课程设计 3：安居客租房数据爬取与可视化（pyquery）

一、准备过程

从安居客租房网站获取房源信息（使用 pyquery 解析库），包括房源标题、价格、位置、户型等数据，并将爬取的数据存储为 CSV 文件。通过对数据进行清洗和分析，结合可视化工具（如 matplotlib 和 seaborn），生成租房价格分布、区域房源数量等图表。

使用 pyquery 进行解析的主要原因是其语法简洁且功能强大，类似于 jQuery 的选择器机制，能够高效地解析 HTML 文档并提取目标数据。对于安居客租房网站这种结构复杂的网页，pyquery 可以通过 CSS 选择器精准定位房源信息（如标题、价格、位置等），极大地简化了解析过程。此外，pyquery 与 Python 生态集成，结合 requests 库可以快速实现从网页请求到数据提取的完整流程，非常适合中小规模的爬虫项目。图 1 为该网站详情。

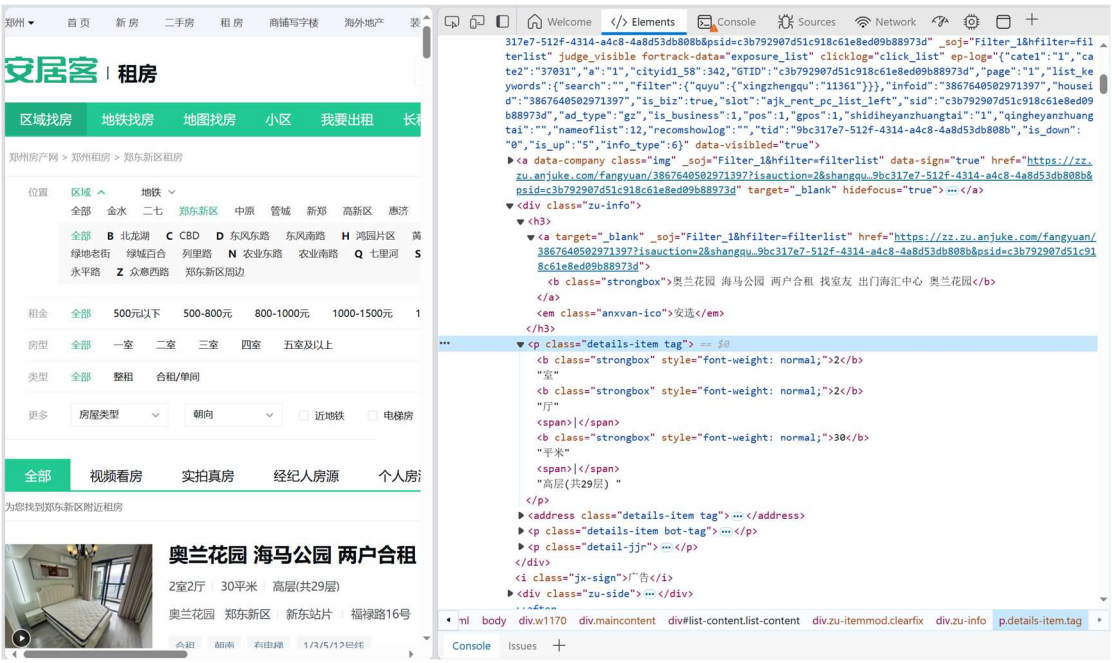

图 1：网站详情

二、实现过程

首先导入必要的库。requests 用于发送 HTTP 请求，time 和 random 用于模拟人类操作以避免反爬虫机制，pandas 用于数据处理，pyquery 用于解析 HTML 文档，matplotlib 和 seaborn 用于数据可视化。

```
import requests
import time
import random
import pandas as pd
from pyquery import PyQuery as pq
```

```
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams['font.sans-serif'] = ['SimHei']
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

通过爬虫从安居客租房网站获取多页房源信息，并提取每套房源的详细数据（如标题、房间数、厅数、面积、位置、价格等）。首先定义了一个 get_url 函数，用于生成目标网站的多页 URL 列表；接着通过 get_info 函数发送 HTTP 请求获取网页内容，并使用 pyquery 解析 HTML 文档，通过 CSS 选择器精准提取房源信息，最终将每套房源的数据存储为列表形式返回。

```
def get_url(url_start, page_nums):
    # 获取网站连接
    urls = []
    urls.append(url_start)
    for i in range(2, page_nums+1):
        urls.append(f'{url_start}/p{i}')
    return urls

def get_info(url):
    '''获取租房详细信息，返回的是一个列表'''
    # 获取网页内容
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36
Edg/131.0.0.0'}
    request1 = requests.get(url, headers=headers)
    # 使用 pyquery 解析网页内容
    d = pq(request1.content)
    d = d('div').filter('.zu-itemmod')
    # 提取房源信息
    detail = []
    for i in d.items():
        title = i('h3 b.strongbox').text()  # 房源标题
        rooms = i('p.details-item.tag b.strongbox').eq(0).text()  # 房间
数
        halls = i('p.details-item.tag b.strongbox').eq(1).text()  # 厅数
        area = i('p.details-item.tag b.strongbox').eq(2).text()  # 面积
        other = i('p.details-item.bot-tag').text()
        location = i('address.details-item.tag a').text()  # 地点
        price = i('div.zu-side strong.price').text()  # 价格
        unit = i('div.zu-side span.unit').text()  # 单位
        contact = i('span.jjr-info').text()  # 经纪人姓名
```

```
        detail.append([title, rooms, halls, other, area, location,
price, unit, contact])
        return detail
```

通过循环遍历生成的 URL 列表，依次爬取安居客租房网站中郑州郑东新区区域的 50 页房源信息，并将每页爬取的数据通过 pandas 的 DataFrame 进行整合。在爬取过程中，通过 time.sleep 和 random 模块实现了随机延迟，以避免触发反爬虫机制。最后，所有爬取到的房源信息保存到 anjuke.csv 文件中。

```
url = 'https://zz.zu.anjuke.com/fangyuan/zhengdongxinqu'
urls = get_url(url, 50)

zufang = pd.DataFrame()
for i in urls:
    zufang = pd.concat([zufang, pd.DataFrame(get_info(i))])
    print(f'Page{i} finished!')
    print(zufang.info())
    time.sleep(random.randint(1,2))

# 保存数据
zufang.to_csv('anjuke.csv')
```

对从安居客租房网站爬取的数据进行清洗和整理，以便后续分析和使用。通过 pd.read_csv 读取之前保存的 CSV 文件，并删除不必要的列。对房源详细信息进行整理，通过自定义的 clean1 函数对原始数据进行拆分和分类处理。clean1 函数从原始字符串中提取出与房源相关的关键信息，包括是否靠近地铁、租房方式以及是否有电梯，并将这些信息分别存储到新的列中。如果原始数据中包含“线”字，相关信息会被提取并存储；如果包含“整租”或“合租”字样，则分别标记为相应的租房方式，否则默认标记为“独立单间”。根据是否包含“有电梯”字样来判断房源是否有电梯设施。

处理完成后，将清洗后的数据重新组织为一个结构化的 DataFrame，并为每一列设置清晰的列名（如“标题”、“室”、“厅”、“面积”、“小区”、“价格”、“联系人”、“地铁”、“出租方式”、“电梯”），以便后续分析和可视化。

```
# 数据清洗
anjuke = pd.read_csv('anjuke.csv')
# 重新排序
anjuke = anjuke.drop(columns=['Unnamed: 0']).drop(columns='7')
# 详细信息整理
anjuke[['9', '10', '11']] = None
def clean1(ori_data):
    clean_data = []
    split_data = ori_data.split(' ')
    # 地铁
    if '线' in ori_data:
```

```
        clean_data.append(split_data[-1])
    else:
        clean_data.append(None)
    # 租房方式
    if '整租' in ori_data:
        clean_data.append('整租')
    elif '合租' in ori_data:
        clean_data.append('合租')
    else:
        clean_data.append('独立单间')
    # 电梯
    if '有电梯' in ori_data:
        clean_data.append('有电梯')
    else:
        clean_data.append('无电梯')
    return clean_data
anjuke['3'] = anjuke['3'].apply(clean1)
for i in range(len(anjuke)):
    anjuke['9'][i] = anjuke['3'][i][0]
    anjuke['10'][i] = anjuke['3'][i][1]
    anjuke['11'][i] = anjuke['3'][i][2]
anjuke = anjuke.drop(columns='3')
# 设置列名
columns = ['标题', '室', '厅', '面积', '小区', '价格', '联系人', '地铁', '
出租方式', '电梯']
anjuke.columns = columns
anjuke.info()
```

清洗处理后的数据基本信息如下：

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1380 entries, 0 to 1379
Data columns (total 10 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   标题      1380 non-null   object
 1   室       1380 non-null   int64
 2   厅       1380 non-null   int64
 3   面积      1380 non-null   float64
 4   小区      1380 non-null   object
 5   价格      1380 non-null   int64
 6   联系人     1380 non-null   object
 7   地铁      1230 non-null   object
 8   出租方式    1380 non-null   object
 9   电梯      1380 non-null   object
dtypes: float64(1), int64(3), object(6)
memory usage: 107.9+ KB
```

图 2：数据基本信息

	标题	室	厅	面积	小区	价格	联系人	地铁	出租方式	电梯
0	双子塔旁 东站附近 东方鼎盛三期 轻奢风格家电齐全随时看房	3	2	30.0	东方鼎盛时代三期	500	李帅旗	1/3/5/12号线	合租	有电梯
1	暖气开放 拎包入住 无杂费 可月付短租 双子塔 升龙 凯丽	3	2	35.0	海马公园(B区)	700	马旭龙	1/3/5号线	合租	有电梯
2	绿地老街 建业总部港 宝龙广场 实图实价 拒绝假房源 可短租	3	2	30.0	老街绿地郑东新苑一期	1000	高俊龙	4/5/12号线	合租	有电梯
3	东十里铺双地铁 三号线 精装一居室 无中介费	1	1	46.2	红星美凯龙(商住楼)	1400	文世兴	3/4/12号线	整租	有电梯
4	新年新气象！！！优惠多多！福利多多！升龙广场 双子塔 东站	2	2	18.0	海马公园(D区)	400	闫浩瑜	1/3/5号线	合租	有电梯

图 3：数据详情

通过数据可视化对清洗后的租房数据进行分布分析。使用 matplotlib 和 seaborn 库创建了一个包含四个子图的画布，每个子图分别展示了租房数据中“室”、“厅”、“面积”和“价格”的分布情况。

设置画布的大小和分辨率，并定义了调色板颜色以增强可视化效果。通过 sns.histplot 函数绘制直方图，分别展示了“室”和“厅”的离散分布（使用 discrete=True 参数），以及“面积”和“价格”的连续分布。每个子图都添加标题以明确展示的内容。结果如图 4 所示。

```
# 数据可视化
plt.figure(figsize=(18, 4), dpi=300)
# 调色板
colors = sns.color_palette('rocket',n_colors=5)
colors2 = sns.color_palette('rocket',n_colors=3)
sns.set_context("notebook")

# 查看各数据分布
plt.subplot(1,4,1)
plt.title('室数据分布')
sns.histplot(x='室', data=anjuke,
             discrete=True, color=colors[0])
plt.subplot(1,4,2)
plt.title('厅数据分布')
sns.histplot(x='厅', data=anjuke,
             discrete=True, color=colors[0])
plt.subplot(1,4,3)
plt.title('面积数据分布')
sns.histplot(x='面积', data=anjuke, color=colors[0])
plt.subplot(1,4,4)
plt.title('价格数据分布')
sns.histplot(x='价格', data=anjuke, color=colors[0])
```

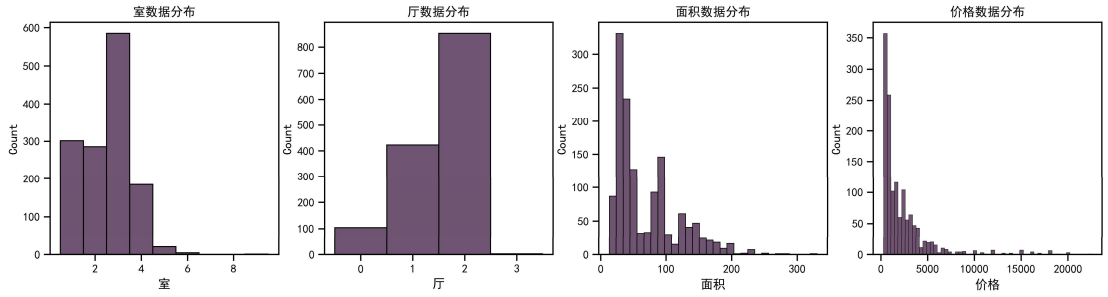

图 4：基本数据可视化

使用 seaborn 的 displot 函数绘制核密度估计（KDE）图，分别展示“面积”和“价格”在不同“出租方式”和“电梯”条件下的分布情况。通过设置multiple="stack"参数，不同类别的分布图叠加在一起，便于比较不同条件下的数据分布特征。

```
sns.displot(x='面积', data=anjuke, hue='出租方式',
            kind='kde', multiple="stack",
            height=10, aspect=1.5, palette=colors2)
sns.displot(x='面积', data=anjuke, hue='电梯',
            kind='kde', multiple="stack",
            height=10, aspect=1.5, palette=colors2)
sns.displot(x='价格', data=anjuke, hue='出租方式',
            kind='kde', multiple="stack",
            height=10, aspect=1.5, palette=colors2)
sns.displot(x='价格', data=anjuke, hue='电梯',
            kind='kde', multiple="stack",
            height=10, aspect=1.5, palette=colors2)
```

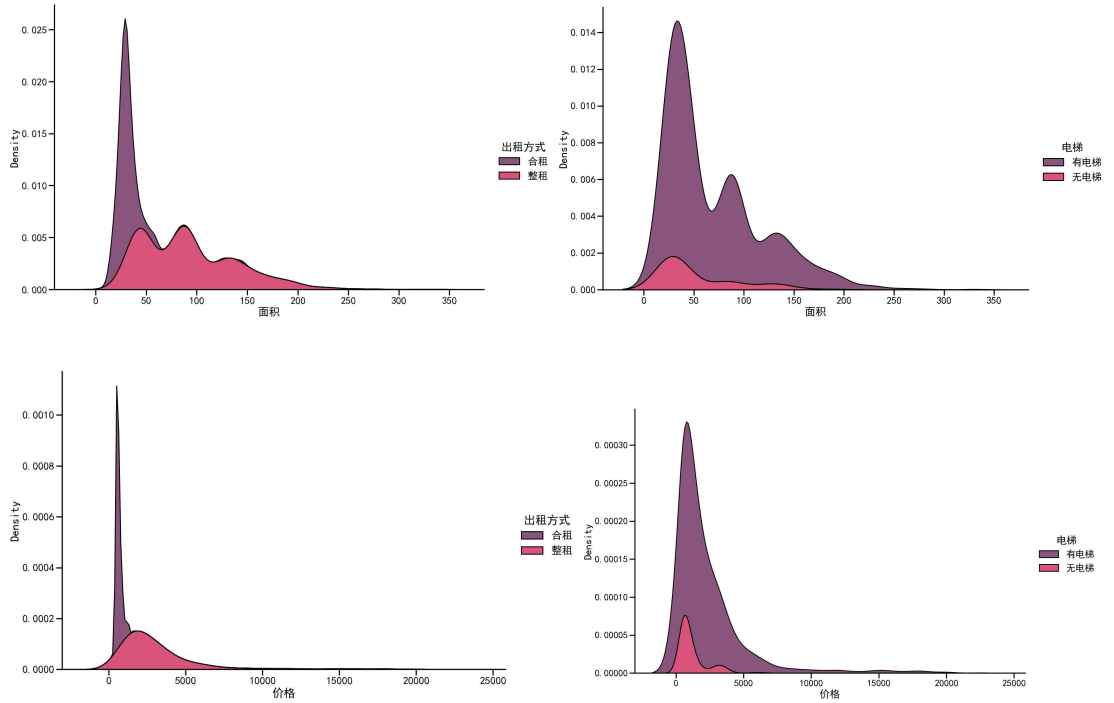

图 5：面积与价格影响因素可视化

使用 catplot 函数绘制了小提琴图（violin plot），展示“面积”与“出租方式”之间的关系，并以“电梯”作为分类变量进行细分。

```
sns.catplot(data=anjuke, x="面积", y="出租方式", hue="电梯",
            kind="violin", height=10, aspect=1.5, palette=colors2)
```

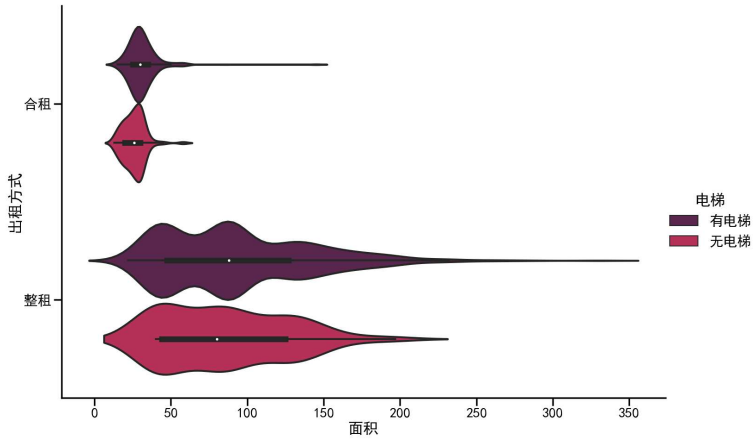

图 6：面积影响因素可视化