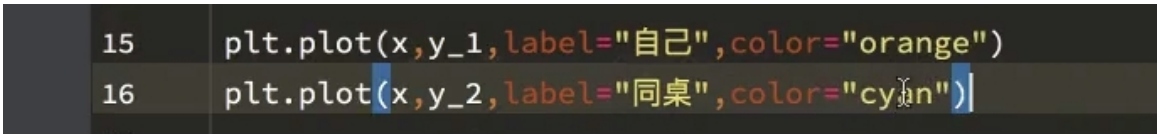


动手

在上一个案例中如果大家希望自定义绘制图形的风格怎么办?

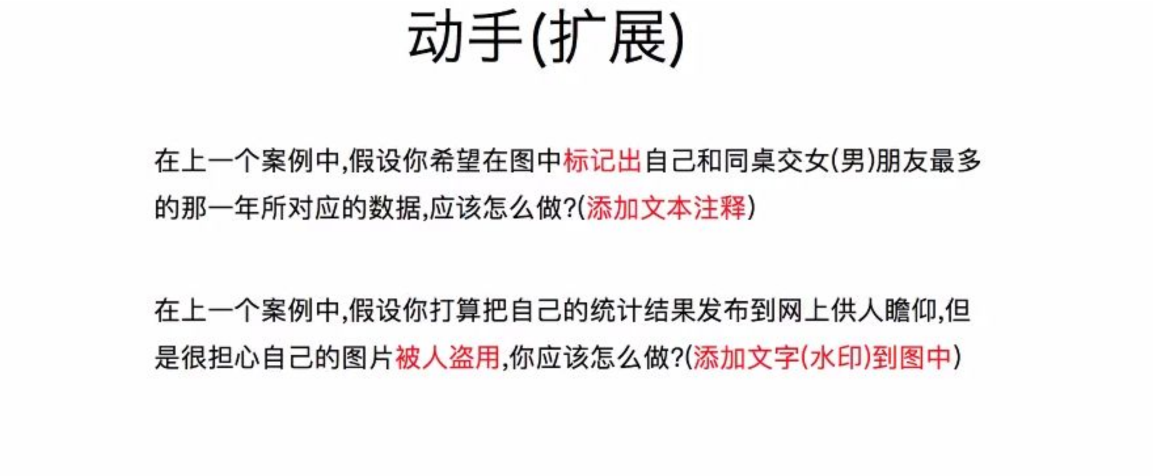
```
plt.plot(
    x, # x
    y, # y

    ->在绘制的时候指定即可
    color='r', # 线条颜色
    linestyle='--', # 线条风格
    linewidth=5, # 线条粗细

    alpha=0.5, # 透明度
)
```

颜色字符	风格字符
r 红色	- 实线
g 绿色	-- 虚线,破折线
b 蓝色	-. 点划线
w 白色	: 点虚线,虚线
	'' 留空或空格,无线条
c 青色	
m 洋红	
y 黄色	
k 黑色	
#00ff00 16进制	
0.8 灰度值字符串	



```
plt.plot(x,y_1,label="自己",color="orange")
plt.plot(x,y_2,label="同桌",color="cyan")
```

2-6. 拓展练习：添加水印

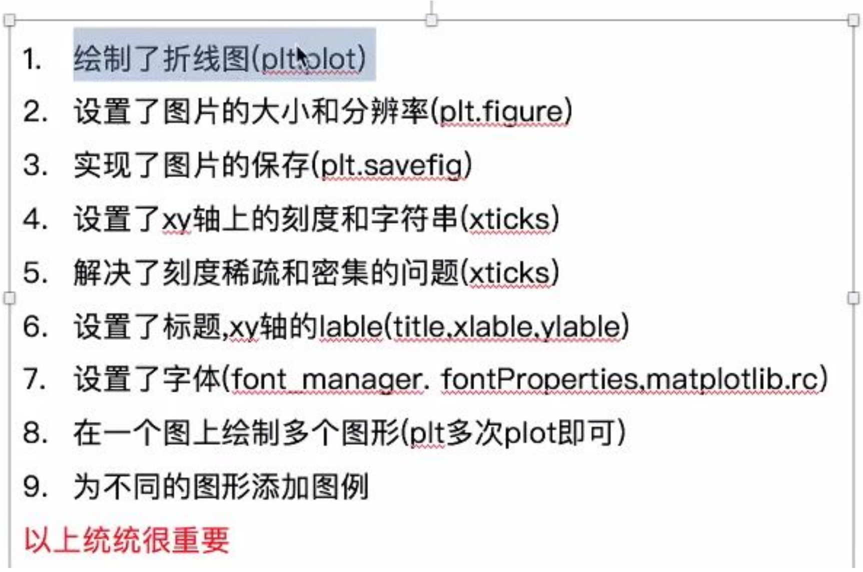


动手(扩展)

在上一个案例中,假设你希望在图中标记出自己和同桌交女(男)朋友最多的那一年所对应的数据,应该怎么做?(添加文本注释)

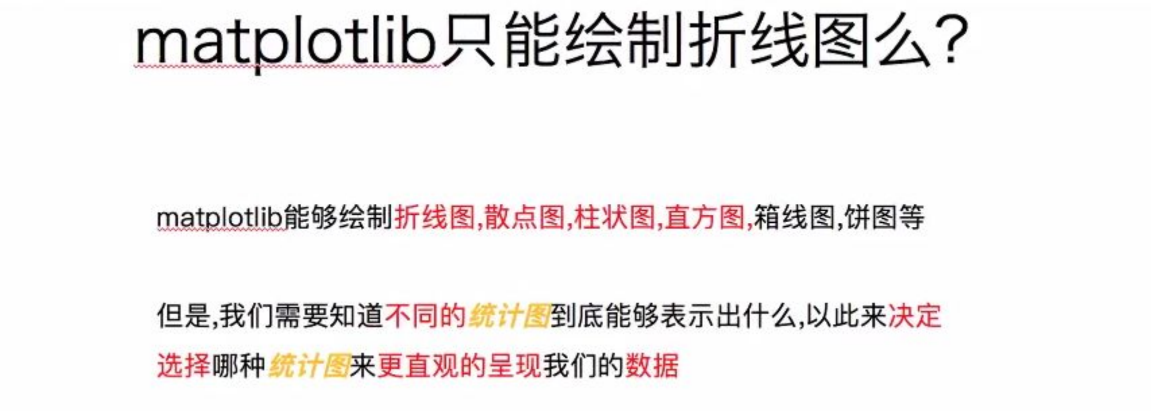
在上一个案例中,假设你打算把自己的统计结果发布到网上供人瞻仰,但是很担心自己的图片被人盗用,你应该怎么做?(添加文字(水印)到图中)

2-7. 总结



1. 绘制了折线图(plt.plot)
2. 设置了图片的大小和分辨率(plt.figure)
3. 实现了图片的保存(plt.savefig)
4. 设置了xy轴上的刻度和字符串(xticks)
5. 解决了刻度稀疏和密集的问题(xticks)
6. 设置了标题,xy轴的lable(title,xlable,ylable)
7. 设置了字体(font_manager. fontProperties,matplotlib.rc)
8. 在一个图上绘制多个图形(plt多次plot即可)
9. 为不同的图形添加图例

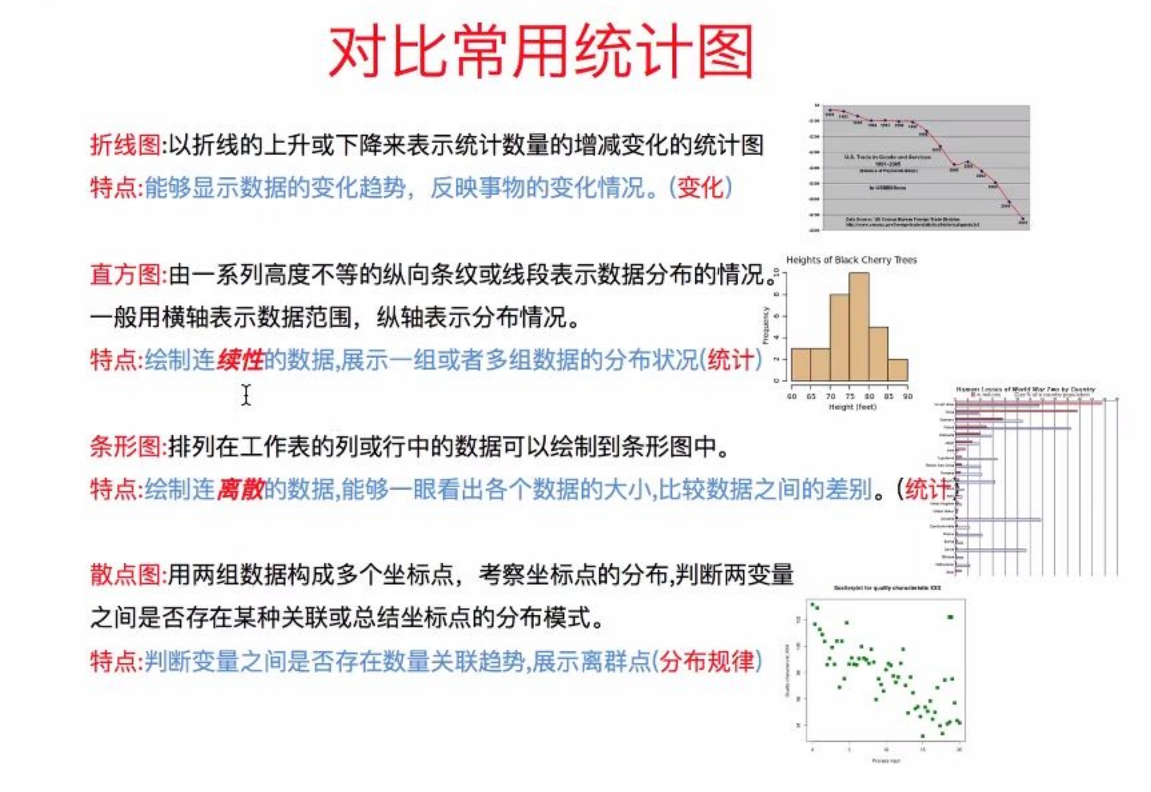
以上统统很重要



matplotlib只能绘制折线图么？

matplotlib能够绘制折线图,散点图,柱状图,直方图,箱线图,饼图等

但是,我们需要知道不同的统计图到底能够表示出什么,以此来决定选择哪种统计图来更直观的呈现我们的数据



对比常用统计图

折线图:以折线的上升或下降来表示统计数量的增减变化的统计图

特点:能够显示数据的变化趋势，反映事物的变化情况。(变化)

直方图:由一系列高度不等的纵向条纹或线段表示数据分布的情况。一般用横轴表示数据范围，纵轴表示分布情况。

特点:绘制连**续性**的数据,展示一组或者多组数据的分布状况(统计)

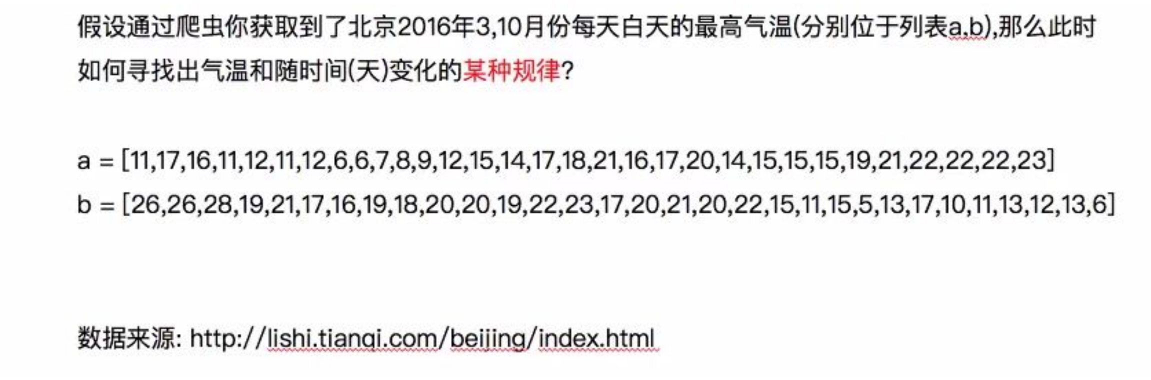
条形图:排列在工作表的列或行中的数据可以绘制到条形图中。

特点:绘制连**离散**的数据,能够一眼看出各个数据的大小,比较数据之间的差别。(统计)

散点图:用两组数据构成多个坐标点，考察坐标点的分布,判断两变量之间是否存在某种关联或总结坐标点的分布模式。

特点:判断变量之间是否存在数量关联趋势,展示离群点(分布规律)

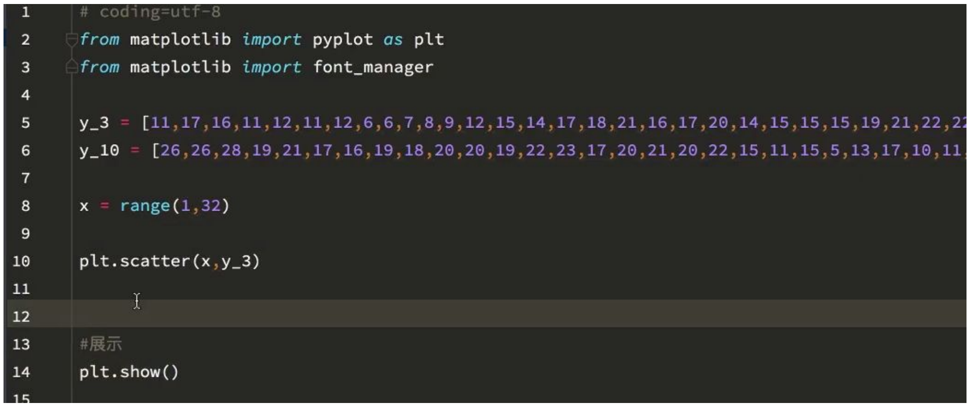
三． 散点图



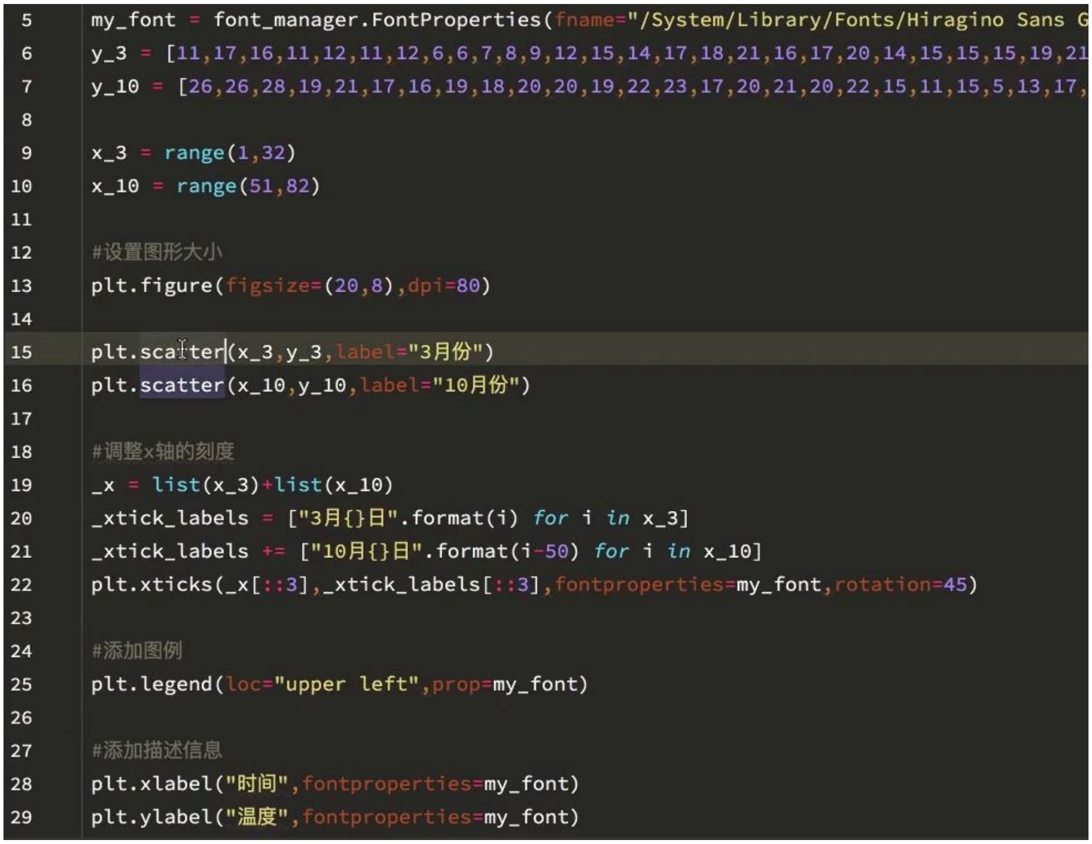
假设通过爬虫你获取到了北京2016年3,10月份每天白天的最高气温(分别位于列表a,b),那么此时如何寻找出气温和随时间(天)变化的某种规律?

a = [11,17,16,11,12,11,12,6,6,7,8,9,12,15,14,17,18,21,16,17,20,14,15,15,15,19,21,22,22,22,23]
b = [26,26,28,19,21,17,16,19,18,20,20,19,22,23,17,20,21,20,22,15,11,15,5,13,17,10,11,13,12,13,6]

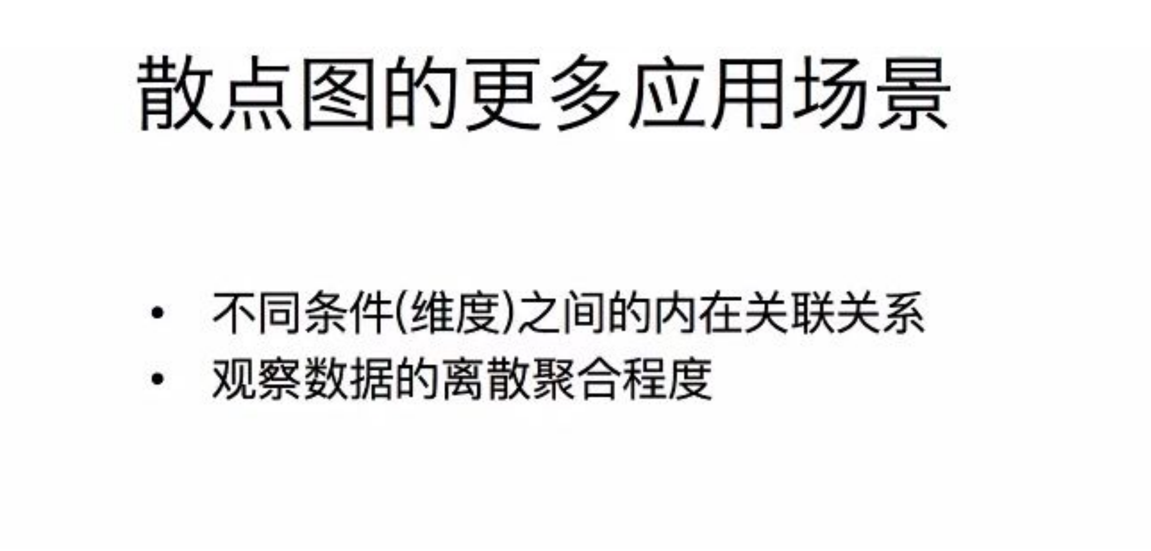
数据来源: http://lishi.tianqi.com/beijing/index.html



```
# coding=utf-8
from matplotlib import pyplot as plt
from matplotlib import font_manager

y_3 = [11,17,16,11,12,11,12,6,6,7,8,9,12,15,14,17,18,21,16,17,20,14,15,15,15,19,21,22,22
y_10 = [26,26,28,19,21,17,16,19,18,20,20,19,22,23,17,20,21,20,22,15,11,15,5,13,17,10,11,

x = range(1,32)

plt.scatter(x,y_3)


#展示
plt.show()
```

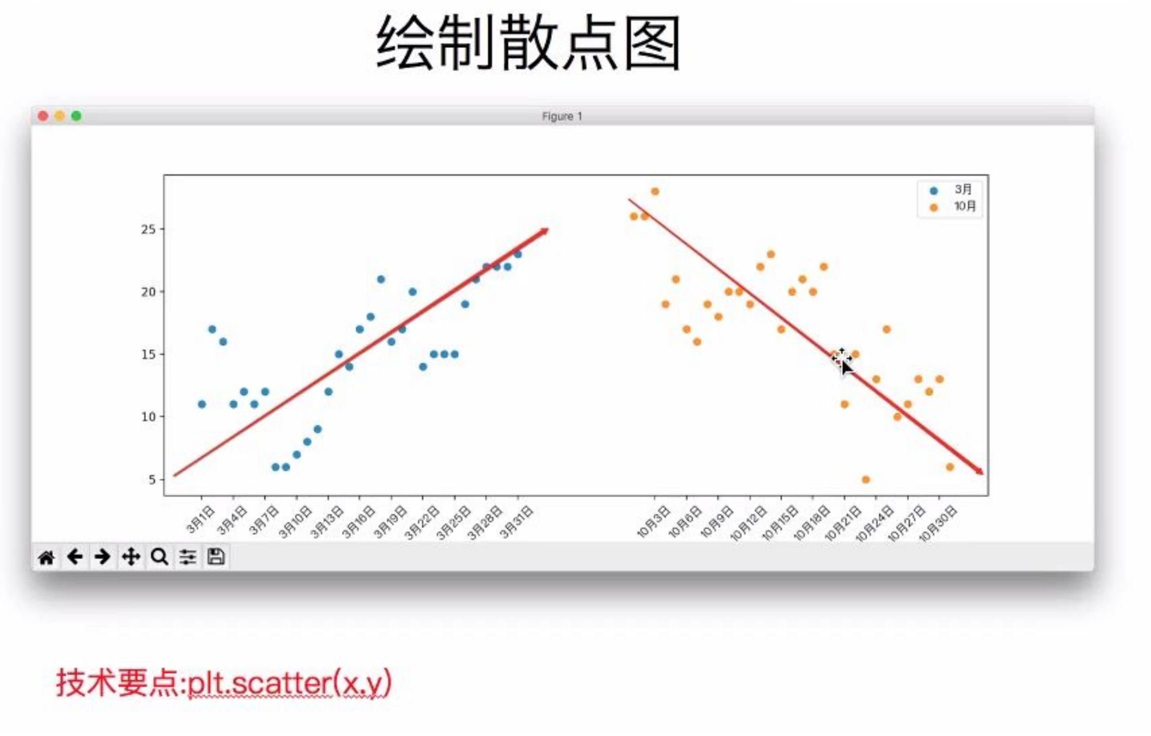


```
my_font = font_manager.FontProperties(fname="/System/Library/Fonts/Hiragino Sans G
y_3 = [11,17,16,11,12,11,12,6,6,7,8,9,12,15,14,17,18,21,16,17,20,14,15,15,15,19,21
y_10 = [26,26,28,19,21,17,16,19,18,20,20,19,22,23,17,20,21,20,22,15,11,15,5,13,17,

x_3 = range(1,32)
x_10 = range(51,82)

#设置图形大小
plt.figure(figsize=(20,8),dpi=80)

plt.scatter(x_3,y_3,label="3月份")
plt.scatter(x_10,y_10,label="10月份")

#调整x轴的刻度
_x = list(x_3)+list(x_10)
_xtick_labels = ["3月{}日".format(i) for i in x_3]
_xtick_labels += ["10月{}日".format(i-50) for i in x_10]
plt.xticks(_x[::3],_xtick_labels[::3],fontproperties=my_font,rotation=45)

#添加图例
plt.legend(loc="upper left",prop=my_font)

#添加描述信息
plt.xlabel("时间",fontproperties=my_font)
plt.ylabel("温度",fontproperties=my_font)
```



散点图的更多应用场景

- 不同条件(维度)之间的内在关联关系
- 观察数据的离散聚合程度

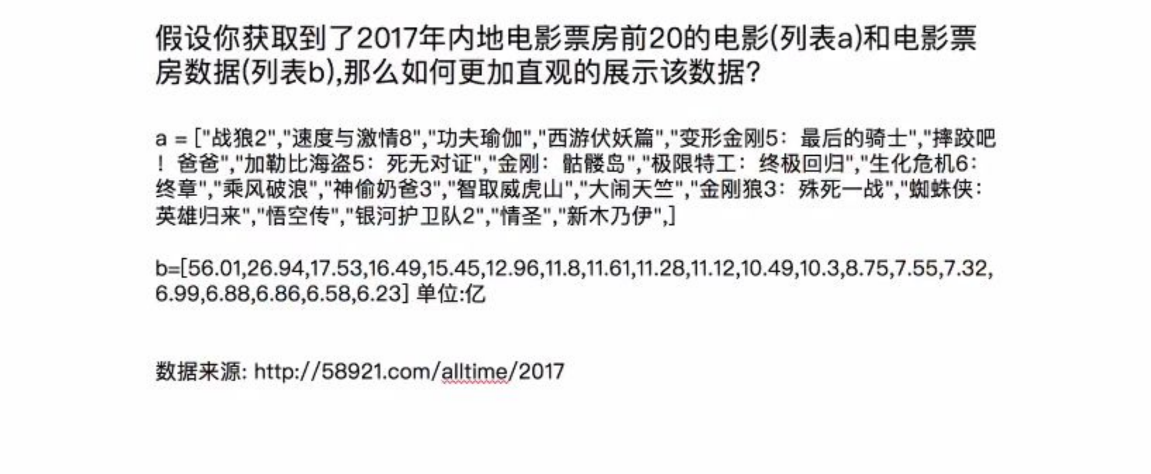


绘制散点图

技术要点:plt.scatter(x,y)

四． 条形图

4-1．单个条形图绘制

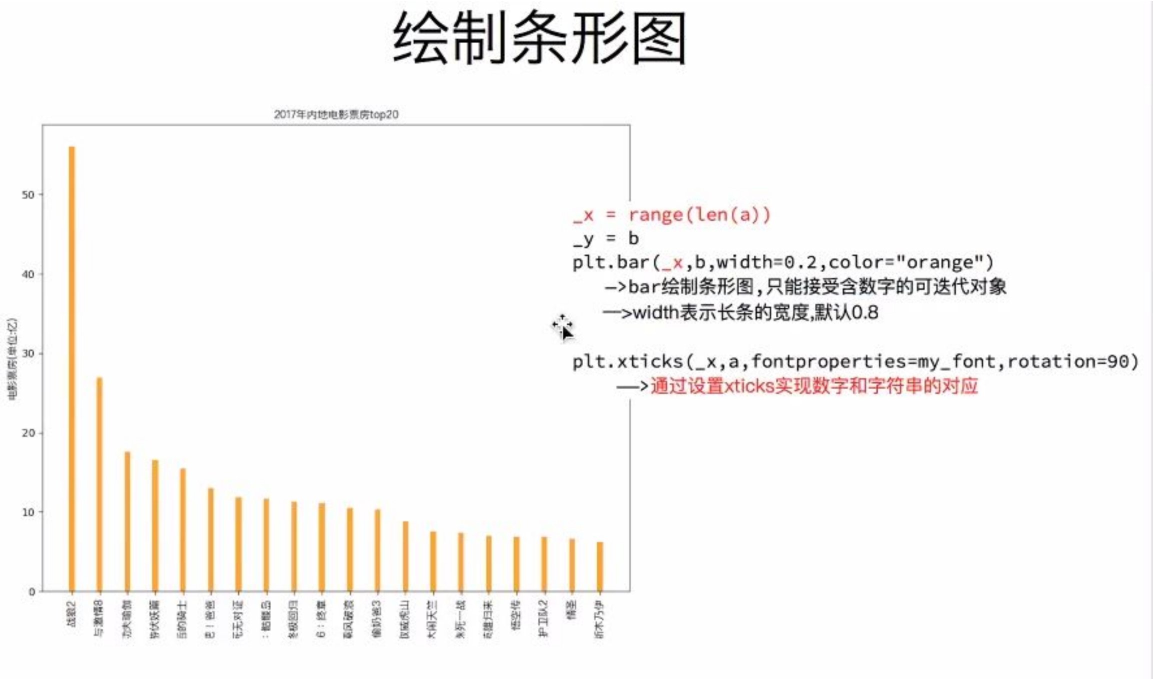


假设你获取到了2017年内地电影票房前20的电影(列表a)和电影票房数据(列表b),那么如何更加直观的展示该数据?

a = ["战狼2","速度与激情8","功夫瑜伽","西游伏妖篇","变形金刚5：最后的骑士","摔跤吧！爸爸","加勒比海盗5：死无对证","金刚：骷髅岛","极限特工：终极回归","生化危机6：终章","乘风破浪","神偷奶爸3","智取威虎山","大闹天竺","金刚狼3：殊死一战","蜘蛛侠：英雄归来","悟空传","银河护卫队2","情圣","新木乃伊",]

b=[56.01,26.94,17.53,16.49,15.45,12.96,11.8,11.61,11.28,11.12,10.49,10.3,8.75,7.55,7.32,6.99,6.88,6.86,6.58,6.23] 单位:亿

数据来源: http://58921.com/alltime/2017



绘制条形图

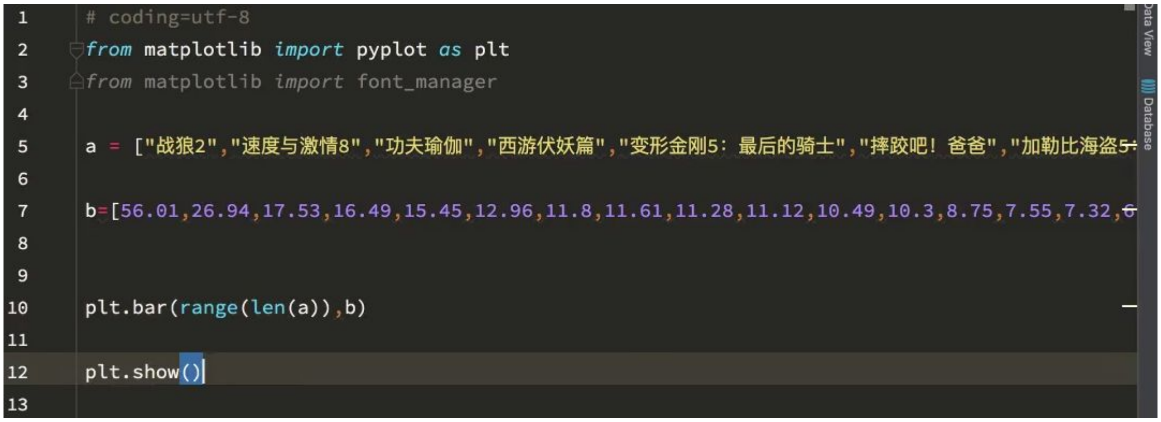
```
_x = range(len(a))
_y = b
plt.bar(_x,b,width=0.2,color="orange")
```

—>bar绘制条形图,只能接受含数字的可迭代对象

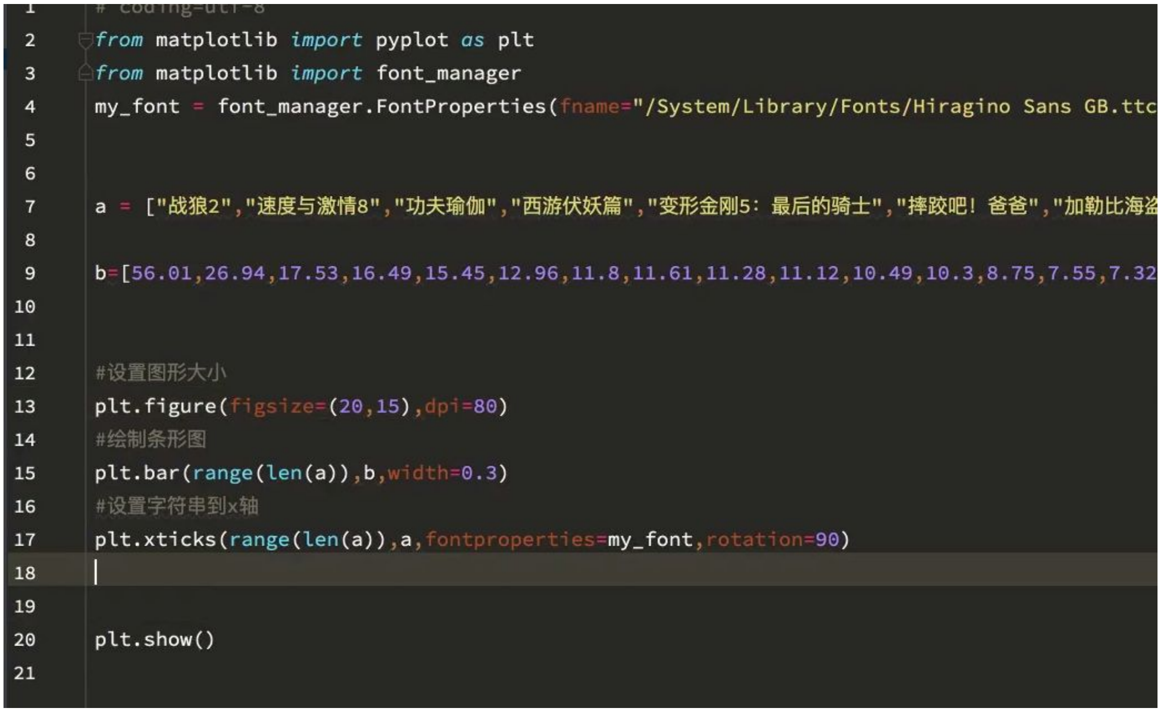
—>width表示长条的宽度,默认0.8

```
plt.xticks(_x,a,fontproperties=my_font,rotation=90)
```

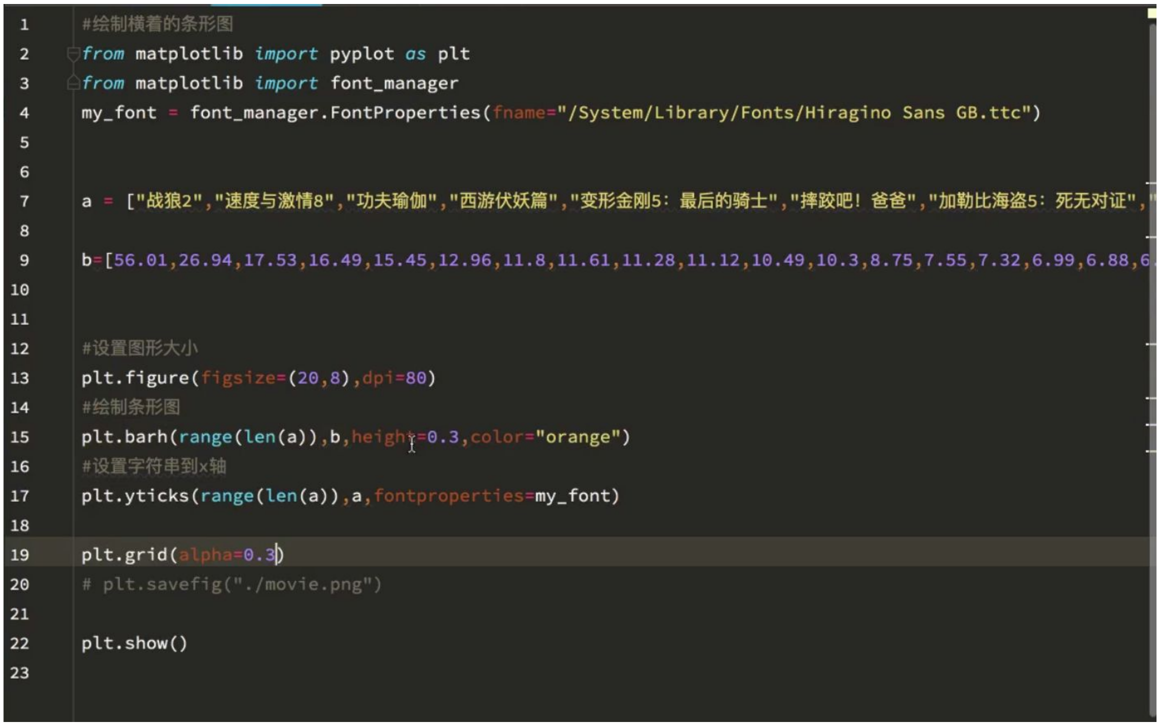
—>通过设置xticks实现数字和字符串的对应



```
# coding=utf-8
from matplotlib import pyplot as plt
from matplotlib import font_manager

a = ["战狼2","速度与激情8","功夫瑜伽","西游伏妖篇","变形金刚5: 最后的骑士","摔跤吧! 爸爸","加勒比海盗5

b=[56.01,26.94,17.53,16.49,15.45,12.96,11.8,11.61,11.28,11.12,10.49,10.3,8.75,7.55,7.32,6


plt.bar(range(len(a)),b)

plt.show()

```

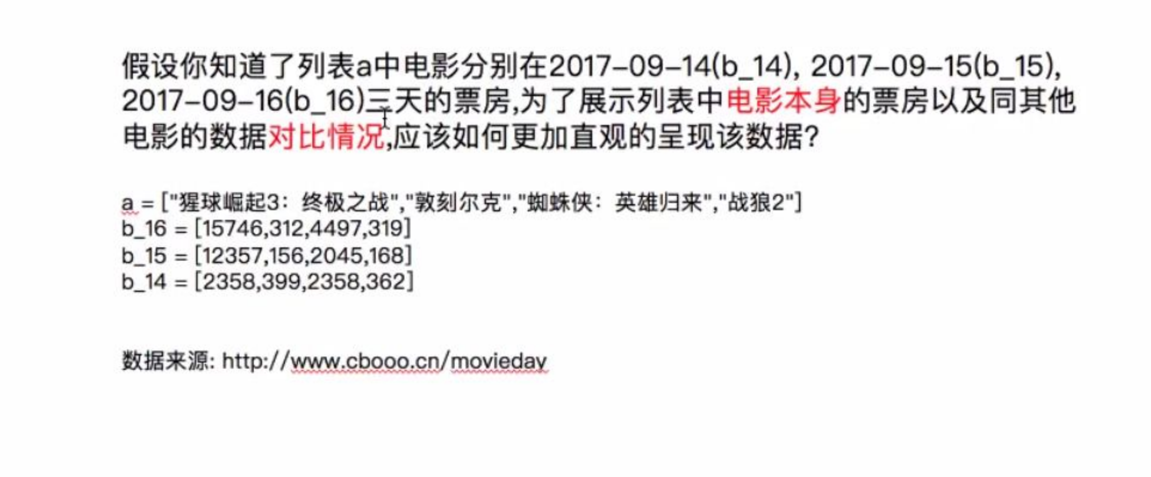


```python
# coding=utf-8
from matplotlib import pyplot as plt
from matplotlib import font_manager
my_font = font_manager.FontProperties(fname="/System/Library/Fonts/Hiragino Sans GB.ttc


a = ["战狼2","速度与激情8","功夫瑜伽","西游伏妖篇","变形金刚5：最后的骑士","摔跤吧！爸爸","加勒比海盗

b=[56.01,26.94,17.53,16.49,15.45,12.96,11.8,11.61,11.28,11.12,10.49,10.3,8.75,7.55,7.32


#设置图形大小
plt.figure(figsize=(20,15),dpi=80)
#绘制条形图
plt.bar(range(len(a)),b,width=0.3)
#设置字符串到x轴
plt.xticks(range(len(a)),a,fontproperties=my_font,rotation=90)


plt.show()

```



```python
#绘制横着的条形图
from matplotlib import pyplot as plt
from matplotlib import font_manager
my_font = font_manager.FontProperties(fname="/System/Library/Fonts/Hiragino Sans GB.ttc")


a = ["战狼2","速度与激情8","功夫瑜伽","西游伏妖篇","变形金刚5：最后的骑士","摔跤吧！爸爸","加勒比海盗5：死无对证","

b=[56.01,26.94,17.53,16.49,15.45,12.96,11.8,11.61,11.28,11.12,10.49,10.3,8.75,7.55,7.32,6.99,6.88,6


#设置图形大小
plt.figure(figsize=(20,8),dpi=80)
#绘制条形图
plt.barh(range(len(a)),b,height=0.3,color="orange")
#设置字符串到x轴
plt.yticks(range(len(a)),a,fontproperties=my_font)

plt.grid(alpha=0.3)
# plt.savefig("./movie.png")

plt.show()

```

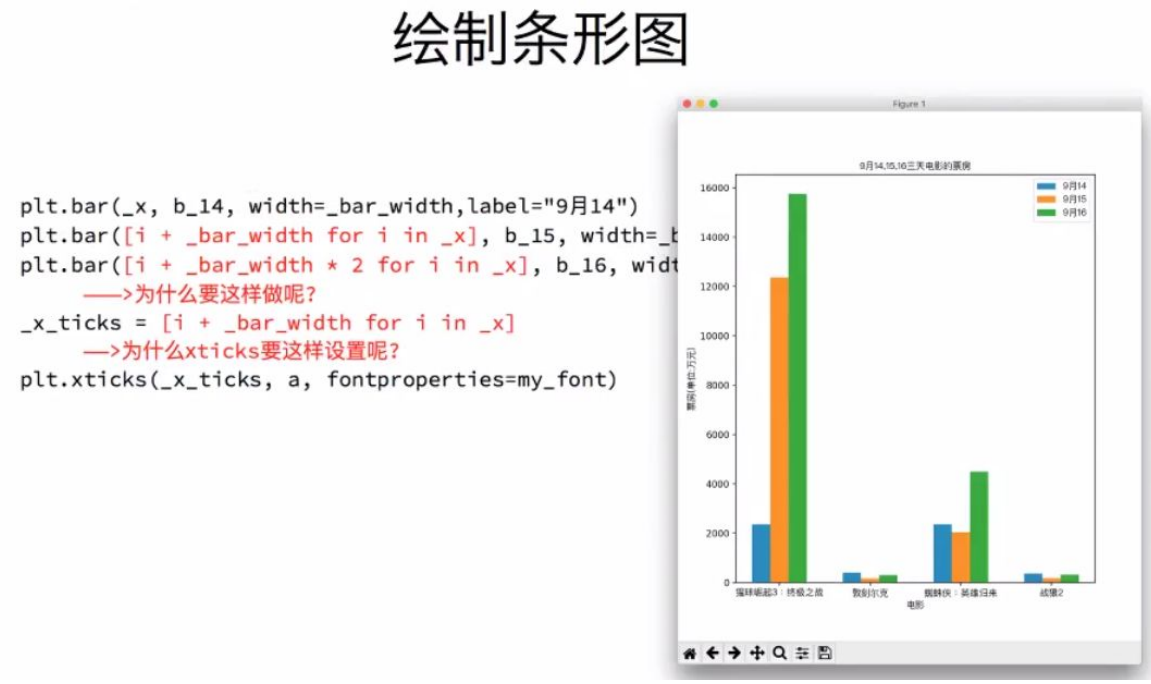
4-2. 多个条形图绘制



假设你知道了列表a中电影分别在2017-09-14(b_14), 2017-09-15(b_15), 2017-09-16(b_16)三天的票房,为了展示列表中电影本身的票房以及同其他电影的数据对比情况,应该如何更加直观的呈现该数据?

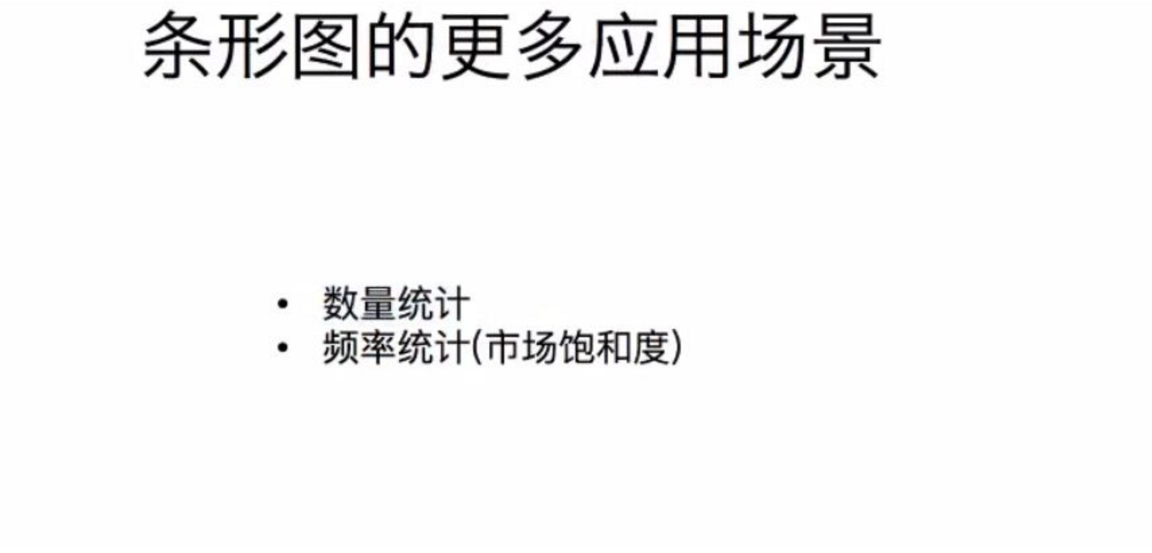
a = ["猩球崛起3：终极之战","敦刻尔克","蜘蛛侠：英雄归来","战狼2"]
b_16 = [15746,312,4497,319]
b_15 = [12357,156,2045,168]
b_14 = [2358,399,2358,362]

数据来源: http://www.cbooo.cn/movieday



绘制条形图

```
plt.bar(_x, b_14, width=_bar_width,label="9月14")
plt.bar([i + _bar_width for i in _x], b_15, width=_b
plt.bar([i + _bar_width * 2 for i in _x], b_16, widt
      ——>为什么要这样做呢?
_x_ticks = [i + _bar_width for i in _x]
      —>为什么xticks要这样设置呢?
plt.xticks(_x_ticks, a, fontproperties=my_font)
```

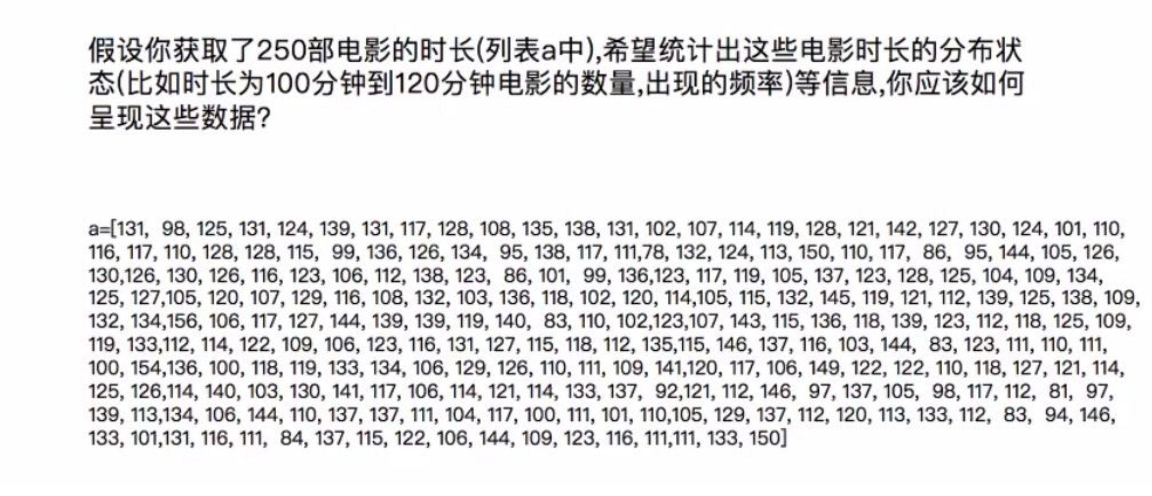


条形图的更多应用场景

- 数量统计
- 频率统计(市场饱和度)

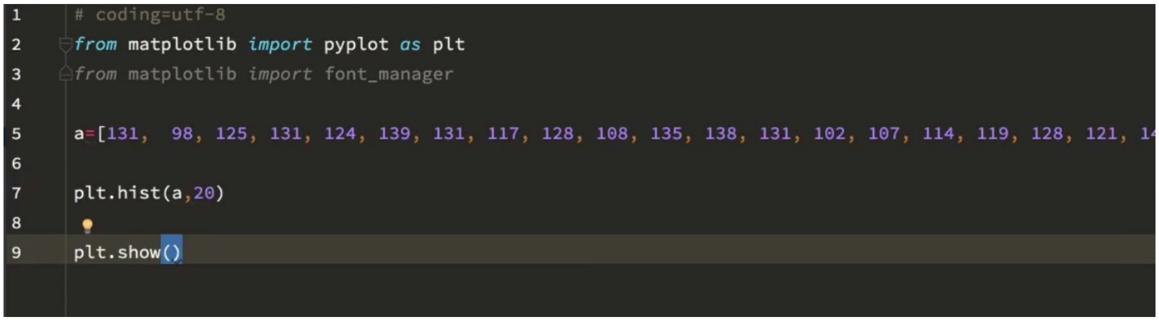
五． 绘制直方图

5-1．常见直方图



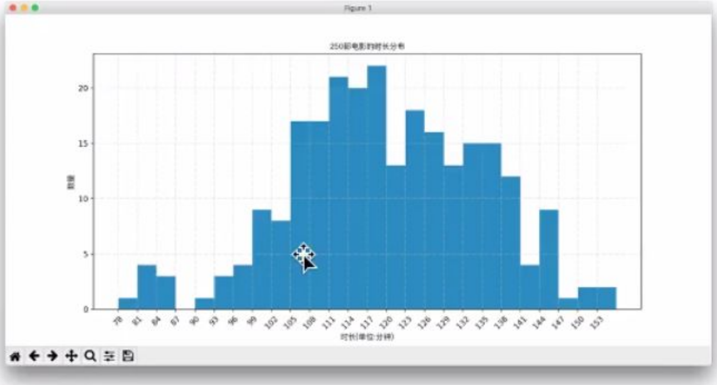
假设你获取了250部电影的时长(列表a中),希望统计出这些电影时长的分布状态(比如时长为100分钟到120分钟电影的数量,出现的频率)等信息,你应该如何呈现这些数据?

a=[131, 98, 125, 131, 124, 139, 131, 117, 128, 108, 135, 138, 131, 102, 107, 114, 119, 128, 121, 142, 127, 130, 124, 101, 110, 116, 117, 110, 128, 128, 115, 99, 136, 126, 134, 95, 138, 117, 111,78, 132, 124, 113, 150, 110, 117, 86, 95, 144, 105, 126, 130,126, 130, 126, 116, 123, 106, 112, 138, 123, 86, 101, 99, 136,123, 117, 119, 105, 137, 123, 128, 125, 104, 109, 134, 125, 127,105, 120, 107, 129, 116, 108, 132, 103, 136, 118, 102, 120, 114,105, 115, 132, 145, 119, 121, 112, 139, 125, 138, 109, 132, 134,156, 106, 117, 127, 144, 139, 139, 119, 140, 83, 110, 102,123,107, 143, 115, 136, 118, 139, 123, 112, 118, 125, 109, 119, 133,112, 114, 122, 109, 106, 123, 116, 131, 127, 115, 118, 112, 135,115, 146, 137, 116, 103, 144, 83, 123, 111, 110, 111, 100, 154,136, 100, 118, 119, 133, 134, 106, 129, 126, 110, 111, 109, 141,120, 117, 106, 149, 122, 122, 110, 118, 127, 121, 114, 125, 126,114, 140, 103, 130, 141, 117, 106, 114, 121, 114, 133, 137, 92,121, 112, 146, 97, 137, 105, 98, 117, 112, 81, 97, 139, 113,134, 106, 144, 110, 137, 137, 111, 104, 117, 100, 111, 101, 110,105, 129, 137, 112, 120, 113, 133, 112, 83, 94, 146, 133, 101,131, 116, 111, 84, 137, 115, 122, 106, 144, 109, 123, 116, 111,111, 133, 150]



```
# coding=utf-8
from matplotlib import pyplot as plt
from matplotlib import font_manager

a=[131,  98, 125, 131, 124, 139, 131, 117, 128, 108, 135, 138, 131, 102, 107, 114, 119, 128, 121, 14

plt.hist(a,20)

plt.show()
```

绘制直方图



把数据分为多少组进行统计???

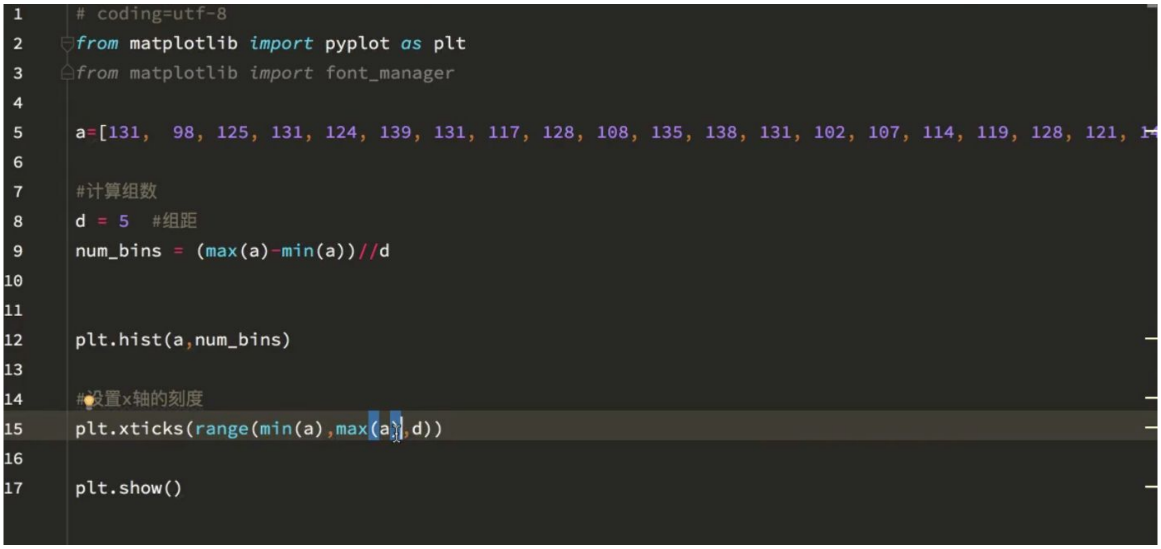
组数要适当,太少会有较大的统计误差,大多规律不明显

组数：将数据分组，当数据在100个以内时，按数据多少常分5-12组。

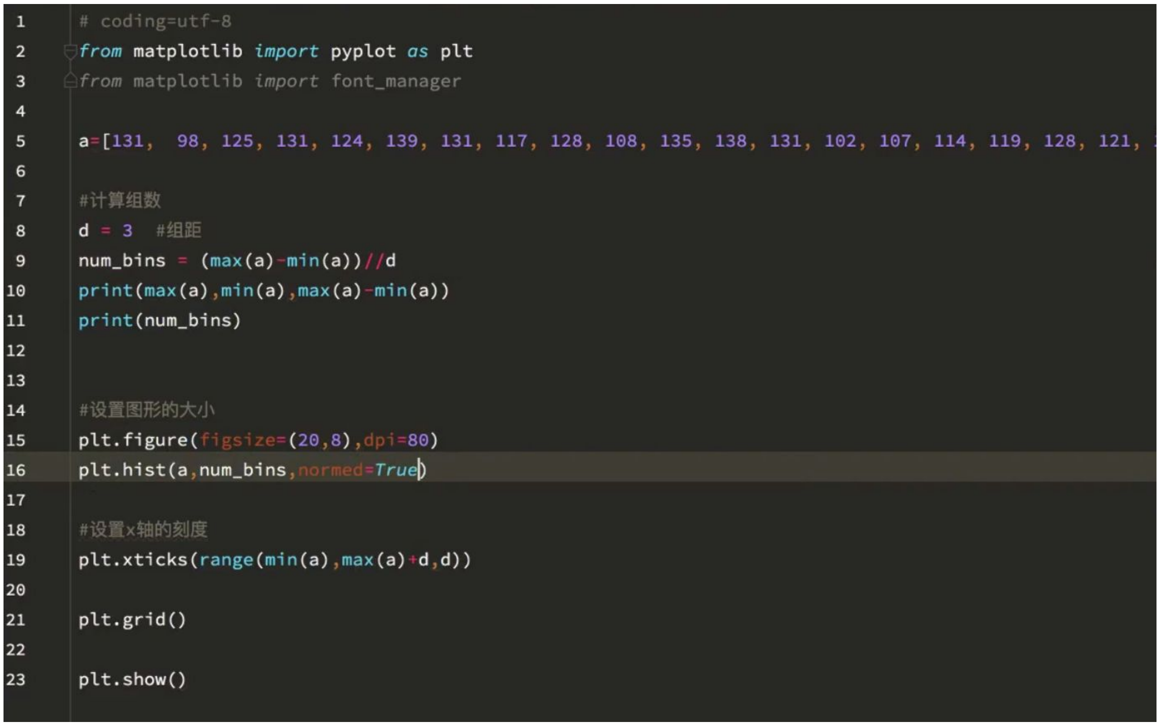
组距：指每个小组的两个端点的距离，

$$组数=\frac{极差}{组距}=\frac{max(a)-min(a)}{bin\_width}$$

```
bin_width = 3 #设置组距为3
num_bins = int((max(a)-min(a))/bin_width) #分为多少组
plt.hist(a, num_bins)
            —>传入需要统计的数据，以及组数即可
#plt.hist(a, [min(a)+i*bin_width for i in range(num_bins)])
            —>可以传入一个列表,长度为组数,值为分组依据,当组距不均匀的时候使用
# plt.hist(a, num_bins, normed=1)
            —>normed:bool 是否绘制频率分布直方图,默认为频数直方图
plt.xticks(list(range(min(a),max(a)))[::bin_width],rotation=45)
plt.grid(True, linestyle = "-.",alpha=0.5)  #显示网格,alpha为透明度
```

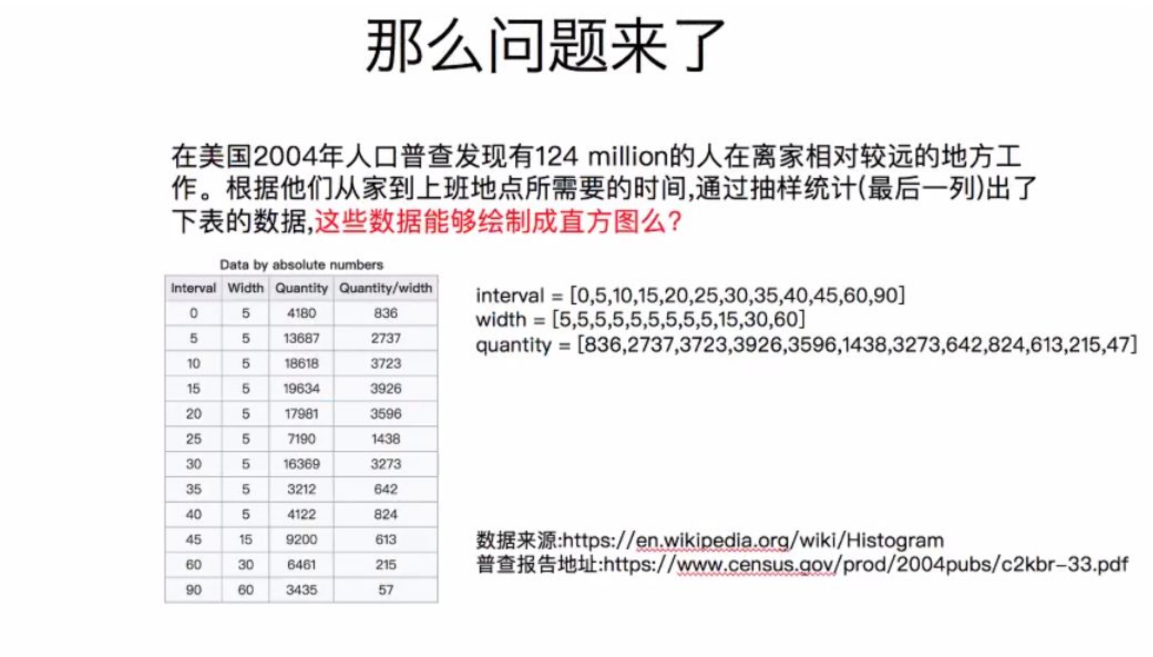


```
# coding=utf-8
from matplotlib import pyplot as plt
from matplotlib import font_manager

a=[131,  98, 125, 131, 124, 139, 131, 117, 128, 108, 135, 138, 131, 102, 107, 114, 119, 128, 121, 1

#计算组数
d = 5  #组距
num_bins = (max(a)-min(a))//d


plt.hist(a,num_bins)

#设置x轴的刻度
plt.xticks(range(min(a),max(a),d))

plt.show()
```



```python
# coding=utf-8
from matplotlib import pyplot as plt
from matplotlib import font_manager

a=[131,  98, 125, 131, 124, 139, 131, 117, 128, 108, 135, 138, 131, 102, 107, 114, 119, 128, 121,

#计算组数
d = 3  #组距
num_bins = (max(a)-min(a))//d
print(max(a),min(a),max(a)-min(a))
print(num_bins)


#设置图形的大小
plt.figure(figsize=(20,8),dpi=80)
plt.hist(a,num_bins,normed=True)

#设置x轴的刻度
plt.xticks(range(min(a),max(a)+d,d))

plt.grid()

plt.show()
```

5-2. 特殊直方图



那么问题来了

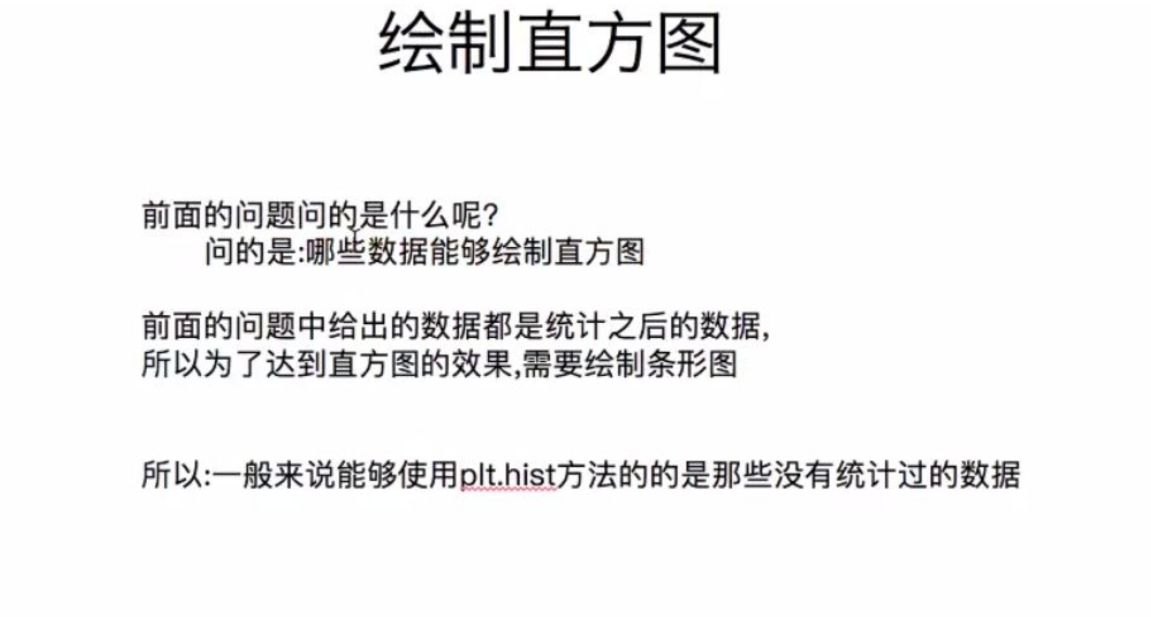
在美国2004年人口普查发现有124 million的人在离家相对较远的地方工作。根据他们从家到上班地点所需要的时间,通过抽样统计(最后一列)出了下表的数据,这些数据能够绘制成直方图么?

Data by absolute numbers

Interval	Width	Quantity	Quantity/width
0	5	4180	836
5	5	13687	2737
10	5	18618	3723
15	5	19634	3926
20	5	17981	3596
25	5	7190	1438
30	5	16369	3273
35	5	3212	642
40	5	4122	824
45	15	9200	613
60	30	6461	215
90	60	3435	57

interval = [0,5,10,15,20,25,30,35,40,45,60,90]
width = [5,5,5,5,5,5,5,5,5,15,30,60]
quantity = [836,2737,3723,3926,3596,1438,3273,642,824,613,215,47]

数据来源:https://en.wikipedia.org/wiki/Histogram
普查报告地址:https://www.census.gov/prod/2004pubs/c2kbr-33.pdf

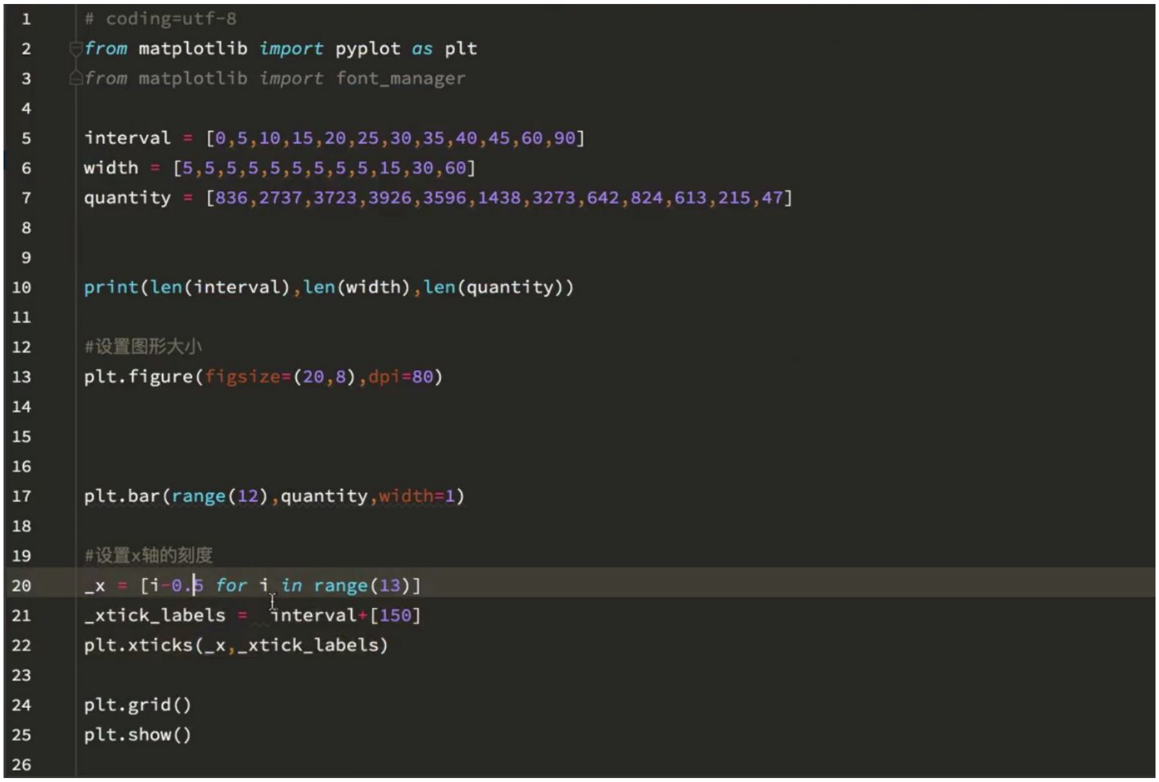


绘制直方图

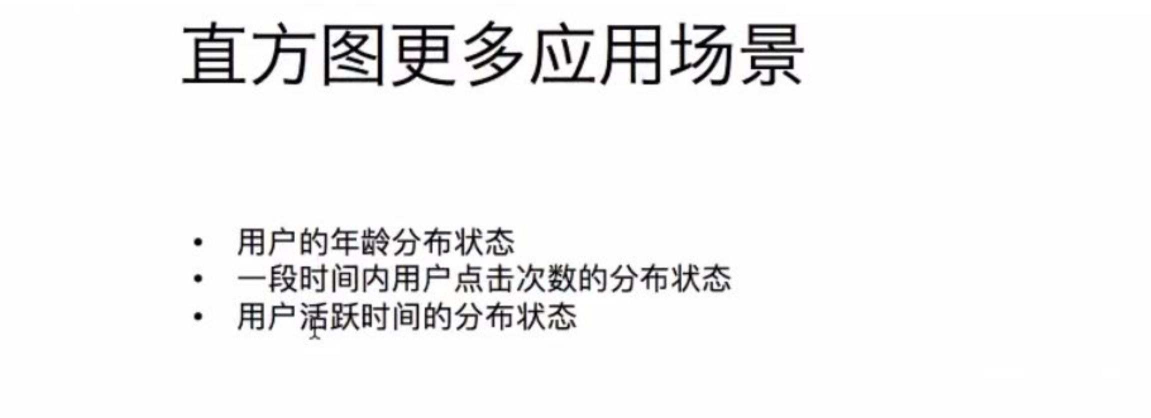
前面的问题问的是什么呢?
问的是:哪些数据能够绘制直方图

前面的问题中给出的数据都是统计之后的数据,
所以为了达到直方图的效果,需要绘制条形图

所以:一般来说能够使用plt.hist方法的的是那些没有统计过的数据



```python
# coding=utf-8
from matplotlib import pyplot as plt
from matplotlib import font_manager

interval = [0,5,10,15,20,25,30,35,40,45,60,90]
width = [5,5,5,5,5,5,5,5,5,15,30,60]
quantity = [836,2737,3723,3926,3596,1438,3273,642,824,613,215,47]


print(len(interval),len(width),len(quantity))

#设置图形大小
plt.figure(figsize=(20,8),dpi=80)



plt.bar(range(12),quantity,width=1)

#设置x轴的刻度
_x = [i-0.5 for i in range(13)]
_xtick_labels = interval+[150]
plt.xticks(_x,_xtick_labels)

plt.grid()
plt.show()

```

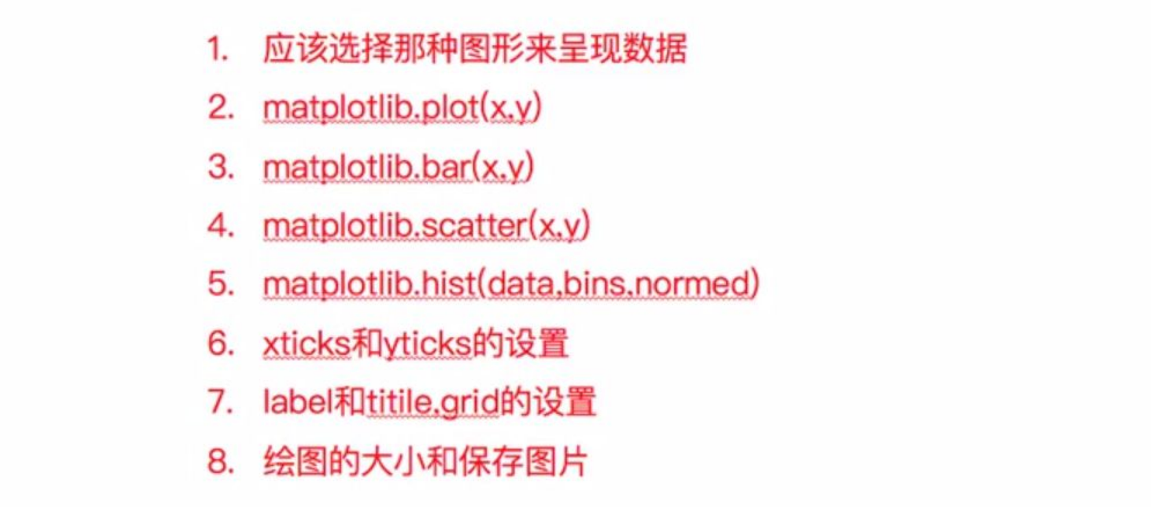


直方图更多应用场景

- 用户的年龄分布状态
- 一段时间内用户点击次数的分布状态
- 用户活跃时间的分布状态

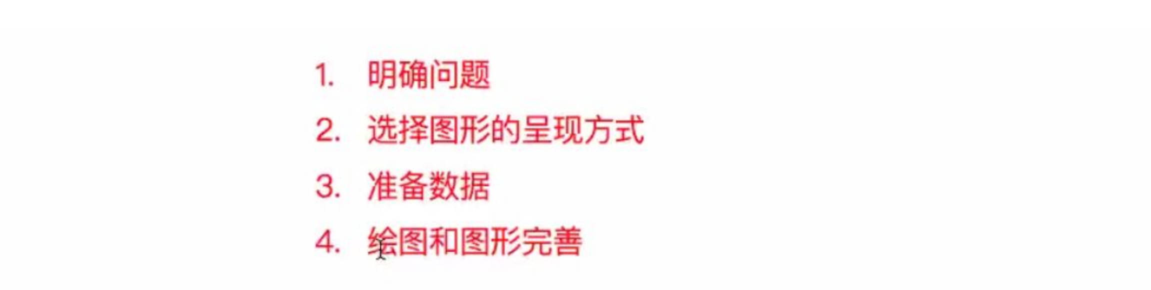
六． Matplotlib 拓展知识

6-1. Matplotlib 常见问题总结



1. 应该选择那种图形来呈现数据
2. matplotlib.plot(x,y)
3. matplotlib.bar(x,y)
4. matplotlib.scatter(x,y)
5. matplotlib.hist(data,bins,normed)
6. xticks和yticks的设置
7. label和titile,grid的设置
8. 绘图的大小和保存图片

6-2. Matplotlib 使用的流程总结



1. 明确问题
2. 选择图形的呈现方式
3. 准备数据
4. 绘图和图形完善

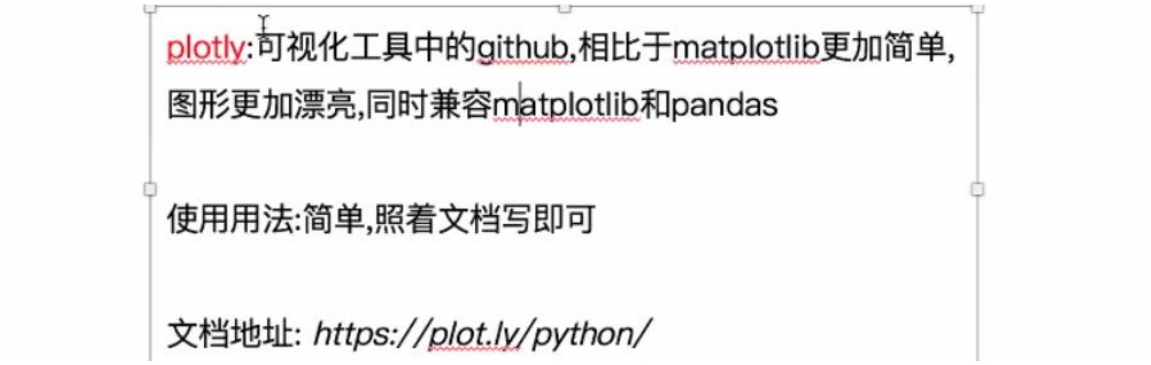
6-3. Matplotlib 更多绘图样式

matplotlib支持的图形是非常多的，如果有其他的需求，我们
可以查看一下url地址：
http://matplotlib.org/gallery/index.html

6-4. 更多的绘图工具

1). 百度的 echart

2). Plotly 模块



plotly:可视化工具中的github,相比于matplotlib更加简单,
图形更加漂亮,同时兼容matplotlib和pandas

使用用法:简单,照着文档写即可

文档地址: *https://plot.ly/python/*