

# Python的matplotlib.pyplot画图的方法总结

库的使用：

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

基本方法：

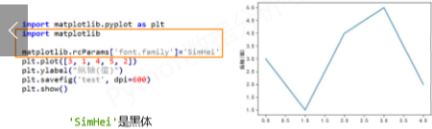
- 文本显示函数

```
plt.xlabel(text,fontproperties,size) # text为内容 (str) 给x轴命名
plt.ylabel(text,fontproperties,size) # text为内容 (str) 给y轴命名
plt.title(text,fontproperties,size) # text为内容 (str) 给图命名
plt.text(x,y,text,fontproperties) # x,y为图中坐标, text为添加文本 在图中任意位置添加文本
```

- 中文显示：

## pyplot的中文显示：第一种方法

pyplot并不默认支持中文显示，需要rcParams修改字体实现



## rcParams的属性

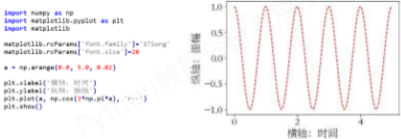
属性	说明
'font.family'	用于显示字体的名字
'font.style'	字体风格，正常'normal'或 斜体'italic'
'font.size'	字体大小，整数字号或者'large'、'x-small'

## 中文字体的种类

rcParams['font.family']

中文字体	说明
'SimHei'	中文黑体
'KaiTi'	中文楷体
'LiSu'	中文隶书
'FangSong'	中文仿宋
'YouYuan'	中文幼圆
'STSong'	华文宋体

## 实例



- 第二种方法：直接改变文本显示函数中的参数

fontproperties = string # string为中文字体的种类，见上图

```
size = 14 # 设置字号
```

- 第三种方法:

```
from matplotlib.font_manager import FontProperties

font =
FontProperties(fname=r"c:\windows\fonts\simsun.ttc", ,size
=14)# 设置字体为14号简宋
'''后面使用文本显示函数时, 将参数fontproperties=font, 并且不再设置size'''
```

- 简单分区:

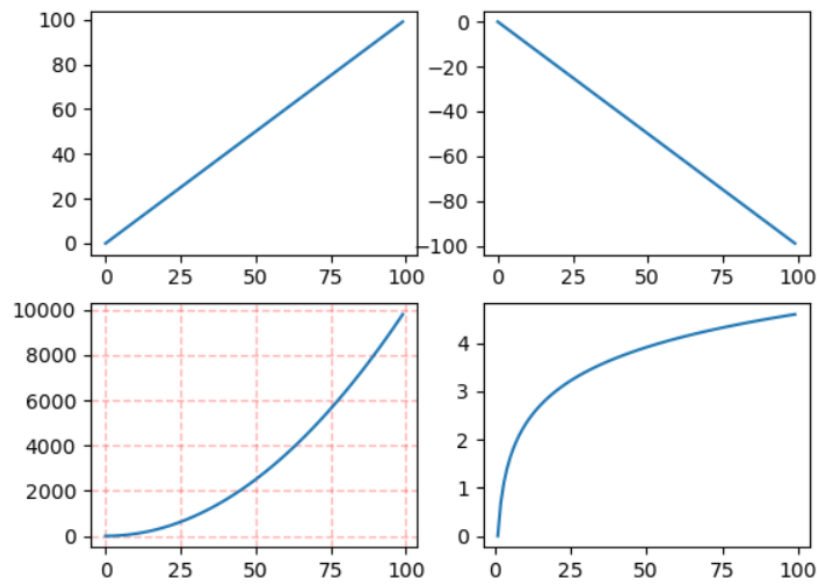
```
# 第一种方法
plt.subplot(x,y,current) # 划分为x行y列, 序号从1开始到x*y,
current指的是当前在哪个区域

# 第二种方法
fig,ax = plt.subplots(x,y) # 划分为x行y列
'''plt.subplots() 函数见名知意, 就是画多个图的意思。这个函数返回的
是一个元组 (tuple)。这个元组中包含了一个figure对象和axes对象集合。
因此, 当我们按上面fig, ax = plt.subplots()这样的用法时, 我们相当
于把返回的 tuple 进行解压 (unpack) 成 fig 和 ax 两个变量。ax 变量
中保存着所有子图的可操作 axe 对象。
摘自: https://www.jianshu.com/p/2da30e247900
'''

# 例子1
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 100)
#划分子图
fig,axes=plt.subplots(2,2)
ax1=axes[0,0]
ax2=axes[0,1]
ax3=axes[1,0]
ax4=axes[1,1]

#作图1
ax1.plot(x, x)
#作图2
ax2.plot(x, -x)
#作图3
ax3.plot(x, x ** 2)
ax3.grid(color='r', linestyle='--', linewidth=1,alpha=0.3)
#作图4
ax4.plot(x, np.log(x))
plt.show()
```



# 例子2

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import matplotlib
```

```
matplotlib.rcParams['font.family'] = 'YouYuan'
```

```
AMean = [75, 77, 85]
```

```
BMean = [72, 88, 87]
```

```
CMean = [70, 85, 88]
```

```
ind = np.arange(0,3,1) # 分成3组
```

```
width = 0.25 # 每个柱子的宽度
```

```
fig, ax = plt.subplots() # 创建一个子图
```

```
# 绘制第一个柱状图A
```

```
rects1 = ax.bar(ind, AMean,
width, hatch='/', edgecolor='black', color='white', label='A')
```

```
# 绘制第二个柱状图B
```

```
rects2 = ax.bar(ind + width, BMean, width,
hatch='*', edgecolor='black', color='white', label='B')
```

```
# 绘制第三个柱状图C
```

```
rects3 = ax.bar(ind+2*width
, CMean, width, hatch='.', edgecolor='black', color='white', label='C')
```

```
ax.set_xticks(ind+width)
```

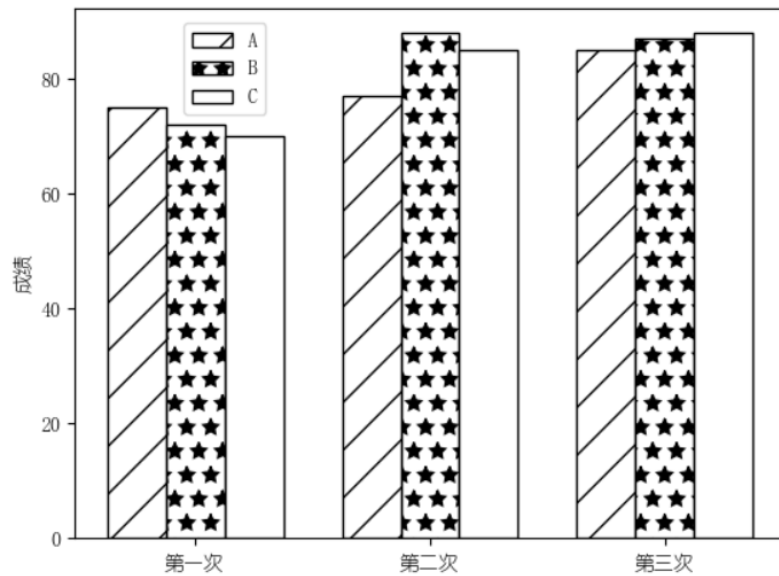
```
ax.set_xticklabels( ('第一次', '第二次', '第三次') )
```

```
plt.ylabel('成绩')
```

```
plt.legend(bbox_to_anchor=(0.14,0.78))
```

```
plt.show()
```

Figure 1



## 1.折线图、曲线图

```
plt.plot(x,y,format_string,**kwargs)
```

```
'''
```

**x**:x轴数据，列表形式或一维数组

**y**:y轴数据，列表形式或一维数组

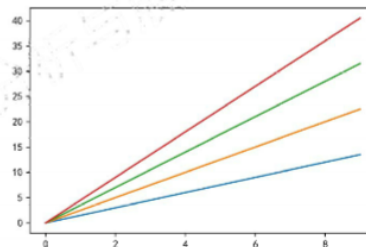
**format\_string**:控制曲线格式的字符串 eg: 'ro-', 'r' 是红色, 'o' 是圆点, '-' 是实线

**\*\*kwargs**:第二组 (x,y,f\_s) 【用来绘制多条线】或者其他参数

```
'''
```

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

a = np.arange(10)
plt.plot(a, a*1.5, a, a*2.5, a, a*3.5, a, a*4.5)
plt.show()
```

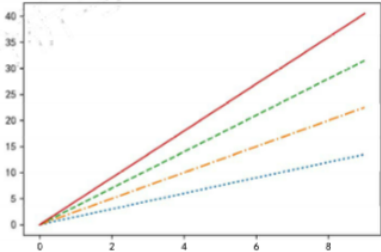


- `format_string`: 控制曲线的格式字符串，可选  
由颜色字符、风格字符和标记字符组成

颜色字符	说明	颜色字符	说明
'b'	蓝色	'm'	洋红色 magenta
'g'	绿色	'y'	黄色
'r'	红色	'k'	黑色
'c'	青绿色 cyan	'w'	白色
'#008000'	RGB某颜色	'0.8'	灰度值字符串

- `format_string`: 控制曲线的格式字符串，可选  
由颜色字符、风格字符和标记字符组成

风格字符	说明
'_'	实线
'--'	破折线
'-.'	点划线
':'	虚线
''	无线条



- `format_string`: 控制曲线的格式字符串，可选

标记字符	说明	标记字符	说明	标记字符	说明
'.'	点标记	'1'	下花三角标记	'h'	竖六边形标记
','	像素标记(极小点)	'2'	上花三角标记	'H'	横六边形标记
'o'	实心圆标记	'3'	左花三角标记	'+'	十字标记
'v'	倒三角标记	'4'	右花三角标记	'x'	x标记
'^'	上三角标记	's'	实心方形标记	'D'	菱形标记
'>'	右三角标记	'p'	实心五角标记	'd'	瘦菱形标记
'<'	左三角标记	'*'	星形标记	' '	垂直线标记

- `color`: 控制颜色, `color='green'`
  - `linestyle`: 线条风格, `linestyle='dashed'`
  - `marker`: 标记风格, `marker='o'`
  - `markerfacecolor`: 标记颜色, `markerfacecolor='blue'`
  - `markersize`: 标记尺寸, `markersize=20`
- .....



```
# 例子1
import matplotlib.pyplot as plt
font = 'STXINGKAI'
```

```

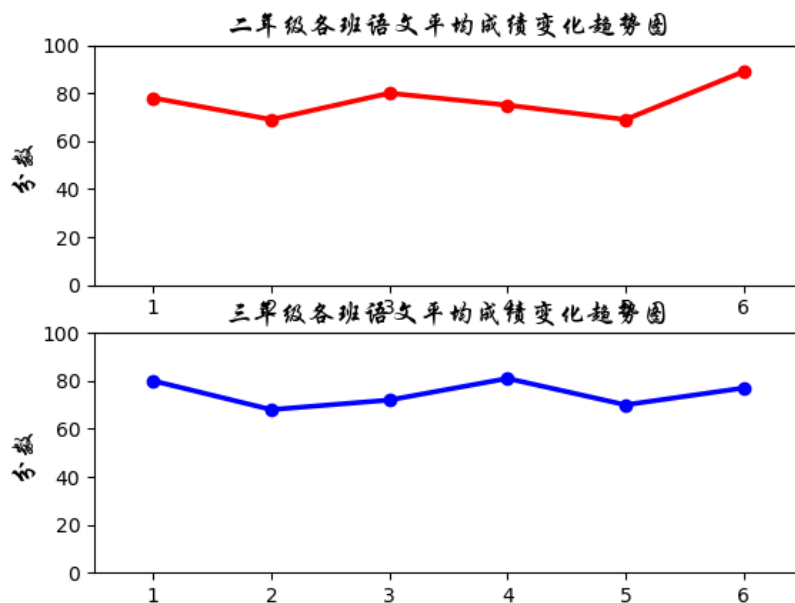
x = [1,2,3,4,5,6]
y1,y2 = [78,69,80,75,69,89],[80,68,72,81,70,77]

plt.subplot(2,1,1)
plt.plot(x, y1, 'ro-',linewidth=2.5)
plt.ylabel(u'分数', fontproperties=font,size = 14)
plt.title(u'二年级各班语文平均成绩变化趋势图',
fontproperties=font, size = 14)
plt.axis([0.5,6.5,0,100])

plt.subplot(2,1,2)
plt.plot(x, y2, 'bo-',linewidth=2.5)
plt.ylabel(u'分数', fontproperties=font,size = 14)
plt.title(u'三年级各班语文平均成绩变化趋势图',
fontproperties=font,size =14)
plt.axis([0.5,6.5,0,100])

plt.show()

```



```

# 例子2
import matplotlib.pyplot as plt # 引用matplotlib.pyplot库
import numpy as np # 引用numpy库
# 设置中文字体为简宋10号字
from matplotlib.font_manager import FontProperties
pic_font =
FontProperties(fname=r"c:\windows\fonts\simsun.ttc",
size=10)

# 会员A、B、C、D每天的伤害数值
memberA = [1127968,1193973,1330385,1211292,1365868]
memberB = [950600,967387,1102495,992571,1132938]
memberC = [797236,872468,922418,896010,950669]

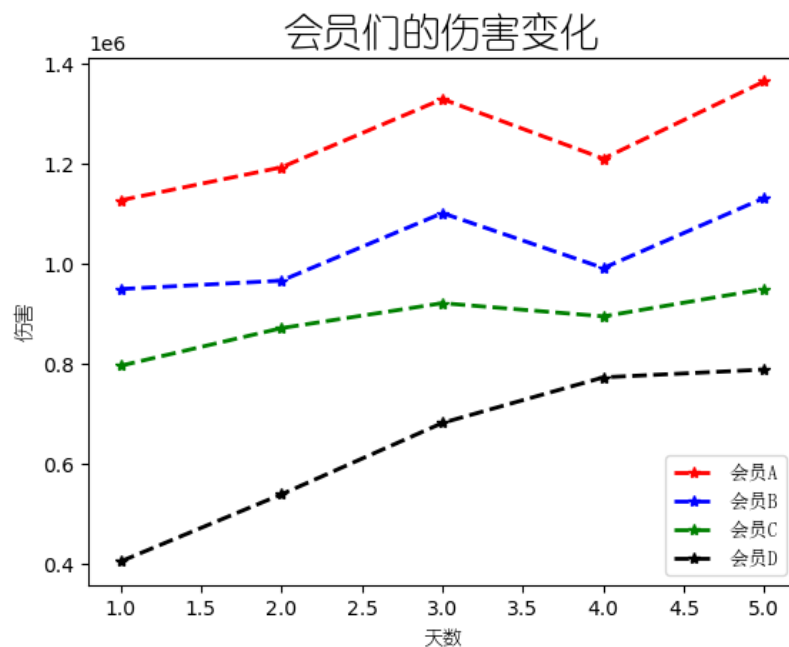
```

```

memberD = [405868,540142,682967,774129,789672]
x = np.arange(1.0,6.0,1.0) # x轴从1.0到5.0的5个值的数组
# 分别画出四个会员每天伤害变化的折线图
a = plt.plot(x,memberA,'r*--',linewidth=2,label='会员A')
b = plt.plot(x,memberB,'b*--',linewidth=2,label='会员B')
c = plt.plot(x,memberC,'g*--',linewidth=2,label='会员C')
d = plt.plot(x,memberD,'k*--',linewidth=2,label='会员D')
# 设置x、y轴名称和图题
plt.xlabel('天数',fontproperties='YouYuan')
plt.ylabel('伤害',fontproperties='YouYuan')
plt.title('会员们的伤害变化',fontproperties='YouYuan',size
= 20)
plt.legend(prop =pic_font) # 设置图例及其中文字体

plt.show()

```



## 2.饼图

```

pie(x, explode=None, labels=None, colors=None,
autopct=None, pctdistance=0.6,
shadow=False, labeldistance=1.1, startangle=None,
radius=None)
...

```

参数：x (每一块)的比例，如果 $\text{sum}(x) > 1$ 会使用 $\text{sum}(x)$ 归一化

labels (每一块)饼图外侧显示的说明文字

explode (每一块)离开中心距离

startangle 起始绘制角度,默认图是从x轴正方向逆时针画起,如设定=90则从y轴正方向画起

shadow表示是否阴影

labeldistance label绘制位置,相对于半径的比例,如<1则绘制在饼图内侧

**autopct** 控制饼图内百分比设置,可以使用**format**字符串或者**format function**,**'%1.1f'**指小数点前后位数(没有会用空格补齐),不写则不标注  
**pctdistance** 类似于**labeldistance**,指定**autopct**的位置刻度  
**radius** 控制饼图半径  
'''

```
# 例子
import matplotlib.pyplot as plt
import numpy as np

from matplotlib.font_manager import FontProperties

data =
[55,63,80,75,69,89,92,91,50,84,86,77,95,81,92,74,60,88,64,
71]

BEST,GOOD,PASS,FAIL = 0,0,0,0
for score in data:
    if score < 60:
        FAIL = FAIL + 1
    elif score < 80:
        PASS = PASS + 1
    elif score < 90:
        GOOD = GOOD + 1
    else:
        BEST = BEST + 1

sizes = BEST,GOOD,PASS,FAIL
labels=[u'优', u'良', u'中',u'差']
#设置各个子块的填充颜色
colors = ['blue', 'orange', 'lime', 'lightgray']
#突出显示第二项,即得分为良的学生所占的比例
explode = (0.1, 0, 0, 0)
pie = plt.pie(sizes, explode=explode, labels=labels,
colors=colors,
autopct='%1.1f%%', shadow=True, startangle=-90)

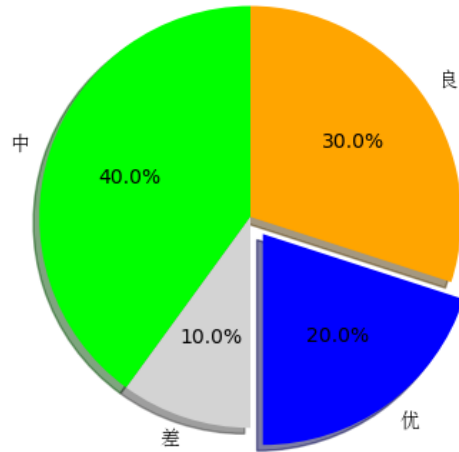
#设置中文
for font in pie[1]:

    font.set_fontproperties(FontProperties(fname=r"c:\windows
\fonts\simsum.ttc"))

plt.show()
'''

或者直接使用第一种设置中文的方法,则上面的for... 可以删去
'''
```





### 3.柱形图

垂直的（水平的调节参数`orientation`，或者直接用`plt.barh()`）：

```
plt.bar(x, height, width=0.8, bottom=None, hold=None,
**kwargs)
'''
x --设置柱状图的在x轴的位置
height --设置柱状图的高度（即y的值）
width --设置柱状图的宽度，默认为 0.8
bottom --设置柱状图底部的 y 轴坐标，默认值为 None
**kwargs:
color --指定柱状图的填充显示颜色，默认值为 None（蓝色）
edgecolor --指定柱状图的边缘显示颜色，默认值为 None
linewidth --指定柱状图的边缘线宽，默认值是 None
hatch --填充方式：1.每一种hatch字符代表填充的形状，其中'*'代表五角星， '.'代表以点填充， 'o'代表圆形填充；2. 重复使用字符，实现的是类似加密，使得填充字符间距更小，当bar_width比较小时看起来效果好，注意并不是越密越好，过密会失去字符特征，应该在尝试中选择效果最好的为准。（摘自：https://www.jianshu.com/p/e8033e25e1aa）
tick_label --指定刻度显示，默认值为 None
xerr --指定柱状图的 x 轴的误差显示。默认值为 None
yerr --指定柱状图的 y 轴的误差显示。默认值为 None
align --设置柱状图的对齐方式，如果是 left，当柱状图的显示为垂直方向时代表左边对齐，当柱状图的显示为水平方向时代表下边对齐
orientation --设置柱状图的朝向，vertical 或 horizontal，默认值为 vertical
log --指定坐标轴的尺度为 log 方式
'''
```

```
# 例子1
# 并列柱状图
```

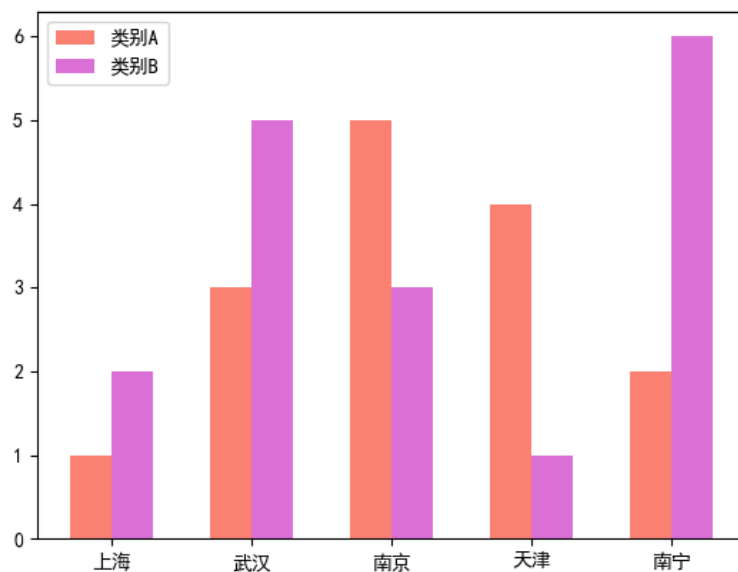
```
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei']#设置字体以便支持中文
import numpy as np

x=np.arange(5)#柱状图在横坐标上的位置
#列出你要显示的数据，数据的列表长度与x长度相同
y1=[1,3,5,4,2]
y2=[2,5,3,1,6]

bar_width=0.3#设置柱状图的宽度
tick_label=['上海','武汉','南京','天津','南宁']

#绘制并列柱状图
plt.bar(x,y1,bar_width,color='salmon',label='类别A')
plt.bar(x+bar_width,y2,bar_width,color='orchid',label='类别B')

plt.legend()#显示图例，即label
plt.xticks(x+bar_width/2,tick_label)#显示x坐标轴的标签,即tick_label,调整位置，使其落在两个直方图中间位置
plt.show()
```



```
# 例子2
import matplotlib.pyplot as plt #导入matplotlib.pyplot模块，别名取为plt
import numpy as np #导入numpy库，别名起为np

#为了在生成的图像中显示中文，需要设置字体属性
from matplotlib.font_manager import FontProperties
#设置字体对象，本例选择的是简宋字体，字号是14
```

```

pic_font =
FontProperties(fname=r"c:\windows\fonts\simsun.ttc",
size=14)

data = [(75,6),(72,5),(70,3),(77,8), (88,10),(85,6),
(85,7),(87,9),(88,8)]

chineseMean ,chineseStd = [],[]
mathMean,mathStd = [],[]
englishMean,englishStd = [],[]

for i in range(0,len(data),3):
    chineseMean.append(data[i]
[0]),chineseStd.append(data[i][1])
    mathMean.append(data[i+1][0]),mathStd.append(data[i+1]
[1])
    englishMean.append(data[i+2]
[0]),englishStd.append(data[i+2][1])

N = 3
plt.rcParams['font.sans-serif'] = ['SimHei']

ind = np.arange(N) # the x locations for the groups
width = 0.25 # the width of the bars

#设置语文、数学、英语的柱状图填充形式
pattern1,pattern2,pattern3 = "/", "*", "."
#这里处理原始数据，将原始数据分为三组，篇幅所限，从略，请查看源码
fig, ax = plt.subplots() #创建一个子图
#绘制语文成绩柱状图
rects1 = ax.bar(ind, chineseMean,
width,hatch=pattern1,edgecolor='black',color='white',
yerr=chineseStd)
#绘制数学成绩柱状图
rects2 = ax.bar(ind + width, mathMean, width,
hatch=pattern2,edgecolor='black',color='white',
yerr=mathStd)
#绘制英语成绩柱状图
rects3=ax.bar(ind+2*width,englishMean,width,hatch=pattern3
,edgecolor='black',color='white', yerr=englishStd)

#设置语文、数学、英语成绩的图例，位置为图表的左上角
ax.legend((rects1[0], rects2[0],rects3[0]), (u'语文', u'数
学',u'英语'), prop={'family':'SimHei','size':8},loc='upper
left',bbox_to_anchor=(0.04, 0.99))
#ax.legend((rects1[0], rects2[0],rects3[0]), (u'语文', u'数
学',u'英语'), prop={'family':'SimHei','size':14},loc='upper
center', bbox_to_anchor=(0.12,1.0),ncol=3)
#定义标注柱形图高度的函数
def autolabel(rects):

```

```

for rect in rects:
    height = rect.get_height()          #获得柱形的高度
    ax.text(rect.get_x() + rect.get_width()/2.,
1.05*height, '%d' % int(height), ha='center', va='bottom')
#添加文本

#调用autolabel函数，在柱状图上标注高度
autolabel(rects1)
autolabel(rects2)
autolabel(rects3)
ax.set_ylabel('分数')

plt.title(u'二年级语文、数学和英语成绩')
# 设置x轴标签
ax.set_xticks(ind+width)
ax.set_xticklabels( ('二年级一班', '二年级二班', '二年级三班')
)

plt.show()

```

