题目

> Made By RingFa1l_08163337

> E. 利用爬虫采集天气信息（4人）
>
> 天气数据存到数据库里；
>
> 数据可视化分析统计：各省会当日高低温、与去年同期比较、多日高低温预报、天气警报等（不必全做出来）。

> 项目/作业报告：题目，设计思路，开发环境配置，运行使用方法，运行效果截图，运行效果分析总结——缺陷、改进方法；
>
> 源代码；数据（数据库、文件、图片等形式）；可执行程序（可选）。

开发环境配置

- Python 3.6.1
- SQLite 3.28.0
- beautifulsoup4 4.6.0
- requests 2.19.1

设计思路

1.爬取数据存入数据库

- 使用requests库获取页面
- 使用BeautifulSoup库解析html
- 使用weather.db存储数据

2.数据可视化分析统计

- 使用pyecharts库画图

七天预报功能

数据来源：www.weather.com.cn （有个坑是今天的天气在早上是高温/低温，晚上就只有一个温度了x

通过观察发现查询的网站城市对应url中的九位编码，获取对应关系存储在json文件中，编写根据城市查询对应编码的函数：

```python
def city2code(city_name):
    with open('city_code.json', 'r') as f:
        data = json.load(f)
    f.close()
    try:
        return data[city_name]
    except:
        sys.exit('no city {} information'.format(city_name))
```

用户输入城市，通过city2code函数得到编码，爬取生成url返回的html，使用BeautifulSoup结构化解析出七天预报的日期、天气状况、高低温、风向和强度信息：

```
city = input("city:")
city_code = city2code(city)
url = "http://www.weather.com.cn/weather/{}.shtml".format(city_code)
res = requests.get(url)
res.encoding = 'utf-8'
bs = BeautifulSoup(res.text,"html.parser")
date = bs.select("li.sky > h1")
desc = bs.select("li.sky > p.wea")
temp = bs.select("li.sky > p.tem")
dir = bs.select("li > p.win > em")
level = bs.select("li > p.win > i")

result = []
for i in range(date.__len__()):
    date1 = date[i].text
    desc1 = desc[i].text
    temp1 = temp[i].stripped_strings
    temp1 = "".join(temp1)
    dir1 = dir[i]
    dir2 = dir1.select("span")
    if len(dir2) == 1:
        direction = "无持续风向"
    else:
        direction = dir2[0].get("title")+"-"+dir2[1].get("title")
    level1 = level[i].text
    result.append([city,date1,desc1,temp1,direction,level1])
```

连接数据库，先搜索原有表单中这个城市的信息并删除，如果没有week_weather表先建立，然后将新的数据存入后关闭连接：

```
conn = sqlite3.connect('weather.db')
cursor = conn.cursor()
try:
    cursor.execute("delete from week_weather where city='{}'".format(city))
    print("delete old data")
except:
    cursor.execute('''create table week_weather (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        city char(100),
        date char(100),
        desc char(100),
        temp char(100),
        direction char(100),
        level char(100)
        );''')
    print("create new table")
for result1 in result:
    print(result1)

    insertsql = "insert into week_weather (city,date,desc,temp,direction,level) VALUES
```

```
('%s','%s','%s','%s','%s','%s')"
    cursor.execute(insertsql %
(result1[0],result1[1],result1[2],result1[3],result1[4],result1[5]))

conn.commit()
cursor.close()
conn.close()
```

可视化脚本连接数据库取出所需城市信息并把高低温文本解析成数字，如果本身信息不全则输出缺失退出运行：

```
x = []
y1 = []
y2 = []
try:
    for row in cursor:
        x.append(row[2])
        h = row[4].split('/')[0]
        if h.find('℃'):
            high = int(h.split('℃')[0])
        else:
            high = int(h)
        try:
            low = int(row[4].split('/')[1].split('℃')[0])
        except:
            low = high
        y1.append(high)
        y2.append(low)
except:
    sys.exit("高低温信息缺失")
```

柱状图绘制并保存在result目录中：

```
bar = Bar()
bar.add_xaxis(x)
bar.add_yaxis("最高温度", y1)
bar.add_yaxis("最低温度", y2)
bar.set_global_opts(title_opts=opts.TitleOpts(title="{}七日高低温预测".format(city),
subtitle="made by Ringfa1l"))
bar.render("result/{}_pre.html".format(city))
```

各省会实时温度功能

数据来源：tianqi.eastday.com

同样是url由城市对应的拼音和一个五位编码组成，首先建立省会字典：

```
dic1 = ['河北省', '山西省', '辽宁省','吉林省', '黑龙江省', '江苏省', '浙江省', '安徽省', '福建省',
'江西省', '山东省', '河南省', '广东省', '湖南省', '湖北省', '海南省', '四川省', '贵州省', '云南省',
'陕西省', '甘肃省', '青海省', '台湾省', '内蒙古自治区', '广西壮族自治区', '西藏自治区', '宁夏回族自治
区', '新疆维吾尔自治区']
dic2 = ['石家庄', '太原', '沈阳', '长春', '哈尔滨', '南京', '杭州', '合肥', '福州', '南昌', '济南',
'郑州', '广州', '长沙', '武汉', '海口', '成都', '贵阳', '昆明', '西安', '兰州', '西宁', '台北', '呼和
浩特', '南宁', '拉萨', '银川', '乌鲁木齐']
dic3 = ['53698', '53772', '54342', '54161', '50953', '58238', '58457', '58321', '58847',
'58606', '54823', '57083', '59287', '57687', '57494', '59758', '56294', '57816', '56778',
'57036', '52889', '52866', '71294', '53463', '59431', '55591', '53614','51463']
dic4 = ['shijiazhuang', 'taiyuan', 'shenyang', 'changchun', 'haerbin', 'nanjing', 'hangzhou',
'hefei', 'fuzhou', 'nanchang', 'jinan', 'zhengzhou', 'guangzhou', 'changsha', 'wuhan', 'haikou',
'chengdu', 'guiyang', 'kunming', 'xian', 'lanzhou', 'xining', 'taibei', 'huhehaote', 'nanning',
'lasa', 'yinchuan', 'wulumuqi']
```

爬取数据：

```
result = []
for i in range(dic1.__len__()):
    url = 'http://tianqi.eastday.com/{}/{}.html'.format(dic4[i],dic3[i])
    res = requests.get(url)
    bs = BeautifulSoup(res.text,"html.parser")
    data = bs.select("div.tempBox > span")
    temp = int(data[0].text)
    result.append([dic1[i],dic2[i],temp])
```

存入数据库live_weather表：

```
conn = sqlite3.connect('weather.db')
cursor = conn.cursor()
try:
    cursor.execute("drop table live_weather;")
    print("delete old data")
except:
    pass
cursor.execute('''create table live_weather (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    province char(100),
    city char(100),
    temp int
    );''')
print("create new table")
for result1 in result:
    print(result1)
    insertsql = "insert into live_weather (province,city,temp) VALUES ('%s','%s','%d')"
    cursor.execute(insertsql % (result1[0],result1[1],result1[2]))
```

取出数据：

```
conn = sqlite3.connect('weather.db')

c = conn.cursor()
```

```
print("opened database successfully")

try:
    cursor = c.execute("SELECT * from live_weather")
    print("data selected successfully")
except:
    sys.exit("No information")

city = []
temp = []
for row in cursor:
    city.append(row[2])
    temp.append(row[3])
```

画地图：

```
def geo_base() -> Geo:
    c = (
        Geo()
        .add_schema(maptype="china")
        .add("省会温度", [list(z) for z in zip(city, temp)])
        .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
        .set_global_opts(
            visualmap_opts=opts.VisualMapOpts(max_=30),
            title_opts=opts.TitleOpts(title="省会城市实时温度", subtitle="made by Ringfa1l"),
        )
    )
    return c

geo_base().render("result/live_temp.html")
```

历史数据比较功能

数据来源：tianqi.eastday.com

使用time库获取当前年月日信息：

```
year = str(int(time.strftime("%Y", time.localtime()))-1)
mon = time.strftime("%m", time.localtime())
day = time.strftime("%d", time.localtime())
```

循环爬取信息（这个网站有两种格式，所以加个判断~）：

```
result = []
for i in range(dic1.__len__()):
    for j in range(1,13):
        monn = int(mon)+j
        if monn > 12:
            monn = monn-12
            yearr = int(year) + 1
        else:
            yearr = int(year)
        url = 'http://tianqi.eastday.com/{}_history/{}_{}{:0>2d}.html'.format(dic4[i],dic3[i],yearr,monn)
        #print(url)
        res = requests.get(url)
        bs = BeautifulSoup(res.text,"html.parser")
        if bs.find_all(class_="item-value") != []:
            data = bs.find_all(class_="item-value")
            high = re.findall(r"\d+",data[0].text)
            low = re.findall(r"\d+",data[1].text)
            #print(high,low)
            try:
                result.append([dic2[i],yearr,monn,int(high[0]),int(low[0])])
            except:
                print(dic2[i]+'-'+str(yearr)+'-'+str(monn)+"data lost")
        else:
            data = bs.select("ul.history2_left > li")
            high = re.findall(r"\d+",data[0].text)
            low = re.findall(r"\d+",data[1].text)
            #print(high,low)
            try:
                result.append([dic2[i],yearr,monn,int(high[0]),int(low[0])])
            except:
                print(dic2[i]+'-'+str(yearr)+'-'+str(monn)+"data lost")
```

传入数据库his_weather表:

```
conn = sqlite3.connect('weather.db')
cursor = conn.cursor()
try:
    cursor.execute("drop table his_weather;")
    print("delete old data")
except:
    pass
cursor.execute('''create table his_weather (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    city char(100),
    mon char(100),
    high int,
    low int
    );''')
print("create new table")
for result1 in result:
    print(result1)
    insertsql = "insert into his_weather (city,mon,high,low) VALUES ('%s','%s','%d','%d')"
    cursor.execute(insertsql % (result1[0],str(result1[1])+'年'+str(result1[2])+'月',result1[3],result1[4]))
```

获取特定城市数据:

```
city = input("请输入一个省会城市:")


conn = sqlite3.connect('weather.db')
```

```
c = conn.cursor()
print("opened database successfully")

try:
    cursor = c.execute("SELECT * from his_weather where city='{}'".format(city))
    print("data selected successfully")
except:
    sys.exit("No information about {}".format(city))

time = []
high = []
low = []

for row in cursor:
    time.append(row[2])
    high.append(row[3])
    low.append(row[4])
```

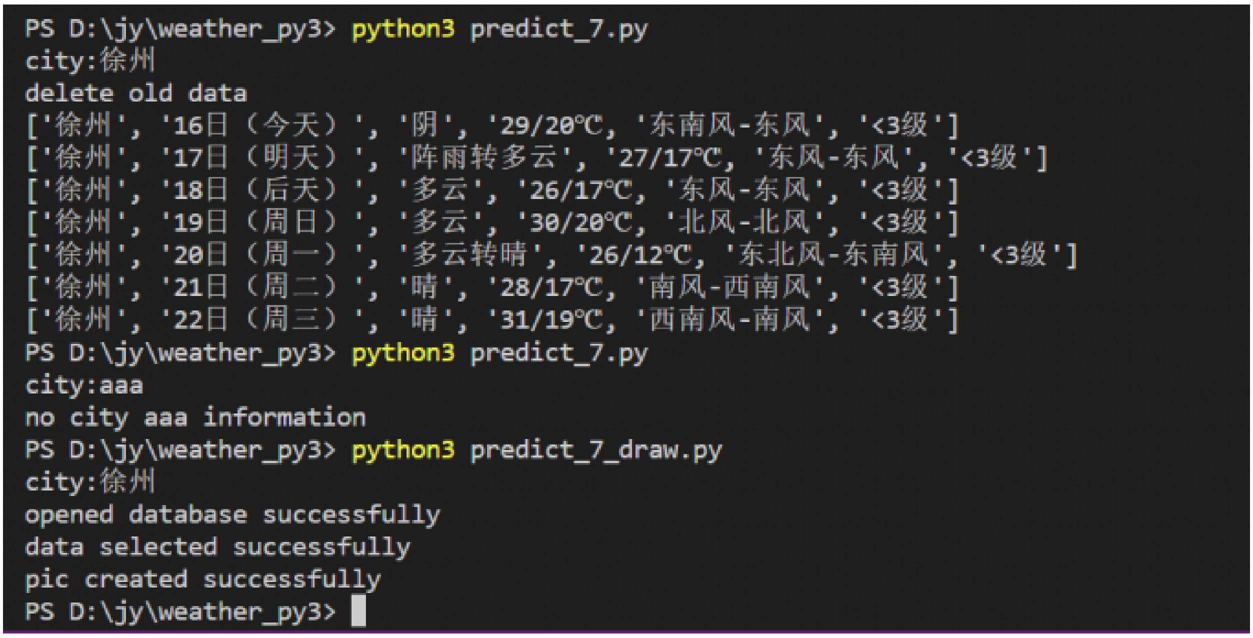
画折线图：

```
def line_base() -> Line:
    c = (
        Line()
        .add_xaxis(time)
        .add_yaxis("月平均最高温度", high)
        .add_yaxis("月平均最低温度", low)
        .set_global_opts(title_opts=opts.TitleOpts(title="{}历史气温比较".format(city),
subtitle="made by Ringfa1l"))
    )
    return c

line_base().render("result/{}_his_compare.html".format(city))
```

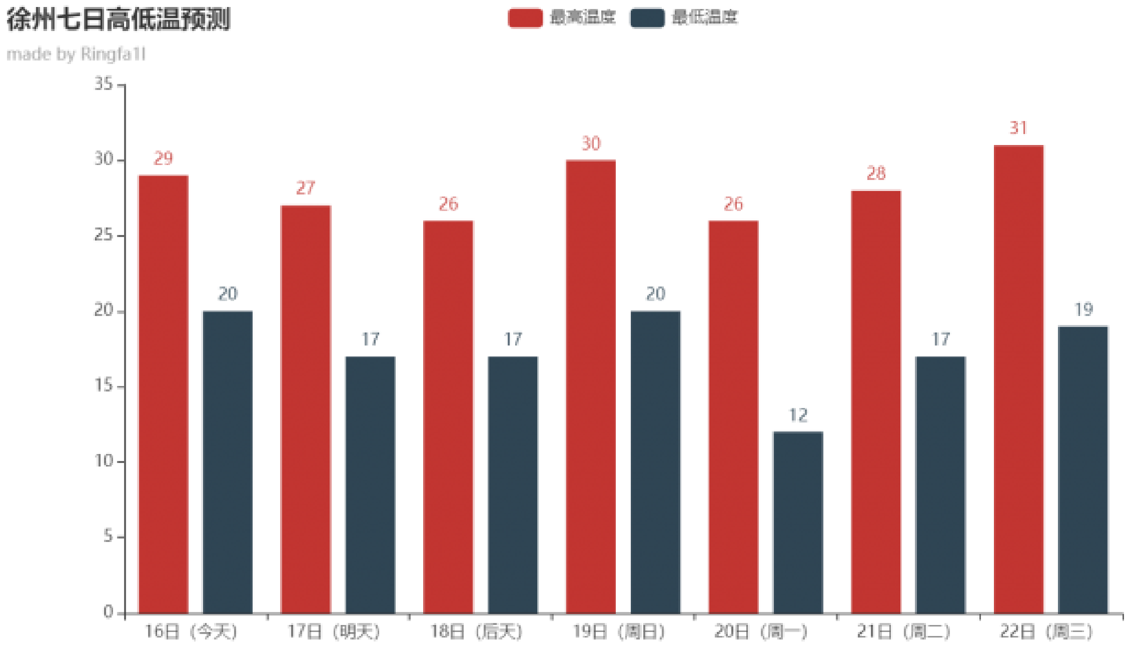
运行使用方法和截图

- 先在目录下新建weather.db数据库
- 运行predict_7.py，按提示输入城市，输入错误会提示

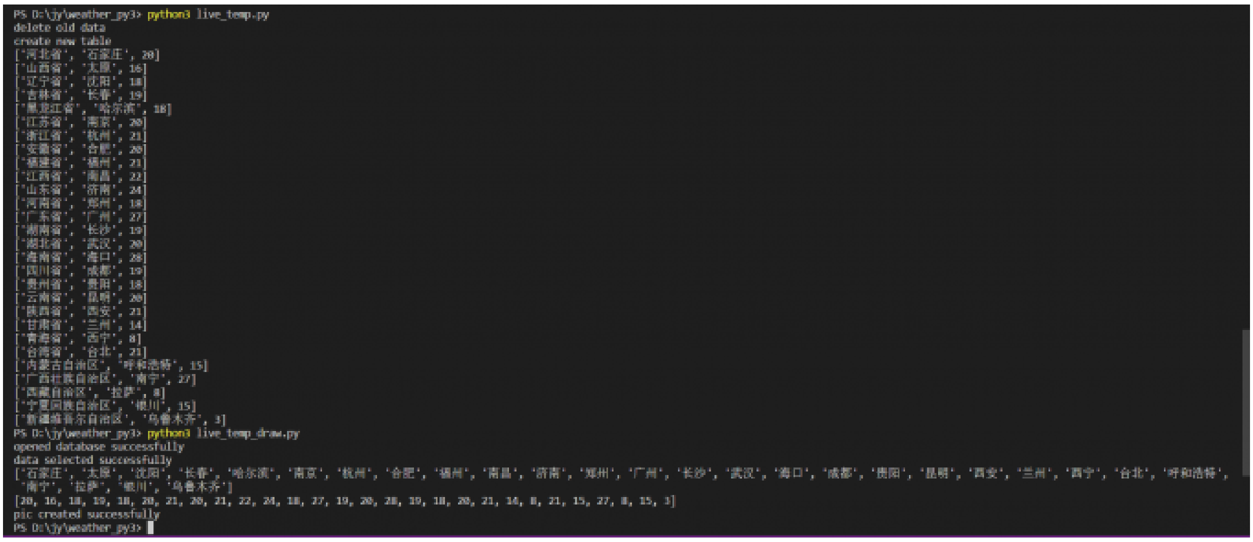


```
PS D:\jy\weather_py3> python3 predict_7.py
city:徐州
delete old data
['徐州', '16日（今天）', '阴', '29/20℃, '东南风-东风', '<3级']
['徐州', '17日（明天）', '阵雨转多云', '27/17℃, '东风-东风', '<3级']
['徐州', '18日（后天）', '多云', '26/17℃, '东风-东风', '<3级']
['徐州', '19日（周日）', '多云', '30/20℃, '北风-北风', '<3级']
['徐州', '20日（周一）', '多云转晴', '26/12℃, '东北风-东南风', '<3级']
['徐州', '21日（周二）', '晴', '28/17℃, '南风-西南风', '<3级']
['徐州', '22日（周三）', '晴', '31/19℃, '西南风-南风', '<3级']
PS D:\jy\weather_py3> python3 predict_7.py
city:aaa
no city aaa information
PS D:\jy\weather_py3> python3 predict_7_draw.py
city:徐州
opened database successfully
data selected successfully
pic created successfully
PS D:\jy\weather_py3>
```

- 运行predict_7_draw.py，按提示输入城市，画出的柱形图在result目录下

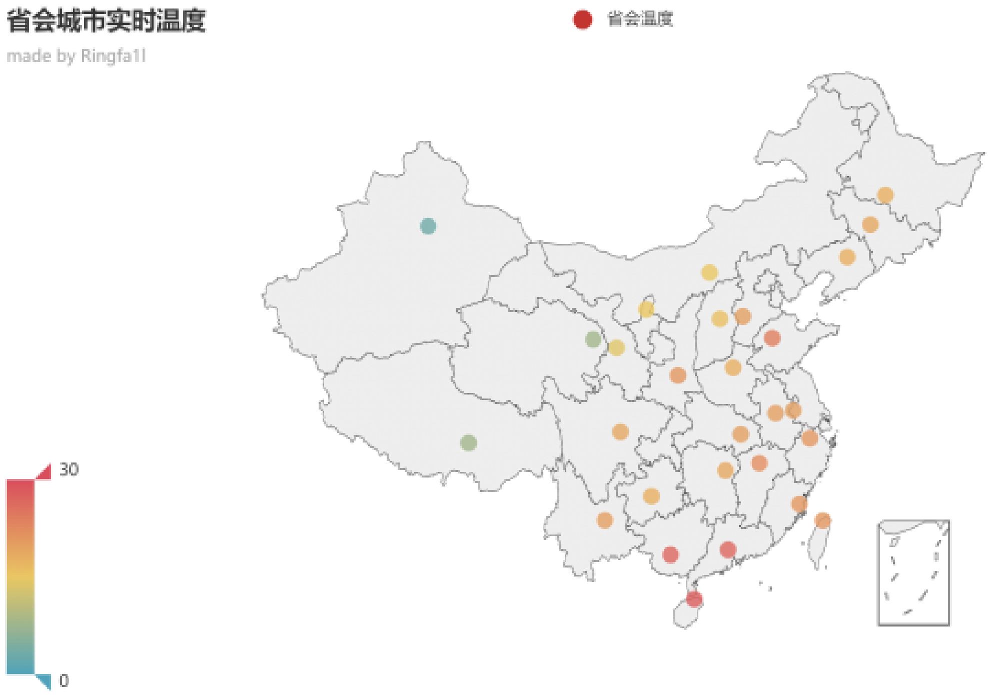


- 运行live_temp.py

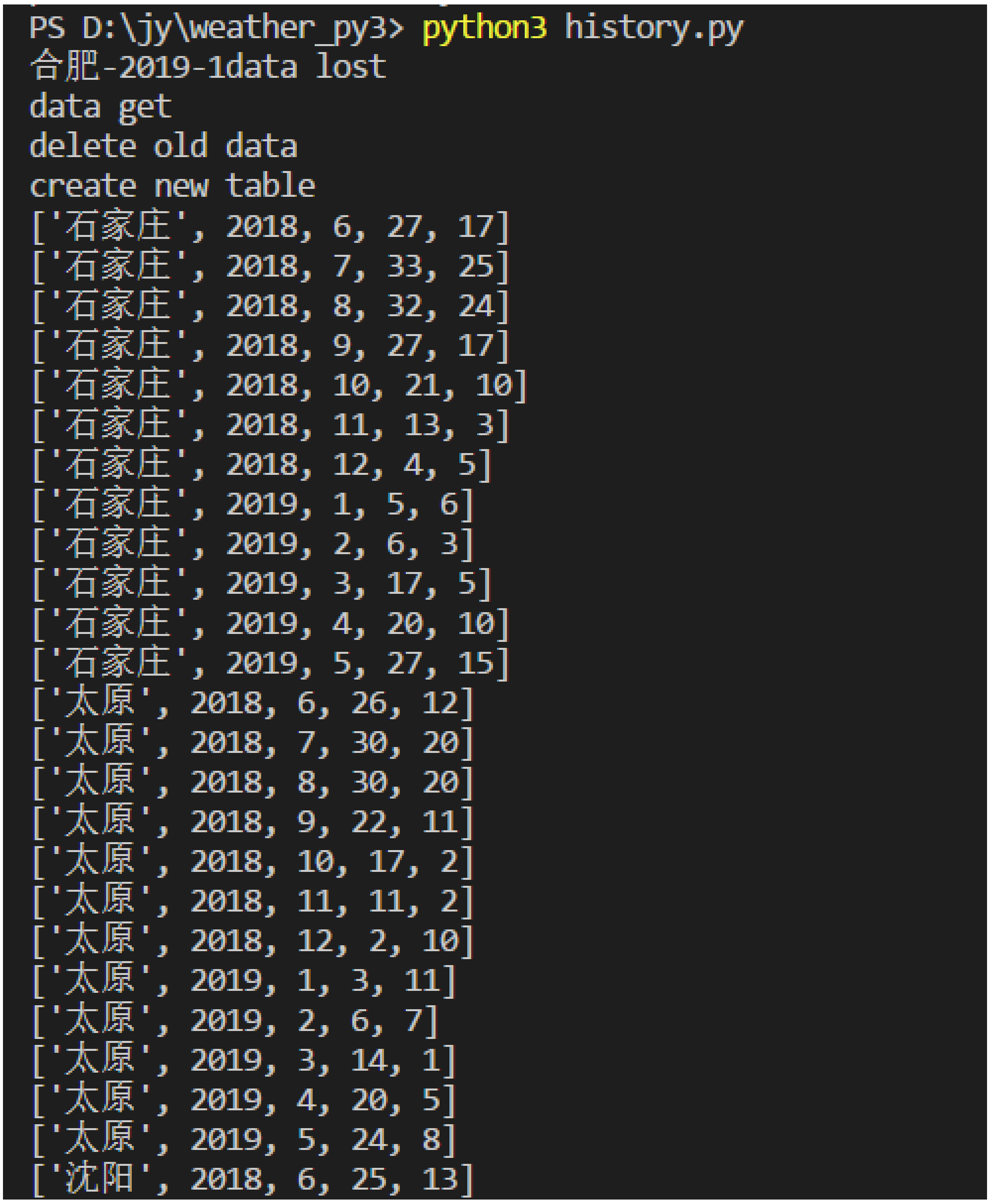


```
PS D:\jy\weather_py3> python3 live_temp.py
delete old data
create new table
['河北省', '石家庄', 20]
['山西省', '太原', 16]
['辽宁省', '沈阳', 18]
['吉林省', '长春', 19]
['黑龙江省', '哈尔滨', 18]
['江苏省', '南京', 20]
['浙江省', '杭州', 21]
['安徽省', '合肥', 20]
['福建省', '福州', 21]
['江西省', '南昌', 22]
['山东省', '济南', 24]
['河南省', '郑州', 18]
['广东省', '广州', 27]
['湖南省', '长沙', 19]
['湖北省', '武汉', 20]
['海南省', '海口', 28]
['四川省', '成都', 19]
['贵州省', '贵阳', 18]
['云南省', '昆明', 20]
['陕西省', '西安', 21]
['甘肃省', '兰州', 14]
['青海省', '西宁', 8]
['台湾省', '台北', 21]
['内蒙古自治区', '呼和浩特', 15]
['广西壮族自治区', '南宁', 27]
['西藏自治区', '拉萨', 8]
['宁夏回族自治区', '银川', 15]
['新疆维吾尔自治区', '乌鲁木齐', 3]
PS D:\jy\weather_py3> python3 live_temp_draw.py
opened database successfully
data selected successfully
['石家庄', '太原', '沈阳', '长春', '哈尔滨', '南京', '杭州', '合肥', '福州', '南昌', '济南', '郑州', '广州', '长沙', '武汉', '海口', '成都', '贵阳', '昆明', '西安', '兰州', '西宁', '台北', '呼和浩特',
 '南宁', '拉萨', '银川', '乌鲁木齐']
[20, 16, 18, 19, 18, 20, 21, 20, 21, 22, 24, 18, 27, 19, 20, 28, 19, 18, 20, 21, 14, 8, 21, 15, 27, 8, 15, 3]
pic created successfully
PS D:\jy\weather_py3> 
```

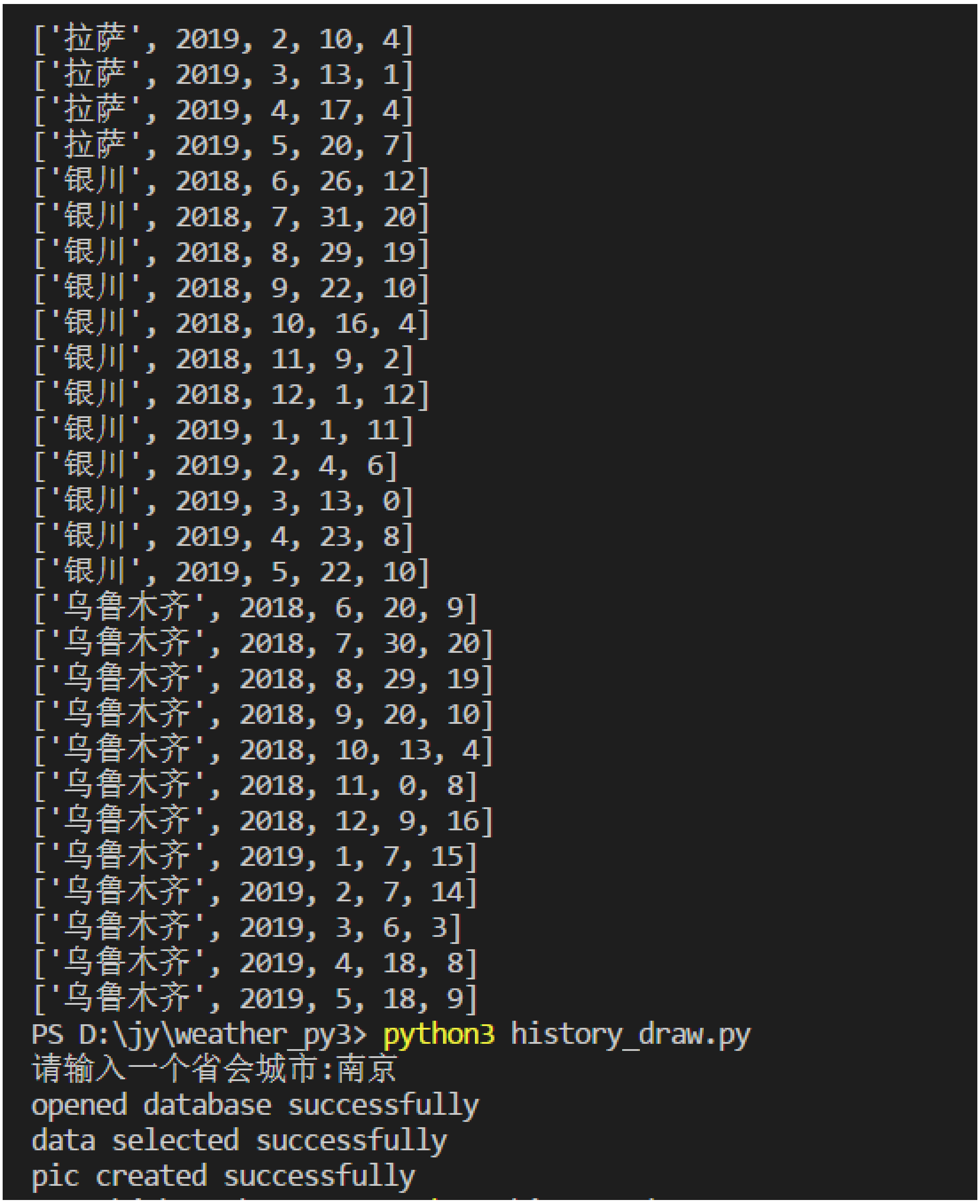
- 运行live_temp_draw.py，画出地图，html格式时移动鼠标可显示省市信息和具体温度值



- 运行history.py，输入省会城市

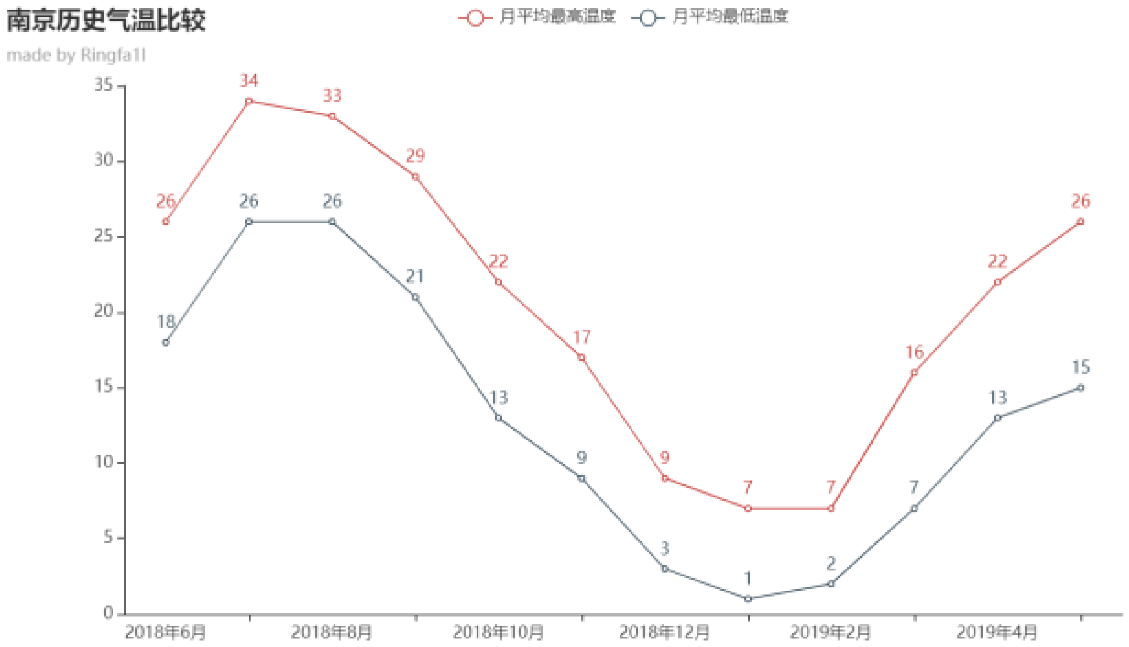


```
PS D:\jy\weather_py3> python3 history.py
合肥-2019-1data lost
data get
delete old data
create new table
['石家庄', 2018, 6, 27, 17]
['石家庄', 2018, 7, 33, 25]
['石家庄', 2018, 8, 32, 24]
['石家庄', 2018, 9, 27, 17]
['石家庄', 2018, 10, 21, 10]
['石家庄', 2018, 11, 13, 3]
['石家庄', 2018, 12, 4, 5]
['石家庄', 2019, 1, 5, 6]
['石家庄', 2019, 2, 6, 3]
['石家庄', 2019, 3, 17, 5]
['石家庄', 2019, 4, 20, 10]
['石家庄', 2019, 5, 27, 15]
['太原', 2018, 6, 26, 12]
['太原', 2018, 7, 30, 20]
['太原', 2018, 8, 30, 20]
['太原', 2018, 9, 22, 11]
['太原', 2018, 10, 17, 2]
['太原', 2018, 11, 11, 2]
['太原', 2018, 12, 2, 10]
['太原', 2019, 1, 3, 11]
['太原', 2019, 2, 6, 7]
['太原', 2019, 3, 14, 1]
['太原', 2019, 4, 20, 5]
['太原', 2019, 5, 24, 8]
['沈阳', 2018, 6, 25, 13]
```



```
['拉萨', 2019, 2, 10, 4]
['拉萨', 2019, 3, 13, 1]
['拉萨', 2019, 4, 17, 4]
['拉萨', 2019, 5, 20, 7]
['银川', 2018, 6, 26, 12]
['银川', 2018, 7, 31, 20]
['银川', 2018, 8, 29, 19]
['银川', 2018, 9, 22, 10]
['银川', 2018, 10, 16, 4]
['银川', 2018, 11, 9, 2]
['银川', 2018, 12, 1, 12]
['银川', 2019, 1, 1, 11]
['银川', 2019, 2, 4, 6]
['银川', 2019, 3, 13, 0]
['银川', 2019, 4, 23, 8]
['银川', 2019, 5, 22, 10]
['乌鲁木齐', 2018, 6, 20, 9]
['乌鲁木齐', 2018, 7, 30, 20]
['乌鲁木齐', 2018, 8, 29, 19]
['乌鲁木齐', 2018, 9, 20, 10]
['乌鲁木齐', 2018, 10, 13, 4]
['乌鲁木齐', 2018, 11, 0, 8]
['乌鲁木齐', 2018, 12, 9, 16]
['乌鲁木齐', 2019, 1, 7, 15]
['乌鲁木齐', 2019, 2, 7, 14]
['乌鲁木齐', 2019, 3, 6, 3]
['乌鲁木齐', 2019, 4, 18, 8]
['乌鲁木齐', 2019, 5, 18, 9]
PS D:\jy\weather_py3> python3 history_draw.py
请输入一个省会城市:南京
opened database successfully
data selected successfully
pic created successfully
```

- 运行history_draw.py，画出历史比较折线图



运行效果分析总结

脚本	缺陷	解决方法
predict_7.py	用户输入不可信	加入是否是可查询城市的判断
history_draw.py	输入不是省会城市	加入数据库中是否有相关信息的判断
history.py	网站提供两种格式	分别进行不同的解析程序
*	网站数据缺失	加入判断并给用户提示
*	可视化结果难以区分	通过格式化创建特定文件名