



【matplotlib绘图库】

人口普查数据图形展示





人口普查数据图形展示

- 数据可视化工作可以帮助决策者发现数据中的趋势、关联和异常，并从数据特征中发现业务优化点和风险，推动业务增长和创新、规避风险。
- **数据可视化工程师**负责将数据转化为直观易懂的图表图形，要具备扎实的数据分析能力，能够理解和解释数据，选择适当的图表类型、颜色、布局等，使数据可视化作品具有吸引力和清晰度，方便用户从中获取数据特征和规律。
- 因此，在数据可视化时要客观、公正；要关注用户体验，设计互动性强、易用性好的数据可视化界面，使用户能够自主地进行数据探索和分析；并同时关注编码规范，在需要的地方添加注释等，增强代码的可读性。





学习目标

- 认识matplotlib, 可以在Python环境中安装matplotlib;
- 掌握matplotlib.pyplot模块常用的绘图函数, 可以绘制一些简单的图表;
- 掌握绘制子图的几种方式, 可以共享子图之间的坐标轴;
- 掌握图表常用辅助元素的设置;
- 掌握默认的图表样式和修改图表样式的方法





目录/CONTENTS



Matplotlib库概述



基础图形绘制



划分画布绘制子图



matplotlib概述



Matplotlib是一个用于绘制图表和可视化数据的Python库，是数据科学、机器学习和科学计算领域中最流行的绘图库之一，支持numpy、pandas等数据结构。它主要用于绘制2D图表，具有丰富的绘制图表、定制图表元素或样式的功能，可以用于生成各种静态、交互式 and 动画图表。matplotlib也可用于绘制一些3D图表。



matplotlib概述

matplotlib库中应用最广泛的是matplotlib.Pyplot模块，该模块中包含折线图、柱形图、饼图等各种绘图函数，使用这些绘图函数可以绘制简单的图表。

函数名	说明	函数名	说明	函数名	说明
Plot	折线图	Bar	柱形图	Barh	条形图
Pie	饼图	Scatter	散点图	Hist	直方图
Boxplot	箱线图	Stackplot	堆叠区域图	polar	雷达图
Clabel	等高线图	Specgram	光谱图	Imshow	栅格图
errorbar	误差棒图	stem	茎叶图	streamplot()	矢量场流线图



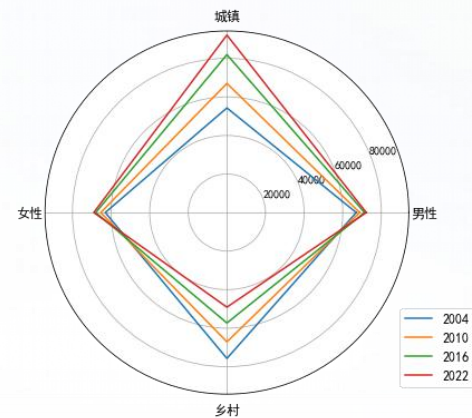
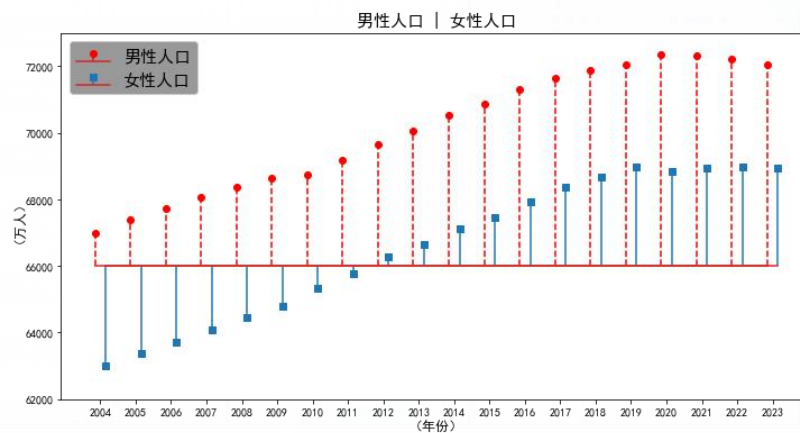
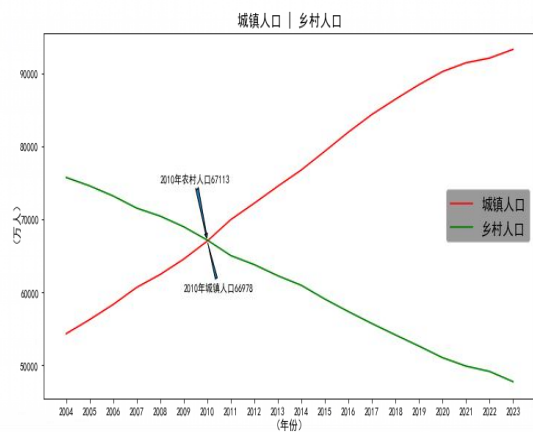
matplotlib.pyplot模块基础图形



- 折线图
- 散点图
- 柱状图
- 条形图
- 茎叶图
- 误差棒图
- 雷达图
- 饼图
- 箱线图
- 直方图



matplotlib.pyplot模块基础图形





环境搭建

安装和升级

安装

```
pip install matplotlib
```

```
conda install matplotlib
```

升级

```
pip install --upgrade matplotlib
```

```
conda update matplotlib
```





环境搭建

导入绘图库

```
from matplotlib import pyplot as plt升级  
from pylab import mpl  
mpl.rcParams['font.sans-serif'] = ['SimHei']  
mpl.rcParams['axes.unicode_minus'] = False
```





创建画布

```
figure(num = None, figsize=None, dpi=None, facecolor=None, edgecolor=None)
```

参数	说明
num	表示图形的编号或名称。数字为编号，字符串为名称。如果是False，则禁止画图形框。
figsize	用于设置画板的尺寸，指定figure的宽和高，类型为整数构成的元组，顺序为（width,height），单位为英寸。如果没有提供，默认为rc figure.figsize。
facecolor	用于设置画板的背景颜色。如果没有提供，默认是rc figure.facecolor
edgecolor	用于显示边框颜色。如果没有提供，默认是 rc figure.edgecolor
dpi	参数指定绘图对象的分辨率，即每英寸多少个像素，缺省值为80。如果没有提供，默认为rc figure.dpi
frameon	布尔型，可选，默认为True。表示是否显示边框。如果是False，则禁止画图形框。



编辑图表组件

添加图表标题

```
title(label, fontdict=None, loc='center', pad=None, **kwargs)
```

参数	说明
label	str, 设置标题文本字符串
fontdict	Dict, 设置文本的外观, 例如文本大小, 文本对齐方式等。默认值: fontdict = {'fontsize': rcParams['axes.titlesize'], 'fontweight': rcParams['axes.titleweight'], 'verticalalignment': 'baseline', 'horizontalalignment': loc}
loc	str, 设置标题的位置, 例如'center', 'left'和'right'
pad	float, 设置标题距轴顶部的偏移量(以磅为单位)。其默认值为“无”。
kwargs	设置其他关键字参数作为文本属性, 例如color, fonstyle, linespacing, backgroundcolor, rotation等



编辑图表组件

设置坐标轴

函数名	说明
xlabel (label,fontdict,labelpad,**kwargs) ylabel (label,fontdict,labelpad,**kwargs)	指定当前图形中X轴或Y轴名称，可以设置其位置、颜色、字体大小等
xlim (min,max) ylim (min,max)	指定当前图形中X轴或Y轴的刻度范围，确定一个数值范围
xticks (ticks,labels,**kwargs) yticks (ticks,labels,**kwargs)	指定当前图形中X轴或Y轴刻度的数量与取值。其中参数： ticks：表示刻度显示的位置列表； Labels：表示指定位置刻度的标签列表。



编辑图表组件

添加网格线

```
grid(b=None, which='major', axis='both', **kwargs)
```

参数	说明
b	是否显示网格线。布尔值或None，可选参数。如果没有关键字参数，则b为True，如果b为None且没有关键字参数，相当于切换网格线的可见性
which	网格线显示的尺度。字符串，可选参数，取值范围为{'major', 'minor', 'both'}，默认为'both'。'major'为主刻度、'minor'为次刻度
axis	选择网格线显示的轴。字符串，可选参数，取值范围为{'both', 'x', 'y'}，默认为'both'
**kwargs	Line2D线条对象属性。比如颜色、宽度等



编辑图表组件

添加参考线

```
axhline(loc=0, min=0, max=1, linestyle='-', **kwargs)
```

```
axvline(loc=0, min=0, max=1, linestyle='-', **kwargs)
```

参数名	说明
loc	表示水平参考线的y坐标位置或垂直参考线的x坐标位置
min	表示参考线的起始位置，默认为0
max	表示参考线的终止位置，默认为1
linestyle	表示参考线的类型，默认为实线



编辑图表组件

图形展示

```
matplotlib.pyplot.show(*, block=None)
```

图形保存

```
matplotlib.pyplot.savefig(url)
```



目录/CONTENTS



Matplotlib库概述



基础图形绘制



划分画布绘制子图



目录

时间：最近10年					
时间	年末总人口(万人)	男性人口(万人)	女性人口(万人)	城镇人口(万人)	乡村人口(万人)
2023年					
2022年	141175	72206	68969	92071	49104
2021年	141260	72311	68949	91425	49835
2020年	141212	72357	68855	90220	50992
2019年	141008	72039	68969	88426	52582
2018年	140541	71864	68677	86433	54108
2017年	140011	71650	68361	84343	55668
2016年	139232	71307	67925	81924	57308
2015年	138326	70857	67469	79302	59024
2014年	137646	70522	67124	76738	60908

注：1981年及以前人口数据为户籍统计数；1982、1990、2000、2010、2020年数据为当年人口普查数据推算数；其余年份数据为年度人口抽样调查推算数据。总人口和按性别分人口中包括现役军人，按城乡分人口中现役军人计入城镇人口。



人口增长趋势图形展示



人口年龄结构变化图形展示

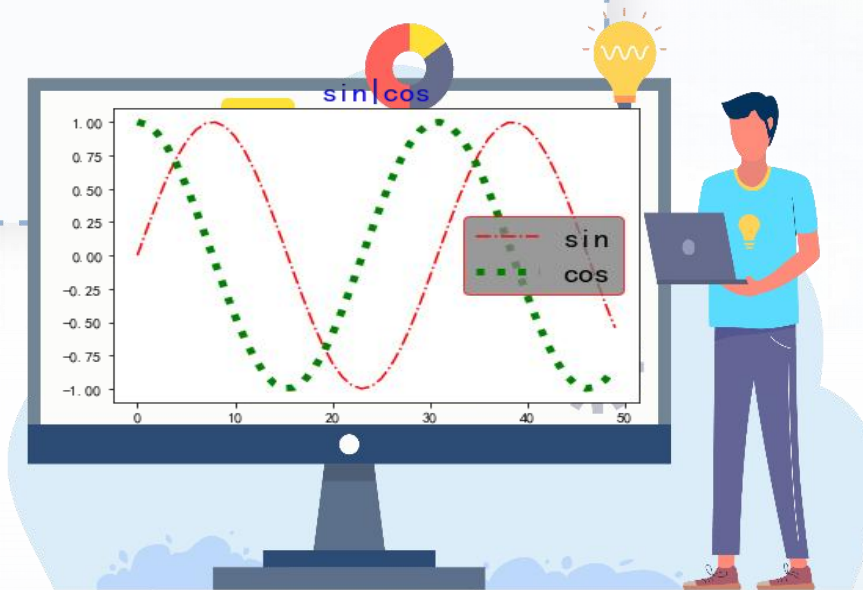


受教育程度数据图形展示



绘制折线图

- 折线图 (Line Chart) 是一种将数据点按照顺序连接起来的图形，可以看作是将散点图按照x轴坐标顺序连接起来的图形。
- 折线图一般以时间序列为轴，适用于展示固定时间间隔的数据变化趋势。





绘制折线图

>>> 基础语法

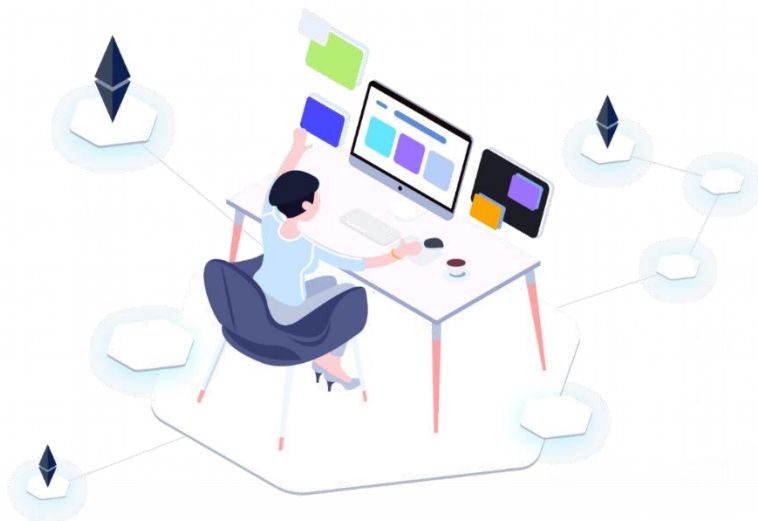
plot (x, y, fmt, scalex=True, scaley=True, data=None, label=None)

参数	说明
X	接收array。表示x轴对应的数据。默认值为range(len(y))。
Y	接收array。表示y轴对应的数据。无默认值
fmt	表示快速设置线条样式的格式字符串。
label	表示应用于图例的标签文本。
scalex, scaley	接收bool。表示这些参数确定视图限制是否适合于数据限制。默认为True
data	接收可索引对象。表示具有标签数据的对象。默认为None
color	接收特定str。表示指定线条的颜色。默认为None 颜色取值可以使用单词或其缩写，如blue或b
linewidth	接收int或float，指定线条宽度
linestyle	接收特定str。表示指定线条类型。默认为“-”。
marker	接收特定str。表示绘制的点的类型。默认为None
marker size	接收int，指定标记点大小
alpha	接收float。表示点的透明度。默认为None



绘制折线图

>>> 人口抽样调查数据



时间：最近10年					
时间	年末总人口(万人)	男性人口(万人)	女性人口(万人)	城镇人口(万人)	乡村人口(万人)
2023年	140967	72032	68935	93267	47700
2022年	141175	72206	68969	92071	49104
2021年	141260	72311	68949	91425	49835
2020年	141212	72357	68855	90220	50992
2019年	141008	72039	68969	88426	52582
2018年	140541	71864	68677	86433	54108
2017年	140011	71650	68361	84343	55668
2016年	139232	71307	67925	81924	57308
2015年	138326	70857	67469	79302	59024
2014年	137646	70522	67124	76738	60908

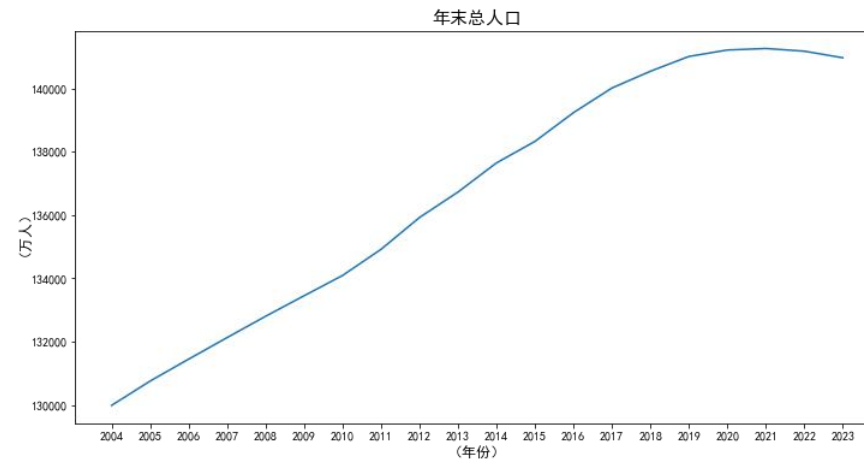
注：1981年及以前人口数据为户籍统计数；1982、1990、2000、2010、2020年数据为当年人口普查数据推算数；其余年份数据为年度人口抽样调查推算数据。总人口和按性别分人口中包括现役军人；按城乡分人口中现役军人计入城镇人口。



绘制折线图

>>> 绘制折线图展示近年来人口增长趋势

```
fig=plt.figure(figsize=(12,6))  
plt.plot(df['年末总人口(万人)'])  
plt.xticks(df.index)  
plt.xlabel(' (年份) ',fontsize=12)  
plt.ylabel(' (万人) ',fontsize=12)  
plt.title('年末总人口',fontsize=15)  
plt.show()
```

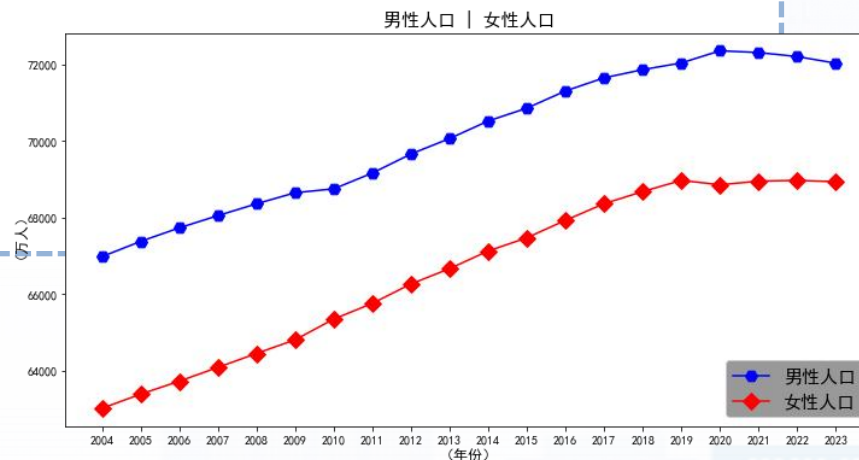




绘制折线图

>>> 绘制折线图比较“男性人口”和“女性人口”数据

```
fig=plt.figure(figsize=(12,6))
plt.plot(df['男性人口(万人)'],color='b',marker='h',markersize=10)
plt.plot(df['女性人口(万人)'],color='r',marker='D',markersize=10)
plt.ylabel(' (万人) ',fontsize=12)
plt.legend(['男性人口','女性人口'],loc=4,fontsize=15,facecolor='gray')
plt.xticks(df.index)
plt.xlabel(' (年份) ',fontsize=12)
plt.ylabel(' (万人) ',fontsize=12)
plt.title('男性人口 | 女性人口',fontsize=15)
plt.show()
```

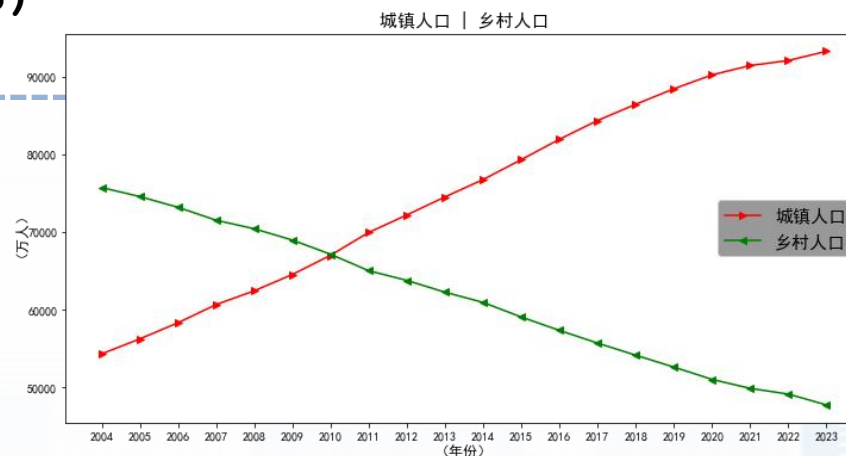




绘制折线图

>>> 绘制折线图比较“城镇人口”和“乡村人口”数据变化

```
fig=plt.figure(figsize=(12,6))
plt.plot(df['城镇人口(万人)'],color='r',marker='>',linestyle='None')
plt.plot(df['乡村人口(万人)'],color='g',marker='<',linestyle='None')
plt.xticks(df.index)
plt.xlabel(' (年份) ',fontsize=12)
plt.ylabel(' (万人) ',fontsize=12)
plt.legend(['城镇人口','乡村人口'],loc=5,fontsize=15,facecolor='gray')
plt.title('城镇人口 | 乡村人口',fontsize=15)
plt.show()
```

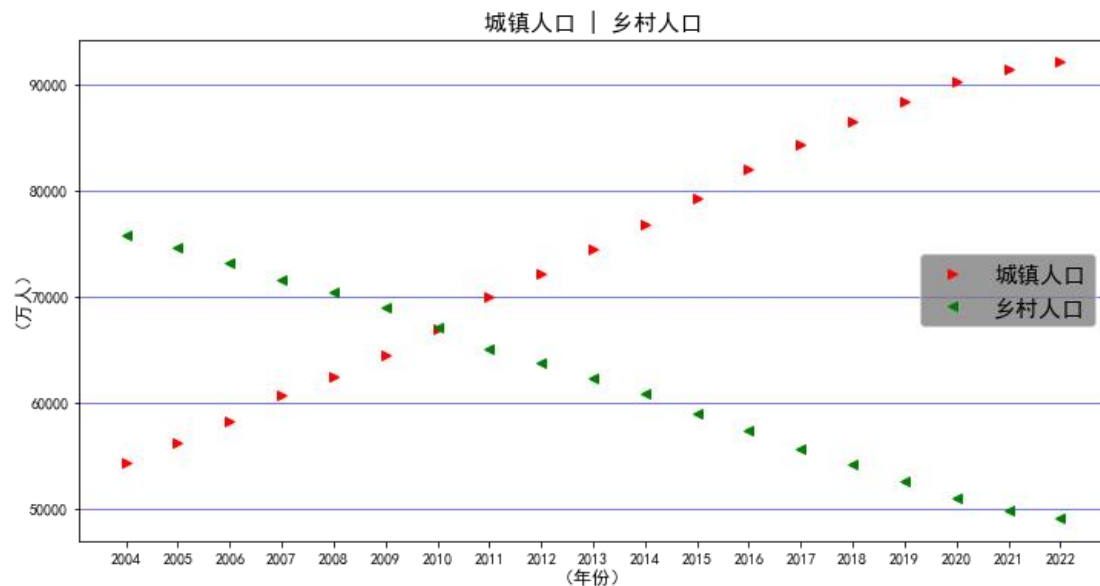




绘制折线图

>>> 添加水平网格线

```
plt.grid(b=True,axis='y',linewidth=0.5,color='b')
```

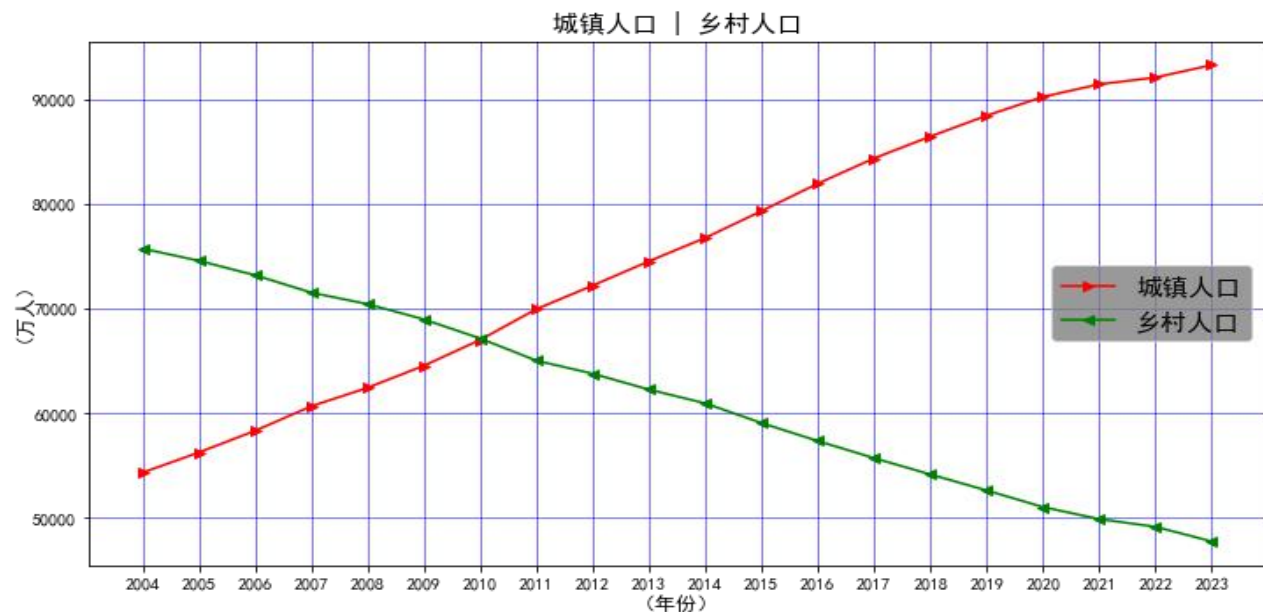




绘制折线图

>>> 同时增加水平网格线和垂直网格线

```
plt.grid(b=True,axis='both',linewidth=0.5,color='b')
```



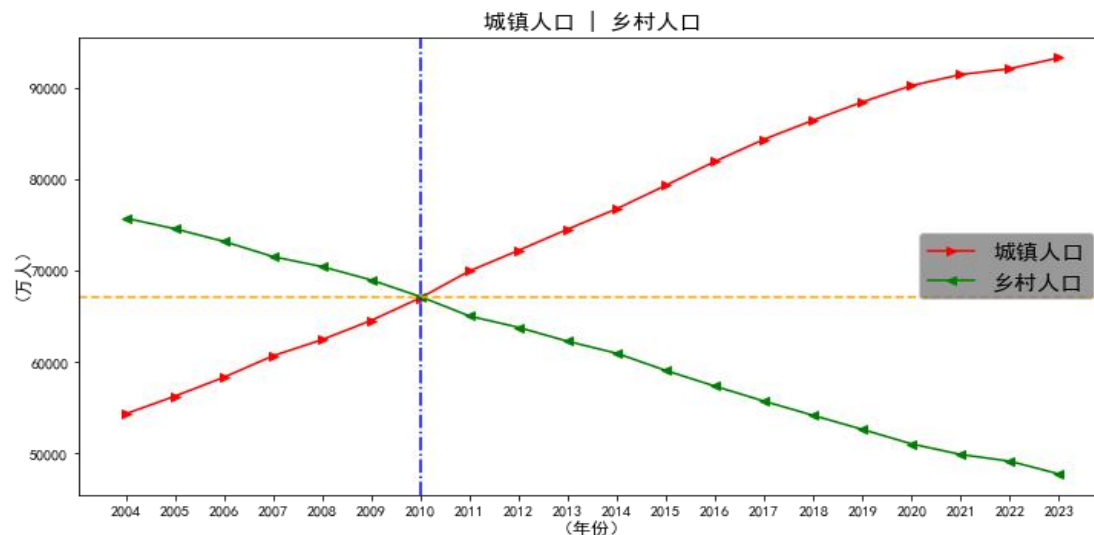


绘制折线图

添加参考线，便于数据之间的比较

```
plt.axhline(y=67000,linestyle='--',color='orange')
```

```
plt.axvline(x=2010,linestyle='-.',color='orange')
```





绘制折线图

>>> 为图表添加指向型注释文本

matplotlib.pyplot.annotate(s,xy, *args,kwargs)**

参数	说明
s	接收str，表示注释文本的内容。
xy	接收元组 (x, y)，表示被注释的点的坐标位置。
xytext	接收元组 (x, y)，表示注释文本所在的坐标位置。
arrowprops	接收dict，表示指示箭头的属性字典。包括width、headwidth、headlength、shrink、arrowstyle等。 arrowstyle表示箭头类型，可取值：-、->、<->、<-、 - 、- 、- >、< -、< - >、fancy、simple、wedge
bbox	接收dict，表示注释文本的边框属性字典。



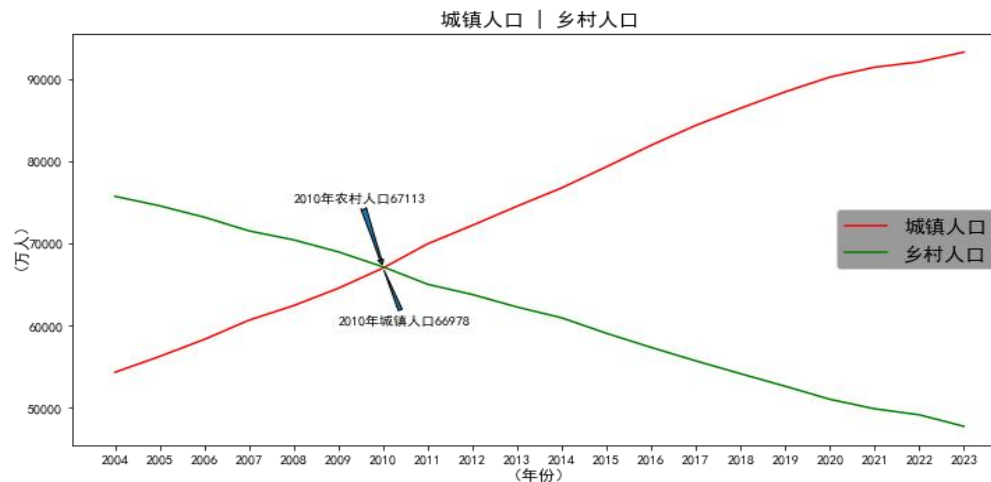


绘制折线图

>>> 给“城镇人口|农村人口”折线图添加注释文本

```
plt.annotate('2010年城镇人口66978', (2010,66978),(2009,60000),  
            arrowprops=dict(arrowstyle="wedge"))
```

```
plt.annotate('2010年农村人口67113', (2010,67113),(2008,75000),  
            arrowprops=dict(arrowstyle="fancy"))
```





绘制散点图

- 散点图 (scatter diagram) 又称为散点分布图, 由若干数据点组成。主要用于判断两变量之间是否存在某种关联, 或者总结数据点的分布模式。
- 每个数据点有位置特征: 横坐标和纵坐标。利用坐标点 (散点) 的分布形态反映特征间的统计关系。





绘制散点图

>>> 基础语法

```
scatter( x, y, s=None, c=None, marker=None, cmap=None, norm=None,
        vmin=None, vmax=None, alpha=None, linewidths=None,
        verts=None, edgecolors=None, data=None )
```



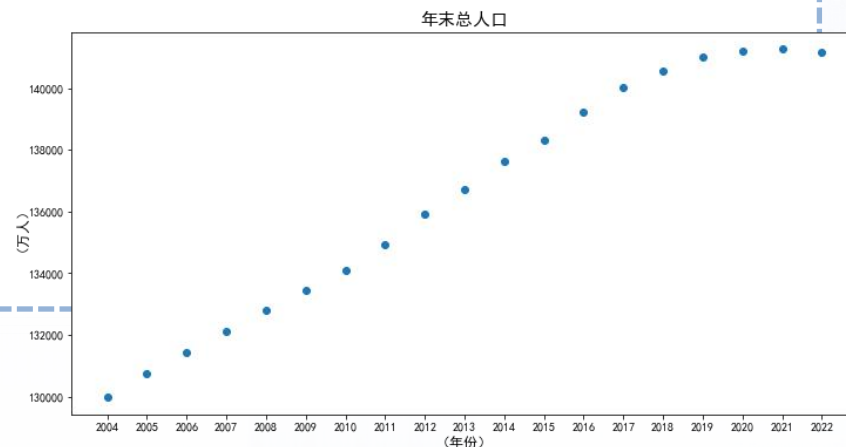
参数名	说明
x, y	表示数据点的位置 (x,y)。必备参数。x, y都不能为空。x, y支持多维数组，但是x, y的维度必须一致，多维数组会被展平 (flatten) 为一维数组。
s	表示数据点标记的大小。可选参数。默认值为rcParams['lines.markersize'] ** 2。
c	表示数据点的颜色。 可选参数。默认是蓝色' b'，表示的是标记的颜色，或者可以是一个表示颜色的字符，或者是一个长度为n的表示颜色的序列等等如 'b' =blue, 'y' =yellow, 'k' =black等。
marker	表示数据点标记的样式，默认为圆形' o'：rcParams["scatter.marker"] ('o')
alpha	表示标记的透明度。取值范围为[0, 1]，0为透明，1为不透明。默认值为None。
linewidths	表示数据点的描边（边缘线）宽度。浮点数或类数组结构。默认值为rcParams["lines.linewidth"] (1.5)。
edgecolors	表示数据点的描边（边缘线）颜色。取值为{'face', 'none', None}、颜色、颜色序列。默认值为rcParams["scatter.edgecolors"] ('face')。'face': 标记边缘颜色与标记颜色相同'none': 没有边线。
vmin, vmax	表示亮度的最小值和最大值。当norm存在的时候忽略。



绘制散点图

>>> 绘制散点图展示近20年人口增长趋势

```
df=pd.read_excel('人口数据.xlsx','总人口',index_col=0)
fig=plt.figure(figsize=(12,6))
plt.scatter(df.index,df['年末总人口(万人)'])
plt.xticks(df.index)
plt.xlabel(' (年份) ',fontsize=12)
plt.ylabel(' (万人) ',fontsize=12)
plt.title('年末总人口',fontsize=15)
plt.show()
```

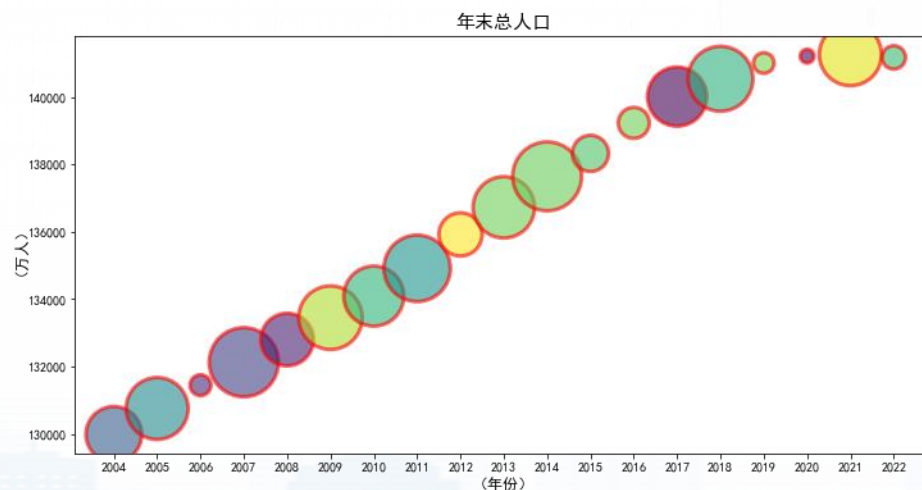




绘制散点图

>>> 绘制气泡图展示近年来人口增长趋势

```
plt.scatter(df.index, df['年末总人口(万人)'],  
            s=np.random.randint(50, 3000, 19),  
            c=np.random.rand(19), alpha=0.6,  
            linewidths=3, edgecolors='r')
```

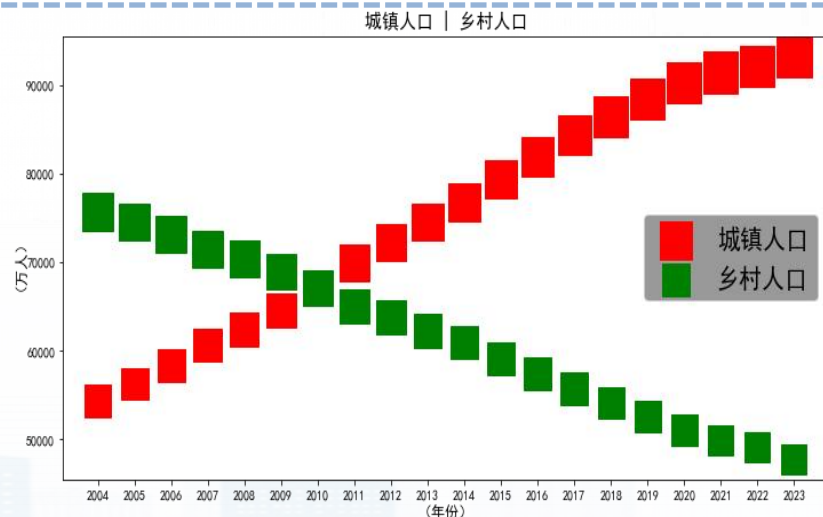




绘制散点图

>>> 绘制散点图比较“城镇人口”和“乡村人口”数据变化

```
plt.scatter(df.index, df['城镇人口(万人)'], color='r',  
            marker='s', s=df['城镇人口(万人)']/100 )  
  
plt.scatter(df.index, df['乡村人口(万人)'], color='g',  
            marker='s', s=df['乡村人口(万人)']/100 )
```

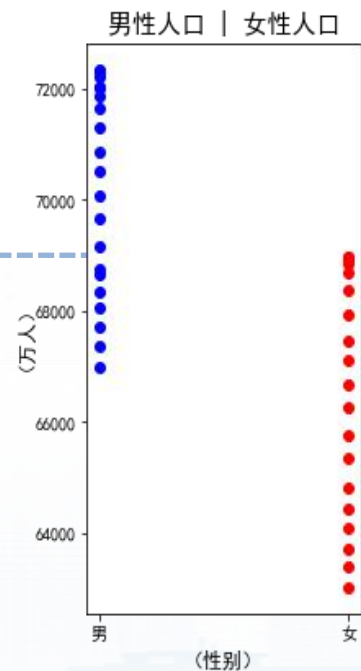




绘制散点图

>>> 绘制散点图比较“男性人口”和“女性人口”数据

```
df=pd.read_excel('人口数据.xlsx','总人口',index_col=0)
fig=plt.figure(figsize=(3,6))
plt.scatter(np.array(['男'] * 20),df['男性人口(万人)'],c='blue')
plt.scatter(np.array(['女'] * 20),df['女性人口(万人)'],c='red')
plt.xlabel(' (性别) ',fontsize=12)
plt.ylabel(' (万人) ',fontsize=12)
plt.title('男性人口 | 女性人口',fontsize=15)
plt.show()
```

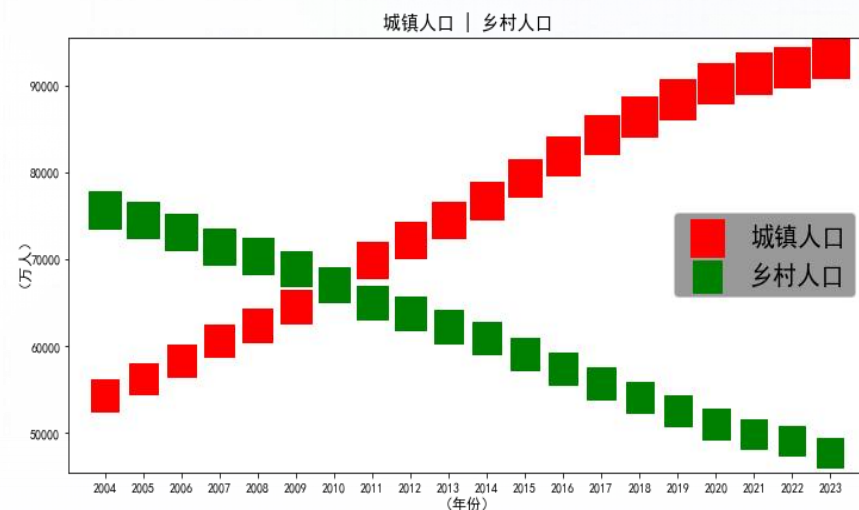


>>> 绘制散点图和折线图

比较“城镇人口”和“乡村人口”数据变化

```
plt.scatter(df.index, df['城镇人口(万人)'], color='r', marker='s', s=df['城镇人口(万人)']/100)
```

```
plt.scatter(df.index, df['乡村人口(万人)'], color='g', marker='s', s=df['乡村人口(万人)']/100)
```





绘制柱形图

- 柱形图（Bar Chart）是由一系列宽度相等的纵向矩形条（柱体）组成的图表。
- 柱形图一般以数据分类为轴，每类数据一个柱体；使用柱体的高度表示数值，以此反映不同分类数据之间的差异。





绘制柱形图

>>> 基础语法

bar(x, height, width=0.8, bottom=None, align=' center' , data=None, tick_label=None, xerr=None, yerr=None, **kwargs)

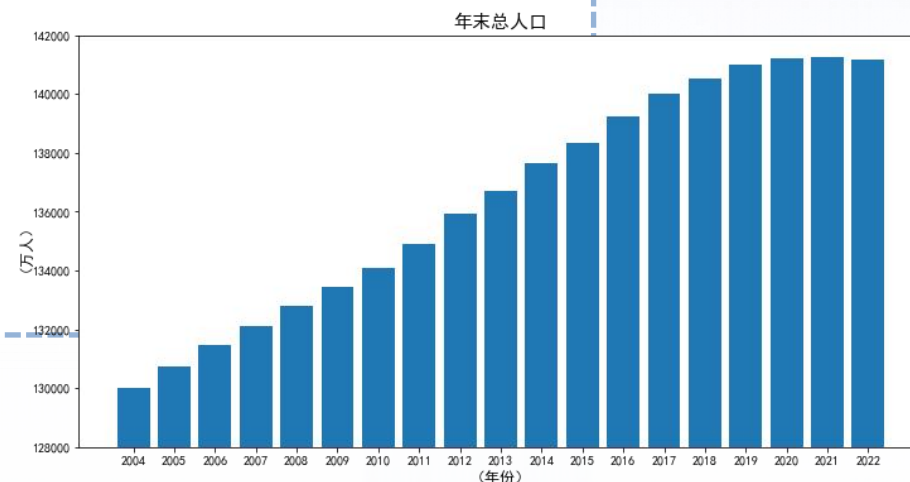
参数	说明
x	表示柱形的x轴坐标值。
height	表示柱形的高度。
width	表示柱形的宽度，默认值为0.8。
bottom	柱形左侧的x坐标值，默认值为0。
align	表示柱形的对齐方式，，可选“center”和“edge”。默认值为“center”，即柱形与刻度线居中对齐；“edge”表示将柱形的左端与刻度线对齐
tick_label	表示柱形对应的刻度标签。
xerr, yerr	若未设为None，则需要为柱形图添加水平/垂直误差棒。
color	接收特定str或包含颜色字符串的list。表示柱形图颜色。默认为None



绘制柱形图

>>> 绘制柱形图展示近年来人口增长趋势

```
df=pd.read_excel('人口数据.xlsx','总人口',index_col=0)
fig=plt.figure(figsize=(12,6))
plt.bar(df.index,df['年末总人口(万人)'])
plt.xticks(df.index)
plt.ylim([128000,142000])
plt.ylabel(' (万人) ',fontsize=12)
plt.xlabel(' (年份) ',fontsize=12)
plt.title('年末总人口',fontsize=15)
plt.show()
```

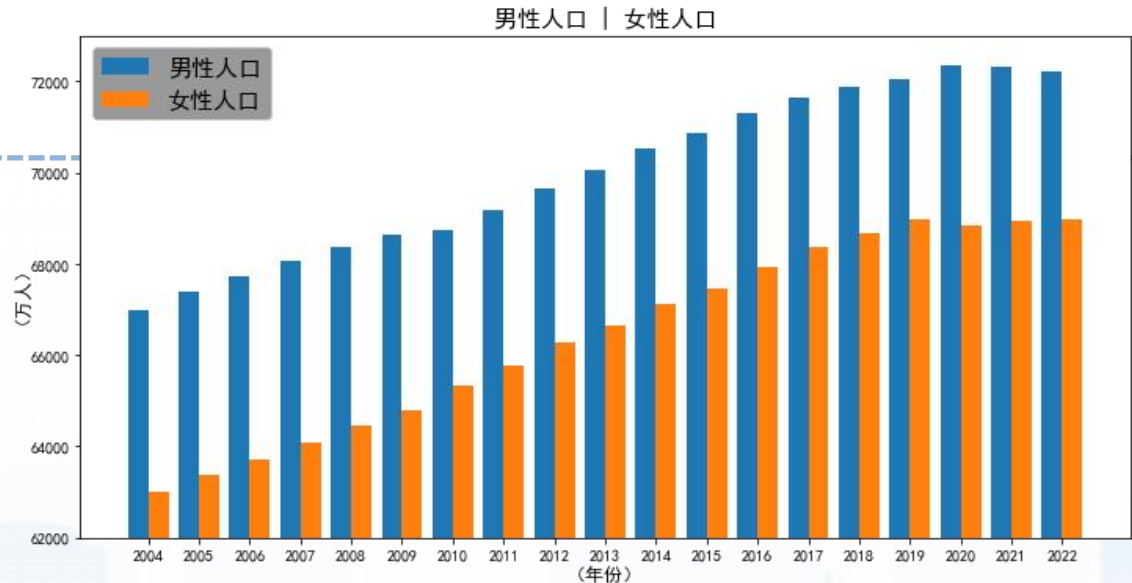




绘制柱形图

>>> 绘制柱形图比较“男性人口”和“女性人口”数据

```
plt.bar(df.index-0.2,df['男性人口(万人)'],width=0.4)
plt.bar(df.index+0.2,df['女性人口(万人)'],width=0.4)
plt.legend(['男性人口','女性人口'],loc=0,fontsize=15,facecolor='gray')
plt.title('男性人口 | 女性人口',fontsize=15)
plt.xticks(df.index)
plt.ylim([62000,73000])
```

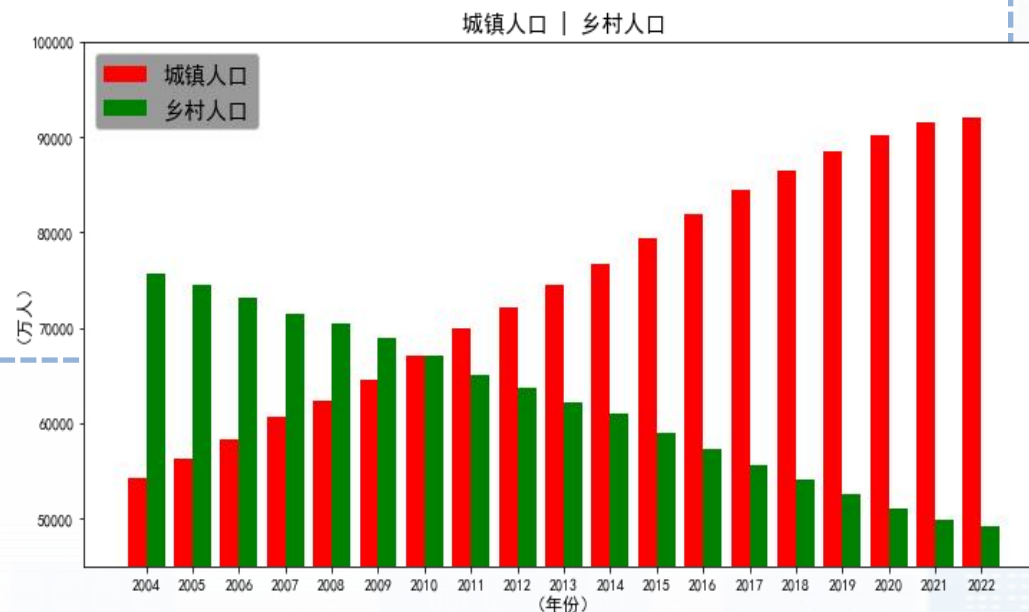




绘制柱形图

>>> 绘制柱形图比较“城镇人口”和“乡村人口”数据变化

```
plt.bar(df.index-0.2,df['城镇人口(万人)'],color='r',width=0.4)
plt.bar(df.index+0.2,df['乡村人口(万人)'],color='g',width=0.4)
plt.legend(['城镇人口','乡村人口'],loc=0,fontsize=15,facecolor='gray')
plt.title('城镇人口 | 乡村人口',fontsize=15)
plt.xticks(df.index)
plt.ylim([45000,100000])
plt.ylabel(' (万人) ',fontsize=12)
plt.xlabel(' (年份) ',fontsize=12)
plt.show()
```

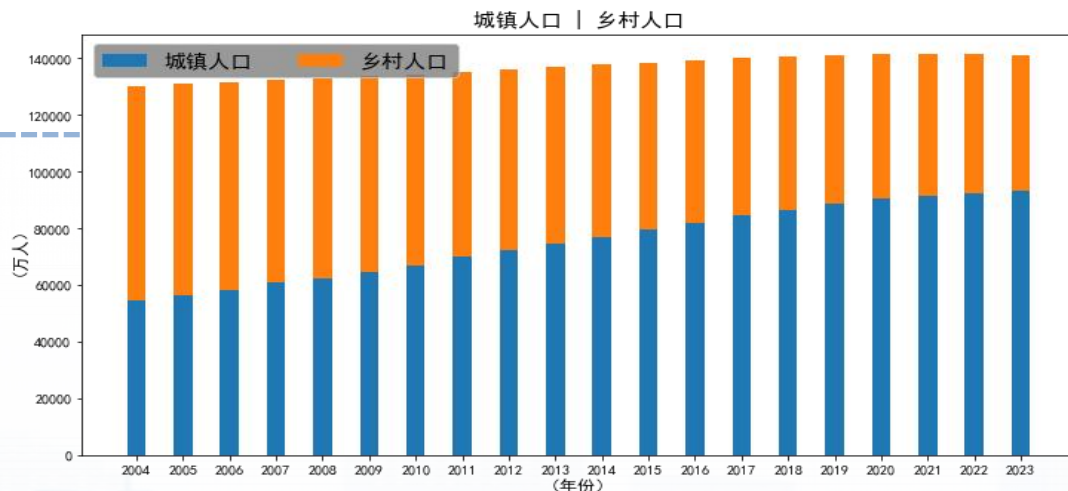




绘制柱形图

>>> “城镇人口” 和 “乡村人口” 柱形堆叠图

```
plt.bar(df.index, df['城镇人口 (万人)'], width=0.4)
plt.bar(df.index, df['乡村人口 (万人)'], width=0.4, bottom=df['城镇人口 (万人)'])
plt.legend(['城镇人口', '乡村人口'], loc='upper
left', fontsize=15, facecolor='gray', ncol=2)
```

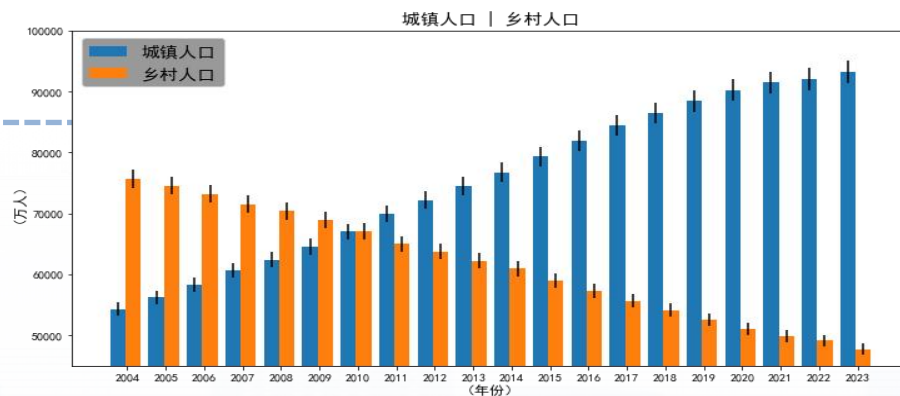




绘制柱形图

>>> 添加误差棒

```
plt.bar(df.index-0.25,df['城镇人口(万人)'],width=0.4,  
        yerr=df['城镇人口(万人)']*0.02)  
  
plt.bar(df.index+0.15,df['乡村人口(万人)'],width=0.4,  
        yerr=df['乡村人口(万人)']*0.02)
```





绘制条形图

- 条形图是横置的柱形图，由一系列高度相等、长短不一的横向矩形条组成。
- 条形图一般以数据分类为轴，每类数据一个矩形条；使用矩形条的长度表示数值，以此反映不同分类数据之间的差异。





绘制条形图

>>> 基础语法

```
barh( y, width, height=0.8, left=None, align='center' ,  
      tick_label=None,xerr=None,yerr=None, **kwargs )
```



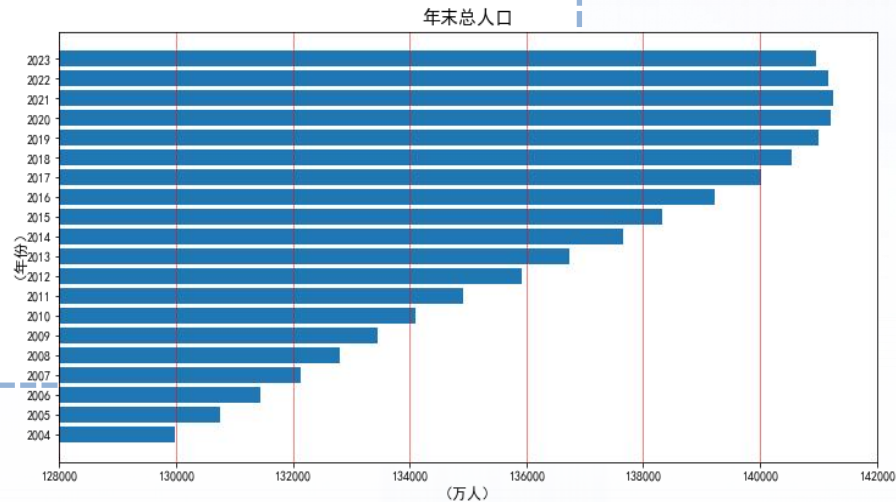
参数	说明
y	表示条形的y值。
width	表示条形的宽度。
height	表示条形的高度，默认值为0.8。
color	接收特定str或包含颜色字符串的list。表示柱形图颜色。默认为None
left	条形左侧的x坐标值，默认值为0。
align	表示条形的对齐方式，，可选“center”和“edge”。默认值为“center”，即条形与刻度线居中对齐；“edge”表示将条形的左端与刻度线对齐
tick_label	表示条形对应的刻度标签。
xerr, yerr	若未设为None，则需要为条形图添加水平/垂直误差棒。



绘制条形图

>>> 绘制条形图展示近20年人口增长趋势

```
df=pd.read_excel('人口数据.xlsx','总人口',index_col=0)
fig=plt.figure(figsize=(12,6))
plt.barh(df.index,df['年末总人口(万人)'])
plt.grid(b=True,axis='x',linewidth=0.5,color='r')
plt.title('年末总人口',fontsize=15)
plt.xlim([128000,142000])
plt.yticks(df.index)
plt.xlabel('(万人)',fontsize=12)
plt.ylabel('(年份)',fontsize=12)
plt.show()
```

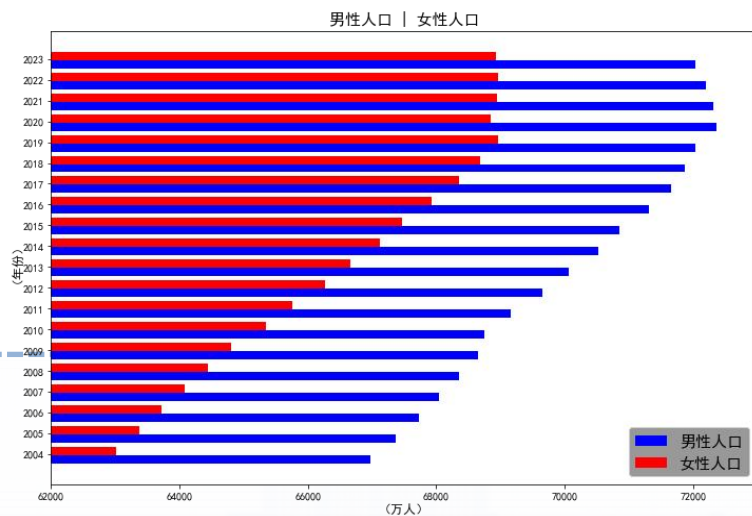




绘制条形图

>>> 绘制条形图比较“男性人口”和“女性人口”数据

```
plt.barh(df.index-0.2,df['男性人口(人)'],color='b',height=0.4)
plt.barh(df.index+0.2,df['女性人口(万人)'],color='r',height=0.4)
plt.legend(['男性人口','女性人口'], loc=0, fontsize=15, facecolor='gray')
plt.title('男性人口 | 女性人口',fontsize=15)
plt.yticks(df.index)
plt.xlim([62000,73000])
plt.xlabel(' (万人) ',fontsize=12)
plt.ylabel(' (年份) ',fontsize=12)
plt.show()
```

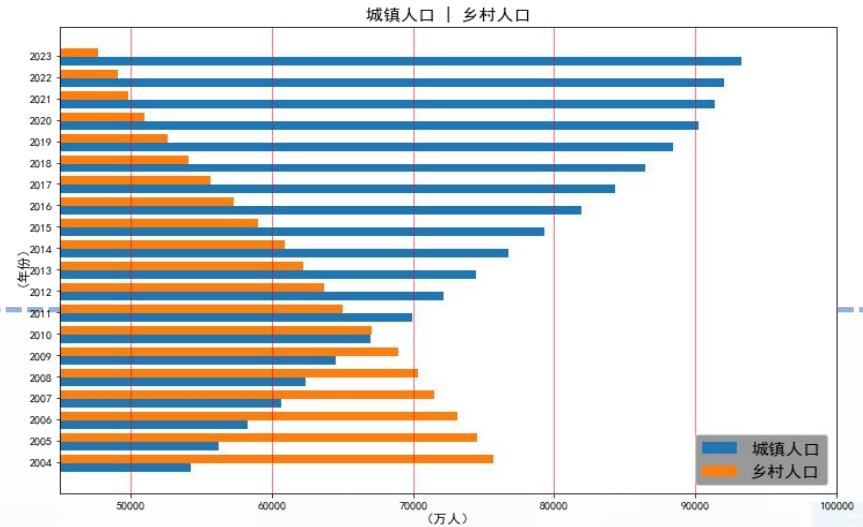




绘制条形图

>>> 绘制条形图比较“城镇人口”和“乡村人口”数据变化

```
plt.barh(df.index-0.2,df['城镇人口(万人)'],height=0.4)
plt.barh(df.index+0.2,df['乡村人口(万人)'],height=0.4)
plt.legend(['城镇人口','乡村人'], loc='best',fontsize=15,facecolor='gray')
plt.title('城镇人口 | 乡村人口',fontsize=15)
plt.grid(b=True,axis='x',linewidth=0.5,color='r')
plt.yticks(df.index)
plt.xlim([45000,100000])
plt.xlabel(' (万人) ',fontsize=12)
plt.ylabel(' (年份) ',fontsize=12)
plt.show()
```

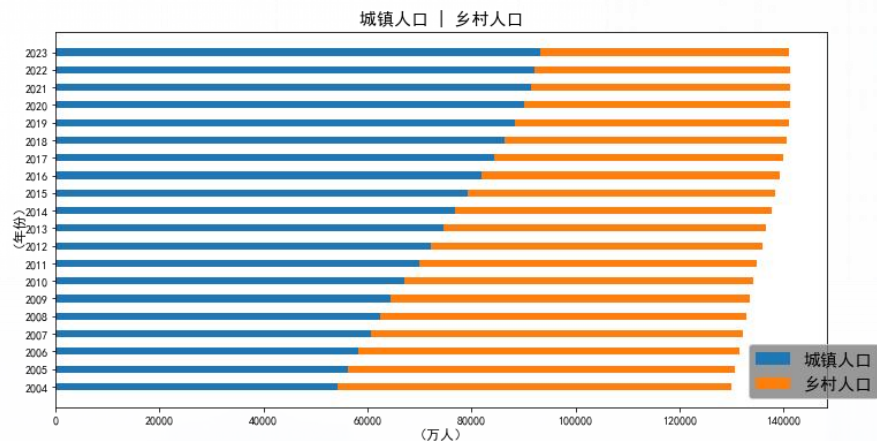




绘制条形图

>>> “城镇人口” 和 “乡村人口” 条形堆叠图

```
plt.barh(df.index, df['城镇人口 (万人)'], height=0.4)
plt.barh(df.index, df['乡村人口 (万人)'], height=0.4, left=df['城镇人口 (万人)'])
plt.legend(['城镇人口', '乡村人口'], loc='lower
right', fontsize=15, facecolor='gray', bbox_to_anchor=(1.08, 0))
```

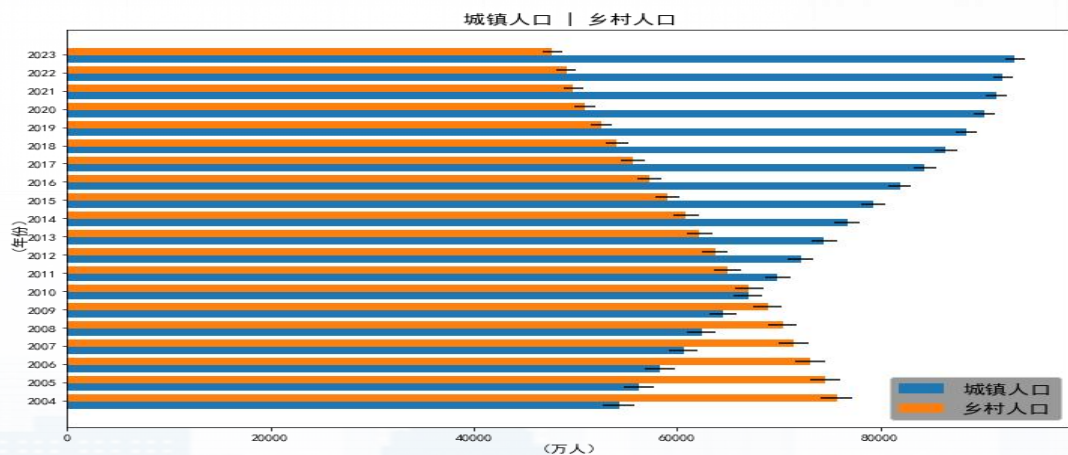




绘制条形图

>>> 添加误差棒

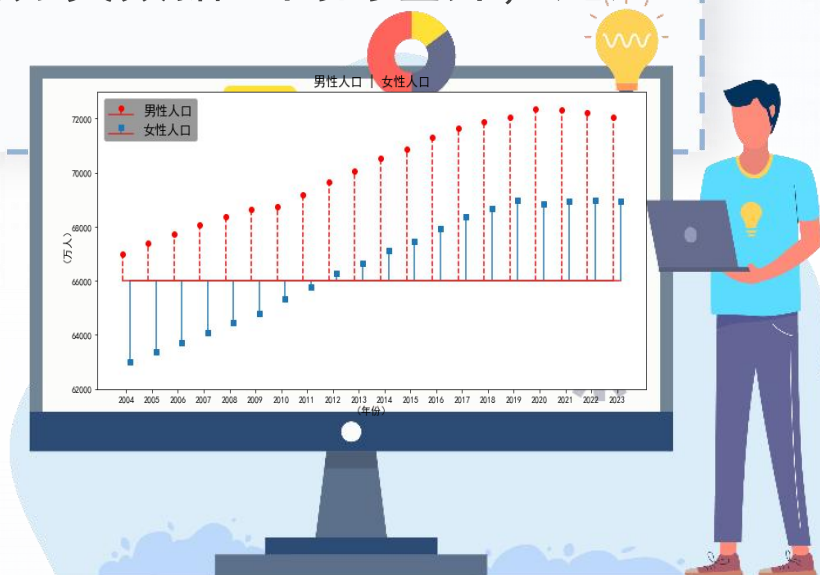
```
plt.barh(df.index-0.25, df['城镇人口(万人)'],  
         height=0.4, xerr=df['乡村人口(万人)']*0.02)  
  
plt.barh(df.index+0.15, df['乡村人口(万人)'],  
         height=0.4, xerr=df['乡村人口(万人)']*0.02)
```





绘制茎叶图

- 茎叶图从外观来看更像是火柴，由基线、茎线、茎头三部分构成，根据基线（baseline）上的位置（locs）绘制从基线到杆头（heads）的茎线，并在茎头（heads）处放置标记，又称为棉棒图、火柴杆图、大头针图等。
- 茎叶图是柱形图的变形。以茎头的位置反映不同分类数据之间的差异，适用于展示中小规模数据集各分类之间的比较。





绘制茎叶图

>>> 基础语法

**stem([loc], head , linefmt=None, markerfmt=None, basefmt=None,
bottom=0, label=None, use_line_collection=True,
orientation='vertical', data=None)**



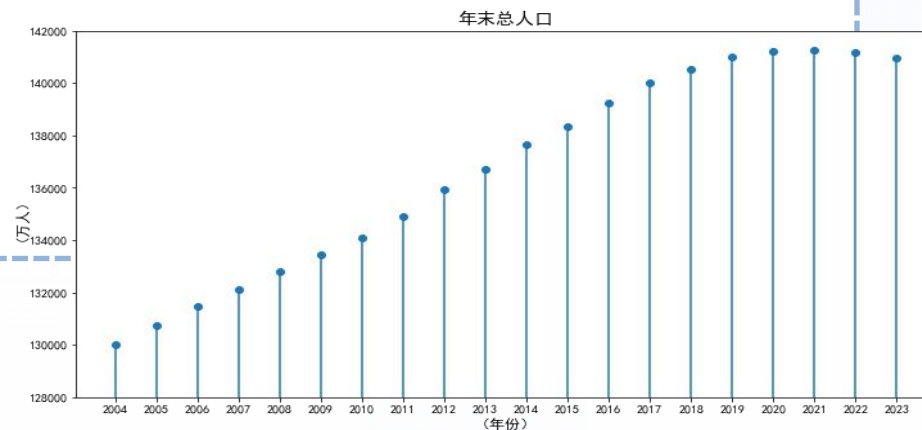
参数	说明
loc	接收array，表示茎线的位置，对于水平茎叶图，该参数即茎线的y轴位置。默认值为range(len(heads))。可选参数。
head	接收array，表示茎头的位置，对于垂直茎叶图，该参数即茎头的y值，对于水平茎叶图，该参数即茎头的x值。必备参数。
linefmt	接收str，表示定义茎线的颜色和线形，取值同plot函数。默认值为'C0-'。可选参数。
markerfmt	接收str，表示定义茎头标记的形状和颜色。默认值为'C0o'。可选参数。
basefmt	接收str，表示定义基线的格式字符串。默认值为'C3-'。 可选参数。
bottom	接收float，表示基线的位置。默认值为0。可选参数。
label	接收str，表示图例中使用的标签。默认值None。可选参数。
use_line_collection	接收bool，表示控制线条对象类型。默认值True。可选参数。



绘制茎叶图

>>> 绘制茎叶图展示近年来人口增长趋势

```
df=pd.read_excel('人口数据.xlsx','总人口',index_col=0)
fig=plt.figure(figsize=(12,6))
plt.stem(df.index,df['年末总人口(万人)'],use_line_collection=True)
plt.title('年末总人口',fontsize=15)
plt.ylim([128000,142000])
plt.xticks(df.index)
plt.ylabel(' (万人) ',fontsize=12)
plt.xlabel(' (年份) ',fontsize=12)
plt.show()
```

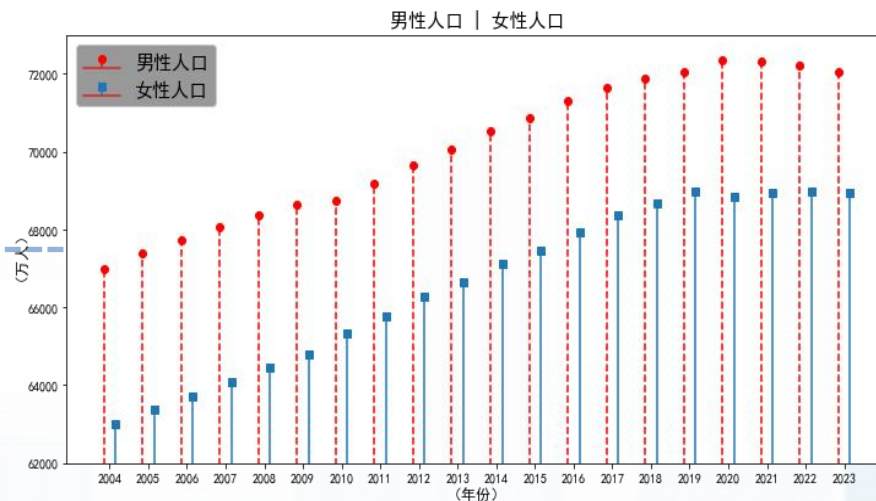




绘制茎叶图

绘制茎叶图比较“男性人口”和“女性人口”数据

```
plt.stem(df.index-0.15,df['男性人口(万人)'],  
         linefmt='r--',markerfmt='ro',use_line_collection=True)  
plt.stem(df.index+0.15,df['女性人口(万人)'],  
         markerfmt='s',use_line_collection=True)  
plt.legend(['男性人口','女性人口'],loc=0,fontsize=15,facecolor='gray')  
plt.title('男性人口 | 女性人口',fontsize=15)  
plt.ylabel('(万人)',fontsize=12)  
plt.xlabel('(年份)',fontsize=12)  
plt.xticks(df.index)  
plt.ylim([62000,73000])  
plt.show()
```

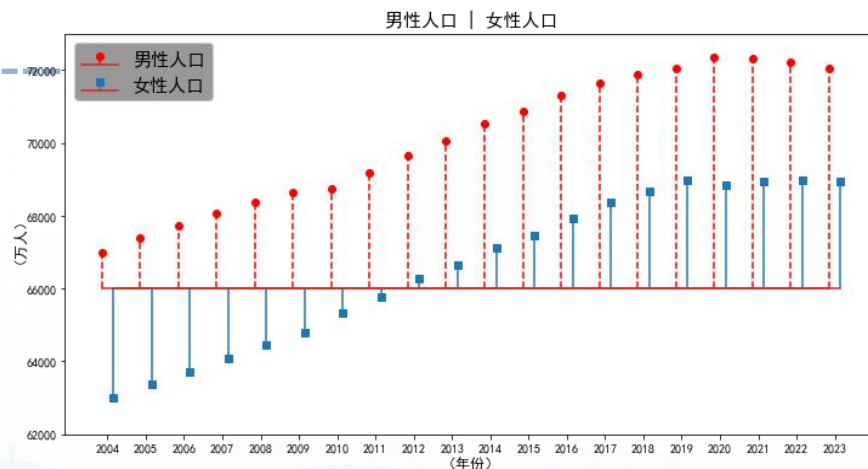




绘制茎叶图

>>> 绘制茎叶图比较“男性人口”和“女性人口”数据

```
plt.stem(df.index-0.15,df['男性人口(万人)'],linefmt='r--',  
         markerfmt='ro',use_line_collection=True,bottom=66000)  
plt.stem(df.index+0.15,df['女性人口(万人)'],  
         markerfmt='s',use_line_collection=True,bottom=66000)
```

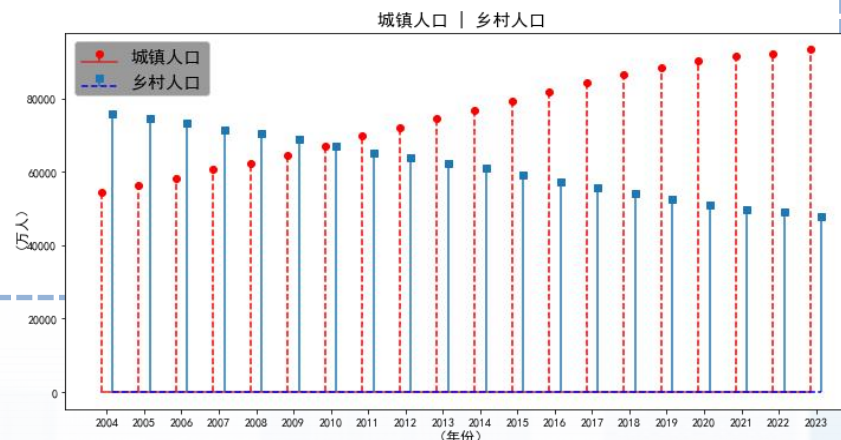




绘制茎叶图

绘制茎叶图比较“城镇人口”和“乡村人口”数据变化

```
plt.stem(df.index-0.15,df['城镇人口(万人)'],linefmt='r--',
         markerfmt='ro', use_line_collection=True )
plt.stem(df.index+0.15,df['乡村人口(万人)'],
         markerfmt='s',use_line_collection=True,basefmt='b--')
plt.legend(['城镇人口','乡村人口'],loc=0,fontsize=15,facecolor='gray')
plt.title('城镇人口 | 乡村人口',fontsize=15)
plt.xticks(df.index)
plt.ylabel(' (万人) ',fontsize=12)
plt.xlabel(' (年份) ',fontsize=12)
plt.show()
```

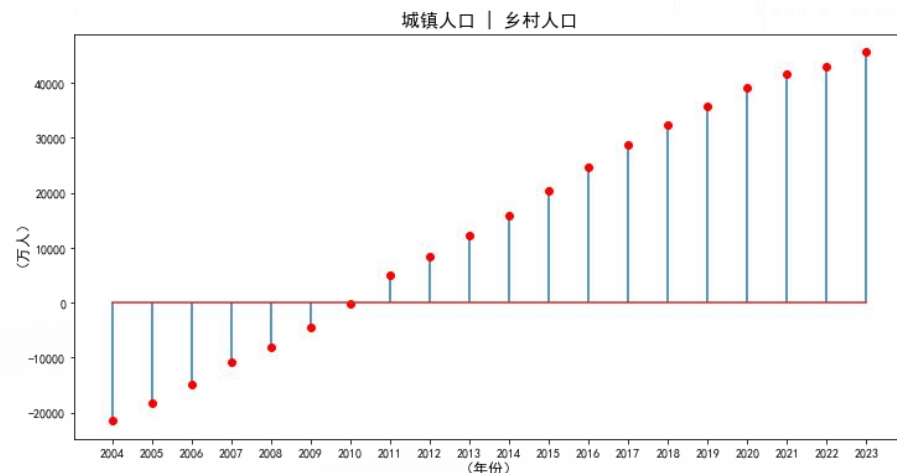




绘制茎叶图

>>> 绘制茎叶图比较“城镇人口”和“乡村人口”数据变化

```
plt.stem(df.index, df['城镇人口(万人)'] - df['乡村人口(人)'],  
         markerfmt='ro', use_line_collection=True)
```





>>> 绘制误差棒图

`matplotlib.pyplot.errorbar()`

- 误差棒图是使用误差棒注明被测量数据的不确定度大小的图表，用于表示测量数据中客观存在的测量偏差。

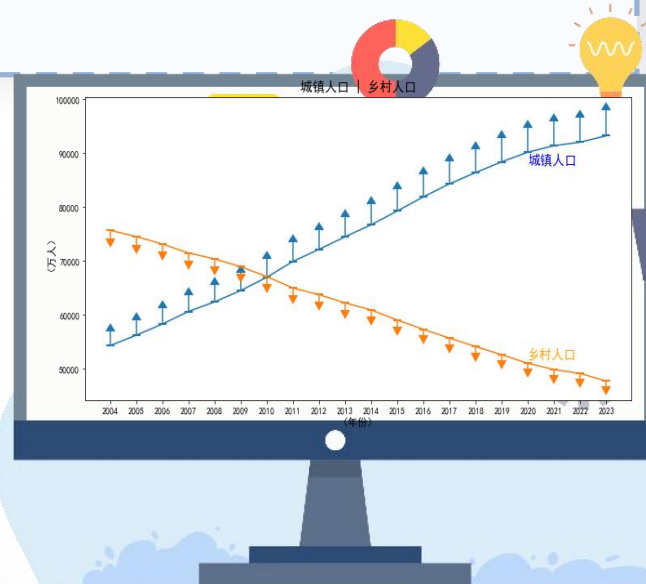
此处可以放一个动态的误差棒图素材





绘制误差棒图

- ❖ 误差棒图是使用误差棒注明被测量数据的不确定度大小的图表，用于表示测量数据中客观存在的测量偏差。
- ❖ 误差棒是在表示测量值大小的方向上的一条线段，它以被测量数据的平均值为中点，线段长度的一半为不确定度





绘制误差棒图

>>> 基础语法

errorbar(x, y, yerr=None, xerr=None, fmt='', ecolor=None, data=None, **kwargs)



参数	说明
x, y	表示数据点的位置。
xerr, yerr	表示数据的误差范围。
fmt	表示数据点的标记样式和数据点之间连接线的样式。默认为 '-' 实线。若设置为标记点样式，则不再显示折线。
marker	表示数据点的标记样式
markersize	表示数据点的标记样式大小
elinewidth	表示误差棒的线条宽度。
ecolor	表示误差棒的颜色
capsize	表示误差棒边界横杆的大小。默认无
capthick	表示误差棒边界横杆的厚度。
uplims, lolims	表示y轴误差棒上下限
xuplims, xlolims	表示x轴误差棒上下限

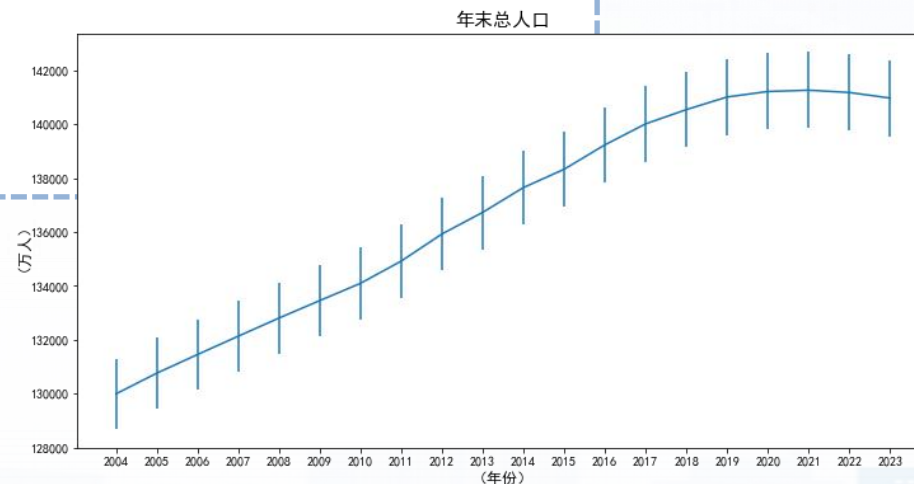


绘制误差棒图

>>> 绘制误差棒图展示近20年人口增长趋势

```
df=pd.read_excel('人口数据.xlsx','总人口',index_col=0)
fig=plt.figure(figsize=(12,6))
plt.errorbar(df.index,df['年末总人口(万人)'],
             yerr=df['年末总人口(万人)']*0.01)

plt.xticks(df.index)
plt.xlabel(' (年份) ',fontsize=12)
plt.ylabel(' (万人) ',fontsize=12)
plt.title('年末总人口',fontsize=15)
plt.show()
```

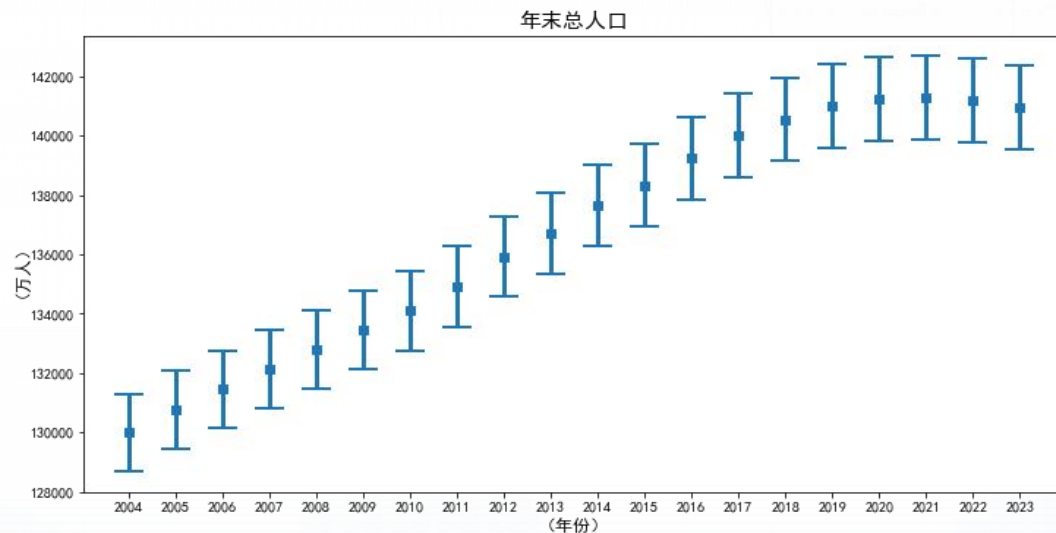




绘制误差棒图

>>> 绘制误差棒图展示近20年人口增长趋势

```
plt.errorbar(df.index, df['年末总人口(万人)'],  
             yerr=df['年末总人口(万人)']*0.01,  
             fmt='s', elinewidth=3, capsize=10, capthick=2)
```

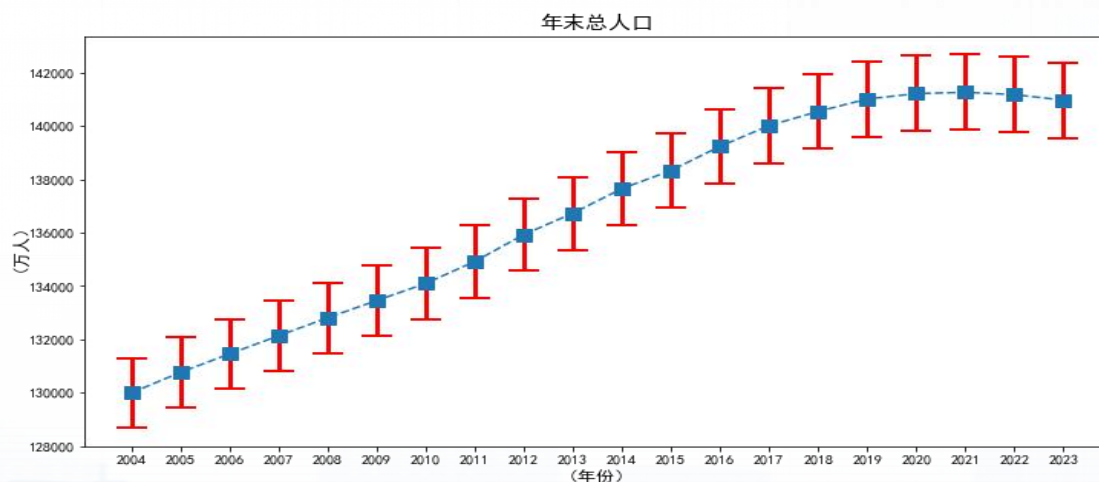




绘制误差棒图

>>> 绘制误差棒图展示近20年人口增长趋势

```
plt.errorbar(df.index, df['年末总人口(万人)'],  
             yerr=df['年末总人口(万人)']*0.01,  
             fmt='--', elinewidth=3, capsize=10, capthick=2,  
             ecolor='r', marker='s', markersize=10)
```

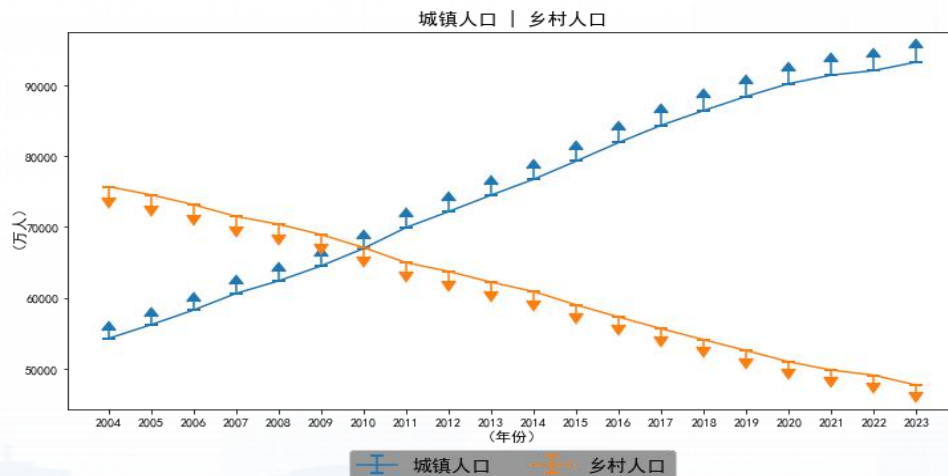




绘制误差棒图

>>> 绘制误差棒图比较“城镇人口”和“乡村人口”数据变化

```
plt.errorbar(df.index, df['城镇人口(万人)'], yerr=df['城镇人口(万人)']*0.02,  
             capsize=5, capthick=2, lolims=True)  
plt.errorbar(df.index, df['乡村人口(万人)'], yerr=df['乡村人口(万人)']*0.02,  
             capsize=5, capthick=2, uplims=True)
```





>>> 添加无指向型注释文本

```
matplotlib.pyplot.text(x, y, s, fontdict=None, **kwargs)
```

参数	说明
x, y	接收float，表示注释文本的x轴和y轴坐标位置。
s	接收str，表示注释文本的内容。
fontdict	接收dict，表示字体格式



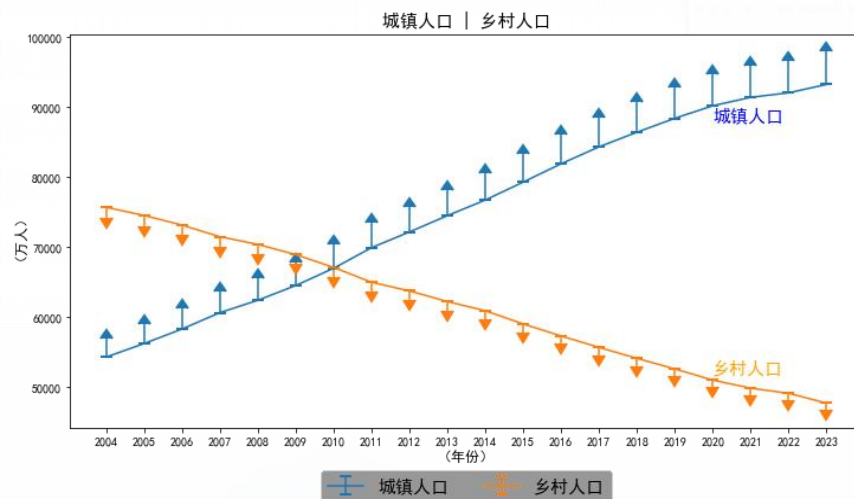


绘制误差棒图

>>> 添加无指向型注释文本

```
plt.text(2020, 52000, '乡村人口', {'fontsize': 15, 'color': 'orange'})
```

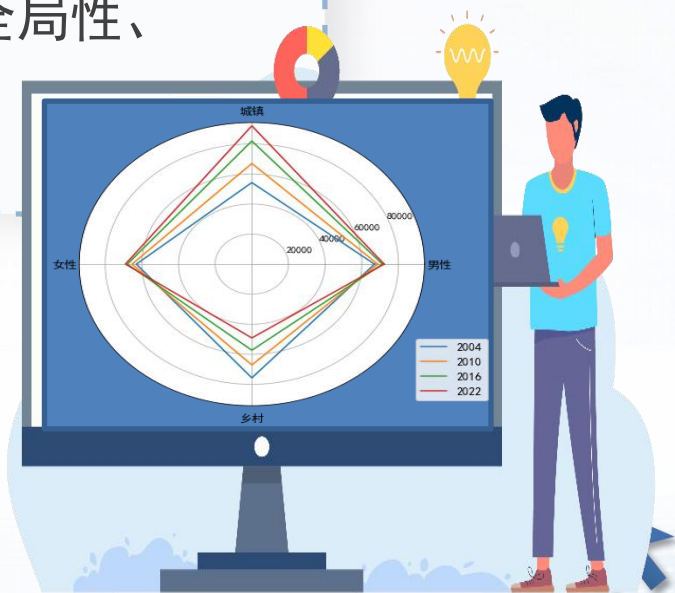
```
plt.text(2020, 88000, '城镇人口', {'fontsize': 15, 'color': 'blue'})
```





绘制雷达图

- 雷达图又称蜘蛛网图、星状图、极区图，由一组坐标轴和多个等距同心圆或多边形组成，是一种表现多维（4维以上）数据的图表。
- 雷达图的坐标轴起始于圆心，每个坐标轴代表一个指标，连接多个维度的数据点，围成一个多边形，适用于对多指标对象做出全局性、整体性评价。





>>> 基础语法

`matplotlib.pyplot.polar(theta, r, **kwargs)`



参数	说明
theta	表示每个数据点所在射线与极径的夹角。
r	表示每个数据点到原点的距离。

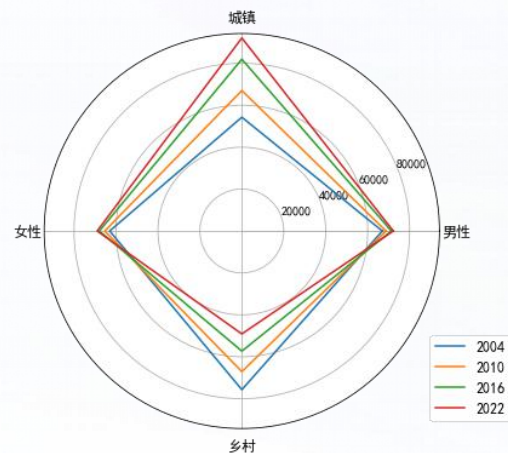




绘制雷达图

绘制雷达图展示2004年、2010年、2016年、2022年城镇人口、农村人口、男性人口、女性人口

```
df=pd.read_excel('人口数据.xlsx','总人口',index_col=0)
fig=plt.figure(figsize=(8,8))
df1=df.iloc[:,6,[1,3,2,4]]
data=df1.T
radar_labels=df1.columns
dim_num=len(radar_labels)
angles=np.linspace(0,2*np.pi,dim_num,endpoint=False)
plt.figure(figsize=(6,6))
plt.polar(angles,data,linestyle='None',marker='*',markersize=10,alpha=0.6)
plt.thetagrids(angles*180/np.pi,labels=['城镇','男性','乡村','女性'],fontsize=12)
plt.legend(df1.index,loc=4,bbox_to_anchor=(1.4,0),fontsize=12)
plt.title('城镇人口 乡村人口 男性人口 女性人口',fontsize=15)
plt.show()
```

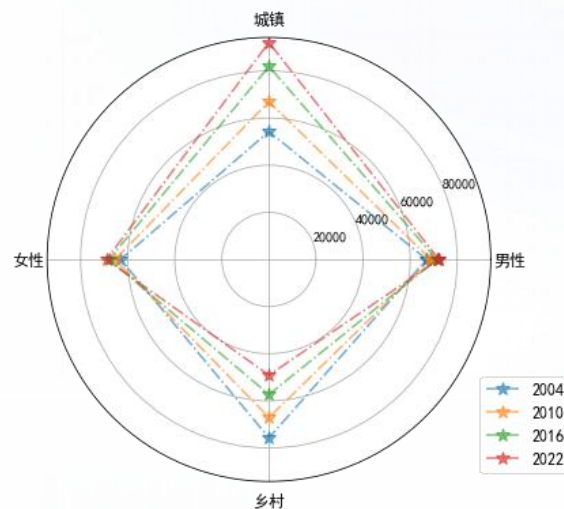




绘制雷达图

绘制雷达图展示2004年、2010年、2016年、2022年城镇人口、农村人口、男性人口、女性人口

- 雷达图的数据点可以使用marker参数设置标注类型，并可以同时设置标注的其他属性参数；也可以使用linestyle设置线型并设置线段其他属性。
- 修改绘制雷达图代码，指定参数linestyle='-.'设置线型，指定marker='*'设置标注点类型，并同时指定了标注大小markersize=10和透明度alpha=0.6：
- `plt.polar(angles2, data, linestyle='-.', marker='*', markersize=10, alpha=0.6)`

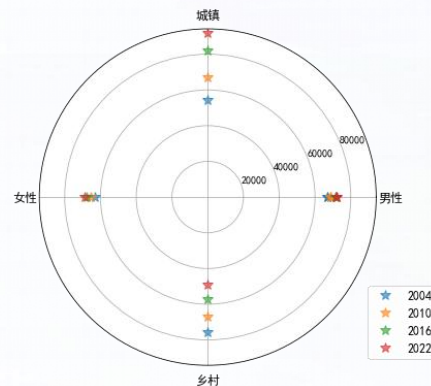




绘制雷达图

>>> 绘制雷达图展示2004年、2010年、2016年、2022年 城镇人口、农村人口、男性人口、女性人口

```
df1=df.iloc[:,6,[1,3,2,4]]
data=df1.T
radar_labels=df1.columns
dim_num=len(radar_labels)
angles=np.linspace(0,2*np.pi,dim_num,endpoint=False)
plt.figure(figsize=(6,6))
plt.polar(angles,data,linestyle='None',marker='*',markersize=10,alpha=0.6)
plt.thetagrids(angles*180/np.pi,labels=['城镇','男性','乡村','女性'],fontsize=12)
plt.legend(df1.index,loc=4,bbox_to_anchor=(1.4,0),fontsize=12)
plt.title('城镇人口 乡村人口 男性人口 女性人口',fontsize=15)
plt.show()
```



因为图形只展示数据点不连线，因此也不需要闭环，则数据及标签也不需要处理



目录

时间	年末总人口(万人)	0-14岁人口(万人)	15-64岁人口(万人)	65岁及以上人口(万人)	总抚养
2023年					
2022年	141175	23908	96289	20978	
2021年	141260	24678	96526	20056	
2020年	141212	25277	96871	19064	
2019年	141008	23689	99552	17767	
2018年	140541	23751	100065	16724	
2017年	140011	23522	100528	15961	
2016年	139232	23252	100943	15037	
2015年	138326	22824	100978	14524	
2014年	137646	22712	101032	13902	

注：1. 1981年及以前人口数据为户籍统计数；1982、1990、2000、2010、2020年数据为当年人口普查数据推算数；其余年份数据为年度人口抽样调查推算数据；2011—2019年数据根据2020年人口普查数据修订。
2. 总人口和按性别分人口中包括现役军人，按城乡分人口中现役军人计入城镇人口。



人口增长趋势图形展示



人口年龄结构变化图形展示



受教育程度数据图形展示



目录

时间	年末总人口(万人)	0-14岁人口(万人)	15-64岁人口(万人)	65岁及以上人口(万人)	总抚养
2023年					
2022年	141175	23908	96289	20978	
2021年	141260	24678	96526	20056	
2020年	141212	25277	96871	19064	
2019年	141008	23689	99552	17767	
2018年	140541	23751	100065	16724	
2017年	140011	23522	100528	15961	
2016年	139232	23252	100943	15037	
2015年	138326	22824	100978	14524	
2014年	137646	22712	101032	13902	

注：1. 1981年及以前人口数据为户籍统计数；1982、1990、2000、2010、2020年数据为当年人口普查数据推算数；其余年份数据为年度人口抽样调查推算数据；2011—2019年数据根据2020年人口普查数据修订。
2. 总人口和按性别分人口中包括现役军人，按城乡分人口中现役军人计入城镇人口。



人口增长趋势图形展示



人口年龄结构变化图形展示



受教育程度数据图形展示



绘制区域堆叠图

- 堆叠区域图是由若干条折线与折线或水平坐标轴之间的填充区域组成的图表，用于强调每个部分变化的趋势；堆叠柱形图和堆叠条形图是由若干个以颜色或线条填充、高度不一的纵向矩形条或横向矩形条堆叠而成的图表，主要用于反映各构成部分在总体中的比重。
- 堆叠区域图可以很方便地比较数据间不同情况下的差异（分工比例等），显示完整的数据以进行可视化。它显示了每个数据相互堆叠以及每个数据如何构成完整的图形。它显示数据的各种组成部分，类似于饼图，具有x-label, y-label和标题，其中各个部分可以用不同的颜色表示。



>>> 基础语法

stackplot(x,y,labels=(),baseline='zero',data=None,*args,kwargs)**

参数	说明
x	形状为(N,)的类数组结构，即尺寸为N的一维数组。必备参数。表示X轴的数据，可以是一维数组。
y	形状为(M,N)的类数组结构，即尺寸为(M,N)的二维数组。必备参数。表示Y轴的数据，可以是二维数组或一维数组序列。y参数有两种应用方式。stackplot(x, y1, y2, y3): y1,y2,y3,y4均为一维数组且长度为N。stackplot(x, y): y的形状为(M, N)
labels	为每个数据系列指定标签。长度为N的字符串列表。表示每个填充区域的标签。
baseline	字符串，取值范围为{'zero','sym','wiggle','weighted_wiggle'}。默认值为'zero'。可选参数。表示计算基线的方法。zero表示恒定零基线（以0为基线），绘制简单的堆叠区域图；sym表示对称于零基线（以0上下对称），主题河流图；wiggle表示所有序列的最小化平方斜率之和；weighted_wiggle表示执行相同的操作，类似于'wiggle'，但是增加各层的大小作为权重。绘制出的图形也被称为流图（streamgraph）
colors	颜色列表或元组。每组区域所使用的颜色，循环使用



绘制区域堆叠图

➤➤➤ 绘制区域堆叠图根据需要设置参数baseline。
将对对应代码分别修改为：

```
plt.stackplot(x,y,baseline='zero')  
plt.title('baseline=zero',fontsize=15)
```

或：

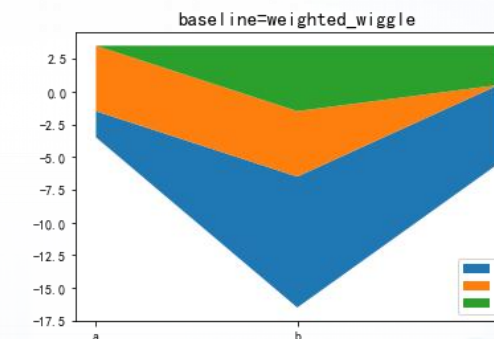
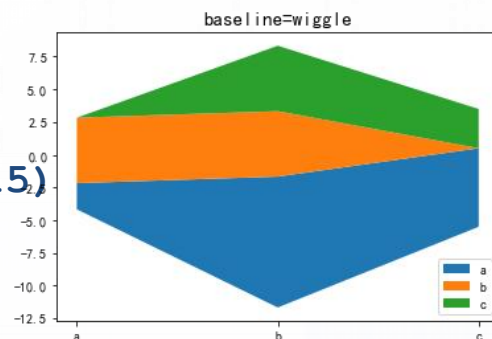
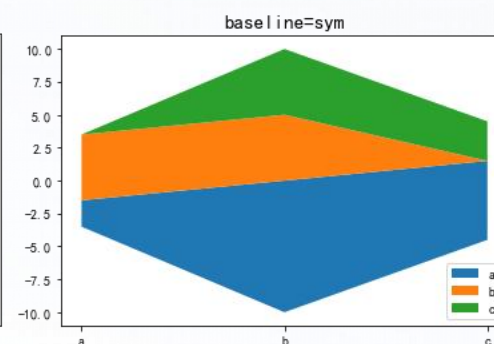
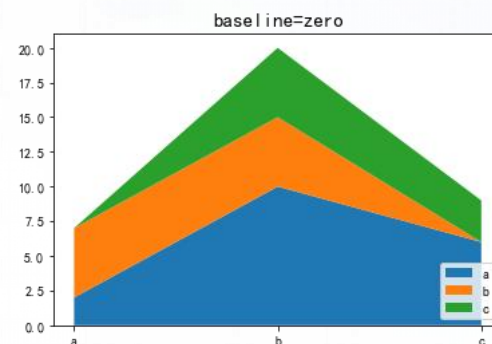
```
plt.stackplot(x,y,baseline='sym')  
plt.title('baseline=sym',fontsize=15)
```

或：

```
plt.stackplot(x,y,baseline='wobble')  
plt.title('baseline=wobble',fontsize=15)
```

或：

```
plt.stackplot(x,y,baseline='weighted_wobble')  
plt.title('baseline=weighted_wobble',fontsize=15)
```

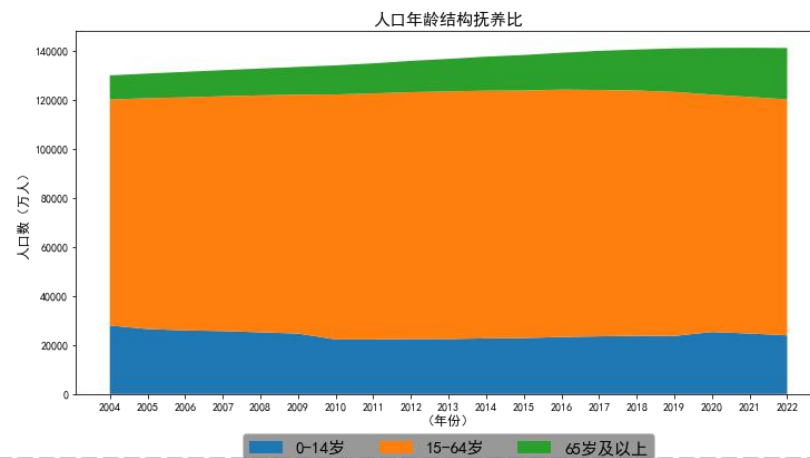




绘制区域堆叠图

>>> 绘制区域堆叠图对比历年各年龄段数据

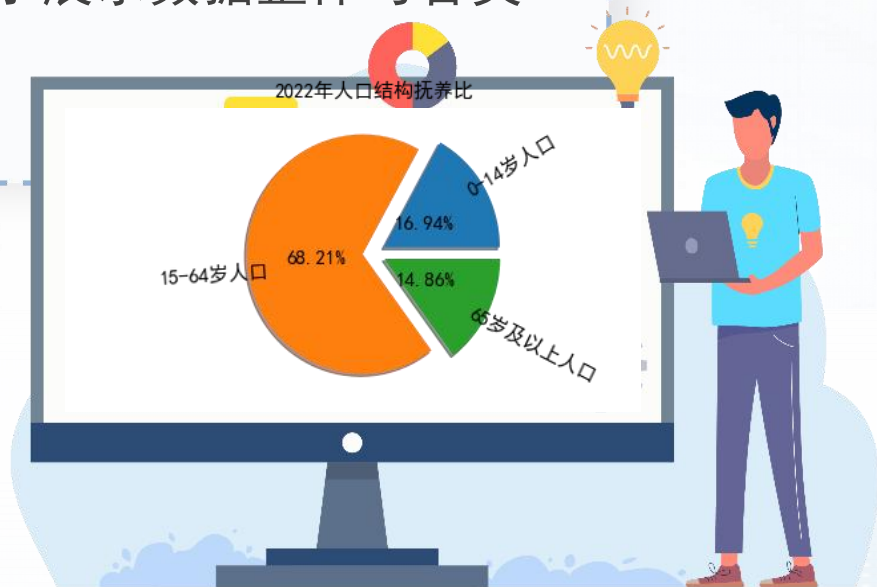
```
df=pd.read_Excel('人口数据.xlsx','人口年龄结构抚养比',index_col=0)
fig=plt.figure(figsize=(12,6))
labels=['0-14岁','15-64岁','65岁及以上']
plt.stackplot(df.index,df['0-14岁人口(万人)'],df['15-64岁人口(万人)'],df['65岁及以上人口(万人)'])
plt.xticks(df.index)
plt.ylabel('人口数(万人)',fontsize=12)
plt.xlabel('(年份)',fontsize=12)
plt.legend(labels,loc=8,fontsize=15,
facecolor='gray',ncol=3,
bbox_to_anchor=(0.5,-0.21))
plt.title('人口年龄结构抚养比',fontsize=15)
plt.show()
```





绘制饼图或圆环图

- 饼图是由若干个面积大小不一、颜色不同的扇形组成的圆形图表，它使用圆表示数据的总量，组成圆的每个扇形表示数据中各项占总量的比例大小，主要用于显示数据中各项大小与各项总和的比例。
- 饼图中的圆与扇形分别代表整体与部分，适用于展示数据整体与各类数据的占比关系。





绘制饼图或圆环图

>>> 基础语法

`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=None )`

参数	说明
x	接收类似一维数组结构数据，指定用于绘制饼图各个扇形的尺寸
explode	接收类似一维数组结构数据，指定每个扇形相对于饼圆半径的偏移距离，取值为小数，默认值为None
labels	字符串列表。指定每个扇形的标签。默认值为None
colors	接收类似一维数组结构数据，指定每个扇形的颜色。颜色会循环使用。默认值为None，使用当前色彩循环
autopct	接收特定str或可调用对象，指定扇形内显示的标签字符串。默认值为None。如果值为格式字符串，标签将被格式化，如果值为函数，将被直接调用。可通过格式字符串指定小数点后的位数
pctdistance	接收float。指定扇形内标签与圆心的距离。默认值为0.6，autopct不为None则该参数生效



绘制饼图或圆环图

>>> 基础语法

`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=None )`

参数	说明
shadow	接收bool，指定饼图下是否显示阴影。默认值为False
labeldistance	接收float。指定扇形外标签与圆心的距离（相对于半径的比例）。默认值为1.1。如果设置为None，标签不会显示，但是图例可以使用标签
rotatelabels	接收bool，指定扇形外标签是否按扇形角度旋转。默认为False
startangle	接收float。指定扇形起始绘制角度。默认值为0，即从X轴的正方向逆时针旋转绘制
counterclock	接收bool，指定角度是否逆时针旋转。默认值为True
radius	接收float。指定圆形半径。默认值为1



绘制饼图或圆环图

>>> 基础语法

`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=None )`

参数	说明
wedgeprops	接收dict。指定扇形属性。例如，通过wedgeprops={'width':0.5}将宽度设为0.5。默认值为None
textprops	接收dict。指定图表中的文本属性。默认值为None
center	接收浮点数元组(float, float)。指定饼图中心点位置坐标。默认值为(0, 0)
frame	接收bool，指定是否绘制子图边框。默认为False
normalize	接收bool，指定是否归一化。默认值为None。True：完全饼图，对x进行归一化， $\text{sum}(x) == 1$ 。False：如果 $\text{sum}(x) \leq 1$ ，绘制不完全饼图。如果 $\text{sum}(x) > 1$ ，抛出ValueError异常



绘制饼图或圆环图

>>> 绘制饼图，观察近年来我国人口年龄结构的变化趋势

```
df=pd.read_Excel('人口数据.xlsx','人口年龄结构抚养比',index_col=0)
data=df.loc[2022,['0-14岁人口(万人)','15-64岁人口(万人)','65岁及以上人口(万人)']]
plt.pie(data)
labels=['0-14岁人口','15-64岁人口','65岁及以上人口']
plt.legend(labels,loc='lower center',fontsize=15,facecolor='gray',
           ncol=3,bbox_to_anchor=(0.5,-0.1))
plt.title('2022年人口结构抚养比',fontsize=15)
```

2022年人口结构抚养比



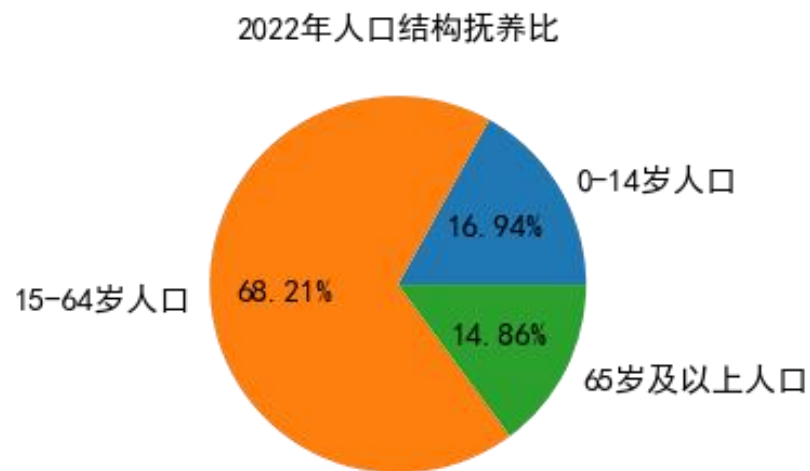
0-14岁人口 15-64岁人口 65岁及以上人口



绘制饼图或圆环图

>>> 设置参数给饼图添加数据文本

```
plt.pie(data, autopct='%.2f%%', labels=labels, textprops={'fontsize':15})
```



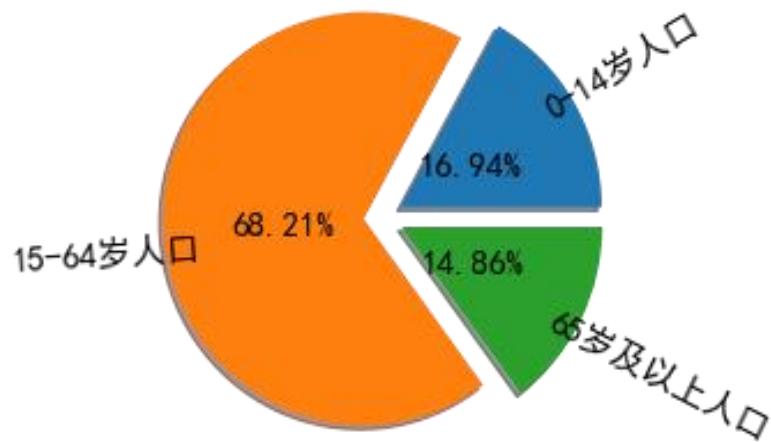


绘制饼图或圆环图

>>> 设置参数使扇形分离

```
plt.pie( data, autopct='% .2f%%', textprops={'fontsize':15},  
        labels=labels, pctdistance=0.4, rotatelabels=45,  
        labeldistance=0.8, explode=(0.1, 0.1, 0.1), shadow=True )
```

2022年人口结构抚养比



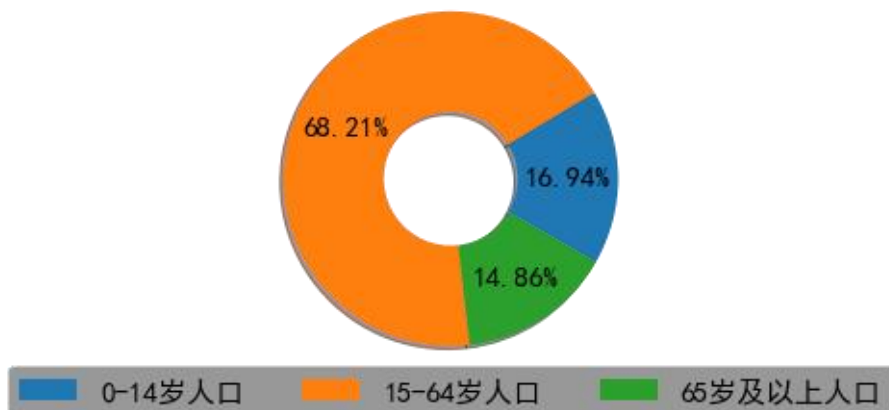


绘制饼图或圆环图

>>> 设置参数wedgeprops绘制圆环图

```
plt.pie( data, autopct='%.2f%%', pctdistance=0.7, rotatelabels=45,  
        labeldistance=0.8, textprops={'fontsize':15},  
        shadow=True, startangle=-30, wedgeprops={'width':0.6} )
```

2022年人口结构抚养比

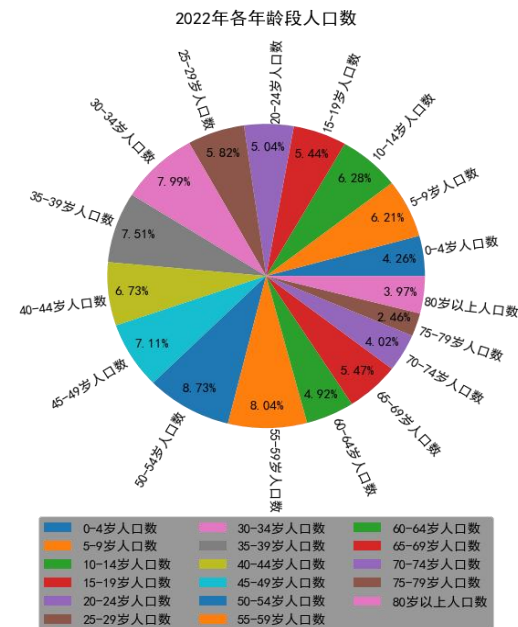




绘制饼图或圆环图

>>> 绘制各年龄段人口数饼图

```
plt.pie( data,radius=2,autopct='%.2f%%',  
         pctdistance=0.85,rotatelabels=45,  
         labels=labels,labeldistance=1,  
         textprops={'fontsize':15} )  
plt.title('2022年各年龄段人口数',fontsize=20,x=0.5,y=1.8)  
plt.legend(labels,loc='lower center',fontsize=15,  
           facecolor='gray', ncol=3,  
           bbox_to_anchor=(0.5,-1.4))  
plt.show()
```



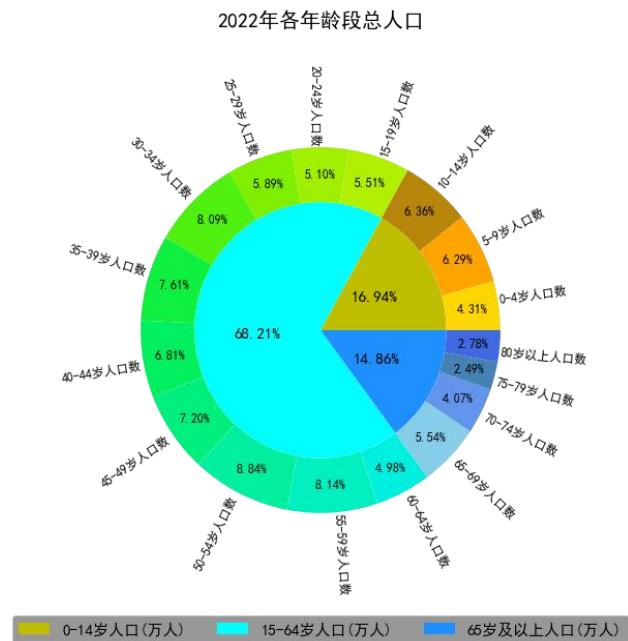


绘制饼图或圆环图

>>> 两个饼图叠加，绘制同心圆

```
labels_inner=data_inner.index
colors_inner=['y','cyan','dodgerblue']
plt.pie( data_inner,autopct='%.2f%%',pctdistance=0.5,
        colors=colors_inner, radius=1.4,
        textprops={'fontsize':15})
```

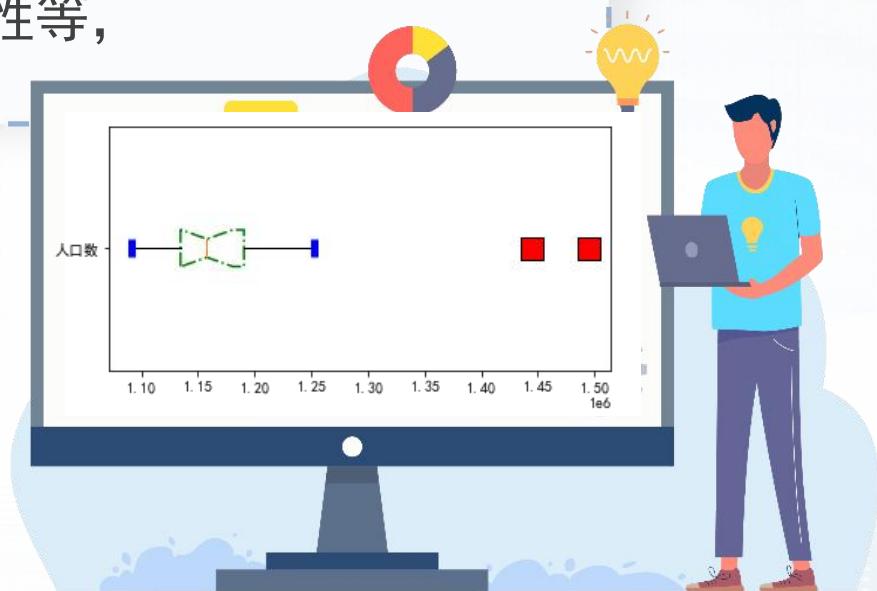
```
labels_outer=labels2[:-3]
colors_outer= ['gold','orange','darkgoldenrod',
               '#b0f000','#a0f000','#80f000','#50f010','#10f040',
               '#00f060','#00f080','#00f0a0','#00f0c0','#00f0e0',
               'skyblue','cornflowerblue','steelblue','royalblue']
plt.pie(data_outer,autopct='%.2f%%',pctdistance=0.85,
        textprops={'fontsize':12},colors=colors_outer,
        wedgeprops={'width':0.6},rotatelabels=45,
        labels=labels_outer,labeldistance=1,radius=2)
```





绘制箱线图

- 箱线图（boxplot）又称箱形图，因形状如箱子而得名。
- 箱线图用来展示数据分布的分散程度，还可以展示多组数据分布特征的比较，也可以粗略地看出数据是否具有对称性等，

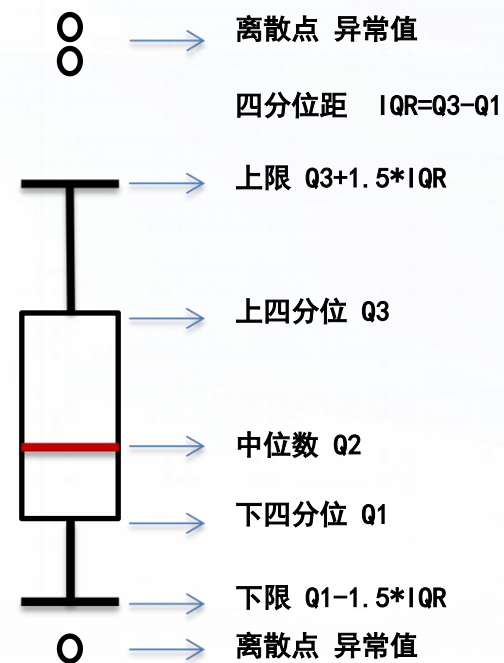




绘制箱线图

箱线图由六种数值组成：

- 异常值 (outlier/flier)
- 最小值 (minimum)
- 下四分位数 (Q1, 即第25%分位数)
- 中位数 (median, 即第50%分位数)
- 上四分位数 (Q3, 即第75%分位数)
- 最大值 (maximum)





>>> 基础语法

```
boxplot( x, notch=None, positions=None, whis=None, widths=None,
        patch_artist=None, bootstrap=None, usermedians=None,
        labels=None, conf_intervals=None, sym=None, vert=None,
        meanline=None, showmeans=None, showcaps=None,
        showbox=None, showfliers=None, boxprops=None,
        flierprops=None, medianprops=None, meanprops=None,
        zorder=None, whiskerprops=None, manage_ticks=True,
        data=None, autorange=False, *, capprops=None )
```

参数	说明
x	接收array或向量序列。表示输入数据。必备参数
labels	接收array。表示每个数据集的标签，默认值为'None'。可选参数
zorder	接收float，表示箱线图的叠放次序。默认值为Line2D. zorder = 2。可选参数



绘制箱线图

>>> 基础语法

参数	说明
showfliers	接收bool，表示是否显示离群值。默认值为True。可选参数
sym	接收str，离群点的默认标记符号。默认值为'None'。可选参数。取值为''隐藏离群点，取值为'None'时，取值为'b+'
flierprops	接收dict，表示离群点的样式。默认值为None。可选参数
showcaps	接收bool，表示是否显示箱须两端的横杠。默认值为True。可选参数
capprops	接收dict，表示箱须横杠的样式。默认值为None。可选参数
whiskerprops	接收dict，表示箱须的样式。默认值为None。可选参数
autorange	接收bool，默认值为False。可选参数。当取值为True且数据分布满足上四分位数（75%）和下四分位数（25%）相等，whis设置为(0, 100)，即箱须端点为数据的最大值和最小值
showbox	接收bool，表示是否显示箱体。默认值为True。可选参数
notch	接收bool，表示控制箱体中央是否有V形凹槽。当取值为True时，箱体中央有V形凹槽，凹槽表示中位数的置信区间；取值为False时，箱体为矩形。类型为布尔值，默认值为False。可选参数
vert	接收bool，箱体的方向，当取值为True时，绘制垂直箱体，当取值为False时，绘制水平箱体。类型为布尔值，默认值为True。可选参数
positions	接收array。表示箱体的位置。刻度和极值会自动匹配箱体位置。可选参数。默认值为range(1, N+1)，N为箱线图的个数
widths	接收float或array，表示箱体的宽度。默认值为0.5或0.15*极值间的距离
boxprops	接收dict，表示箱体的样式。默认值为None。可选参数
patch_artist	接收bool，表示控制箱体的生成对象。默认值为False。可选参数。当取值为False时，箱体由Line2D生成，否则，箱体由Patch对象生成



绘制箱线图

基础语法

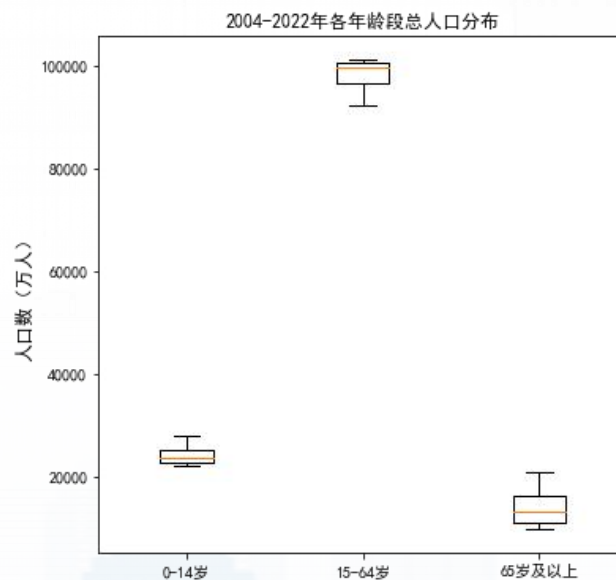
参数	说明
medianprops	接收dict, 表示中位数的样式。默认值为None。可选参数
manage_ticks	接收bool, 表示控制刻度和标签位置, 默认值为True。可选参数。取值为True时, 刻度和标签位置自动匹配箱线图的位置
showmeans	接收bool, 表示是否显示算术平均值。默认值为False。可选参数
medianprops	接收dict, 表示中位数的样式。默认值为None。可选参数
meanprops	接收dict, 表示算术平均值的样式。默认值为None。可选参数
manage_ticks	接收bool, 表示控制刻度和标签位置, 默认值为True。可选参数。取值为True时, 刻度和标签位置自动匹配箱线图的位置
meanline	接收bool, 表示均值显示为线还是点。默认值为False。可选参数。当取值为True, 且showmeans、shownotches参数均为True, 时显示为线, 线条属性受meanprops参数控制; 否则显示为点
meanprops	接收dict, 表示算术平均值的样式。默认值为None。可选参数
bootstrap	接收int, 表示是否使用bootstrap方法计算中位数置信区间。可选参数。当参数取值为None时, 中位数置信区间由某种高斯渐进逼近算法确定。否则, 采用bootstrap方法求中位数95%置信区间, bootstrap 参数定义了抽样次数。建议取值范围为1000-10000
usermedians	表示是否指定中位数。类型为一维类数组结构。可选参数。对于一维数组, 数组元素个数为1, 对于多维数组, 长度等于len(x)。如果元素不为None, 则将该值强制设置为对应数据集的中位数; 如果元素为None, 则由matplotlib生成中位数
conf_intervals	接收array。表示是否指定置信区间。形状为 (len(x), 2)。可选参数。如果元素不为None, 则将该值强制设置为箱体V型凹槽位置 (只有当notch参数为True时绘制); 如果元素为None, 箱体V型凹槽的位置由其他参数计算, 例如bootstrap
whis	接收float或float二元组, 表示箱须的位置。默认值为1.5。 可选参数。浮点数, 则下箱须位于高于 $Q1 - whis * (Q3 - Q1)$ 的最低数据处, 上箱须位于低于 $Q3 + whis * (Q3 - Q1)$ 下方的最高数据处, 其中Q1和Q3分别为下四分位数和上四分位数。默认值whis=1.5对应于箱线图的原始定义。浮点数元组, 则表示要在绘制箱须的百分位数 (例如, (5, 95))。将其设置为 (0, 100) 则箱须覆盖整个数据范围。当 $Q1 == Q3$ 时, 如果autorange为True, whis将自动设置为 (0, 100)。箱须范围之外的数据将被视为异常值, 绘制为点



绘制箱线图

➤➤➤ 绘制箱线图，观察近年来我国人口年龄结构变化趋势

```
plt.boxplot( [df['0-14岁人口(万人)'],  
              df['15-64岁人口(万人)'],  
              df['65岁及以上人口(万人)']],  
            labels=['0-14岁', '15-64岁', '65岁及以上'])
```

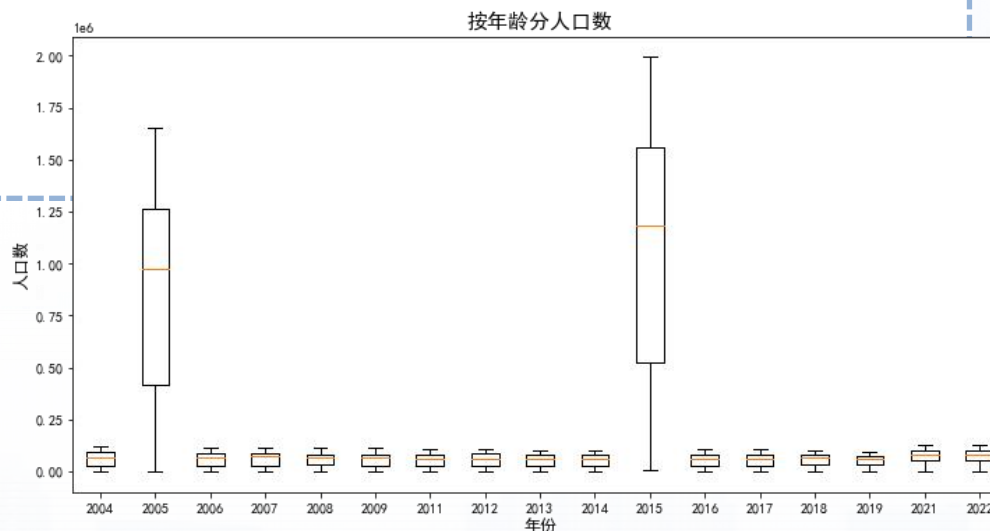




绘制箱线图

>>> 根据采样人口数绘制箱线图

```
df=pd.read_Excel('人口数据.xlsx','按年龄分人口数',index_col=0)
data=df.iloc[:,1:].T
fig=plt.figure(figsize=(12,6))
plt.boxplot(data,labels=data.columns)
plt.title('按年龄分人口数',fontsize=15)
plt.ylabel('人口数',fontsize=12)
plt.xlabel('年份',fontsize=12)
plt.show()
```

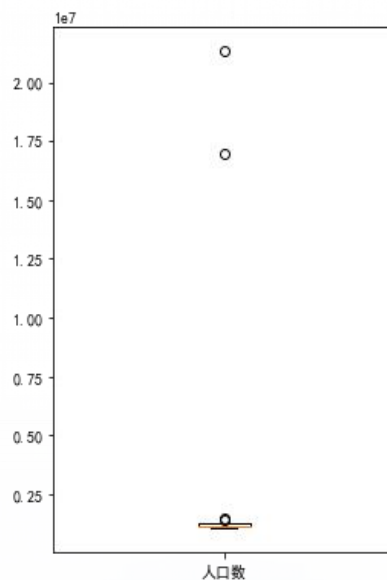




绘制箱线图

>>> 箱线图的离群点（异常值）

```
plt.boxplot(df['人口数'], labels=['人口数'])
```

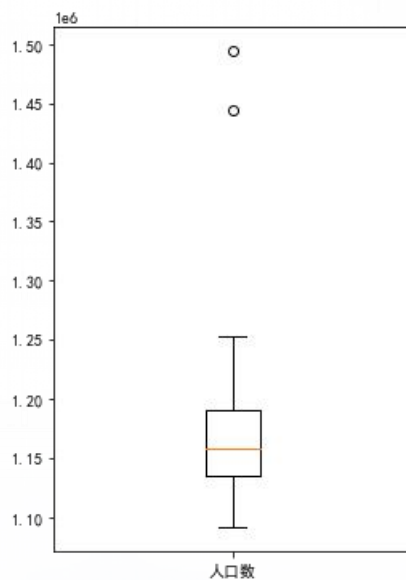




绘制箱线图

>>> 箱线图的离群点（异常值）

```
plt.boxplot(df['人口数'].drop([2005,2015]),labels=['人口数'])
```

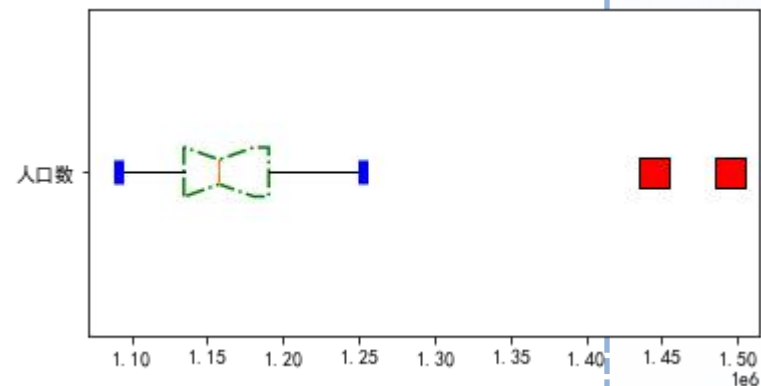




绘制箱线图

>>> 设置箱形图的外观

```
plt.boxplot(df['人口数'].drop([2005,2015]),labels=['人口数'],  
            vert = False,  
            notch = True,  
            flierprops = { "marker": "s",  
                           "markerfacecolor": "red",  
                           "markersize": 15},  
            capprops = {"linestyle":"--",  
                        "color":"b",  
                        "linewidth":5},  
            boxprops = { 'color': 'g',  
                          'linestyle':'-.',  
                          'linewidth':1.5},  
            meanline = {'color': 'red',  
                        'linewidth': '1.5'})
```





绘制直方图

直方图又称质量分布图，是由一系列高低不等的纵向矩形条或线段组成的图表，通常将连续的数据分割成若干区间，然后计算每个区间内数据点的数量，反映数据的分布和波动情况。数据点数量通常用柱形图表示，柱子的高度表示该区间内数据点的数量





绘制直方图

>>> 基础语法

hist(x,bins=None,range=None,density=None,weights=None,bottom=None)

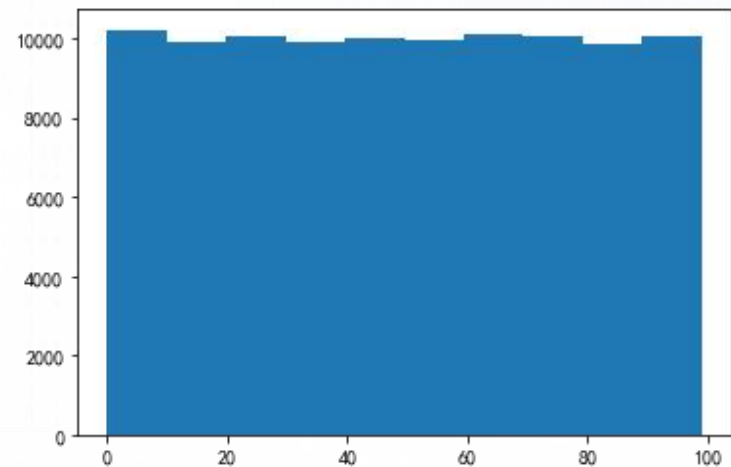
参数	说明
x	表示X轴的数据
bins	表示矩形条的个数，默认为10
range	表示X轴数据的范围，若未设置范围，默认数据范围为(x.min(),x.max())
bottom	表示Y轴数据刻度的起始位置
cumulative	表示是否计算累计频数或频率
histtype	表示直方图的类型，支持'bar'、'barstacked'、'step'、'stepfilled' 四种取值 'bar'为默认值，代表传统的直方图；'barstacked'代表堆叠直方图；'step'代表未填充的线条直方图；'stepfilled'代表填充的线条直方图。
align	表示矩形条边界的对齐方式，可设置为'left'、'mid'或'right'，默认为'mid'。
orientation	表示矩形条的摆放方式，默认为'vertical'，即垂直方向。水平方向：horizontal
rwidth	表示矩形条宽度的百分比，默认为0。若histtype的值为'step'或'stepfilled'，则直接忽略rwidth参数的值
stacked	表示是否将多个矩形条以堆叠形式摆放
facecolor	直方图颜色
edgecolor	直方图边框颜色
alpha	透明度



绘制直方图

>>> 数据均匀分布

```
data=np.random.randint(0,100,100000)  
  
plt.hist(data)  
  
plt.show()
```

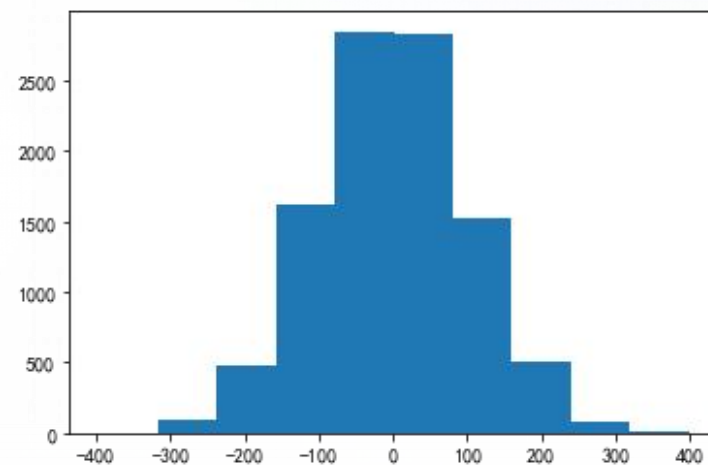




绘制直方图

>>> 数据正态分布

```
data=np.random.normal(0,10  
0,10000)  
  
plt.hist(data)  
  
plt.show()
```

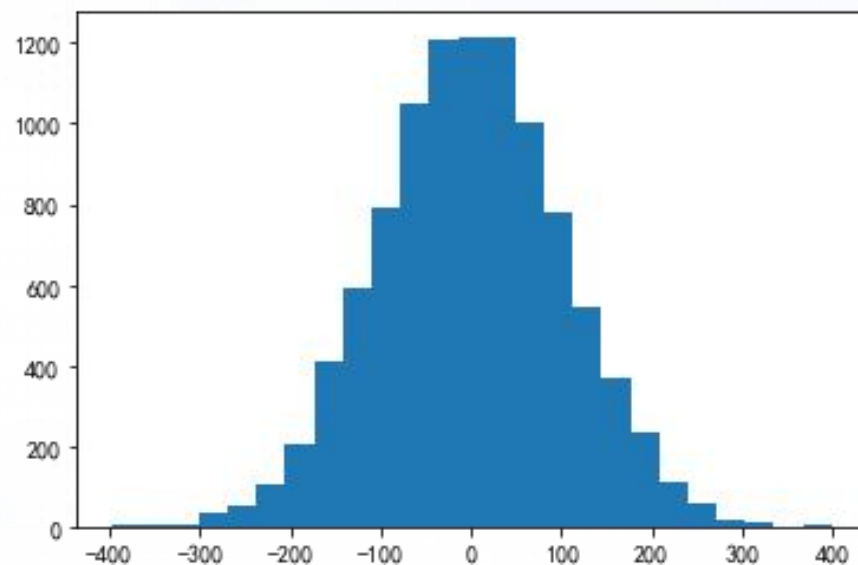




绘制直方图

➤➤➤ 参数`bin=25`，设置柱形的数量

```
plt.hist(data,bins=25)
```

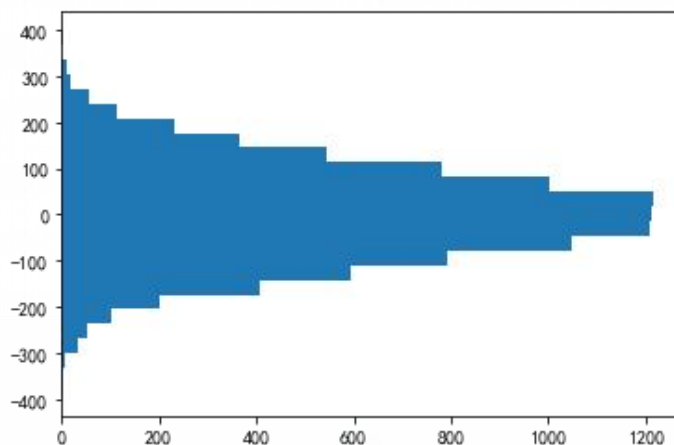




绘制直方图

➤➤➤ 参数`orientation='horizontal'`，设置柱形方向

```
plt.hist(data,bins=25,orientation='horizontal')
```





目录/CONTENTS



Matplotlib库概述



基础图形绘制



划分画布绘制子图



目录



绘制单子图 `subplot()`



添加子图 `add_subplot()`



绘制多子图 `subplots()`



>>> 基础语法

```
pyplot.subplot(nrows, ncols, index, **kwargs)
```

参数：

nrows, ncols : 表示子网格的行数、列数

index : 表示子网格区域的索引



绘制单子图

>>> 演示案例

<code>subplot(2, 2, 1)</code>	<code>subplot(2, 2, 2)</code>
<code>subplot(2, 2, 3)</code>	<code>subplot(2, 2, 4)</code>

`subplot(221)` 等价于 `subplot(2, 2, 1)`



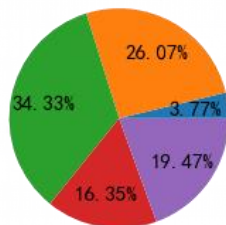


绘制单子图

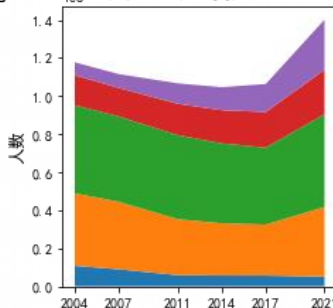
>>> 绘制图形观察“按受教育程度分人口数”

```
fig = plt.figure(figsize=(16,8))
```

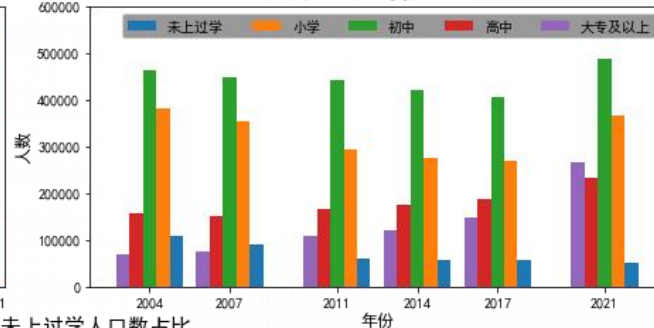
2022年各受教育程度人数占比



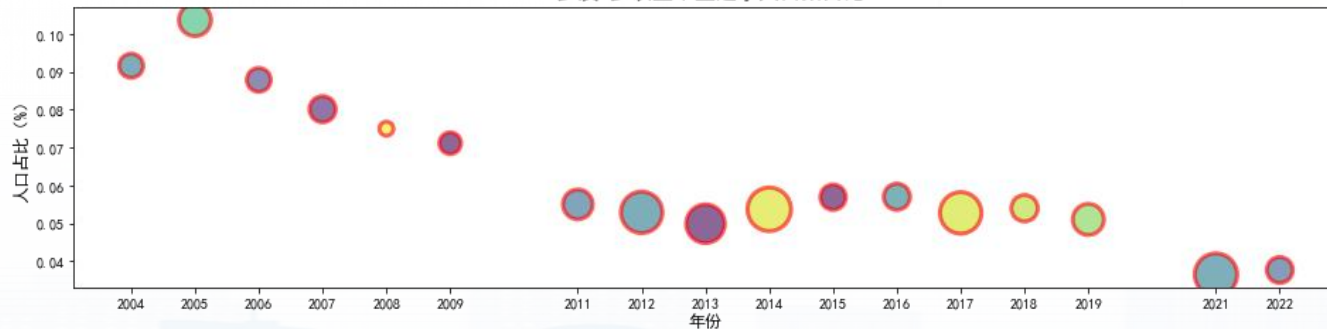
各受教育程度人口数



各受教育程度人口数



6岁及6岁以上未上过学人口数占比

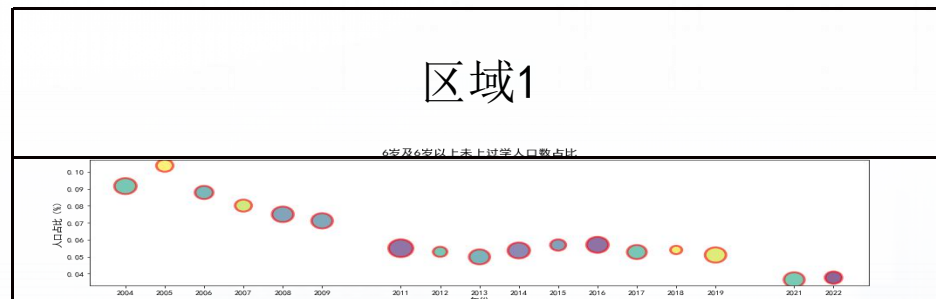




绘制单子图

>>> 绘制图形观察 “按受教育程度分人口数”

```
plt.subplot(2,1,2)
plt.scatter(df.index,df['6岁及6岁以上未上过学人口数']/df['6岁及6岁以上人口数'],
            c=np.random.rand(17),s=np.random.randint(50,1000,17),
            alpha = 0.6,linewidths=3,edgecolors='r' )
plt.ylabel('人口占比 (%) ',fontsize=12)
plt.xlabel('年份',fontsize=12)
plt.xticks(df.index)
plt.title('6岁及6岁以上未上过学人口数占比',fontsize=15)
```

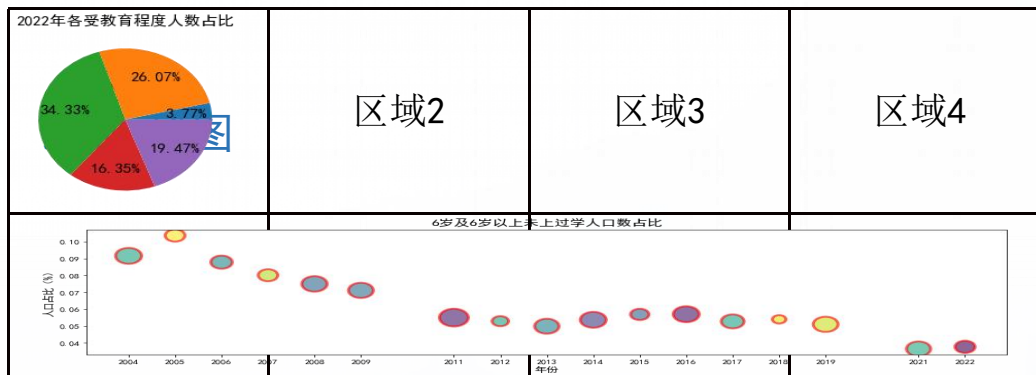




绘制单子图

>>> 绘制图形观察 “按受教育程度分人口数”

```
plt.subplot(2,4,1)
data3=df.iloc[-1,3::3]
p_inner=plt.pie( data3,autopct='%.2f%%',textprops={'fontsize':15},pctdistance=0.7 )
plt.title('2022年各受教育程度人数占比',fontsize=15)
```

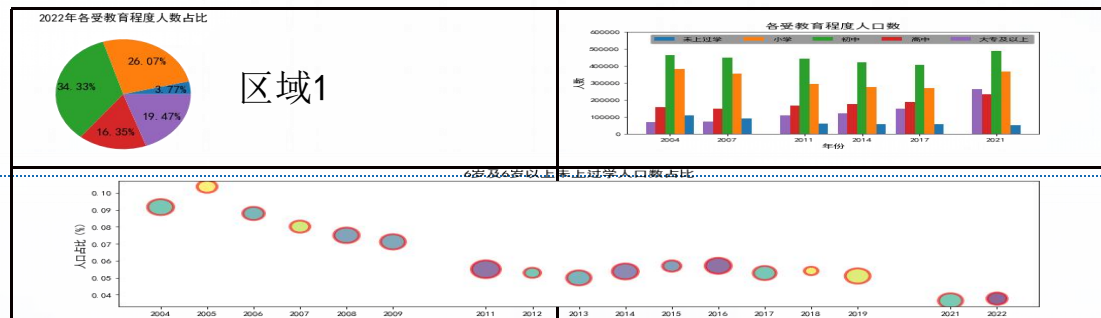




绘制单子图

>>> 绘制图形观察 “按受教育程度分人口数”

```
plt.subplot(2,2,2)
plt.bar(data.index+1,data['6岁及6岁以上未上过学人口数'],width=0.5)
plt.bar(data.index+0.5,data['6岁及6岁以上小学人口数'],width=0.5)
plt.bar(data.index,data['6岁及6岁以上初中人口数'],width=0.5)
plt.bar(data.index-0.5,data['6岁及6岁以上高中人口数'],width=0.5)
plt.bar(data.index-1,data['6岁及6岁以上大专及以上学历人口数'],width=0.5)
plt.legend(['未上过学','小学','初中','高中','大专及以上学历'],loc=1,fontsize=10,facecolor='gray',ncol=5)
plt.xticks(data.index)
plt.ylim([0,600000])
plt.ylabel('人数',fontsize=12)
plt.xlabel('年份',fontsize=12)
plt.title('各受教育程度人口数',fontsize=15)
```

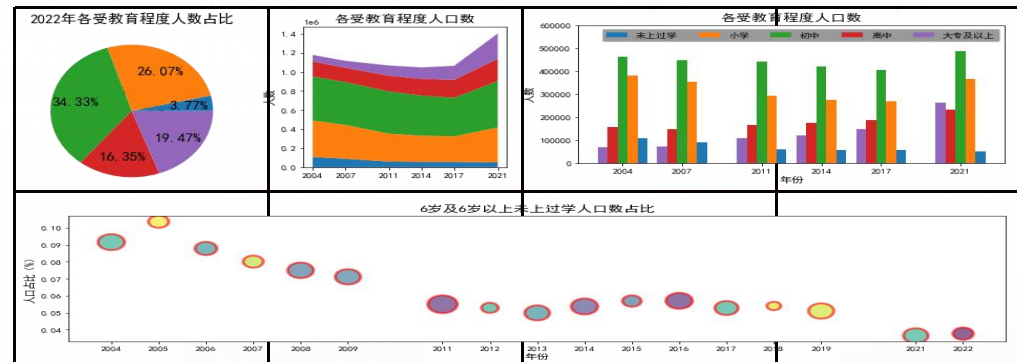




绘制单子图

>>> 绘制图形观察 “按受教育程度分人口数”

```
plt.subplot(2,4,2)
plt.stackplot(data.index,data['6岁及6岁以上未上过学人口数'],
data['6岁及6岁以上小学人口数'],data['6岁及6岁以上初中人口数'],
data['6岁及6岁以上高中人口数'],data['6岁及6岁以上大专及以上人口数'])
plt.xticks(data.index)
plt.ylabel('人数',fontsize=12)
plt.title('各受教育程度人口数',fontsize=15)
```





目录

指标	2022年	2021年	2019年	2018年	2017年	2016年
15岁及以上人口数(人口抽样调查)(人)	1199109	1232677	908609	951685	952893	965321
15岁及以上男性人口数(人口抽样调查)(人)	607967	625238	459235	481358	482331	488944
15岁及以上女性人口数(人口抽样调查)(人)	591141	607439	449374	470327	470561	476376
15岁及以上文盲人口数(人口抽样调查)(人)	40704	39531	41696	47007	46221	50980
15岁及以上男性文盲人口数(人口抽样调查)(人)	9614	9386	10185	11641	11668	13402
15岁及以上女性文盲人口数(人口抽样调查)(人)	31089	30145	31511	35366	34554	37578

注： 2005年、2015年为1%人口抽样调查样本数据，其他年份为1‰人口变动调查样本数据。具体抽样比详见出版物中《中国统计年鉴》。



人口增长趋势图形展示



人口年龄结构变化图形展示



受教育程度数据图形展示



目录



绘制单子图 `subplot()`



添加子图 `add_subplot()`



绘制多子图 `subplots()`



添加子图

>>> 基础语法

```
add_subplot(* args, ** kwargs)
```



参数：

***args参数：**表示一个三位数的实数或三个独立的实数，用于描述子图的位置。比如“nrows, ncols, index”，其中nrows和ncols表示将Figure对象分割成nrows行ncols列的区域，index表示当前选中的要操作的区域。

返回：一个axes类的子对象，可以详细设置坐标轴各参数。



添加子图

>>> 演示案例

<code>add_subplot(2, 2, 1)</code>	<code>add_subplot(2, 2, 2)</code>
<code>add_subplot(2, 2, 3)</code>	<code>add_subplot(2, 2, 4)</code>

`add_subplot(221)`

等价于

`add_subplot(2, 2, 1)`

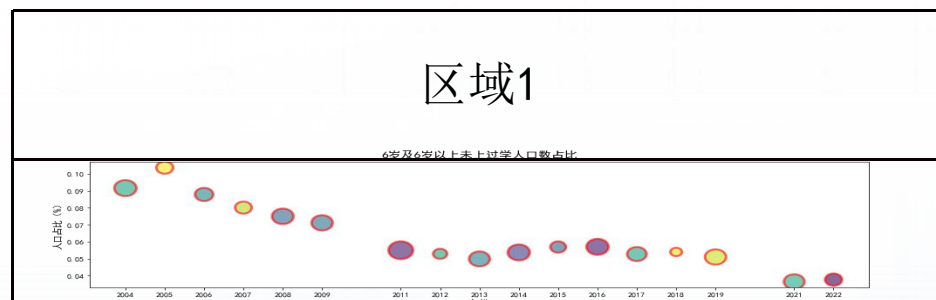




添加子图

>>> 绘制图形观察 “按受教育程度分人口数”

```
plt.add_subplot(2,1,2)
plt.scatter(df.index,df['6岁及6岁以上未上过学人口数']/df['6岁及6岁以上人口数'],
            c=np.random.rand(17),s=np.random.randint(50,1000,17),
            alpha = 0.6,linewidths=3,edgecolors='r' )
plt.ylabel('人口占比 (%) ',fontsize=12)
plt.xlabel('年份',fontsize=12)
plt.xticks(df.index)
plt.title('6岁及6岁以上未上过学人口数占比',fontsize=15)
```

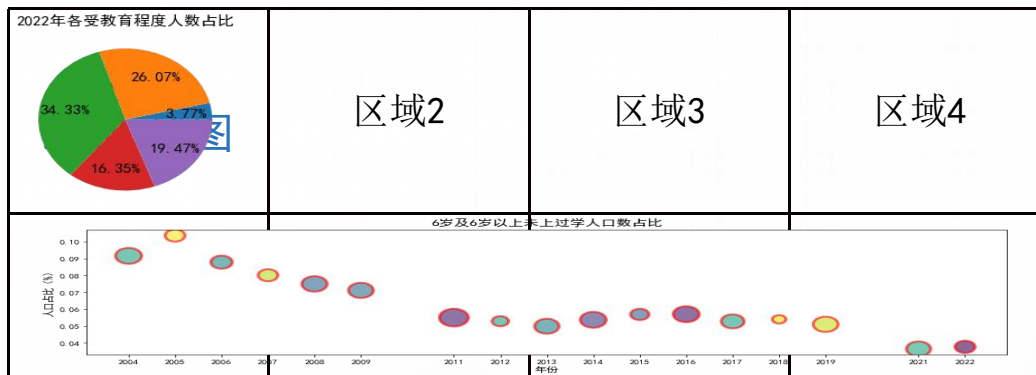




添加子图

>>> 绘制图形观察 “按受教育程度分人口数”

```
plt.add_subplot(2,4,1)
data3=df.iloc[-1,3::3]
p_inner=plt.pie( data3,autopct='%.2f%%',textprops={'fontsize':15},pctdistance=0.7 )
plt.title('2022年各受教育程度人数占比',fontsize=15)
```

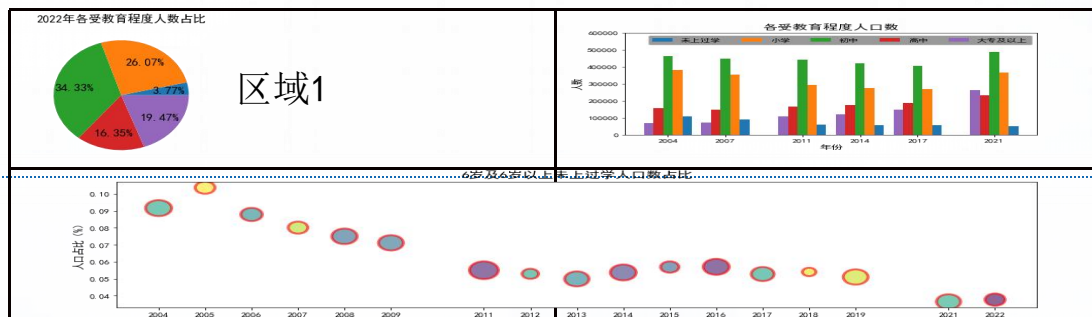




添加子图

>>> 绘制图形观察 “接受教育程度分人口数”

```
plt.add_subplot(2,2,2)
plt.bar(data.index+1,data['6岁及6岁以上未上过学人口数'],width=0.5)
plt.bar(data.index+0.5,data['6岁及6岁以上小学人口数'],width=0.5)
plt.bar(data.index,data['6岁及6岁以上初中人口数'],width=0.5)
plt.bar(data.index-0.5,data['6岁及6岁以上高中人口数'],width=0.5)
plt.bar(data.index-1,data['6岁及6岁以上大专及以上学历人口数'],width=0.5)
plt.legend(['未上过学','小学','初中','高中','大专及以上学历'],loc=1,fontsize=10,facecolor='gray',ncol=5)
plt.xticks(data.index)
plt.ylim([0,600000])
plt.ylabel('人数',fontsize=12)
plt.xlabel('年份',fontsize=12)
plt.title('各受教育程度人口数',fontsize=15)
```

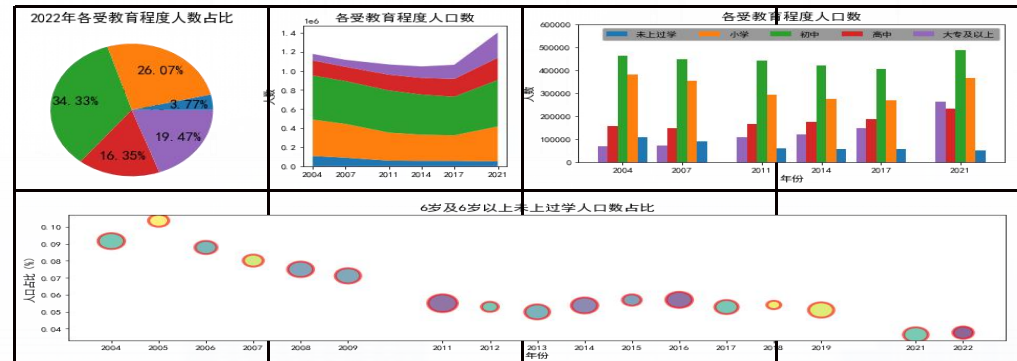




添加子图

>>> 绘制图形观察 “按受教育程度分人口数”

```
plt.add_subplot(2,4,2)
plt.stackplot(data.index,data['6岁及6岁以上未上过学人口数'],
data['6岁及6岁以上小学人口数'],data['6岁及6岁以上初中人口数'],
data['6岁及6岁以上高中人口数'],data['6岁及6岁以上大专及以上人口数'])
plt.xticks(data.index)
plt.ylabel('人数',fontsize=12)
plt.title('各受教育程度人口数',fontsize=15)
```





目录



绘制单子图 `subplot()`



添加子图 `add_subplot()`



绘制多子图 `subplots()`



>>> 基础语法

```
subplots( nrows = 1, ncols = 1, sharex = False, sharey = False,  
          squeeze = True, subplot_kw = None, gridspec_kw = None )
```

参数：

nrows, ncols : 表示子网格的行数、列数

sharex, sharey : 表示各子区域网格是否共享X轴或Y轴，默认False

返回值：

一个元组，元组的第一个元素为Figure对象（画布），第二个元素为Axes对象（子图，包含坐标轴和画的图）或Axes对象数组。如果创建的是单个子图，则返回的是一个Axes对象，否则返回的是一个Axes对象数组。

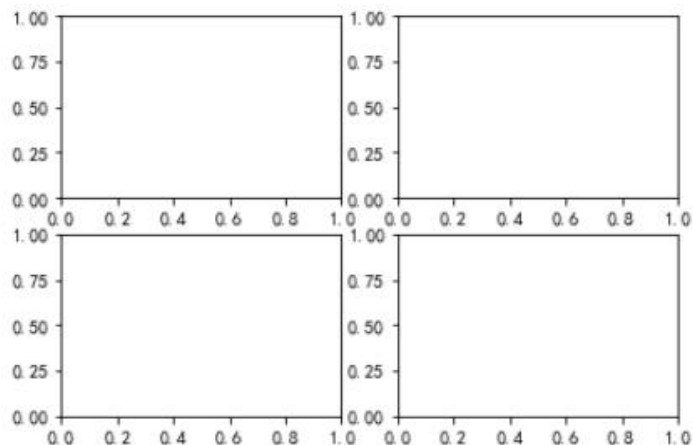


绘制多子图

>>> 演示案例

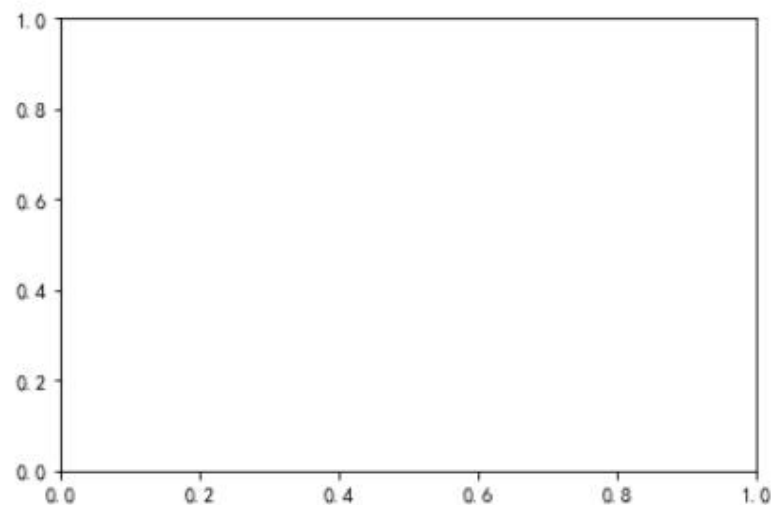
`plt.subplots(2,2)`

```
(<Figure size 432x288 with 4 Axes>,  
array([[<AxesSubplot:>, <AxesSubplot:>],  
       [<AxesSubplot:>, <AxesSubplot:>]], dtype=object))
```



`plt.subplots(1,1)`

```
(<Figure size 432x288 with 1 Axes>, <AxesSubplot:>)
```

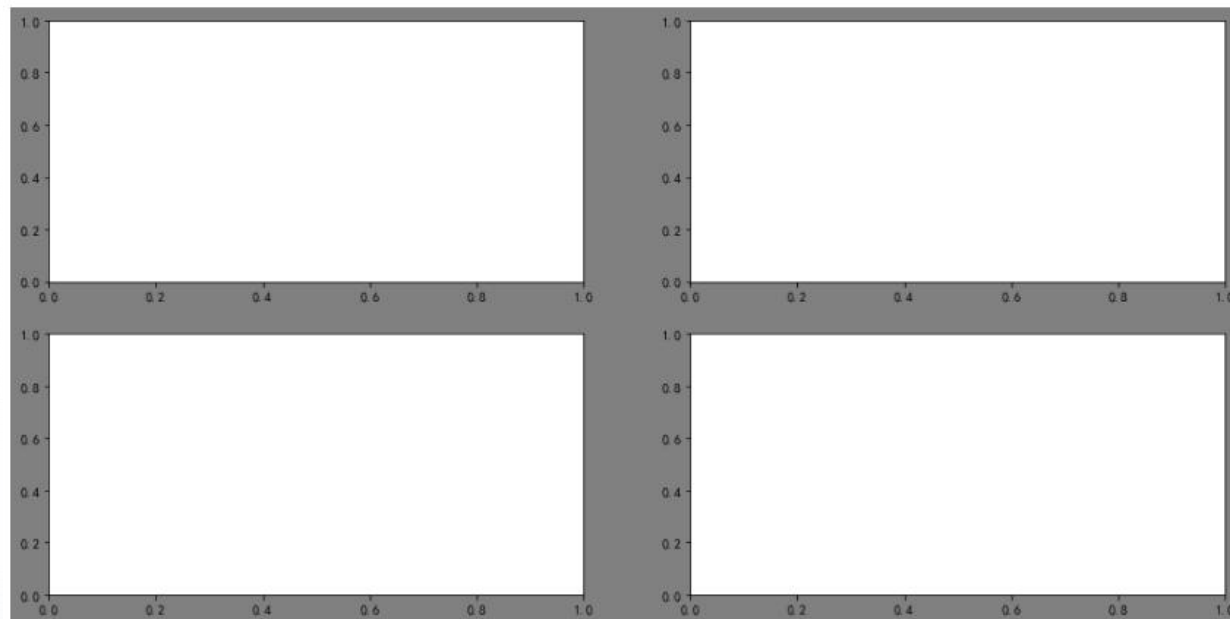




绘制多子图

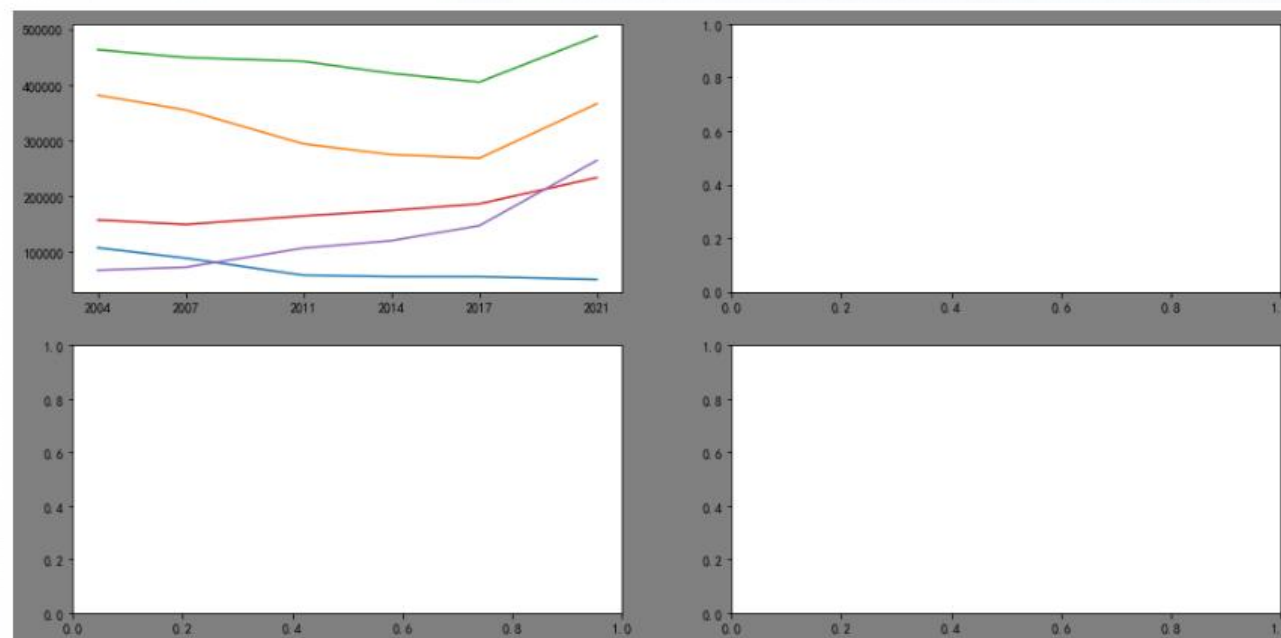
>>> 演示案例

```
fig = plt.figure(figsize=(16,8),facecolor='gray')  
axes_array = fig.subplots(2,2)
```



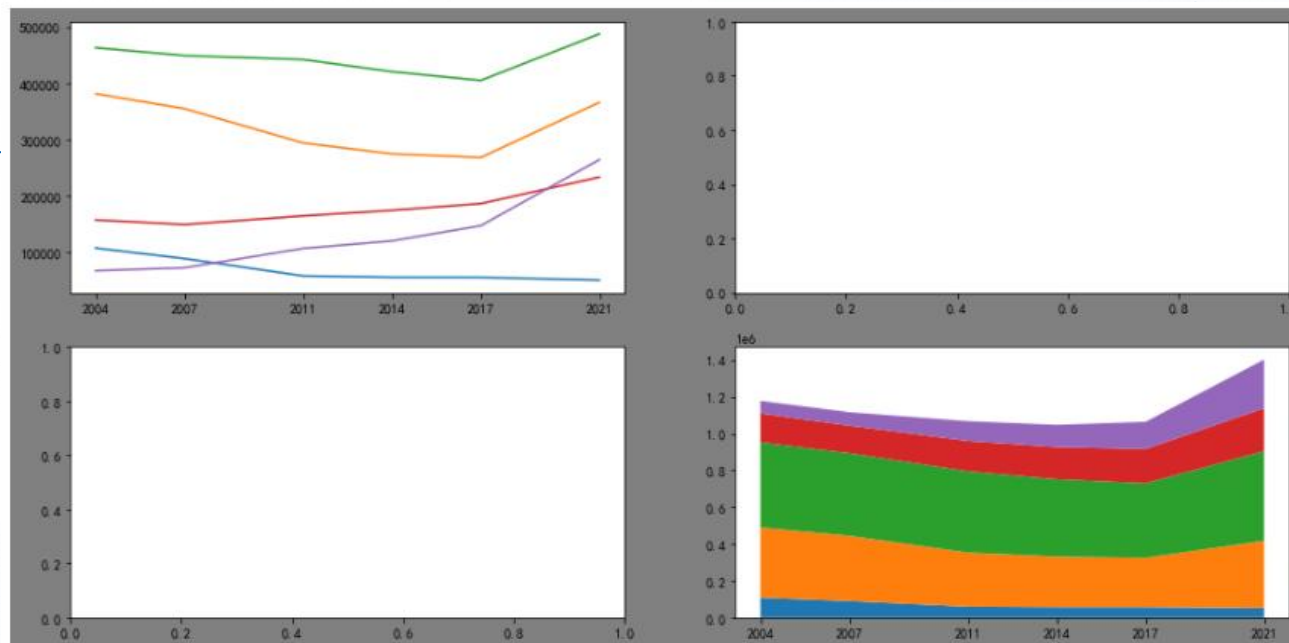
>>> 绘制图形观察“按受教育程度分人口数”

```
axes00=axes_arr[0,0]  
axes00.plot(df)  
axes00.set_xticks(df.index)  
axes00.set_xticklabels(ticklabels)
```



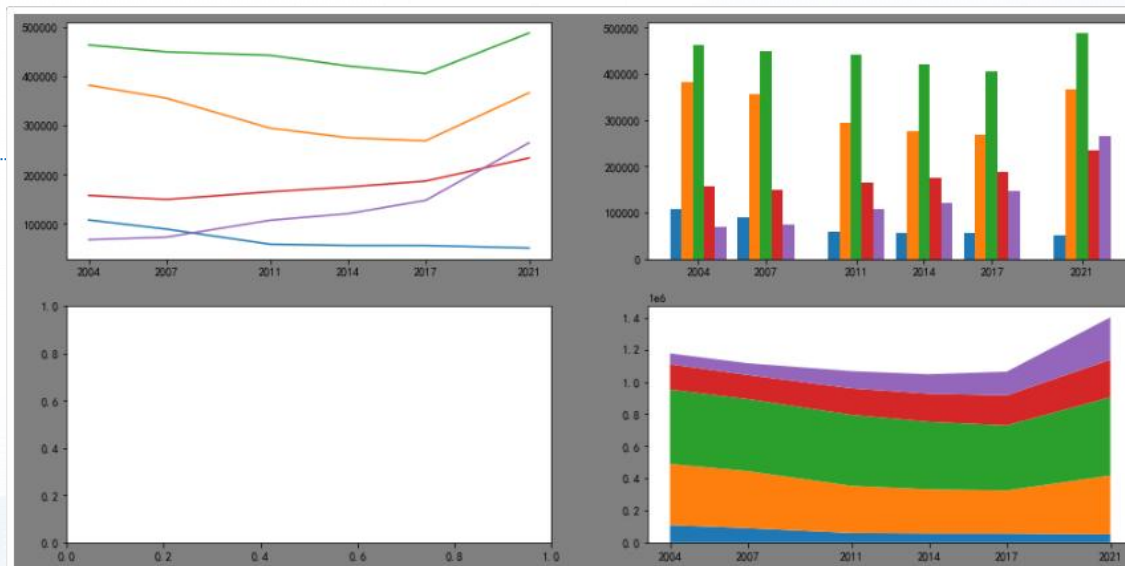
>>> 绘制图形观察 “按受教育程度分人口数”

```
axes11=axes_arr[1,1]  
axes11.stackplot( df.index,df['6岁及6岁以上未上过学人口数'],df['6岁及6岁以上小学人口数'],  
                  df['6岁及6岁以上初中人口数'],df['6岁及6岁以上高中人口数'],  
                  df['6岁及6岁以上大专及以上学历人口数'])  
axes11.set_xticks(df.index)  
axes11.set_xticklabels(ticklabels)
```



>>> 绘制图形观察 “按受教育程度分人口数”

```
axes01=axes_arr[0,1]
axes01.bar(df.index-1,df['6岁及6岁以上未上过学人口数'],width=0.5)
axes01.bar(df.index-0.5,df['6岁及6岁以上小学人口数'],width=0.5)
axes01.bar(df.index,df['6岁及6岁以上初中人口数'],width=0.5)
axes01.bar(df.index+0.5,df['6岁及6岁以上高中人口数'],width=0.5)
axes01.bar(df.index+1,df['6岁及6岁以上大专及以上学历人口数'],width=0.5)
axes01.set_xticks(df.index)
axes01.set_xticklabels(ticklabels)
```



目录



绘制单子图 `subplot()`



添加子图 `add_subplot()`



绘制多子图 `subplots()`

多子图共享坐标轴

- `sharex`
- `sharey`

多子图共享坐标轴

```
subplots( nrows = 1, ncols = 1, sharex = False, sharey = False,  
          squeeze = True, subplot_kw = None, gridspec_kw = None )
```

参数`sharex`和`sharey`使各子图共享X轴或Y轴。

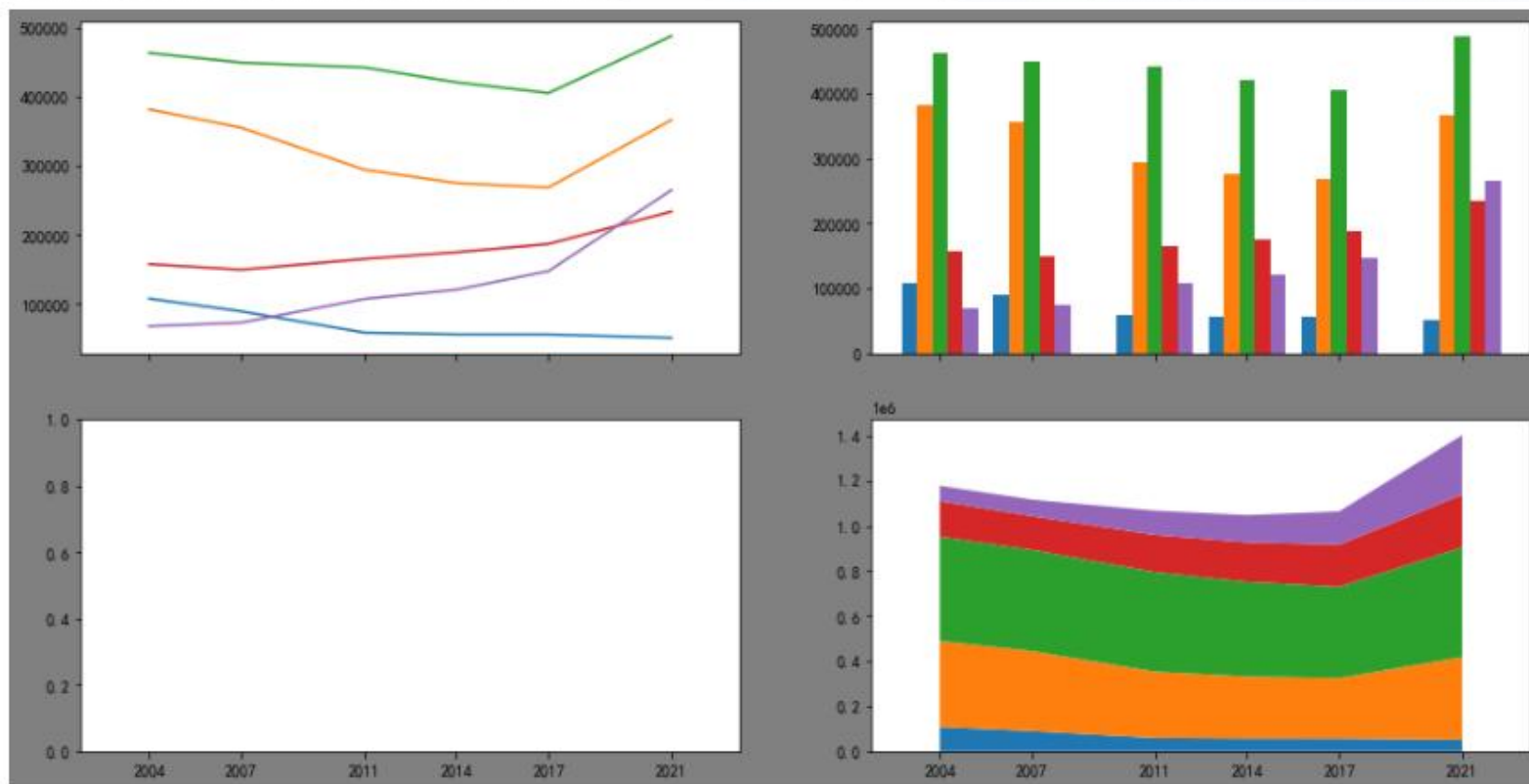
取值：

- True或'all'：表示所有子图之间共享X轴或Y轴。
- False或'none'：表示所有子图之间不共享X轴或Y轴。
- 'row'：表示每一行的子图之间共享X轴或Y轴。
- 'col'：表示每一列的子图之间共享X轴或Y轴。



>>> 多子图共享X轴

```
axes_arr=fig.subplots(2,2,sharex=True)
```





总结

技术准备

`figure()` 创建画布

`title()` 添加标题

`axhline()` 添加参考线

`grid()` 添加网格

`show()` 展示图形

`save()` 保存图形

划分画布绘制子图

`subplot()` 绘制单子图

`add_subplot()` 添加子图

`subplots()` 绘制多子图

图形绘制

`plot()` 绘制折线图

`plot()` 函数绘制折线图

`scatter()` 函数绘制散点图

`bar()` 函数绘制柱形图

`barh()` 函数绘制条形图使用

`stem()` 函数绘制茎叶图

`errorbar()` 函数绘制误差棒图

`polar()` 函数绘制雷达图

`stackplot()` 函数绘制堆叠区域图

`pie()` 函数绘制饼图和圆环图

`boxplot()` 函数绘制箱线图

`hist()` 绘制直方图



[人口普查数据图形展示] matplotlib.pyplot模块

主讲：孙 睿

