爬取租房数据并可视化分析

以北京租房数据统计分析作为案例，演示一个基本的完整数据分析过程：

- `Pandas` 的读写操作
- 使用预处理技术过滤数据。
- 使用 `Matplotlib` 库绘制各种图表。
- 基于数据进行分析。

近年来随着经济的快速发展，一线城市的资源和就业机会吸引了很多外来人口，使其逐渐成为人口密集的城市之一。绝大多数人是以租房的形式解决居住问题。

本文将租房网站上北京地区的租房数据作为参考，运用所学到的数据分析知识，带领大家一起来分析真实数据，并以图表的形式得到以下统计指标：

- (1) **统计每个区域的房源总数量，并使用热力图分析房源位置分布情况。**
- (2) **使用条形图分析哪种户型的数量最多、更受欢迎。**
- (3) **统计每个区域的平均租金，并结合柱状图和折线图分析各区域的房源数量和租金情况。**
- (4) **统计面积区间的市场占有率，并使用饼图绘制各区间所占的比例。**

1. 数据基本介绍

目前网络上有很多的租房平台，比如自如、爱屋吉屋、房天下、链家等，其中，链家是目前市场占有率最高的公司，通过链家平台可以便捷且全面地提供可靠的房源信息。如下图所示:

通过网络爬虫技术，爬取链家网站中列出的租房信息，具体包括所属区域、小区名称、房屋、价格、房屋面积、户型。由于链家的网站数据没有做太多的反爬，所以具体数据采集的过程就不赘述了（提示：浅尝辄止，避免对服务器造成干扰）。将爬到的数据下载到本地，并保存在“链家北京租房数据.csv”文件中，打开该文件后可以看到里面有很多条（本案例爬取的数据共计 8224 条)信息，具体如下图所示。

	A	B	C	D	E	F
1	区域	小区名称	户型	面积(㎡)	价格(元/月)	
2	东城	万国城MOMA	1室0厅	59.11平米	10000	
3	东城	北官厅胡同2号院	3室0厅	56.92平米	6000	
4	东城	和平里三区	1室1厅	40.57平米	6900	
5	东城	菊儿胡同	2室1厅	57.09平米	8000	
6	东城	交道口北二条35号院	1室1厅	42.67平米	5500	
7	东城	西营房	2室1厅	54.48平米	7200	
8	东城	地坛北门	1室1厅	33.76平米	6000	
9	东城	安外东河沿	1室1厅	37.62平米	5600	
10	东城	清水苑	1室1厅	45.61平米	6200	
11	东城	李村东里	2室1厅	57.35平米	5700	
12	东城	幸福北里	2室1厅	51.15平米	6500	
13	东城	保利蔷薇	2室1厅	97.11平米	10000	
14	东城	东板桥西巷	2室1厅	52.86平米	5800	
15	东城	本家润园三期	2室1厅	63.09平米	7800	
16	东城	营房西街	2室1厅	62.95平米	7500	
17	东城	新景家园西区	1室1厅	57.24平米	7500	
18	东城	东花市北里东区	2室1厅	85.36平米	8800	
19	东城	幸福家园一期	5室2厅	226.86平米	29000	
20	东城	景泰西里西区	1室1厅	60.3平米	6200	
21	东城	海晟名苑北区	1室1厅	70.86平米	12000	
22	东城	和平新城一期	2室1厅	122.76平米	14500	
23	东城	太华公寓	2室2厅	152.24平米	17000	
24	东城	官书院	2室1厅	92.01平米	16000	
25	东城	幸福家园二期	2室1厅	65.25平米	7800	
26	东城	安外大街3号院	1室1厅	33.77平米	5500	
27	东城	中海紫御公馆	2室2厅	90.15平米	13000	
28	东城	海晟名苑北区	1室0厅	45.62平米	9000	
29	东城	凯景铭座	3室1厅	156.2平米	16000	
30	东城	永定门东街西里	2室1厅	53.26平米	5000	

2. 数据读取

准备好数据后，我们便可以使用 Pandas 读取保存在 CSV 文件的数据，并将其转换成 DataFrame 对象展示，便于后续操作这些数据。

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号
```

首先，读取数据：

```
import pandas as pd
import numpy as np

# 读取链家北京租房信息
file_data = pd.read_csv('./data/链家北京租房数据.csv')
```

```
# 完整数据获取：关注@公众号：数据STUDIO
# 后台联系云朵君获取~
file_data.head()
```

读取效果如下：

	区域	小区名称	户型	面积(㎡)	价格(元/月)
0	东城	万国城MOMA	1室0厅	59.11平米	10000
1	东城	北官厅胡同2号院	3室0厅	56.92平米	6000
2	东城	和平里三区	1室1厅	40.57平米	6900
3	东城	菊儿胡同	2室1厅	57.09平米	8000
4	东城	交道口北二条35号院	1室1厅	42.67平米	5500
...	...	...	...	...	...
8218	顺义	怡馨家园	3室1厅	114.03平米	5500
8219	顺义	旭辉26街区	4房间2卫	59平米	5000
8220	顺义	前进花园玉兰苑	3室1厅	92.41平米	5800
8221	顺义	双裕小区	2室1厅	71.81平米	4200
8222	顺义	樱花园二区	1室1厅	35.43平米	2700

8223 rows × 5 columns

3. 数据预处理

尽管从链家官网上直接爬取下来的数据大部分是比较规整的，但或多或少还是会存在一些问题，不能直接用做数据分析。为此，在使用前需要对这些数据进行一系列的检测与处理，包括处理重复值和缺失值、统一数据类型等，以保证数据具有更高的可用性。

3.1 重复值和空值处理

预处理的前两步就是检查缺失值和重复值。如果希望检查准备的数据中是否存在重复的数据，则可以通过 Pandas 中的 duplicated()方法完成。接下来，通过 `duplicated()`方法对北京租房数据进行检测，只要有重复的数据就会映射为 True，具体代码如下。

```
# 重复数据检测
file_data.duplicated()
```

由于数据量相对较多，所以在 Jupyter NoteBook 工具中有一部分数据会省略显示，但是从输出结果中仍然可以看到有多条返回结果为 True 的数据，这表明有重复的数据。这里，处理重复数据的方式是将其删除。接下来，使用 `drop_duplicates()`方法直接删除重复的数据，具体代码如下。

```
# 删除重复数据
file_data = file_data.drop_duplicates()
```

与上一次输出的行数相比，可以很明显地看到减少了很多条数据，只剩下了 5773 条数据。

对数据重复检测完成之后，便可以检测数据中是否存在缺失值，我们可以直接使用 dropna()方法检测并删除缺失的数据，具体代码如下。

```
# 删除缺失数据
file_data = file_data.dropna()
```

经过缺失数据检测之后，可以发现当前数据的总行数与之前相比没有发生任何变化。因此我们断定准备好的数据中并不存在缺失的数据。

3.2 数据转换类型

在这套租房数据中，“面积(㎡)”一列的数据里面有中文字符，说明这一列数据都是字符串类型的。为了方便后续对面积数据进行数学运算，所以需要将“面积(m)”一列的数据类型转换为 float 类型，具体代码如下。

```
# 创建一个空数组
data_new = np.array([])
# 取出“面积”一列数据，将每个数据末尾的中文字符去除  fild_data.info()
data = file_data['面积(㎡)'].values
for i in data:
                data_new = np.append(data_new, np.array(i[:-2]))
# 通过 astype()方法将 str 类型转换为 float64 类型
data = data_new.astype(np.float64)
# 用新的数据替换
file_data.loc[:,'面积(㎡)']= data
```

除此之外，在“户型”一列中，大部分数据显示的是“室*厅*”，只有个别数据显示的是"*房间*卫”(比如索引 8219 对应的一行)。为了方便后期的使用，需要将“房间"替换成"室"，以保证数据的一致性。

接下来，使用 Pandas 的 replace() 方法完成替换数据的操作，具体代码如下。

```
# 获取“户型”一列数据
housetype_data = file_data['户型']
temp_list = []
# 通过 replace()方法进行替换
for i in housetype_data:
    new_info = i.replace('房间','室')
    temp_list.append(new_info)
file_data.loc[:,'户型'] = temp_list
```

通过比较处理前与处理后的数据可以发现，索引为 8219 的户型数据已经由“4 房间 2 卫”变成“4 室 2 卫”，说明数据替换成功。

4. 图表分析

数据经过预处理以后，便可以用它们来做分析了，为了能够更加直观地看到数据的变化，这里，我们采用图表的方式来辅助分析。

4.1 房源数量、位置分布分析

如果希望统计各个区域的房源数量，以及查看这些房屋的分布情况，则需要先获取各个区的房源。为了实现这个需求，可以将整个数据按照“区域”一列进行分组。

为了能够准确地看到各区域的房源数量，这里只需要展示“区域”与“数量”这两列的数据即可。因此，先创建一个空的 DataFrame 对象，然后再将各个区域计算的总数量作为该对象的数据进行展示，具体代码如下。

```
# 创建一个 DataFrame 对象，该对象只有两列数据：区域和数量
# 完整数据获取：关注@公众号：数据 STUDIO
# 后台联系云朵君获取~
new_df = pd.DataFrame({'区域':file_data['区域'].unique(),'数量':[0]*13})
```

接下来，通过 Pandas 的 `groupby()`方法将 file data 对象按照“区域”一列进行分组，并利用 `count()`方法统计每个分组的数量，具体代码如下。

```
# 按“区域”列将 file_data 进行分组，并统计每个分组的数量
groupy_area = file_data.groupby(by='区域').count()
new_df['数量'] = groupy_area.values
```

通过 `sort_values()`方法对 new_df 对象排序，按照从大到小的顺序进行排列，具体代码如下。

```
# 按“数量”一列从大到小排列
new_df.sort_values(by=['数量'], ascending=False)
```

通过输出的排序结果可以看出，房源数量位于前的区域分别是朝阳区、海淀区、丰台区。

4.2 户型数量分析

随着人们生活水平的提高，以及各住户的生活需求，开发商设计出了各种各样的户型供人们居住。接下来，我们来分析一下户型，统计租房市场中哪种户型的房源数量偏多，并筛选出数量大于 50 的户型。

首先，我们定义一个函数来计算各种户型的数量，具体代码如下。

```
# 定义函数，用于计算各户型的数量
def all_house(arr):
    key = np.unique(arr)
    result = {}
    for k in key:
        mask = (arr == k)
        arr_new = arr[mask]
        v = arr_new.size
        result[k] = v
    return result

# 获取户型数据
house_array = file_data['户型']
house_info = all_house(house_array)
```

程序输出了一个字典，其中，字典的键表示户型的种类，值表示该户型的数量。

使用字典推导式将户型数量大于 50 的元素筛选出来，并将筛选后的结果转换成 DataFrame 对象，具体代码如下。

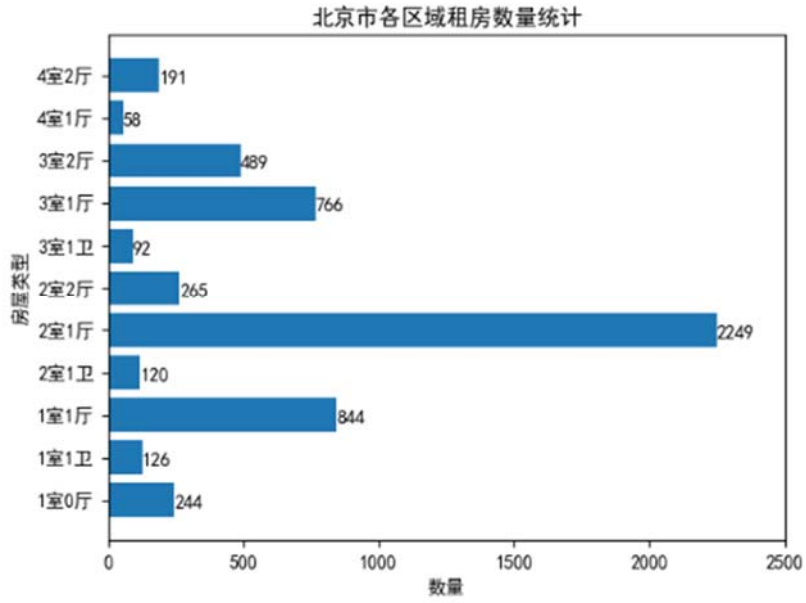
```
# 使用字典推导式
house_type = dict((key, value) for key, value
in house_info.items() if value > 50)
show_houses = pd.DataFrame({'户型':[x for x in  house_type.keys()],'数量
':[x for x in house_type.values()]})
```

为了能够更直观地看到户型数量间的差异，我们可以使用条形图进行展示，其中，条形图纵轴坐标代表户型种类，横坐标代表数量体代码如下

```
import matplotlib.pyplot as plt

house_type = show_houses['户型']
house_type_num = show_houses['数量']
plt.barh(range(11), house_type_num, height=0.7, color='steelblue', alph
a=0.8)
plt.yticks(range(11), house_type)
plt.xlim(0,2500)  # 把x轴坐标延长到2500
```

```
plt.xlabel("数量")
plt.ylabel("户型种类")
plt.title("北京地区各户型房屋数量")
for x, y in enumerate(house_type_num):
    plt.text(y + 0.2, x - 0.1, '%s' % y)
plt.show()
```

运行结果如下图所示。



通过图可上以清晰地看出，整个租房市场中户型数量较多分别为“2 室 1 厅”、“1 室 1 厅”、“3 室 1 厅”的房屋，其中，“2 室 1 厅”户型的房屋在整个租房市场中是数量最多的。

4.3 平均租金分析

为了进一步剖析房屋的情况，接下来，我们来分析一下各地区目前的平均租金情况。计算各区域房租的平均价格与计算各区域户型数量的方法大同小异，首先创建一个 DataFrame 对象，具体代码如下。

```
# 新建一个DataFrame对象，设置房租总金额和总面积初始值为0
df_all = pd.DataFrame({'区域':file_data['区域'].unique(),'房租总金额
':[0]*13,'总面积(㎡)':[0]*13})
```

接下来，按照“区域”一列进行分组，然后调用 sum()方法分别对房租金额和房屋面积执行求和计算，具体代码如下:

```
# 求总金额和总面积
sum_price = file_data['价格(元/月)'].groupby(file_data['区域']).sum()
```

```
sum_area = file_data['面积(㎡)'].groupby(file_data['区域']).sum()
df_all['房租总金额'] = sum_price.values
df_all['总面积(㎡)'] = sum_area.values
```

计算出各区域房租总金额和总面积之后，便可以对每平方米的租金进行计算。在 df_all 对象的基础上增加一列，该列的名称为“每平方米租金(元)”，数据为求得的每平方米的平均价格，具体代码如下。

```
# 计算各区域每平米房租价格,并保留两位小数
df_all['每平米租金(元)'] = round(df_all['房租总金额'] / df_all ['总面积(㎡)'], 2)
```

为了能更加全面地了解到各个区域的租房数量与平均租金，我们可以将之前创建的 new_df 对象(各区域房源数量)与 df_all 对象进行合并展示，由于这两个对象中都包含“区域”一列，所以这里可以采用主键的方式进行合并，也就是说通过 merge()函数来实现，具体代码如下。

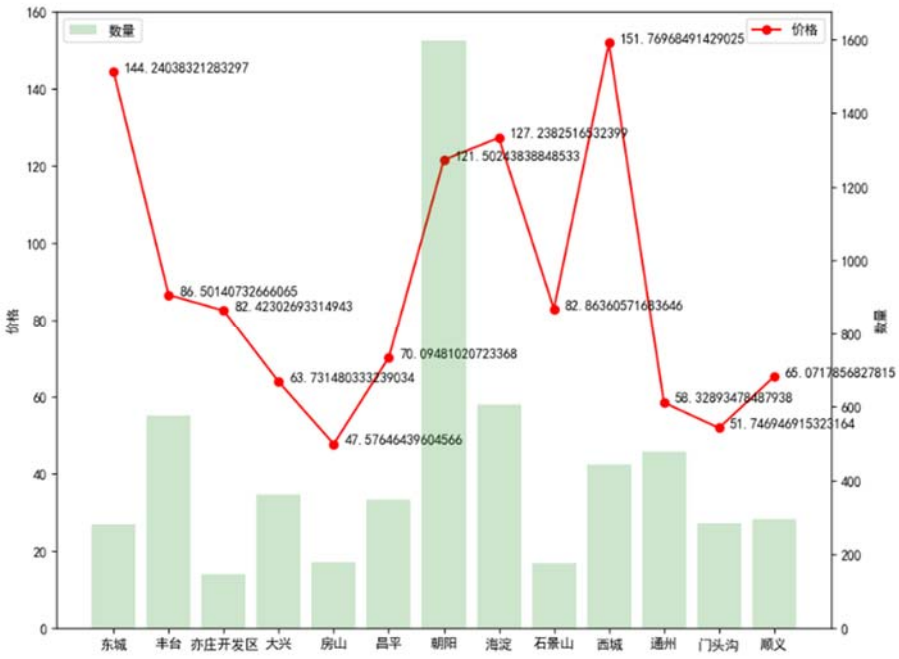
```
# 合并new_df 与df_all
df_merge = pd.merge(new_df, df_all)
```

合并完数据以后，就可以借用图表来展示各地区房屋的信息，其中，房源的数量可以用柱状图中的条柱表示，每平方米租金可以用折线图中的点表示，具体代码如下。

```
num= df_merge['数量']    # 数量
price=df_merge['每平米租金(元)'] # 价格
l=[i for i in range(13)]

lx=df_merge['区域']
fig = plt.figure(figsize=(10, 8), dpi=100)

# 显示折线图
ax1 = fig.add_subplot(111)
ax1.plot(l, price,'or-',label='价格')  # "or-" 显示那个小红圆点
for i,(_x,_y) in enumerate(zip(l,price)):
    plt.text(_x,_y,price[i])
ax1.set_ylim([0, 200])
ax1.set_ylabel('价格')
plt.legend(loc='upper left')

# 显示条形图
ax2 = ax1.twinx()  # 显示次坐标轴 ax2=ax1.twinx()
plt.bar(l,num,alpha=0.3,color='green',label='数量')
ax2.set_ylabel('数量')
plt.legend(loc="upper right")
plt.xticks(l,lx)
```

```
plt.show()
```

运行结果如下：



从图中可以看出，西城区、东城区、海淀区、朝阳区的房租价格相对较高，这主要是因为东城区和西城区作为北京市的中心区，租金相比其他几个区域自然偏高一些，而海淀区租金较高的原因推测可能是海淀区名校较多，也是学区房最火热的地带，朝阳区内的中央商务区聚集了大量的世界 500 强公司，因此这四个区域的房租相对其他区域较高。

4.4 面积区间分析

下面我们将房屋的面积数据按照一定的规则划分成多个区间，看一下各面积区间的上情况，便于分析租房市场中哪种房屋类型更好出租，哪个面积区间的相房人数最多

要想将数据划分为若干个区间，则可以使用 Pame 中的 `cut()`函数来实现，首先，使用 `max()`与 `min()`方法分别计算出房屋面积的最大值和最小值，具体代码如下。

```
# 查看房屋的最大面积和最小面积
print('房屋最大面积是%d 平米'%(file_data['面积(㎡)'].max()))
print('房屋最小面积是%d 平米'%(file_data['面积(㎡)'].min()))

# 查看房租的最高值和最小值
```

```
print('房租最高价格为每月%d 元'%(file_data['价格(元/月)'].max()))
print('房屋最低价格为每月%d 元'%(file_data['价格(元/月)'].min()))
```

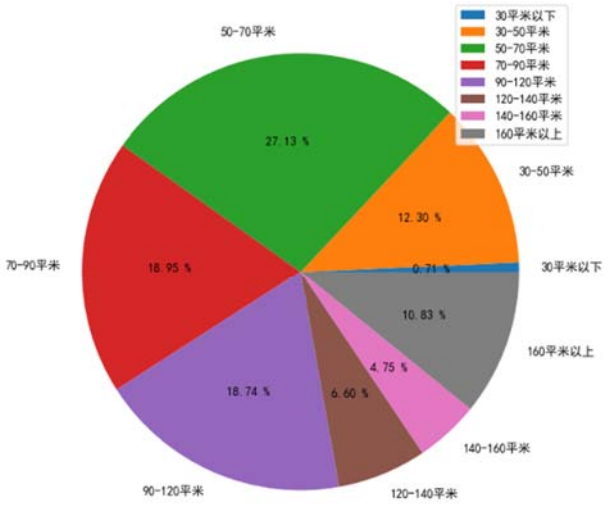
在这里，我们参照链家网站的面积区间来定义，将房屋面积划分为 8 个区间。然后使用 describe()方法显示各个区间出现的次数(counts 表示)以及频率(freps 表示)，具体代码如下。

```
# 面积划分
area_divide = [1, 30, 50, 70, 90, 120, 140, 160, 1200]
area_cut = pd.cut(list(file_data['面积(㎡)']), area_divide)
area_cut_data = area_cut.describe()
```

接着，使用饼图来展示各面积区间的分布情况，具体代码如下。

```
area_percentage = (area_cut_data['freqs'].values)*100

labels  = ['30 平米以下', '30-50 平米', '50-70 平米', '70-90 平米',
'90-120 平米','120-140 平米','140-160 平米','160 平米以上']

plt.figure(figsize=(20, 8), dpi=100)
plt.axes(aspect=1)  # 显示的是圆形,如果不加,是椭圆形
plt.pie(x=area_percentage, labels=labels, autopct='%.2f %%', shadow=Tru
e)
plt.legend(loc='upper right')
plt.show()
```

运行结果如图所示：



通过上图可以看出，50-70 平方米的房屋在租房市场中占有率最大。总体看来，租户主要以 120 平方米以下的房屋为租住对象，其中 50~70 平方米以下的房屋为租户的首选对象。

总结

通过对于北京地区的二手房信息的数据进行分析，中间数据预处理的过程尤为重要，包括但不限于数据清洗等操作，除此之外还使用到大量 pandas 内置函数，对于数据进行分组聚合达到想要的效果，从而进行数据分析以及可视化，感兴趣的同学可以尝试下。