实践报告：
成都租房数据分析与可视化

学号：2213189

姓名：李懋

日期：2024.10.11

南开大学 软件学院

目 录

目录

1 实验介绍

根据课程网站上给出的例子,爬取成都市的租房价格信息,保存到 CSV 文件中.再将已有的数据进行数据分析和可视化,进而对成都市的租房情况有一个初步的了解.

2 实验目的

- 利用爬虫技术从链家网站上爬取成都市租房价格情况,将数据爬下储存到 CSV 文件中.后续再对数据读取,预处理清洗,可视化分析等操作.
- 进而可以直观的对成都租房价格,区域分布等等数据有个更深入的了解,帮助人们根据自己的条件,选择租房的最优解.

3 实验流程

1) 爬取成都市房价信息

- 使用 Selenium 库来爬取链家网站上成都市的租房信息，并将结果保存到一个 CSV 文件中。

2) 对已有的租房信息进行数据分析与可视化

1. 利用 pandas 库读取文件
2. 进行数据预处理,过滤数据
3. 使用 Matplotlib Seaborn 库进行数据可视化操作

4 实验结果或结论

1) 爬取成都市租房价格信息

4.1.1 导入相关的库

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import csv
import time
```

4.1.2 初始化 CSV 文件

```
# 初始化CSV文件
csv_file = open('chengdu_rent.csv', 'w', newline='', encoding='utf-8')
csv_writer = csv.writer(csv_file)
csv_writer.writerow(['标题', '价格', '面积', '区域', '链接'])
```

4.1.3 初始化 Selenium WebDriver

```
# 初始化Selenium WebDriver
driver = webdriver.Chrome()
```

4.1.4 爬取多页数据

```
# 爬取多页数据
for page in range(1, 101):  # 假设爬取前100页
    url = f'https://cd.lianjia.com/zufang/pg{page}/'
    print(f'正在爬取第{page}页: {url}')

    driver.get(url)
    time.sleep(2)  # 等待页面加载
```

4.1.5 查找租房信息,并在每段爬取后暂停一段时间

```
# 查找租房信息
rent_items = driver.find_elements(By.CSS_SELECTOR, value: 'div.content__list--item')

for item in rent_items:
    try:
        title = item.find_element(By.CSS_SELECTOR, value: 'p.content__list--item--title').text.strip()
        price = item.find_element(By.CSS_SELECTOR, value: 'span.content__list--item-price').text.strip()
        area = item.find_element(By.CSS_SELECTOR, value: 'p.content__list--item--des').text.strip().split('/')[1].strip()
        region = item.find_element(By.CSS_SELECTOR, value: 'p.content__list--item--des').text.strip().split('/')[0].strip()
        link = item.find_element(By.CSS_SELECTOR, value: 'a.content__list--item--aside').get_attribute('href')

        csv_writer.writerow([title, price, area, region, link])
    except Exception as e:
        print(f'解析错误: {e}')

# 每页爬取后暂停一段时间
time.sleep(2)
```

4.1.6 爬取完成,关闭 CSV 文件和 WebDriver

```
40  # 关闭CSV文件
41  csv_file.close()
42
43  # 关闭WebDriver
44  driver.quit()
45  print('爬取完成，数据已保存到chengdu_rent.csv')
```

2) 可视化分析

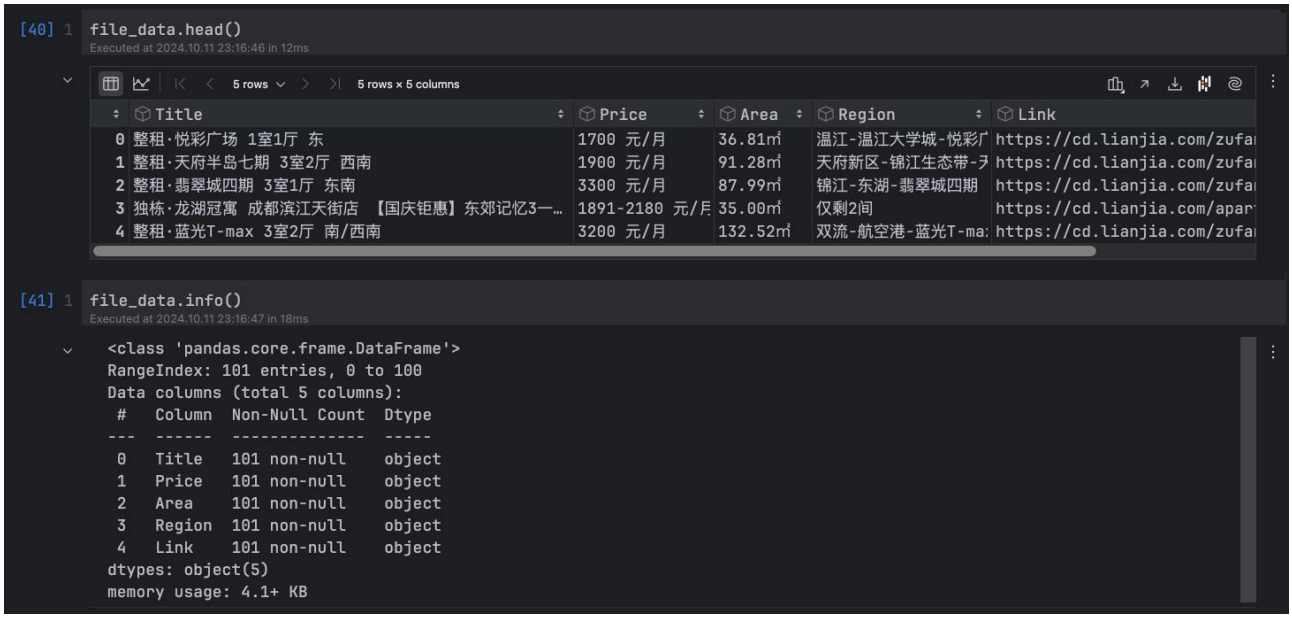
4.2.1 读取数据

- 使用 pandas 读取 CSV 文件中的数据

```
import pandas as pd

file_data=pd.read_csv('./chengdu_rent_1.csv')
Executed at 2024.10.11 23:16:46 in 18ms
```

- 查看数据文件中信息



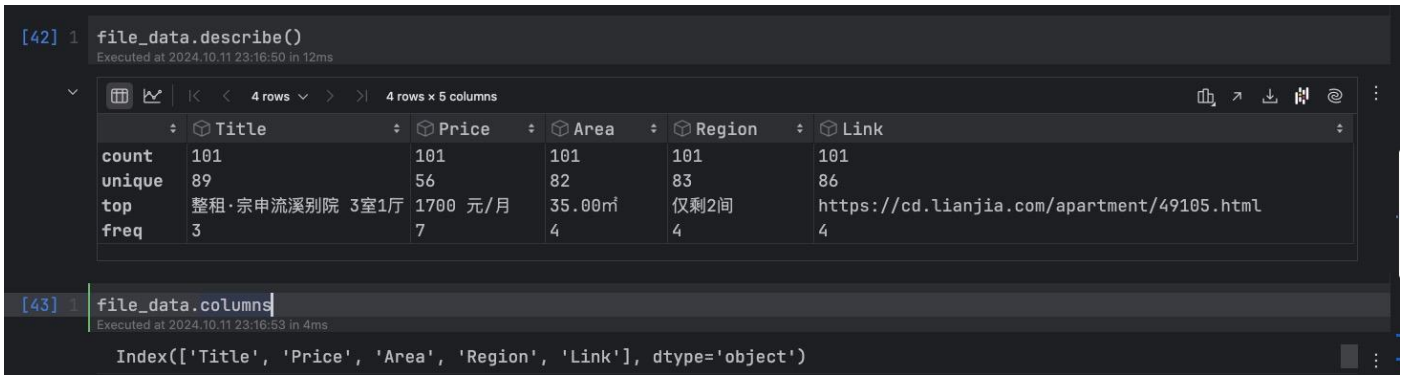
```
[40] 1 file_data.head()
Executed at 2024.10.11 23:16:46 in 12ms
```

5 rows 5 rows × 5 columns

	Title	Price	Area	Region	Link
0	整租·悦彩广场 1室1厅 东	1700 元/月	36.81㎡	温江-温江大学城-悦彩广	https://cd.lianjia.com/zufa
1	整租·天府半岛七期 3室2厅 西南	1900 元/月	91.28㎡	天府新区-锦江生态带-天	https://cd.lianjia.com/zufa
2	整租·翡翠城四期 3室1厅 东南	3300 元/月	87.99㎡	锦江-东湖-翡翠城四期	https://cd.lianjia.com/zufa
3	独栋·龙湖冠寓 成都滨江天街店 【国庆钜惠】东郊记忆3一…	1891-2180 元/月	35.00㎡	仅剩2间	https://cd.lianjia.com/apar
4	整租·蓝光T-max 3室2厅 南/西南	3200 元/月	132.52㎡	双流-航空港-蓝光T-max	https://cd.lianjia.com/zufa

```
[41] 1 file_data.info()
Executed at 2024.10.11 23:16:47 in 18ms
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Title   101 non-null    object
 1   Price   101 non-null    object
 2   Area    101 non-null    object
 3   Region  101 non-null    object
 4   Link    101 non-null    object
dtypes: object(5)
memory usage: 4.1+ KB
```



```
[42] 1 file_data.describe()
Executed at 2024.10.11 23:16:50 in 12ms
```

4 rows 4 rows × 5 columns

	Title	Price	Area	Region	Link
count	101	101	101	101	101
unique	89	56	82	83	86
top	整租·宗申流溪别院 3室1厅	1700 元/月	35.00㎡	仅剩2间	https://cd.lianjia.com/apartment/49105.html
freq	3	7	4	4	4

```
[43] 1 file_data.columns
Executed at 2024.10.11 23:16:53 in 4ms
```

```
Index(['Title', 'Price', 'Area', 'Region', 'Link'], dtype='object')
```

4.2.2 数据预处理

- 对数据集中的“价格”和“面积”列进行清洗和处理，以便后续分析。

```
[49] 1  # 清洗价格列
     2  def clean_price(price):
     3      # 移除"元/月"并去掉空格
     4      price = price.replace('元/月', '').replace(' ', '')
     5      # 如果价格是范围（如 '1891-2180'），取平均值
     6      if '-' in price:
     7          low, high = map(float, price.split('-'))
     8          return (low + high) / 2
     9      else:
    10          return float(price)
    11
    12  file_data['Price'] = file_data['Price'].astype(str).apply(clean_price)
    13
    14  # 清洗面积列
    15  file_data['Area'] = file_data['Area'].astype(str).str.replace('㎡', '').astype(float)
    16
    17  # 输出数据检查
    18  print(file_data.head())
    Executed at 2024.10.11 23:20:00 in 8ms
```

```
Executed at 2024.10.11 23:20:00 in 8ms

                                                Title   Price    Area  \
0                                  整租·悦彩广场 1室1厅 东  1700.0   36.81
1                                 整租·天府半岛七期 3室2厅 西南  1900.0   91.28
2                                  整租·翡翠城四期 3室1厅 东南  3300.0   87.99
3  独栋·龙湖冠寓 成都滨江天街店 【国庆钜惠】东郊记忆3一个月起租可月付短租 0中介今日可看房 开间  2035.5   35.00
4                                整租·蓝光T-max 3室2厅 南/西南  3200.0  132.52

                Region                                              Link
0      温江-温江大学城-悦彩广场  https://cd.lianjia.com/zufang/CD19559224339154...
1  天府新区-锦江生态带-天府半岛七期  https://cd.lianjia.com/zufang/CD19554644353634...
2        锦江-东湖-翡翠城四期  https://cd.lianjia.com/zufang/CD16993487308864...
3                  仅剩2间       https://cd.lianjia.com/apartment/49105.html
4     双流-航空港-蓝光T-max  https://cd.lianjia.com/zufang/CD19558896193125...
```

4.2.3 数据分析

1. 统计数据集中每个区域的租房信息数量

3.1 统计数据集中每个区域的租房信息数量

```
1  region_count = file_data['Region'].value_counts().reset_index()
2  region_count.columns = ['Region', 'Count']
3  print(region_count)
Executed at 2024.10.11 23:20:00 in 7ms
```

```
                Region  Count
0                 仅剩2间      4
1       温江-温江大学城-悦彩广场      3
2      双流-双流城区-宗申流溪别院      3
3         成华-万象城-千居朝阳      2
4         高新-远大-嘉年华青年城      2
..                   ...    ...
78        武侯-红牌楼-双楠悦天地      1
79     郫都-橡树湾-华润橡树湾A区      1
80        高新-中和-军安卫士花园      1
81          锦江-东湖-中港广场      1
82  锦江-川师-邮电职业技术学院宿舍      1

[83 rows x 2 columns]
```

2. 计算数据集中每个区域的平均租金

3.2 计算数据集中每个区域的平均租金

```
average_rent = file_data.groupby('Region')['Price'].mean().reset_index()
average_rent.columns = ['Region', 'Average_Rent']
print(average_rent)
```

Executed at 2024.10.11 23:20:00 in 6ms

```
                Region  Average_Rent
0                 仅剩2间      1974.875
1                 仅剩3间      2090.500
2       双流-九龙湖-棠湖华府公园      2000.000
3        双流-公兴-荷韵欣苑西苑      1420.000
4        双流-华府-洲际华府广场      1900.000
..                   ...          ...
78  龙泉驿-东山-东山国际新城H区二期      1800.000
79       龙泉驿-大面-世茂城一期      2200.000
80       龙泉驿-大面-炜岸城一期      2100.000
81       龙泉驿-大面-首创万卷山      1890.000
82    龙泉驿-西河-银诚东方国际一期      1600.000

[83 rows x 2 columns]
```

3. 计算数据集中每个区域的平均租金，将结果存储在一个新的 DataFrame 中，并打印出来

3.3 计算数据集中每个区域的平均租金，将结果存储在一个新的DataFrame中，并打印出来

```
[63] area_bins = [0, 30, 50, 70, 90, 120, 150, 200]
     area_labels = ['0-30', '30-50', '50-70', '70-90', '90-120', '120-150', '150+']
     file_data['Area_Category'] = pd.cut(file_data['Area'], bins=area_bins, labels=area_labels)
     area_distribution = file_data['Area_Category'].value_counts().reset_index()
     area_distribution.columns = ['Area_Category', 'Count']
     print(area_distribution)
Executed at 2024.10.11 23:30:24 in 13ms
```

	Area_Category	Count
0	30-50	29
1	90-120	20
2	70-90	15
3	120-150	13
4	0-30	12
5	50-70	11
6	150+	1

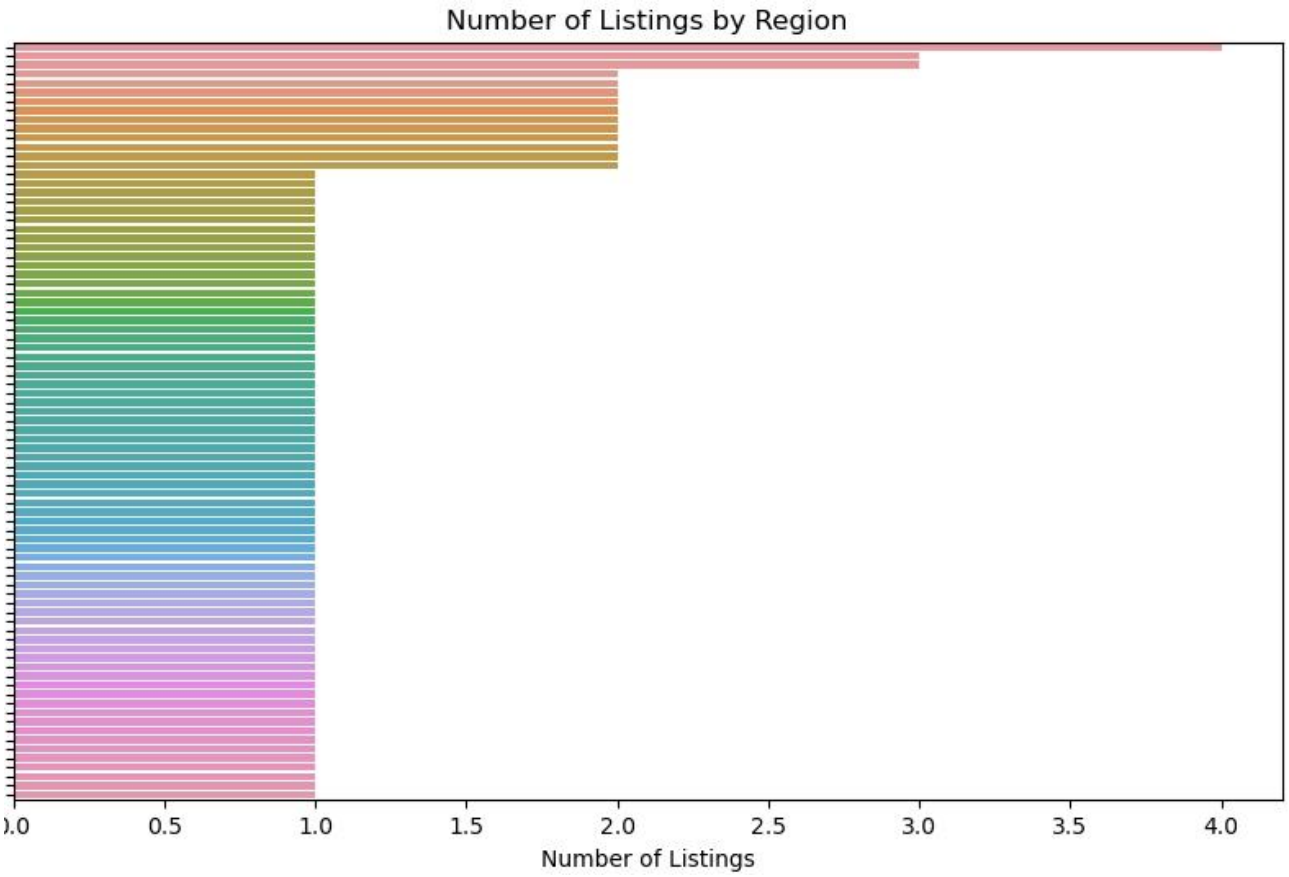
4. 图表展示不同区域的房源数量

3.4 图表展示不同区域的房源数量

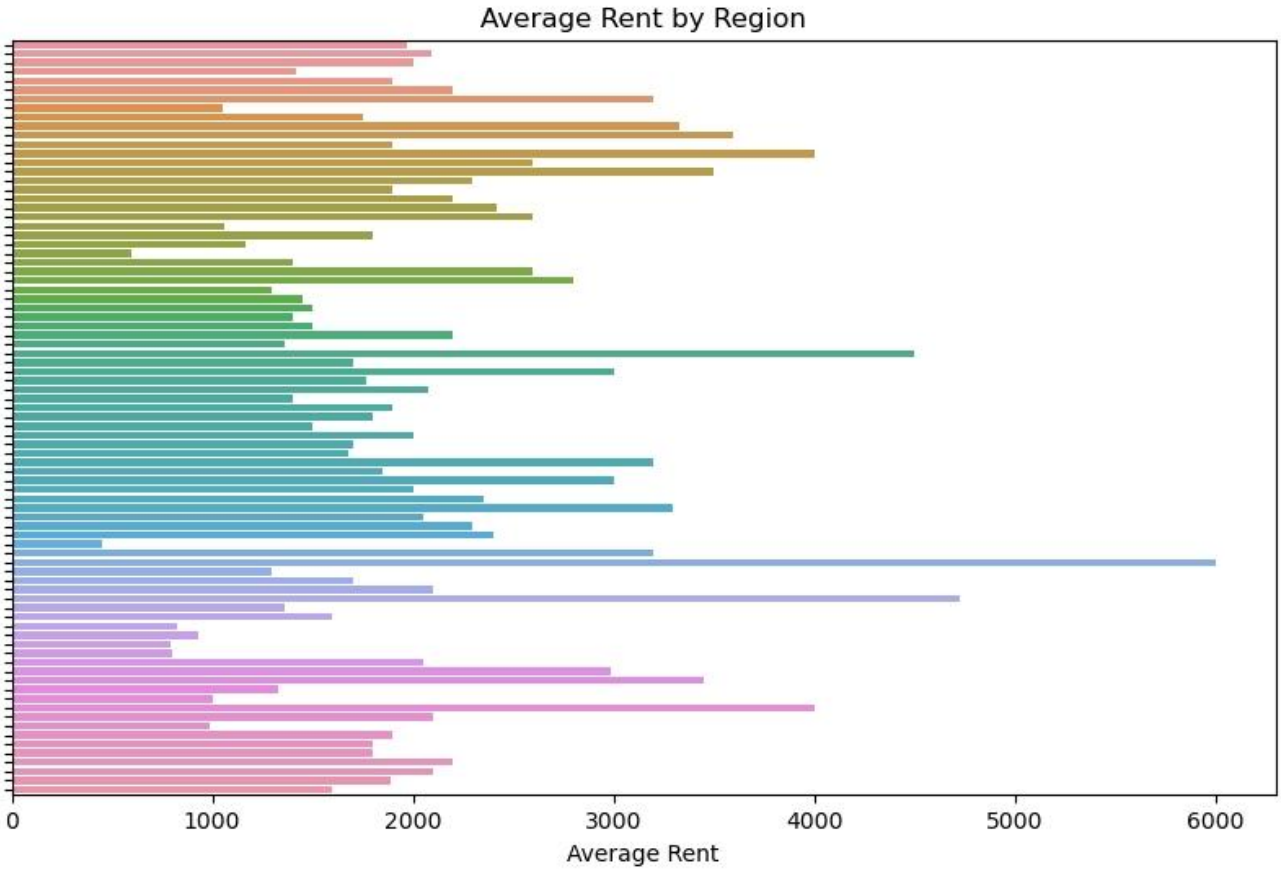
```
[64] import seaborn as sns
     import matplotlib.pyplot as plt

     # 房源数量按区域
     plt.figure(figsize=(10, 6))
     sns.barplot(x='Count', y='Region', data=region_count)
     plt.title('Number of Listings by Region')
     plt.xlabel('Number of Listings')
     plt.ylabel('Region')
     plt.show()

Executed at 2024.10.11 23:30:24 in 412ms
```



5. 条形图展示不同区域的平均租金



3) 实验总结

4.3.1 步骤总结

1. 爬取数据:使用 selenium 库直接从链家平台上爬取信息,存储在 CSV 文件中
2. 数据分析与可视化

a) 读取数据:使用 pandas 库

b) 数据预处理,清洗

c) 使用 matplotlib seaborn 库实现数据可视化分析

4.3.2 遇到的困难

1. 在爬取数据的过程中,如果在短时间内多次访问服务器的话,会被拦截,导致爬虫失败

2. matplotlib seaborn 库在 macOS 系统上的中文字体显示出了问题,由于近期才换了 macOS,还在配置环境和习惯中,所以这方面花费了一些时间

3. 实验过程中python解释器出现了问题,搜了很多教程也没得到解决,最后通过重启 Pycharm 问题自然就解决了.

4. word 书写实验报告还是感觉不是很高效,可能是我使用 word 不太熟练,感觉使用 Markdown 或 LaTeX 更加高效,排版更加美观.

5. Hw1 所有代码和相关文件全部上传到 Github 仓库 https://github.com/KaiHaverz/dataScience_learningRecord