Python 数据处理：pandas

可视化图表

2024

目录

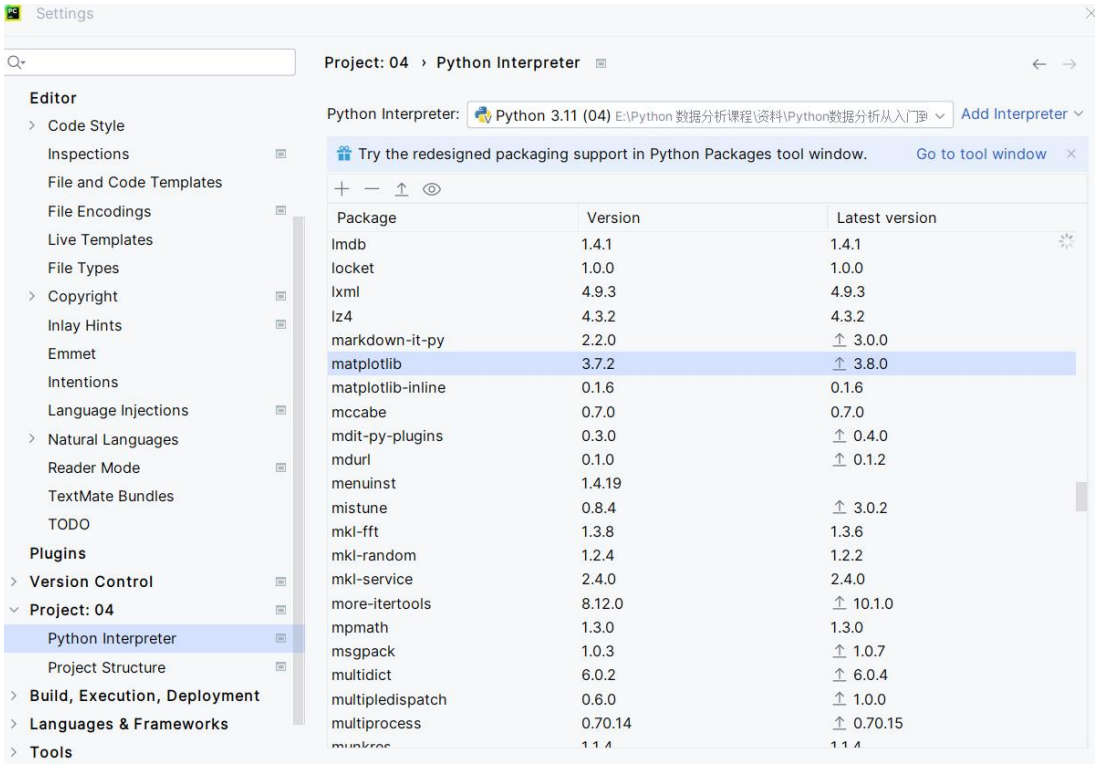
CONTENTS

01

matplotlib简介

1.matplotlib简介

- Matplotlib是一个基于Python的二维绘图库，是最基本的python绘图库
- Matplotlib中绘图函数最好基于list，np.array数据或者np.ma.masked_array作为输入类型。



02

数图表常用设置

2. 图表常用设置

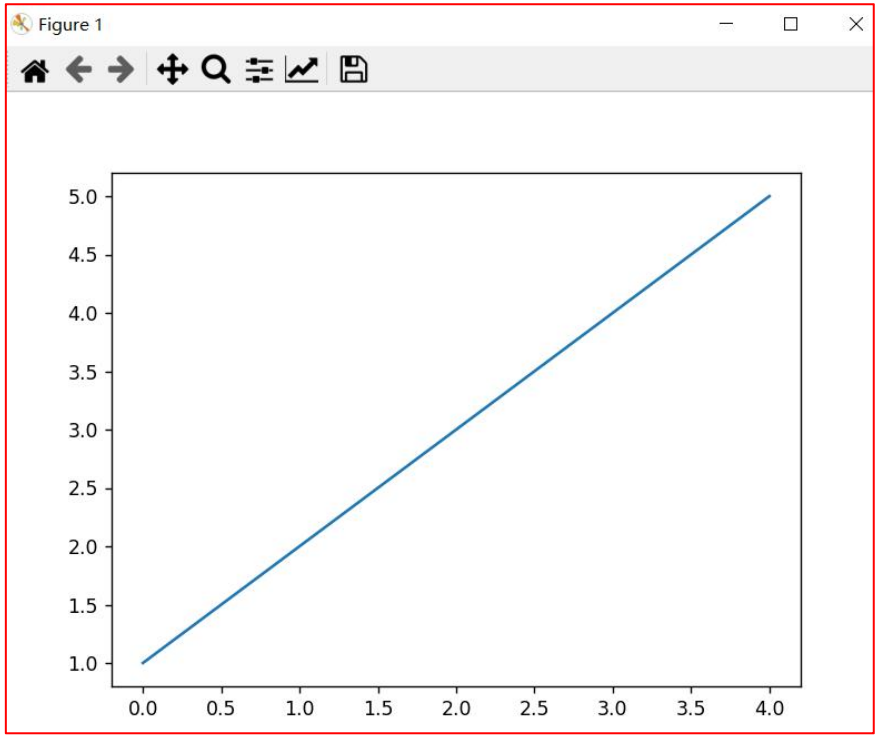
- matplotlib.pyplot 导入（主要使用plot函数）
- plt.plot(x,y,format_string,**kwargs)
  - x,y :x,y 轴数据
  - format_string:控制格式的字符串，包括颜色，线条样式等

- 第一个图表

代码

```
import matplotlib.pyplot as plt
#折线图
plt.plot([1, 2, 3, 4,5])
plt.show()
```

输出



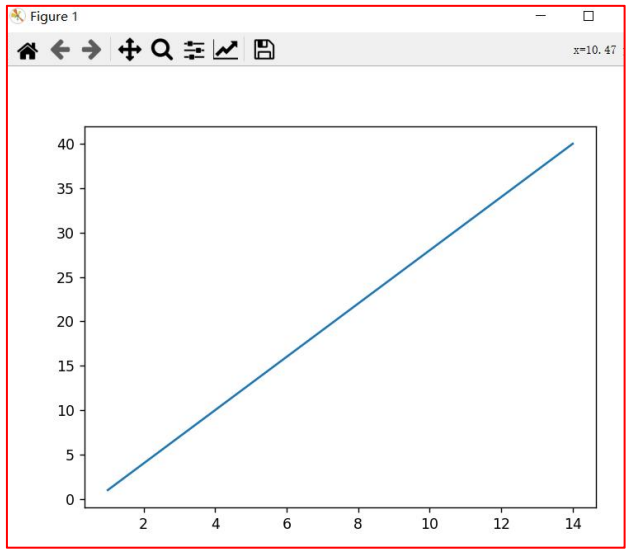
2. 图表常用设置

- 简单折线

代码

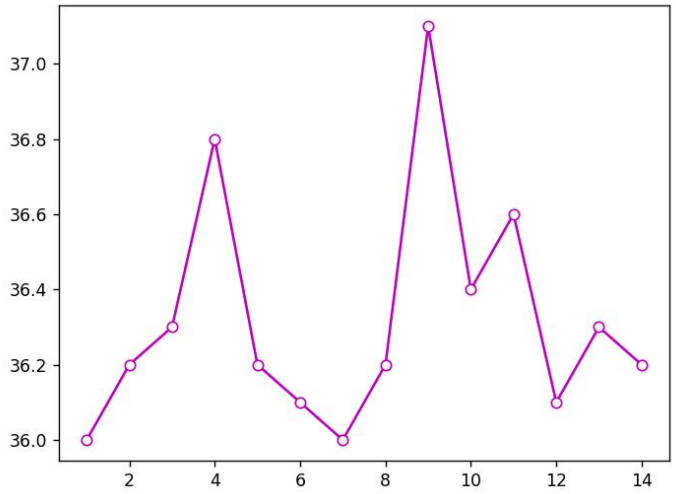
```
import matplotlib.pyplot as plt
#折线图
#range()函数创建整数列表
x = range(1,15,1)
y = range(1,42,3)
plt.plot(x,y)
plt.show()
```

输出



- 导入数据，绘图

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_excel('体温.xls') #导入Excel文件
#折线图
x =df['日期']                    #x轴数据
y=df['体温']                     #y轴数据
plt.plot(x,y,color='m',linestyle='-',marker='o',mfc='w')
plt.show()
```



- color：颜色

设置值	说明	设置值	说明
b	蓝色	m	洋红色
g	绿色	y	黄色
r	红色	k	黑色
c	蓝绿色	w	白色
#FFFF00	黄色，十六进制颜色值	0.5	灰度字符串

- linestyle：线条样式
  - “-”：实线，默认值
  - “--”：双划线
  - “-.”：点划线
  - “：”：虚线

- marker：标记样式

设置值	说明	设置值	说明	设置值	说明
.	点标记	1	下三角标记	h	竖六边形标记
,	像素标记	2	上三角标记	H	横六边形标记
o	实心圆标记	3	左花三角标记	+	加号标记
v	倒三角标记	4	右花三角标记	X	叉号标记
^	上三角标记	s	实心正方形标记	D	大菱形标记
>	右三角标记	p	实心五角星标记	d	小菱形标记
<	左三角标记	*	星形标记	|	垂直线标记

- mfc：填充颜色

2. 图表常用设置

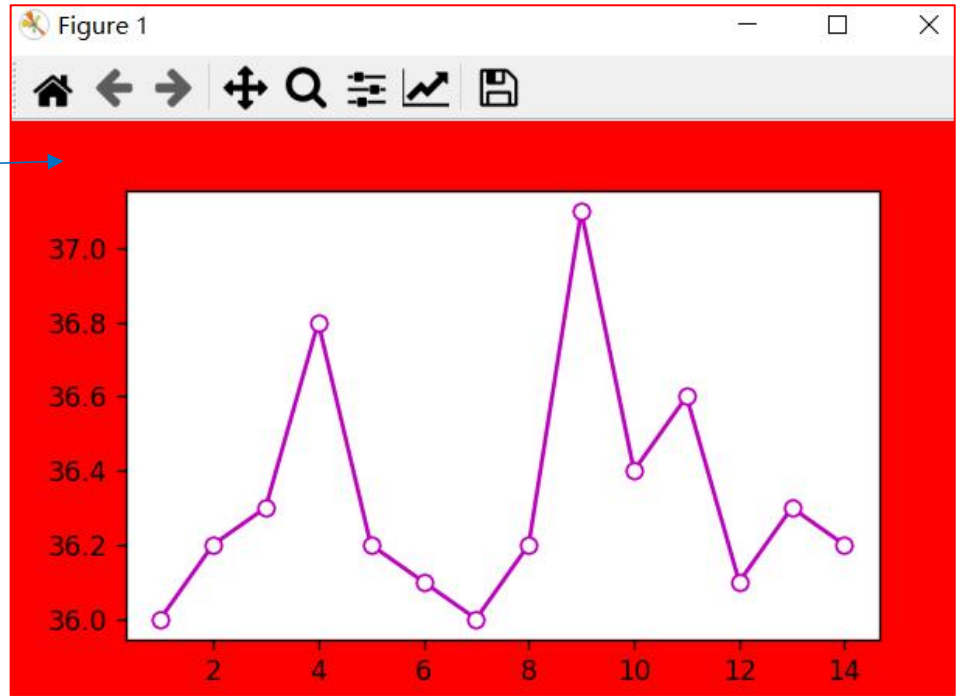
- 设置画布颜色

代码

```
import pandas as pd
import matplotlib.pyplot as plt
fig=plt.figure(figsize=(5,3),facecolor='r')
#导入Excel文件
df=pd.read_excel('体温.xls')
#折线图
x=df['日期']                #x轴数据
y=df['体温']                #y轴数据
plt.plot(x,y,color='m',linestyle='-',marker='o',mfc='w')
plt.show()
```

图形大小：
500*300像素

输出



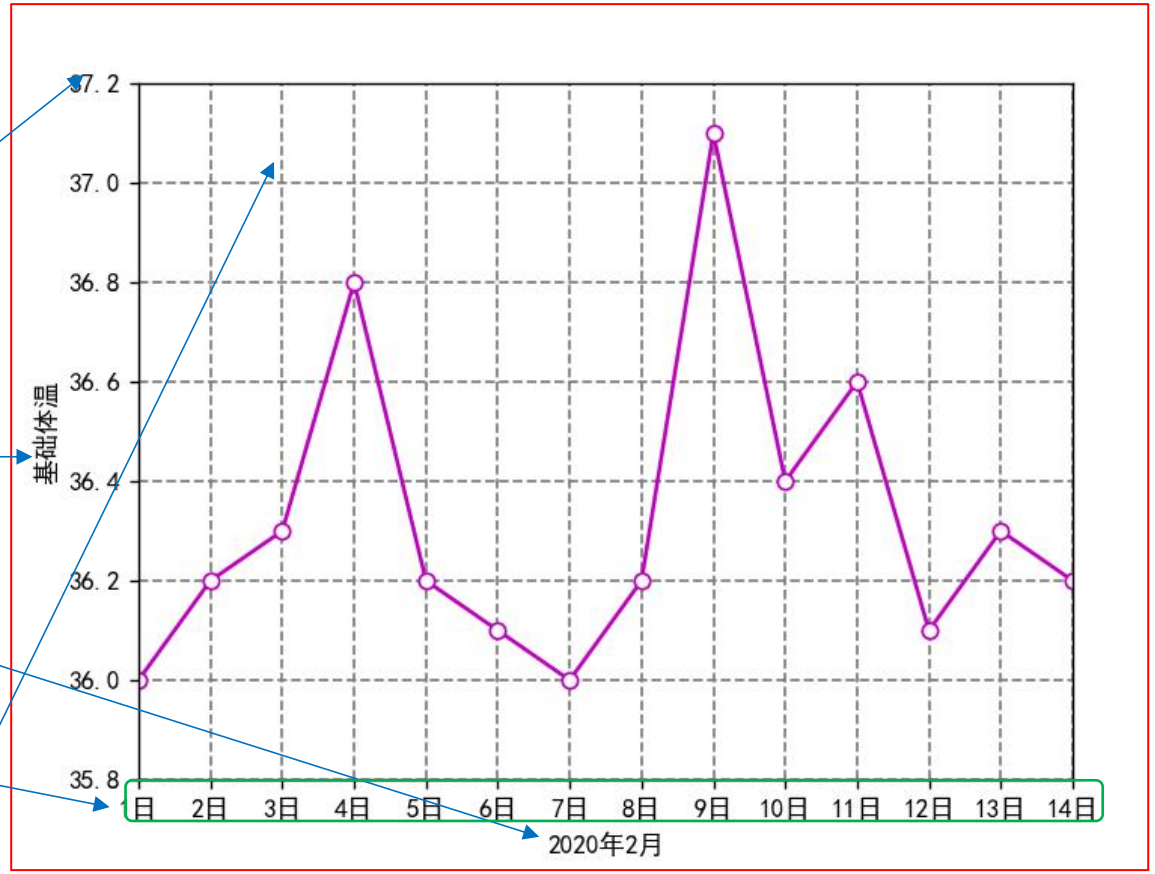
2. 图表常用设置

- 设置坐标标题，坐标刻度，网格等

代码

```
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif']=['SimHei'] #解决中文乱码
df=pd.read_excel('体温.xls')                #导入Excel文件
#折线图
x=df['日期']                    #x轴数据
y=df['体温']                    #y轴数据
plt.plot(x,y,color='m',linestyle='-',marker='o',mfc='w')
plt.xlabel('2020年2月')          #x轴标题
plt.ylabel('基础体温')           #y轴标题
#设置x轴刻度及标签
dates=['1日','2日','3日','4日','5日',
       '6日','7日','8日','9日','10日',
       '11日','12日','13日','14日']
plt.xticks(range(1,15,1),dates)
#坐标轴范围
plt.xlim(1,14)
plt.ylim(35.8,37.2)
plt.grid(color='0.5',linestyle='--',linewidth=1,axis='both')
plt.show()
```

输出



2. 图表常用设置

- 添加文本标签
  - plt.text(x,y,s,fontdict=None,withdash=Flash, **kwargs)
    - x:x坐标轴的值
    - y:y坐标轴的值
    - s：字符串，注释的内容
    - **kwargs：关键字参数：如 字体大小：fontsize=12；垂直对齐方式：ha= ‘center’ ；水平对齐方法：va= ‘bottom’

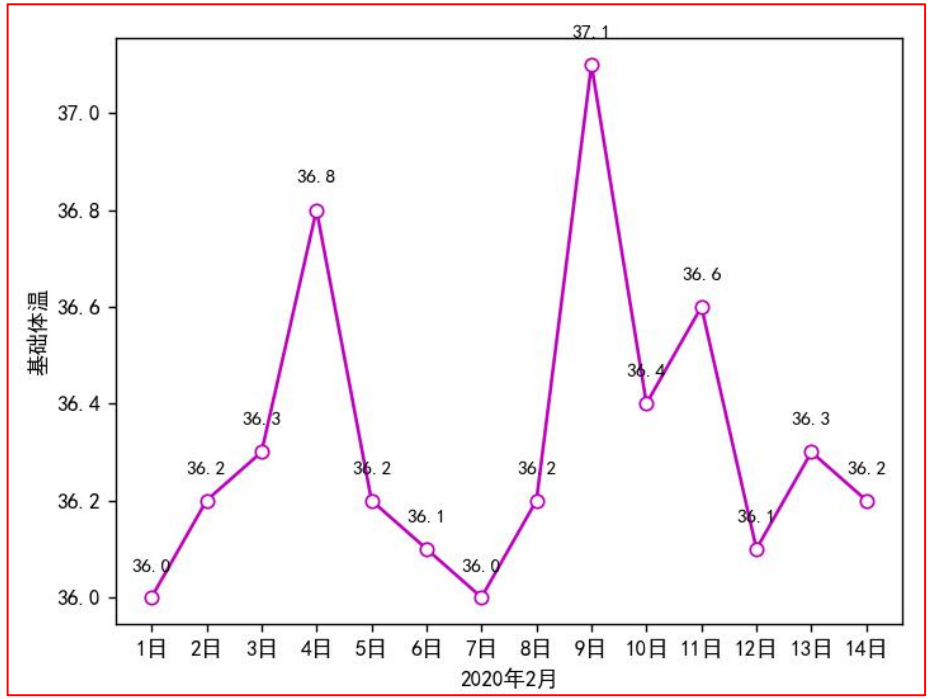
代码

```
for a,b in zip(x,y):
    plt.text(a,b+0.05,'%.1f'%b,ha = 'center',va = 'bottom',fontsize=9)
```

a，b在 x,y 组合里面遍历

浮点b保留一位小数，%为占位符

输出



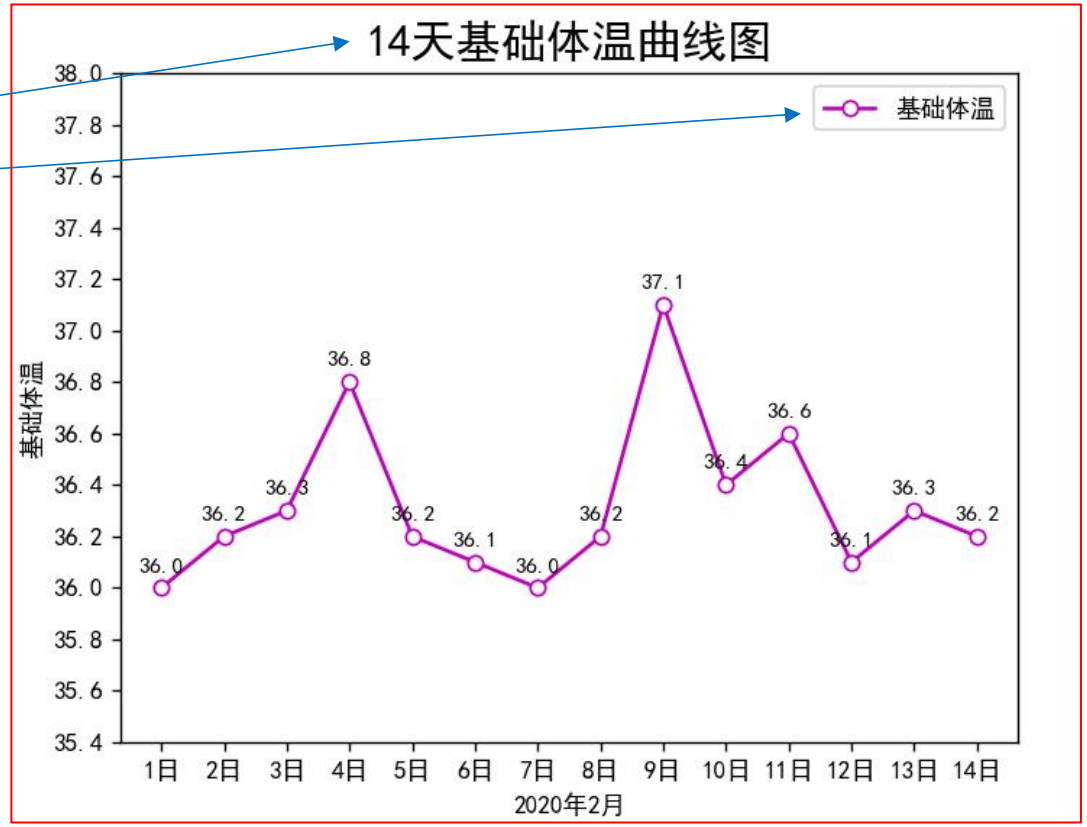
2. 图表常用设置

- 图题和图例
  - plt.title(label,fontdict=None,loc='center',pad='None',**kwargs)
  - plt.legend()

```
plt.title('14天基础体温曲线图',fontsize='18')
#图例
plt.legend(('基础体温',),loc='upper right',fontsize=10)
plt.show()
```

代码

输出



03

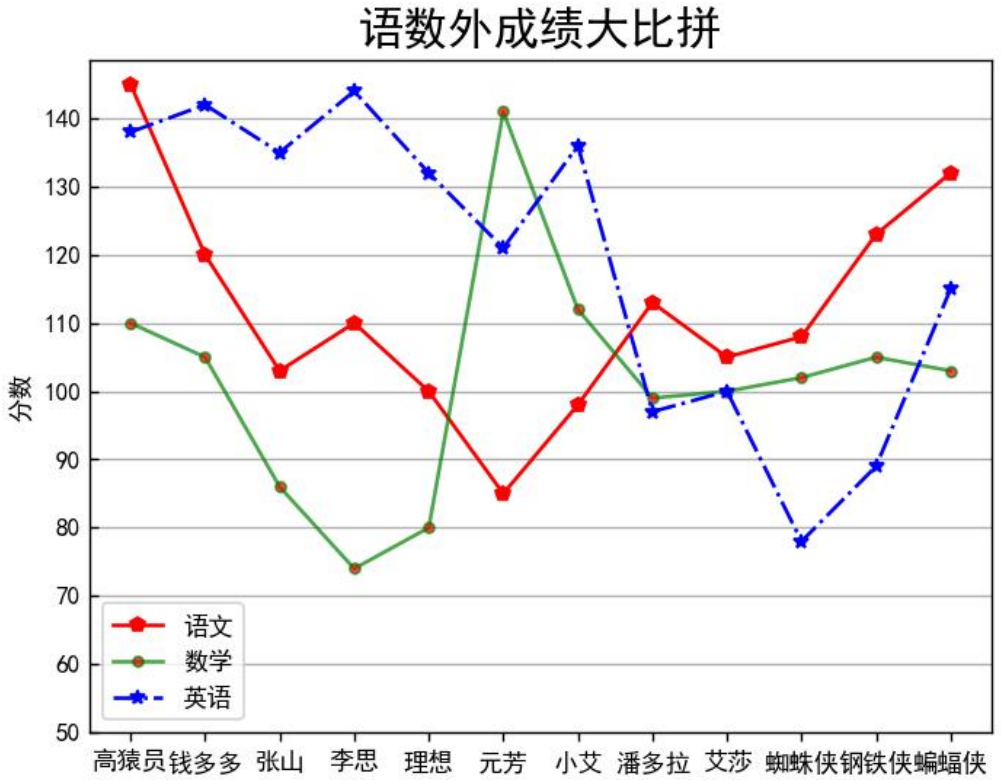
常用图表绘制

- 折线图

代码

```
import pandas as pd
import matplotlib.pyplot as plt
df1=pd.read_excel('data.xls')                    #导入Excel文件
#多折线图
x1=df1['姓名']
y1=df1['语文']
y2=df1['数学']
y3=df1['英语']
plt.rcParams['font.sans-serif']=['SimHei']       #解决中文乱码
plt.rcParams['xtick.direction'] = 'out'          #x轴的刻度线向外显示
plt.rcParams['ytick.direction'] = 'in'           #y轴的刻度线向内显示
plt.title('语数外成绩大比拼',fontsize='18')      #图表标题
plt.plot(x1,y1,label='语文',color='r',marker='p')
plt.plot(x1,y2,label='数学',color='g',marker='.',mfc='r',ms=8,alpha=0.7)
plt.plot(x1,y3,label='英语',color='b',linestyle='-.',marker='*')
plt.grid(axis='y')                               #显示网格关闭y轴
plt.ylabel('分数')
plt.yticks(range(50,150,10))
plt.legend(['语文','数学','英语'])               #图例
plt.show()
```

输出

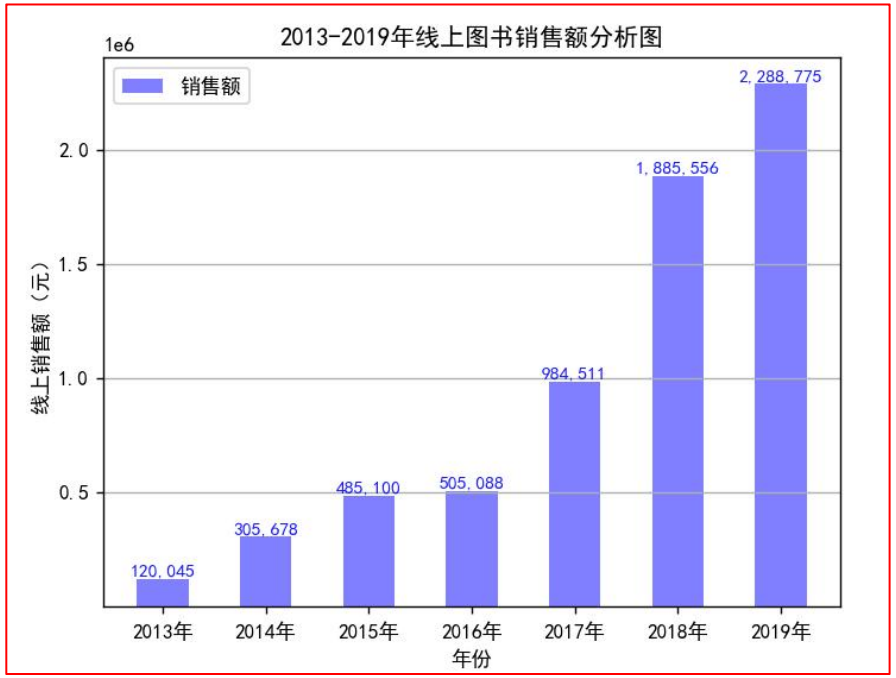


- 绘制柱状图
  - plt.bar(x,height,width,align='center',bottom，**kwargs)
    - x:x轴数据
    - height：柱子的高度，y轴数据
    - width：浮点型，柱子的宽度，默认0.8，可以指定固定值
    - bottom：柱子底部坐标
    - align:对齐方式，如center，edge，默认center
    - **kwargs：关键字参数，color， alpha（透明度），label（每个柱子显示的标签）

代码

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('books.xlsx')
plt.rcParams['font.sans-serif']=['SimHei'] #解决中文乱码
x=df['年份']
height=df['销售额']
plt.grid(axis="y", which="major")  # 生成虚线网格
plt.xlabel('年份')
plt.ylabel('线上销售额（元）')
plt.title('2013-2019年线上图书销售额分析图')
plt.bar(x,height,width = 0.5,align='center',color = 'b',alpha=0.5,bottom=1)
#设置每个柱子的文本标签,format(b,',')格式化销售额为千位分隔符格式
for a,b in zip(x,height):
    plt.text(a, b,format(b,','), ha='center',
             va= 'bottom',fontsize=9,color = 'b',alpha=0.9)
plt.legend(['销售额'])
plt.show()
```

输出

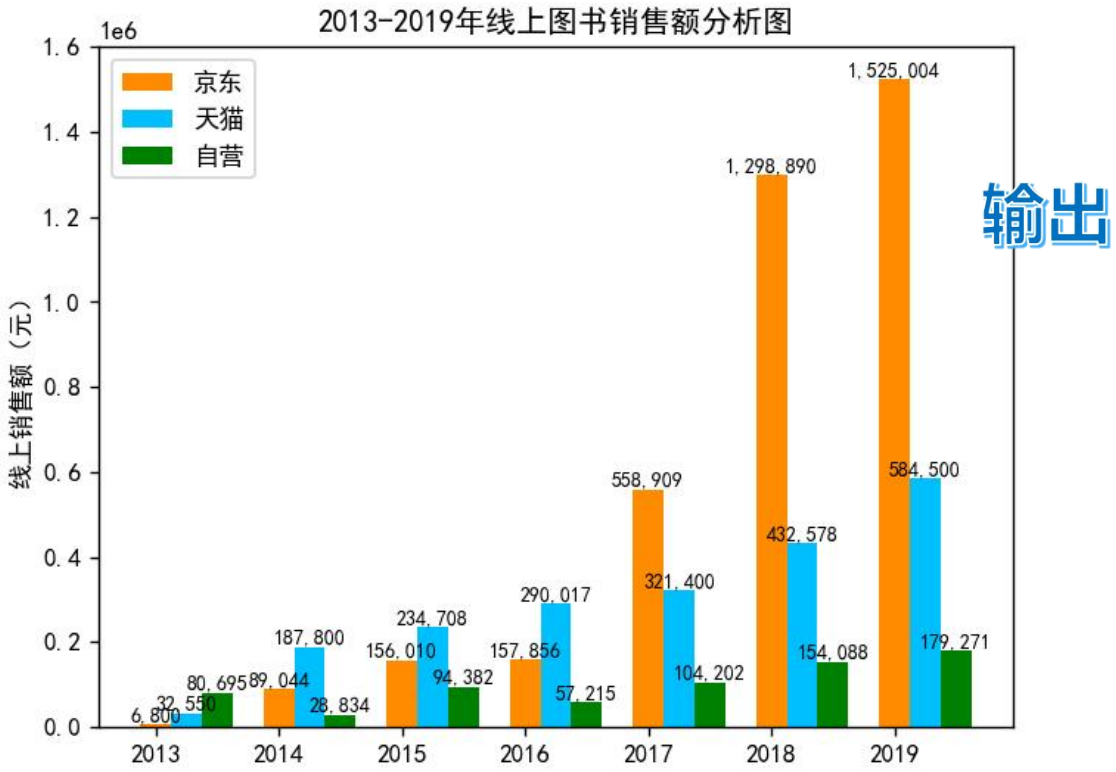


- 多柱状图

代码

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('books.xlsx',sheet_name='Sheet2')
plt.rcParams['font.sans-serif']=['SimHei'] #解决中文乱码
x=df['年份']
y1=df['京东']
y2=df['天猫']
y3=df['自营']
width =0.25  #柱子宽度，若显示n个柱子，则width值需小于1/n ，否则柱子会出现重叠
#y轴标签
plt.ylabel('线上销售额（元）')
#图表标题
plt.title('2013-2019年线上图书销售额分析图')
plt.bar(x,y1,width = width,color = 'darkorange')
plt.bar(x+width,y2,width = width,color = 'deepskyblue')
plt.bar(x+2*width,y3,width = width,color = 'g')
#设置每个柱子的文本标签,format(b,',')格式化销售额为千位分隔符格式
for a,b in zip(x,y1):
    plt.text(a, b,format(b,','), ha='center', va= 'bottom',fontsize=8)
for a,b in zip(x,y2):
    plt.text(a+width, b,format(b,','), ha='center', va= 'bottom',fontsize=8)
for a, b in zip(x, y3):
    plt.text(a + 2*width, b, format(b, ','), ha='center', va='bottom', fontsize=8)
plt.legend(['京东','天猫','自营'])#图例
plt.show()
```

输出

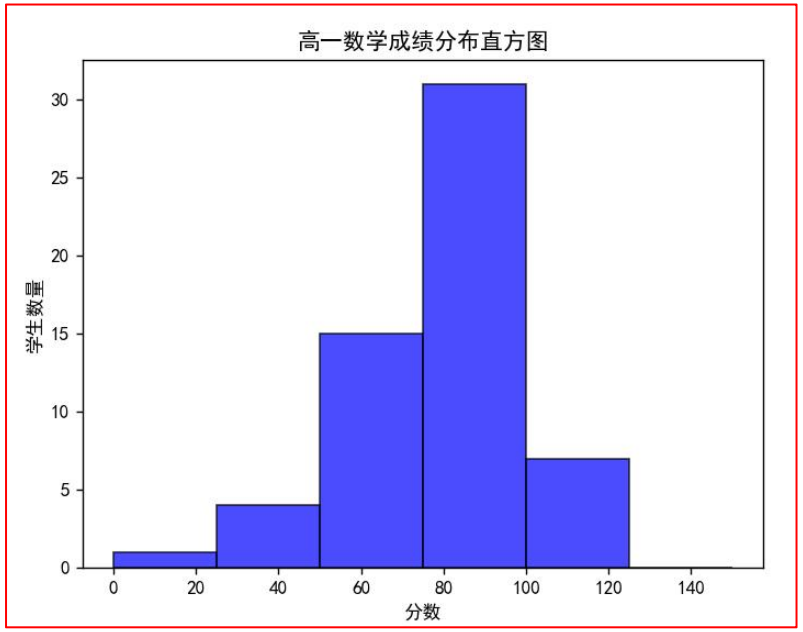


- 直方图
  - plt.hist(x, bins=None, range=None, density=False, weights=None, cumulative=False, bottom=None, histtype='bar',align='mid', orientation='vertical', rwidth=None, log=False, color=None, label=None, stacked=False, *, data=None, **kwargs,)
  - x:数据集，最终的直方图将对数据集进行统计
  - bins：统计数据的区间分布
  - density：默认为None，如果为True，则显示频率统计结果

代码

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('grade1.xls')
plt.rcParams['font.sans-serif']=['SimHei']
x=df['得分']
plt.xlabel('分数')
plt.ylabel('学生数量')
# 显示图标题
plt.title("高一数学成绩分布直方图")
plt.hist(x, bins = [0,25,50,75,100,125,150],
         facecolor="blue", edgecolor="black", alpha=0.7)
plt.show()
```

输出

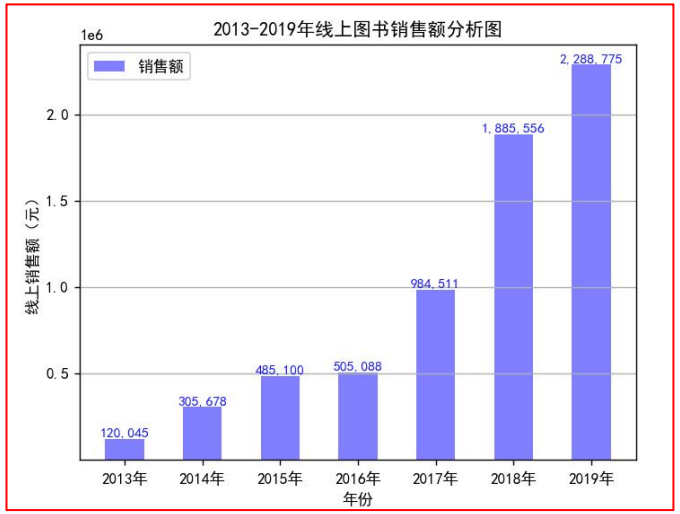


3.常用图表绘制

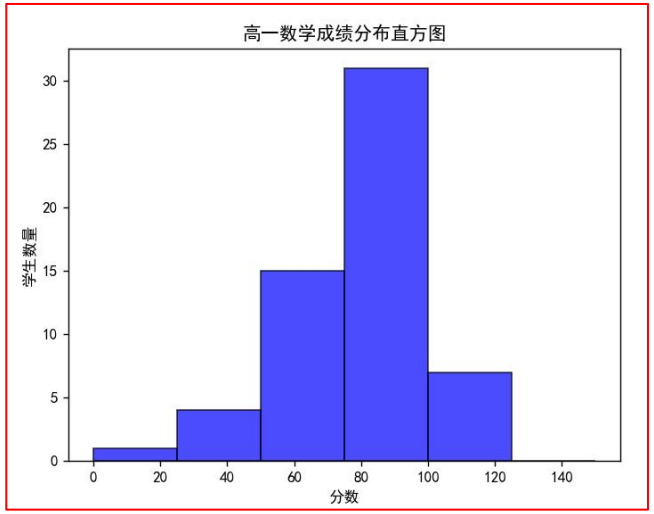
柱状图与直方图的对比与区别

- 直方图展示数据的分布，柱状图比较数据的大小。
- 直方图X轴为定量数据，柱状图X轴为分类数据。因此，直方图上的每个条形都是不可移动的，X轴上的区间是连续的、固定的。而柱状图上的每个条形是可以随意排序的，有的情况下需要按照分类数据的名称排列，有的则需要按照数值的大小排列。
- 直方图柱子无间隔，柱状图条形有间隔
- 直方图条形宽度可不一，柱状图条形宽度须一致。
- 柱状图条形的宽度因为没有数值含义，所以宽度必须一致。但是在直方图中，条形的宽度代表了区间的长度，根据区间的不同，条形的宽度可以不同，但理论上应为单位长度的倍数。
- https://blog.csdn.net/Px01lh8/article/details/108030451

年份	销售额
2013年	120,045.00
2014年	305,678.00
2015年	485,100.00
2016年	505,088.00
2017年	984,511.00
2018年	1,885,556.00
2019年	2,288,775.00



序号	得分
1	107
2	77
3	94
4	87
5	90
6	95
7	92
8	93
9	86
10	101
11	85
12	76
13	96
14	99
15	110
16	89
17	101
18	110



- 绘制饼图
- plt.pie(x，explode=None, labels=None, colors=None, autopct=None，pctdistance=0.6, shadow=False，labeldistance=1.1, startangle=0，radius=1， counterclock=True, wedgeprops=None，textprops=None， center=(0, 0)， frame=False, rotatelabels=False， *， normalize=True， hatch=None， data=None,)
  - x：每一块饼图的比例，如果sum（x）>1，会归一化处理
  - labels：每一块饼图侧面显示的说明文字
  - explode：每一块饼形图离中心的距离。
  - startangle：起始绘制角度，默认是从x轴正方向逆时针画起，如设置值为90，则从y轴正方向画起。
  - shadow：在饼形图下面画一个阴影，默认值为False，即不画阴影。
  - labeldistance：标记的绘制位置，相对于半径的比例，默认值为1.1，如＜1，则绘制在饼图内侧。
  - autopct：设置饼图百分比，可以使用格式化字符串或format函数。如＇％1.1f＇保留小数点的后1位。
  - pctdistance：类似于labeldistance参数，指定百分比的位置刻度，默认值为0.6。
  - radius：饼图半径，默认值为1。

- counterclock：指定指针方向，布尔型，可选参数，默认值为True，表示逆时针；如果值为False，则表示顺时针。
- wedgeprops：字典类型，可选参数，默认值为None。字典传递给wedge对象用来画一个饼图。例如 wedgeprops = {'linewidth'：2} 设置 wedge 线宽为2。
- textprops：设置标签和比例文字的格式，字典类型，可选参数，默认值为None。传递给text对象的字典参数。
- center：浮点类型的列表，可选参数，默认值为（0,0），表示图表中心位置。
- frame：布尔型，可选参数，默认值为False，不显示轴框架（也就是网格）；如果值为True，则显示轴框架，与grid函数配合使用。在实际应用中建议使用默认设置，因为显示轴框架会干扰饼形图效果。
- rotatelabels：布尔型，可选参数，默认值为False；如果值为True，则旋转到每个标签到指定的角度。

- 基础饼图实例

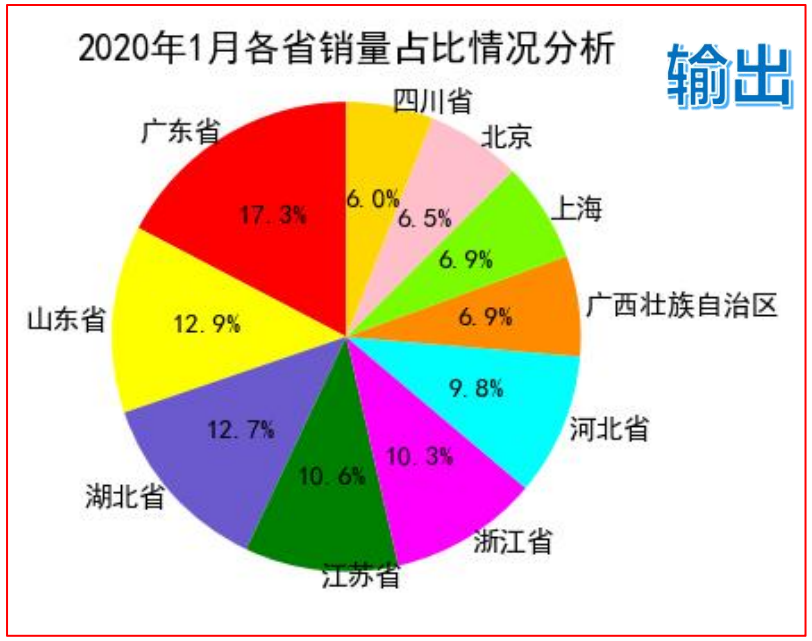
代码

```
import pandas as pd
from matplotlib import pyplot as plt
df1 = pd.read_excel('data2.xls')
plt.rcParams['font.sans-serif']=['SimHei'] #解决中文乱码
plt.figure(figsize=(5,3)) #设置画布大小
labels = df1['省']
sizes = df1['销量']
#设置饼形图每块的颜色
colors = ['red', 'yellow', 'slateblue', 'green',
          'magenta','cyan','darkorange','lawngreen','pink','gold']
plt.pie(sizes, #绘图数据
        labels=labels,#添加区域水平标签
        colors=colors,# 设置饼图的自定义填充色
        labeldistance=1.02,#设置各扇形标签（图例）与圆心的距离
        autopct='%.1f%%',# 设置百分比的格式，这里保留一位小数
        startangle=90,# 设置饼图的初始角度
        radius = 0.5, # 设置饼图的半径
        center = (0.2,0.2), # 设置饼图的原点
        textprops = {'fontsize':9, 'color':'k'}, # 设置文本标签的属性值
        pctdistance=0.6)# 设置百分比标签与圆心的距离
plt.axis('equal')# 设置x，y轴刻度一致，保证饼图为圆形
plt.title('2020年1月各省销量占比情况分析')
plt.show()
```

数据

省	销量
广东省	10777
山东省	8018
湖北省	7921
江苏省	6582
浙江省	6396
河北省	6096
广西壮族自	4303
上海	4277
北京	4071
四川省	3723

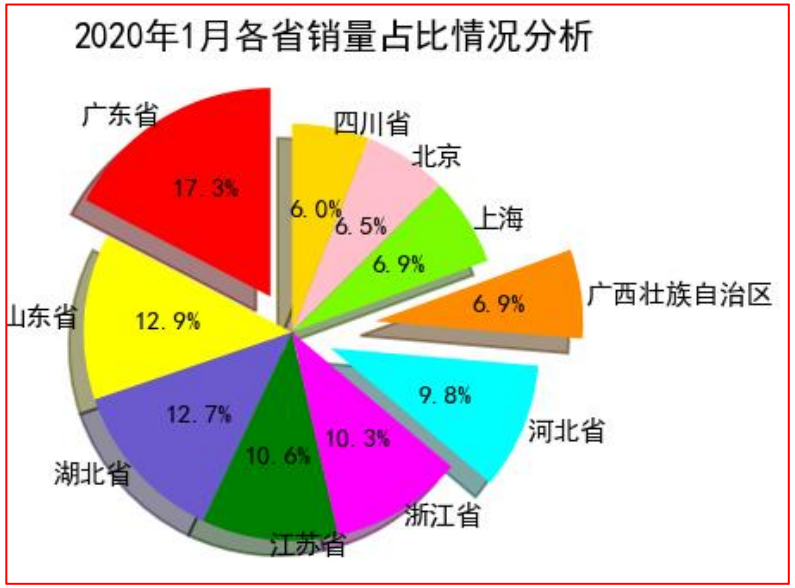
输出



- 一些细节

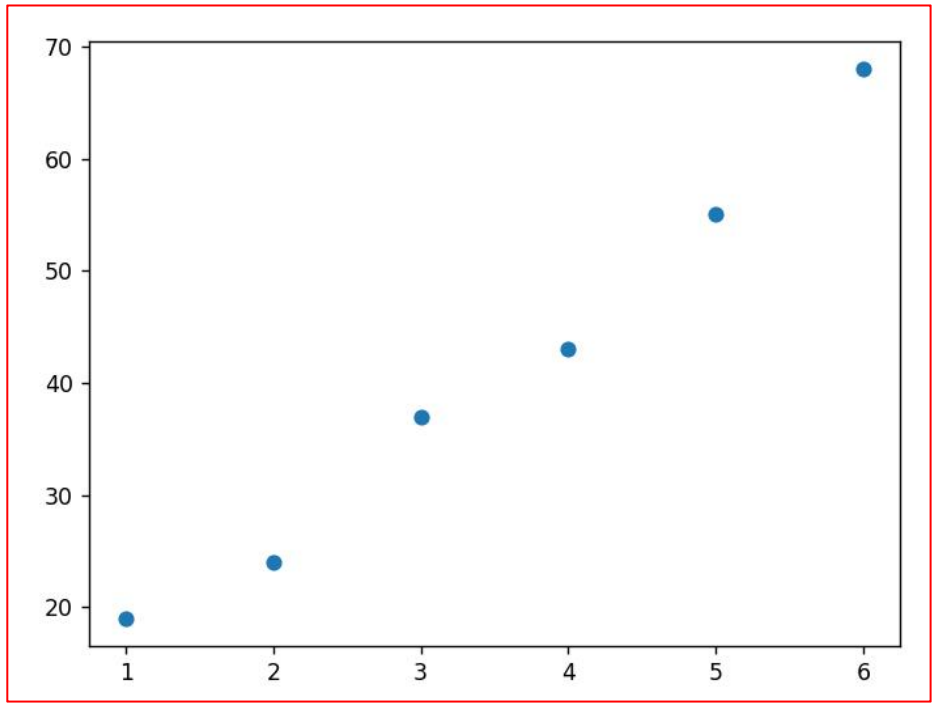
```
explode=(0.1,0,0,0,0,0.1,0.2,0,0,0),# 设置爆炸视图
shadow=True)   #设置阴影
```

```
wedgeprops={'width':0.4,'edgecolor':'k'})   #设置中空
```



- 散点图
  - plt.scatter(x, y, s=None, c=None, marker=None, cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, *, edgecolors=None, plotnonfinite=False, data=None, **kwargs,)

```
import matplotlib.pyplot as plt
x=[1,2,3,4,5,6]
y=[19,24,37,43,55,68]
plt.scatter(x, y)
plt.show()
```

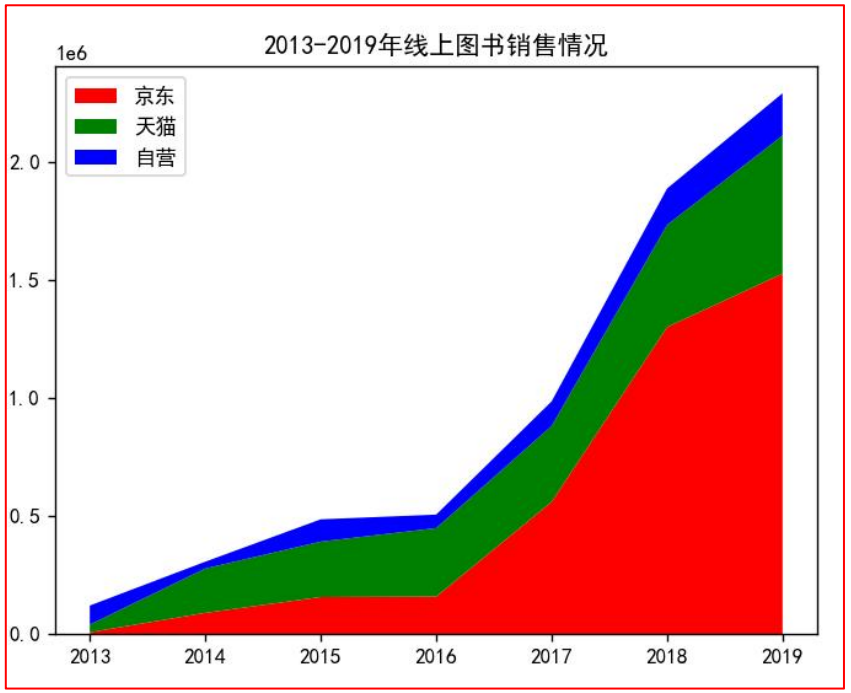


- 绘制面积图
  - plt.stackplot(x, *args, labels=(), colors=None, baseline='zero', data=None, **kwargs,)
    - args：可以是多个y轴数据

代码

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('books.xlsx',sheet_name='Sheet2')
plt.rcParams['font.sans-serif']=['SimHei'] #解决中文乱码
x=df['年份']
y1=df['京东']
y2=df['天猫']
y3=df['自营']
#图表标题
plt.title('2013-2019年线上图书销售情况')
plt.stackplot(x,  y1,y2,y3,colors=['r','g','b'])
#图例
plt.legend(['京东','天猫','自营'],loc='upper left')
plt.show()
```

输出



- **绘制热力图：** imshow()其实就是将数组的值以图片的形式展示出来,数组的值对应着不同的颜色深浅,而数值的横纵坐标就是数组的索引。
  - plt.imshow(X, cmap=None，norm=None, *，aspect=None，interpolation=None, alpha=None， vmin=None， vmax=None， origin=None，extent=None, interpolation_stage=None， filternorm=True， filterrad=4.0， resample=None, url=None， data=None， **kwargs,)
    - X：数据，以纵横坐标为索引的二维数组
    - cmap：colormap，渲染图谱，import matplotlib.cm as cm ； dir(cm)
    - interpolation：渲染方法，'antialiased', 'nearest', 'bilinear'， 'bicubic', 'spline16','spline36', 'hanning', 'hamming', 'hermite'， 'kaiser', 'quadric', 'catrom','gaussian', 'bessel', 'mitchell'， 'sinc', 'lanczos', 'blackman'

3.常用图表绘制

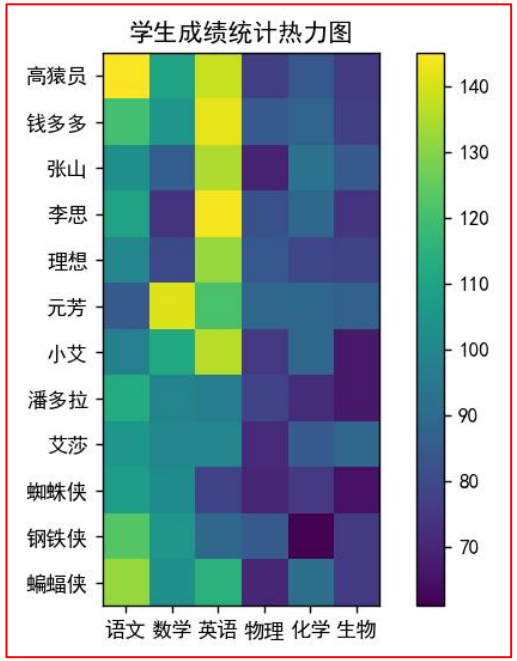
代码

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_excel('data1.xls',sheet_name='高二一班')
plt.rcParams['font.sans-serif']=['SimHei']
X = df.loc[:,"语文":"生物"].values
name=df['姓名']
plt.imshow(X)
#plt.imshow(X,cmap="RdBu")
plt.xticks(range(0,6,1),['语文','数学','英语','物理',
                         '化学','生物'])#设置x轴刻度标签
plt.yticks(range(0,12,1),name)#设置y轴刻度标签
plt.colorbar()   #显示颜色条
plt.title('学生成绩统计热力图')
plt.show()
```

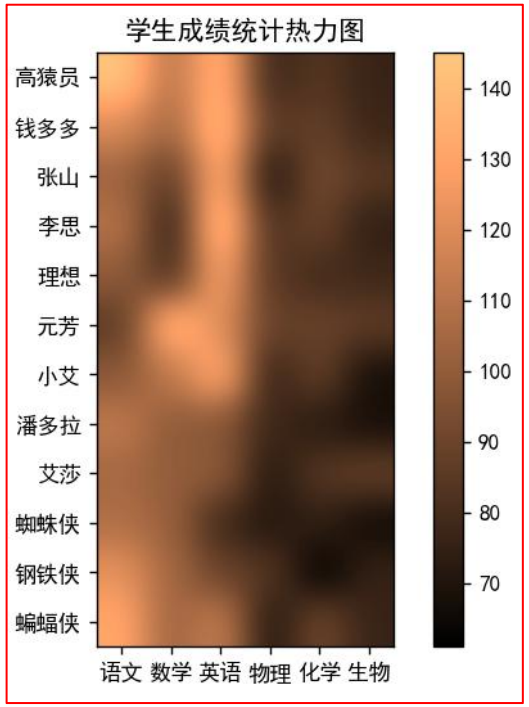
数据

	A	B	C	D	E	F	G
1	姓名	语文	数学	英语	物理	化学	生物
2	高猿员	145	110	138	77	84	76
3	钱多多	120	105	142	85	88	77
4	张山	103	86	135	69	93	85
5	李思	110	74	144	82	89	74
6	理想	100	80	132	84	79	78
7	元芳	85	141	121	89	89	87
8	小艾	98	112	136	75	89	67
9	潘多拉	113	99	97	78	72	67
10	艾莎	105	100	100	71	85	89
11	蜘蛛侠	108	102	78	70	75	65
12	钢铁侠	123	105	89	85	61	76
13	蝙蝠侠	132	103	115	70	92	76

输出



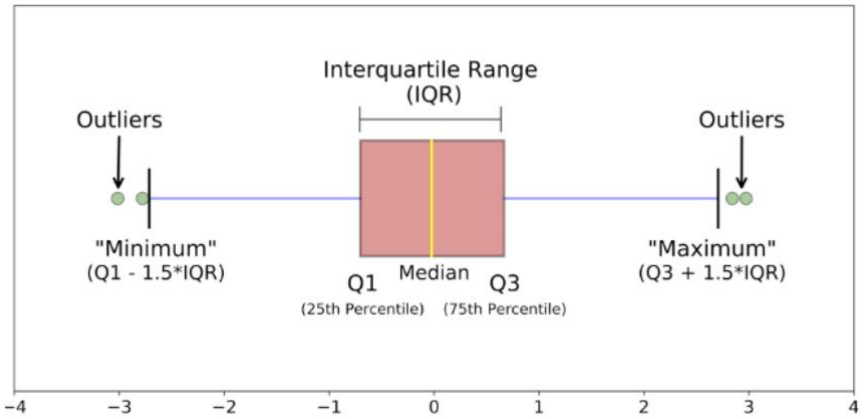

```
plt.imshow(X,cmap="copper",interpolation='gaussian')
```



3.常用图表绘制

绘制箱型图

- plt.boxplot(x, notch=None, sym=None, vert=None, whis=None, positions=None, widths=None, patch_artist=None, bootstrap=None, usermedians=None, conf_intervals=None, meanline=None, showmeans=None, showcaps=None, showbox=None, showfliers=None, boxprops=None, labels=None, flierprops=None, medianprops=None, meanprops=None, capprops=None, whiskerprops=None,
- x：指定要绘制箱形图的数据。
- notch：是否以凹口的形式展现箱形图，默认非凹口。
- sym：指定异常点的形状，默认为（+）号显示。
- vert：是否需要将箱形图垂直摆放，默认垂直摆放。
- whis：指定上下限与上下四分位的距离，默认为1.5倍的四分位差
- positions：指定箱形图的位置，默认为,1,2…]。
- widths：指定箱形图的宽度，默认为0.5。
- patch _ artist：是否填充箱体的颜色。
- meanline：是否用线的形式表示均值，默认用点来表示。
- showmeans：是否显示均值，默认不显示。
- showcaps：是否显示箱形图顶端和末端的两条线，默认显示。
- showbox：是否显示箱形图的箱体，默认显示。
- showfliers：是否显示异常值，默认显示。
- boxprops：设置箱体的属性，如边框色、填充色等。
- labels：为箱形图添加标签，类似于图例的作用。
- filerprops：设置异常值的属性，如异常点的形状、大小、填充色等。
  medianprops：设置中位数的属性，如线的类型、粗细等。
- meanprops：设置均值的属性，如点的大小、颜色等。
- capprops：设置箱形图顶端和末端线条的属性，如颜色、粗细等



箱线图是一种基于五位数摘要（“下限”，第一四分位数（Q1），中位数，第三四分位数（Q3）和“上限”）显示数据分布的标准化方法。

1.中位数（Q2 / 50百分位数）：数据集的中间值；

2.第一个四分位数（Q1 / 25百分位数）

3.第三四分位数（Q3 / 75百分位数）

4.四分位间距（IQR）：第25至第75个百分点的距离；

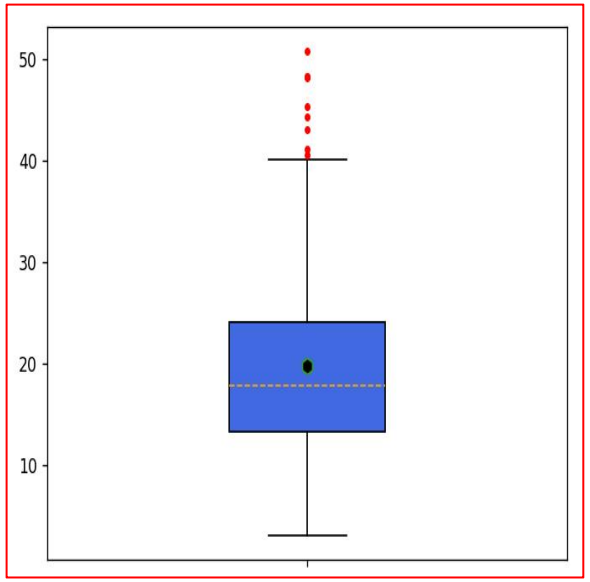
5.晶须（蓝色显示）

6.离群值（显示为绿色圆圈）

3.常用图表绘制

```
import matplotlib.pyplot as plt
import pandas as pd
df=pd.read_excel('tips.xlsx')
plt.boxplot(x = df['总消费'], # 指定绘制箱线图的数据
            whis = 1.5, # 指定1.5倍的四分位差
            widths = 0.3, #指定箱线图中箱子的宽度为0.3
            patch_artist = True, #填充箱子颜色
            showmeans = True, #显示均值
            boxprops = {'facecolor':'RoyalBlue'}, # 指定箱子的填充色为宝蓝色
            flierprops = {'markerfacecolor':'red', 'markeredgecolor':'red', 'markersize':3}, # 指定异常值的填充色、边框色和大小
            meanprops = {'marker':'h','markerfacecolor':'black', 'markersize':8},# 指定均值点的标记符号（六边形）、填充色和大小
            medianprops = {'linestyle':'--','color':'orange'}, # 指定中位数的标记符号（虚线）和颜色
            labels = ['']) # 去除x轴刻度值
plt.show()
# 计算下四分位数和上四分位
Q1 = df['总消费'].quantile(q = 0.25)
Q3 = df['总消费'].quantile(q = 0.75)
# 基于1.5倍的四分位差计算上下限对应的值
low_limit = Q1 - 1.5*(Q3 - Q1)
up_limit = Q3 + 1.5*(Q3 - Q1)
# 查找异常值
val=df['总消费'][(df['总消费'] > up_limit) | (df['总消费'] < low_limit)]
print('异常值如下：')
print(val)
```

	A	B	C	D	
1	总消费	小费	性别	是否吸烟	
2	16.99	1.01	女性	No	Su
3	24.59	3.61	女性	No	Su
4	35.26	5	女性	No	Su
5	14.83	3.02	女性	No	Su
6	10.33	1.67	女性	No	Su
7	16.97	3.5	女性	No	Su
8	20.29	2.75	女性	No	Sa
9	15.77	2.23	女性	No	Sa
10	19.65	3	女性	No	Sa
11	15.06	3	女性	No	Sa
12	20.69	2.45	女性	No	Sa
13	16.93	3.07	女性	No	Sa
14	10.29	2.6	女性	No	Su
15	34.81	5.2	女性	No	Su

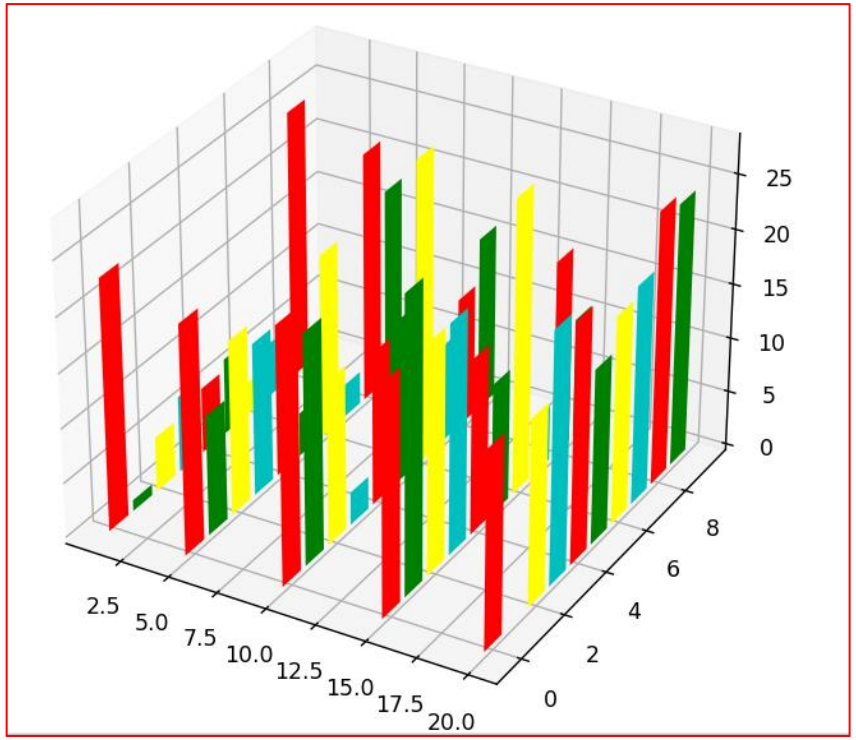


- 3D图
  - 3D图需要有mpl_toolkits 工具包
- 3D柱状图 实例

代码

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.axes3d import Axes3D
import numpy as np
fig = plt.figure()
#axes3d = Axes3D(fig)
axes3d = Axes3D(fig,auto_add_to_figure=False)
fig.add_axes(axes3d) #以上两行解决了版本不兼容问题
zs = [1, 5, 10, 15, 20]
for z in zs:
    x = np.arange(0, 10)
    y = np.random.randint(0, 30, size=10)
    axes3d.bar(x, y, zs=z, zdir='x', color=['r', 'green', 'yellow', 'c'])
plt.show()
```

输出

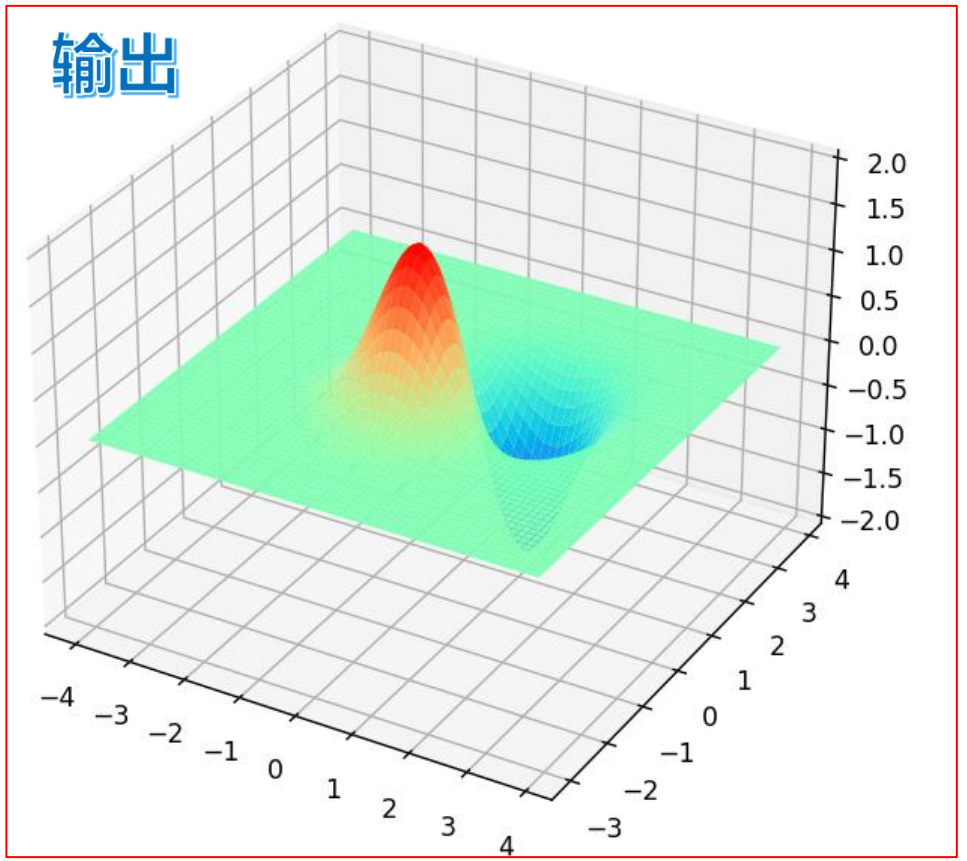


- 绘制3D曲面图实例

代码

```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
#ax = Axes3D(fig)
ax = Axes3D(fig,auto_add_to_figure=False)
fig.add_axes(ax) #以上两行解决了版本不兼容问题
delta = 0.125
# 生成代表X轴数据的列表
x = np.arange(-4.0, 4.0, delta)
# 生成代表Y轴数据的列表
y = np.arange(-3.0, 4.0, delta)
# 对x、y数据执行网格化
X, Y = np.meshgrid(x, y)
Z1 = np.exp(-X**2 - Y**2)
Z2 = np.exp(-(X - 1)**2 - (Y - 1)**2)
# 计算Z轴数据（高度数据）
Z = (Z1 - Z2) * 2
# 绘制3D图形
ax.plot_surface(X, Y, Z,
    rstride=1,  # rstride（row）指定行的跨度
    cstride=1,  # cstride(column)指定列的跨度
    cmap=plt.get_cmap('rainbow'))  # 设置颜色映射
# 设置Z轴范围
ax.set_zlim(-2, 2)
plt.show()
```

输出

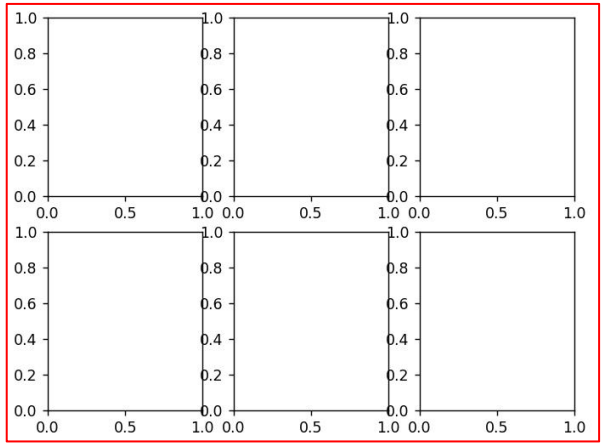


- 绘制多个子图表 subplot ， subplots， add_subplot
  - plt.subplot(*args, **kwargs)
  - subplot将画布分成n*m个区域，一次只画一个子图

代码

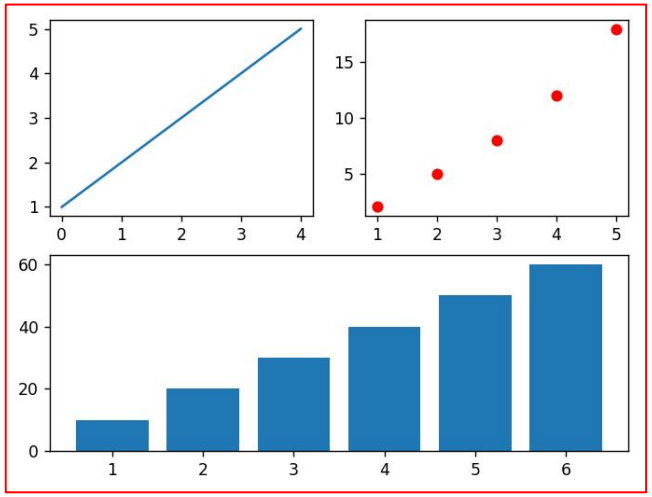
```
import matplotlib.pyplot as plt
plt.subplot(2,3,1)
plt.subplot(2,3,2)
plt.subplot(2,3,3)
plt.subplot(2,3,4)
plt.subplot(2,3,5)
plt.subplot(2,3,6)
plt.show()
```

输出



每画一个子图，都需要调用一次subplot

```
import matplotlib.pyplot as plt
#第1个子图表-折线图
plt.subplot(2,2,1)
plt.plot([1, 2, 3, 4,5])
#第2个子图表-散点图
plt.subplot(2,2,2)
plt.plot([1, 2, 3, 4,5], [2, 5, 8, 12,18], 'ro')
#第3个子图表-柱形图
plt.subplot(2,1,2)
x=[1,2,3,4,5,6]
height=[10,20,30,40,50,60]
plt.bar(x,height)
plt.show()
```



3.常用图表绘制

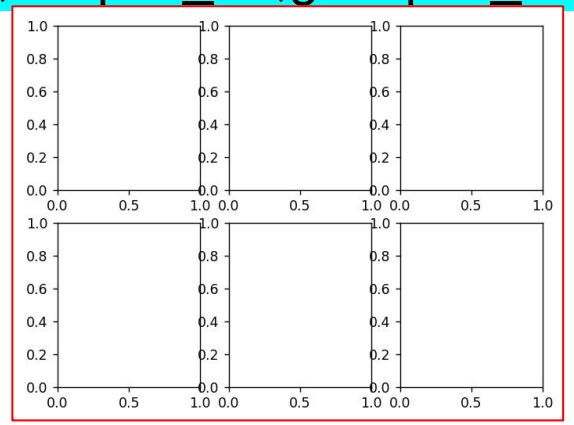
- plt.subplots（nrows,ncols,sharex,sharey,squeeze,subplot_kw,gridspec_kw,**fig_kw

代码

```
import matplotlib.pyplot as plt
figure,axes=plt.subplots(2,3)
plt.show()
```

2*3子图

输出



figure：绘制图表的画布
axes：坐标对象

```
import matplotlib.pyplot as plt
figure,axes=plt.subplots(2,2)
axes[0,0].plot([1, 2, 3, 4,5])   #折线图
axes[0,1].plot([1, 2, 3, 4,5], [2, 5, 8, 12,18], 'ro')#散点图
#柱形图
x=[1,2,3,4,5,6]
height=[10,20,30,40,50,60]
axes[1,0].bar(x,height)
#饼形图
x = [2,5,12,70,2,9]
axes[1,1].pie(x,autopct='%1.1f%%')
plt.show()
```