一：二手房源数据分析

1、房源数据介绍

1、house.csv文件

	index	title	community	years	housetype	square	floor	taxtype	totalPrice	unitPrice	followInfo
0	0	宝星华庭一层带花园，客厅挑高，通透四居室。房主自荐	宝星国际三期	底层(共22层)2010年建板塔结合	4室1厅	298.79平米	底层(共22层)2010年建板塔结合	距离15号线望京东站680米房本满五年	2598	86951	53人关注 / 共44次带看 / 一年前发布
1	1	三面采光全明南北朝向 正对小区绿地花园	顶秀青溪	中楼层(共11层)2008年建板塔结合	3室2厅	154.62平米	中楼层(共11层)2008年建板塔结合	距离5号线立水桥站1170米房本满两年随时看房	1000	64675	323人关注 / 共579次带看 / 一年前发布
2	2	沁园公寓 三居室 距离苏州街地铁站383米	沁园公寓	低楼层(共24层)1999年建塔楼	3室2厅	177.36平米	低楼层(共24层)1999年建塔楼	距离10号线苏州街站383米房本满五年	1200	67659	185人关注 / 共108次带看 / 一年前发布
3	3	金星园东南向户型，四居室设计，中间楼层	金星园	中楼层(共28层)2007年建塔楼	4室2厅	245.52平米	中楼层(共28层)2007年建塔楼	距离机场线三元桥站1153米房本满五年	1650	67205	157人关注 / 共35次带看 / 一年前发布

2、community_describe.csv文件

	index	id	community	district	bizcircle	tagList	onsale
0	0	1111000004310	什坊院甲3号院	海淀	田村	NaN	0
1	1	1111027373682	大慧寺6号院	海淀	白石桥	NaN	2
2	2	1111027373683	东花市北里东区	东城	东花市	近地铁1号线王府井站	0
3	3	1111027373684	东花市北里西区	东城	东花市	近地铁7号线广渠门内站	7
4	4	1111027373685	东花市北里中区	东城	东花市	近地铁2号线朝阳门站	9

2、分析业务问题

1. 数据读取及描述性分析，得到房价及平米的数值型描述
2. 删除车位信息
3. 数据分析1：价格最高的5个别墅，删除别墅信息
4. 数据分析2：找出数据中的住房户型分布
5. 数据分析3：找出关注人数最多的五套房子
6. 数据分析4：户型和关注人数分布
7. 数据分析5：面积分布
8. 数据分析6：各个行政区房源单价均价
9. 数据分析7：各个行政区的房源总价对比
10. 数据分析8：按照地铁信息对各个区域每平米均价排序，柱形图绘制
11. 数据分析9：按小区均价排序
12. 综合：紧邻望京地铁站,三室一厅，400万-500万，大于80平米的房子¶

二：知识点补充

1、使用apply方法聚合数据

使用方法对对象进行聚合操作其方法和方法也相同，不同之处在于方法相比方法传入的函数只能够作用于整个或者，而无法像一样能够对不同字段应用不同函数获取不同结果

DataFrame.apply(func, axis=0)

```
import numpy as np
import pandas as pd
np.random.seed(123)
df=pd.DataFrame(np.random.randn(4,5),columns=list('abcde'))
df
# 求每列的最大值与最小值的差
a = df.apply(lambda x:x.max()-x.min(),axis=0)
# 求每行的最大值与最小值的差
b = df.apply(lambda x:x.max()-x.min(),axis=1)
a
b
```

2、主键合并数据

和数据库的join一样，merge函数也有左连接（left）、右连接（right）、内连接（inner）和外连接（outer）

pandas.merge(left, right, how='inner', on=None, left_on=None, right_on=None, left_index=False, right_index=False, sort=False, suffixes=('*x*', 'y'), copy=True, indicator=False)

参数名称	说明
left	接收DataFrame或Series。表示要添加的新数据。无默认。
right	接收DataFrame或Series。表示要添加的新数据。无默认。。
how	接收inner，outer，left，right。表示数据的连接方式。默认为inner。
on	接收string或sequence。表示两个数据合并的主键（必须一致）。默认为None。
left_on	接收string或sequence。表示left参数接收数据用于合并的主键。默认为None。
right_on	接收string或sequence。表示right参数接收数据用于合并的主键。默认为None。
left_index	接收boolean。表示是否将left参数接收数据的index作为连接主键。默认为False。
right_index	接收boolean。表示是否将right参数接收数据的index作为连接主键。默认为False。
sort	接收boolean。表示是否根据连接键对合并后的数据进行排序。默认为False。

```
import pandas as pd
import numpy as np
# 依据一组key合并
#定义资料集并打印出
left = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K3'],
                     'A': ['A0', 'A1', 'A2', 'A3'],
                     'B': ['B0', 'B1', 'B2', 'B3']})
right = pd.DataFrame({'key': ['K0', 'K1', 'K2', 'K4'],
                      'C': ['C0', 'C1', 'C2', 'C3'],
                      'D': ['D0', 'D1', 'D2', 'D3']})
left
right
#练习on参数，how参数
```

```
#依据key column合并，并打印出结果 res
# res=pd.merge(left,right)
res=pd.merge(left,right,on='key',how='left')
res


## 练习参数left_on，right_index
left = pd.DataFrame({'A': ['A0', 'A1', 'A2', 'A3'],
                     'B': ['B0', 'B1', 'B2', 'B3'],
                   'key': ['K0', 'K1', 'K0', 'K1']})

right = pd.DataFrame({'C': ['C0', 'C1'],
                      'D': ['D0', 'D1']},
                     index=['K0', 'K1'])
result = pd.merge(left, right, left_on='key', right_index=True, how='left',
sort=True)
result


#练习参数left_index，right_index
left = pd.DataFrame({'A': ['A0', 'A1', 'A2'],
                     'B': ['B0', 'B1', 'B2']},
                      index=['K0', 'K1', 'K2'])

right = pd.DataFrame({'C': ['C0', 'C2', 'C3'],
                      'D': ['D0', 'D2', 'D3']},
                        index=['K0', 'K2', 'K3'])
pd.merge(left,right,how= 'left',left_index=True,right_index=True)
```