安徽省人口数据分析

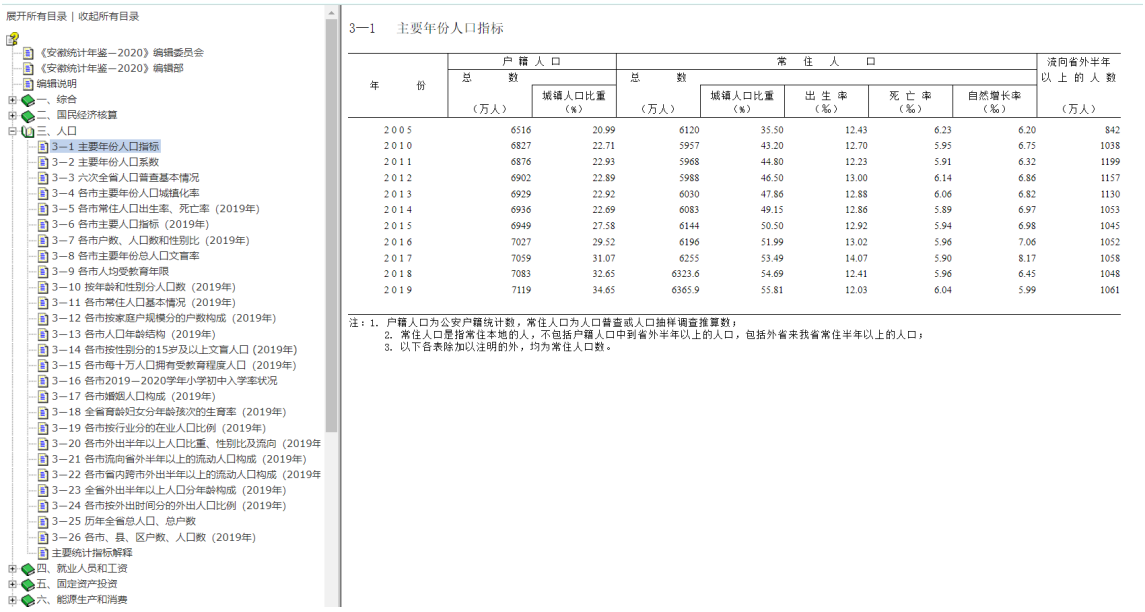
需求分析

近年来有关人口生育率的问题一直是社会的焦点问题，而安徽在人口问题上更是全国的舆论焦点。。。。

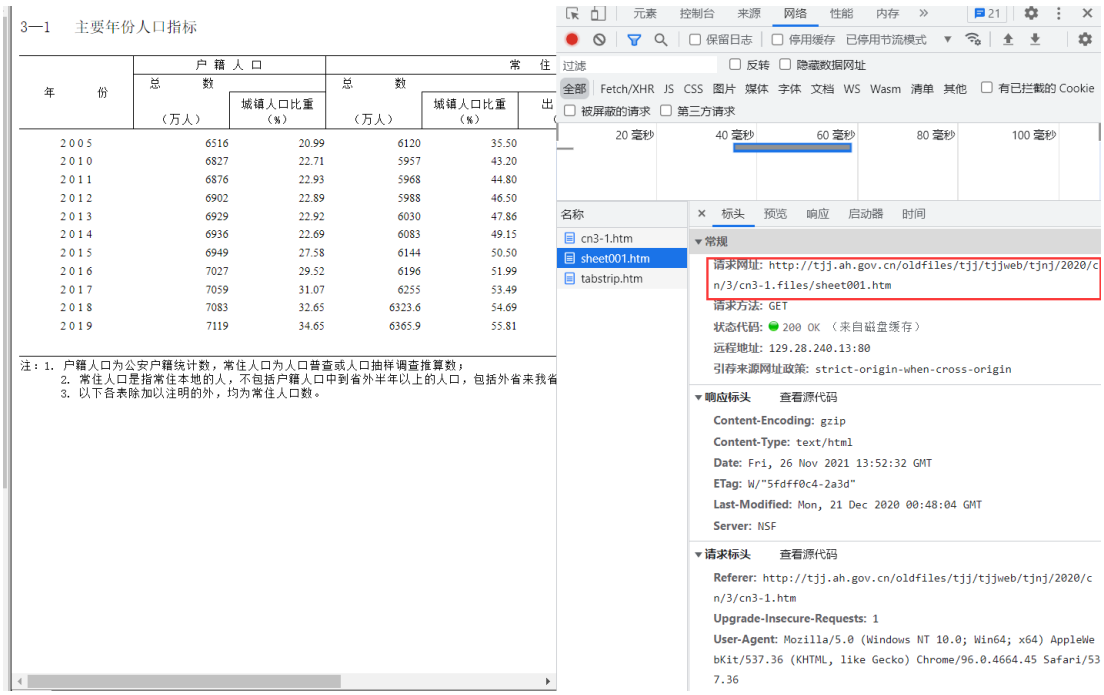
数据获取

数据形式分析

本次我们从安徽统计局获取人口数据，进行数据可视化呈现，我们访问安徽统计局查询数据，刚好安徽统计局已经对数据进行了集中展示，并编写了统计年鉴，我们查询安徽统计年鉴—2020中人口部分。

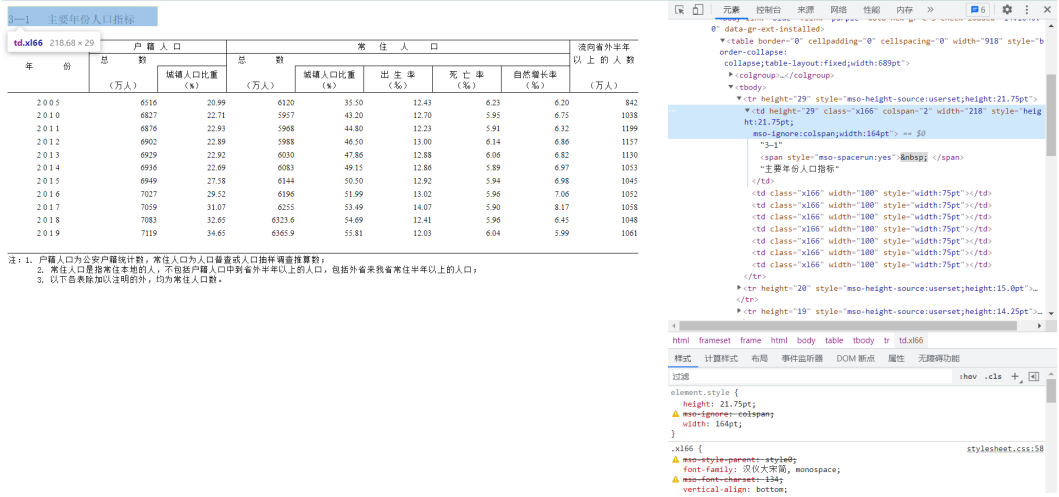


开始F12浏览器调试，刷新页面并监听，发现是通过局部刷新来加载页面，并且标明了请求网址，即数据保存的页面。同时还通过网址比对发现个链接中只有cn3-1，cn3-2部分不同且恰好对应表格编号

- 

进入数据实际所在页面，对元素进行解析

- 发现tr中height="29"属性对应的是表格名称



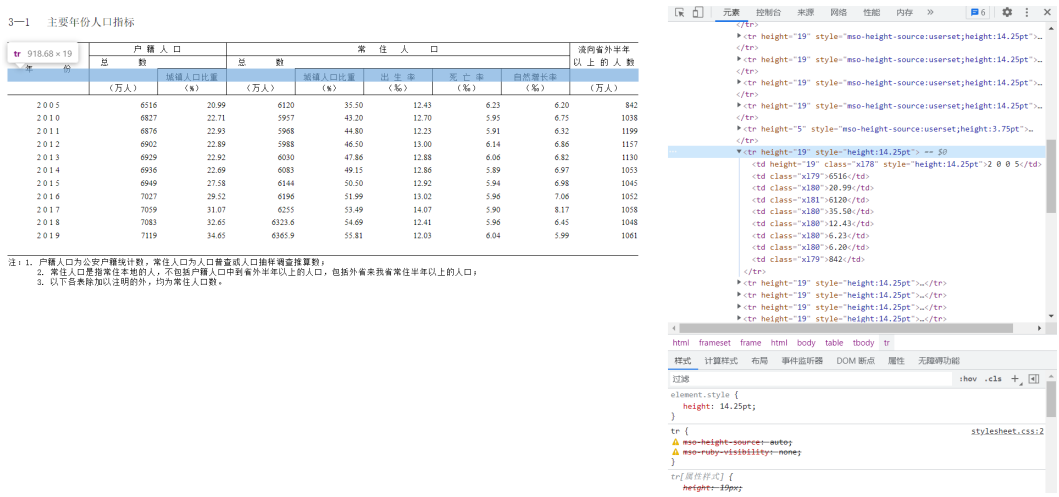
3—1　主要年份人口指标

td.xl66 218.68 × 29

年份	户籍人口 总数（万人）	户籍人口 城镇人口比重（%）	常住人口 总数（万人）	常住人口 城镇人口比重（%）	常住人口 出生率（‰）	常住人口 死亡率（‰）	常住人口 自然增长率（‰）	流向省外半年以上的人数（万人）
2005	6516	20.99	6120	35.50	12.43	6.23	6.20	842
2010	6827	22.71	5957	43.20	12.70	5.95	6.75	1038
2011	6876	22.93	5968	44.80	12.23	5.91	6.32	1199
2012	6902	22.89	5988	46.50	13.00	6.14	6.86	1157
2013	6929	22.92	6030	47.86	12.88	6.06	6.82	1130
2014	6936	22.69	6083	49.15	12.86	5.89	6.97	1053
2015	6949	27.58	6144	50.50	12.92	5.94	6.98	1045
2016	7027	29.52	6196	51.99	13.02	5.96	7.06	1052
2017	7059	31.07	6255	53.49	14.07	5.90	8.17	1058
2018	7083	32.65	6323.6	54.69	12.41	5.96	6.45	1048
2019	7119	34.65	6365.9	55.81	12.03	6.04	5.99	1061

注：1. 户籍人口为公安户籍统计数，常住人口为人口普查或人口抽样调查推算数；
2. 常住人口是指常住本地的人，不包括户籍人口中到省外半年以上的人口，包括外省来我省常住半年以上的人口；
3. 以下各表除加以注明的外，均为常住人口数。

- tr中height="29"属性对应的是各数据代表含义以及数据项



3—1　主要年份人口指标

tr 918.68 × 19

年份	户籍人口 总数（万人）	户籍人口 城镇人口比重（%）	常住人口 总数（万人）	常住人口 城镇人口比重（%）	常住人口 出生率（‰）	常住人口 死亡率（‰）	常住人口 自然增长率（‰）	流向省外半年以上的人数（万人）
2005	6516	20.99	6120	35.50	12.43	6.23	6.20	842
2010	6827	22.71	5957	43.20	12.70	5.95	6.75	1038
2011	6876	22.93	5968	44.80	12.23	5.91	6.32	1199
2012	6902	22.89	5988	46.50	13.00	6.14	6.86	1157
2013	6929	22.92	6030	47.86	12.88	6.06	6.82	1130
2014	6936	22.69	6083	49.15	12.86	5.89	6.97	1053
2015	6949	27.58	6144	50.50	12.92	5.94	6.98	1045
2016	7027	29.52	6196	51.99	13.02	5.96	7.06	1052
2017	7059	31.07	6255	53.49	14.07	5.90	8.17	1058
2018	7083	32.65	6323.6	54.69	12.41	5.96	6.45	1048
2019	7119	34.65	6365.9	55.81	12.03	6.04	5.99	1061

注：1. 户籍人口为公安户籍统计数，常住人口为人口普查或人口抽样调查推算数；
2. 常住人口是指常住本地的人，不包括户籍人口中到省外半年以上的人口，包括外省来我省常住半年以上的人口；
3. 以下各表除加以注明的外，均为常住人口数。

由此发现可以使用request进行数据请求，使用Xpath进行数据提取，再使用openpyxl进行数据保存

这里顺便说明一点，如果提取时使用utf-8编码格式会出现乱码，使用常见的gbk也会出现乱码，后来查看页面源代码才发现使用的竟然是我国上世纪八十年代设计的gb2312。

数据下载程序设计

```
from lxml import etree
from openpyxl import Workbook
import requests


def getdownload(url):
    # UA伪装
    header = {
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36 Edg/96.0.1054.29"
    }
    # 发起请求，带3参数
    r = requests.get(url=url, headers=header)
    # 将编码格式调成gb2312
    r.encoding = "gb2312"
    # 转成文本文件
    htm_text = r.text
    # 将文件本地保存下来，同时类型修改为html，编码格式改为utf-8 本文件仅做临时保存用，数据会储存在csv文件中
    with open("tmp.html", 'w', encoding='utf-8') as f:
        f.write(htm_text)


def getdata(id):
    # 转成标准的xml
    parser = etree.HTMLParser(encoding="utf-8")
    tree = etree.parse('tmp.html', parser=parser)
    # 解析出标题 大部分是 x166属性
    title = tree.xpath('/html/body/table/tr/td[@class="xl66"]/text()')
    # 小部分是x165属性
    if len(title) == 0:
        # 只有一个14号表格是x165属性
        classattribute = 'xl65' if id != 14 else 'xl67'
        title = tree.xpath(f'/html/body/table/tr/td[@class="{classattribute}"]/text()')
```

```
    # 返回的title是一个列表，转成一个str格式
    title = ''.join(title)
    print(title)
    # 获取表格中有效数据部分共有多少行
    elementnum = len(tree.xpath('/html/body/table/tr[@height="19"]'))
    ws = wb.create_sheet(title=f'{title}')  # 新建sheel插入到最后
    # 对所有行有效元素进行一一提取
    for i in range(1, elementnum + 1):
        tdele = tree.xpath(f'/html/body/table/tr[@height="19"][{i}]/td/text()')
        print(tdele)
        # 逐行添加数据
        ws.append(tdele)


if __name__ == '__main__':
    # 实例化表格
    wb = Workbook()
    for id in range(1, 27):
        url = f'http://tjj.ah.gov.cn/oldfiles/tjj/tjjweb/tjnj/2020/cn/3/cn3-{id}.files/sheet001.htm'
        getdownload(url)
        getdata(id)

    wb.save('ahpopulation.xlsx')
```

至此，得到了一个有26个sheel的xlsx表格,共有如下数据:

表格id	表格名	表格id	表格名
3-1	主要年份人口指标	3-14	各市按性别分的15岁及以上文盲人口 (2019年)
3-2	主要年份人口系数	3-15	各市每十万人口拥有受教育程度人口（2019年）
3-3	六次全省人口普查基本情况	3-16	各市2019—2020学年小学初中入学率状况
3-4	各市主要年份人口城镇化率	3-17	各市婚姻人口构成（2019年）
3-5	各市常住人口出生率、死亡率（2019年）	3-18	全省育龄妇女分年龄孩次的生育率（2019年）
3-6	各市主要人口指标（2019年）	3-19	各市按行业分的在业人口比例（2019年）
3-7	各市户数、人口数和性别比（2019年）	3-20	各市外出半年以上人口比重、性别比及流向（2019年）
3-8	各市主要年份总人口文盲率	3-21	各市流向省外半年以上的流动人口构成（2019年）
3-9	各市人均受教育年限	3-22	各市省内跨市外出半年以上的流动人口构成（2019年）
3-10	按年龄和性别分人口数（2019年）	3-23	全省外出半年以上人口分年龄构成（2019年）
3-11	各市常住人口基本情况（2019年）	3-24	各市按外出时间分的外出人口比例（2019年）
3-12	各市按家庭户规模分的户数构成（2019年）	3-25	历 年 全 省 总 人 口、总 户 数
3-13	各市人口年龄结构（2019年）	3-26	各市、县、区户数、人口数（2019年）

格式转换

在数据分析中我们一般使用csv文件，而且得到的数据中表头也需要修改，以及个别表格中“总计”二字出现各占一格情况，也需手动修改。此次也是用了try except 进行异常捕获，整个避免程序因某个表格出错而中断，在此次格式转换中采取读取时后立刻以行为单位写入csv。

```
import csv
from openpyxl import load_workbook


def getSheet(sheetName,header,before,after):
    '''表格转换函数,四参数：sheet名称,首行名称，前索引，后索引'''
    try:
        # 获取指定的表单
        ws = wb[sheetName]
        # 创建对应的.csv文件，从3或4开始切片，追加模式，utf-8编码,新建""一行
        with open(f'{sheetName[4:]}.csv', mode="a", encoding='utf-8', newline="") as f:
            # 创建filepencil，用来在问价上写入数据
            fp = csv.writer(f)
            # 写入表头
            fp.writerow(header)
            for index in range(before,after+1):
                rowitems = []
                for index,item in enumerate(ws[index]):
                    rowitem = item.value if index != 0 else item.value.replace(' ', '')
                    rowitems.append(rowitem)
                fp.writerow(rowitems)
    except Exception as e:
```

```
        print(e)


if __name__ == '__main__':
    # 加载表格文件
    wb = load_workbook('ahpopulation.xlsx')
    # # 主要年份人口指标
    # header31 = ['年份', '户籍人口总数（万人）', '城镇人口比重%', '常住人口总数（万人）', '城镇人口比重%',
    #  '出生率‰', '死亡率‰', '自然增长率‰', '流向省外半年以上的人数（万人）']
    # getSheet(sheetName='3-1主要年份人口指标', header=header31, before=5, after=15)

    # # 主要年份人口系数
    # header32 = ['年份', '少年儿童系数','', ''老年系数','', ''老少比','', ''少年儿童抚养系数',
    #             '老年抚养系数', '总抚养系数', '年龄中位数（岁）']
    # getSheet(sheetName='3-2主要年份人口系数', header=header32, before=3, after=13)

    # # 各市主要年份人口城镇化率
    # header32 = ['地区', '2010', '2015', '2017', '2018', '2019']
    # getSheet(sheetName='3-4各市主要年份人口城镇化率', header=header32, before=1, after=17)

    # # 3-5各市常住人口出生率、死亡率（2019年）
    # header35 = ['地区', '出生率（‰）', '死亡率（‰）', '自然增长率（‰）']
    # getSheet(sheetName='3-5各市常住人口出生率、死亡率（2019年）', header=header35, before=1, after=17)

    # # 3-7各市户数、人口数和性别比（2019年）
    # header37 = ['地区（万户）', '户数（万户）', '人口数', '男', '性别比（女=100）']
    # getSheet(sheetName='3-7各市户数、人口数和性别比（2019年）', header=header37, before=1, after=17)

    # # 3-10按年龄和性别分人口数（2019年）
    # header310 = ['年龄', '人口总数（人）', '男性总数', '女性总数', '占总人口比重（%）', '男性比重', '女性比重', '性别比
（女=100）']
    # getSheet(sheetName='3-10按年龄和性别分人口数（2019年）', header=header310, before=1, after=15)

    # # 3-12各市按家庭户规模分的户数构成（2019年）
    # header312 = ['地区', '家庭户规模（人/户）', '一人户', '二人户', '三人户', '四人户', '五人户', '六人及六人以上户（人/
户）']
    # getSheet(sheetName='3-12各市按家庭户规模分的户数构成（2019年）', header=header312, before=2, after=18)

    # # 3-21各市流向省外半年以上的流动人口构成（2019年）
    # header321 = ['地区', '合计', '江苏', '浙江', '上海', '广东', '北京', '福建', '山东', '天津', '河南', '河北', '新
疆', '辽宁', '湖北', '陕西', '流向其他省市']
    # getSheet(sheetName='3-21各市流向省外半年以上的流动人口构成（2019年）', header=header321, before=2, after=18)

    # # 3-25历 年 全 省 总 人 口、总 户 数
    # header325 = ['年份','总户数','合计总人口','男性人口','女性人口','性别比(女=100)','城镇人口','乡村人口']
    # getSheet(sheetName='3-25历 年 全 省 总 人 口、总 户 数', header=header325, before=2, after=17)

    # 3-26各市、县、区户数、人口数（2019年）
    header326 = ['地区', '总户数（万人）', '户籍人口（万人）', '男户籍人口', '女户籍人口', '性别比（女=100）', '城镇人口',
'常住人口']
    getSheet(sheetName='3-26各市、县、区户数、人口数（2019年）', header=header326, before=1, after=137)
```

可视化分析

```
# 导包
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
```

3—1主要年份人口指标

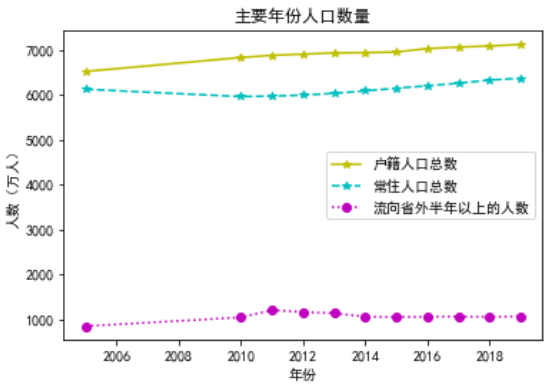
```
年份,户籍人口总数（万人）,城镇人口比重%,常住人口总数（万人）,城镇人口比重%,出生率%,死亡率%,自然增长率%,流向省外半年以上的人数（万人）
2005,6516,20.99,6120,35.50,12.43,6.23,6.20,842,
2010,6827,22.71,5957,43.20,12.70,5.95,6.75,1038,
2011,6876,22.93,5968,44.80,12.23,5.91,6.32,1199,
2012,6902,22.89,5988,46.50,13.00,6.14,6.86,1157,
2013,6929,22.92,6030,47.86,12.88,6.06,6.82,1130,
2014,6936,22.69,6083,49.15,12.86,5.89,6.97,1053,
2015,6949,27.58,6144,50.50,12.92,5.94,6.98,1045,
2016,7027,29.52,6196,51.99,13.02,5.96,7.06,1052,
2017,7059,31.07,6255,53.49,14.07,5.90,8.17,1058,
2018,7083,32.65,6323.6,54.69,12.41,5.96,6.45,1048,
2019,7119,34.65,6365.9,55.81,12.03,6.04,5.99,1061,
```

```
# 3-1主要年份人口指标
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('主要年份人口指标.csv',index_col=False )
df.columns = ['年份', '户籍人口总数（万人）', '城镇人口比重%', '常住人口总数（万人）', '城镇人口比重%', '出生率‰', '死亡率‰', '自然增长率‰', '流向省外半年以上的人数（万人）']
df
```

	年份	户籍人口总数（万人）	城镇人口比重%	常住人口总数（万人）	城镇人口比重%	出生率‰	死亡率‰	自然增长率‰	流向省外半年以上的人数（万人）
0	2005	6516	20.99	6120.0	35.50	12.43	6.23	6.20	842
1	2010	6827	22.71	5957.0	43.20	12.70	5.95	6.75	1038
2	2011	6876	22.93	5968.0	44.80	12.23	5.91	6.32	1199
3	2012	6902	22.89	5988.0	46.50	13.00	6.14	6.86	1157
4	2013	6929	22.92	6030.0	47.86	12.88	6.06	6.82	1130
5	2014	6936	22.69	6083.0	49.15	12.86	5.89	6.97	1053
6	2015	6949	27.58	6144.0	50.50	12.92	5.94	6.98	1045
7	2016	7027	29.52	6196.0	51.99	13.02	5.96	7.06	1052
8	2017	7059	31.07	6255.0	53.49	14.07	5.90	8.17	1058
9	2018	7083	32.65	6323.6	54.69	12.41	5.96	6.45	1048
10	2019	7119	34.65	6365.9	55.81	12.03	6.04	5.99	1061

```
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
x = df['年份']

y1 = df['户籍人口总数（万人）']
y2 = df['常住人口总数（万人）']
y3 = df['流向省外半年以上的人数（万人）']

fig, ax = plt.subplots()  # Create a figure and an axes.

ax.plot(x, y1, 'y*-', label='户籍人口总数')
ax.plot(x, y2, 'c*--', label='常住人口总数')
ax.plot(x, y3, 'mo:', label='流向省外半年以上的人数')
ax.set_xlabel('年份')  # Add an x-label to the axes.
ax.set_ylabel('人数（万人）')  # Add a y-label to the axes.
ax.set_title("主要年份人口数量")  # Add a title to the axes.
ax.legend()  # Add a legend.
plt.show()
```

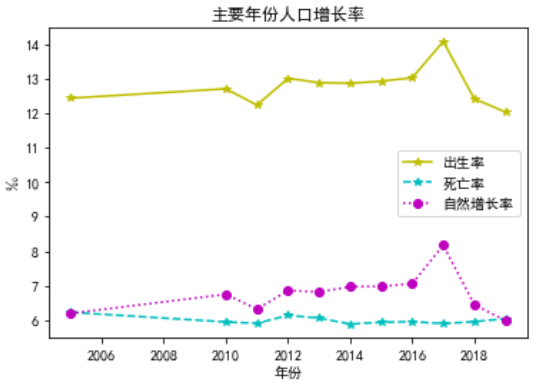
```
<Figure size 1296x720 with 0 Axes>
```



3—2主要年份人口系数

```
年份,少年儿童系数,", 老年系数",", 老少比",", 少年儿童抚养系数",老年抚养系数,总抚养系数,年龄中位数（岁）
2005,23.07,10.08,43.69,34.51,15.08,49.59,34.32,
2010,17.77,10.23,57.57,24.68,14.21,38.89,36.36,
2011,18.59,11.41,61.36,26.56,16.30,42.86,38.83,
2012,18.35,12.08,65.83,26.37,17.36,43.73,39.79,
2013,18.51,12.24,66.15,26.72,17.68,44.40,40.12,
2014,18.68,11.71,62.72,26.83,16.83,43.66,39.42,
2015,18.21,11.73,64.41,25.99,16.74,42.73,39.68,
2016,18.31,12.00,65.54,26.27,17.22,43.49,38.63,
2017,18.60,12.38,66.56,26.95,17.94,44.89,39.62,
2018,18.85,12.97,68.81,27.65,19.02,46.67,40.08,
2019,18.91,13.93,73.66,28.16,20.74,48.90,41.91,
```

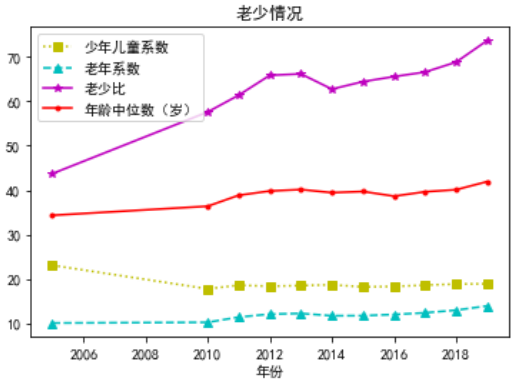
```
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
x = df['年份']
y1 = df['出生率‰']
y2 = df['死亡率‰']
y3 = df['自然增长率‰']
fig, ax = plt.subplots()  # Create a figure and an axes.
ax.plot(x, y1, 'y*-', label='出生率')  # Plot some data on the axes.
ax.plot(x, y2, 'c*--', label='死亡率')  # Plot more data on the axes...
ax.plot(x, y3, 'mo:', label='自然增长率')  # ... and some more.
ax.set_xlabel('年份')  # Add an x-label to the axes.
ax.set_ylabel('‰')  # Add a y-label to the axes.
ax.set_title("主要年份人口增长率")  # Add a title to the axes.
ax.legend()  # Add a legend.
```




```
#3-2主要年份人口系数
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('主要年份人口系数.csv',index_col=False )
df.columns = ['年份', '少年儿童系数', '老年系数', '老少比', '少年儿童抚养系数', '老年抚养系数', '总抚养系数', '年龄中位数（岁）']
df
```

	年份	少年儿童系数	老年系数	老少比	少年儿童抚养系数	老年抚养系数	总抚养系数	年龄中位数（岁）
0	2005	23.07	10.08	43.69	34.51	15.08	49.59	34.32
1	2010	17.77	10.23	57.57	24.68	14.21	38.89	36.36
2	2011	18.59	11.41	61.36	26.56	16.30	42.86	38.83
3	2012	18.35	12.08	65.83	26.37	17.36	43.73	39.79
4	2013	18.51	12.24	66.15	26.72	17.68	44.40	40.12
5	2014	18.68	11.71	62.72	26.83	16.83	43.66	39.42
6	2015	18.21	11.73	64.41	25.99	16.74	42.73	39.68
7	2016	18.31	12.00	65.54	26.27	17.22	43.49	38.63
8	2017	18.60	12.38	66.56	26.95	17.94	44.89	39.62
9	2018	18.85	12.97	68.81	27.65	19.02	46.67	40.08
10	2019	18.91	13.93	73.66	28.16	20.74	48.90	41.91

```
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
x = df['年份']
y1 = df['少年儿童系数']
y2 = df['老年系数']
y3 = df['老少比']
y4 = df['年龄中位数（岁）']
fig, ax = plt.subplots()  # Create a figure and an axes.
ax.plot(x, y1, 'ys:', label='少年儿童系数')  # Plot some data on the axes.
ax.plot(x, y2, 'c^--', label='老年系数')  # Plot more data on the axes...
ax.plot(x, y3, 'm*-', label='老少比')  # ... and some more.
ax.plot(x, y4, 'r.-', label='年龄中位数（岁）')  # Plot some data on the axes.
ax.set_xlabel('年份')  # Add an x-label to the axes.
ax.set_ylabel('')  # Add a y-label to the axes.
ax.set_title("老少情况")  # Add a title to the axes.
ax.legend()  # Add a legend.
```



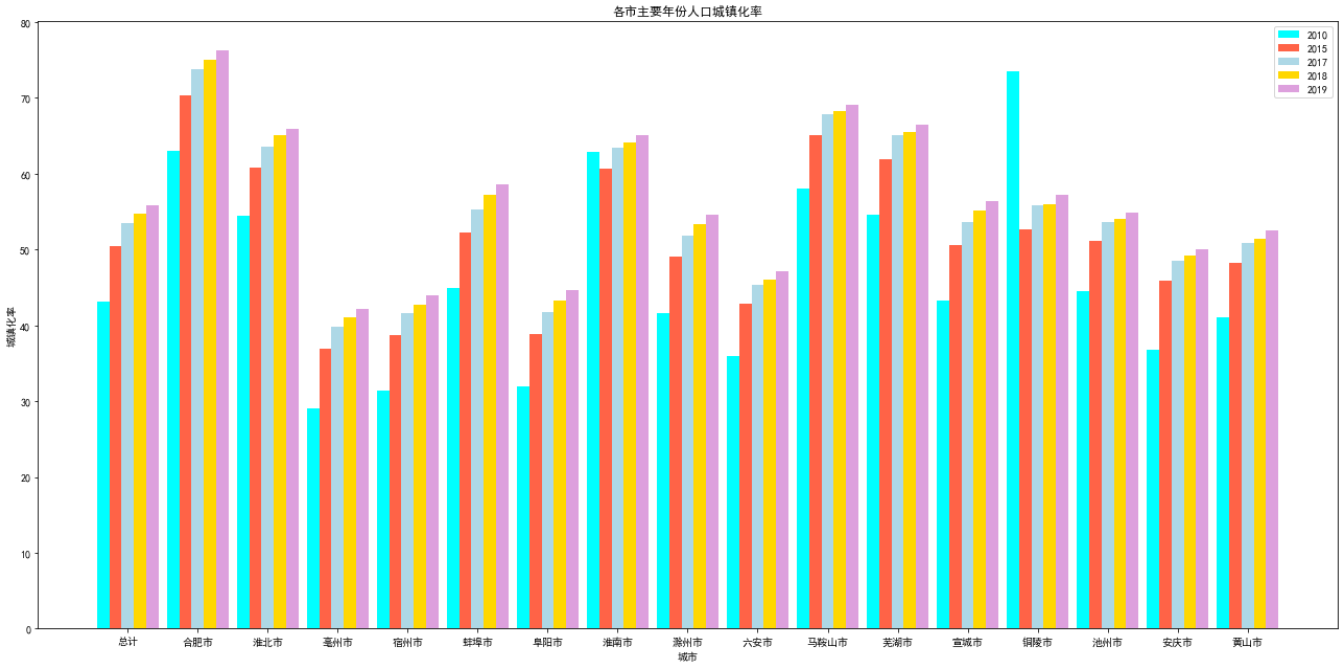
3—4各市主要年份人口城镇化率

```
地区,2010,2015,2017,2018,2019
总计,43.20,50.50,53.49,54.69,55.81
合肥市,63.00,70.40,73.75,74.97,76.33
淮北市,54.50,60.76,63.61,65.11,65.88
亳州市,29.10,36.96,39.77,41.01,42.22
宿州市,31.40,38.73,41.56,42.74,43.96
蚌埠市,45.00,52.22,55.31,57.22,58.58
阜阳市,31.90,38.81,41.75,43.29,44.62
淮南市,62.90,60.67,63.46,64.11,65.04
滁州市,41.60,49.02,51.89,53.42,54.54
六安市,35.90,42.81,45.41,46.08,47.09
马鞍山市,58.00,65.15,67.89,68.25,69.12
芜湖市,54.60,61.96,65.05,65.54,66.41
宣城市,43.30,50.64,53.69,55.21,56.33
铜陵市,73.50,52.73,55.79,55.99,57.16
池州市,44.50,51.11,53.67,54.10,54.92
安庆市,36.80,45.87,48.57,49.22,49.98
黄山市,41.10,48.28,50.90,51.46,52.49
```

```
#3-4各市主要年份人口城镇化率
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('各市主要年份人口城镇化率.csv',index_col=False )
df.columns =  ['地区', '2010', '2015', '2017', '2018', '2019']
df
```

	地区	2010	2015	2017	2018	2019
0	总计	43.2	50.50	53.49	54.69	55.81
1	合肥市	63.0	70.40	73.75	74.97	76.33
2	淮北市	54.5	60.76	63.61	65.11	65.88
3	亳州市	29.1	36.96	39.77	41.01	42.22
4	宿州市	31.4	38.73	41.56	42.74	43.96
5	蚌埠市	45.0	52.22	55.31	57.22	58.58
6	阜阳市	31.9	38.81	41.75	43.29	44.62
7	淮南市	62.9	60.67	63.46	64.11	65.04
8	滁州市	41.6	49.02	51.89	53.42	54.54
9	六安市	35.9	42.81	45.41	46.08	47.09
10	马鞍山市	58.0	65.15	67.89	68.25	69.12
11	芜湖市	54.6	61.96	65.05	65.54	66.41
12	宣城市	43.3	50.64	53.69	55.21	56.33
13	铜陵市	73.5	52.73	55.79	55.99	57.16
14	池州市	44.5	51.11	53.67	54.10	54.92
15	安庆市	36.8	45.87	48.57	49.22	49.98
16	黄山市	41.1	48.28	50.90	51.46	52.49

```
labels = ['总计', '合肥市', '淮北市', '亳州市', '宿州市', '蚌埠市', '阜阳市', '淮南市', '滁州市', '六安市', '马鞍山市', '芜湖市', '宣城市', '铜陵市', '池州市', '安庆市', '黄山市']
width = 0.35  # the width of the bars

x = np.arange(len(labels))  # the label locations

fig, ax = plt.subplots(figsize=(18, 9))
rects1 = ax.bar(x-width ,df['2010'] , width/2, label='2010',color='cyan')
rects2 = ax.bar(x-width/2, df['2015'], width/2, label='2015',color='tomato')
rects3 = ax.bar(x, df['2017'], width/2, label='2017',color='lightblue')
rects4 = ax.bar(x+width/2, df['2018'], width/2, label='2018',color='gold')
rects5 = ax.bar(x+width, df['2019'], width/2, label='2019',color='plum')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_xlabel('城市')
ax.set_ylabel('城镇化率')
ax.set_title('各市主要年份人口城镇化率')
plt.xticks(x, labels)
ax.legend()


fig.tight_layout()

plt.show()
```



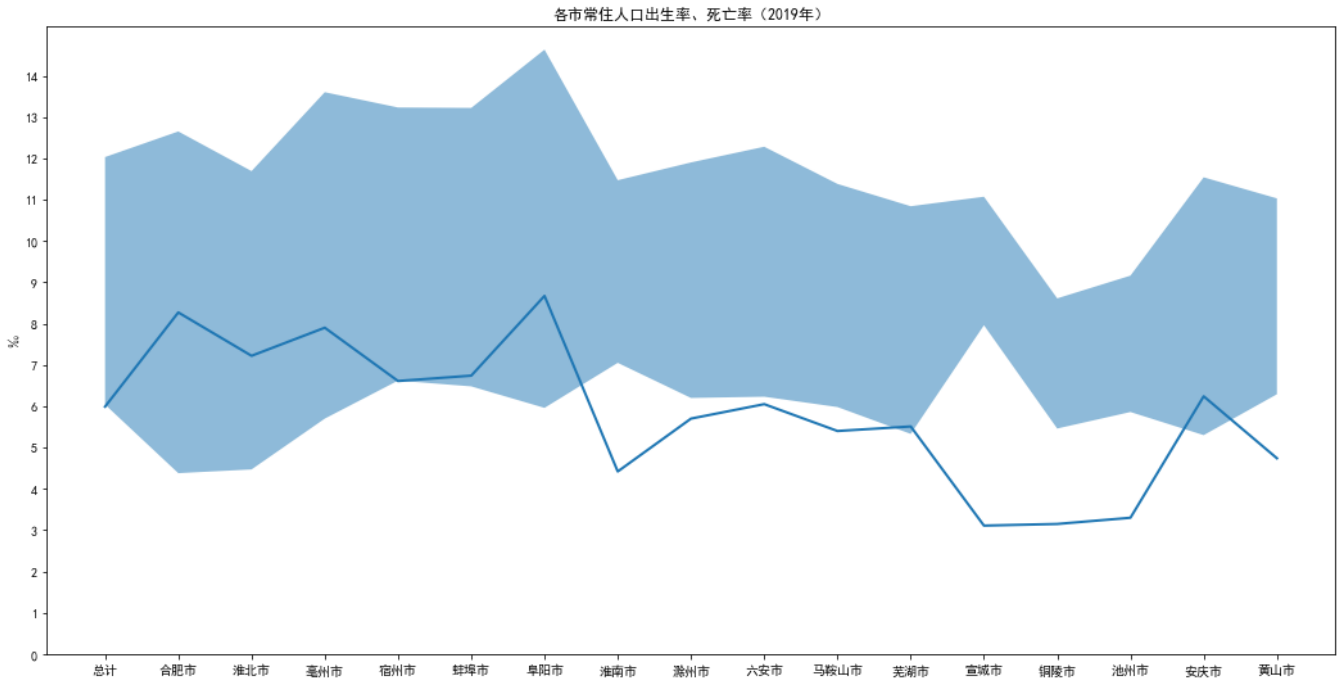
3—5各市常住人口出生率、死亡率（2019年）

-
```csv
地区,出生率（‰）,死亡率（‰）,自然增长率（‰）
总计,12.03,6.04,5.99
合肥市,12.65,4.38,8.27
淮北市,11.69,4.47,7.22
亳州市,13.60,5.70,7.90
宿州市,13.23,6.62,6.61
蚌埠市,13.22,6.48,6.74
阜阳市,14.63,5.96,8.67
淮南市,11.47,7.05,4.42
滁州市,11.90,6.20,5.70
六安市,12.28,6.23,6.05
马鞍山市,11.38,5.98,5.40
芜湖市,10.84,5.33,5.51
宣城市,11.07,7.96,3.11
铜陵市,8.61,5.46,3.15
池州市,9.16,5.86,3.30
安庆市,11.54,5.30,6.24
黄山市,11.03,6.29,4.74
```

```python
# 3-5各市常住人口出生率、死亡率（2019年）
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('各市常住人口出生率、死亡率（2019年）.csv',index_col=False )
df.columns =  ['地区', '出生率（‰）', '死亡率（‰）', '自然增长率（‰）']
df
```

	地区	出生率（‰）	死亡率（‰）	自然增长率（‰）
0	总计	12.03	6.04	5.99
1	合肥市	12.65	4.38	8.27
2	淮北市	11.69	4.47	7.22
3	亳州市	13.60	5.70	7.90
4	宿州市	13.23	6.62	6.61
5	蚌埠市	13.22	6.48	6.74
6	阜阳市	14.63	5.96	8.67
7	淮南市	11.47	7.05	4.42
8	滁州市	11.90	6.20	5.70
9	六安市	12.28	6.23	6.05
10	马鞍山市	11.38	5.98	5.40
11	芜湖市	10.84	5.33	5.51
12	宣城市	11.07	7.96	3.11
13	铜陵市	8.61	5.46	3.15
14	池州市	9.16	5.86	3.30
15	安庆市	11.54	5.30	6.24
16	黄山市	11.03	6.29	4.74

```
# make data
np.random.seed(1)
x = np.arange(len(labels))  # the label locations


# plot
fig, ax = plt.subplots(figsize=(18, 9))

ax.fill_between(x, df['出生率（‰）'], df['死亡率（‰）'], alpha=.5, linewidth=0)
ax.plot(x, df['自然增长率（‰）'], linewidth=2,label='自然增长率')
plt.ylabel('‰')
plt.xticks(x, df['地区'])
plt.yticks(np.arange(15))
plt.title("各市常住人口出生率、死亡率（2019年）")
plt.show()
```



3—7各市户数、人口数和性别比（2019年）

```
地区（万户）,户数（万户）,人口数,男,性别比（女=100）
总计,2176.29,7119.37,3694.33,107.86
合肥市,255.33,770.44,395.24,105.34
淮北市,73.04,218.72,112.25,105.43
亳州市,179.68,662.99,347.91,110.42
宿州市,197.85,658.27,342.28,108.32
蚌埠市,113.40,386.30,200.09,107.45
阜阳市,285.36,1077.28,561.09,108.70
淮南市,124.39,390.82,204.89,110.20
滁州市,140.69,455.35,236.31,107.88
六安市,189.85,591.07,311.86,111.69
马鞍山市,74.60,229.14,117.85,105.90
芜湖市,130.94,389.84,200.87,106.29
宣城市,98.95,278.77,143.70,106.38
铜陵市,54.13,170.58,87.54,105.41
池州市,51.79,162.29,82.94,104.53
安庆市,155.39,528.58,273.36,107.11
黄山市,50.91,148.92,76.16,104.66
```

```
#各市户数、人口数和性别比（2019年）
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('各市户数、人口数和性别比（2019年）.csv',index_col=False )
df.columns = ['地区（万户）', '户数（万户）', '人口数', '男', '性别比（女=100）']
df
```

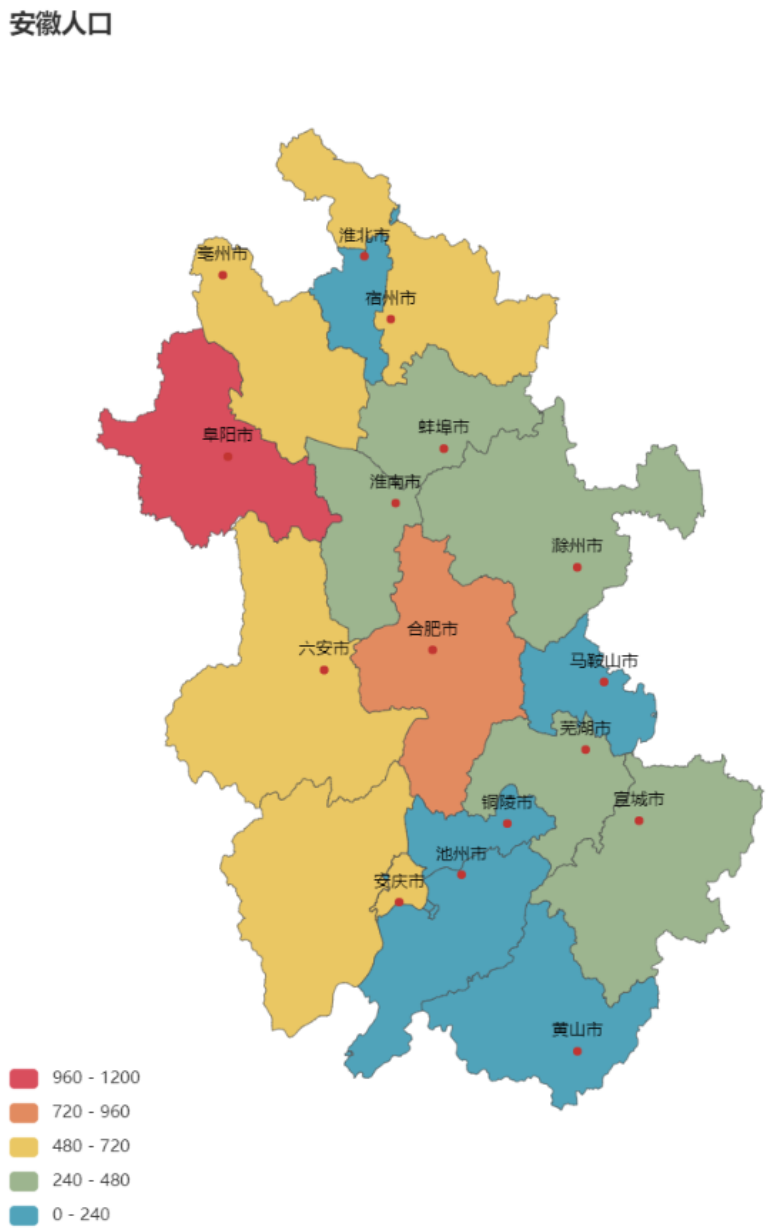
	地区（万户）	户数（万户）	人口数	男	性别比（女=100）
0	总计	2176.29	7119.37	3694.33	107.86
1	合肥市	255.33	770.44	395.24	105.34
2	淮北市	73.04	218.72	112.25	105.43
3	亳州市	179.68	662.99	347.91	110.42
4	宿州市	197.85	658.27	342.28	108.32
5	蚌埠市	113.40	386.30	200.09	107.45
6	阜阳市	285.36	1077.28	561.09	108.70
7	淮南市	124.39	390.82	204.89	110.20
8	滁州市	140.69	455.35	236.31	107.88
9	六安市	189.85	591.07	311.86	111.69
10	马鞍山市	74.60	229.14	117.85	105.90
11	芜湖市	130.94	389.84	200.87	106.29
12	宣城市	98.95	278.77	143.70	106.38
13	铜陵市	54.13	170.58	87.54	105.41
14	池州市	51.79	162.29	82.94	104.53
15	安庆市	155.39	528.58	273.36	107.11
16	黄山市	50.91	148.92	76.16	104.66

3—10按年龄和性别分人口数（2019年）

```
年龄,人口总数（人）,男性总数,女性总数,占总人口比重（%）,男性比重,女性比重,性别比（女=100）
总计,636590,319757,316832,100.00,50.23,49.77,100.96
0—4,37002,18590,18412,5.81,2.92,2.89,100.96
5—9,42670,22754,19916,6.70,3.57,3.13,114.25
10—14,40707,22086,18621,6.39,3.47,2.93,118.61
15—19,32634,17651,14982,5.13,2.77,2.35,117.82
20—24,29589,15765,13824,4.65,2.48,2.17,114.04
25—29,40166,21369,18797,6.31,3.36,2.95,113.69
30—34,40878,19943,20934,6.42,3.13,3.29,95.27
35—39,35918,16932,18986,5.64,2.66,2.98,89.18
40—44,43077,20781,22294,6.77,3.26,3.50,93.21
45—49,61939,30675,31264,9.73,4.82,4.91,98.11
50—54,67674,33334,34338,10.63,5.24,5.39,97.08
55—59,47141,22994,24147,7.41,3.61,3.79,95.22
60—64,28519,14493,14026,4.48,2.28,2.20,103.33
65+,88677,43210,45470,13.93,6.79,7.14,95.03
```

```
from pyecharts.charts import Map
from pyecharts import options as opts
import pandas as pd


# print(df) 可以检验一下数据是否导入正确
area = df['地区（万户）']
value = df['人口数']
k = list(df['地区（万户）'])[1:]

v = list(df['人口数'])[1:]

data_pair = [(k[i], v[i]) for i in range(len(k))]
print(data_pair)

map = Map(init_opts=opts.InitOpts(width="600px", height="860px"))  # 创建地图，其中括号内可以调整大小，也可以修改主题颜色。
map.add("安徽人口", data_pair, maptype="安徽")  # 添加安徽地图
map.set_global_opts(  # 设置全局配置项#
    title_opts=opts.TitleOpts(title="安徽人口"),  # 添加标题
    visualmap_opts=opts.VisualMapOpts(max_=1200, is_piecewise=True),  # 最大数据范围 并且使用分段
    legend_opts=opts.LegendOpts(is_show=False),  # 是否显示图例
)
map.render('安徽人口.html')   # map.render_notebook()直接在notebook中显示# map.render('map1.html') 将地图以html形式保存在工作目录下
```

```
[('合肥市', 770.44), ('淮北市', 218.72), ('亳州市', 662.99), ('宿州市', 658.27), ('蚌埠市', 386.3), ('阜阳市', 1077.28), ('淮南市', 390.82), ('滁州市', 455.35), ('六安市', 591.07), ('马鞍山市', 229.14), ('芜湖市', 389.84), ('宣城市', 278.77), ('铜陵市', 170.58), ('池州市', 162.29), ('安庆市', 528.58), ('黄山市', 148.92)]
```

- 

[安徽人口.html]

3—12各市按家庭户规模分的户数构成（2019年）

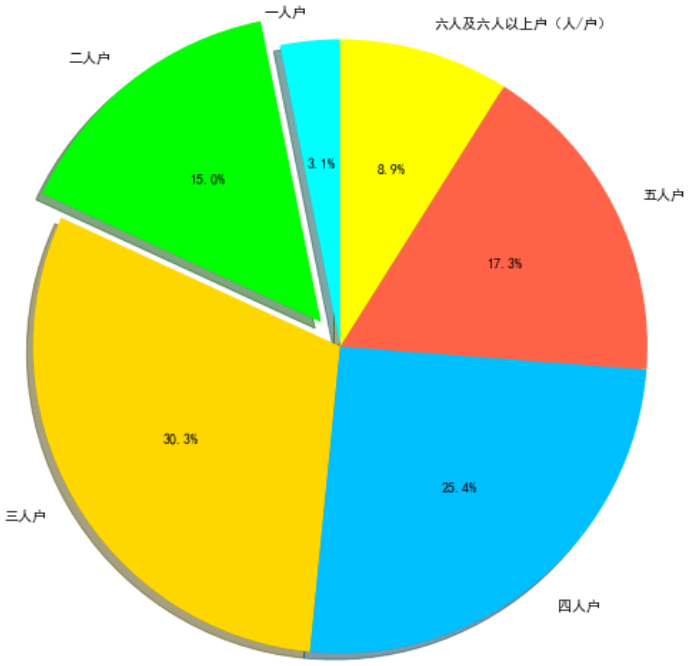
-
```
地区,家庭户规模（人/户）,一人户,二人户,三人户,四人户,五人户,六人及六人以上户（人/户）
总计,3.04,14.45,29.26,24.49,16.66,8.63,6.50
合肥市,2.86,15.74,29.74,27.11,16.35,6.69,4.38,
淮北市,3.05,12.62,29.99,25.95,17.50,7.70,6.27,
亳州市,3.53,12.23,24.71,17.80,22.07,11.55,11.64,
宿州市,3.12,14.40,30.03,20.63,19.40,8.46,7.08,
蚌埠市,3.09,14.21,29.20,23.70,17.32,8.47,7.11,
阜阳市,3.44,12.67,24.64,19.53,21.60,10.70,10.86,
淮南市,2.90,16.97,32.23,23.02,15.42,6.74,5.61,
滁州市,3.17,12.43,28.28,24.50,16.80,10.16,7.82,
六安市,3.22,13.61,26.88,22.22,18.18,10.58,8.53,
马鞍山市,2.81,16.45,33.79,25.55,12.27,7.30,4.65,
芜湖市,2.92,15.01,29.03,28.00,14.93,8.41,4.63,
宣城市,2.69,18.68,35.71,23.15,11.76,7.15,3.55,
铜陵市,2.77,14.35,31.84,31.31,14.10,5.65,2.74,
池州市,2.89,15.23,31.65,25.58,14.80,8.05,4.70,
安庆市,3.32,11.17,23.99,25.68,19.12,11.40,8.63,
黄山市,2.81,16.29,31.77,27.34,12.53,7.72,4.36,
```

```
#各市按家庭户规模分的户数构成（2019年）
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('各市按家庭户规模分的户数构成（2019年）.csv',index_col=False )
df.columns = ['地区', '家庭户规模（人/户）', '一人户', '二人户', '三人户', '四人户', '五人户', '六人及六人以上户（人/户）']
df
```

	地区	家庭户规模（人/户）	一人户	二人户	三人户	四人户	五人户	六人及六人以上户（人/户）
0	总计	3.04	14.45	29.26	24.49	16.66	8.63	6.50
1	合肥市	2.86	15.74	29.74	27.11	16.35	6.69	4.38
2	淮北市	3.05	12.62	29.99	25.95	17.50	7.70	6.27
3	亳州市	3.53	12.23	24.71	17.80	22.07	11.55	11.64
4	宿州市	3.12	14.40	30.03	20.63	19.40	8.46	7.08
5	蚌埠市	3.09	14.21	29.20	23.70	17.32	8.47	7.11
6	阜阳市	3.44	12.67	24.64	19.53	21.60	10.70	10.86
7	淮南市	2.90	16.97	32.23	23.02	15.42	6.74	5.61
8	滁州市	3.17	12.43	28.28	24.50	16.80	10.16	7.82
9	六安市	3.22	13.61	26.88	22.22	18.18	10.58	8.53
10	马鞍山市	2.81	16.45	33.79	25.55	12.27	7.30	4.65
11	芜湖市	2.92	15.01	29.03	28.00	14.93	8.41	4.63
12	宣城市	2.69	18.68	35.71	23.15	11.76	7.15	3.55
13	铜陵市	2.77	14.35	31.84	31.31	14.10	5.65	2.74
14	池州市	2.89	15.23	31.65	25.58	14.80	8.05	4.70
15	安庆市	3.32	11.17	23.99	25.68	19.12	11.40	8.63
16	黄山市	2.81	16.29	31.77	27.34	12.53	7.72	4.36

```
data=pd.DataFrame(df)
header = [ '一人户', '二人户', '三人户', '四人户', '五人户', '六人及六人以上户（人/户）']
data.loc[0].values[1:-1] # 只选用第一行全省数据
```

```
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = header
sizes = data.loc[0].values[1:-1]
explode = (0, 0.1, 0, 0,0,0)  # only "explode" the 2nd slice (i.e. 'Hogs')
colors = ['cyan','lime','gold','deepskyblue','tomato','yellow']
fig1, ax1 = plt.subplots(figsize=(9, 9))
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90,colors=colors)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



3—21各市流向省外半年以上的流动人口构成（2019年）

```
地区,合计,江苏,浙江,上海,广东,北京,福建,山东,天津,河南,河北,新疆,辽宁,湖北,陕西,流向其他省市
总计,95400,30.02,25.72,21.79,4.35,3.31,2.12,1.66,1.28,0.98,1.08,0.62,0.69,0.87,0.89,4.62
合肥市,3062,33.36,14.78,24.99,2.68,7.38,1.96,1.15,2.15,0.93,1.03,2.33,0.53,0.53,1.12,5.1
淮北市,1703,31.53,25.33,15.74,4.63,4.58,1.28,3.52,1,2.46,0.56,0.56,0.28,0.73,1.12,6.75
亳州市,7452,34.03,28.79,12.46,5.35,1.26,2.09,2.88,1.43,1.11,2.24,0.97,0.87,0.33,0.61,5.61
宿州市,8511,40.16,25.81,10.74,2.39,2.93,2.18,1.54,1.63,1.15,1.07,0.83,0.25,0.56,4.84,3.96
蚌埠市,7390,29.5,31.9,22.53,4.58,2.05,3.05,0.72,0.86,0.41,0.45,0.19,0.17,0.51,0.28,2.83
阜阳市,14878,17.06,38.21,16.88,6.51,3.16,1.84,2.3,2.04,1.9,2.14,1.57,1.17,1.36,0.47,3.41
淮南市,8287,26.05,12.94,47.46,3.51,1.25,2.56,0.52,0.42,0.68,0.31,0.25,0.19,0.49,0.56,2.83
滁州市,6113,52.75,17.69,20.09,2.17,1.27,0.89,0.95,0.28,0.33,0.68,0.11,0.39,0.26,0.16,2
六安市,7411,47.31,18.49,23.25,2.89,1.78,0.59,0.74,0.41,0.76,0.38,0.23,0.1,0.65,0.17,2.25
马鞍山市,4429,45.81,8.75,12.35,3.29,10.08,1.27,2.02,0.62,1.14,2,0.71,0.64,1.68,0.95,8.75
芜湖市,4825,28.35,8.05,30.78,5.6,10.04,1.42,1.82,2.9,0.57,1.18,0.2,0.43,1.24,0.91,6.53
宣城市,5577,19.48,29.23,38.07,2.79,1.57,1.19,1.24,0.48,0.44,0.27,0.07,0.15,0.89,0.44,3.73
铜陵市,2819,41.77,20.92,14.41,4.93,3.74,0.88,0.54,0.57,0.51,0.71,0.13,0.47,1.28,0.67,8.43
池州市,3343,22.02,32.97,22.22,4.01,2.33,2.05,1.76,0.88,0.83,0.63,0.23,1.11,0.68,0.71,7.6
安庆市,6658,13.67,26.01,16.47,7.54,5.23,6.67,3.29,3.1,1.31,1.27,0.41,3.14,1.81,0.47,9.61
黄山市,3783,12.28,56.32,16.68,3.42,1.53,2.16,1.76,0.3,0.25,0.13,0.05,0.2,0.75,0.2,4.02
```

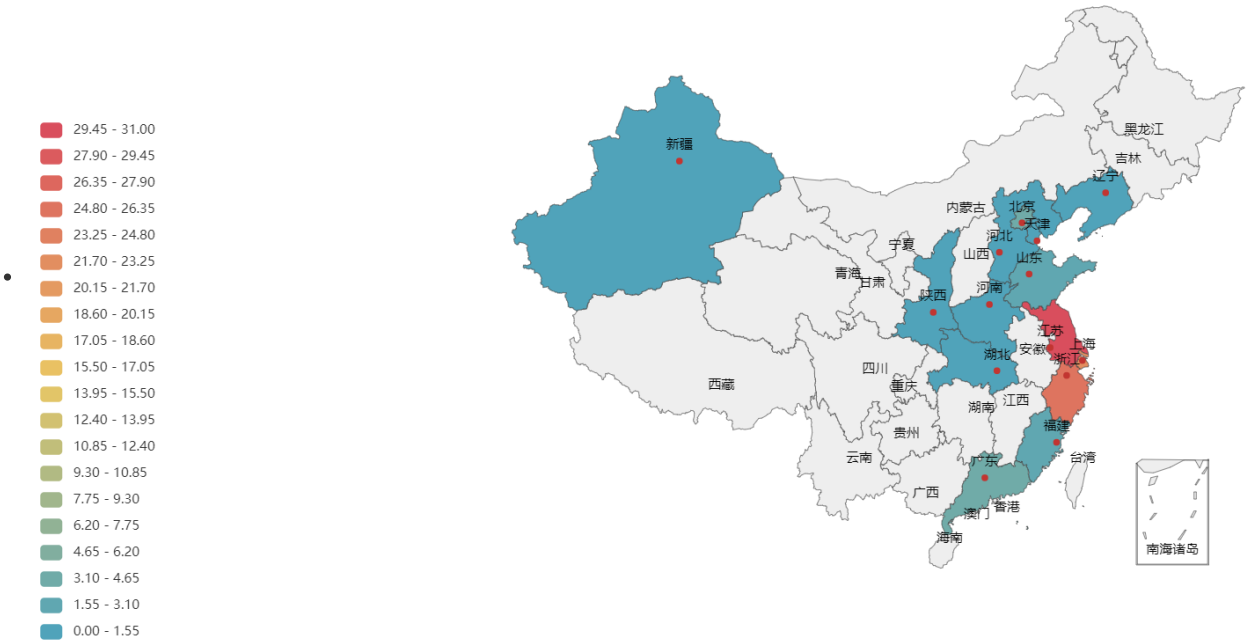
```
#各市流向省外半年以上的流动人口构成（2019年）
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('各市流向省外半年以上的流动人口构成（2019年）.csv',index_col=False )
df.columns = ['地区', '合计', '江苏', '浙江', '上海', '广东', '北京', '福建', '山东', '天津', '河南', '河北', '新疆', '辽宁', '湖北', '陕西',
'流向其他省市']
df
```

	地区	合计	江苏	浙江	上海	广东	北京	福建	山东	天津	河南	河北	新疆	辽宁	湖北	陕西	流向其他省市
0	总计	95400	30.02	25.72	21.79	4.35	3.31	2.12	1.66	1.28	0.98	1.08	0.62	0.69	0.87	0.89	4.62
1	合肥市	3062	33.36	14.78	24.99	2.68	7.38	1.96	1.15	2.15	0.93	1.03	2.33	0.53	0.53	1.12	5.10
2	淮北市	1703	31.53	25.33	15.74	4.63	4.58	1.28	3.52	1.00	2.46	0.56	0.56	0.28	0.73	1.12	6.75
3	亳州市	7452	34.03	28.79	12.46	5.35	1.26	2.09	2.88	1.43	1.11	2.24	0.97	0.87	0.33	0.61	5.61
4	宿州市	8511	40.16	25.81	10.74	2.39	2.93	2.18	1.54	1.63	1.15	1.07	0.83	0.25	0.56	4.84	3.96
5	蚌埠市	7390	29.50	31.90	22.53	4.58	2.05	3.05	0.72	0.86	0.41	0.45	0.19	0.17	0.51	0.28	2.83
6	阜阳市	14878	17.06	38.21	16.88	6.51	3.16	1.84	2.30	2.04	1.90	2.14	1.57	1.17	1.36	0.47	3.41
7	淮南市	8287	26.05	12.94	47.46	3.51	1.25	2.56	0.52	0.42	0.68	0.31	0.25	0.19	0.49	0.56	2.83
8	滁州市	6113	52.75	17.69	20.09	2.17	1.27	0.89	0.95	0.28	0.33	0.68	0.11	0.39	0.26	0.16	2.00
9	六安市	7411	47.31	18.49	23.25	2.89	1.78	0.59	0.74	0.41	0.76	0.38	0.23	0.10	0.65	0.17	2.25
10	马鞍山市	4429	45.81	8.75	12.35	3.29	10.08	1.27	2.02	0.62	1.14	2.00	0.71	0.64	1.68	0.95	8.75
11	芜湖市	4825	28.35	8.05	30.78	5.60	10.04	1.42	1.82	2.90	0.57	1.18	0.20	0.43	1.24	0.91	6.53
12	宣城市	5577	19.48	29.23	38.07	2.79	1.57	1.19	1.24	0.48	0.44	0.27	0.07	0.15	0.89	0.44	3.73
13	铜陵市	2819	41.77	20.92	14.41	4.93	3.74	0.88	0.54	0.57	0.51	0.71	0.13	0.47	1.28	0.67	8.43
14	池州市	3343	22.02	32.97	22.22	4.01	2.33	2.05	1.76	0.88	0.83	0.63	0.23	1.11	0.68	0.71	7.60
15	安庆市	6658	13.67	26.01	16.47	7.54	5.23	6.67	3.29	3.10	1.31	1.27	0.41	3.14	1.81	0.47	9.61
16	黄山市	3783	12.28	56.32	16.68	3.42	1.53	2.16	1.76	0.30	0.25	0.13	0.05	0.20	0.75	0.20	4.02

```
data=pd.DataFrame(df)
header =['地区', '合计', '江苏', '浙江', '上海', '广东', '北京', '福建', '山东', '天津', '河南', '河北', '新疆', '辽宁', '湖北', '陕西', '流向其他省市']
data.loc[0].values[2:-1] # 只选用第一行全省数据
```

```
# print(df) 可以检验一下数据是否导入正确
area = header[2:-1]
value = data.loc[0].values[2:-1]


data_pair = [(area[i], value[i]) for i in range(len(area))]
print(data_pair)

map = Map(init_opts=opts.InitOpts(width="1720px", height="920px"))  # 创建地图，其中括号内可以调整大小，也可以修改主题颜色。
map.add("安徽人口", data_pair, maptype="china")  # 添加安徽地图
map.set_global_opts(  # 设置全局配置项#
    title_opts=opts.TitleOpts(title="来此的安徽人安徽"),  # 添加标题
    visualmap_opts=opts.VisualMapOpts(max_=31, is_piecewise=True,split_number = 20,),  # 最大数据范围 并且使用分段
    legend_opts=opts.LegendOpts(is_show=False),  # 是否显示图例
)
map.render('省外半年以上人口分布.html')  # map.render_notebook()直接在notebook中显示# map.render('map1.html') 将地图以html形式保存在工作目录下
```

```
[('江苏', 30.02), ('浙江', 25.72), ('上海', 21.79), ('广东', 4.35), ('北京', 3.31), ('福建', 2.12), ('山东', 1.66), ('天津', 1.28), ('河南', 0.98), ('河北', 1.08), ('新疆', 0.62), ('辽宁', 0.69), ('湖北', 0.87), ('陕西', 0.89)]
```



省外半年以上人口分布.html

```
import pandas as pd
from pyecharts import options as opts
from pyecharts.charts import Geo
from pyecharts.datasets import register_url
from pyecharts.globals import ChartType, SymbolType


#各市流向省外半年以上的流动人口构成（2019年）
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('各市流向省外半年以上的流动人口构成（2019年）.csv',index_col=False )
df.columns = ['地区', '合计', '江苏', '浙江', '上海', '广东', '北京', '福建', '山东', '天津', '河南', '河北', '新疆', '辽宁', '湖北', '陕西', '流向其他省市']
data=pd.DataFrame(df)

header =['江苏', '浙江', '上海', '广东', '北京', '福建', '山东', '天津', '河南', '河北', '新疆', '辽宁', '湖北', '陕西']
data = data.loc[0].values[2:-1] # 只选用第一行全省数据

data_pair = [(header[i], data[i]) for i in range(len(header))]

print(data_pair)

location_pair = [("安徽", header[i]) for i in range(len(header))]
print(location_pair)
title = "省外半年以上去向示意图"
```

```
c = (
    Geo(init_opts=opts.InitOpts(width="1720px", height="920px"))
    .add_schema(
            maptype='china',    #设置国家名字
            itemstyle_opts = opts.ItemStyleOpts(color = 'skyblue',border_color = '#111'),)  #设置地图颜色和边框色
    .add(
        "目的地",   #第一个add数据的标题
        data_pair, #目的地数据
        type_ = ChartType.EFFECT_SCATTER,  #使用点的样式，并设置点的颜色，点的大小都是一样的！
        symbol_size = 6,  #设置点的大小
        color = 'blue',) #点的颜色
    .add(
        "连线",
        location_pair, #本处，目的地 元组
        type_ = ChartType.LINES,
        effect_opts = opts.EffectOpts(
                symbol = SymbolType.ARROW, symbol_size = 6, color = 'cyan'),  #线上的小箭头的颜色
        linestyle_opts = opts.LineStyleOpts(curve = 0.2))  #设置两点间线缆的弯曲度
    .set_series_opts(label_opts = opts.LabelOpts(is_show = False))  #安徽->** 显示在线上，设为不显示
    .set_global_opts(title_opts=opts.TitleOpts(title=title),
                     toolbox_opts = opts.ToolboxOpts())
    .render(r"省外半年以上去向示意图.html")
)
```

```
[('江苏', 30.02), ('浙江', 25.72), ('上海', 21.79), ('广东', 4.35), ('北京', 3.31), ('福建', 2.12), ('山东', 1.66), ('天津', 1.28), ('河南', 0.98), ('河北', 1.08), ('新疆', 0.62), ('辽宁', 0.69), ('湖北', 0.87), ('陕西', 0.89)]
[('安徽', '江苏'), ('安徽', '浙江'), ('安徽', '上海'), ('安徽', '广东'), ('安徽', '北京'), ('安徽', '福建'), ('安徽', '山东'), ('安徽', '天津'), ('安徽', '河南'), ('安徽', '河北'), ('安徽', '新疆'), ('安徽', '辽宁'), ('安徽', '湖北'), ('安徽', '陕西')]
```

- **省外半年以上去向示意图**

省外半年以上去向示意图.html

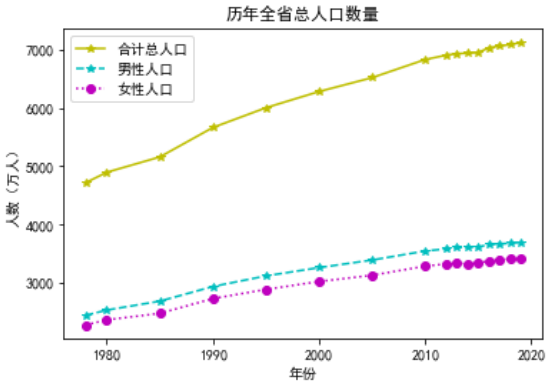
3—25历年全省总人口、总户数

```csv
年份,总户数,合计总人口,男性人口,女性人口,性别比(女=100),城镇人口,乡村人口
1978,1018,4713,2439,2274,107.27,504,4209
1980,1051,4893,2530,2363,107.10,556,4337
1985,1174,5156,2683,2473,108.46,724,4432
1990,1445,5661,2934,2727,107.57,843,4818
1995,1551,6000,3116,2884,108.08,1044,4956
2000,1656,6278,3258,3020,107.87,1230,5048
2005,1849,6516,3388,3127,108.34,1368,5148
2010,2093,6827,3543,3283,107.92,1550,5276
2012,2139,6902,3580,3322,107.78,1580,5322
2013,2144,6929,3599,3330,108.08,1588,5341
2014,2123,6936,3610,3326,108.57,1574,5362
2015,2132,6949,3615,3334,108.43,1917,5032
2016,2142,7027,3652,3374,108.24,2075,4952
2017,2149,7059,3666,3393,108.04,2193,4866
2018,2158,7083,3676,3407,107.91,2313,4770
2019,2176,7119,3694,3425,107.86,2467,4652
```

```
#历 年 全 省 总 人 口、总 户 数
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('历 年 全 省 总 人 口、总 户 数.csv',index_col=False )
df.columns = ['年份','总户数','合计总人口','男性人口','女性人口','性别比(女=100)','城镇人口','乡村人口']
df
```

	年份	总户数	合计总人口	男性人口	女性人口	性别比(女=100)	城镇人口	乡村人口
0	1978	1018	4713	2439	2274	107.27	504	4209
1	1980	1051	4893	2530	2363	107.10	556	4337
2	1985	1174	5156	2683	2473	108.46	724	4432
3	1990	1445	5661	2934	2727	107.57	843	4818
4	1995	1551	6000	3116	2884	108.08	1044	4956
5	2000	1656	6278	3258	3020	107.87	1230	5048
6	2005	1849	6516	3388	3127	108.34	1368	5148
7	2010	2093	6827	3543	3283	107.92	1550	5276
8	2012	2139	6902	3580	3322	107.78	1580	5322
9	2013	2144	6929	3599	3330	108.08	1588	5341
10	2014	2123	6936	3610	3326	108.57	1574	5362
11	2015	2132	6949	3615	3334	108.43	1917	5032
12	2016	2142	7027	3652	3374	108.24	2075	4952
13	2017	2149	7059	3666	3393	108.04	2193	4866
14	2018	2158	7083	3676	3407	107.91	2313	4770
15	2019	2176	7119	3694	3425	107.86	2467	4652

```
x =list(df['年份'])
y1 = list(df['合计总人口'])
y2 = list(df['男性人口'])
y3 = list(df['女性人口'])

fig, ax = plt.subplots()  # Create a figure and an axes.

ax.plot(x, y1, 'y*-', label='合计总人口')
ax.plot(x, y2, 'c*--', label='男性人口')
ax.plot(x, y3, 'mo:', label='女性人口')
ax.set_xlabel('年份')  # Add an x-label to the axes.
ax.set_ylabel('人数（万人）')  # Add a y-label to the axes.
ax.set_title("历年全省总人口数量")  # Add a title to the axes.
ax.legend()  # Add a legend.
```

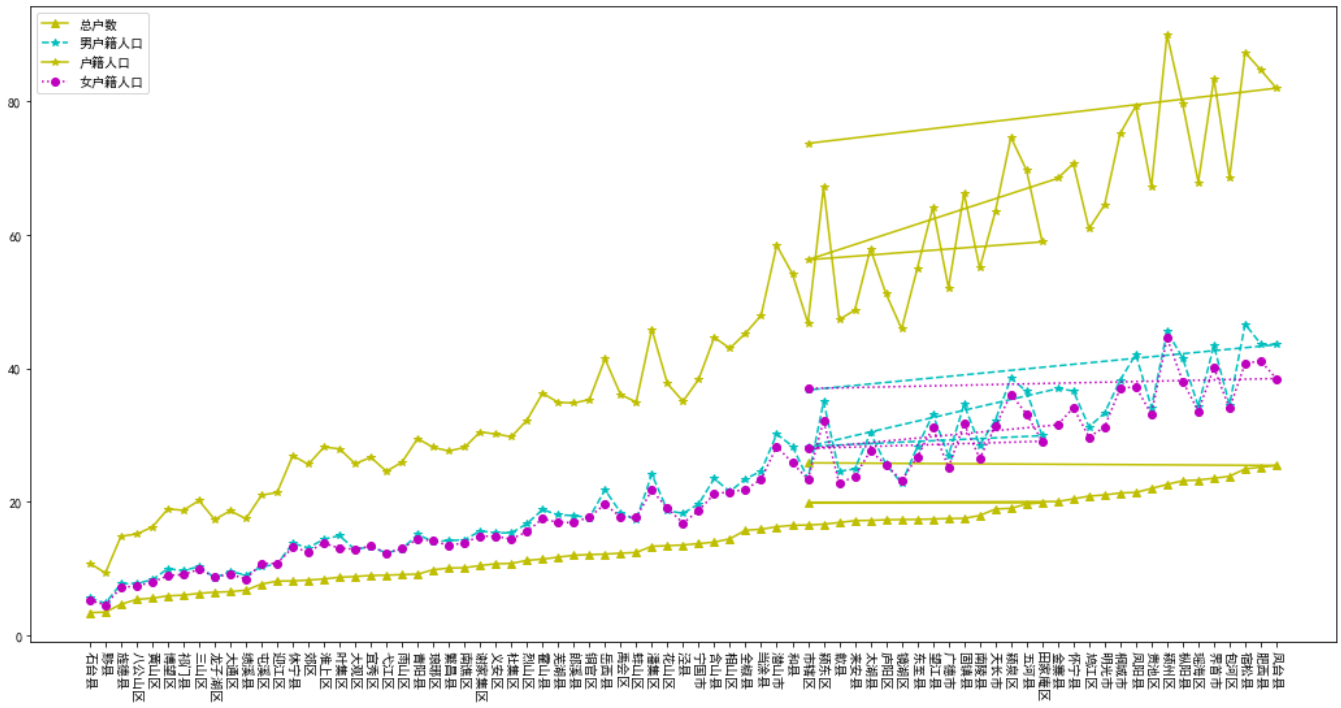


3—26各市、县、区户数、人口数（2019年）

```
地区,总户数（万人）,户籍人口（万人）,男户籍人口,女户籍人口,性别比（女=100）,城镇人口,常住人口
总计,2176.29,7119.37,3694.33,3425.04,107.86,2467.15,6365.9
合肥市,255.33,770.44,395.24,375.21,105.34,404.99,818.9
市辖区,100.48,290.82,146.68,144.15,101.75,279.52,400.3
瑶海区,23.24,67.85,34.28,33.57,102.10,64.77,101.6
庐阳区,17.28,51.14,25.59,25.55,100.17,49.85,69.2
蜀山区,36.15,103.35,52.31,51.04,102.49,98.37,131.2
包河区,23.80,68.49,34.50,33.99,101.51,66.53,98.3
巢湖市,31.07,86.04,44.21,41.82,105.72,25.48,80.3
长丰县,26.28,79.61,41.30,38.31,107.79,21.82,67.4
肥东县,34.52,108.12,56.75,51.37,110.48,24.34,90.5
肥西县,25.17,84.69,43.62,41.07,106.23,23.36,79.0
庐江县,37.81,121.16,62.67,58.49,107.15,30.46,101.4
淮北市,73.04,218.72,112.25,106.47,105.43,120.05,227.0
市辖区,36.37,104.88,53.51,51.37,104.15,71.30,120.3
杜集区,10.74,29.72,15.31,14.40,106.33,18.10,34.4
相山区,14.43,42.97,21.50,21.47,100.18,36.38,51.1
烈山区,11.20,32.20,16.69,15.51,107.63,16.82,34.8
濉溪县,36.67,113.84,58.74,55.10,106.62,48.75,106.7
亳州市,179.68,662.99,347.91,315.07,110.42,132.97,526.3
谯城区,54.05,169.94,88.22,81.71,107.96,35.32,153.1
涡阳县,36.43,172.01,90.52,81.49,111.07,37.90,131.9
蒙城县,35.42,146.46,77.02,69.45,110.90,31.43,116.1
```

```
# 3-26各市、县、区户数、人口数（2019年）
# 设定index_col=False 保证pandas用第一列作为行索引
df = pd.read_csv('各市、县、区户数、人口数（2019年）.csv',index_col=False )
df.columns = ['地区', '总户数（万人）', '户籍人口（万人）', '男户籍人口', '女户籍人口', '性别比（女=100）', '城镇人口', '常住人口']
df = df[1:]
df = df.sort_values(by='总户数（万人）')
df
```

	地区	总户数（万人）	户籍人口（万人）	男户籍人口	女户籍人口	性别比（女=100）	城镇人口	常住人口
129	徽州区	3.31	9.58	4.89	4.68	104.42	4.02	9.8
111	石台县	3.37	10.76	5.56	5.20	106.93	3.32	10.0
132	黟县	3.44	9.32	4.78	4.54	105.42	2.93	8.3
101	旌德县	4.63	14.82	7.70	7.12	108.19	4.02	12.5
53	八公山区	5.34	15.13	7.74	7.38	104.91	12.84	17.7
...	...	...	...	...	...	...	...	...
18	亳州市	179.68	662.99	347.91	315.07	110.42	132.97	526.3
67	六安市	189.85	591.07	311.86	279.21	111.69	137.88	487.3
23	宿州市	197.85	658.27	342.28	315.99	108.32	160.01	570.0
1	合肥市	255.33	770.44	395.24	375.21	105.34	404.99	818.9
38	阜阳市	285.36	1077.28	561.09	516.19	108.70	263.88	825.9

133 rows × 8 columns

```
x = df['地区'][1:80]
y1 = df['总户数（万人）'][1:80]
y2 = df['男户籍人口'][1:80]
y3 = df['户籍人口（万人）'][1:80]
y4 = df['女户籍人口'][1:80]

fig, ax = plt.subplots(figsize=(18, 9))

ax.plot(x, y1,'y^-',label='总户数')

ax.plot(x, y2,'c*--', label='男户籍人口')
ax.plot(x, y3, 'y*-',label='户籍人口')
ax.plot(x, y4,'mo:', label='女户籍人口')
ax.legend()  # Add a legend.

plt.xticks(rotation=270)
plt.show()
```