Python期末大作业：租房数据分析

1. 项目背景

本项目的目标是抓取链家官网北上广深4个一线城市，再加上苏州的全部租房数据并分析。

2. 实验环境

- Windows11
- Visual Studio Code

数据来源：链家官网，中经数据，人社通

3. 数据爬取

数据爬取使用Scrapy框架，基于之前作业爬取链家二手房数据的程序。

经过观察，链家网站最多显示100页，以万为单位的数据量显示不全，必须通过先获得区域 (district) 再获得板块(plate) 缩小范围获得url，以爬取全部数据。同时，不同区会显示相同的板块，需要去重。

工作流程：

1. 初始化爬虫
2. 开始请求
3. 解析district
4. 解析plate
5. 解析租房信息

核心代码：

```
class LianjiaSpider(scrapy.Spider):
    """
    爬取流程：城市首页 -> 区域列表 -> 板块列表 -> 房源详情
    """
    name = "rent_house"
    allowed_domains = ["lianjia.com"]
    city_name = "bj"
    # city_name = "sh"
    # city_name = "gz"
    # city_name = "sz"
    # city_name = "su"

    def start_requests(self):
        url = get_districts(self.city_name)
        yield  SeleniumRequest(url=url,  callback=self.parse_district,  meta={'city_name': self.city_name, 'proxy': self.settings.get('HTTP_PROXY')})

    def parse_district(self, response):
        city_name = response.meta['city_name']
        html_string = response.text
        html_tree = etree.HTML(html_string)
        districts = html_tree.xpath('//ul[@data-target="area" and @class=""]/li/a/@href')
        districts = [district.split("/")[2] for district in districts]

        d_urls = get_start_urls(city_name, districts)
        for url in d_urls:
            yield  SeleniumRequest(url=url,  callback=self.parse_plate,  meta={'city_name': city_name, 'proxy': response.meta['proxy']})
```

```
    def parse_plate(self, response):
        city_name = response.meta['city_name']
        html_string = response.text
        html_tree = etree.HTML(html_string)
        plates = html_tree.xpath('//ul[@data-target="area" and not(@class)]/li/a/@href')
        plates = [plate.split("/")[2] for plate in plates]

        seen_plates = set() # 去重
        for plate in plates:
            if plate not in seen_plates:
                seen_plates.add(plate)
                url = "https://{}.lianjia.com/zufang/{}/pg1/".format(city_name, plate)
                yield  SeleniumRequest(url=url,  callback=self.parse,  meta={'city_name':
city_name, 'plate': plate, 'proxy': response.meta['proxy']})

    def parse(self, response):
        city_name = response.meta['city_name']
        plate = response.meta['plate']
        # 检查页面是否为空
        isempty = bool(response.xpath('//div[@class="content__list--empty"]'))
        if not isempty:
            page = response.url.split("/")[-2]
            for each in response.xpath('//div[@class="content__list--item"]'):
                item = RentItem()
                item['plate']       =       each.xpath('.//p[@class="content__list--item--
des"]/a[2]/text()').get()
                if item['plate'] is not None:
                    item['plate'] = item['plate'].strip()
```

```
                description = each.xpath('.//p[@class="content__list--item--des"]').xpath('string(.)').get()
                if description:
                    description = ''.join(description).replace('\n', '').strip()
                    parts = [part.strip() for part in description.split('/') if part.strip()]
                    if len(parts) >= 4:
                        item['orien'] = ' '.join(parts[2].split())
                        item['house_type'] = parts[3]
                        item['area'] = parts[1].replace('㎡', '').strip() + "平方米"

                item['rental'] = each.xpath('.//span/em/text()').get() + "元/月"
                yield item

            if page != "pg100":
                next_page = int(page[2:]) + 1
                next_url = "https://{}.lianjia.com/zufang/{}/pg{}/".format(city_name, plate, next_page)
                yield scrapy.Request(next_url, callback=self.parse, meta={'city_name': city_name, 'plate': plate, 'page': next_page, 'proxy': response.meta['proxy']})
```

本次爬虫直接沿用了之前作业写的Selenium和header等设置，删除了cookie，添加了隧道代理IP的使用。

最终爬取数据量：

北京 90971

上海 53597

广州 106282

深圳 53743

苏州 50547

4. 数据处理

直接爬取到的数据中，包含重复数据和板块为 null 的与其他租房信息格式不同的数据，以及爬虫时遇到的数据损坏，所以有必要进行数据清洗。在后续数据分析时发现的一些极端数据（如面积大于2000平方米的数据）是真实的并没有在清洗时删除。同时，保留面积和房租的数值部分。

核心代码：

```
# 过滤板块为 null 的数据和非法数据
filtered_data = [item for item in data if item.get('plate') is not None and item.get('area')
and item['area'][0].isdigit()]


# 面积和房租转换为数值
    for item in filtered_data:
        item['area'] = float(item['area'].replace('平方米', ''))
        item['rental'] = float(item['rental'].replace('元/月', '').replace(',', ''))


# 数据量大，使用 set 去重提高性能
seen = set()
cleaned_data = []
for item in filtered_data:
    unique_key    =    (item.get('plate'),    item.get('orien'),    item.get('house_type'),
item.get('area'))
    if unique_key not in seen:
```

```
        seen.add(unique_key)
        cleaned_data.append(item)
```

5. 数据分析及可视化

5.1. 房租

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。采用合适的图或表形式进行展示。

设计：

• 读取数据：从 cleandata 目录中读取每个城市的租金数据文件（如 bj-data.json）使用 jsonlines 库逐行读取数据，并将其存储在 data 列表中

• 统计数据：总体租金和单位面积租金的均价、最高价、最低价、中位数

• 可视化：使用 matplotlib 库生成直方图，每个城市两个直方图分别是总体租金和单位面积租金频率分布

核心代码：

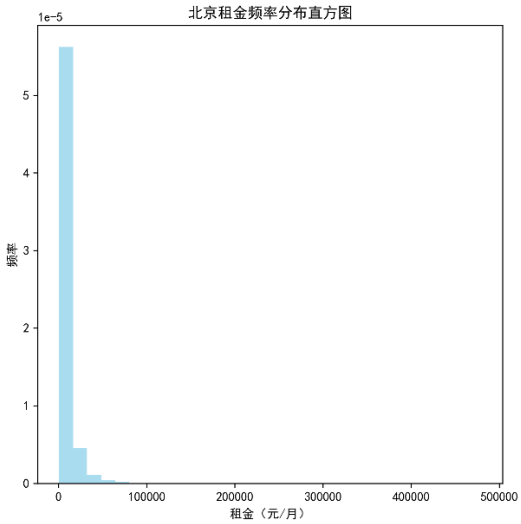
```
for city in city_names:
    # 读取数据 pass

    rental = [item['rental'] for item in data]
    area = [item['area'] for item in data]

    # 计算总体租金的统计数据（均价、最高价、最低价、中位数） pass
    # 计算单位面积租金的统计数据（均价、最高价、最低价、中位数） pass
    # 存储每个城市的统计数据 pass
```

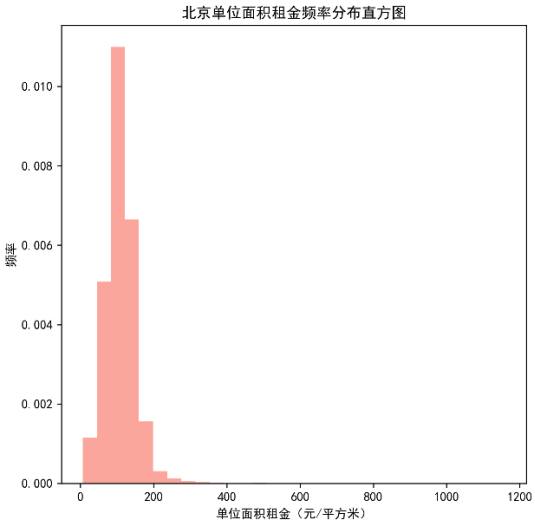
```
    # 绘制直方图
    plt.figure(figsize=(12, 6), dpi=300)

    # 总体租金直方图
    plt.subplot(1, 2, 1)
    plt.hist(rental, bins=30, color='skyblue', alpha=0.7, density=True)
    plt.title(f'{city_labels[city]}租金频率分布直方图')
    plt.xlabel('租金（元/月）')
    plt.ylabel('频率')

    # 单位面积租金直方图
    plt.subplot(1, 2, 2)
    plt.hist(rental_per_area, bins=30, color='salmon', alpha=0.7, density=True)
    plt.title(f'{city_labels[city]}单位面积租金频率分布直方图')
    plt.xlabel('单位面积租金（元/平方米）')
    plt.ylabel('频率')

    # 保存 pass
```

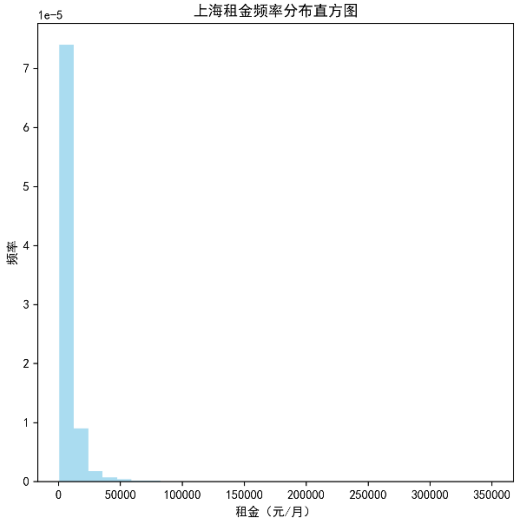
```
(test) PS C:\Users\NFsam\Desktop\11\CodePython\lianjiaSpider> python analyze1.py
北京:
租金均价：9617.05，最高价：480000.00，最低价：500.00，中位数：6900.00
单位面积租金均价：109.11，最高价：1160.54，最低价：6.20，中位数：106.45
上海:
租金均价：7983.30，最高价：350000.00，最低价：800.00，中位数：5600.00
单位面积租金均价：108.42，最高价：4100.00，最低价：9.15，中位数：102.02
广州:
租金均价：4946.60，最高价：250000.00，最低价：150.00，中位数：3600.00
单位面积租金均价：64.58，最高价：1000.00，最低价：2.17，中位数：57.33
深圳:
租金均价：8287.28，最高价：320000.00，最低价：960.00，中位数：5900.00
单位面积租金均价：99.94，最高价：990.98，最低价：1.00，中位数：90.91
苏州:
租金均价：3440.33，最高价：220000.00，最低价：300.00，中位数：2800.00
单位面积租金均价：39.04，最高价：279.07，最低价：1.00，中位数：35.37
```

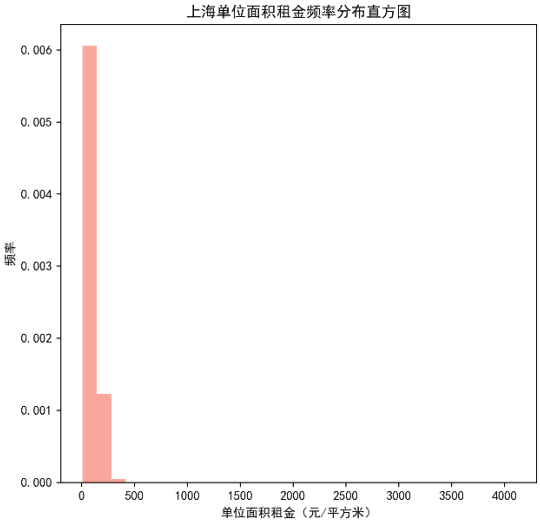
根据统计信息进行分析：

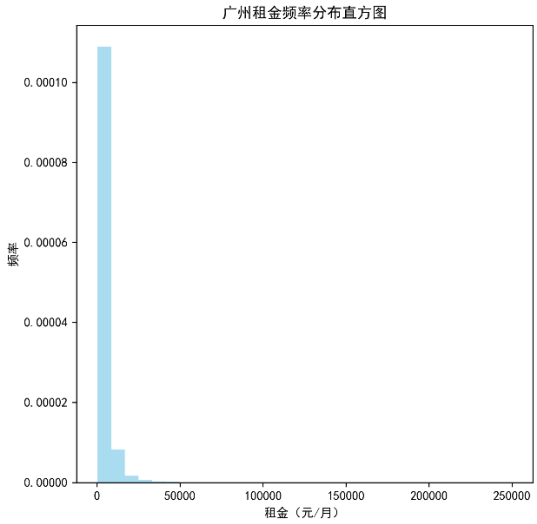
2022211414-黎昱彤

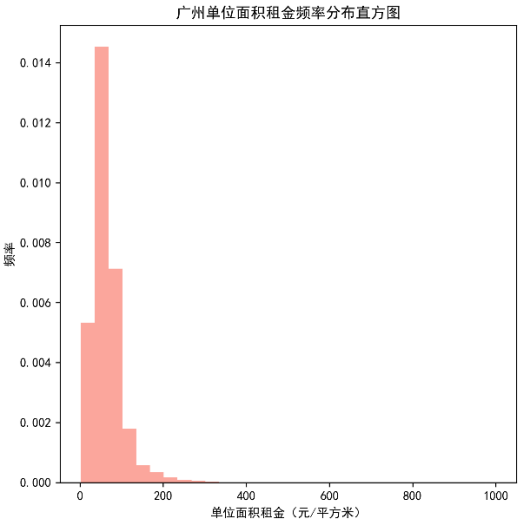


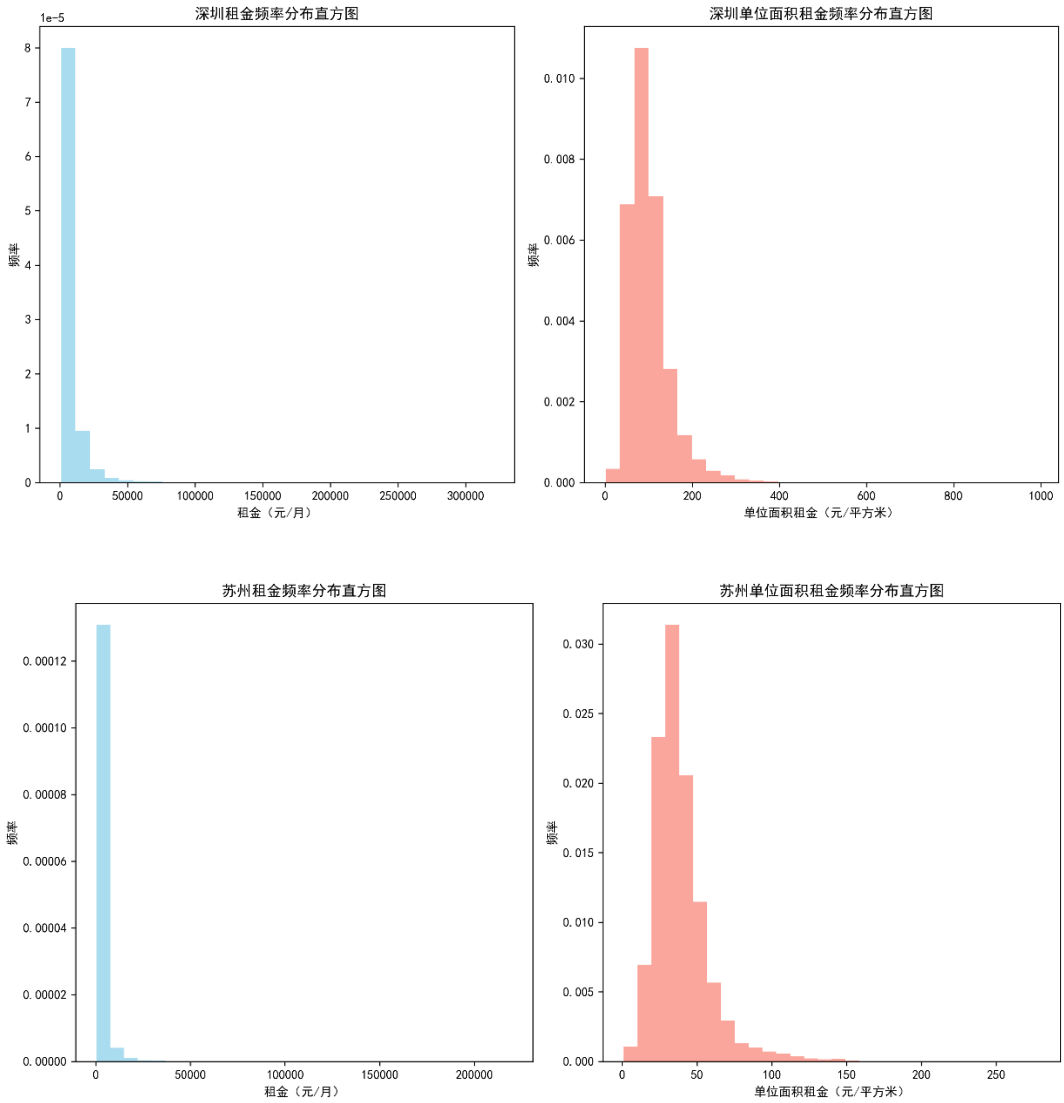












1. 均价

   北京 > 深圳 > 上海 > 广州 > 苏州

2. 最高价

   北京 > 上海 > 深圳 > 广州 > 苏州

3. 最低价

   深圳 > 上海 > 北京> 苏州 > 广州

4. 中位数

北京 > 深圳 > 上海 > 广州 > 苏州

5. 单位面积租金

基本维持和房租一样的规律，均价、最高价、中位数：北京、上海、深圳高于广州和苏州。

总体来看，可能是由经济发展、人口密度、城市面积导致的房租差异。北京、上海、深圳经济更发达，可能吸引人才从外地迁入，土地资源紧张，导致人口密度更大，房屋需求大，房租水平高。广州可能由于房屋供应更充足，苏州可能由于经济发展相对滞后，二者房租水平较低。

5.2. 房型

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。

设计：

• 读取数据：根据房型将租金数据分类存储在不同的列表中

• 统计数据：定义 calculate_stats 计算租金数据的统计信息，每种房型调用

• 可视化：使用 matplotlib 库生成直方图，每个城市用三种颜色表示三种房型租金分布

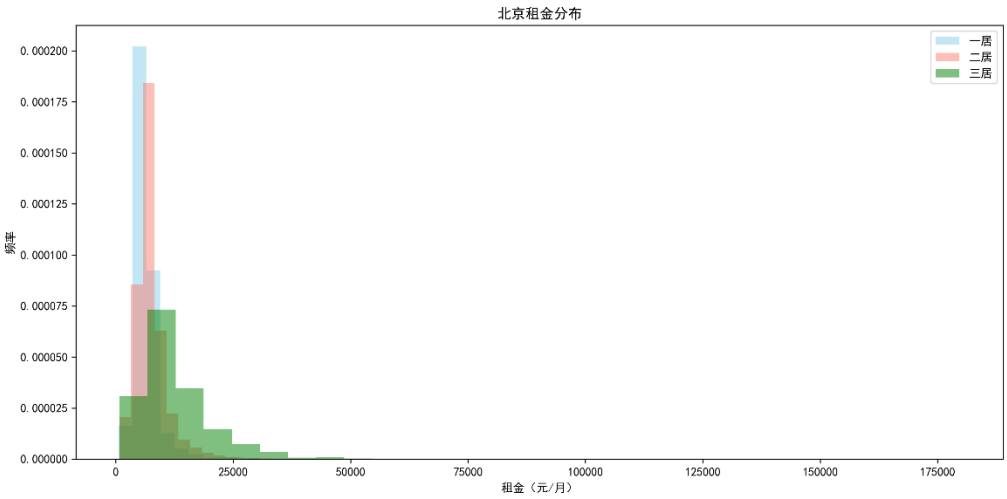
核心代码：

```
for city in city_names:
    # 读取数据（筛选一居、二居、三居） pass


    # 计算租金统计数据
    def calculate_stats(rental_data):
        mean = np.mean(rental_data)
        max_value = np.max(rental_data)
        min_value = np.min(rental_data)
        median = np.median(rental_data)
```

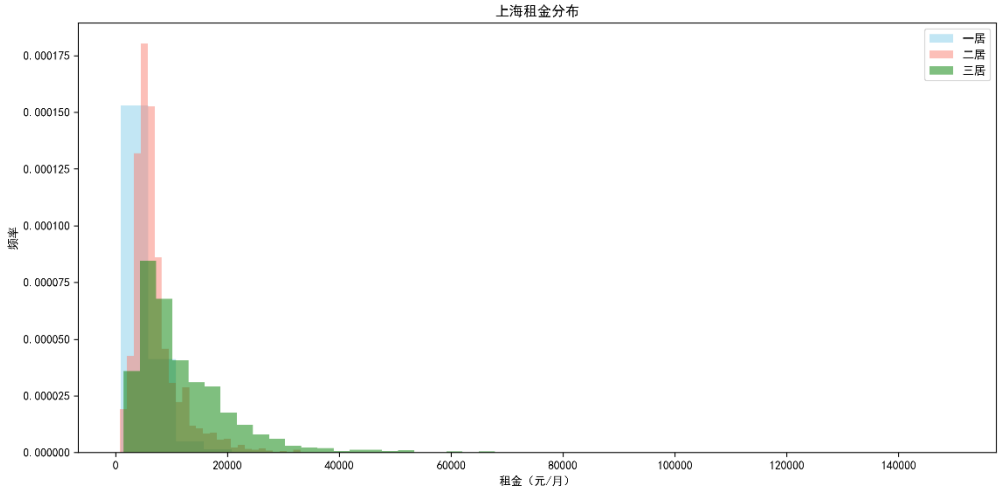
```
    return mean, max_value, min_value, median


stats_1b = calculate_stats(rental_1b)
stats_2b = calculate_stats(rental_2b)
stats_3b = calculate_stats(rental_3b)
statistics[city] = {
    '1b': stats_1b,
    '2b': stats_2b,
    '3b': stats_3b
}

# 绘制直方图
plt.figure(figsize=(12, 6), dpi=300)
plt.hist(rental_1b, bins=30, color='skyblue', alpha=0.5, label='一居', density=True)
plt.hist(rental_2b, bins=30, color='salmon', alpha=0.5, label='二居', density=True)
plt.hist(rental_3b, bins=30, color='green', alpha=0.5, label='三居', density=True)

# 图表设置
plt.title(f'{city_labels[city]}租金分布')
plt.xlabel('租金（元/月）')
plt.ylabel('频率')
plt.legend()

# 保存 pass
```

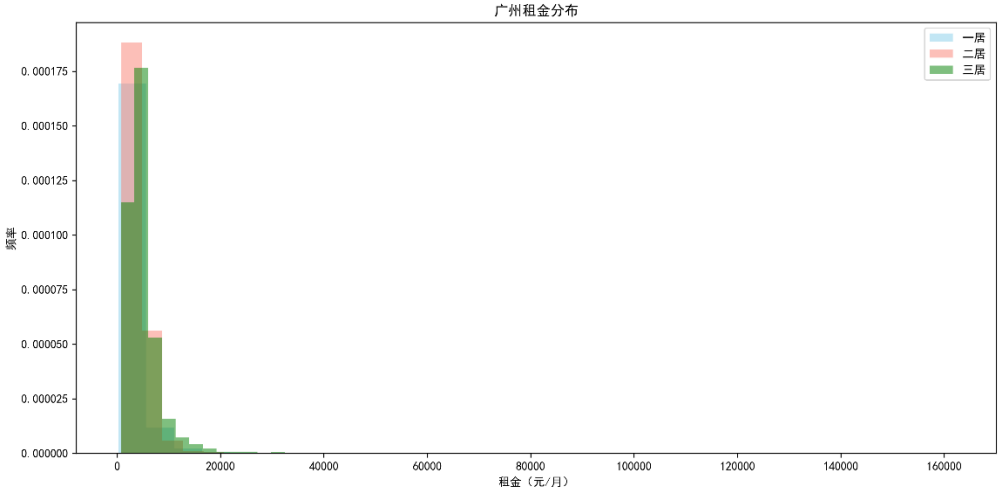
```
(test) PS C:\Users\NFsam\Desktop\11\CodePython\lianjiaSpider> python analyze2.py
北京：
一居
均价：6429.93，最高价：90000.00，最低价：500.00，中位数：5900.00
二居
均价：7549.18，最高价：76000.00，最低价：750.00，中位数：6990.00
三居
均价：12731.29，最高价：180000.00，最低价：800.00，中位数：10500.00
上海：
一居
均价：5266.21，最高价：150000.00，最低价：900.00，中位数：4700.00
二居
均价：7114.10，最高价：38000.00，最低价：800.00，中位数：5930.00
三居
均价：12141.42，最高价：88000.00，最低价：1500.00，中位数：9500.00
广州：
一居
均价：3361.67，最高价：162000.00，最低价：200.00，中位数：2500.00
二居
均价：4084.65，最高价：120000.00，最低价：800.00，中位数：3600.00
三居
均价：5168.47，最高价：80000.00，最低价：720.00，中位数：4320.00
深圳：
一居
均价：4530.05，最高价：80000.00，最低价：1300.00，中位数：3800.00
二居
均价：6649.37，最高价：65000.00，最低价：1300.00，中位数：6000.00
三居
均价：8496.94，最高价：145000.00，最低价：1300.00，中位数：7000.00
苏州：
一居
均价：2261.30，最高价：40000.00，最低价：300.00，中位数：2000.00
二居
均价：2661.05，最高价：33000.00，最低价：600.00，中位数：2500.00
三居
均价：3370.32，最高价：80000.00，最低价：500.00，中位数：3100.00
```

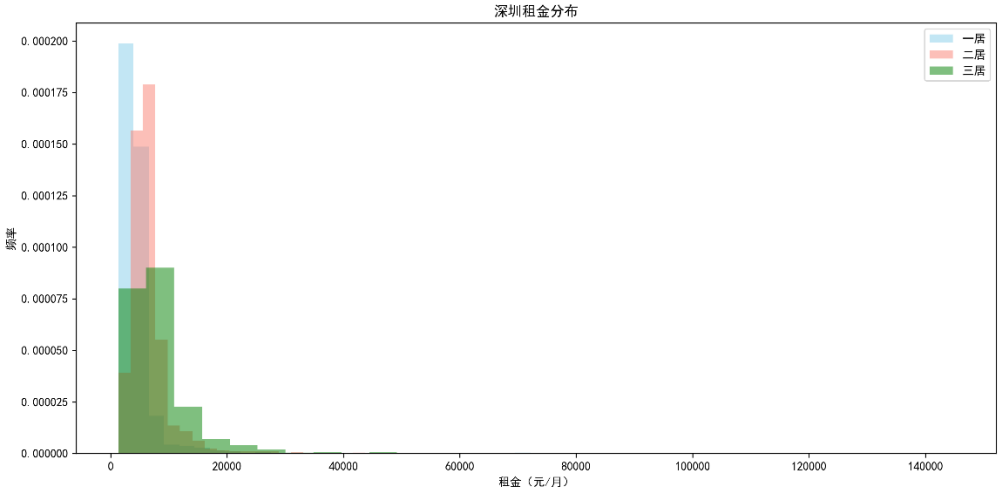
根据统计信息进行分析：

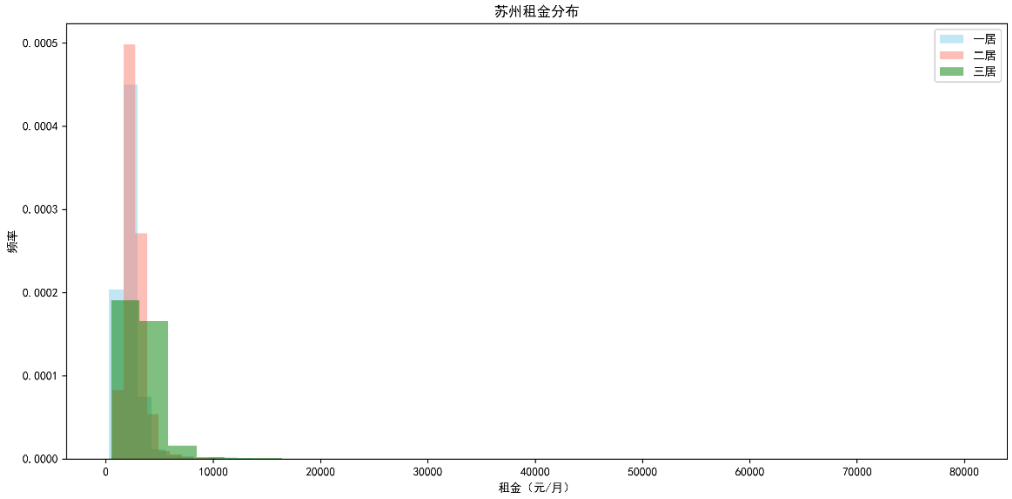


2022211414-黎昱彤









1. 均价和中位数

   北京 > 上海 > 深圳 > 广州 > 苏州

   北京的均价整体最高，尤其三居远超其他城市。苏州的均价明显低于北京、上海、深圳和广州。特别是一居和二居的均价相对较低，显示出其房地产市场相对平稳，且价格较为亲民。

2. 最高价和最低价

   根据不同房型有些特殊，最高价和最低价都是苏州最低，广州也处于较低水平，但相较于苏州，广州的低价房型数量较多，体现出其市场的价格较为多样化。

总体来看，北京的房价无论是均价还是最高价均领先于其他城市，反映了其作为首都和一线城市的地位；上海紧随其后，深圳在高端房型市场上也具有较强的竞争力；广州在价格波动上较大，但低价房型较为丰富；苏州则以较低的房价和相对平稳的市场为特色。

5.3. 板块

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。

设计：

- 读取数据：同5.1

•统计数据：计算每个板块的平均租金，存储在 plate_avg_rentals 字典中；将板块按平均租金从高到低排序，存储在 sorted_plates 列表中

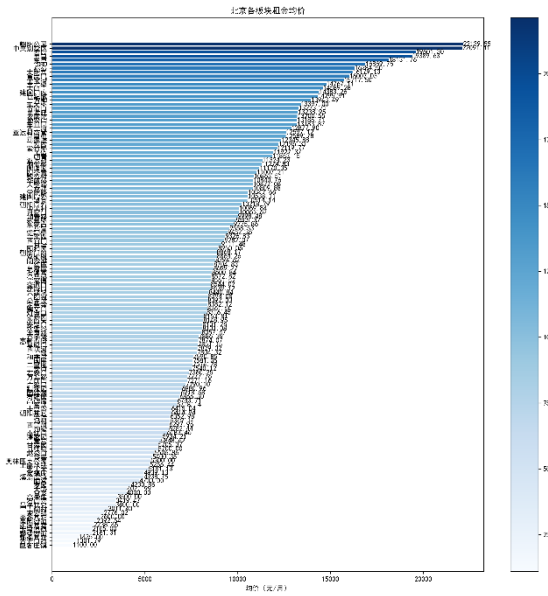
• 可视化：使用 matplotlib 库生成柱状图，每个城市生成一个柱状图，显示不同板块的平均租金；在柱状图旁边添加颜色条，显示颜色深浅与租金高低的对应关系

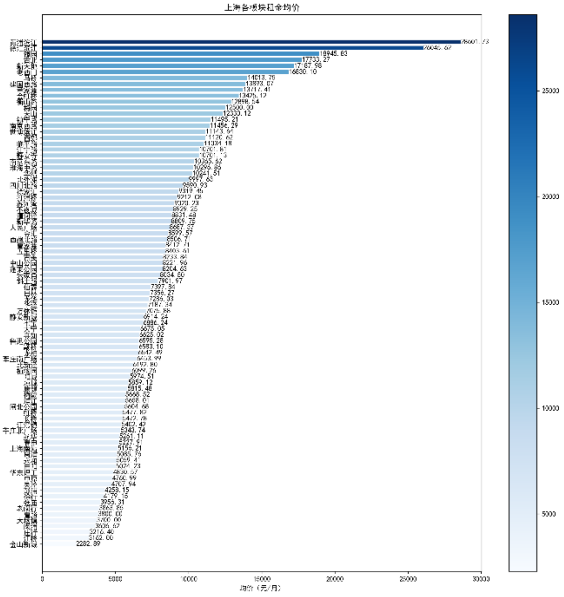
核心代码：

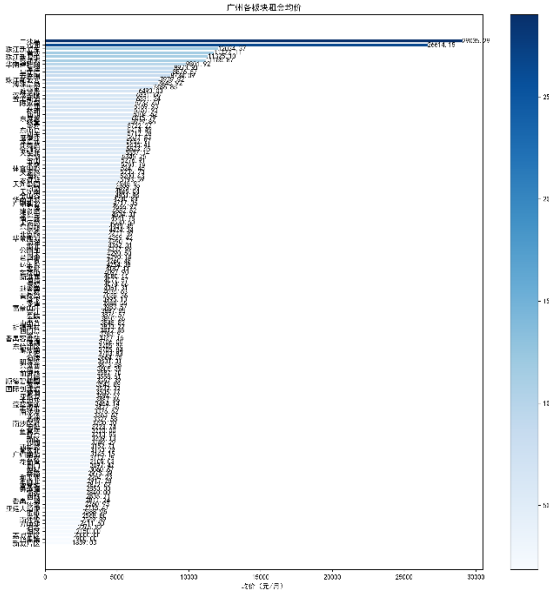
```
for city in city_names:
    # 读取数据 pass
    # 计算每个板块的均价并排序 pass

    # 绘制板块均价柱状图
    fig, ax = plt.subplots(figsize=(12, 12), dpi=300)
    plates, avg_rentals = zip(*sorted_plates)

    # 创建渐变色 pass

    bars = ax.barh(plates, avg_rentals, color=colors, height=0.7)
    for bar, avg_rental in zip(bars, avg_rentals):
        ax.text(bar.get_width() + 10, bar.get_y() + bar.get_height() / 2,
f'{avg_rental:.2f}', va='center')
    ax.set_xlabel('均价（元/月）')
    ax.set_title(f'{city_labels[city]}各板块租金均价')
    ax.invert_yaxis() # 反转 Y 轴，使均价最高的板块在最上面
    # 颜色条
    cbar = fig.colorbar(sm, ax=ax)

    # 保存 pass
```

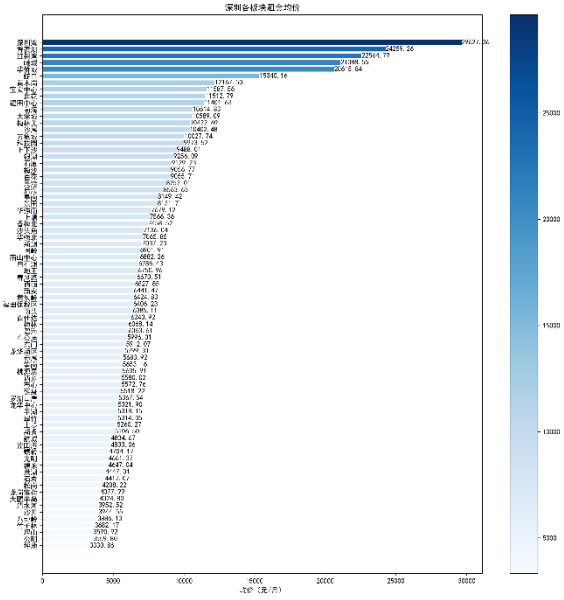
根据统计信息进行分析：

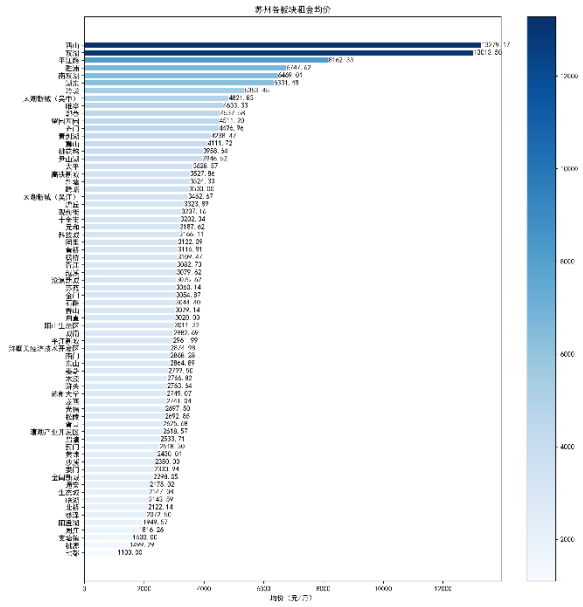
2022211414-黎昱彤











• 北京：房租最高的板块在朝阳公园、中央别墅区、西单、东单等核心区域，分析核心区域含有外交设施，地段优越、交通便利、配套设施完善，吸引了高端人才和企业入住。昌平、平谷等郊区房租较低，主要原因是距离市中心较远、交通不便、配套设施不完善。

• 五个城市各个板块的租金水平均有较大差异，从颜色上看，广州各个板块差异最大，大部分租金较低，有个别板块租金很高；北京总体租金更高。

5.4. 朝向与租金

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。

设计：

• 读取数据：同5.1

• 统计数据：遍历数据，提取每个房源的朝向、租金和面积，计算单位面积租金并其添加到对应朝向的租金列表中；对每个朝向的单位面积租金列表计算平均值，再排序

• 可视化：使用 matplotlib 库生成箱线图。箱线图能显示数据的分布情况，包括中位数、四分位数、最大值和最小值。通过箱线图，可以直观地看到不同朝向的单位面积租金分布情况，并找出最高和最低的朝向。

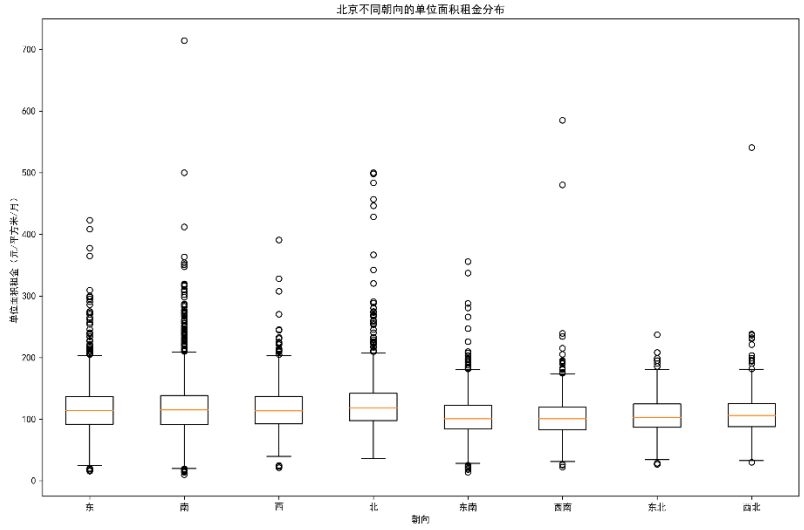
核心代码：

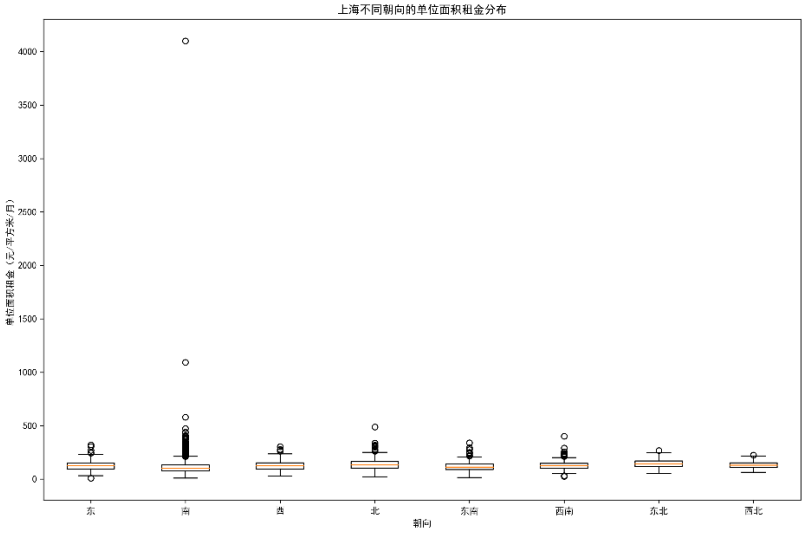
```
for city in city_names:

    # 读取数据  pass

    # 计算每个朝向的单位面积租金 pass


    # 绘制箱线图

    fig, ax = plt.subplots(figsize=(12, 8), dpi=300)

    data_to_plot = [orientation_rentals[orientation] for orientation in orientation_labels]

    ax.boxplot(data_to_plot, labels=orientation_labels)

    ax.set_xlabel('朝向')

    ax.set_ylabel('单位面积租金（元/平方米/月）')

    ax.set_title(f'{city_labels[city]}不同朝向的单位面积租金分布')


    # 保存 pass
```

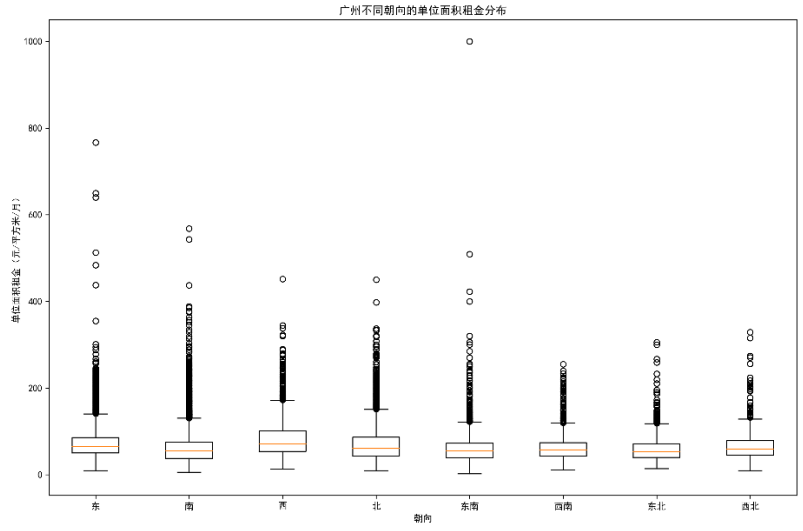
```
(test) PS C:\Users\NFsam\Desktop\11\CodePython\lianjiaSpider> python analyze4.py
北京:
北 东 西 南 西北 东北 东南 西南
最高：北，平均单位面积租金：123.95 元/平方米/月
最低：西南，平均单位面积租金：103.66 元/平方米/月
上海:
东北 北 西北 西南 西 东 东南 南
最高：东北，平均单位面积租金：147.32 元/平方米/月
最低：南，平均单位面积租金：107.49 元/平方米/月
广州:
西 东 北 西北 西南 南 东北 东南
最高：西，平均单位面积租金：88.65 元/平方米/月
最低：东南，平均单位面积租金：60.01 元/平方米/月
深圳:
西 北 东 东北 西北 南 西南 东南
最高：西，平均单位面积租金：129.02 元/平方米/月
最低：东南，平均单位面积租金：91.09 元/平方米/月
苏州:
北 西北 西 东 东北 西南 南 东南
最高：北，平均单位面积租金：57.54 元/平方米/月
最低：东南，平均单位面积租金：35.80 元/平方米/月
```

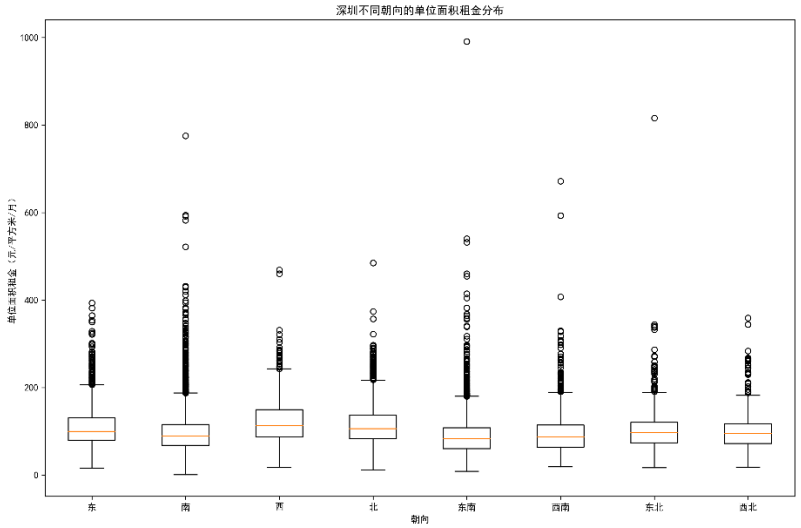
根据统计信息进行分析：

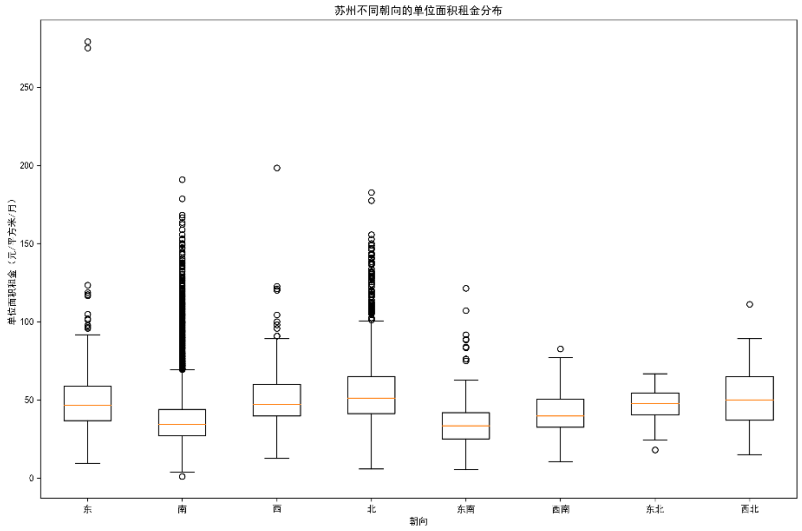
2022211414-黎昱彤











- 北京：最高朝向-北；最低朝向-西南
- 上海：最高朝向-东北；最低朝向-南
- 广州：最高朝向-西；最低朝向-东南
- 深圳：最高朝向-西；最低朝向-东南
- 苏州：最高朝向-北；最低朝向-东南

总体来看，五个城市的具体朝向的租金差异并不完全一致，但是偏向北的朝向租金更高，偏向南的朝向租金更低。广东省的两个城市保持一致。对于不同城市来说，地理位置、经济背景、租赁市场的成熟度和政府政策等都在影响朝向租金的分布和波动，反

映出城市的经济发展水平、房屋供需、区域规划、市场需求和基础设施等多方面因素的影响。

5.5. GDP 与租金

查询各个城市的人均GDP，分析并展示其和单位面积租金分布的关系。

设计：

• 用折线图展示各个城市的单位面积租金和人均GDP

• 将租房性价比定义为单位面积租金与人均GDP的比值，用柱状图展示各个城市租房性价比

核心代码：

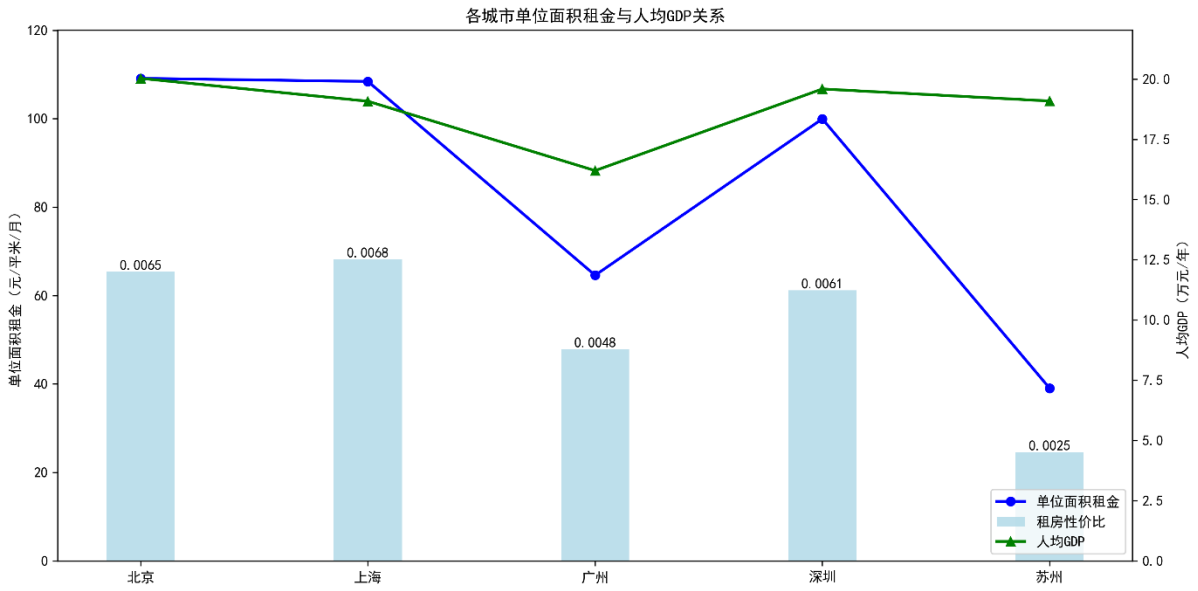
```
for city in city_names:
    # 读取数据  pass


rental_affordability = {}
for city in city_names:
    rental_affordability[city] = city_avg_rentals[city] / (city_gdp_person[city] * 10000 / 12)


# 创建双 Y 轴图表
fig, ax1 = plt.subplots(figsize=(12, 6), dpi=300)
ax2 = ax1.twinx()


# 绘制柱状图（租房性价比）
x = np.arange(len(city_names))
width = 0.3
rental_affordability_scaled = [rental_affordability[city] * 10000 for city in city_names] # 调整柱子的高度
```

```
rects = ax1.bar(x, rental_affordability_scaled, width, label='租房性价比', alpha=0.8, color='lightblue')

# 单位面积租金的折线图
ax1.plot(x, [city_avg_rentals[city] for city in city_names], 'bo-', label='单位面积租金', linewidth=2)

# 人均GDP的折线图
ax2.plot(x, [city_gdp_person[city] for city in city_names], 'g^-', label='人均 GDP', linewidth=2)

# 设置x轴标签
ax1.set_xticks(x)
ax1.set_xticklabels([city_labels[city] for city in city_names])

# 设置y轴范围
ax1.set_ylim(0, max(city_avg_rentals.values()) * 1.1)
ax2.set_ylim(0, max(city_gdp_person.values()) * 1.1)

# 设置标题和标签 pass
# 图例 pass

# 在柱状图上添加数值标签
for i, rect in enumerate(rects):
    height = rect.get_height()
    original_value = rental_affordability[city_names[i]]
    ax1.text(rect.get_x() + rect.get_width() / 2., height,
             f'{original_value:.4f}',
             ha='center', va='bottom')
```

```
# 保存 pass
```

```
(test) PS C:\Users\NFsam\Desktop\11\CodePython\lianjiaSpider> python analyze5.py
各城市租房性价比排名:
苏州: 0.0025
广州: 0.0048
深圳: 0.0061
北京: 0.0065
上海: 0.0068
```

根据统计信息进行分析：



人均GDP较高的城市（如北京、上海、深圳）其单位面积租金也相对较高，而人均GDP较低的城市（如苏州）单位面积租金则显著降低。

苏州的租房性价比是最高的，可能的原因如下：

- 苏州的人均GDP虽然较低，但其租金也远远低于其他一线城市，表明住房成本相对于收入水平来说更低
- 苏州作为新一线城市，虽然经济发展较快，但整体房价和租金仍然保持在较低水平。这可能与其产业结构和人口需求有关。相比北上深，苏州的产业更偏向于制造业和传统经济，导致租金压力较小
- 供需平衡较好

5.6. 工资与租金

查询各个城市的平均工资，分析并展示其和单位面积租金分布的关系。

设计：

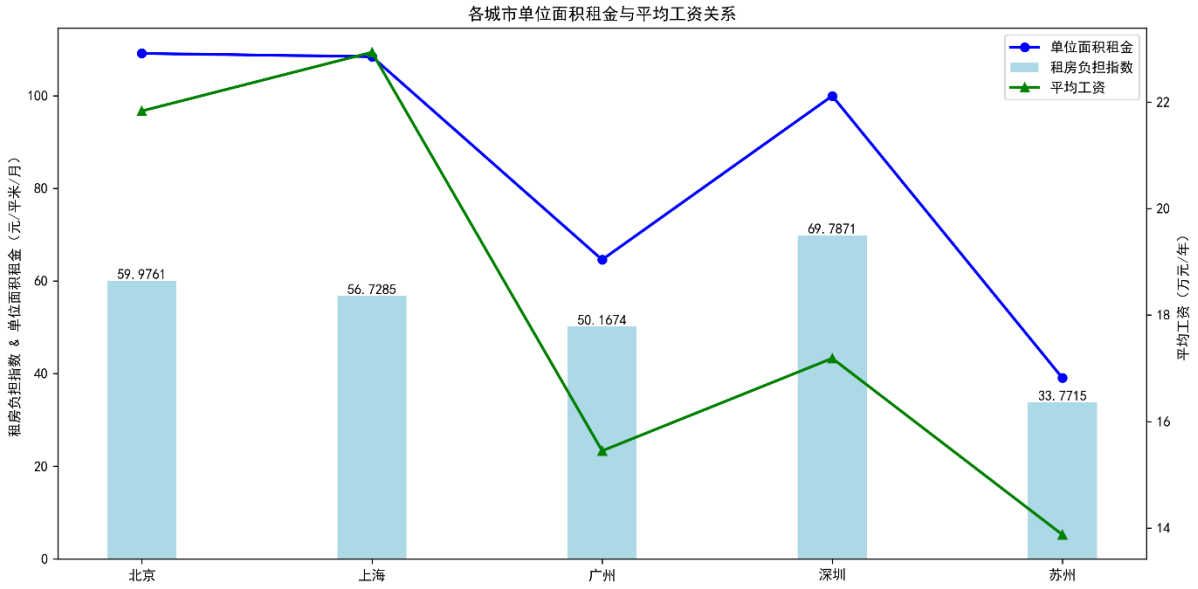
• 用折线图展示各个城市的单位面积租金和平均工资

• 将租房的负担定义为单位面积租金与平均工资的比值，用柱状图展示各个城市租房负担

核心代码：和5.5同理，略

```
(test) PS C:\Users\NFsam\Desktop\11\CodePython\lianjiaSpider> python analyze6.py
各城市租房负担排名：
深圳
租房负担指数：0.0070，平均工资：171854 元/年，单位面积租金：99.94 元/平米/月
北京
租房负担指数：0.0060，平均工资：218312 元/年，单位面积租金：109.11 元/平米/月
上海
租房负担指数：0.0057，平均工资：229337 元/年，单位面积租金：108.42 元/平米/月
广州
租房负担指数：0.0050，平均工资：154475 元/年，单位面积租金：64.58 元/平米/月
苏州
租房负担指数：0.0034，平均工资：138732 元/年，单位面积租金：39.04 元/平米/月
```

根据统计信息进行分析：



‘平均工资’折线和‘单位面积租金’折线的趋势基本一致，和5.5的‘人均GDP’折线相比，可以说明平均工资和单位面积租金大致成正相关关系。

深圳的租房负担是最重的，可能的原因如下：

- 深圳的高收入主要集中于高科技产业，这些行业集中的核心区域租金价格极高。说明租金的涨幅相比工资更为明显
- 深圳作为一线城市，人口流入持续增加，土地供应有限，导致住房需求旺盛，租金水平长期维持高位

结合5.5，苏州租房性价比是最高，租房负担最轻。相较于一线城市，苏州为租房者提供了更舒适、经济的居住条件。

6. 实验总结

本次实验大概 60%的时间花在应对链家反爬机制上。我尝试了各种方法，总是很慢（200 items/min）我任务是 Selenium 动态获取的问题，但是爬到 2 万会被重定向崩溃。最后尝试的是代理池，接入这一点也费了一番功夫。频繁切换 IP 确实有所帮助，最后能很快地爬到数据页没再触发 302。当最终进入分析与可视化阶段时，实验推进顺利了很多。可视化分析不仅揭示了各城市租房的各种差异，还让我对城市发展的特点有了新的认识。虽然这次实验我认为在繁琐的技术细节中进行了过度消耗，但我相信这些经历将在未来的数据科学学习和实践中，为我打下坚实的基础。