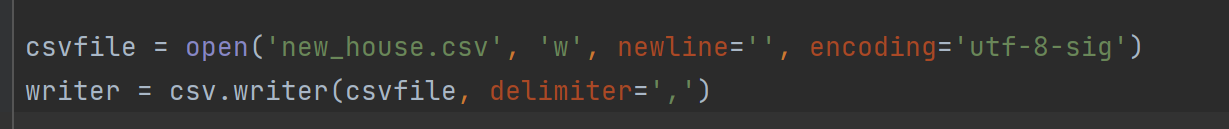
# **一、对新房数据进行预处理**

实现代码文件：convert.py

程序思路：

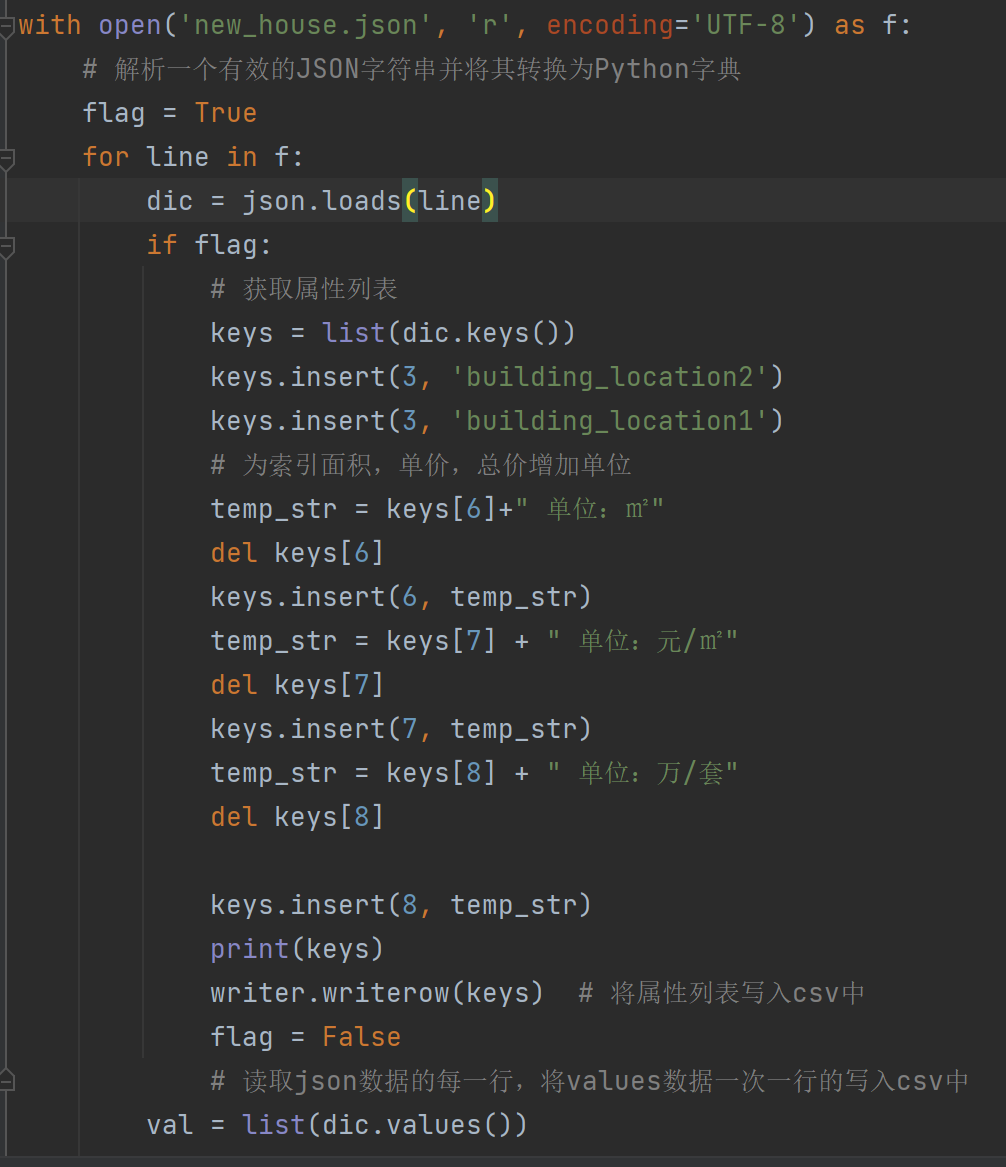
首先以‘w’方式打开（若不存在则新建）new\_house.csv文件，并做好写入数据的准备：



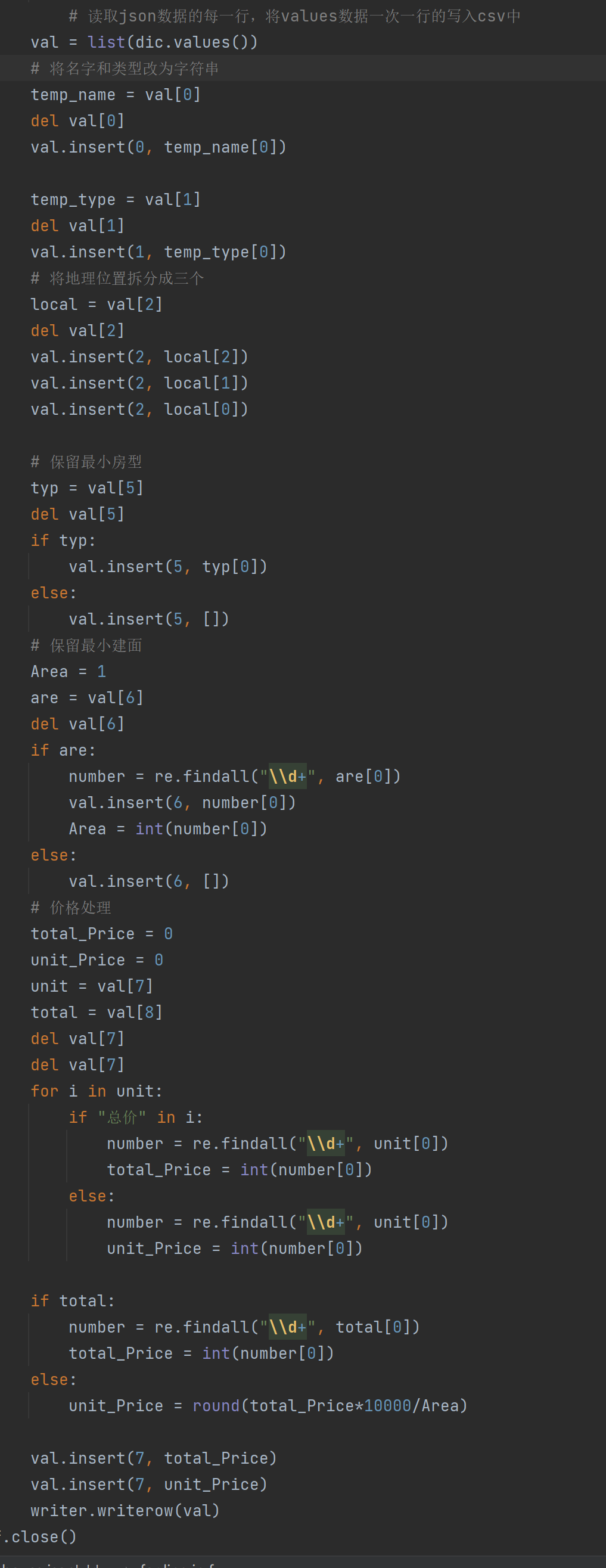
然后以‘r’方式打开new\_house.json文件：



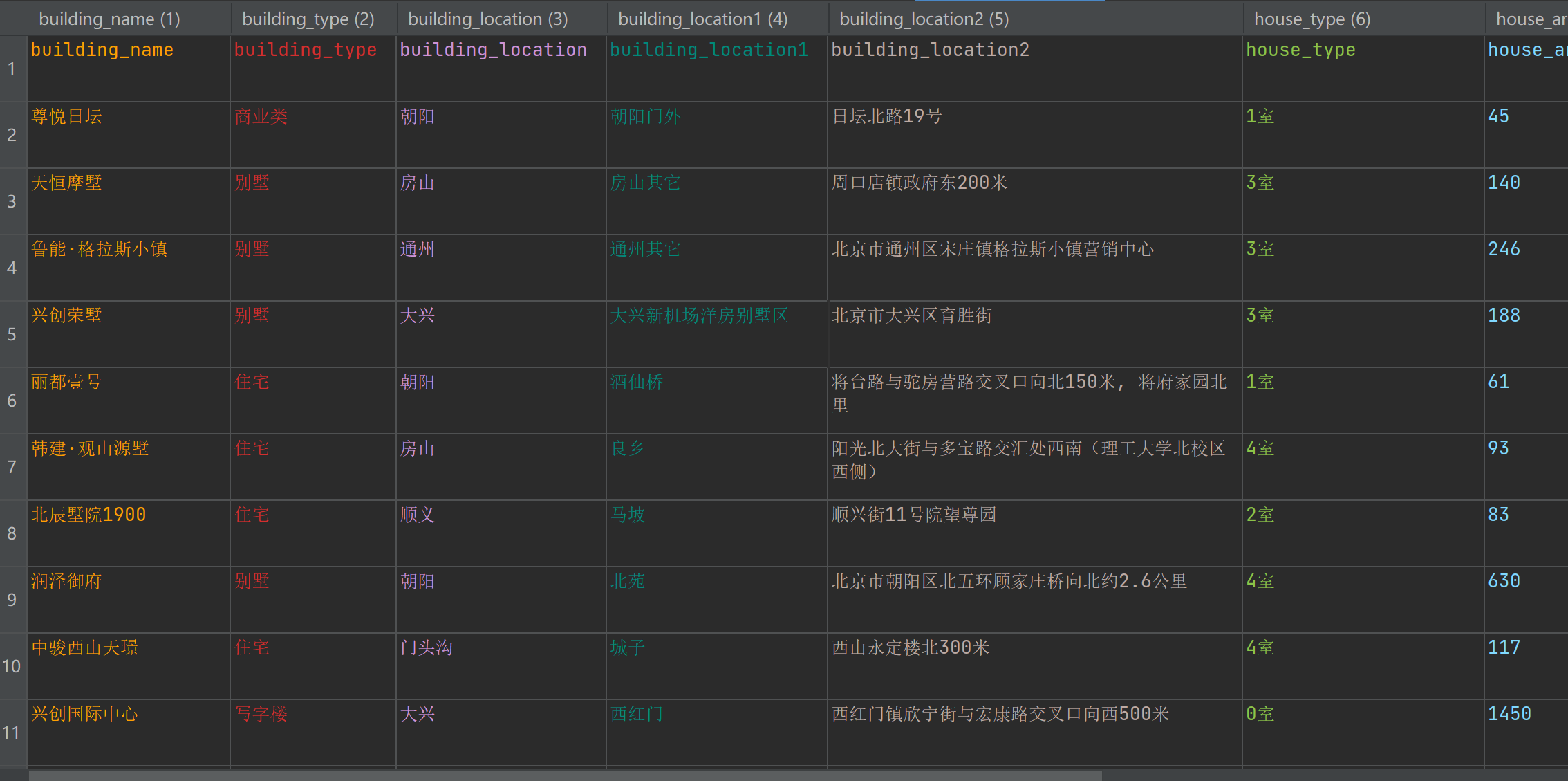
之后先向new\_house.csv文件写入属性列表，但要对其进行一定的操作以满足要求，即通过对列表进行删改来达到要求，flag变量用来确保只写入一次属性列表：



后面就开始每次获取一条数据的所有键值，并对其按要求进行修改，通过转换成列表进行修改在写入完成：



到此，就完成了预处理工作，可以得到如下的new\_house.csv文件：



# **二、将处理后的数据进行可视化展示**

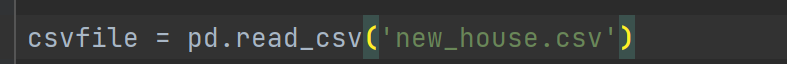
**1. 绘制楼盘价格分布的散点图，纵横轴分别代表单价和总价，楼盘**

**类型通过散点的颜色区分 ：**

实现代码文件：new\_house\_Scatter.py

程序思路：

首先将csv数据读入进来：



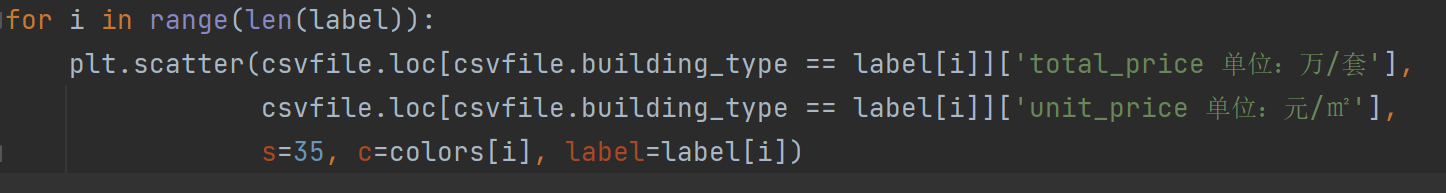
定义五种颜色的列表，用来给点上不同的颜色：



然后获取需要区分的类型，即楼盘类型，这里通过unique函数去重：



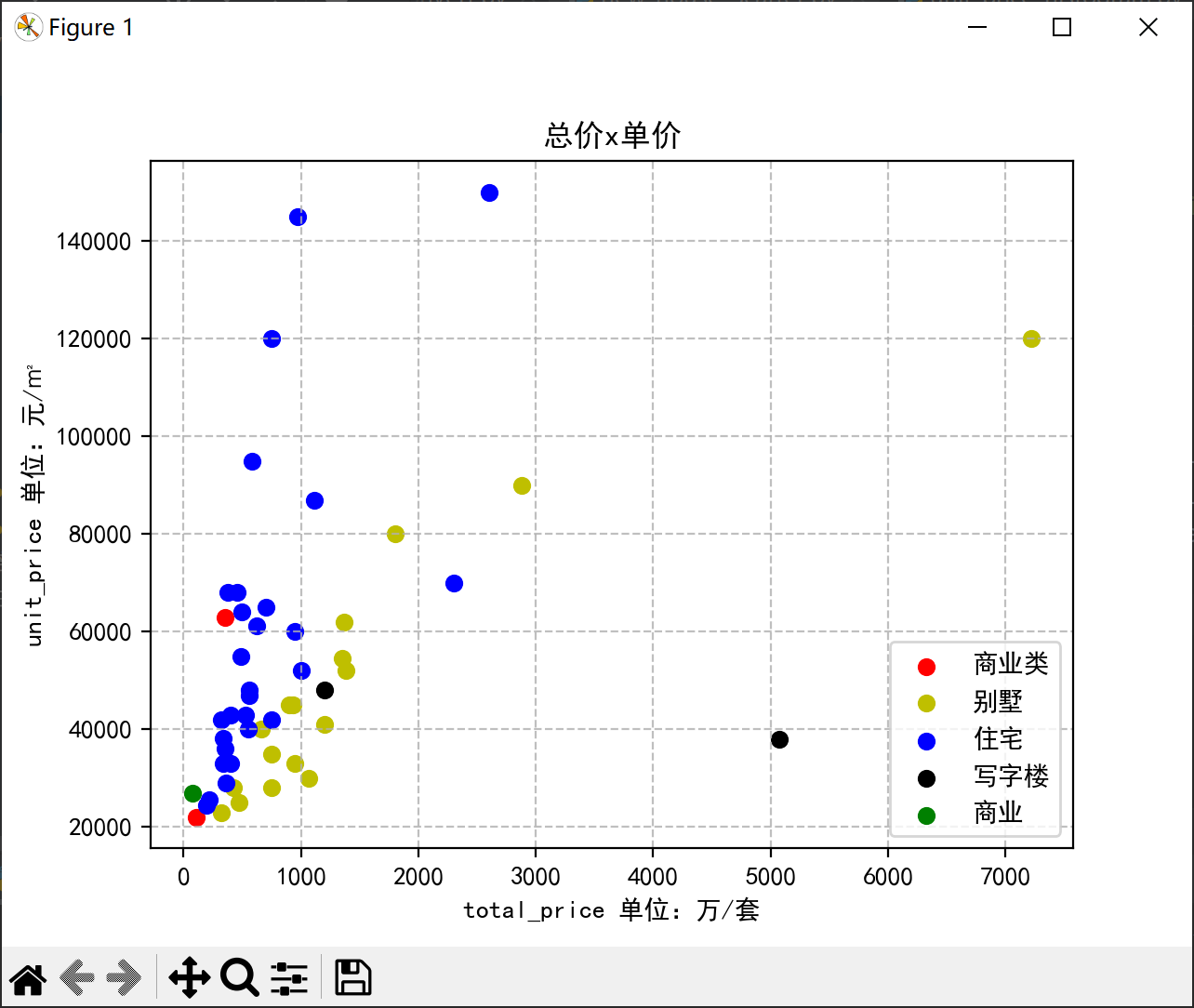
然后开始跑循环制图，通过loc获取到每种类型的每条数据的均价和总价，并设置图例



最后添加一些标题之类的，然后展示：



可以得到下图：



**2. 绘制各行政区楼盘的直方图，每个行政区一个柱子，高度代表平**

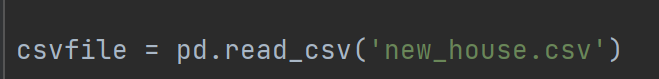
**均单价，宽度代表楼盘数量；将单价换为总价再绘制一张直方图**

实现代码文件：unit\_price\_histogram.py（高度为单价）

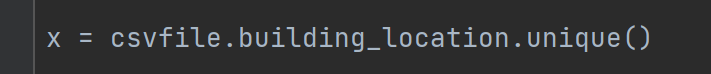
total\_price\_histogram.py（高度为总价）

下面讲一下unit\_price\_histogram.py的思路：

首先读入csv数据：



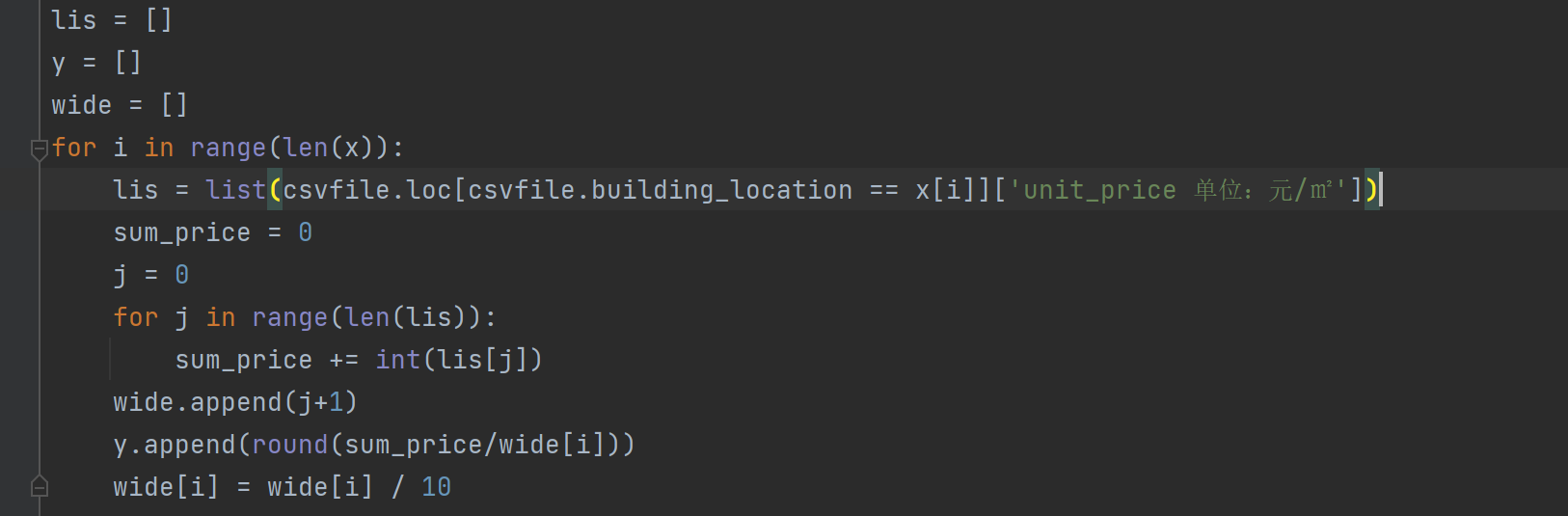
然后把行政区去重存起来，作为x轴：



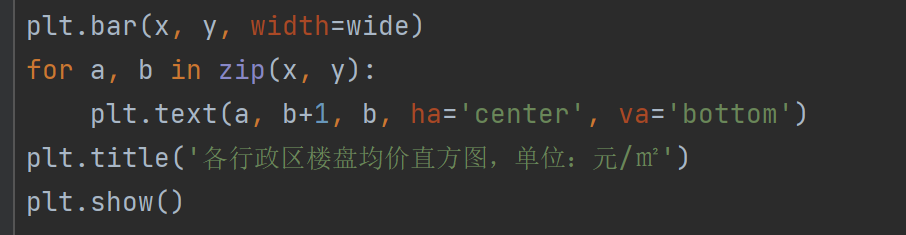
然后开始为y轴和带宽处理数据，先读入单价的数据，存到列表，

然后循环相加得到总的单价，之后就可以计算均价，并写入y轴和带宽的列表。

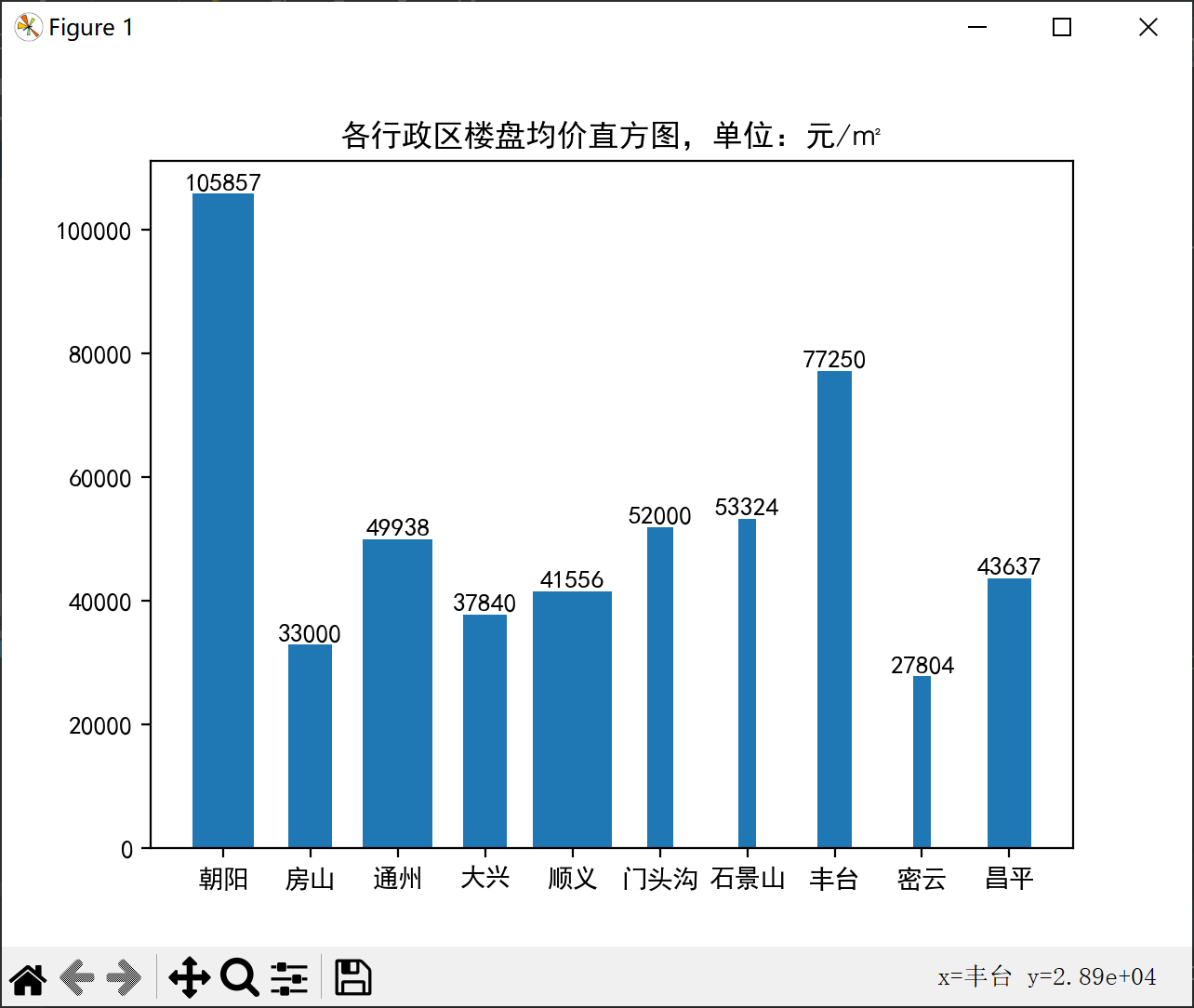
带宽的处理：这里发现各行政区的楼盘数量均没有超过10，于是就都除以10，求得一个小于1的比例，直接写入就行



然后绘图，并为每个柱子头添加一个数据，表示该柱的大小，最后展示：



可以得到下图：



total\_price\_histogram.py的思路和上面一样，得到下图：

