Python 数据分析

班级:2017211314　　学号:2017213508　　学生:蒋雪枫

一. 任务

数据处理是学习 Python 必须掌握的一门技能,本次课堂中杨老师给我们简单介绍了以下有关的知识与技术,并布置了如下任务:

作业1:

1.在链家官网上，找一个你感兴趣的城市，将其二手房数据爬取下来（取前100页3000个数据，如果不够3000个则取全部），并进行清洗整理（去掉不必要的空格，算出单价，房屋建成年份用整数表示等）；

2.查看总价和均价的统计量分析;

3.完成总价和单价的分布分析：要求画出直方图;

4.进行帕累托分析：看看是否符合二八定律。

二. 网站爬虫(深圳市)

Item 定义:

```
8   import scrapy
9
10  class XinfangItem(scrapy.Item):
11      name = scrapy.Field()
12      information = scrapy.Field()
13      totalprice = scrapy.Field()
14      unitprice = scrapy.Field()
```

网页数据获取:

```
class xinfangSpider(scrapy.spiders.Spider):
    name = "fuck"
    allowed_domains = ["sz.lianjia.com/"]

    start_urls = []
    for page in range(1,100):
        url = "https://sz.lianjia.com/ershoufang/pg{}/".format(page)

        start_urls.append(url)

    def parse(self,response):
        item = XinfangItem()
```

```
            for i in range(1,30):
                item['name'] =
response.xpath('//*[@id="content"]/div[1]/ul/li[{}]/div[1]/div[1]/a/text()'.format(i)).extract()
                item['information'] =
response.xpath('//*[@id="content"]/div[1]/ul/li[{}]/div[1]/div[3]/div/text()'.format(i)).extract()
                item['totalprice'] =
response.xpath('//*[@id="content"]/div[1]/ul/li[{}]/div[1]/div[6]/div[1]/span/text()'.format(i)).e
xtract()
                item['unitprice'] =
response.xpath('//*[@id="content"]/div[1]/ul/li[{}]/div[1]/div[6]/div[2]/span/text()'.format(i)).e
xtract()

                if(item['name'] and item['information'] and item['totalprice'] and
item['unitprice']):
                    yield (item)
```

获取了数据之后再统一处理:

```
#打开 csv 文件
fileNameStr = 'dealsecondhand.csv'
orig_df = pd.read_csv(fileNameStr,encoding='utf-8',dtype=str)

#1.将 name,information,totalprice,unitprice 列去掉空格
orig_df['name'] = orig_df['name'].str.strip()
orig_df['information'] = orig_df['information'].str.strip()
orig_df['totalprice'] = orig_df['totalprice'].str.strip()
orig_df['unitprice'] = orig_df['unitprice'].str.strip()

#2.处理 unitprice 中的字
orig_df['unitprice'] = orig_df['unitprice'].str.replace("单价","")
orig_df['unitprice'] = orig_df['unitprice'].str.replace("元/平米","")

#3.将后两列转为整数，总价乘 10000
orig_df['unitprice'] = orig_df['unitprice'].astype(np.int)
orig_df['totalprice'] = orig_df['totalprice'].astype(np.float)
orig_df['totalprice'] = orig_df['totalprice'].astype(np.int)
orig_df['totalprice'] = orig_df['totalprice'] * 10000

#4.输出到文件
orig_df.to_csv("dealsecondhand.csv",encoding = 'gbk')
```

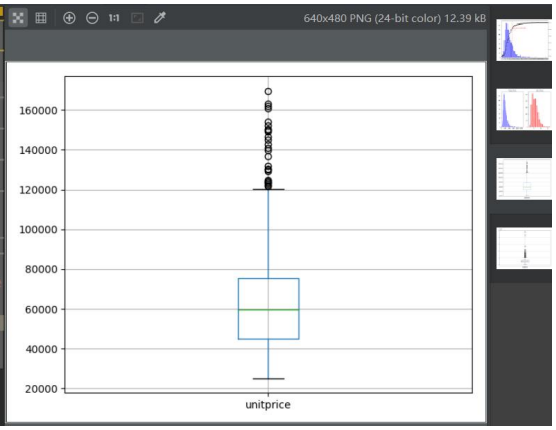
部分数据展示:

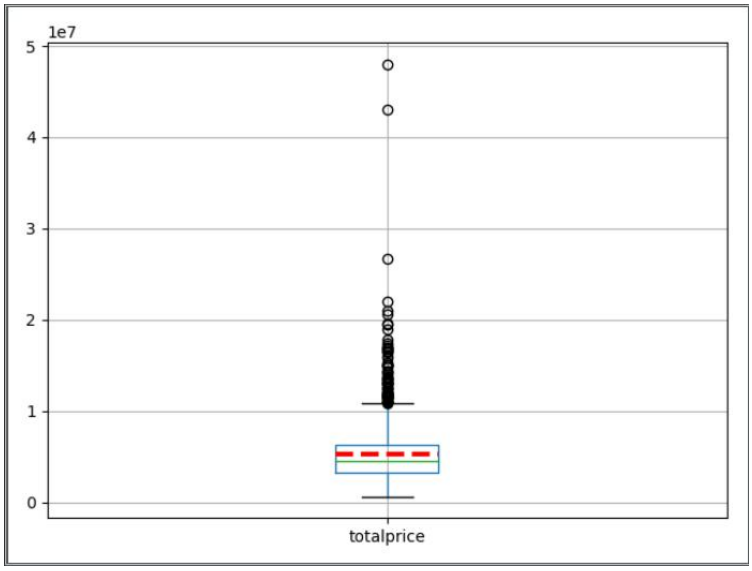
	A	B	C	D	E
13	11	交通方便，实用三房，满五唯一税少	3室2厅 | 79.83平米 | 西南 | 简装 | 中楼层(共12层) | 2004年建 | 塔楼	5950000	74534
14	12	信和自由广场 经典小两房 满五年唯一	2室1厅 | 50.27平米 | 东北 | 精装 | 中楼层(共26层) | 2006年建 | 塔楼	4380000	87130
15	13	深业东城国际 3室1厅 280万	3室1厅 | 80.96平米 | 南 | 精装 | 低楼层(共33层) | 2013年建 | 塔楼	2800000	34585
16	14	海富花园 精装两房 高楼层 居家舒适	2室1厅 | 72.54平米 | 东北 | 精装 | 高楼层(共32层) | 1991年建 | 板塔结合	3180000	43838
17	15	此房是满二，正规一房一厅，翠竹地铁,	1室1厅 | 41.9平米 | 南 | 简装 | 高楼层(共32层) | 2005年建 | 板塔结合	2950000	70406
18	16	前海东岸 三室二厅一卫 精装修 全部	3室2厅 | 88.4平米 | 东南 | 精装 | 低楼层(共38层) | 2018年建 | 塔楼	8500000	96154
19	17	和兴花园 厅出阳台中装3房 户型方正	3室2厅 | 92.61平米 | 南 | 精装 | 中楼层(共7层) | 2002年建 | 板塔结合	3450000	37253
20	18	金地上塘道一期 2室1厅 410万	2室1厅 | 64.8平米 | 东南 | 精装 | 高楼层(共25层) | 2010年建 | 塔楼	4100000	63272
21	19	满五年 户型方正 朝南向 安静舒适 采	2室1厅 | 83.52平米 | 东南 | 简装 | 高楼层(共7层) | 1999年建 | 板塔结合	3800000	45499
22	20	这套房子正南客厅出阳台，双阳台视野	2室1厅 | 81.49平米 | 东南 | 毛坯 | 高楼层(共25层) | 2007年建 | 板楼	2300000	28225
23	21	楼下壹方天地，楼上你的温馨有爱之家	3室1厅 | 78.8平米 | 西南 西 南 | 简装 | 低楼层(共49层) | 2017年建 | 板塔结合	5200000	65990
24	22	本大厦少有客厅带阳台，主卧有阳台 方	3室2厅 | 95.6平米 | 西南 | 精装 | 低楼层(共24层) | 1996年建 | 塔楼	5300000	55440
25	23	近地铁物业，满五年，红本在手看房有	3室2厅 | 101.77平米 | 北 | 简装 | 中楼层(共12层) | 2002年建 | 塔楼	2750000	27022
26	24	高楼层 东南向 复式两房 不算一层面积	2室2厅 | 43.9平米 | 西北 | 简装 | 低楼层(共15层) | 2004年建 | 板塔结合	3150000	71754
27	25	星海名城一起方正大三房、安静看花园	3室2厅 | 92平米 | 东北 | 精装 | 低楼层(共9层) | 2001年建 | 板塔结合	6800000	73914
28	26	阳光两房满五唯一本在手 交通便利 生	2室1厅 | 72.44平米 | 东北 | 精装 | 中楼层(共11层) | 2005年建 | 板塔结合	2980000	41138
29	27	红本在手，业主性格好，诚意出售，拎	2室1厅 | 47.6平米 | 西南 | 简装 | 高楼层(共32层) | 2006年建 | 板塔结合	3250000	68278
30	28	桃源居十一区 3室2厅 508万	3室2厅 | 85平米 | 东南 | 精装 | 低楼层(共18层) | 2008年建 | 塔楼	5080000	59765
31	29	公务员社区，户型通透，采光好，业主	2室1厅 | 72.6平米 | 东南 | 简装 | 中楼层(共8层) | 1997年建 | 板塔结合	4200000	57852
32	30	海月社区顶楼复式，品牌精装，带私家	4室2厅 | 149.14平米 | 南 | 精装 | 高楼层(共6层) | 2002年建 | 板塔结合	16800000	112646
33	31	鸿翔花园精装两房 ，满两年，地铁9号	2室1厅 | 73.68平米 | 西南 西北 | 简装 | 中楼层(共32层) | 2006年建 | 塔楼	5400000	73290
34	32	满五年红本，装修保养好，看房方便，	1室1厅 | 45.37平米 | 西南 | 精装 | 低楼层(共31层) | 2000年建 | 板塔结合	2400000	52899
35	33	宝安碧海，高尔夫球场碧海湾公园边上	3室2厅 | 88.01平米 | 东南 | 精装 | 低楼层(共30层) | 2012年建 | 塔楼	6000000	68175
36	34	地铁口物业花园小区， 满五年红本诚心	3室1厅 | 80.02平米 | 南 | 简装 | 中楼层(共18层) | 2007年建 | 板楼	6500000	81230
37	35	精装修大两房，南北通，通风采光，好	2室1厅 | 72.79平米 | 东南 | 精装 | 中楼层(共29层) | 1999年建 | 板塔结合	2800000	38467

三. 统计量数据分析与箱线图

	Unnamed: 0	totalprice	unitprice
count	1300.000000	1.300000e+03	1300.000000
mean	649.500000	5.263023e+06	62607.080769
std	375.421985	3.333603e+06	22653.136360
min	0.000000	6.500000e+05	25000.000000
25%	324.750000	3.300000e+06	44957.000000
50%	649.500000	4.500000e+06	59579.000000
75%	974.250000	6.300000e+06	75289.500000
max	1299.000000	4.800000e+07	169645.000000

```python
import numpy as np
import pandas as pd
from pandas import Series,DataFrame
import matplotlib.pyplot as plt

#1.打开CSV文件
fileNameStr = 'dealsecondhand.csv'
df = pd.read_csv(fileNameStr,encoding='gbk')

#2.查看总价和均价的统计量分析
print(df.describe())
#生成箱型图
df.boxplot(column='unitprice')
plt.show()
f = df.boxplot(column='totalprice',meanline=True,showme
#画出均值线
for mean in f['means']:
    mean.set(color='r', linewidth=3)
plt.show()
```





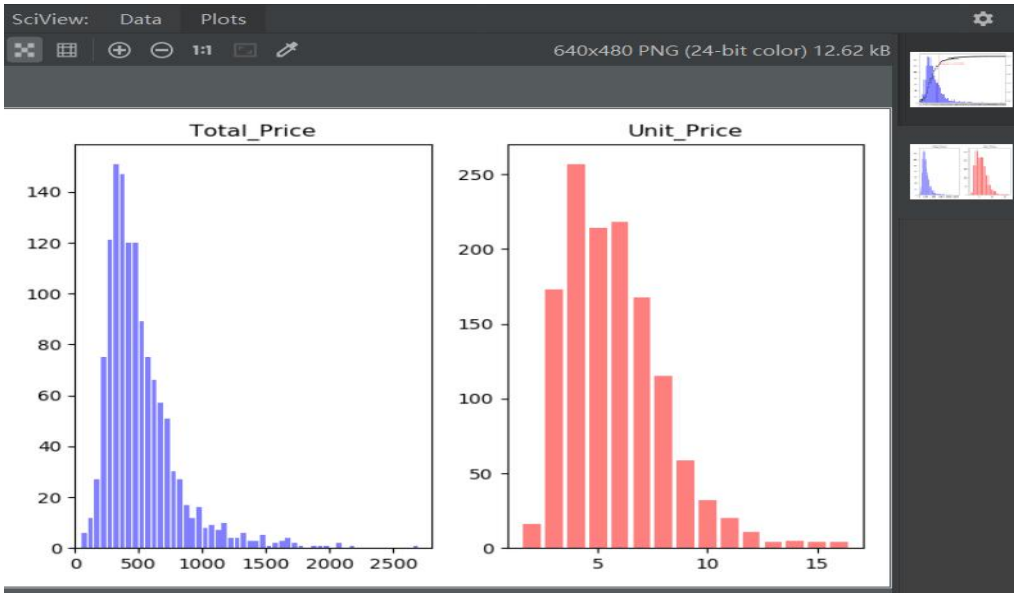
四. 总价与单价分布分析

以下两个子任务均在一份 py 代码中完成:

```
#生成 2 个子图
fig = plt.figure()
ax1 = fig.add_subplot(121)   #展示总价信息
ax2 = fig.add_subplot(122)   #展示单价信息

#打开 CSV 文件
fileNameStr = 'dealsecondhand.csv'
df = pd.read_csv(fileNameStr,encoding='gbk')

def count_elements(scores): #定义转换函数，统计每个数值对应多少个
    scorescount = {}   #定义一个字典对象
    for i in scores:
        scorescount[int(i)] = scorescount.get(int(i), 0) + 1 #累加每个整数数值的个数
    return scorescount

'''
#part1 展示总价和单价的分布直方图
df['totalprice'] = df['totalprice'] / 1000000
counted1 = count_elements(df["totalprice"])
ax1.set_title("Total_Price")
ax1.bar(counted1.keys(),counted1.values(),0.8,alpha=0.5,color='b')

df['unitprice'] = df['unitprice'] / 10000
counted2 = count_elements(df["unitprice"])
ax2.set_title("Unit_Price")
ax2.bar(counted2.keys(),counted2.values(),0.8,alpha=0.5,color='r')
```

```
plt.show()
'''
#part2，对总价信息进行区间化的处理

ax1.set_title("Total_Price")
sections = list(np.arange(0,2900,50))
print(len(sections))
mylabels = list(np.arange(25,2850,50))   #生成 x 轴上的标签
print(len(mylabels))
#result1 = pd.cut(df.totalprice,sections) #使用 cut 函数分组汇总
df['totalprice'] = df['totalprice'] / 10000
result1 = pd.cut(df.totalprice,sections,labels=mylabels) #使用 cut 函数分组汇总
print("------------result1--------------")
print(result1)
print(type(result1))

print("------------result1.value_counts--------------")
print(result1.value_counts())

result2=result1.value_counts().sort_index()   #按照索引值进行排序
print("------------result2-------------")
print(result2)

ax1.set_xlim(0,2800) #设定 x 轴的起止范围
ax1.bar(result2.index,result2.values,40,alpha=0.5,color='b')   #40 表示取值宽度 50*0.8

df['unitprice'] = df['unitprice'] / 10000
counted2 = count_elements(df["unitprice"])
ax2.set_title("Unit_Price")
ax2.bar(counted2.keys(),counted2.values(),0.8,alpha=0.5,color='r')

plt.show()

#part3: 二八定律

plt.figure()
df_counts = pd.Series(result2.values)   #频数，每个区间中房子的个数
print("----------df_counts-------------")
print(df_counts)

df_freq = df_counts/df_counts.sum()      #频率，每个区间的房子个数/总个数，即每个区间
的占比
print("----------df_freq------------")
print(df_freq)
```

```
cum_ratio = df_freq.cumsum()                #累计频率，累计百分比
print("----------df_cum_freq-------------")
print(cum_ratio)

df_counts.plot(kind = 'bar', color = 'b', alpha = 0.8, width = 0.6)   #频数的直方图
key = cum_ratio[cum_ratio>0.8].index[0]   #找到大于 80%的累计频率对应的索引号
key_num = df_counts.index.tolist().index(key)   #找到对应的索引序号
print('超过 80%累计占比的节点值： ',(key+1)*50)
print('超过 80%累计占比的节点值索引位置为： ',key_num)
print('------')
cum_ratio.plot(style = '--ko', secondary_y=True)   #累计频率的曲线图
plt.axvline(key_num,   color = 'r', linestyle = '--', alpha = 0.8) #把 80%占比的参考线画出来，直接是 key_num，因为它是 X 轴的索引值
plt.text(key_num+1,cum_ratio[key],'cumsum is: %.3f%%' % (cum_ratio[key]*100), color = 'r')   #文字提示信息

plt.show()
```



五. 帕累托分析

实现代码如上,结论一目了然.