PYTHON

数据分析

一、编译环境：

PyCharm 2019.3 (Community Edition)
Build #PC-193.5233.109, built on November 28, 2019
Runtime version: 11.0.4+10-b520.11 amd64
VM: OpenJDK 64-Bit Server VM by JetBrains s.r.o
Windows 10 10.0
GC: ParNew, ConcurrentMarkSweep
Memory: 1963M
Cores: 8
Registry:
Non-Bundled Plugins:
python版本：3.7（Anaconda3）

二、源代码：

items.py:

```
import scrapy



class Q19ScrapyItem(scrapy.Item):

    name = scrapy.Field()
    type = scrapy.Field()
    size = scrapy.Field()
    time = scrapy.Field()
    sum_price = scrapy.Field()
    avg_price = scrapy.Field()
     pass
```

spider.py:

```
import scrapy
import re
from q19_scrapy.q19_scrapy.items import Q19ScrapyItem



class mySpider(scrapy.spiders.Spider):
    name = "loufang"
    allowed_domains = ["bj.lianjia.com"]
    start_urls = []
    for page in range(1, 101):
        url = "https://bj.lianjia.com/ershoufang/pg{}/".format(page)
        start_urls.append(url)

        def parse(self, response):
            item = Q19ScrapyItem()
            for each in response.xpath("/html/body/div[4]/div[1]/ul/*"):
                item['name'] = each.xpath("div/div[1]/a/text()").extract()
                item['type'] = []
                item['size'] = []
                item['time'] = []
                detail = each.xpath("div[1]/div[3]/div/text()").extract()
                if detail:
                    split = detail[0].split('|')
                    for i in split:
                        i = i.strip()
                        if '室' in i and '厅' in i:
                            item['type'] = [i]
                        if '平米' in i:
                            item['size'] = [re.findall(r'-?\d+\.?\d*e?-?\d*?', i)[0]]
                        if '年建' in i:
                            item['time'] = [re.findall(r'\d+', i)[0]]
                item['sum_price'] = each.xpath("div[1]/div[6]/div[1]/span/text()").extract()
                item['avg_price'] = []
                if item['size'] and item['sum_price']:
                    item['avg_price'] = ['%.2f' % (float(item['sum_price'][0]) / float(item['size'][0]))]
                if item['name'] and item['sum_price'] and item['avg_price'] and \
                        item['type'] and item['size'] and item['time']:
                    yield item
```

chart.py:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator

plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

fig1 = plt.figure()

fileNameStr = 'MyData.csv'
```

```
df = pd.read_csv(fileNameStr, encoding='utf-8', dtype=str)
df['avg_price'] = df['avg_price'].astype(np.float)
df['sum_price'] = df['sum_price'].astype(np.float)
df.drop(df[df['sum_price'] > 2000].index, inplace=True)
print("------------describe---------------")
print(df.describe())   #  查看统计信息


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


ax1 = plt.subplot(2, 2, 1)
counted1 = count_elements(df['sum_price'])
plt.title("总价")
plt.bar(counted1.keys(), counted1.values(), 0.8, alpha=0.5, color='b')

ax2 = plt.subplot(2, 2, 2)
counted2 = count_elements(df['avg_price'])
plt.title("单价")
plt.bar(counted2.keys(), counted2.values(), 0.8, alpha=0.5, color='b')
ax2.xaxis.set_major_locator(MultipleLocator(1))

ax3 = plt.subplot(2, 2, 3)
plt.title("分组总价")
sections = list(np.arange(0, 2000, 50))
print(len(sections))
my_labels = list(np.arange(25, 1975, 50))
print(len(my_labels))
result1 = pd.cut(df.sum_price, sections, labels=my_labels)
print("------------result1--------------")
print(result1)
print(type(result1))

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

result2 = result1.value_counts().sort_index()
print("------------result2-------------")
print(result2)
```

```
plt.bar(result2.index, result2.values, 40, alpha=0.5, color='b')

fig2 = plt.figure()
df_counts = pd.Series(result2.values)
print("----------df_counts-------------")
print(df_counts)

df_freq = df_counts / df_counts.sum()
print("----------df_freq-------------")
print(df_freq)

cum_ratio = df_freq.cumsum()
print("----------df_cum_freq-------------")
print(cum_ratio)

df_counts.plot(kind='bar', color='b', alpha=0.8, width=0.6)
key = cum_ratio[cum_ratio > 0.8].index[0]
key_num = df_counts.index.tolist().index(key)
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)
plt.axvline(key_num, color='r', linestyle='--', alpha=0.8)
plt.text(key_num + 1, cum_ratio[key], 'cumsum is: %.3f%%' % (cum_ratio[key] * 100), color='r')
fig2.tight_layout()

plt.show()
```

三、结果及截图：

MyData.json:

{"name": ["满五唯一央产房，小区单独管理，采光无遮挡"], "type": ["3室1厅"], "size": ["57"], "time": ["1982"], "sum_price": ["810"], "avg_price": ["142105"]}
{"name": ["商品房满五年唯一 观景电梯 采光好 视野开阔 精装修"], "type": ["2室1厅"], "size": ["90.88"], "time": ["2004"], "sum_price": ["432"], "avg_price": ["47535"]}
{"name": ["西城区70年产权小户型,南向 满五唯一高楼层"], "type": ["1室0厅"], "size": ["25.3"], "time": ["2006"], "sum_price": ["350"], "avg_price": ["138339"]}
{"name": ["全南向两居 格局方正 中高楼层 采光好"], "type": ["2室1厅"], "size": ["71.5"], "time": ["2011"], "sum_price": ["316"], "avg_price": ["44195"]}
{"name": ["三环新城6号院 3室1厅 656万"], "type": ["3室1厅"], "size": ["122.88"], "time": ["2005"], "sum_price": ["656"], "avg_price": ["53385"]}
{"name": ["东方生活区 2室1厅 低楼层 南北通透 无遮挡"], "type": ["2室1厅"], "size": ["51.63"], "time": ["1984"], "sum_price": ["229"], "avg_price": ["44354"]}
{"name": ["西城广外中间楼层南北通透明厨明卫两居室。"], "type": ["2室1厅"], "size": ["47.58"], "time": ["1970"], "sum_price": ["412"], "avg_price": ["86591"]}

{"name": ["安贞西里南北通透两居室 双阳台 精装修 离公园近"], "type": ["2 室 1 厅"], "size": ["64.23"], "time": ["1985"], "sum_price": ["513"], "avg_price": ["79869"]}
{"name": ["满五唯一，东南 双阳台，中高层，视野采光无挡"], "type": ["1 室 1 厅"], "size": ["58.02"], "time": ["2004"], "sum_price": ["365"], "avg_price": ["62909"]}
{"name": ["随时可看,业主诚意出售,精装修,静待有缘人"], "type": ["2 室 1 厅"], "size": ["65.65"], "time": ["1997"], "sum_price": ["349"], "avg_price": ["53160"]}

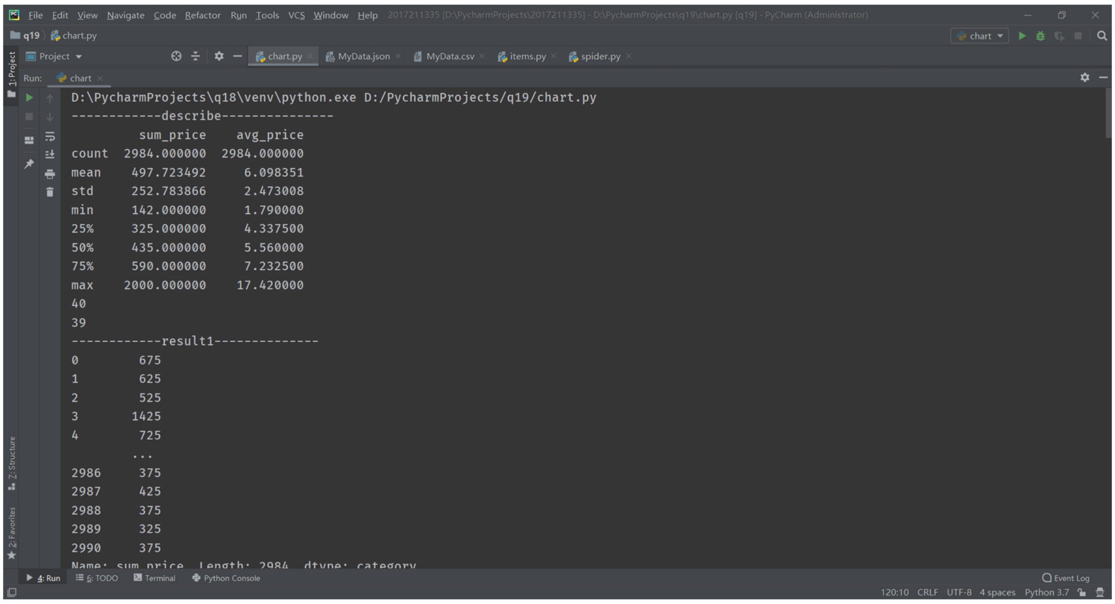
…………

MyData.csv:

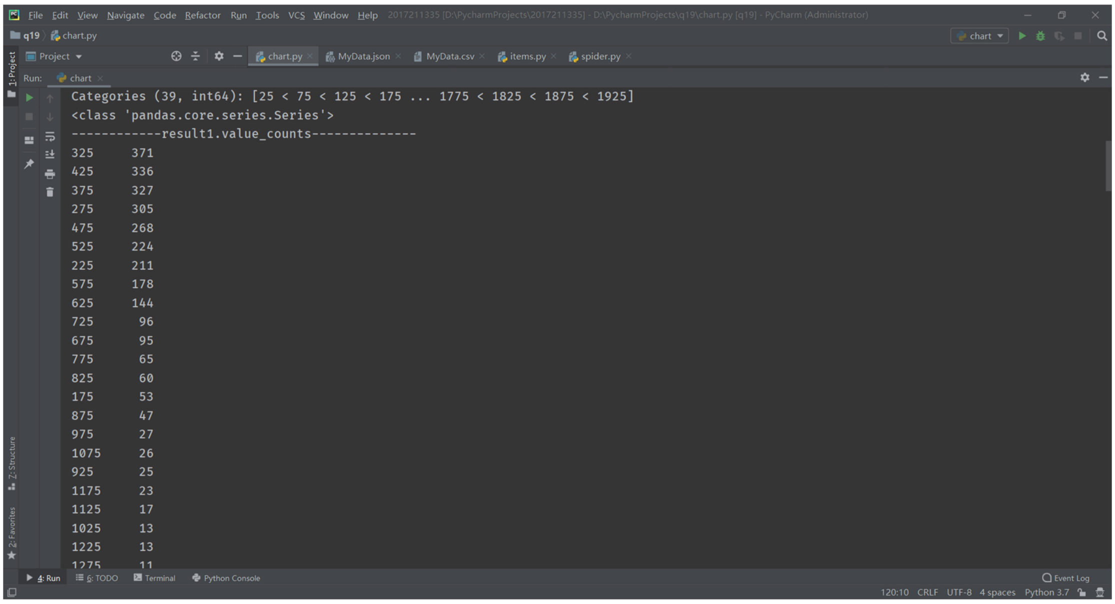
```
"name","type","size","time","sum_price","avg_price"
"永安东里低楼层三居室，满五年公房","3 室 1 厅","89.51","1998","668","7.46"
"电梯一室一厅，诚意出售，随时看房","1 室 1 厅","53.31","1998","620","11.63"
"文慧园 2 室 1 厅 545 万","2 室 1 厅","68.48","2001","545","7.96"
"此房满五唯一 南北通透，客厅朝南，落地窗，采光好","3 室 1 厅","157.76","2004","1450","9.19"
"明厨明卫 高层视野好 南北双阳台双通透","2 室 2 厅","104.64","2009","730","6.98"
"满五年，南北通透，全明格局，中间楼层，视野不错","2 室 1 厅","83.86","2000","395","4.71"
"房子是满五年唯一，两梯三户，纯板楼，朝南户型","1 室 2 厅","75.14","2009","585","7.79"
"荣丰二期正南向开间，满二唯一，无遮挡","1 室 1 厅","32.58","2004","378","11.60"
"此房是满五唯一商品房 户型方正 正规三居 交通便利","3 室 1 厅","73.8","1994","438","5.93"
"东城区商品社区 精装房 带电梯 房本满五年","1 室 1 厅","65.39","2009","523","8.00"
"芳群园一区 国家电网宿舍 正南向中间楼层两居室","2 室 1 厅","59.73","1992","399","6.68"
"大红门 角门 西马金润一区 通透两居室 明卫 电梯房","2 室 2 厅","103.73","2006","580","5.59"
"中直、信访办社区管理，带电梯，南北通透无遮挡","3 室 1 厅","77.5","1986","1080","13.94"
"小两居 万达商圈  万达对面  通州北苑 配套齐全","2 室 1 厅","49.81","1982","215","4.32"
"店长力荐好房，保利东南两居，户型方正，视野开阔。","2 室 1 厅","89.87","2010","418","4.65"
```

…………

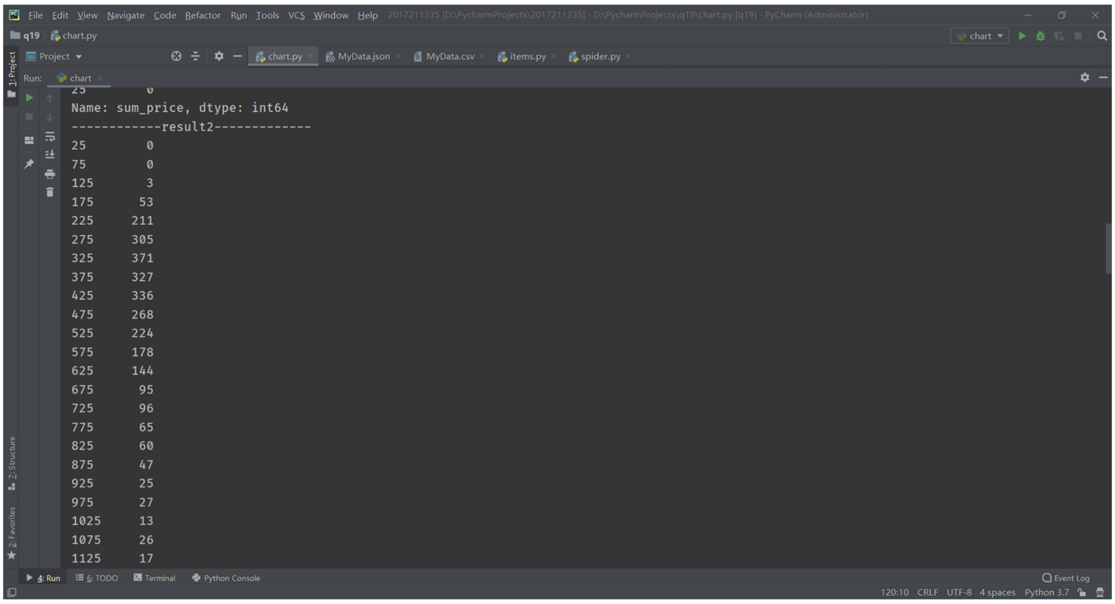
统计量分析:



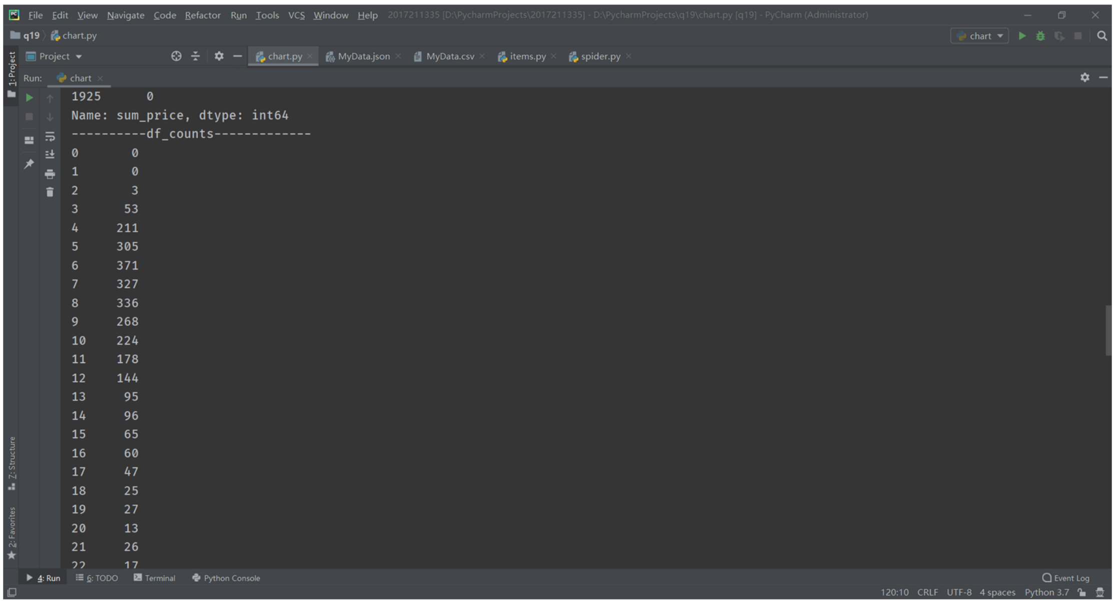
```
D:\PycharmProjects\q18\venv\python.exe D:/PycharmProjects/q19/chart.py
------------describe---------------
         sum_price    avg_price
count  2984.000000  2984.000000
mean    497.723492     6.098351
std     252.783866     2.473008
min     142.000000     1.790000
25%     325.000000     4.337500
50%     435.000000     5.560000
75%     590.000000     7.232500
max    2000.000000    17.420000
40
39
------------result1--------------
0        675
1        625
2        525
3       1425
4        725
        ...
2986     375
2987     425
2988     375
2989     325
2990     375
Name: sum_price, Length: 2984, dtype: category
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help 2017211335 [D:\PycharmProjects\2017211335] - D:\PycharmProjects\q19\chart.py [q19] - PyCharm (Administrator)
q19 chart.py
chart
Project
chart.py
MyData.json
MyData.csv
items.py
spider.py
Run: chart
1: Project
Z: Structure
2: Favorites
4: Run
6: TODO
Terminal
Python Console
Event Log
120:10 CRLF UTF-8 4 spaces Python 3.7

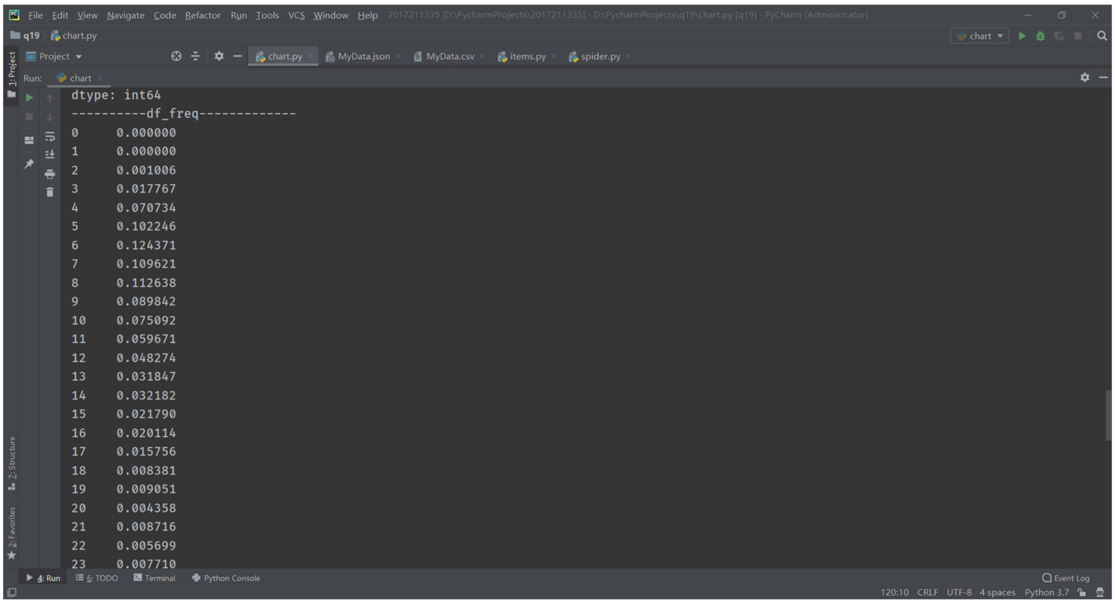
```
Categories (39, int64): [25 < 75 < 125 < 175 ... 1775 < 1825 < 1875 < 1925]
<class 'pandas.core.series.Series'>
------------result1.value_counts-------------
325     371
425     336
375     327
275     305
475     268
525     224
225     211
575     178
625     144
725      96
675      95
775      65
825      60
175      53
875      47
975      27
1075     26
925      25
1175     23
1125     17
1025     13
1225     13
1275     11
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help 2017211335 [D:\PycharmProjects\2017211335] - D:\PycharmProjects\q19\chart.py [q19] - PyCharm (Administrator)
q19 chart.py
chart
Project
chart.py
MyData.json
MyData.csv
items.py
spider.py
Run: chart
1: Project
Z: Structure
2: Favorites
4: Run
6: TODO
Terminal
Python Console
Event Log
120:10 CRLF UTF-8 4 spaces Python 3.7

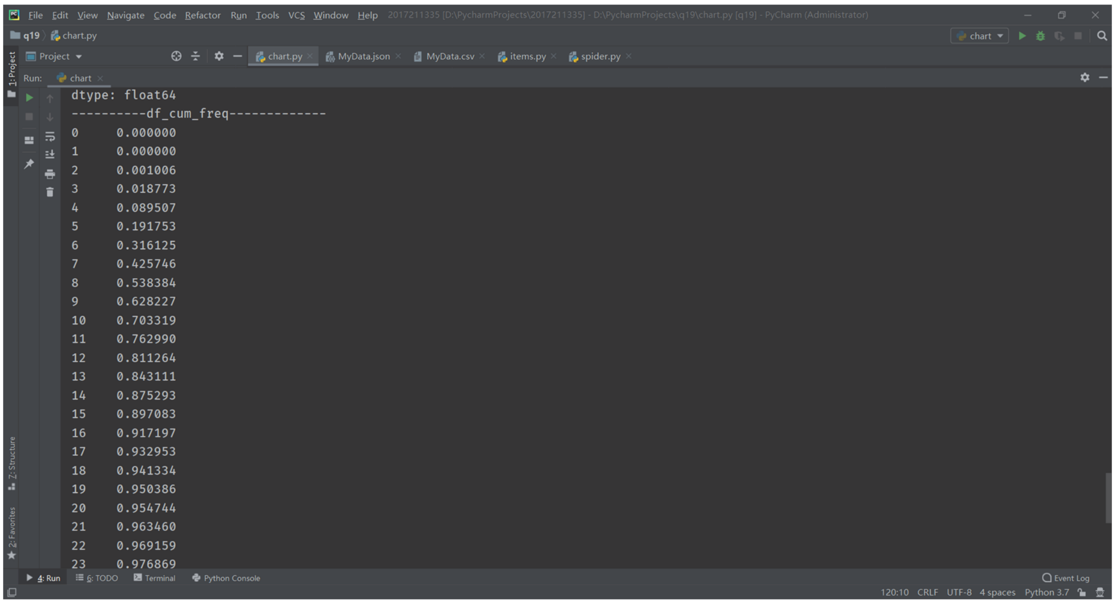
```
25      0
Name: sum_price, dtype: int64
------------result2------------
25        0
75        0
125       3
175      53
225     211
275     305
325     371
375     327
425     336
475     268
525     224
575     178
625     144
675      95
725      96
775      65
825      60
875      47
925      25
975      27
1025     13
1075     26
1125     17
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help 2017211335 [D:\PycharmProjects\2017211335] - D:\PycharmProjects\q19\chart.py [q19] - PyCharm (Administrator)
q19 chart.py
chart
Project
chart.py
MyData.json
MyData.csv
items.py
spider.py
Run: chart
1: Project
Z: Structure
2: Favorites
4: Run
6: TODO
Terminal
Python Console
Event Log
120:10 CRLF UTF-8 4 spaces Python 3.7

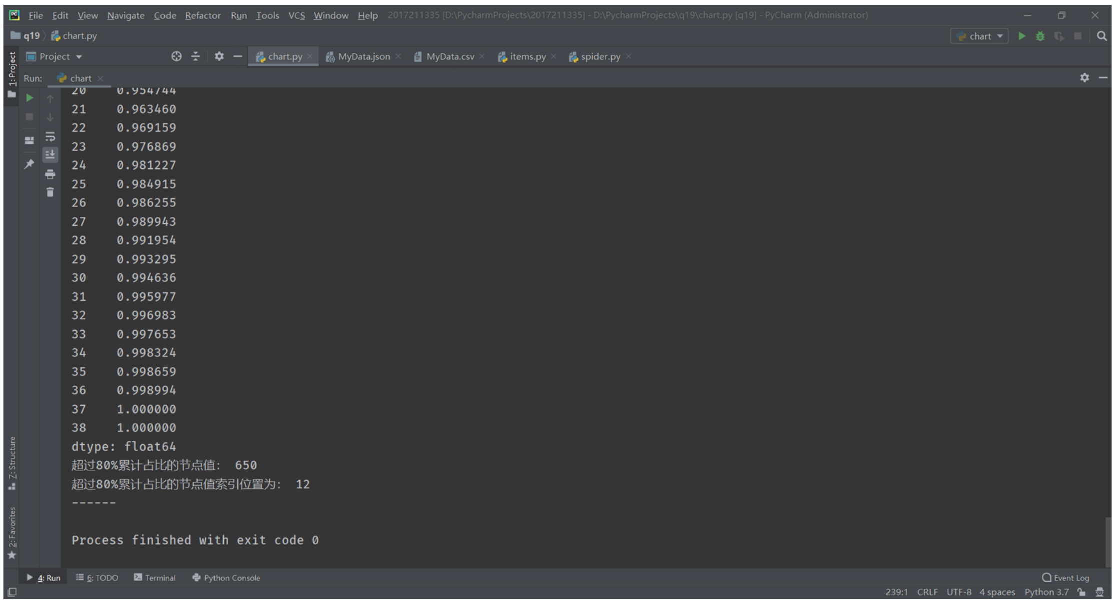
```
1925      0
Name: sum_price, dtype: int64
----------df_counts------------
0       0
1       0
2       3
3      53
4     211
5     305
6     371
7     327
8     336
9     268
10    224
11    178
12    144
13     95
14     96
15     65
16     60
17     47
18     25
19     27
20     13
21     26
22     17
```



```
dtype: int64
----------df_freq-------------
0     0.000000
1     0.000000
2     0.001006
3     0.017767
4     0.070734
5     0.102246
6     0.124371
7     0.109621
8     0.112638
9     0.089842
10    0.075092
11    0.059671
12    0.048274
13    0.031847
14    0.032182
15    0.021790
16    0.020114
17    0.015756
18    0.008381
19    0.009051
20    0.004358
21    0.008716
22    0.005699
23    0.007710
```



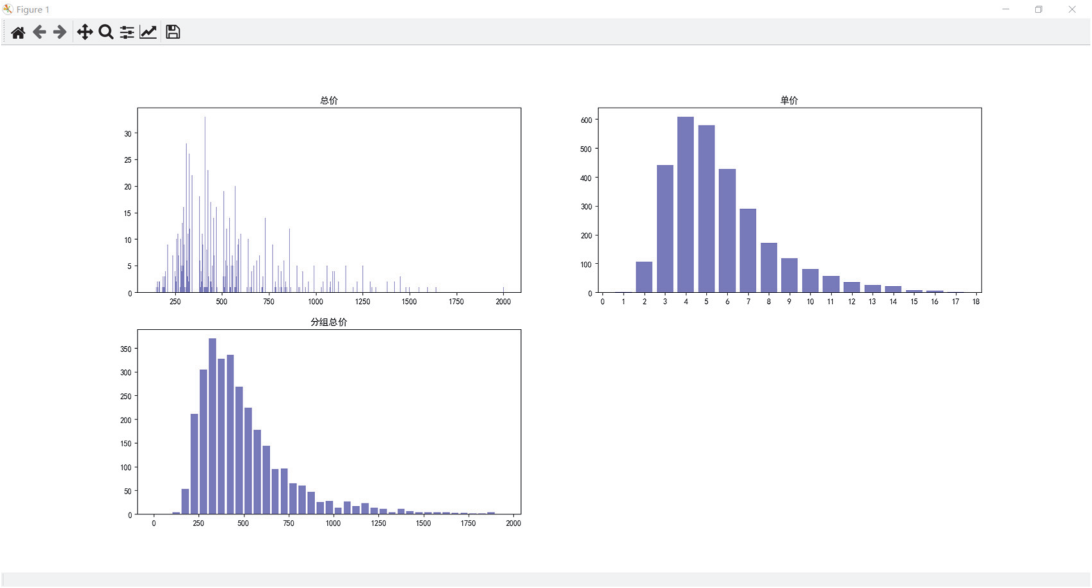
```
dtype: float64
----------df_cum_freq-------------
0     0.000000
1     0.000000
2     0.001006
3     0.018773
4     0.089507
5     0.191753
6     0.316125
7     0.425746
8     0.538384
9     0.628227
10    0.703319
11    0.762990
12    0.811264
13    0.843111
14    0.875293
15    0.897083
16    0.917197
17    0.932953
18    0.941334
19    0.950386
20    0.954744
21    0.963460
22    0.969159
23    0.976869
```

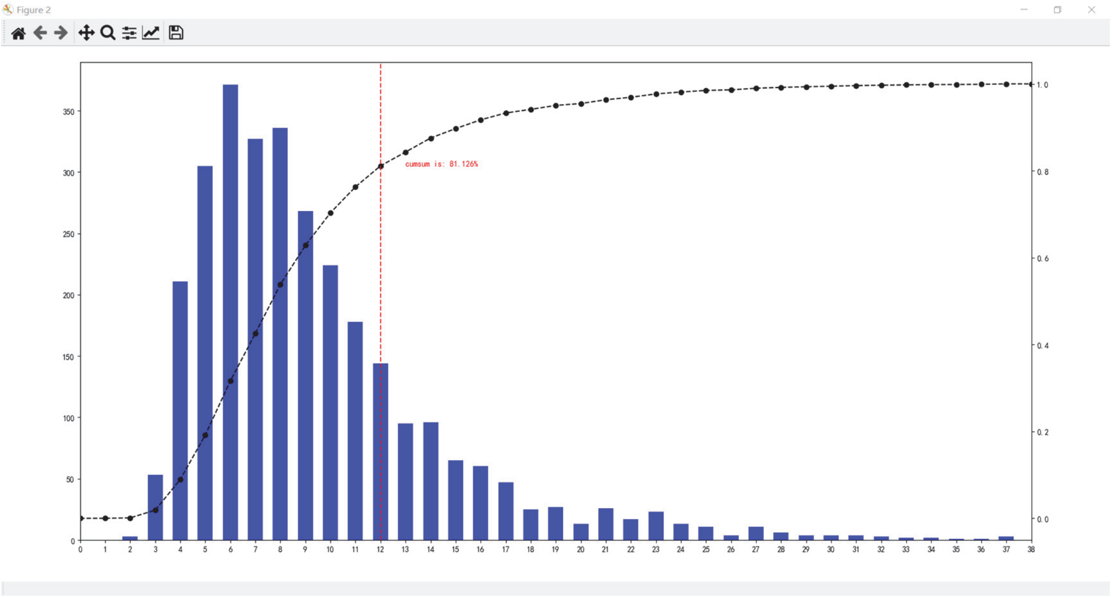


```
20    0.954744
21    0.963460
22    0.969159
23    0.976869
24    0.981227
25    0.984915
26    0.986255
27    0.989943
28    0.991954
29    0.993295
30    0.994636
31    0.995977
32    0.996983
33    0.997653
34    0.998324
35    0.998659
36    0.998994
37    1.000000
38    1.000000
dtype: float64
超过80%累计占比的节点值:  650
超过80%累计占比的节点值索引位置为:  12
------

Process finished with exit code 0
```

图表：





结论：发现比例为 81.126%，符合二八定律。