数据可视化实验报告

计算机科学与技术学院

班　　级： _____

学　　号： _____

姓　　名： _____

指导教师： 何云峰

完成日期： 2024-01-10

实验报告及设计评分细则

评 分 项 目	满分	得分	备注	
文档格式（段落、行间距、缩进、图表、编号等）	15			实 验 报 告 总分
实验方案设计	10			
实验过程	50			
遇到的问题及处理	10			
设计方案存在的不足	5			
心得（含思政）	5			
意见和建议	5			
可视化作品	100			
教师签名			日　　期	

备注：实验过程将从可视化作品的完成度、是否讲述好数据的故事、作品的复杂度等方面进行综合评分。

实验课程总分=可视化*0.6+实验报告*0.4

目　录

1　实验概述 4

1.1　实验名称 4
1.2　实验目的 4
1.3　实验环境 4
1.4　实验内容 4
1.5　实验要求 4

2　总体设计 6

2.1　总体设计思路 6
2.2　总体设计框架 7

3　实验过程 8

3.1　数据预处理 8
3.2　总价视图 9
3.3　二手房数量视图 10
3.4　房屋类型视图 16
3.5　介绍词云视图 20
3.6　地理分布视图 21
3.7　建筑面积与套内面积拟合视图 23

4　设计总结与心得 25

4.1　实验总结 25
4.2　实验心得 25
4.3　意见与建议 26

1 实验概述

1.1 实验名称

设计一个数据可视化作品，讲述数据的故事。

1.2 实验目的

（1）了解数据可视化方案的设计和实现方法；

（2）利用 python 的数据可视化工具库（matplotlib 库、pyecharts 库）进行可视化方案的实现。

1.3 实验环境

软件：python。

可用库：数据处理库 NumPy、pandas；可视化库 matplotlib 库、pyecharts 库等。

可视化视图：pyecharts 中提供了 Bar(柱状图/条形图)；Bar3D(3D 柱状图)；Boxplot(箱形图)；EffectScatter(涟漪散点图)；Funnel (漏斗图)；Gauge(仪表盘)；Geo(地理坐标系)；Graph(关系图)；HeatMap(热力图)；Kline(K 线图)；Line(折线/面积图)；Line3D(3D 折线图)；Liquid(水球图)；Map(地图)；Parallel (平行坐标系；Pie(饼图)；Polar(极坐标系)；Radar(雷达图)；Sankey(桑基图)；Scatter(散点图等，可以根据需要进行选择。

1.4 实验内容

（1）在备选的 4 个数据集中任意选择其中一个数据集，对数据集中的数据进行数据分析和预处理，设计合理的可视化方案。

（2）利用 python 的可视化工具库实现可视化方案。

（3）完成实验报告，讲述数据的故事。

1.5 实验要求

（1）对数据集中的数据进行分析和整理，提取出其中的有效信息，完成数据的预处理；

（2）设计合理的可视化视图，选择合适的视觉通道，表达数据的某个特性；

（3）完成不少于 6 个可视化视图，较为完整地描述数据集。

2 总体设计

2.1 总体设计思路

数据集选择和说明：

选择的数据集是杭州二手房价数据集。

数据集包含十个文件，每个文件对应一个区域，每个文件包含约 3000 条数据。

数据集的字段包括小区名称、区域位置、经度、纬度、总价、单价、看房时间、链家编号、关注度、房屋户型、所在楼层、建筑面积、户型结构、套内面积、建筑类型、房屋朝向、建筑结构、装修情况、梯户比例、配备电梯、挂牌时间、交易权属、上次交易、房屋用途、房屋年限、产权所属、抵押信息、房本备件、房源核验统一编码、查询房管备案记录、核心卖点、小区介绍、周边配套、税费解析、用水类型、用电类型、燃气价格、户型介绍、适宜人群、装修描述、售房详情、交通出行、别墅类型、权属抵押。

总体设计思路：

数据预处理：对数据集进行必要的预处理，包括数据清洗、缺失值处理、异常值处理等。

数据分析和可视化：通过统计分析和可视化方法对数据集进行探索，了解数据的分布、特征和相关性。

结果分析和总结：对实验结果进行分析和总结，讨论视图的优缺点以及改进方向。

数据预处理：

数据清洗：检查数据中的重复值、异常值和错误值，并进行相应处理。

缺失值处理：对缺失值进行填充或删除，选择合适的方法保证数据的完整性和准确性。

数据转换：将数据中的非数值型特征进行编码或转换为数值型特征，以便于后续的分析和建模过程。

数据分析和可视化：

统计分析：计算数据集的基本统计特征，如均值、中位数、标准差等，以了解数据的整体情况。

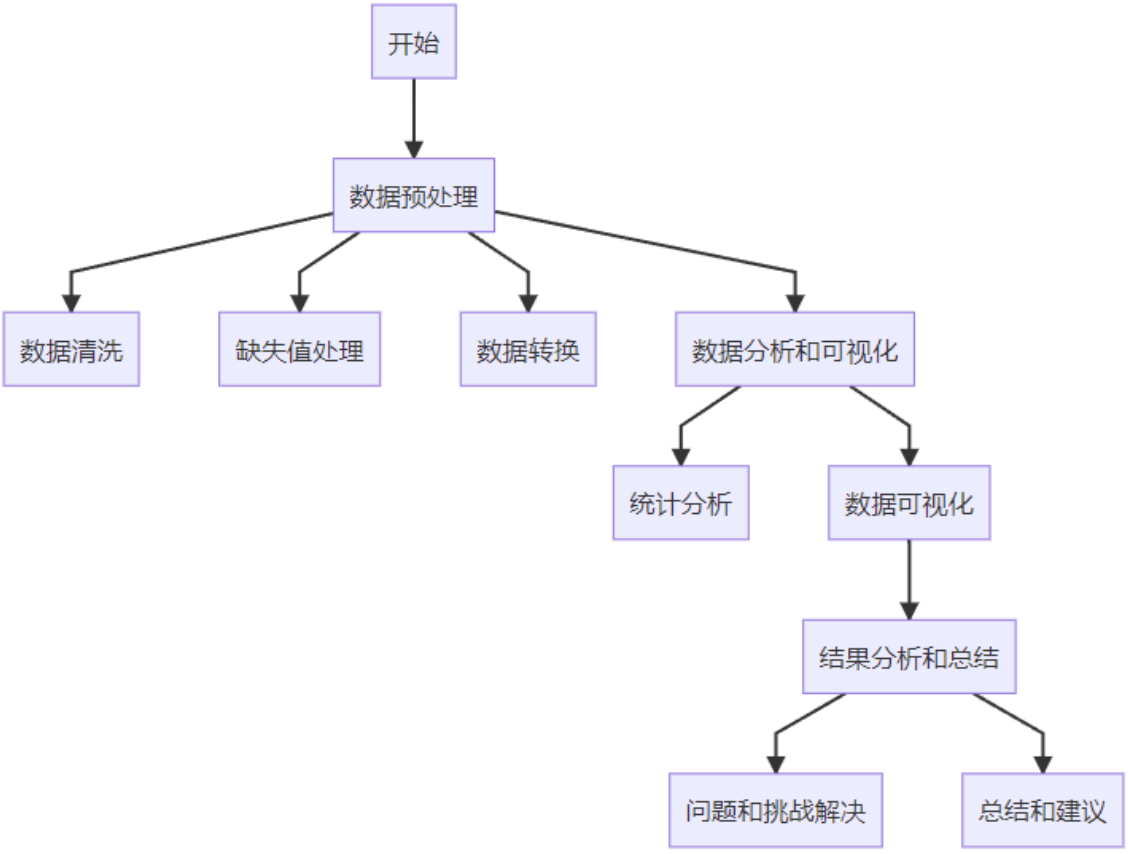
数据可视化：使用适当的图表和图像展示数据的分布、关系和趋势，如直方

图、散点图、箱线图等。

结果分析和总结：

对实验结果进行分析和总结，讨论可视化的优点、局限性以及改进方向。
针对实验中遇到的问题和挑战，提出解决方案和改进策略。
总结实验的经验教训，对未来类似问题的研究提出建议。

2.2 总体设计框架



3 实验过程

3.1 数据预处理

（1） 数据预处理设计

数据预处理设计包括以下步骤：

1. 合并数据：将十个文件的数据进行合并。合并是基于列名进行的，这意味着具有相同列名的列将被合并在一起，形成一个新的数据集。合并过程中使用常见的数据处理工具或编程语言（如 Python 中的 pandas 库）来执行此操作。

2. 文件名映射：将每个文件的文件名映射为一个新的字段，并将该字段的值赋予该文件内的所有数据。这将帮助我在合并后的数据集中区分每个文件的数据来源，以便后续分析和识别。

3. 缺失数据处理：当某个数据没有分析所需的字段时，我选择将该数据暂时剔除。这意味着如果某行数据在分析所需的字段上存在缺失值，该行数据将被从数据集中移除。这种处理方法可以确保在进行分析时只使用完整的数据。

4. 输出合并后的数据集：最后，我将输出一个合并后的数据集，该数据集包含了十个文件的数据，并且每个文件名都映射到其所包含的数据的地区字段。这个合并后的数据集可以用于后续的数据分析和建模。

（2）数据预处理的实现

```
import os
import pandas as pd

# 获取 data 文件夹下所有 CSV 文件的文件名
folder_path = 'data/'
csv_files = [file for file in os.listdir(folder_path) if
file.endswith('.csv')]

# 创建一个空的 DataFrame 用于存储合并的结果
merged_data = pd.DataFrame()

# 遍历每个 CSV 文件并进行合并
for file in csv_files:
    # 读取 CSV 文件
    file_path = os.path.join(folder_path, file)
    data = pd.read_csv(file_path,encoding='GBK')

    # 添加文件名列
```

```
    print(file)
    data['区域'] = file[0:-9]

    # 将当前 CSV 文件的数据合并到结果 DataFrame 中
    merged_data = pd.concat([merged_data, data], ignore_index=True)

# 将合并结果保存为新的 CSV 文件
merged_data.to_csv('merged_data.csv', index=False,encoding='utf-8')
```

3.2 总价视图

（1） 设计思路及设计过程

中位数价格法比算术平均法更能反映房地产市场的真实情况，因为中位数较平均数的优点是其不易受到极大数和极小数的影响，反映了一组数据的平均水平，因此这里使用中位数来表示房价水平。总价视图的设计思路是通过条形图展示不同区域和位置的房价中位数，以便用户可以比较不同地区和位置的房价水平。

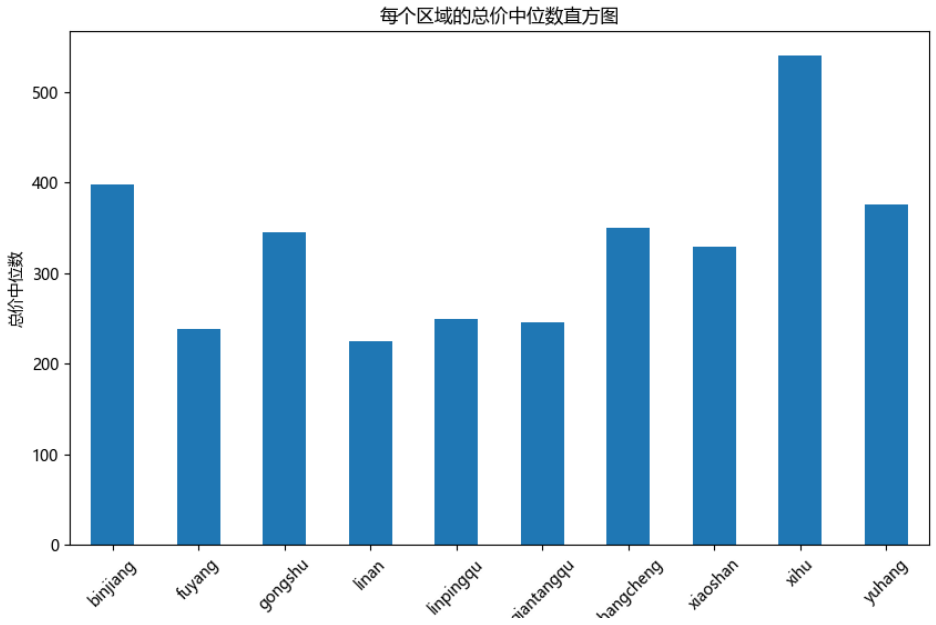


图 3-1 区域总价条形图

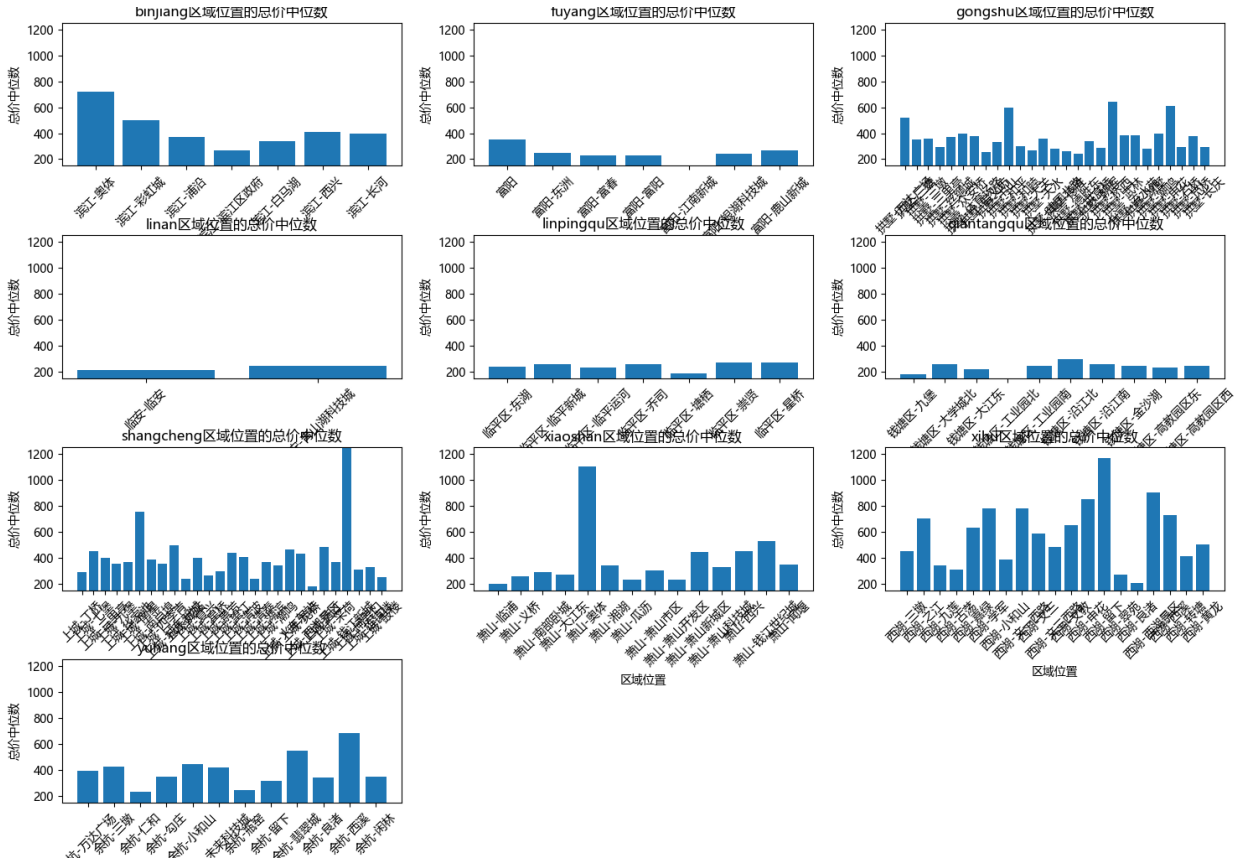


图 3-2 区域位置总价

（2）视图分析

（举例说明可视化视图给用户展示了什么信息。）

用户可以看到柱状图中每个区域或位置的条形高度，较高的条形表示该区域或位置的房价中位数较高，而较低的条形表示房价中位数较低。

用户可以通过比较不同条形的高度，确定哪个区域或位置的房价更贵或更便宜。从图中可以看出，房价的水平在地区之间大致分为三个梯度，西湖属于第一梯度，滨江、拱墅、杭城、萧山和余杭属于第二梯度，其他地区属于第三梯度。

同时，根据区域位置总价中每个子图，可以观察到区域内的发展是否平衡，例如西湖总体房价较高，但仍有翠苑和良渚两个地方的房价处于较低水平，而萧山虽然总体房价不高，但是萧山-奥体的房价却非常高，遥遥领先于该地区的其他地方。

此外，用户还可以通过观察不同区域或位置的条形在图表上的相对位置，了解房价差异是否与地理位置有关。

3.3 二手房数量视图

（1） 设计思路及设计过程

二手房数量视图的设计思路是通过饼图展示每个地区的二手房数量比例，并

在每个地区内再生成饼图展示各个地方的二手房数量比例，以便用户可以直观地了解不同地区和地方的房屋分布情况。

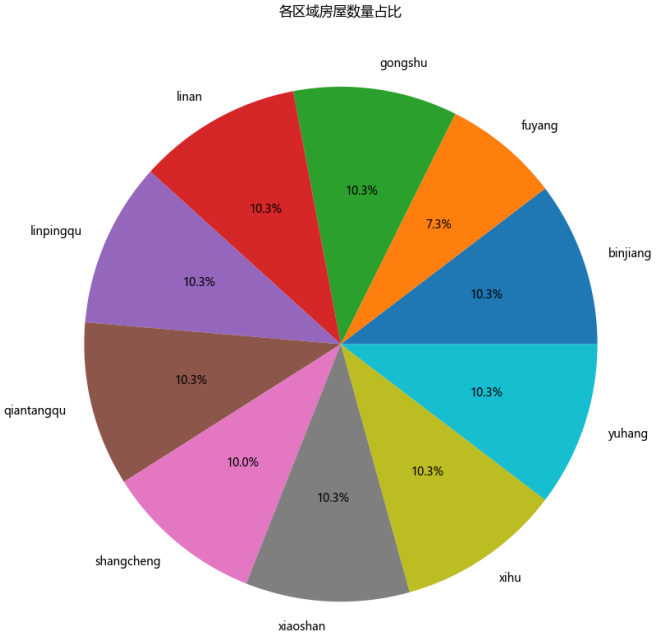


图 3-3 各区域房屋数量占比

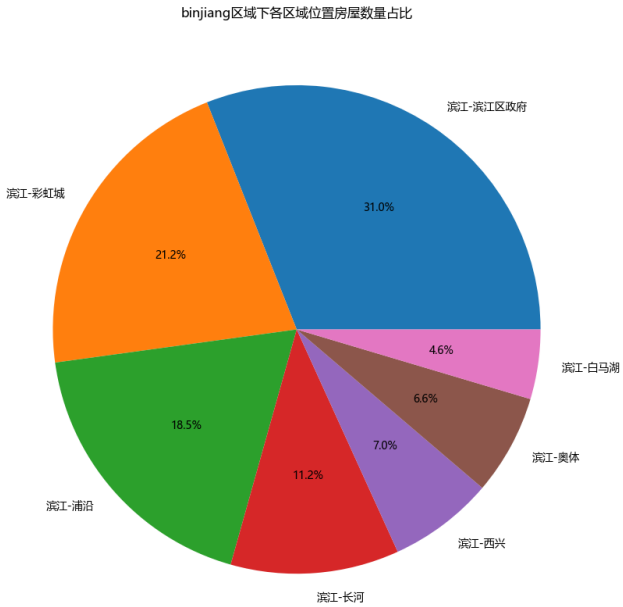


图 3-4 滨江房屋数量占比

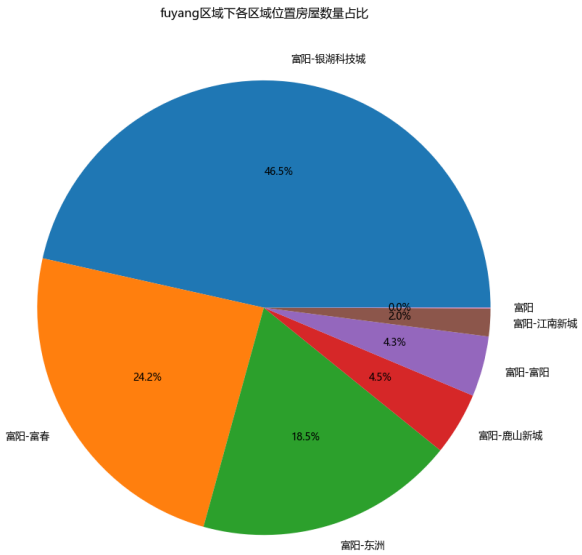


图 3-5 富阳房屋数量占比

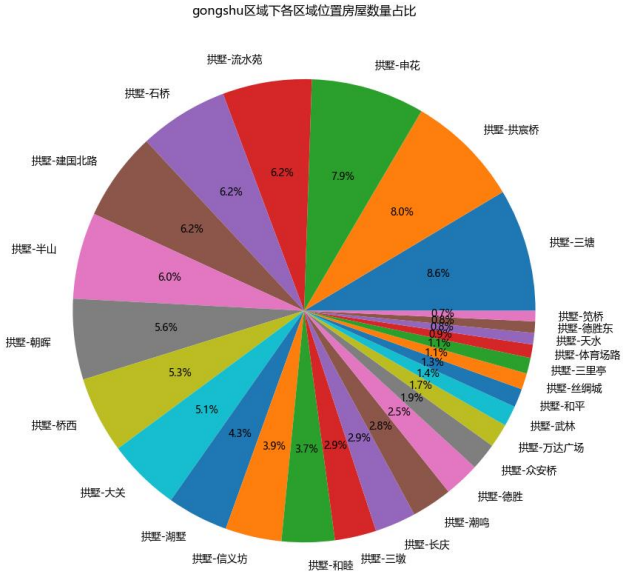


图 3-6 拱墅房屋数量占比

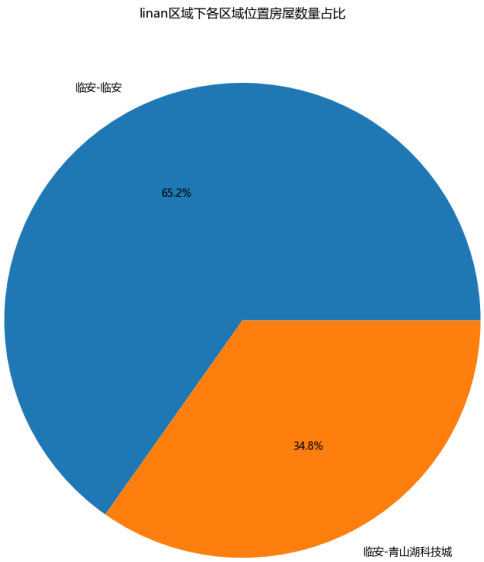


图 3-7 临安房屋数量占比

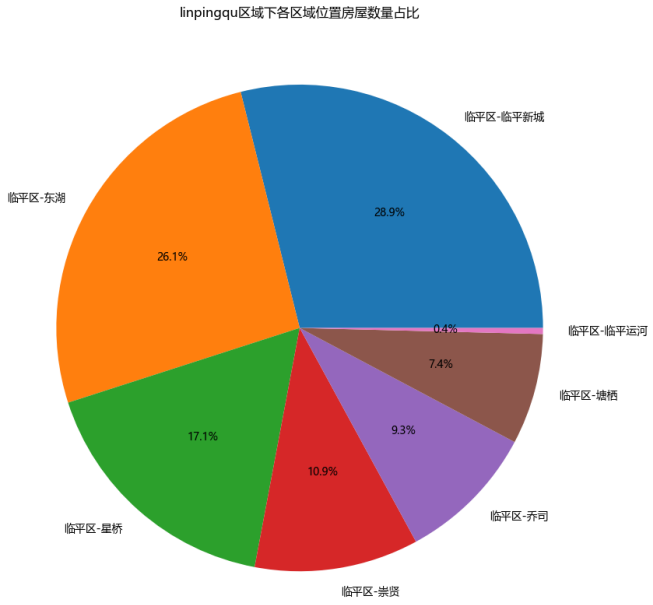


图 3-8 临平区房屋数量占比

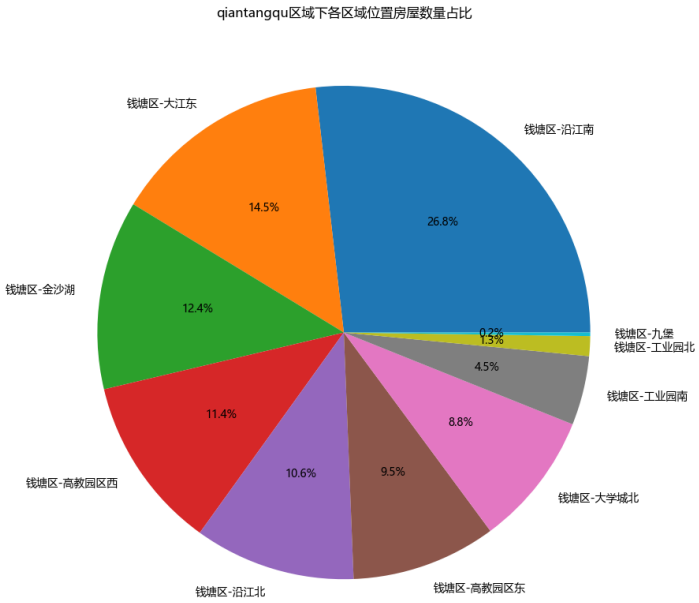


图 3-9 钱塘区房屋数量占比

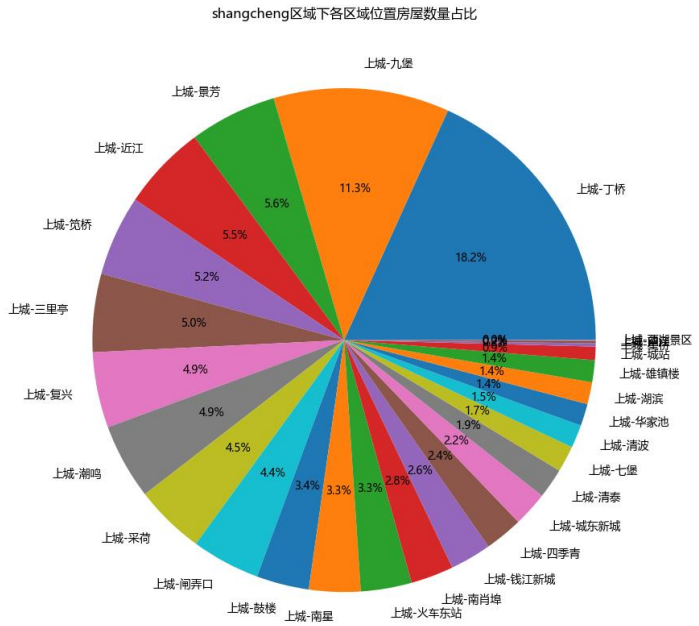


图 3-10 上城房屋数量占比

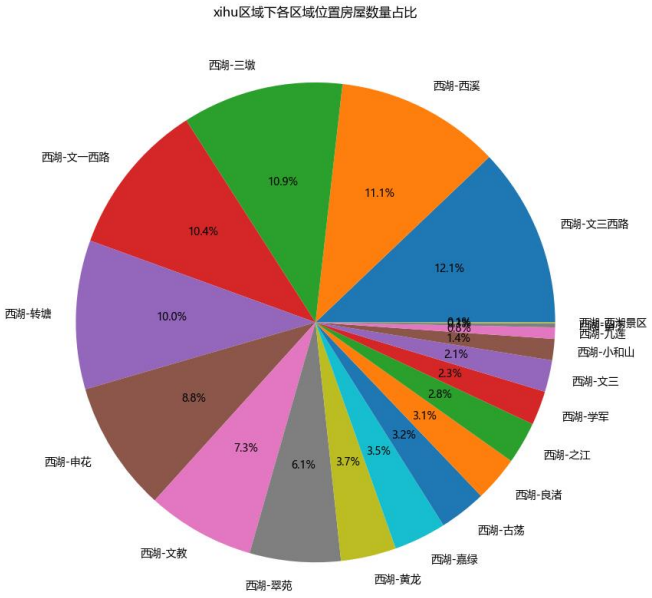


图 3-11 西湖房屋数量占比

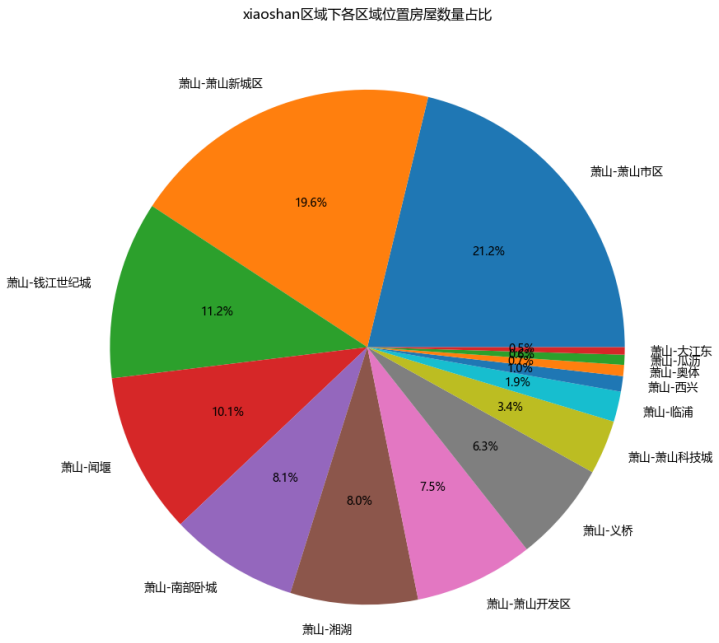


图 3-12 萧山房屋数量占比

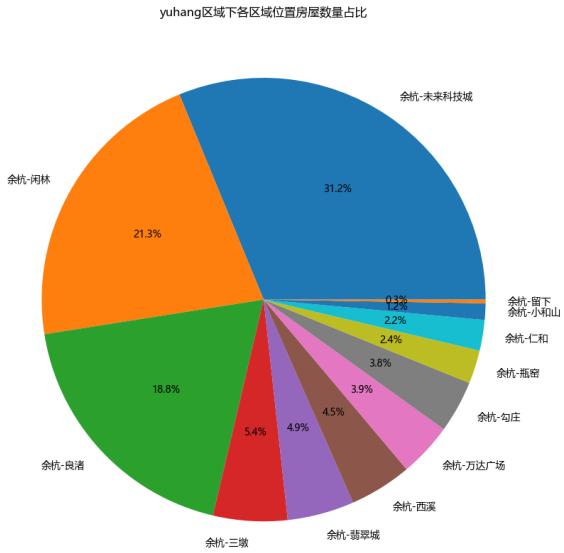


图 3-13 余杭房屋数量占比

（2） 视图分析

视图展示了不同地区和地方的二手房数量比例。

首先，用户可以观察整体饼图，每个扇区代表一个地区，扇区的面积或角度表示该地区的二手房数量占比。通过比较不同地区扇区的大小，用户可以了解哪些地区的二手房数量较多或较少。然后，用户可以选择一个地区，观察该地区内的饼图，每个内部扇区代表一个地方，扇区的面积或角度表示该地方的二手房数量占比。

通过比较不同地方扇区的大小，用户可以了解哪些地方的二手房数量较多或较少。例如，通过观察各区域房屋数量占比饼图可以发现各区域的二手房数量相对是比较均衡的，观察某个区域各个

通过这样的可视化视图，用户可以直观地了解二手房在不同地区和地方的数量比例，并进行比较分析。

3.4 房屋类型视图

（1） 设计思路及设计

房屋类型视图的设计思路是通过条形图和饼图展示房屋的户型、户型结构和建筑类型，以便用户可以了解不同类型的房屋在整体中的分布情况

房屋户型饼图：饼图的每个扇区表示一个户型，扇区的面积或角度与该户型的数量成比例。

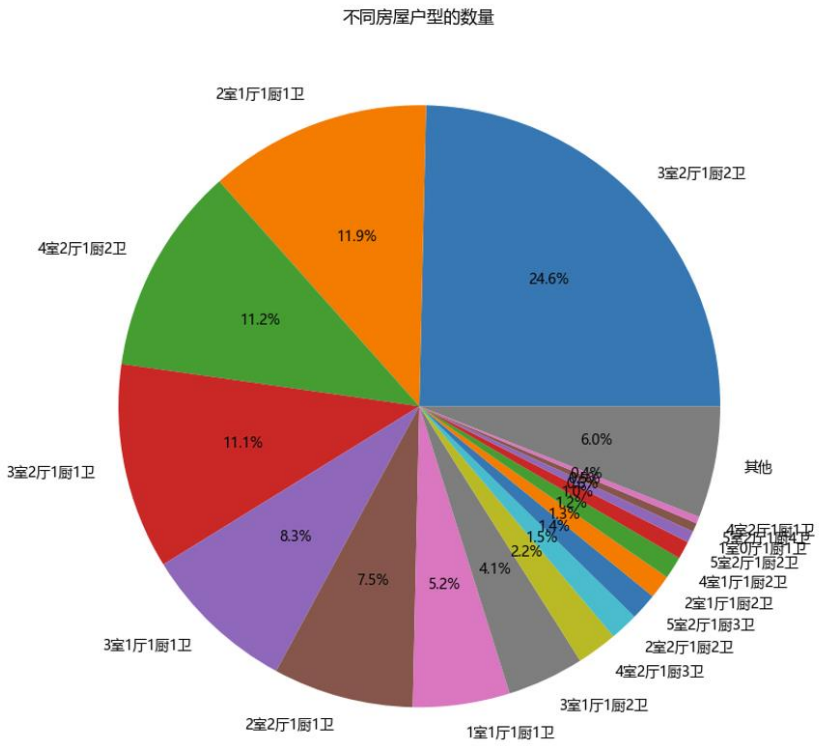


图 3-14 房屋户型饼图

房屋户型条形图：横轴表示不同的户型，纵轴表示户型的数量。每个户型在横轴上有一个对应的条形，条形的高度表示该户型的数量。

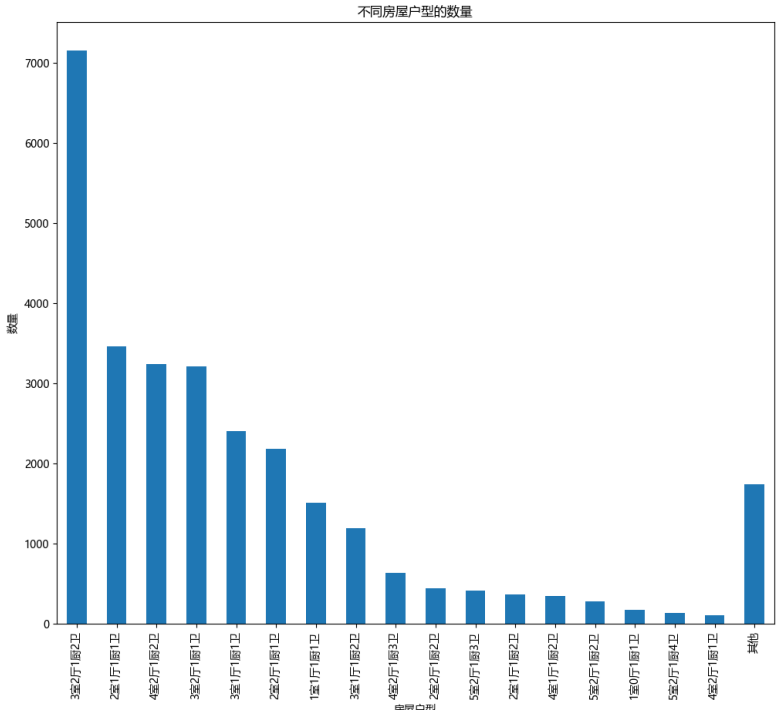


图 3-15 房屋户型条形图

户型结构饼图：饼图的每个扇区表示一个户型结构，扇区的面积或角度与该

户型结构的数量成比例。

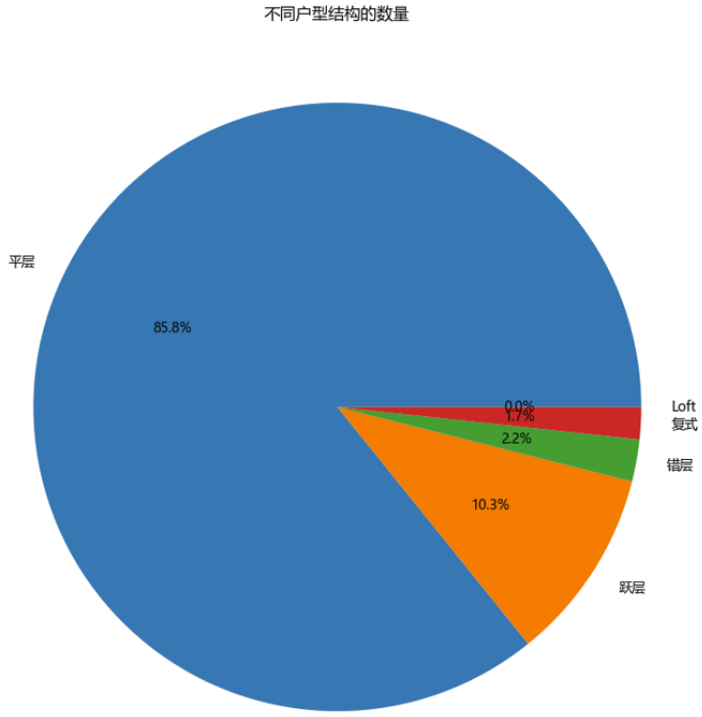


图 3-16 户型结构饼图

户型结构条形图：横轴表示不同的户型结构，纵轴表示户型结构的数量。每个户型结构在横轴上有一个对应的条形，条形的高度表示该户型结构的数量。

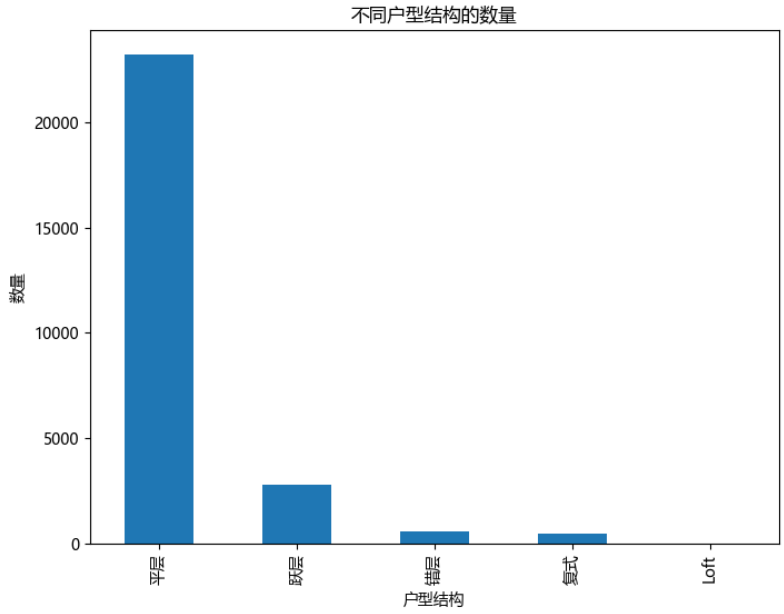


图 3-17 户型结构条形图

建筑类型饼图：饼图的每个扇区表示一个建筑类型，扇区的面积或角度与该

建筑类型的数量成比例。

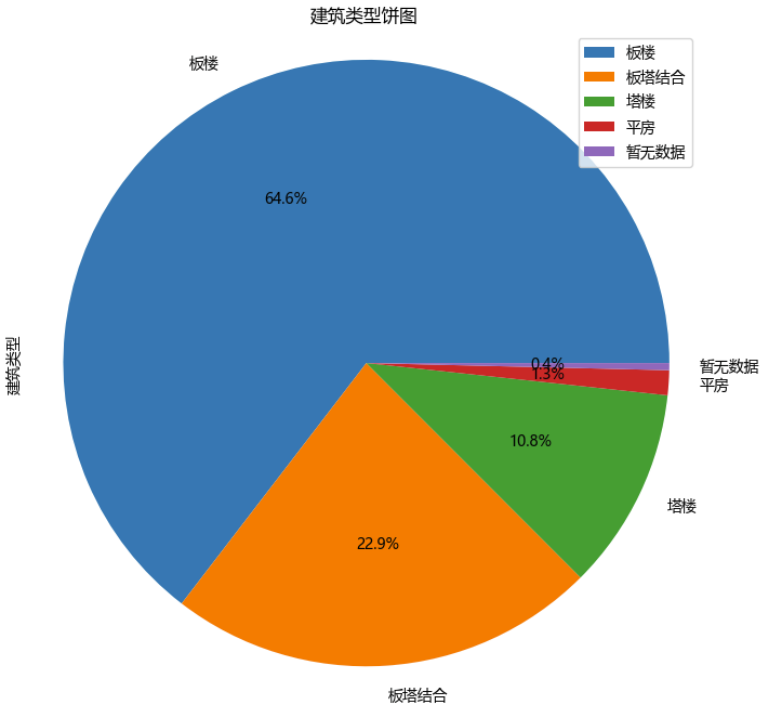


图 3-18 建筑类型饼图

建筑类型条形图：横轴表示不同的建筑类型，纵轴表示建筑类型的数量。每种建筑类型在横轴上有一个对应的条形，条形的高度表示该建筑类型的数量。

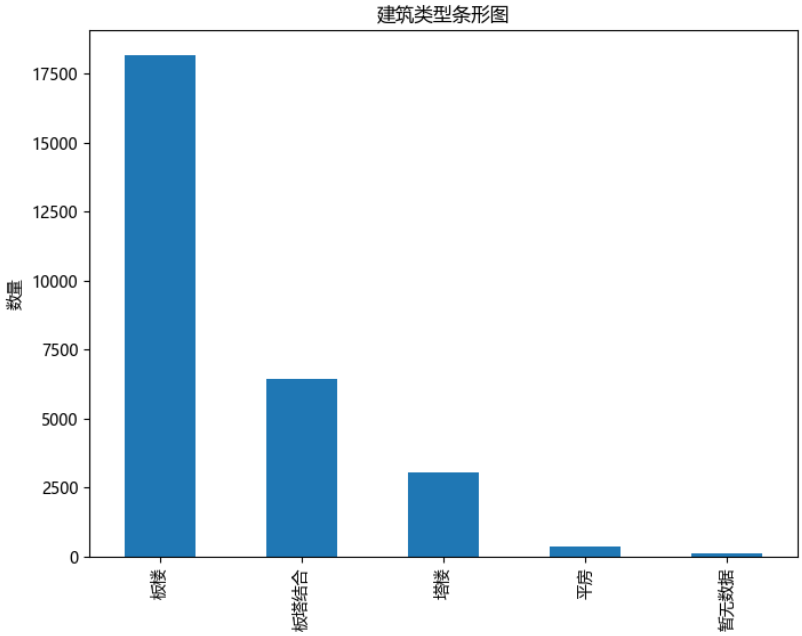


图 3-19 建筑类型条形图

（2） 视图分析

房屋类型视图展示了某个地区的房屋户型、户型结构和建筑类型。

用户可以通过条形图或饼图观察每个图表中不同类型的房屋的数量比例。例如，在房屋户型条形图中，用户可以看到每个户型的条形高度，较高的条形表示该户型的数量较多，而较低的条形表示数量较少的户型。用户还可以通过观察房屋户型饼图的扇区面积或角度，直观地了解每个户型在整体中的占比情况。类似地，用户可以通过户型结构和建筑类型的条形图和饼图，了解不同户型结构和建筑类型的分布情况。

从房屋户型饼图和条形图中，可以观察到 3 室 2 厅 1 厨 2 卫在二手房中占比最多，远高于其他房屋户型，紧随其后的是 2 室 2 厅 1 厨 1 卫、4 室 2 厅 1 厨 2 卫、3 室 2 厅 1 厨 1 卫、3 室 1 厅 1 厨 1 卫这几个房屋户型，同时也可以观察到房屋户型的种类非常多。

从户型结构和建筑类型的饼图和条形图中，可以观察到平层结构以及板楼是主要的户型结构和建筑类型，占比分别为 85.8%和 64.6%。

通过这样的可视化视图，用户可以方便地比较不同类型的房屋在整体中的分布情况，并进行分析和决策。

3.5 介绍词云视图

（1） 设计思路及设计过程

介绍词云视图的设计思路是通过词云展示房屋的核心卖点、周边配套和小区介绍等字段的内容，以便用户可以快速了解房屋的特点和周边环境。根据词频生成词云图，将词频高的词显示为较大的字体，词频低的词显示为较小的字体。

图 3-20 房屋介绍词云

（2） 视图分析

词云视图展示了总体房屋的核心卖点、周边配套和小区介绍。

在词云中，较大字体的词汇可能包括"地铁"、"医院"、"方便"等词语，这表明该房屋的核心卖点是与地铁交通和医疗资源的便利性相关。

较大字体的词汇还可能包括"小区"和"环境"，说明该房屋所在的小区环境优美。

而较小字体的词汇可能包括"农贸市场"、"幼儿园"、"高速"等词语，用户可以得知周边还有农贸市场、幼儿园和高速公路等配套设施。

通过词云视图，用户可以快速了解房屋的特点和周边环境，为购房决策提供参考。此外，还可以从中观察到二手房出售时的介绍主要围绕地理位置，配套设施来展开。

3.6 地理分布视图

（1） 设计思路及设计过程

地理分布视图的设计思路是将小区的地理位置标识在真实地图上，并通过散点图和热力图展示地理分布的信息。

根据小区名称和地理位置信息，在选定的地图上将小区的位置标识出来，可以使用标记或其他符号来表示。

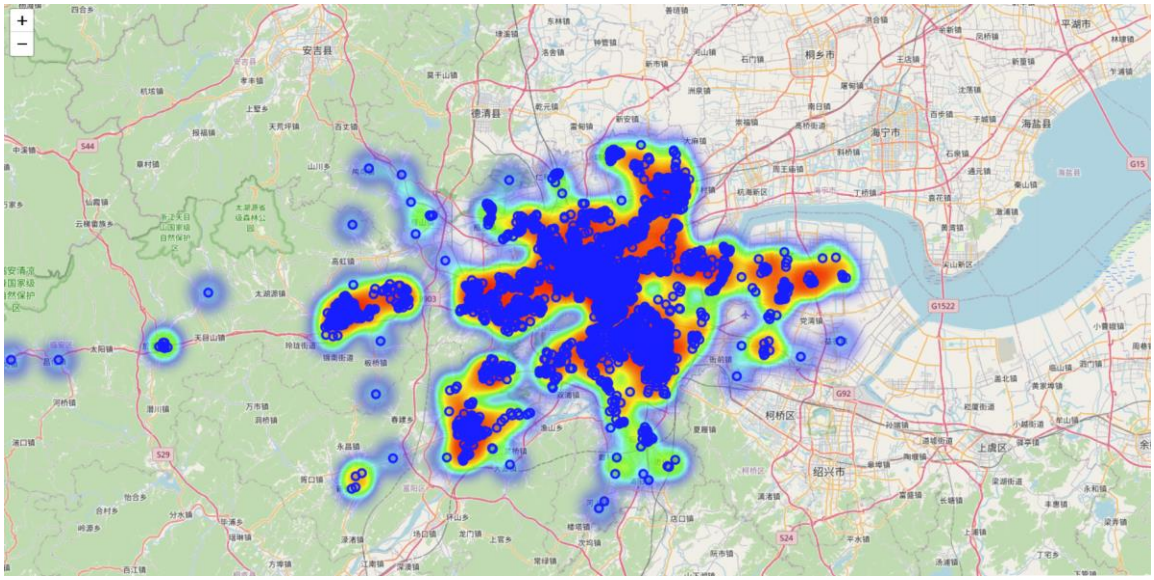


图 3-21 地理位置标识

根据小区的地理位置，绘制散点图，其中每个散点代表一个小区，散点的位置对应小区的地理位置。

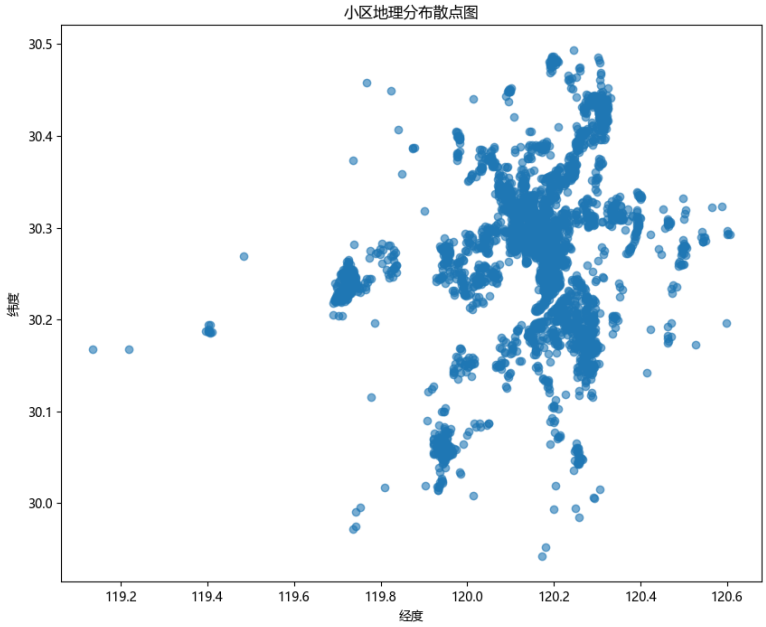


图 3-22 小区地理分布散点图

根据小区的地理位置密度或其他相关指标，绘制热力图。热力图通过不同颜色或渐变色表示地理区域的密集程度，颜色越深表示密度越高。

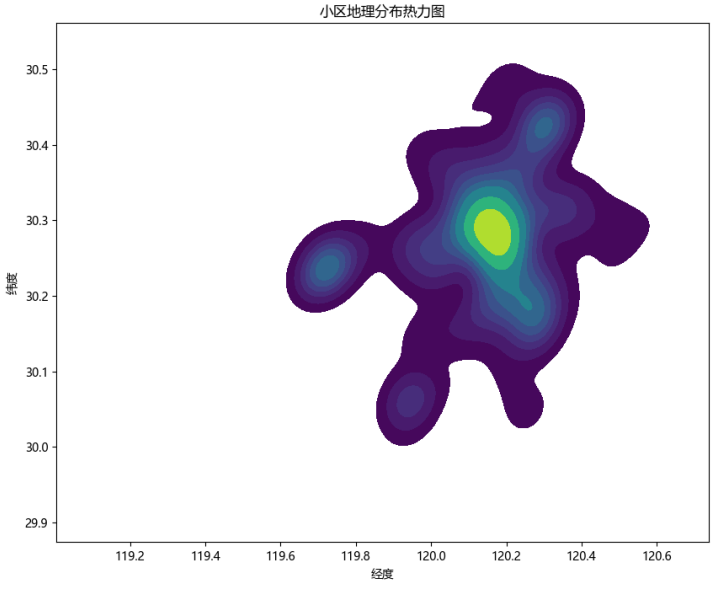


图 3-23 小区地理分布热力图

（2） 交互设计

可以通过鼠标，观察真实地图上不同位置的小区分布情况，也可以进行放大或者缩小，在一个期望的视野范围内进行更加细致的观察。

（3） 视图分析

地理分布视图展示了小区地理分布情况。

在地图上，每个小区以标记或符号的形式标识出来，用户可以看到各个小区在城市地理空间上的位置。

散点图展示了小区的分布情况，如图中密集的散点可能表示该区域有较多的小区集中。

热力图则通过颜色的深浅展示了小区的密度情况，用户可以直观地了解到人口密集的区域和分布趋势。

从地图上可以观察到，西湖的东侧以及北侧，小区数量较多，而西湖的西南侧没有太多的小区，从地图上可以观察到，那里还有村落的存在，相对偏僻和落后。同时，西湖东南侧的钱塘江两岸小区数量也比较多，且密集程度高。此外，临安区是另外一个小区数量多且密集的地方，从地图上可以观察到，这些小区处于道路交汇处，交通比较发达，可能是小区密集的原因之一。

通过地理分布视图，用户可以更好地了解小区的地理位置和分布情况，为选择合适的区域或了解城市的整体格局提供参考。

3.7 建筑面积与套内面积拟合视图

（1） 设计思路及设计过程

建筑面积与套内面积拟合视图的设计思路是通过绘制散点图和曲线，展示建筑面积与套内面积之间的关系以及拟合程度。

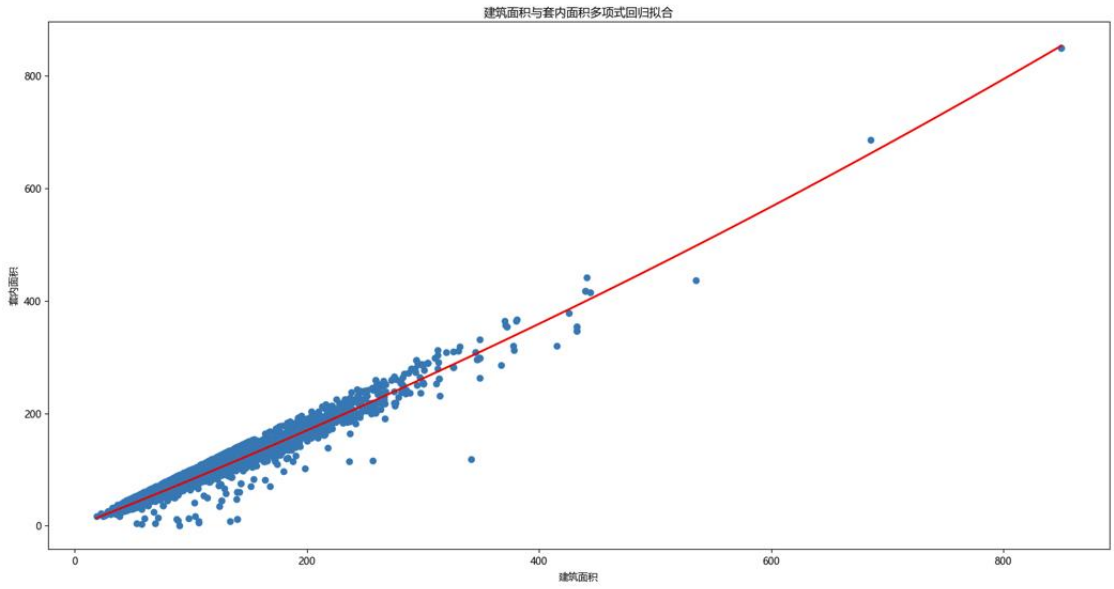


图 3-24 建筑面积与套内面积拟合

（2） 视图分析

在散点图中，每个散点代表一个房屋单位，横轴表示建筑面积，纵轴表示套内面积。

通过观察散点的分布和拟合曲线，用户可以得出以下信息：当建筑面积增加时，套内面积也相应增加，表明建筑面积与套内面积之间存在正相关关系。如果拟合曲线与散点分布接近重合，表示建筑面积与套内面积之间拟合程度较好，可以较准确地预测套内面积。如果拟合曲线与散点分布较分散，则拟合效果较差，建筑面积与套内面积之间的关系较为复杂或不明显。

通过建筑面积与套内面积拟合视图，用户可以直观地了解建筑面积与套内面积之间的关系及其拟合程度，为购房决策提供参考。

4 设计总结与心得

4.1 实验总结

本次实验旨在通过数据预处理和多种可视化视图对二手房数据进行分析和展示。通过实验，我们对数据进行了预处理，包括清洗、整理和合并，以确保数据的完整性和准确性。随后，我们设计和实现了多个可视化视图，以便更直观地展示数据的特征和趋势。

首先，通过总价视图，我们可以了解二手房总价的分布情况。通过直方图、箱线图等图表，我们可以观察到总价的集中趋势、异常值以及不同价格区间的房屋数量分布情况。

其次，通过二手房数量视图，我们可以观察不同地区或时间段内二手房数量的变化趋势。通过折线图、柱状图等图表，我们可以了解二手房市场的热度和趋势，以及不同地区或时间段的供需情况。

接下来，通过房屋类型视图，我们可以了解不同类型的房屋在二手房市场中的分布情况。通过饼图、条形图等图表，我们可以观察各种房屋类型的比例或数量，为购房者或投资者提供决策参考。

通过介绍词云视图，我们可以抓住二手房房源描述中的关键词，进一步了解房屋特点、装修风格或配套设施。这样的可视化视图可以帮助我们了解市场需求和用户偏好。

最后，通过地理分布视图，我们可以展示二手房在地理空间上的分布情况。通过地图或热力图等可视化方式，我们可以了解不同地区或街道的二手房数量或总价分布情况。

通过本次实验，我们学到了数据预处理的基本步骤和技巧，以及如何使用不同类型的可视化视图来展示数据。这些技能对于数据分析和决策支持非常重要。同时，在实验过程中，我们也面临了一些挑战，如处理缺失值、调整图表布局等问题，但通过查阅文档和不断尝试，我们成功地克服了这些问题。

总的来说，本次实验提供了一个综合性的数据分析和可视化实践，加深了我们对二手房数据的理解，并提升了数据处理和可视化的能力。这些技能对于实际的数据工作和决策支持具有重要意义。

4.2 实验心得

在完成本次实验的过程中，我获得了一些宝贵的心得体会。以下是我对实验

的心得总结：

数据预处理是数据分析的关键步骤：数据预处理是确保数据质量和准确性的重要步骤。在本次实验中，我学会了如何处理缺失值、异常值和重复值，并进行数据类型转换和字段合并。这些预处理步骤对于后续的分析和可视化非常关键，因为只有经过清洗和整理的数据才能提供准确的结果和洞察。

可视化是理解数据的有力工具：通过可视化视图，我们可以更直观地理解数据的特征和趋势。在本次实验中，我使用了多种可视化方式，如直方图、折线图、饼图等，来展示不同方面的数据分布和变化。这些图表帮助我更好地把握数据的整体情况，并发现其中的规律和异常。

持续学习和实践是提高技能的关键：在实验过程中，我遇到了一些挑战，如处理缺失值、调整图表布局等。然而，通过查阅文档、学习资料和尝试不同的方法，我逐渐克服了这些难题，并取得了进步。这个过程让我意识到，持续学习和实践是提高技能和解决问题的关键。只有不断锻炼和尝试，我们才能在数据分析领域不断进步。

数据分析的重要性在不断增加：通过本次实验，我更深刻地认识到数据分析在决策和问题解决中的重要性。通过对二手房数据的分析，我们可以了解市场趋势、用户需求和竞争态势，为决策提供有力支持。数据分析能够揭示隐藏在数据背后的信息和规律，帮助我们做出更明智的决策。

综上所述，通过本次实验，我不仅学到了数据预处理和可视化的技能，还培养了持续学习和解决问题的能力。我深刻认识到数据分析在现代社会中的重要性，以及不断提升自己的必要性。我期待将来能够在实际工作和学习中应用这些技能，为解决现实问题做出贡献。

4.3 意见与建议

深入实用工具和技术：在课程中可以涵盖更多实用的数据分析工具和技术，如常用的数据预处理工具、可视化工具、机器学习算法等。这样可以让学生更广泛地了解和掌握不同工具的使用，为实际工作做好准备。

加强团队合作和沟通能力培养：数据分析通常需要团队合作，因此课程可以引入团队项目，培养学生的团队合作能力和沟通能力。这样可以更好地培养学生与他人合作、交流和协作的能力，符合实际工作环境的需求。

原创性声明

本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。

已阅读并同意以下内容。

判定为不合格的一些情形：

（1） 请人代做或冒名顶替者；

（2） 替人做且不听劝告者；

（3） 实验报告内容抄袭或雷同者；

（4） 实验报告内容与实际实验内容不一致者；

（5） 实验代码抄袭者。

作者签名：

附录一 data.py

```
import os
import pandas as pd

# 获取 data 文件夹下所有 CSV 文件的文件名
folder_path = 'data/'
csv_files = [file for file in os.listdir(folder_path) if
file.endswith('.csv')]

# 创建一个空的 DataFrame 用于存储合并的结果
merged_data = pd.DataFrame()

# 遍历每个 CSV 文件并进行合并
for file in csv_files:
    # 读取 CSV 文件
    file_path = os.path.join(folder_path, file)
    data = pd.read_csv(file_path,encoding='GBK')

    # 添加文件名列
    print(file)
    data['区域'] = file[0:-9]

    # 将当前 CSV 文件的数据合并到结果 DataFrame 中
    merged_data = pd.concat([merged_data, data], ignore_index=True)

# 将合并结果保存为新的 CSV 文件
merged_data.to_csv('merged_data.csv', index=False,encoding='utf-8')
```

附录二 attention.py

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import seaborn as sns
from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解
决 plot 不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方
块的问题

# 读取数据
data = pd.read_csv('./merged_data.csv')
```

```
# 绘制散点图
plt.figure(figsize=(10, 8))
sns.scatterplot(data=data, x='经度', y='纬度', size='关注度', sizes=(20, 200), alpha=0.7)
plt.title('位置与关注度散点图')
plt.show()

# 绘制热力图
sns.kdeplot(data=data, x='经度', y='纬度', weights='关注度', cmap='hot', shade=True)
plt.title('热力图')
plt.show()
```

附录三 count.py

```
import pandas as pd
import matplotlib.pyplot as plt

from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解决 plot 不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方块的问题

# 读取数据
data = pd.read_csv('./merged_data.csv')

data_region = data.groupby("区域").size()

# 绘制饼图
plt.figure(figsize=(12, 12))
data_region.plot(kind='pie', autopct='%.1f%%')
plt.ylabel('')
plt.title('各区域房屋数量占比')
plt.show()


# 获取区域位置
locations = data.groupby("区域")["区域位置"].value_counts()
# 获取所有的区域
regions = data["区域"].unique()

# 遍历每个区域，并绘制饼图
for region in regions:
    region_data = locations[region]
```

```
    # 绘制饼图
    plt.figure(figsize=(12, 12))
    region_data.plot(kind='pie', autopct='%.1f%%')
    plt.ylabel('')
    plt.title(f'{region}区域下各区域位置房屋数量占比')
    plt.show()
```

附录四 huxing.py

```
import pandas as pd
import matplotlib.pyplot as plt

from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解决 plot 不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方块的问题

# 读取数据
data = pd.read_csv('./merged_data.csv')

grouped = data["户型结构"].value_counts()
grouped = grouped.drop("暂无数据", errors='ignore')

# 绘制柱状图
plt.figure(figsize=(8, 6))
grouped.plot(kind='bar')
plt.xlabel('户型结构')
plt.ylabel('数量')
plt.title('不同户型结构的数量')

plt.show()

# 绘制饼图
plt.figure(figsize=(12, 12))
grouped.plot(kind='pie', autopct='%.1f%%')
plt.ylabel('')
plt.title('不同户型结构的数量')
plt.show()


grouped = data["房屋户型"].value_counts()
# 将数量低于 100 的房屋户型合并为"其他"
threshold = 100
```

```
other_count = grouped[grouped <= threshold].sum()
grouped = grouped[grouped >= threshold]
grouped['其他'] = other_count


# 绘制柱状图
plt.figure(figsize=(12, 12))
grouped.plot(kind='bar')
plt.xlabel('房屋户型')
plt.ylabel('数量')
plt.title('不同房屋户型的数量')

plt.show()

# 绘制饼图
plt.figure(figsize=(8, 8))
grouped.plot(kind='pie', autopct='%.1f%%')
plt.ylabel('')
plt.title('不同房屋户型的数量')
plt.show()
```

附录五 introduction.py

```
import pandas as pd
from wordcloud import WordCloud
import jieba
import matplotlib.pyplot as plt

from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解决plot不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方块的问题

# 读取数据
data = pd.read_csv('./merged_data.csv')

# 将小区介绍和周边配套的文本数据合并
data['小区介绍'] = data['小区介绍'].fillna('')  # 处理小区介绍列中的缺失值
data['周边配套'] = data['周边配套'].fillna('')  # 处理周边配套列中的缺失值
data['核心卖点'] = data['核心卖点'].fillna('')  # 处理核心卖点列中的缺失值
text_data = data['小区介绍'] + ' ' + data['周边配套'] + ' ' + data['核心卖点']

# 文本分词处理
```

```
text_seg = ' '.join(jieba.cut(' '.join(text_data)))

# 生成文字云图
wordcloud = WordCloud(width=800, height=400, background_color='white',
font_path='C:\Windows\Fonts\msyh.ttc',
collocations=False).generate(text_seg)

# 显示文字云图
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('房屋介绍文字云图')
plt.show()
```

附录六 location.py

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解决 plot 不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方块的问题

# 读取数据
data = pd.read_csv('./merged_data.csv')

# 去重
data = data.drop_duplicates(subset=['小区名称'])

# 绘制散点图
plt.figure(figsize=(10, 8))
plt.scatter(data['经度'], data['纬度'], alpha=0.6)
plt.xlabel('经度')
plt.ylabel('纬度')
plt.title('小区地理分布散点图')
plt.show()

# 绘制热力图
plt.figure(figsize=(10, 8))
sns.kdeplot(data['经度'], data['纬度'], shade=True, cmap='viridis')
plt.xlabel('经度')
plt.ylabel('纬度')
```

```
plt.title('小区地理分布热力图')
plt.show()
```

附录七 price.py

```
import pandas as pd
import matplotlib.pyplot as plt
from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解决plot不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方块的问题

file_path = './merged_data.csv'

data = pd.read_csv(file_path,encoding='utf-8')


# 计算每个区域的单价中位数
median_prices = data.groupby('区域')['单价'].median()

# 绘制直方图
plt.figure(figsize=(10, 6))
median_prices.plot(kind='bar')
plt.xlabel('区域')
plt.ylabel('单价中位数')
plt.title('每个区域的单价中位数直方图')
plt.xticks(rotation=45)
plt.show()

median_prices = data.groupby(['区域', '区域位置'])['单价
'].median().reset_index()

unique_regions = median_prices['区域'].unique()
num_regions = len(unique_regions)

num_cols = 3  # 每行显示的子图数量
num_rows = (num_regions - 1) // num_cols + 1  # 计算行数

fig, axes = plt.subplots(nrows=num_rows, ncols=num_cols, figsize=(15,
5*num_rows))

# 计算纵轴范围
max_price = median_prices['单价'].max()
min_price = median_prices['单价'].min()
```

```
for i, region in enumerate(unique_regions):
    region_data = median_prices[median_prices['区域'] == region]
    row = i // num_cols
    col = i % num_cols
    ax = axes[row][col]
    ax.bar(region_data['区域位置'], region_data['单价'])
    ax.set_xlabel('区域位置')
    ax.set_ylabel('单价中位数')
    ax.set_title(f'{region}区域位置的单价中位数')
    ax.tick_params(axis='x', rotation=45)
    ax.set_ylim(min_price, max_price)  # 统一纵轴范围

# 隐藏多余的子图
if num_regions < num_rows * num_cols:
    for i in range(num_regions, num_rows*num_cols):
        row = i // num_cols
        col = i % num_cols
        fig.delaxes(axes[row][col])

plt.tight_layout()
plt.show()
```

附录八 real_map.py

```
import pandas as pd
import folium
from folium.plugins import HeatMap

# 读取数据
data = pd.read_csv('./merged_data.csv')

# 去重
data = data.drop_duplicates(subset=['小区名称'])

# 创建地图
m = folium.Map(location=[data['纬度'].mean(), data['经度'].mean()],
zoom_start=10)

# 绘制散点图
for _, row in data.iterrows():
    folium.CircleMarker(location=[row['纬度'], row['经度']], radius=5,
color='blue', fill=True).add_to(m)

# 绘制热力图
```

```
heat_data = [[row['纬度'], row['经度'], row['关注度']] for _, row in
data.iterrows()]
HeatMap(heat_data).add_to(m)

# 保存地图
m.save('map.html')
```

附录九 square.py

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from pylab import mpl
from sklearn.preprocessing import PolynomialFeatures

mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解
决 plot 不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方
块的问题

# 读取数据
data = pd.read_csv('./merged_data.csv')

# 去除建筑面积和套内面积字段中的单位和暂无数据
data['建筑面积'] = data['建筑面积'].str.replace('㎡', '')
data['套内面积'] = data['套内面积'].str.replace('㎡', '')
data = data[data['建筑面积'] != '暂无数据']
data = data[data['套内面积'] != '暂无数据']
data['建筑面积'] = pd.to_numeric(data['建筑面积'])
data['套内面积'] = pd.to_numeric(data['套内面积'])
data = data.dropna(subset=['套内面积', '建筑面积'])

x = data['建筑面积'].values.reshape(-1, 1)
y = data['套内面积'].values.reshape(-1, 1)

# 多项式特征转换
poly_features = PolynomialFeatures(degree=2)  # 设置多项式的阶数
x_poly = poly_features.fit_transform(x)

# 多项式回归模型训练
model = LinearRegression()
model.fit(x_poly, y)
```

```
# 预测并绘制曲线
x_pred = np.linspace(x.min(), x.max(), 100).reshape(-1, 1)
x_pred_poly = poly_features.transform(x_pred)
y_pred = model.predict(x_pred_poly)

plt.scatter(x, y)
plt.plot(x_pred, y_pred, color='red', linewidth=2)
plt.xlabel('建筑面积')
plt.ylabel('套内面积')
plt.title('建筑面积与套内面积多项式回归拟合')
plt.show()
```

附录十 total_price.py

```
import pandas as pd
import matplotlib.pyplot as plt
from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解决 plot 不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方块的问题

file_path = './merged_data.csv'

data = pd.read_csv(file_path,encoding='utf-8')


# 计算每个区域的总价中位数
median_prices = data.groupby('区域')['总价'].median()

# 绘制直方图
plt.figure(figsize=(10, 6))
median_prices.plot(kind='bar')
plt.xlabel('区域')
plt.ylabel('总价中位数')
plt.title('每个区域的总价中位数直方图')
plt.xticks(rotation=45)
plt.show()

median_prices = data.groupby(['区域', '区域位置'])['总价
'].median().reset_index()

unique_regions = median_prices['区域'].unique()
num_regions = len(unique_regions)
```

```
num_cols = 3  # 每行显示的子图数量
num_rows = (num_regions - 1) // num_cols + 1  # 计算行数

fig, axes = plt.subplots(nrows=num_rows, ncols=num_cols, figsize=(15,
5*num_rows))

# 计算纵轴范围
max_price = median_prices['总价'].max()
min_price = median_prices['总价'].min()

for i, region in enumerate(unique_regions):
    region_data = median_prices[median_prices['区域'] == region]
    row = i // num_cols
    col = i % num_cols
    ax = axes[row][col]
    ax.bar(region_data['区域位置'], region_data['总价'])
    ax.set_xlabel('区域位置')
    ax.set_ylabel('总价中位数')
    ax.set_title(f'{region}区域位置的总价中位数')
    ax.tick_params(axis='x', rotation=45)
    ax.set_ylim(min_price, max_price)  # 统一纵轴范围

# 隐藏多余的子图
if num_regions < num_rows * num_cols:
    for i in range(num_regions, num_rows*num_cols):
        row = i // num_cols
        col = i % num_cols
        fig.delaxes(axes[row][col])

plt.tight_layout()
plt.show()
```

附录十一 type.py

```
import pandas as pd
import matplotlib.pyplot as plt
from pylab import mpl
mpl.rcParams['font.sans-serif'] = ['Microsoft YaHei'] # 指定默认字体：解
决plot不能显示中文问题
mpl.rcParams['axes.unicode_minus'] = False # 解决保存图像是负号'-'显示为方
块的问题

# 读取数据
data = pd.read_csv('./merged_data.csv')
```

```
# 统计建筑类型的计数
building_counts = data['建筑类型'].value_counts()

# 绘制建筑情况的条形图
plt.figure(figsize=(8, 6))
building_counts.plot(kind='bar')
plt.xlabel('建筑类型')
plt.ylabel('数量')
plt.title('建筑类型条形图')
plt.show()

# 绘制装修类型的饼图
plt.figure(figsize=(8, 8))
building_counts.plot(kind='pie', autopct='%1.1f%%')
plt.axis('equal')
plt.title('建筑类型饼图')
plt.legend()
plt.show()

# 统计装修情况的计数
decoration_counts = data['装修情况'].value_counts()

# 绘制建筑情况的条形图
plt.figure(figsize=(8, 6))
decoration_counts.plot(kind='bar')
plt.xlabel('装修情况')
plt.ylabel('数量')
plt.title('装修情况条形图')
plt.show()

# 绘制装修类型的饼图
plt.figure(figsize=(8, 8))
decoration_counts.plot(kind='pie', autopct='%1.1f%%')
plt.axis('equal')
plt.title('装修情况饼图')
plt.legend()
plt.show()
```