### 本次大作业我主要通过三个阶段来完成，分别是：

1. 爬虫
2. 数据格式转换（json到csv）
3. 数据处理

下面分别对这三个阶段进行分析。

选取的第五个城市是合肥市

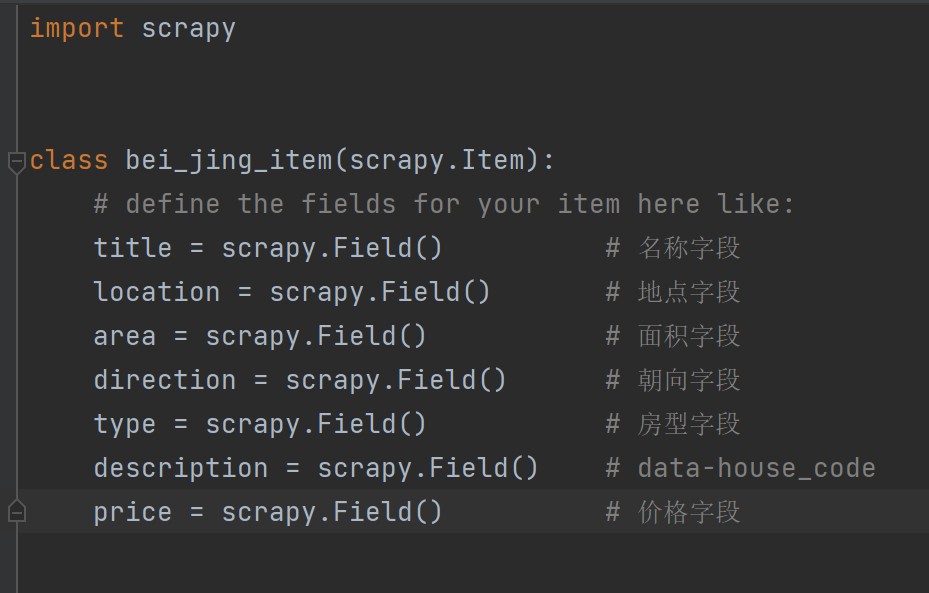
### 一、爬虫

由于一共要爬取5个城市的租房数据，所以使用了5个spider分别爬取这5个城市，并将他们的数据分别存到一个json文件里。

由于这5个城市要爬取的页面数据一样，就都使用了同一个item

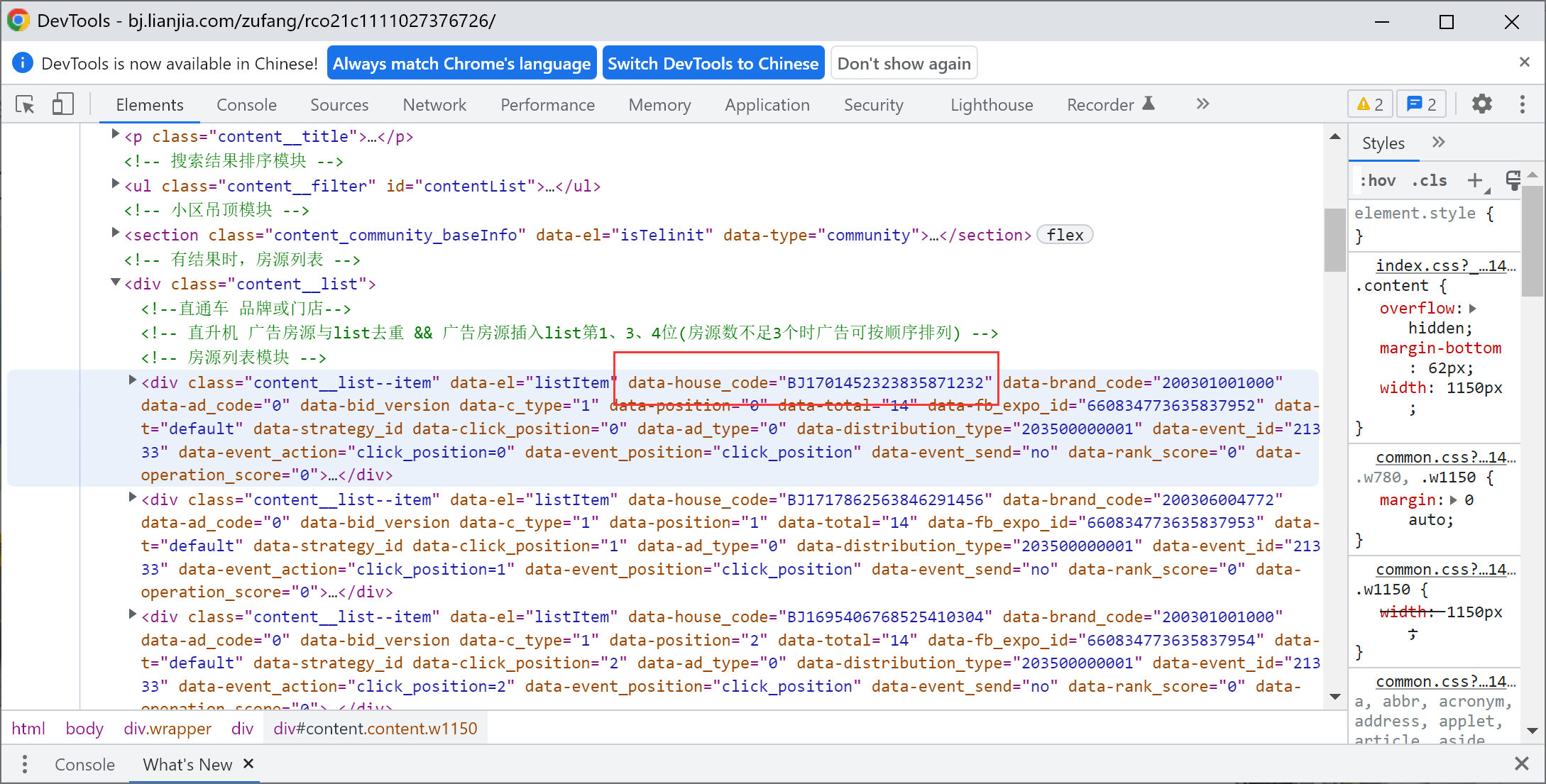
对北京市爬取的spider主要代码如下：

首先是item，



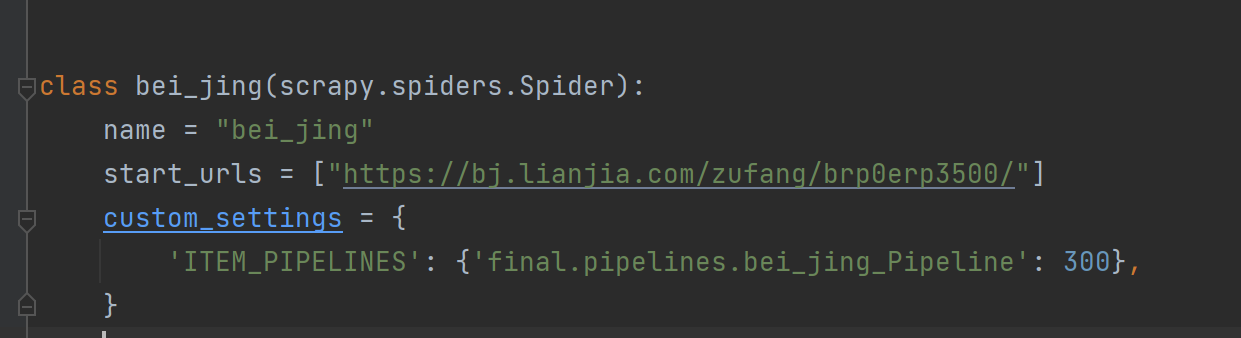
分别对应以下数据：





然后是spider的主要代码：

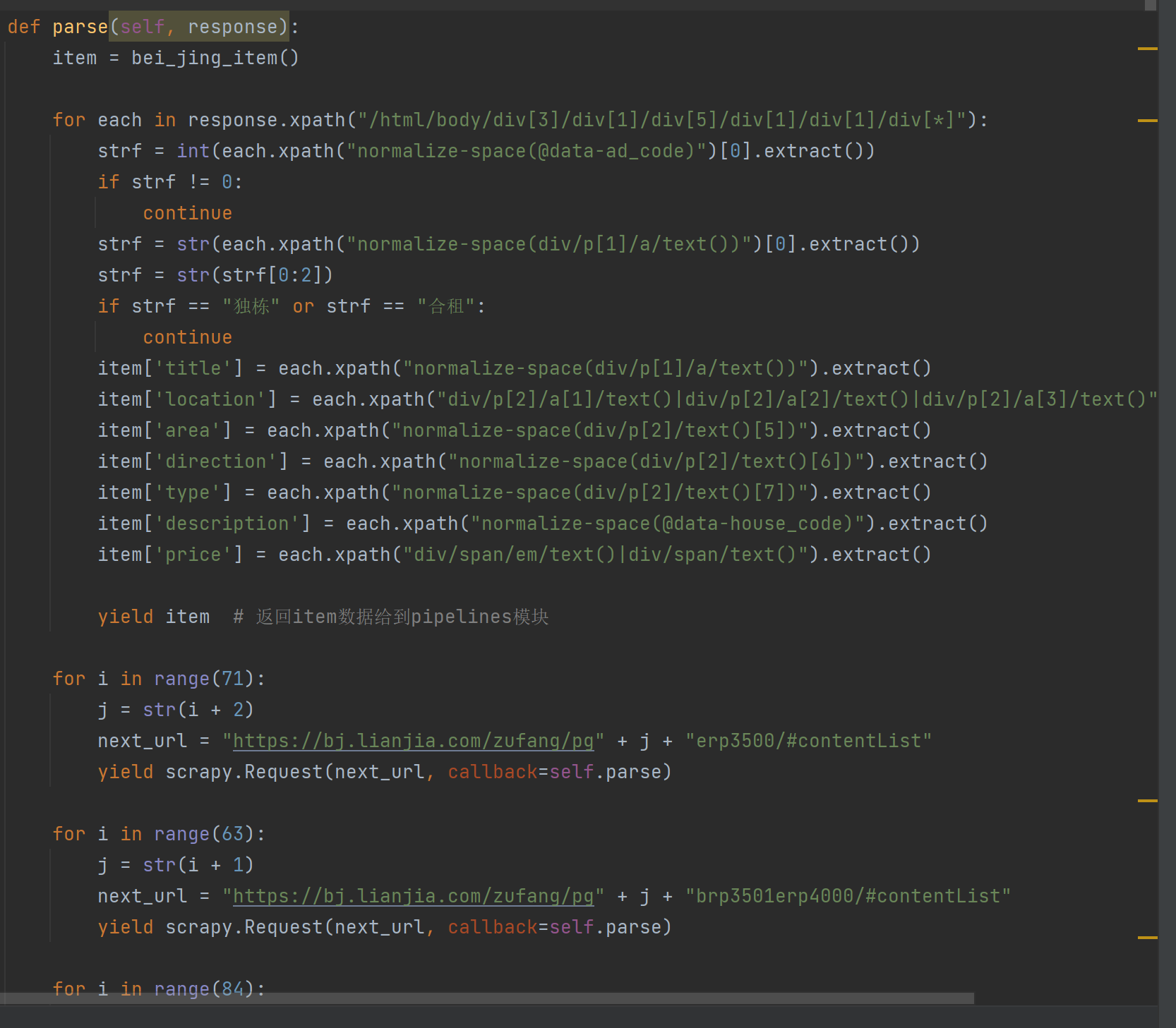
爬取的筛选条件是通过价格加朝向来控制一次显示的页数小于等于100，所以爬取的第一页限制条件是价格从0到3500元每月的。



下面是prase函数代码，首先就是设置xpath，这里通过xpath筛除了独栋和合租的数据，同时也把所有广告数据也过滤了，只保留了整租的数据

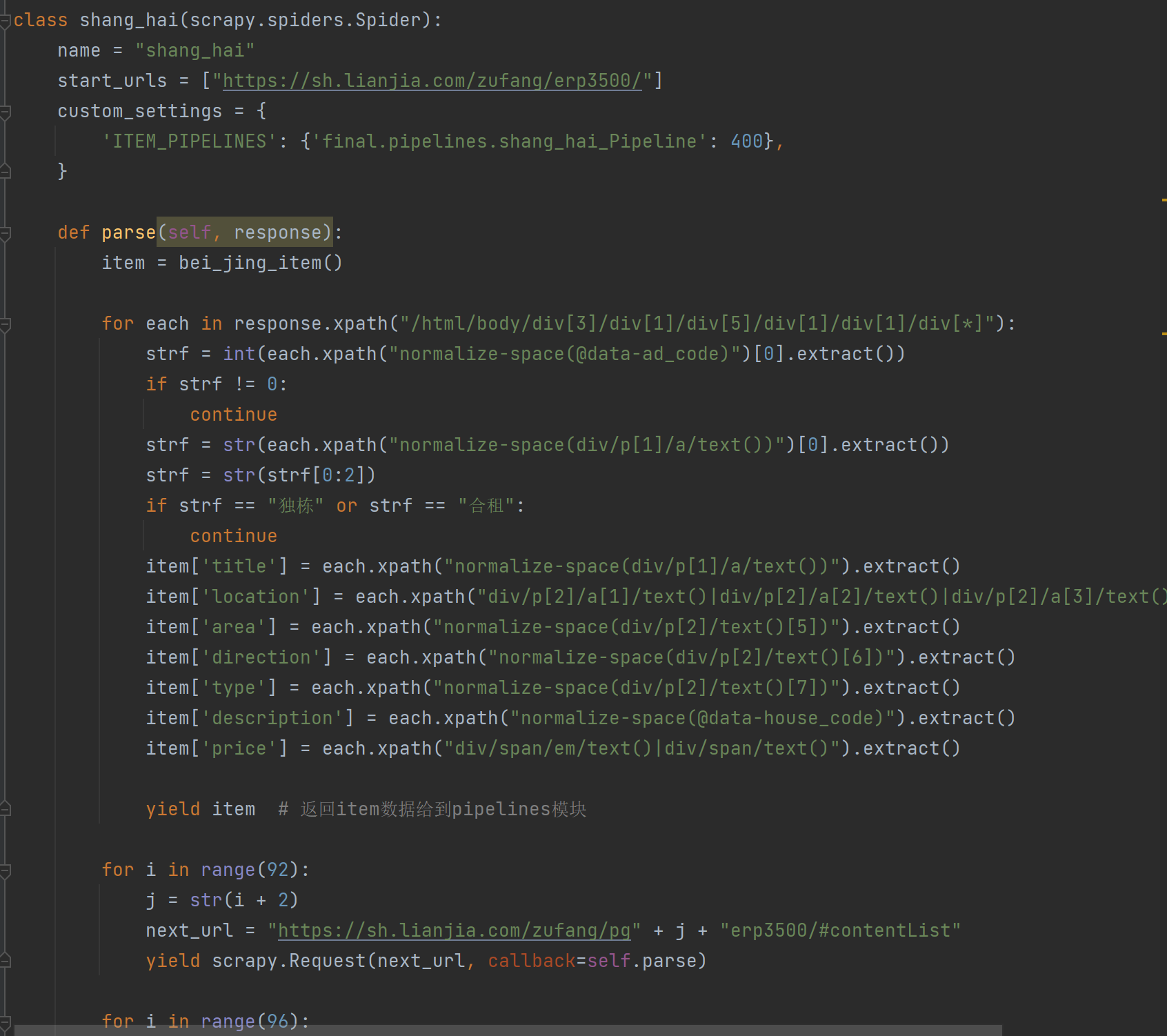
然后就是提取信息到item里面

当设置的第一页爬完后，使用了for循环和yield scrapy.Request()函数来进行下一页读取，这里后续页面都是通过设置价格区间来保证数据不超过100页的，后续代码基本都是for循环爬取下一页，就不在报告里展示了



由于后续四个城市的spider代码基本一样，就不说明了，展示主要代码如下：

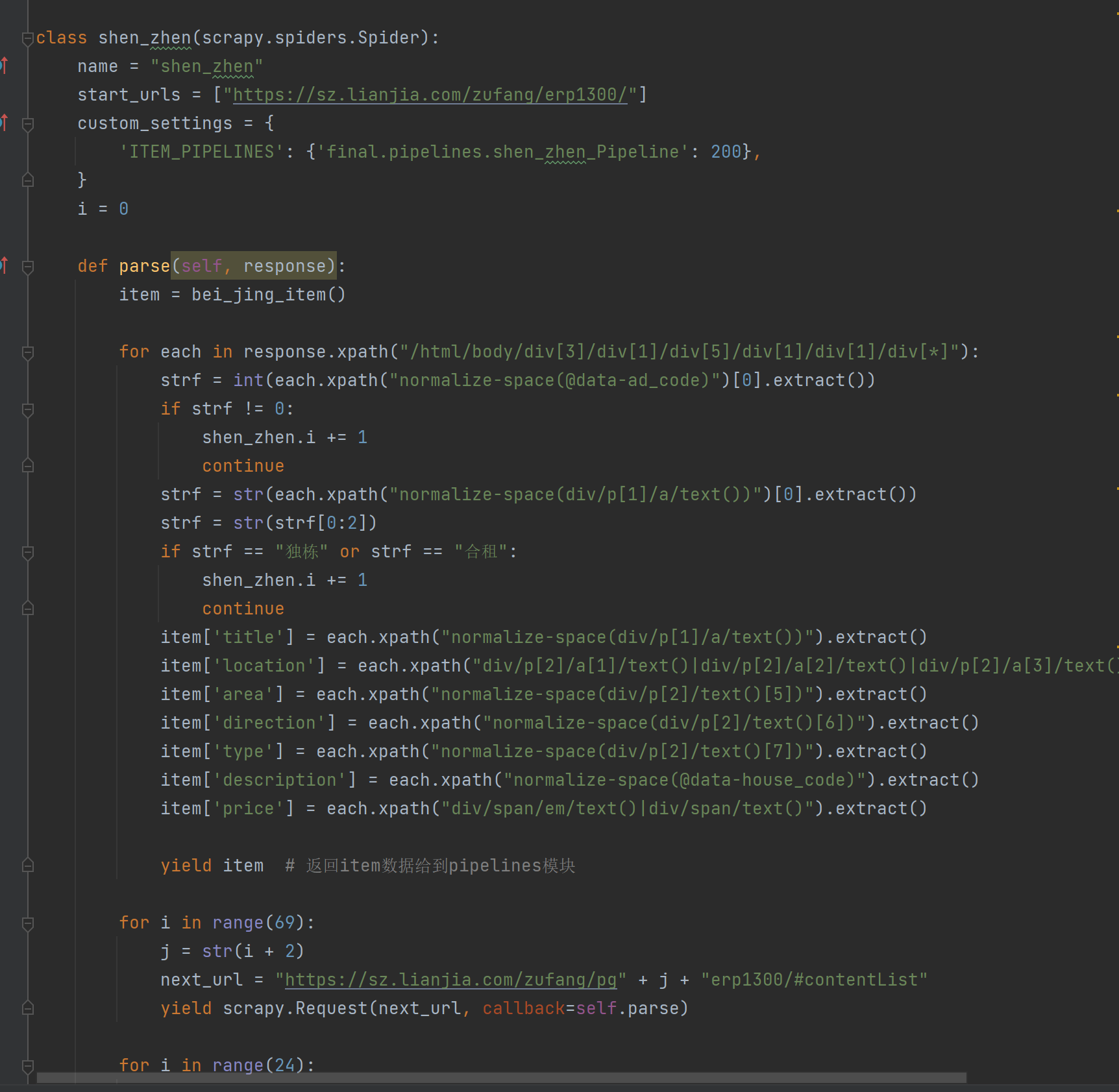
上海的：



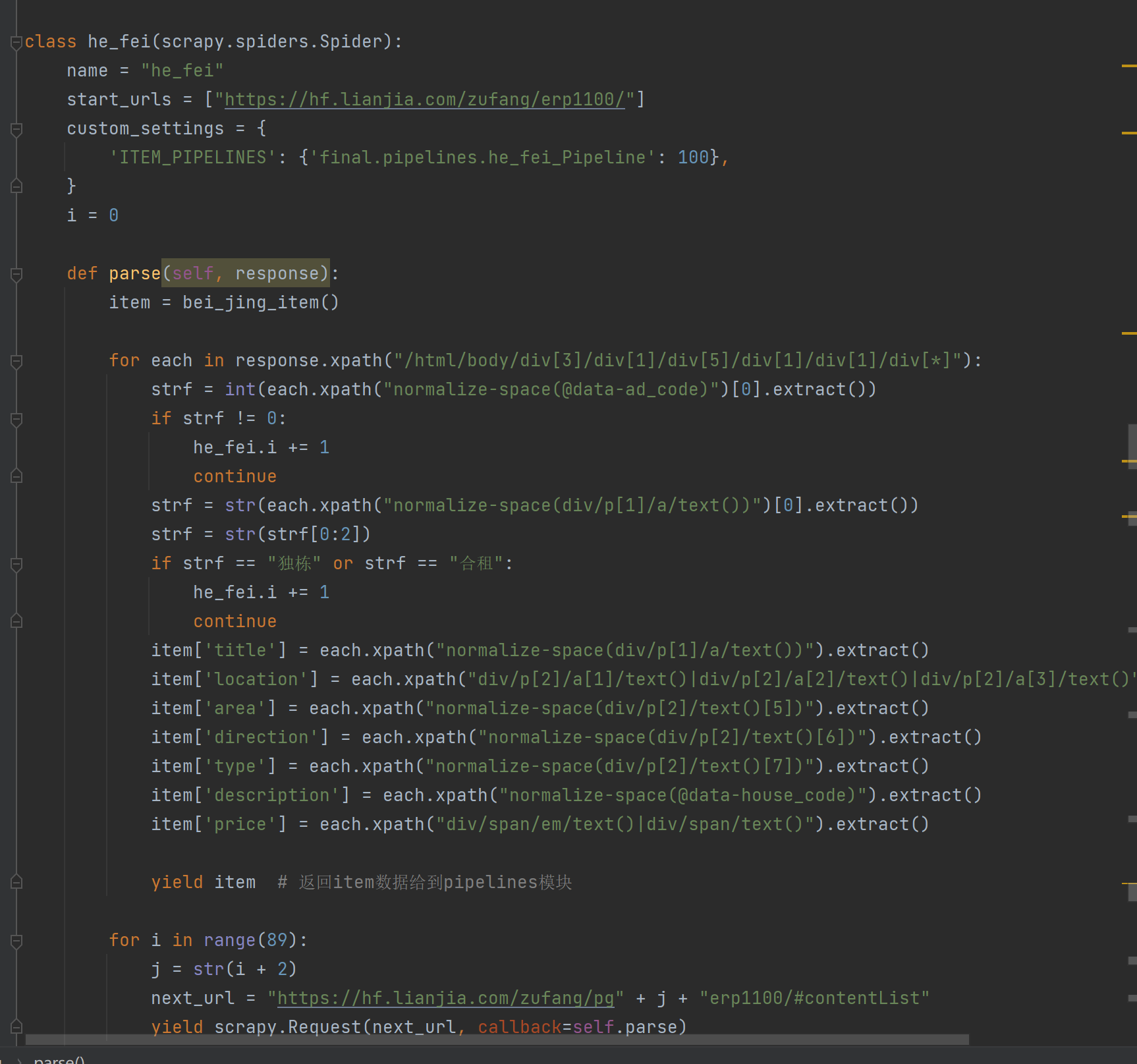
广州的：



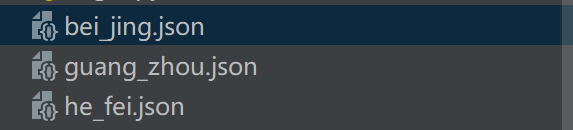
深圳的：



合肥的：



爬取下来的json数据分别保存在：



其中北京市一共有33964条数据

上海市一共有30170条数据

广州市一共有84229条数据

深圳市一共有33600条数据

合肥市一共有25611条数据

爬虫文件名称是spider.py，需要运行begin.py来启动爬虫

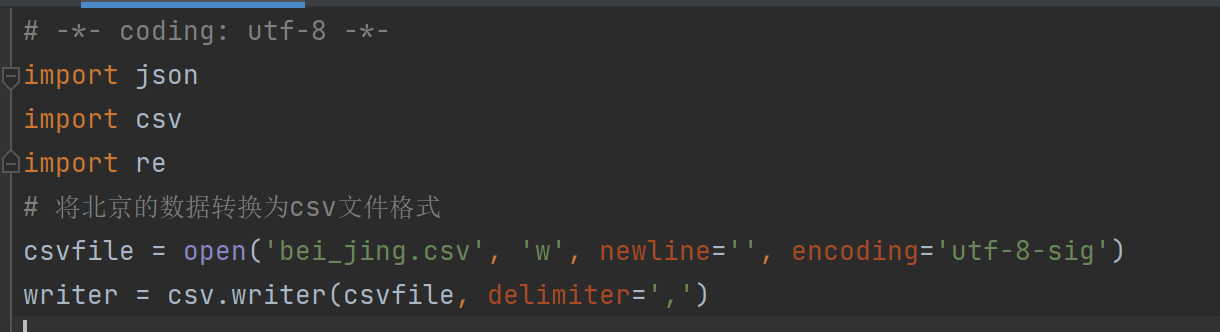
到此，爬虫阶段结束，下面说明一下数据格式转换。

### 二、数据格式转换

五个城市的代码都在convert\_data目录下的convert.py里，得到的csv数据文件也在此目录下

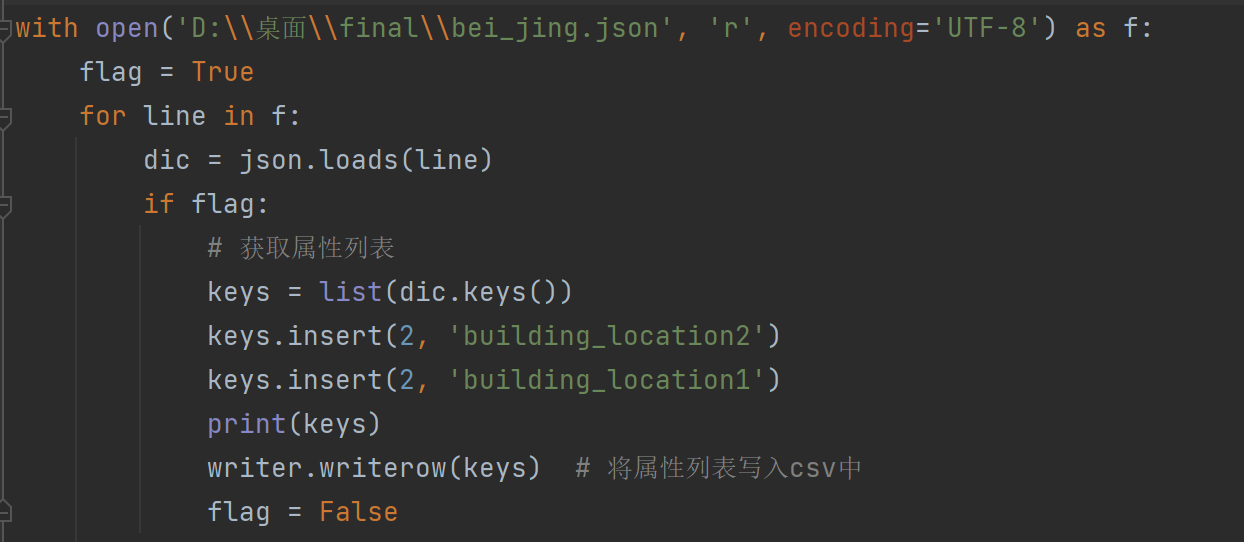
由于五个城市的操作类似，就以北京市的代码进行说明：

首先是设定输出文件的操作：



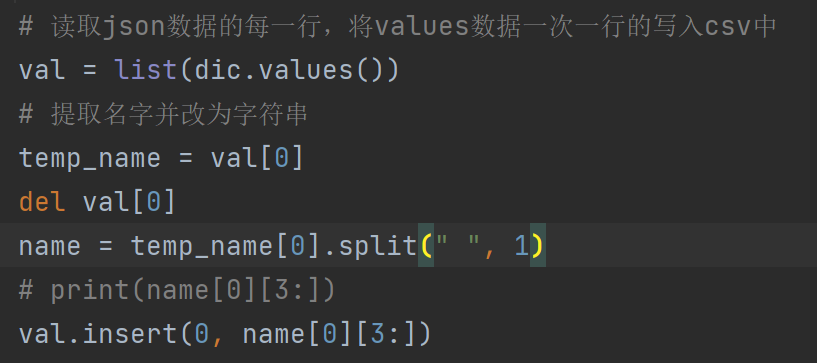
然后打开爬取到的bei\_jing.json文件，通过一行一行的读取，然后对每一行进行修改在写入到csv文件

首先就是设定属性列表，这里添加了两个属性building\_location2和building\_location1，是为了后续把地点字段拆成三个字段做准备

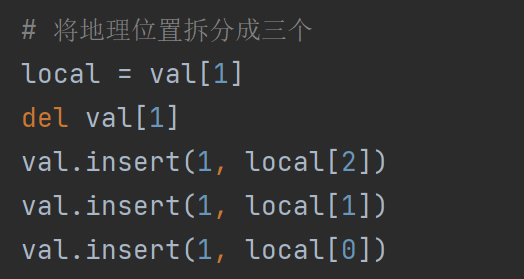


设定好属性后，就开始对每一行数据进行修改

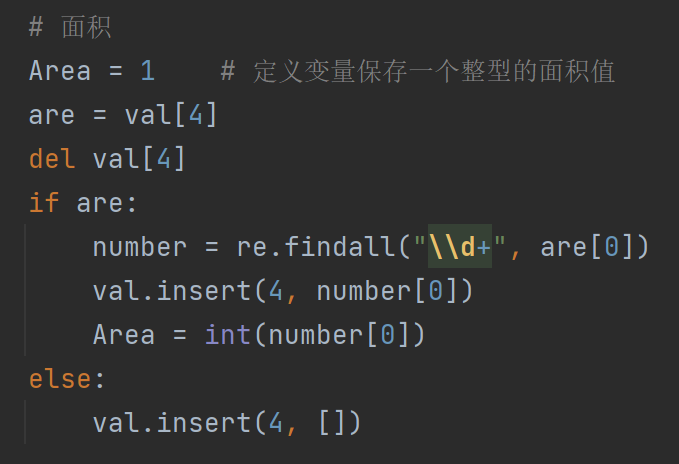
由于爬取到的名称字段有冗余信息，于是通过spilt函数来获取需要的字段：



然后是地点列表拆分成三个字段：



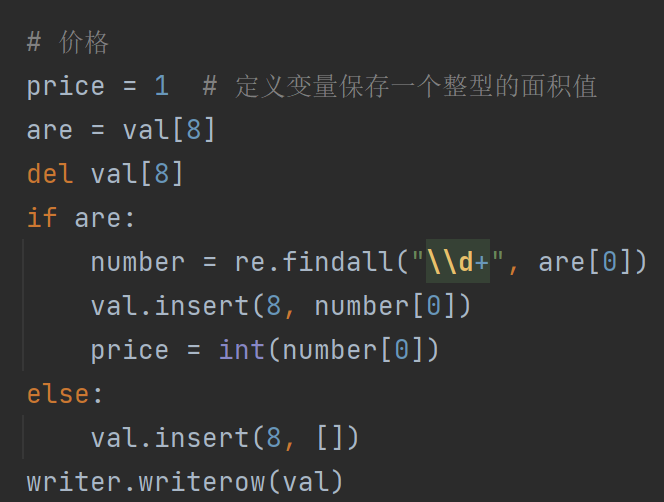
之后是面积字段，这里对面积进行了一次判断，如果该字段没有值，就插入空列表：



然后分别是朝向，房型，房子号码：



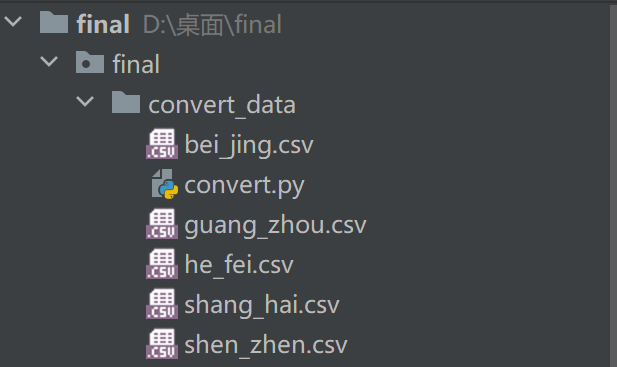
最后是价格，也有空值的判断处理：



之后就是将修改后的一行数据写入csv文件

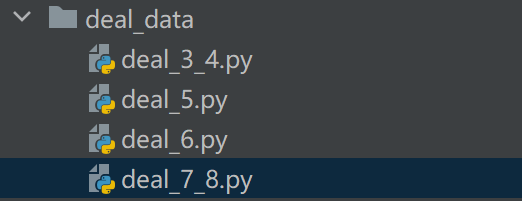
以上，就是北京市数据格式转换的内容，其他五个城市的内容类似，就不赘述了。

这是文件位置：



### 三、数据处理

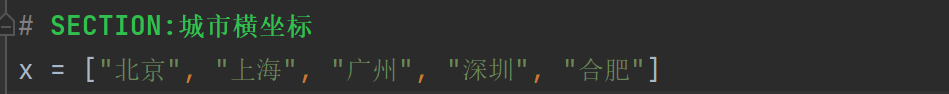
这个部分主要是针对每个问题的要求进行了代码编写，分别是



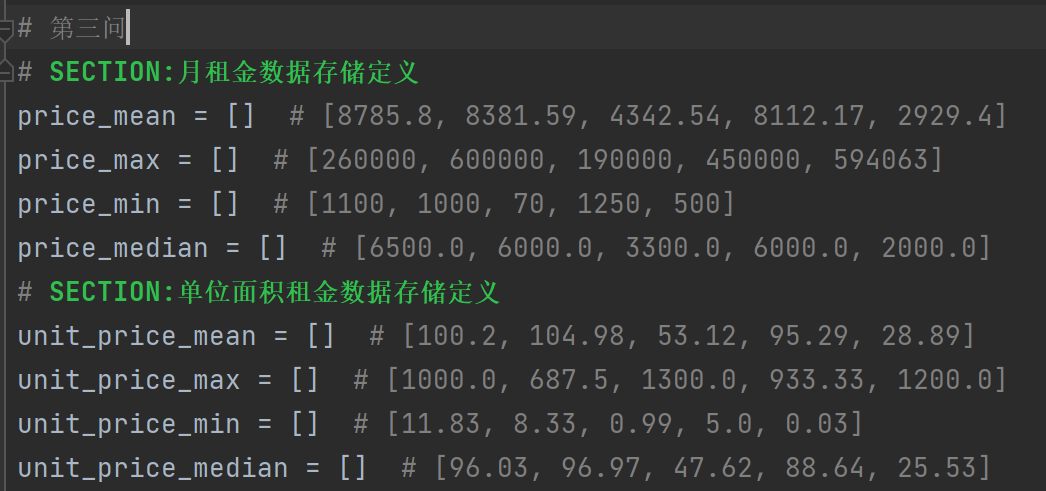
按照名称里的数字对每一问进行了解决

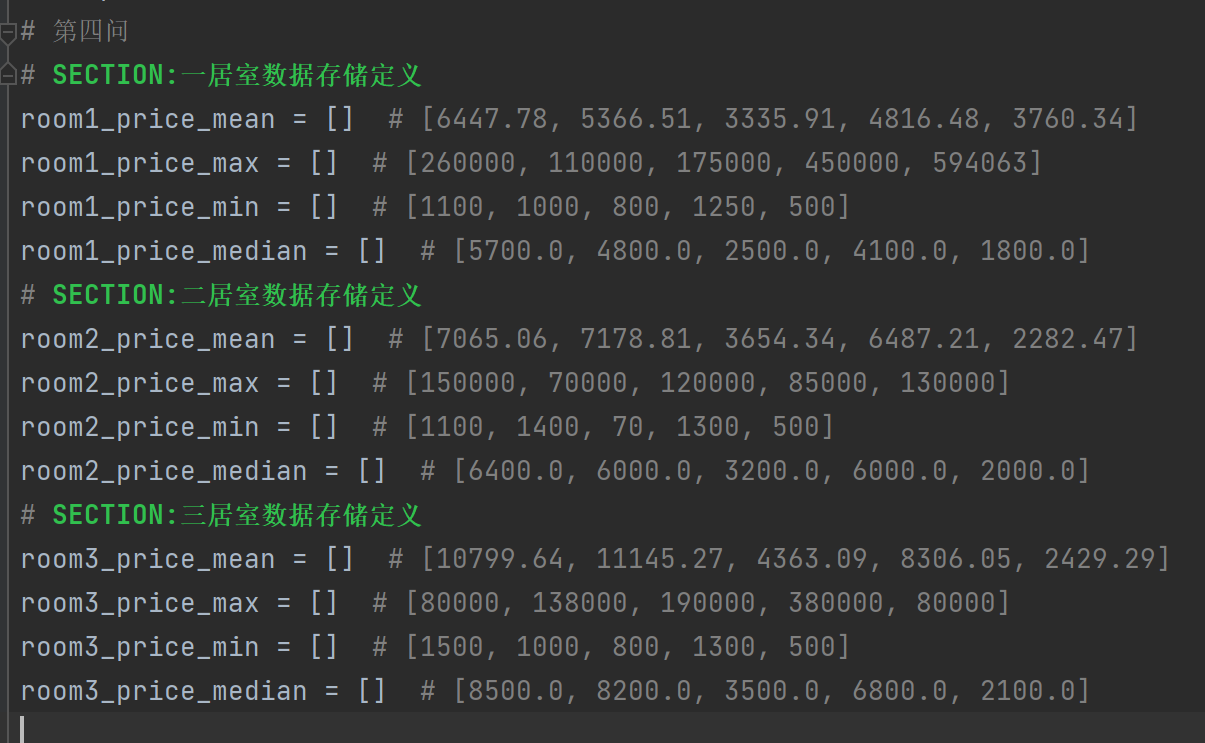
### 首先说明一下deal\_3\_4.py的代码：

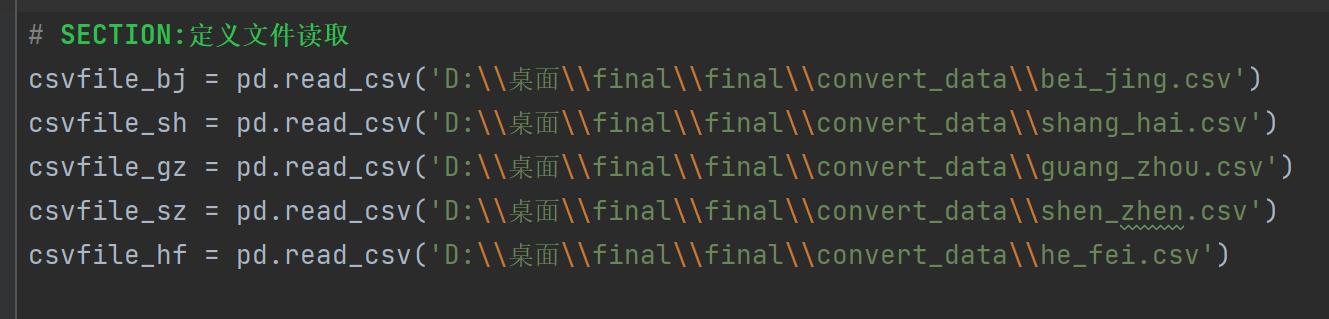
首先是定义了一个列表保存城市的名称，如无特别说明，本作业里面的五个城市就是以下顺序



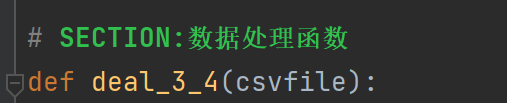
然后是定义了一些数据结构用来保存计算后的数据，每一项后面都给出了计算的结果，这样在绘图或调试时，就可以不用在跑一遍计算过程了，节省时间：





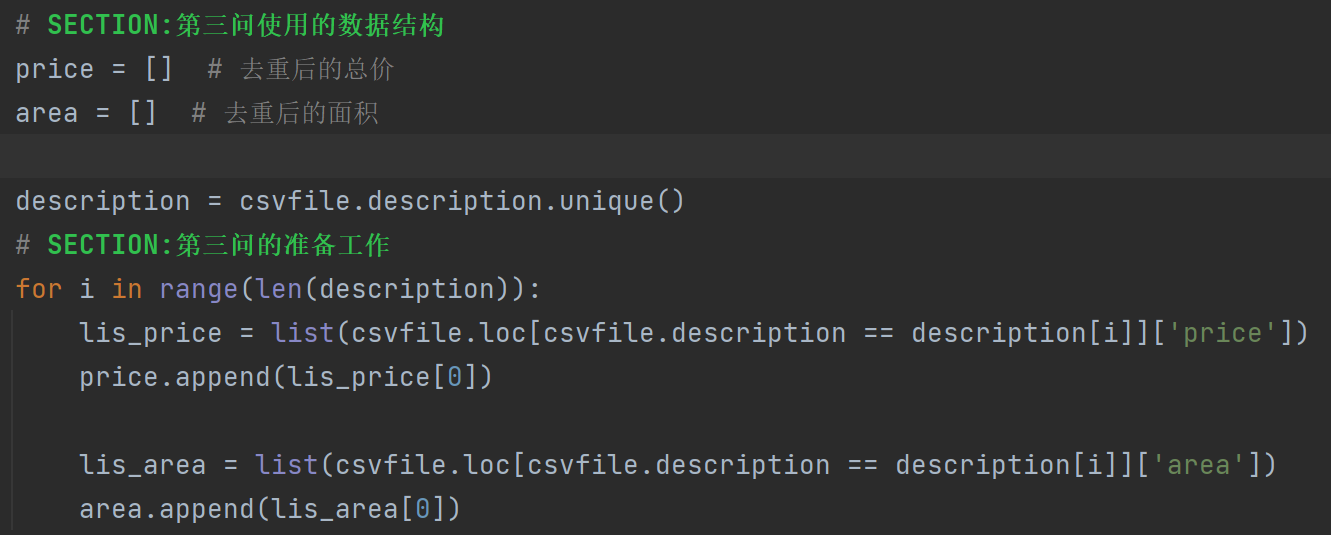


由于要处理五个城市的数据，故设计了函数来避免重复代码，这里是把第三问和第四问的代码写在这一个函数里面了，我用注释进行了标记：

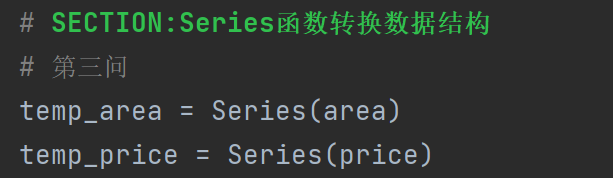


对于第三问，要比较 5 个城市的总体房租情况，包含租金的均价、最高价、最低价、中位数等信息，单位面积租金（元/平米）的均价、最高价、最低价、中位数等信息。

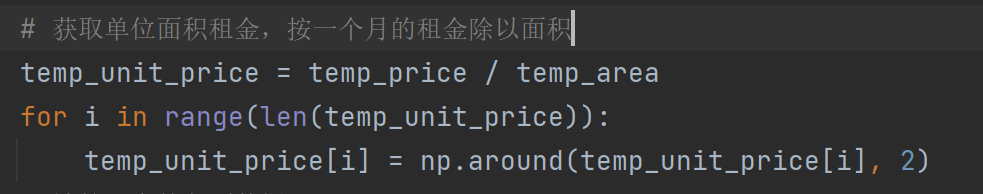
所以就需要获得每个城市的每个房子的价格和面积，但需要进行去重工作，这里是通过csv文件里description列的data-house\_code来去重的：



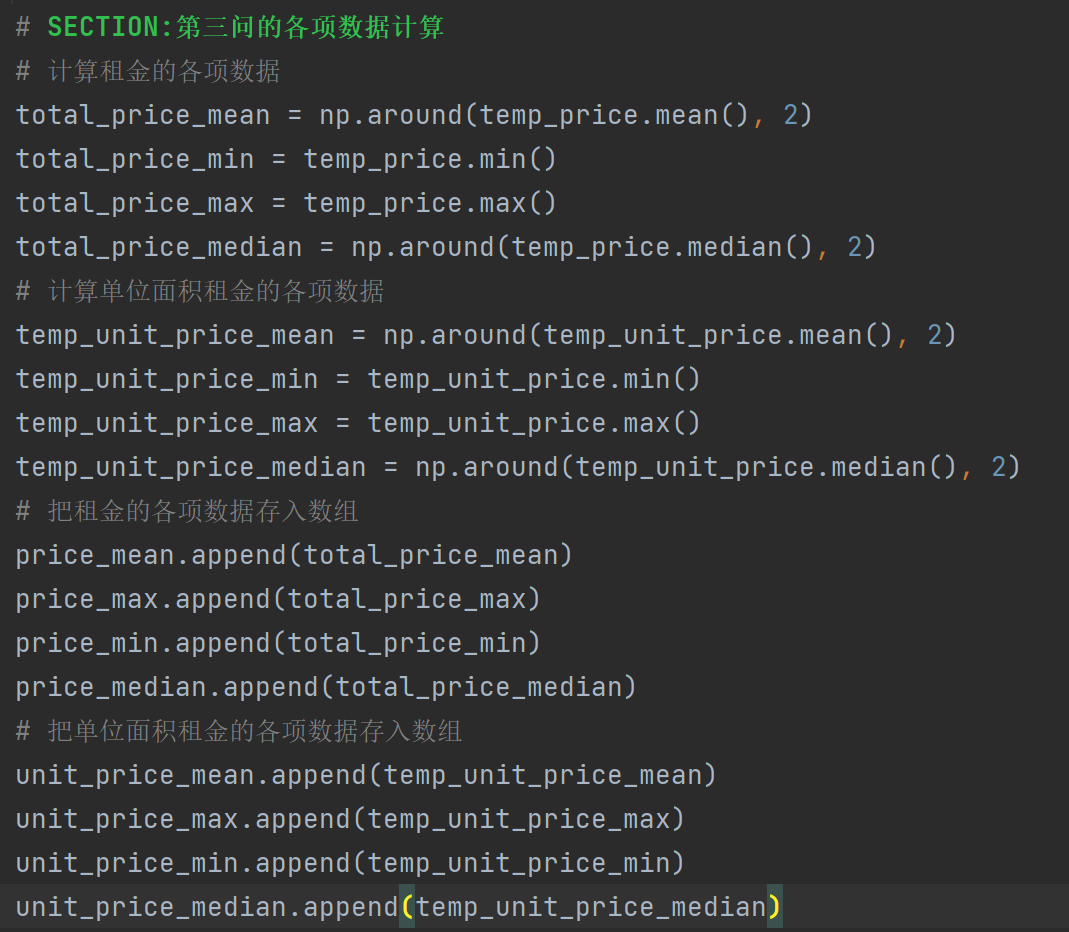
然后把得到的去重后的数据转换为Series类型：



还要获取单位面积的租金，这里是按一个月的租金进行计算的：



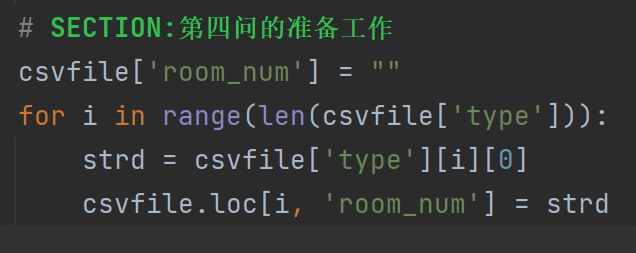
然后就是计算均值等数据，并保存：



### 下面说一下第四问的过程：

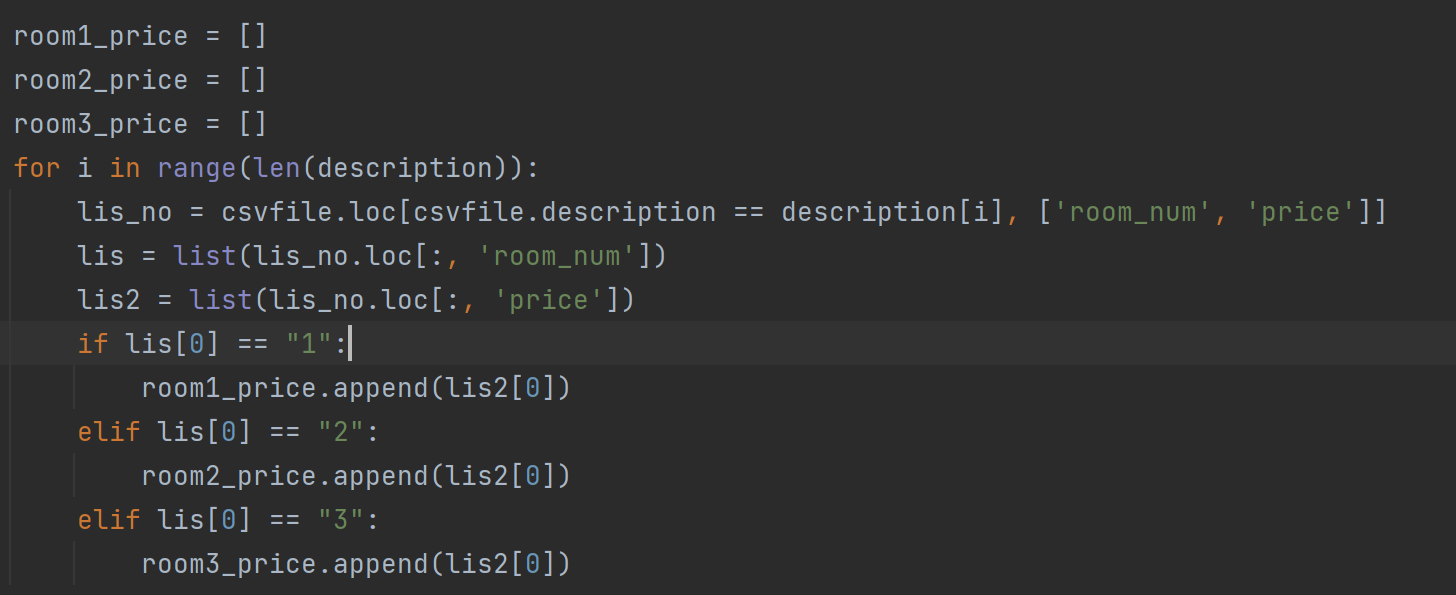
因为要比较 5 个城市一居、二居、三居的情况，包含均价、最高价、最低价、中位数等信息。

所以就需要知道每一个房子的卧室数量，这里新增了一列room\_num用来保存卧室数量：

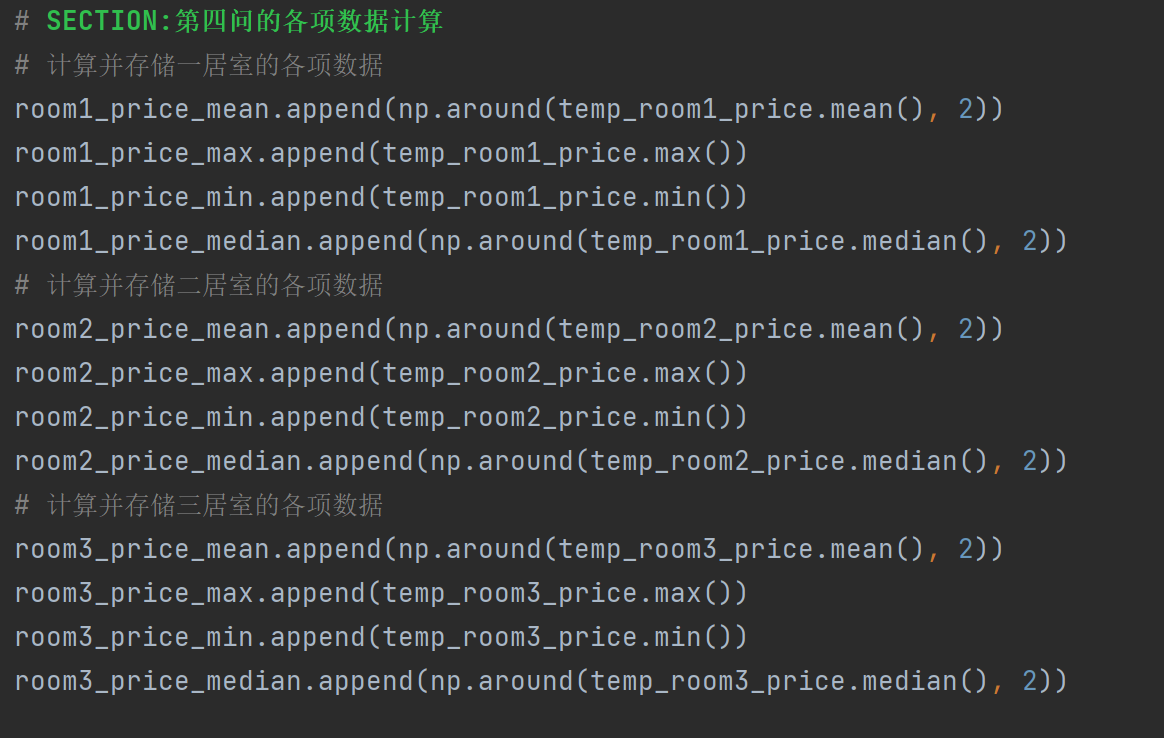


然后先要对数据进行去重处理，这里是也是通过data-house\_code来去掉重复的数据，并提取出'room\_num', 'price'两列数据

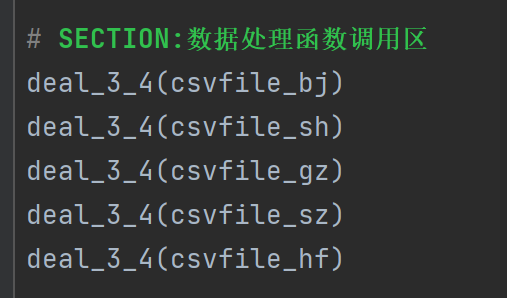
然后通过对room\_num进行判断，来把对应的price保存在相应的列表内：



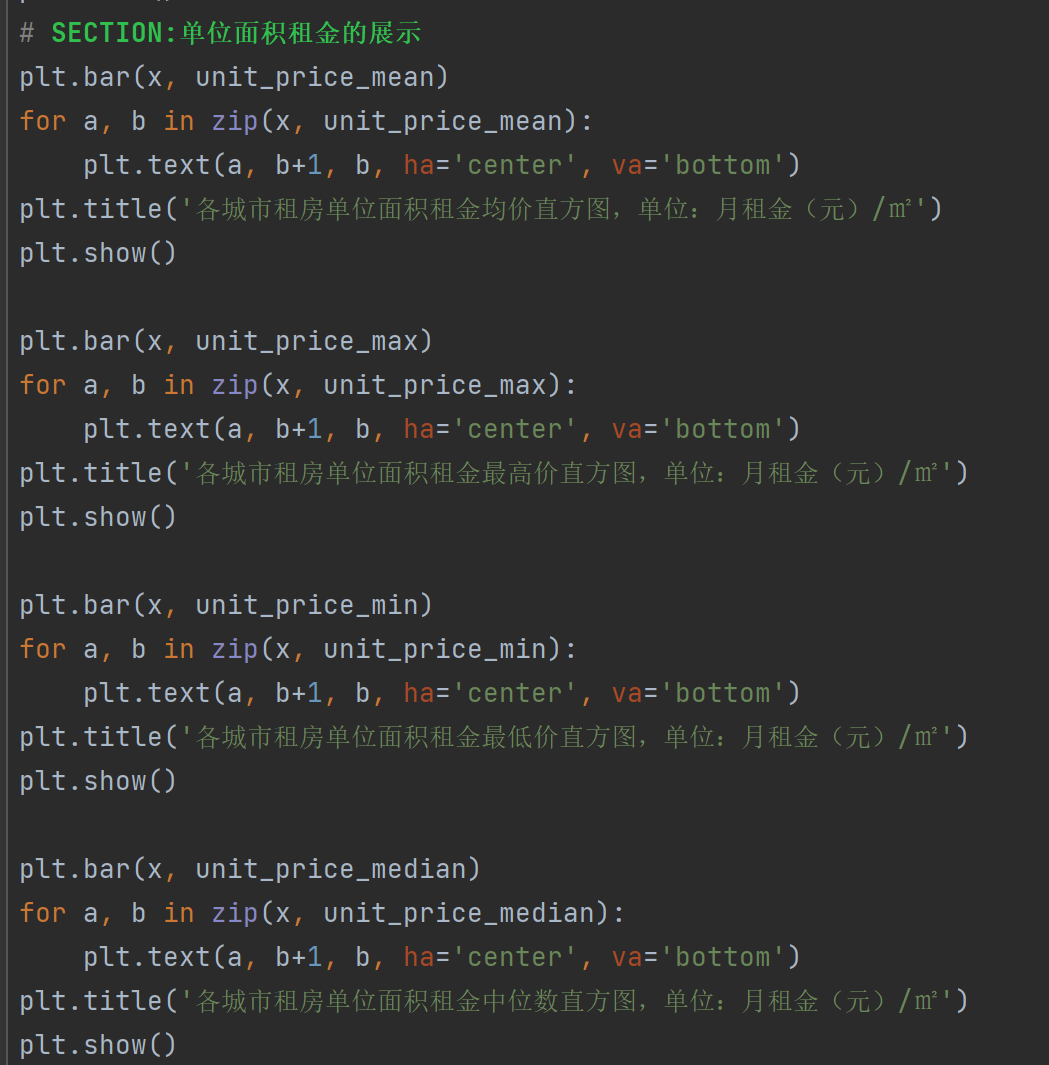
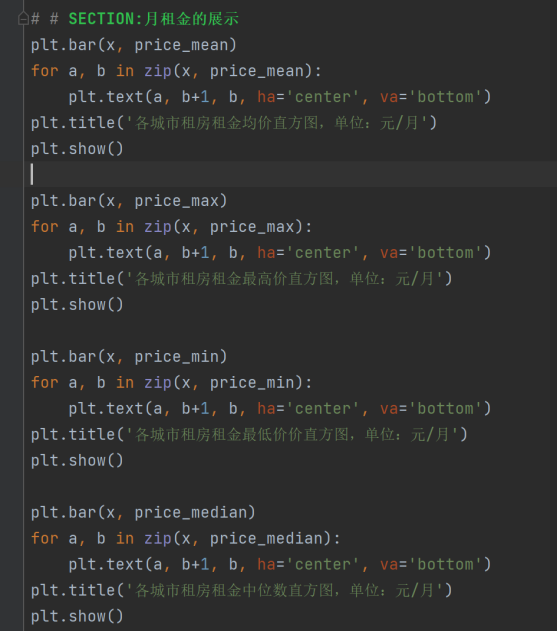
之后就是计算各项数据并保存：

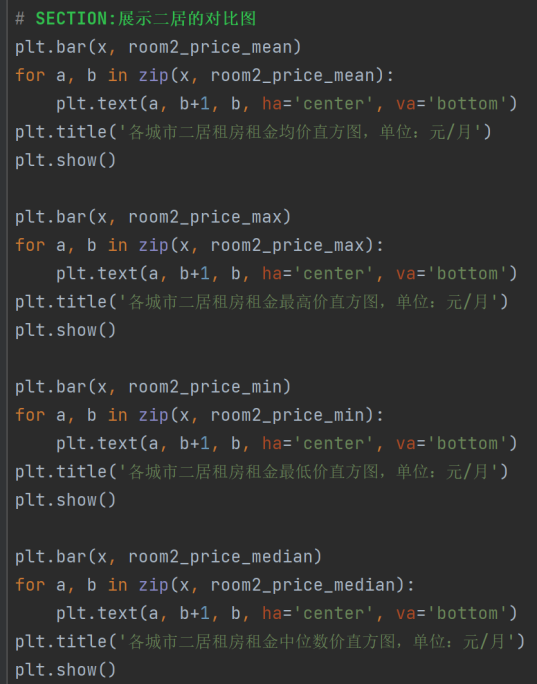
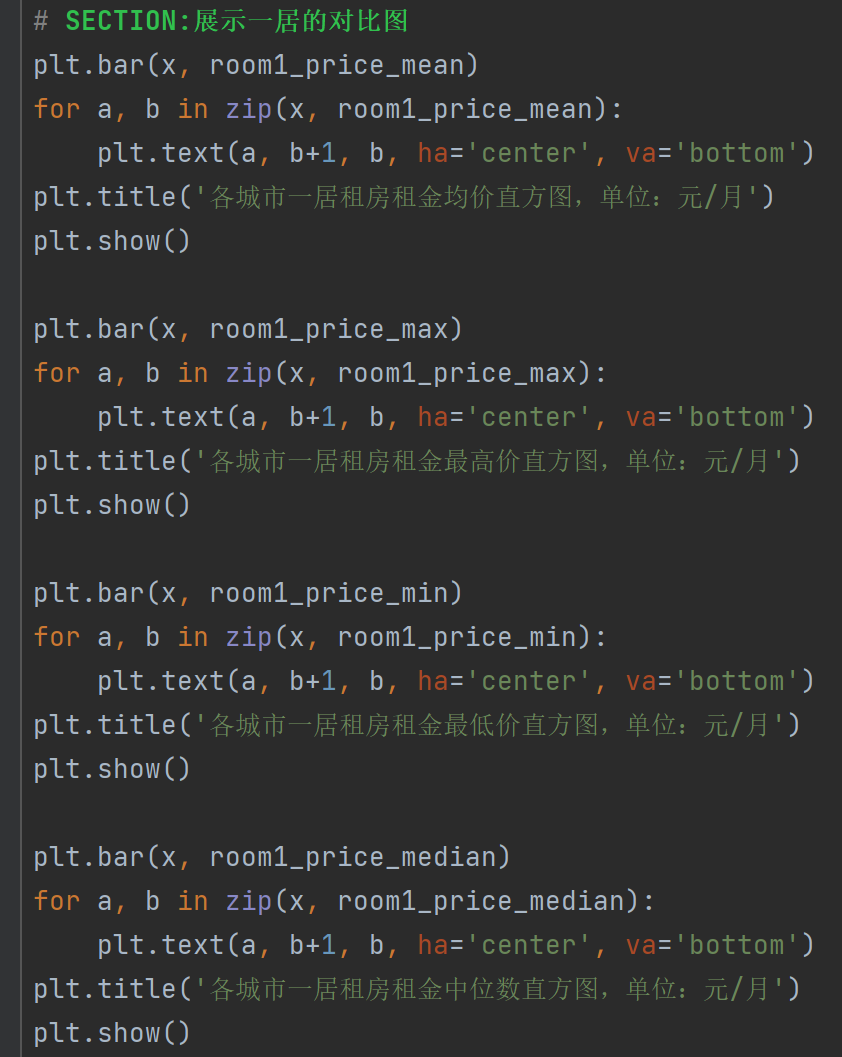


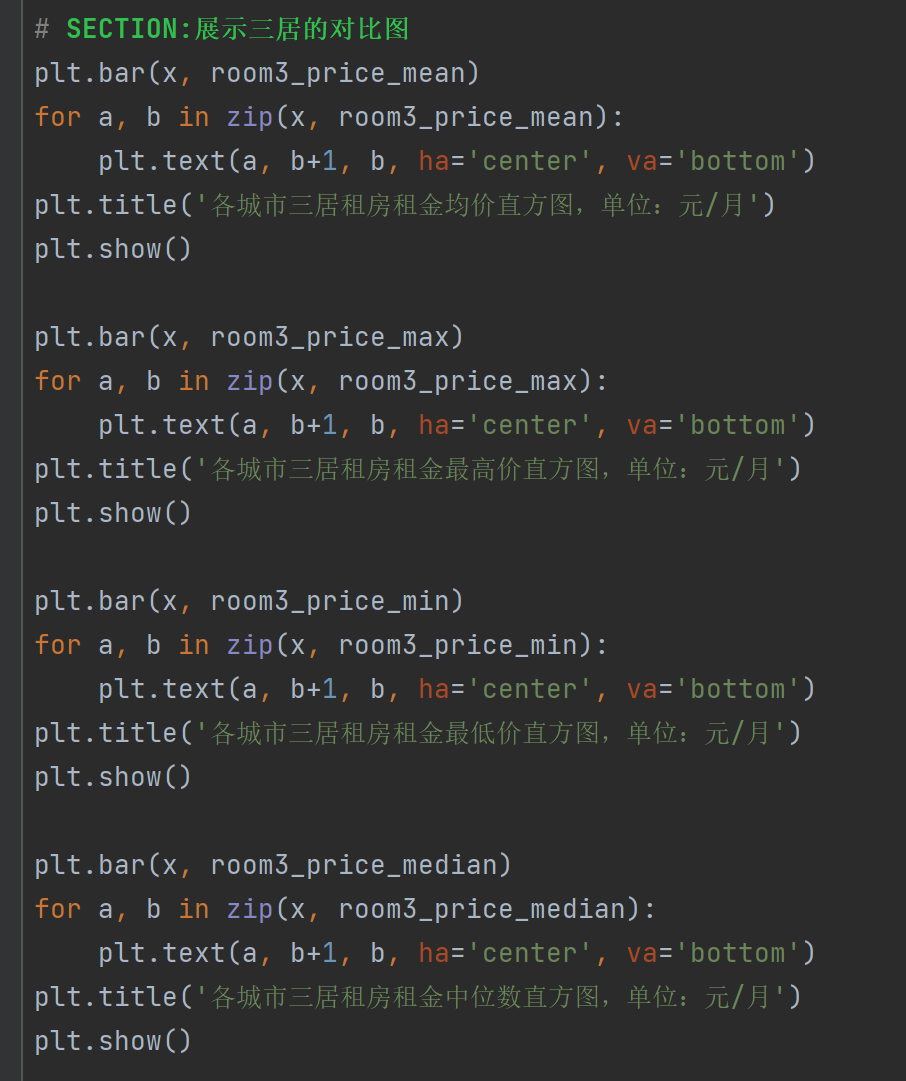
函数定义好之后，就是调用，只有一个参数，就是传入的各城市的DateFrame：



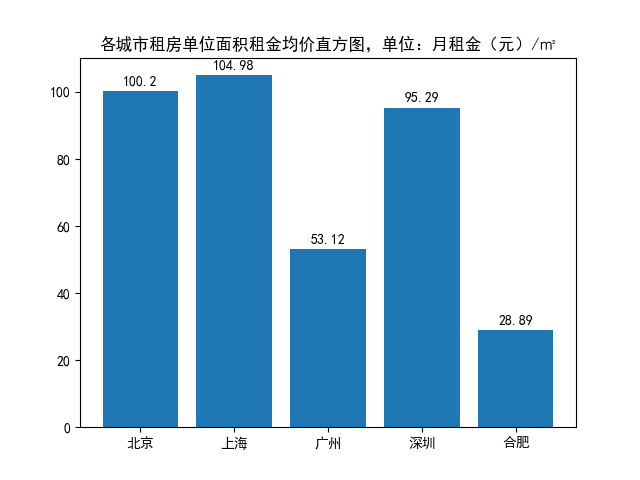
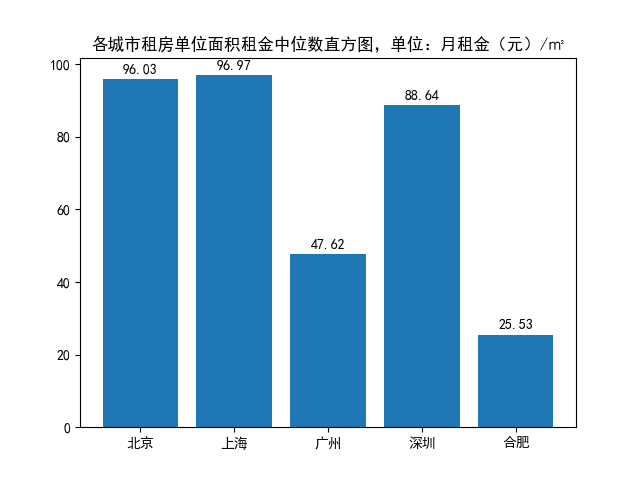
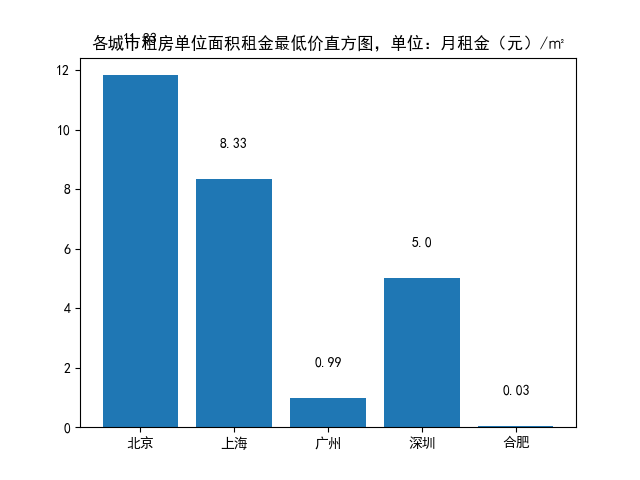
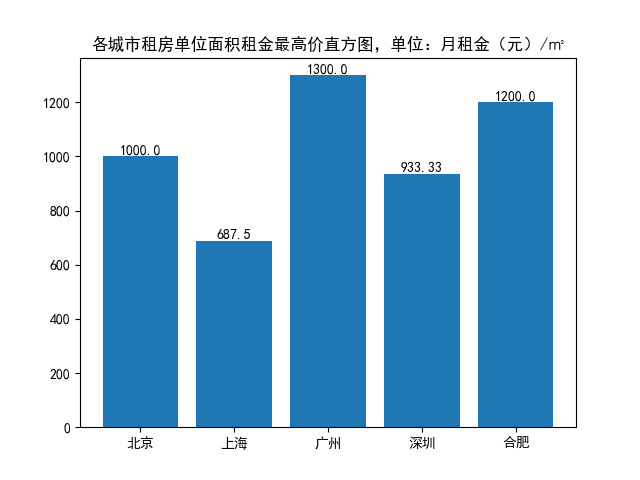
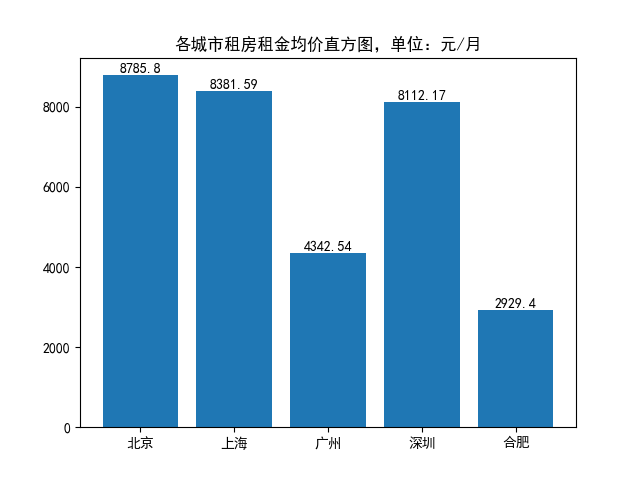
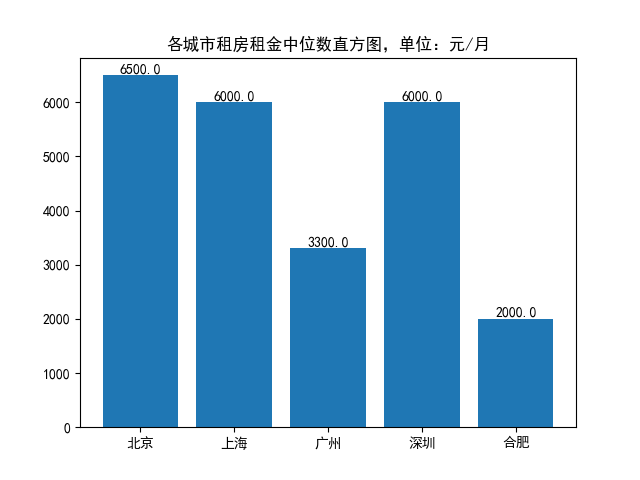
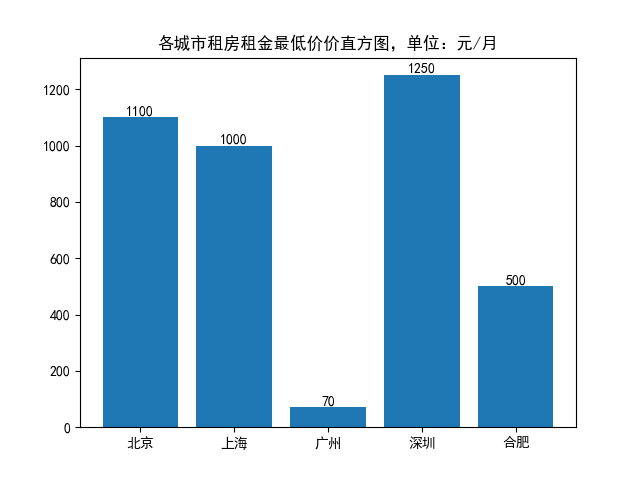
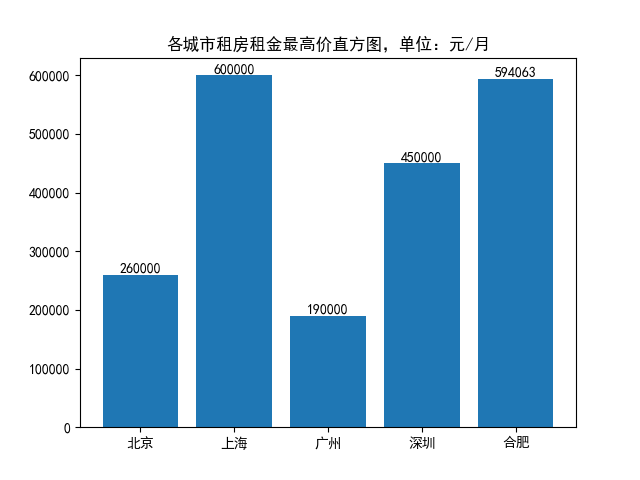
最后就是把得到的数据进行作图展示：

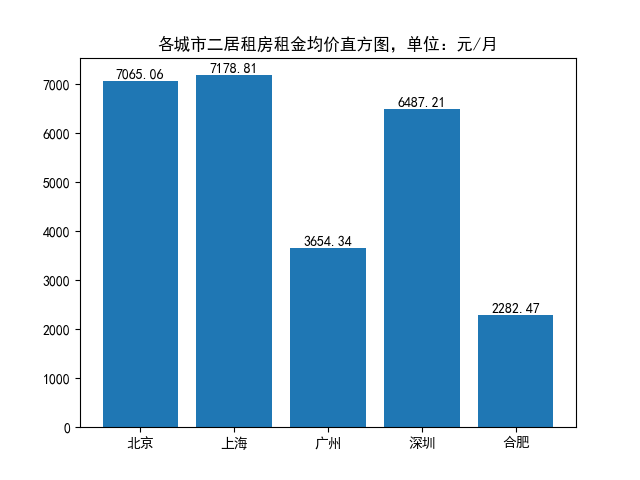
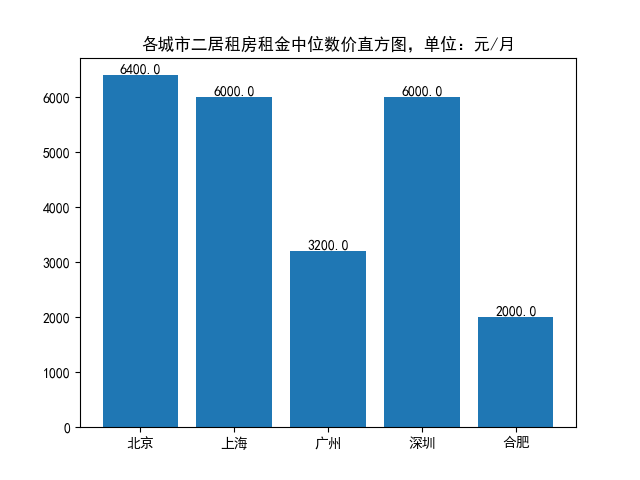
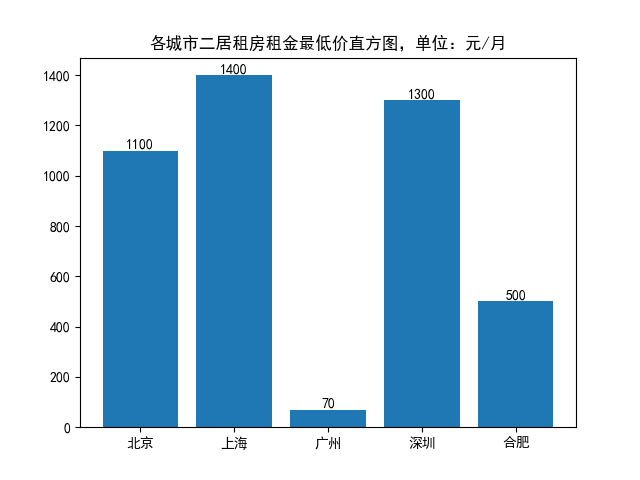
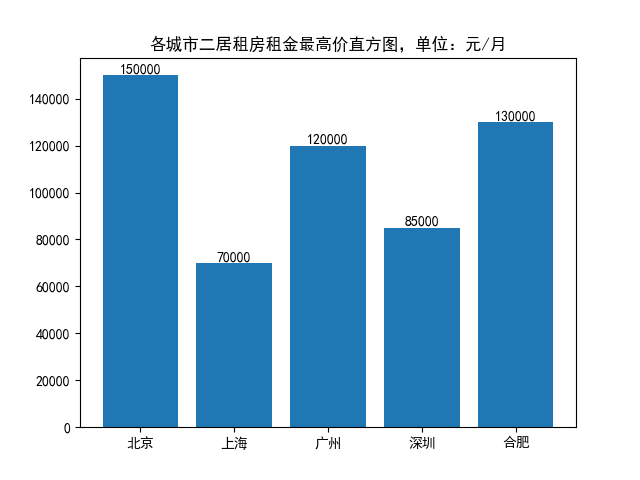
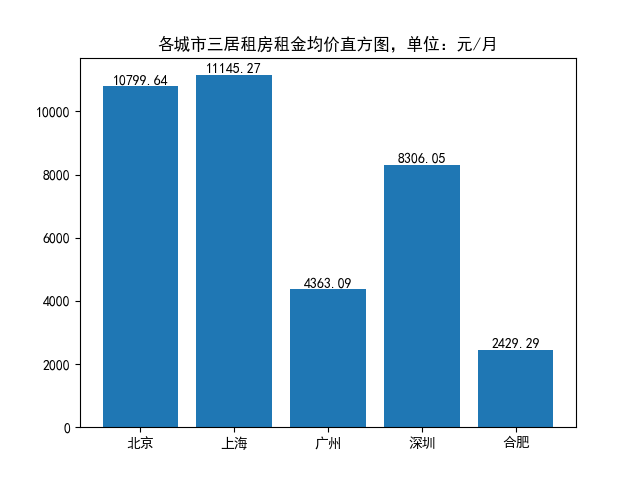
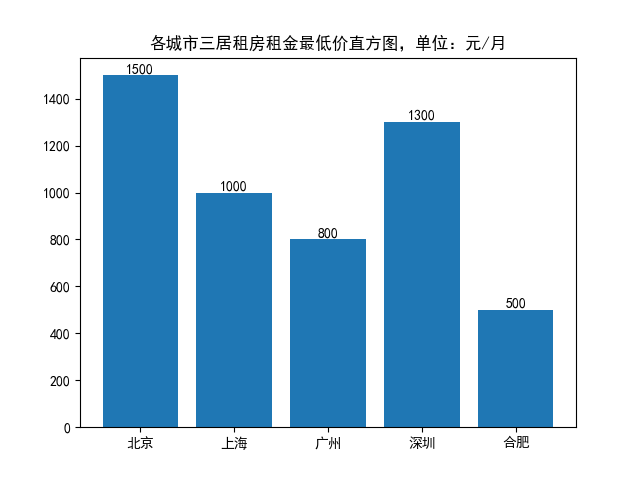
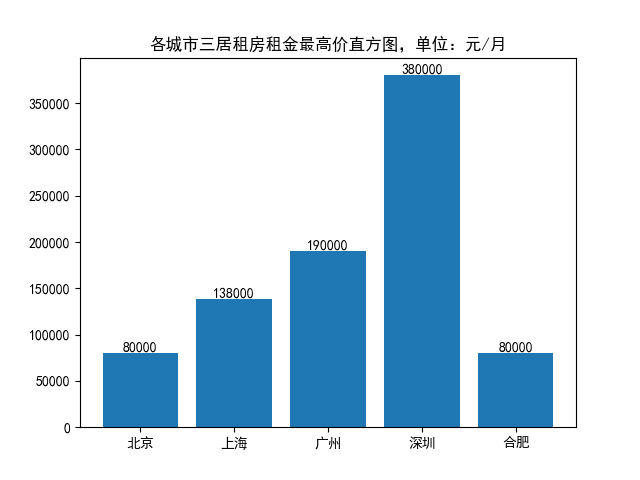
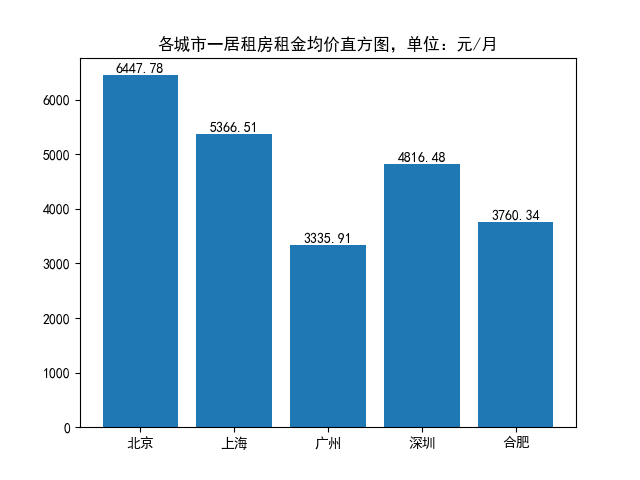
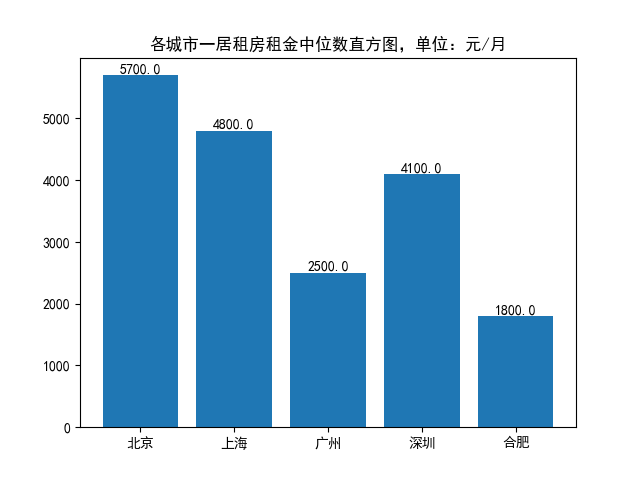
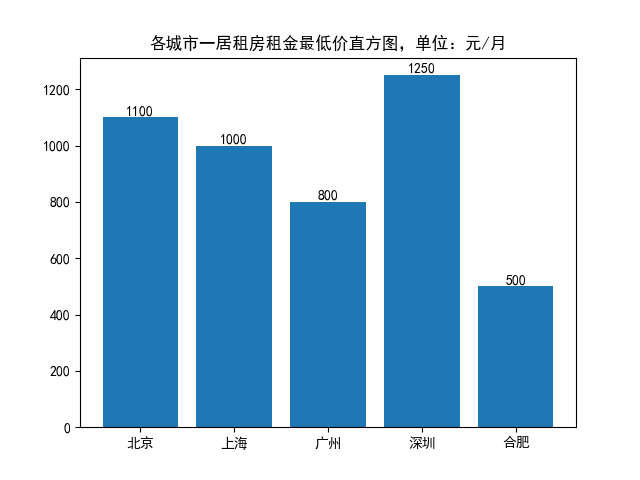
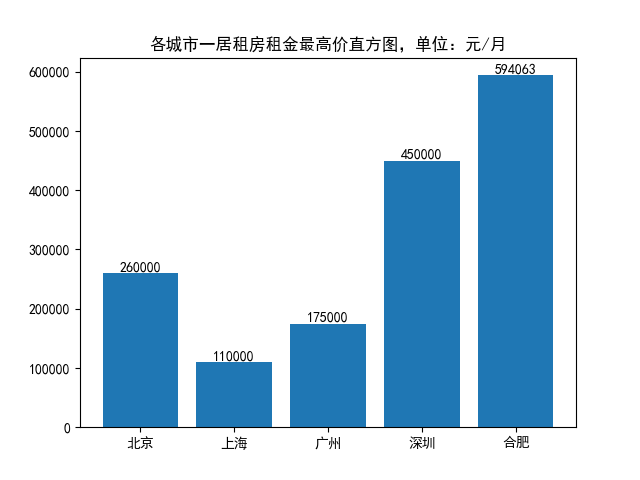






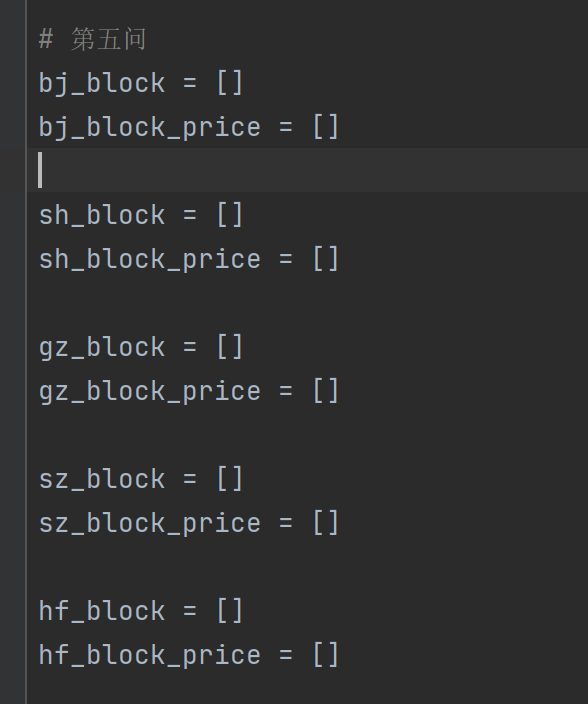
得到如下20张图：



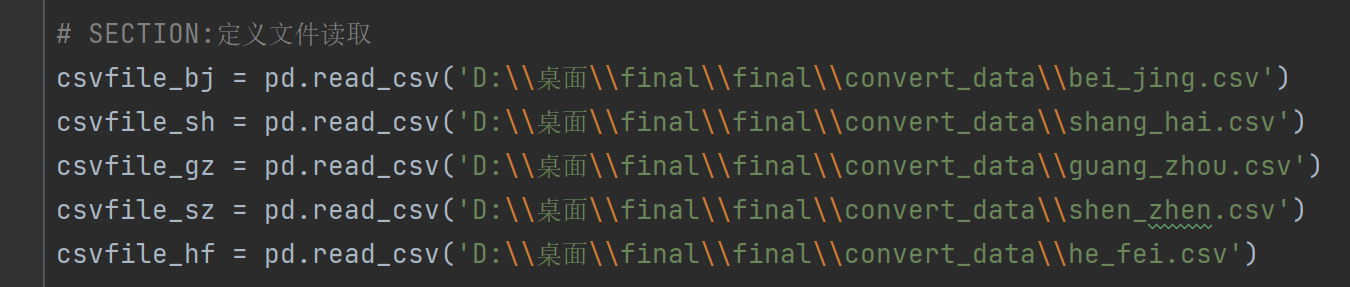


对于第五问，需要计算和分析每个城市不同板块的均价情况，并采用合适的图或表形式进行展示。

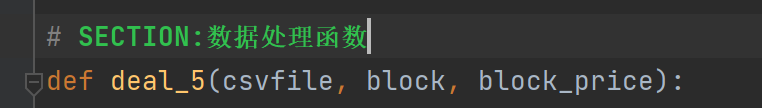
所以就要获得每个城市有哪些板块，以及每个板块里的每个房子的价格，所以设计了如下数据结构来保存这些信息：



依旧是定义文件读取：

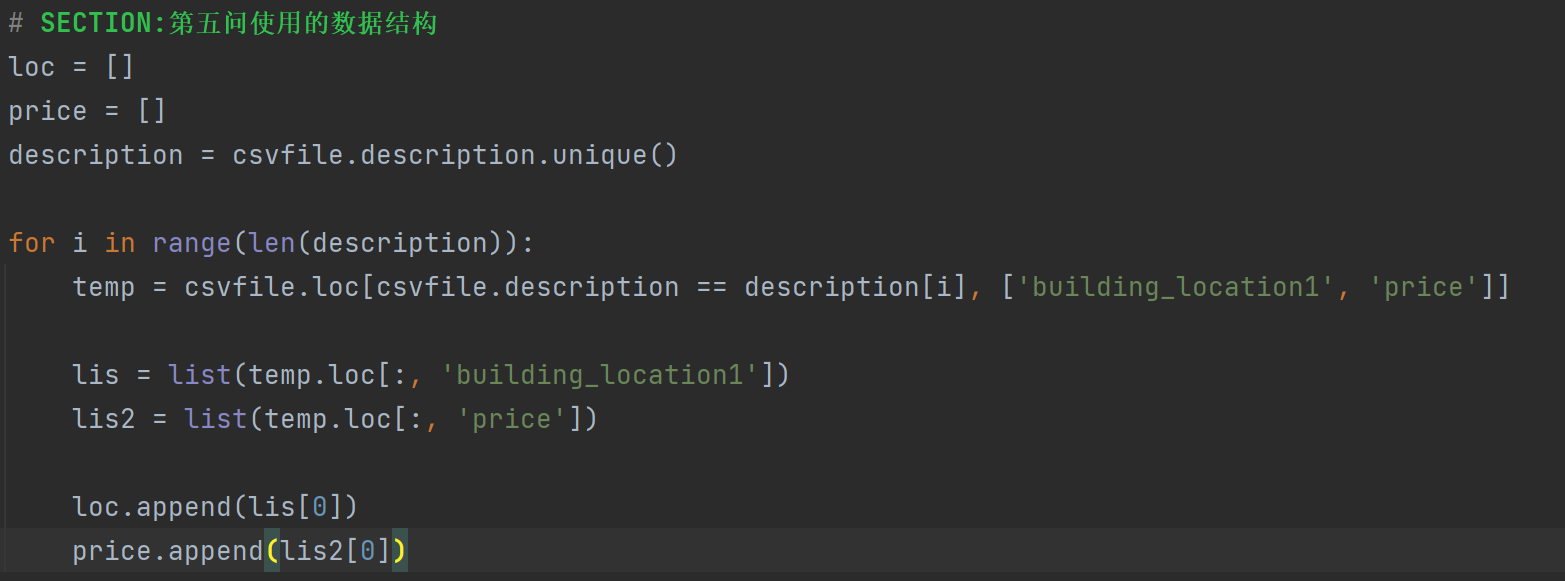


由于五个城市的操作一样，就定义了一个函数来避免重复代码：

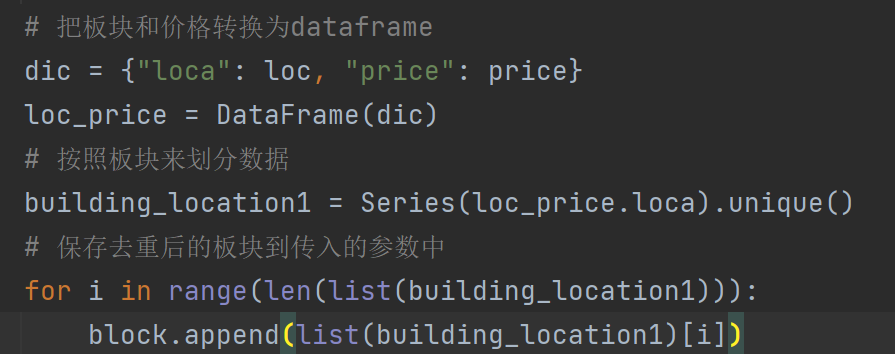


有三个参数，第一参数是对应城市的DateFrame，第二个参数是对应城市用来保存板块的数据结构，第三个参数是对应城市用来保存每个板块的均价的数据结构

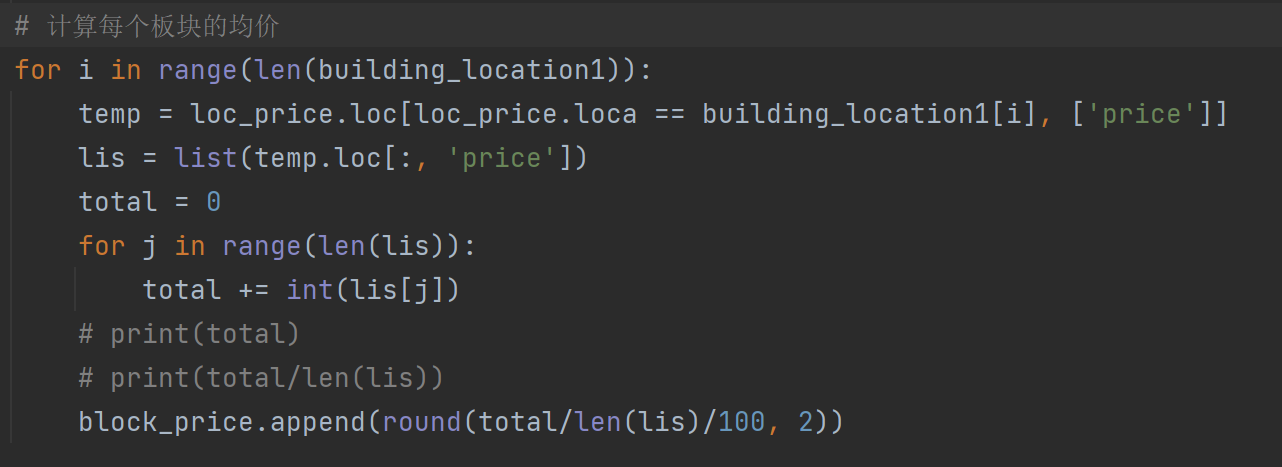
进入函数，首先就是去重，然后提取出'building\_location1', 'price'两列的信息，其中building\_location1保存的就是板块名称：



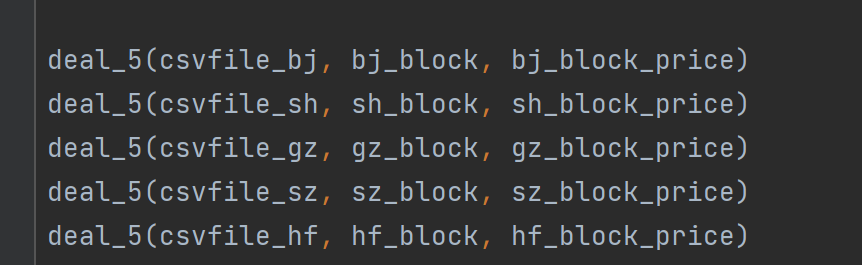
然后是以下操作（注释写了）：



然后计算每个板块的均价，这里由于数据不好展示，就把价格除以了100，所以单位是百元每月



函数到此定义完成，之后就是传参调用：



最后画图，代码重复，只展示一个：



这里没有直接在运行中展示出图片，而是以特定分辨率保存到了本地：

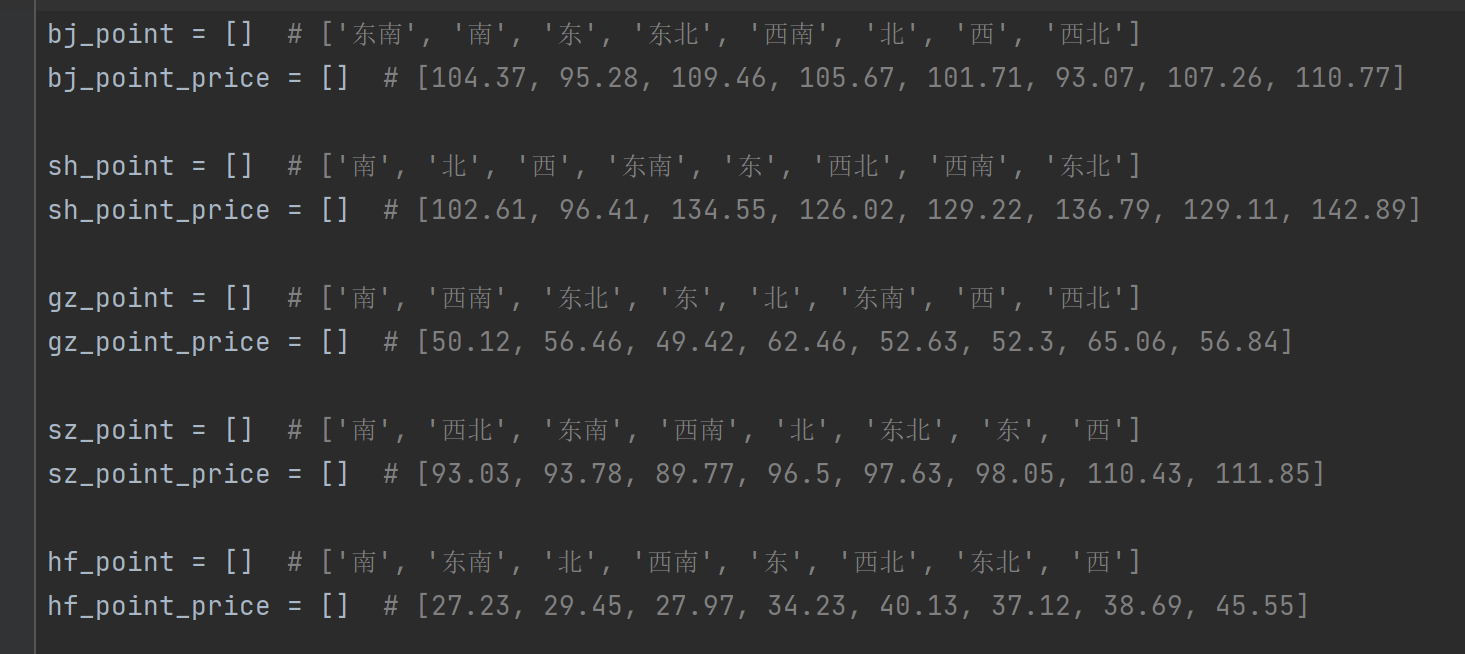
由于图片太长，就不展示了，可以去对文件里查看。

### 下面是第六问的过程：

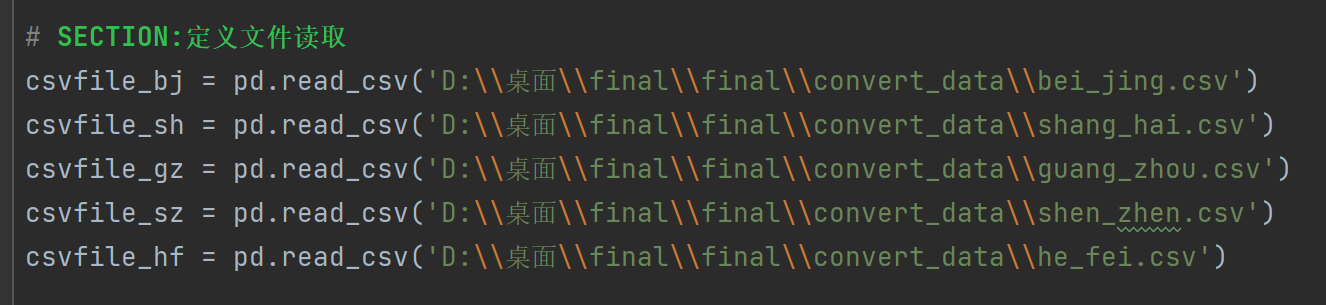
第六问要比较各个城市不同朝向的单位面积租金分布情况，采用合适的图或表形式进行展示。

所以就要获取每个城市每个房子的所有朝向，价格和面积。

为此，定义以下数据结构分别保存每个城市的数据，后面注释给的是计算好的数据，可以直接使用，节省运行时间：



依旧要定义文件读取：



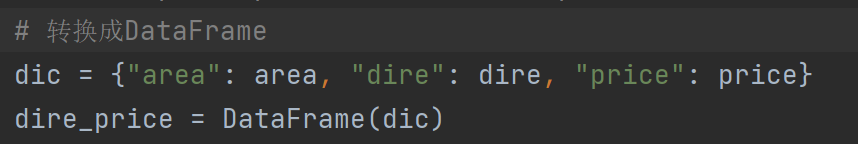
还是定义了一个函数来处理五个城市：



参数传递类似第五问

由于类似第五问，接下来用注释说明代码：



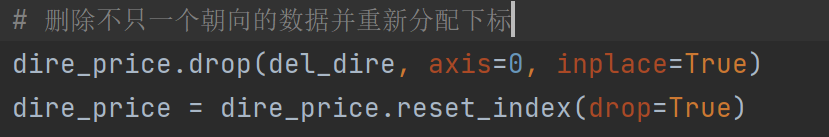


得到了需要的三列数据，但观察数据会发现，一个房子朝向并不只一个，这里就对每个房子的朝向进行判断，由于方向只有八个，最多为两个字符，所以超过两个字符的就需要特殊处理，这里是把这些特殊数据进行spilt按空格切割，并保存到列表里，之后，还要保存这些不止一个朝向的房子的下标，为后续删除这些数据做准备

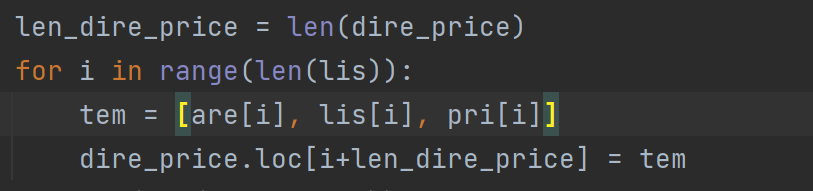
最后，为这些切割出来的朝向保存对应的面积和price，注意循环的长度为切割出来的朝向的数量



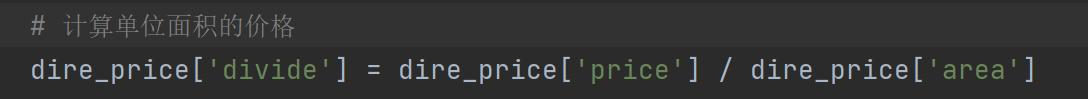
然后删除不只一个朝向的数据并重新分配下标



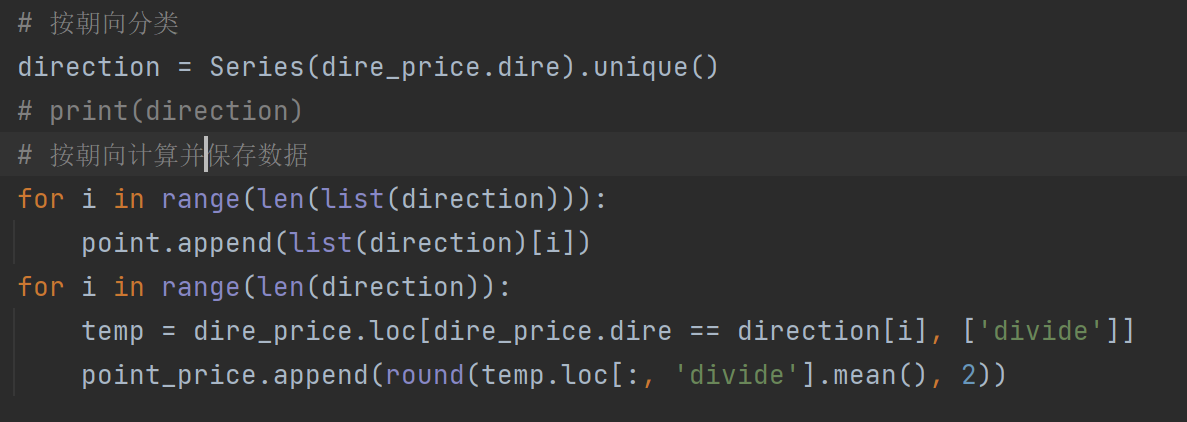
将切割出来的数据加上对应的面积和价格，在存到dataframe中



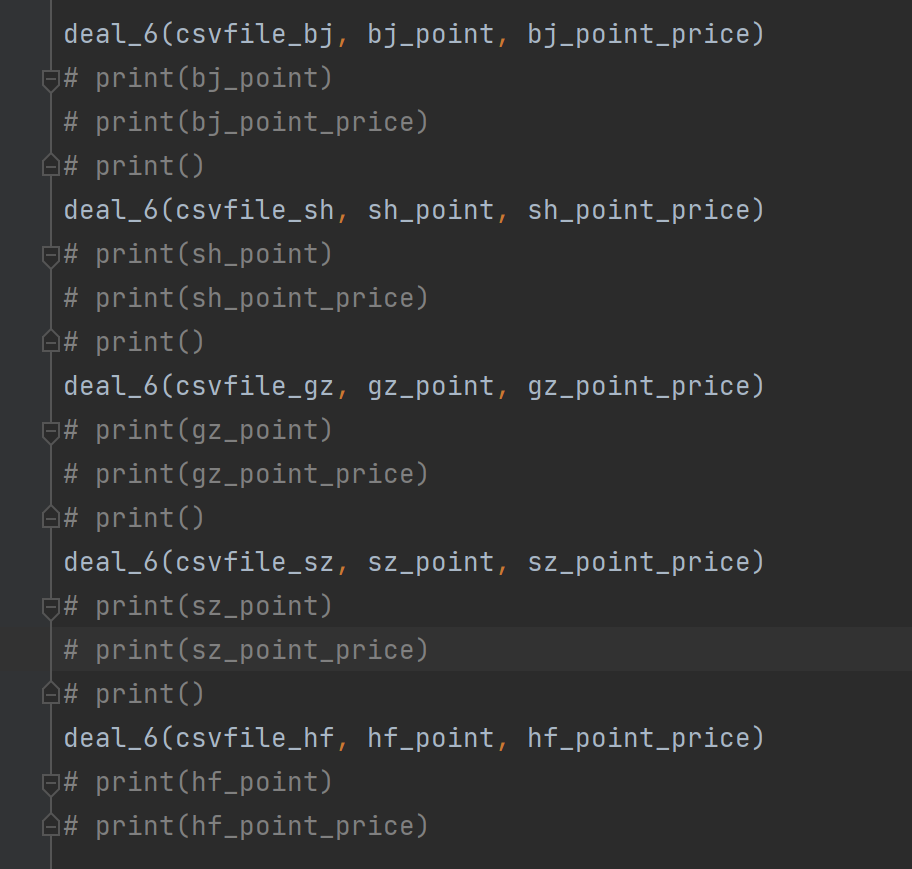
然后计算单位面积的价格



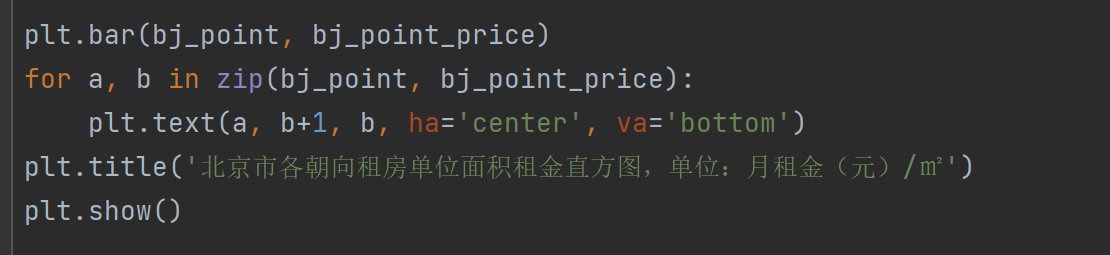
最后，按朝向分类，然后按朝向计算并保存数据



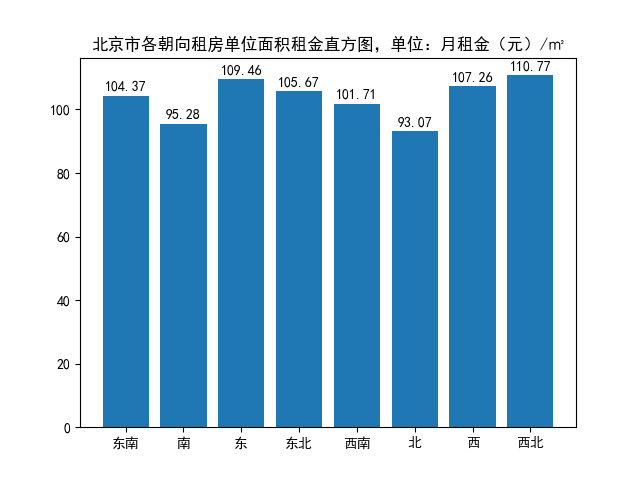
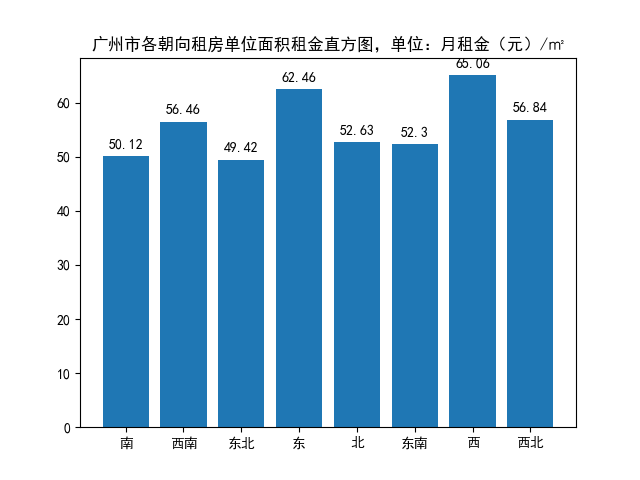
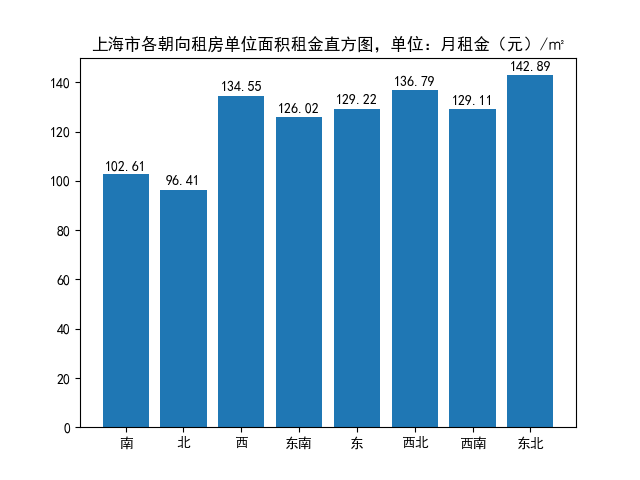
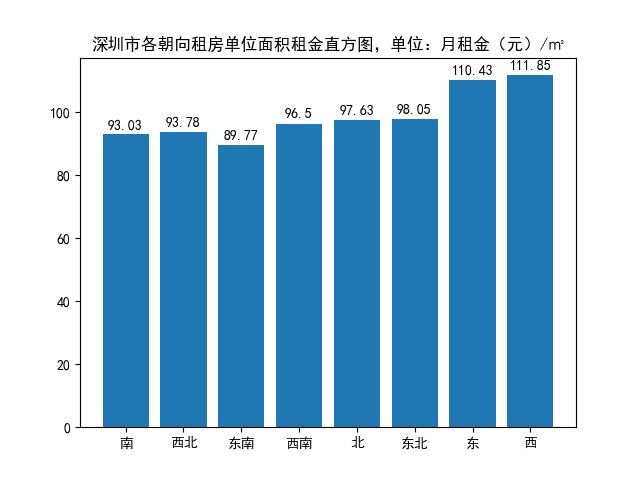
到此，函数完毕，之后就是调用



然后作图（代码重复，只展示一个）：



最后，得到五张图，可以看到，程序结果正确，每个城市都是八个方向：



**哪个方向最高，哪个方向最低？**

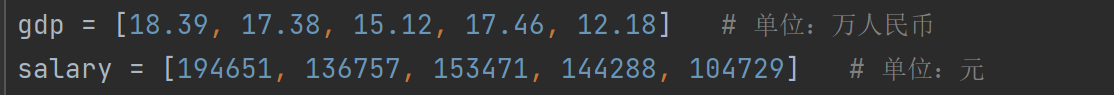
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 北京 | 上海 | 广州 | 深圳 | 合肥 |
| 最高 | 西北 | 东北 | 西 | 西 | 西 |
| 最低 | 北 | 北 | 东北 | 东南 | 南 |

**各个城市是否一致？如果不一致， 你认为原因是什么？**

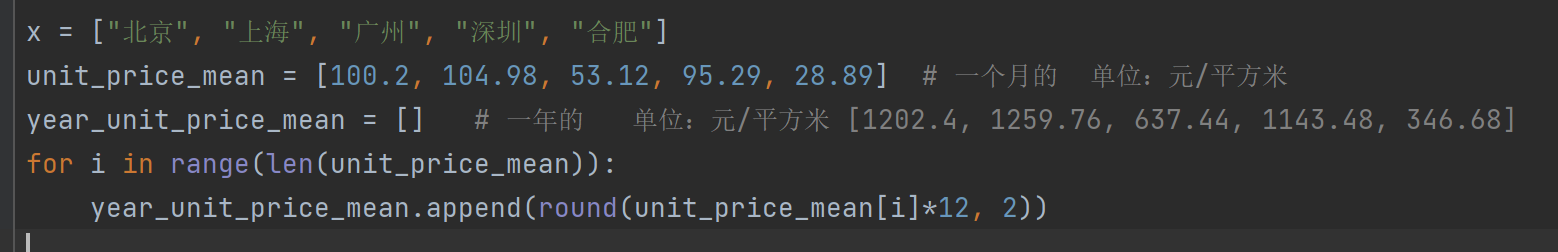
不一致。我认为原因主要与地理位置和当地习惯有关。

### 最后是第七题和第八题

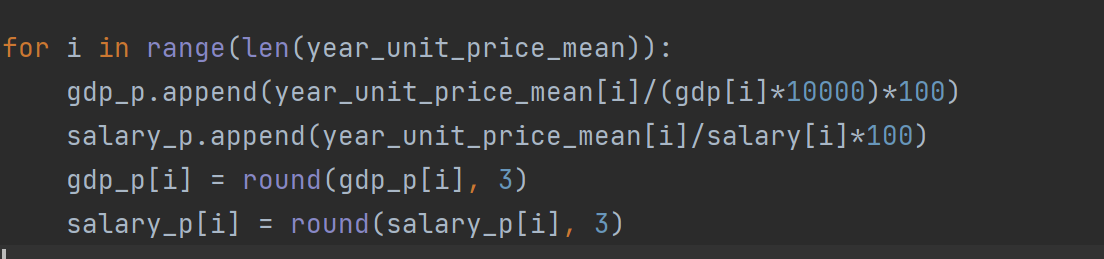
这两题是通过查数据然后绘表，查到的数据如下：



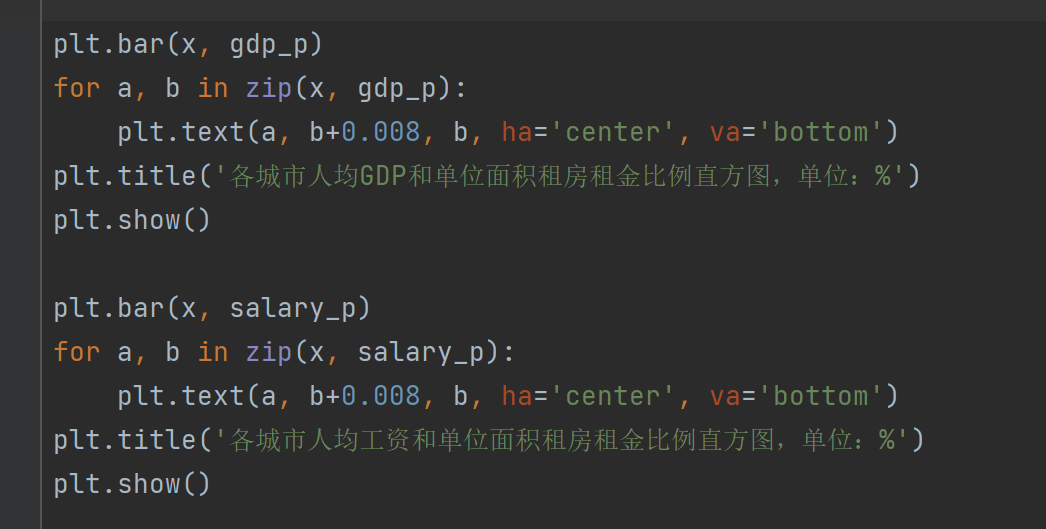
而单位面积的价格之前已经算出来了，这里还计算出了一年的单位面积价格，为了后续画图做准备：



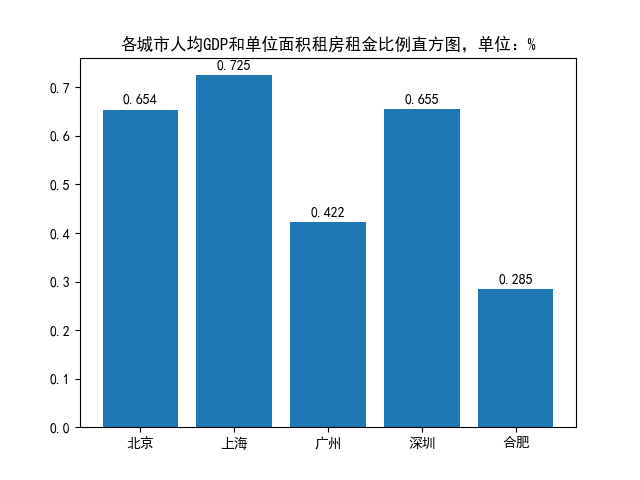
得到了数据，然后就是数据处理，将对应两者相除在乘以100算出百分比：



然后画图：



得到如下两图：



**7.相对而言，在哪个城市租房的性价比最高？**

由图可知，在合肥性价比最高。

**8.相对而言，在哪个城市租房的负担最重？**

在上海负担最重

到此，问题全部解决。