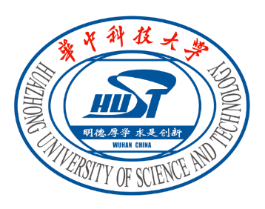
****

**华中科技大学2024～2025学年第一学期**

**“GIS城市大数据”结课论文**

|  |  |
| --- | --- |
| **院（系）：** | **公共管理学院** |
| **专业班级：** | **公共事业管理2401班** |
| **学号：** | **U202416071** |
| **姓名：** | **梅帅** |
| **成绩：** |  |
| **签名：** |  |

**提交日期：2026年1月1日**

### ****长三角地区土地利用变化及建设用地驱动性分析报告****

### ****摘要****

本研究以长三角地区（安徽、江苏、浙江、上海）为研究区，基于多期遥感影像与地理空间数据，运用ArcGIS空间分析技术、土地利用转移矩阵Python数据分析技术，系统分析研究时段内土地利用类型的时空演变特征，并定量识别建设用地扩张的核心驱动因素。

研究结果显示，长三角地区土地利用格局呈现显著的城乡转型特征，建设用地持续扩张且主要来源于耕地转换。在驱动力定量识别上，地形地貌（坡度与高程）对建设用地空间分布表现出最强的解释力，构成了区域扩张的基础性约束；同时，人口集聚与社会经济活力（灯光指数）也是推动建设用地演化的核心驱动因子。

1. ****研究区概况与数据来源****

1.1研究区概况

长三角地区位于我国东部沿海，地理范围涵盖安徽省、江苏省、浙江省及上海市，是我国经济最活跃、城镇化水平最高的区域之一。该区域地势平坦、河网密布，兼具平原、丘陵等多样地形，土地利用类型复杂且动态变化剧烈，是研究土地利用转型与城市化驱动机制的典型区域。

1.2数据来源

基础地理数据：全国省级行政区边界矢量数据（标准地图服务系统）、长三角地区数字高程模型数据。

土地利用数据：Globeland30地类。

社会经济数据：研究时段内长三角四省/市人口数量、灯光亮度数据。

****二、研究方法与技术路线****

2.1核心研究方法

笔者主要运用了四种研究方法：

1. 土地利用分类方法：对研究区土地利用类型进行可视化表达
2. 空间分析方法：运用ArcGISPro的叠加分析、缓冲区分析、空间统计等工具，分析土地利用空间格局变化。
3. 转移矩阵分析法：构建土地利用转移矩阵，量化不同土地利用类型间的转换面积与转移概率，明确建设用地的主要来源。
4. 地理探测器模型法：选取人口密度、距城市中心距离等因子，通过地理探测器的因子探测、交互探测功能，识别建设用地扩张的驱动因素及因子间交互作用强度。

2.2技术路线

具体的操作过程，主要涉及以下四类技术：

1. 数据预处理：包括遥感影像裁剪、校正、配准，矢量数据投影统一及属性完善。
2. 土地利用变化分析：通过转移矩阵、面积变化率、空间格局指数等，分析土地利用时空演变特征。
3. 驱动因子选取与数据标准化：筛选核心驱动因子，完成因子数据的标准化处理与空间化表达。
4. 驱动性分析：运用地理探测器模型，开展因子探测与交互探测，量化驱动因子影响力。
5. ****数据预处理与实验步骤****

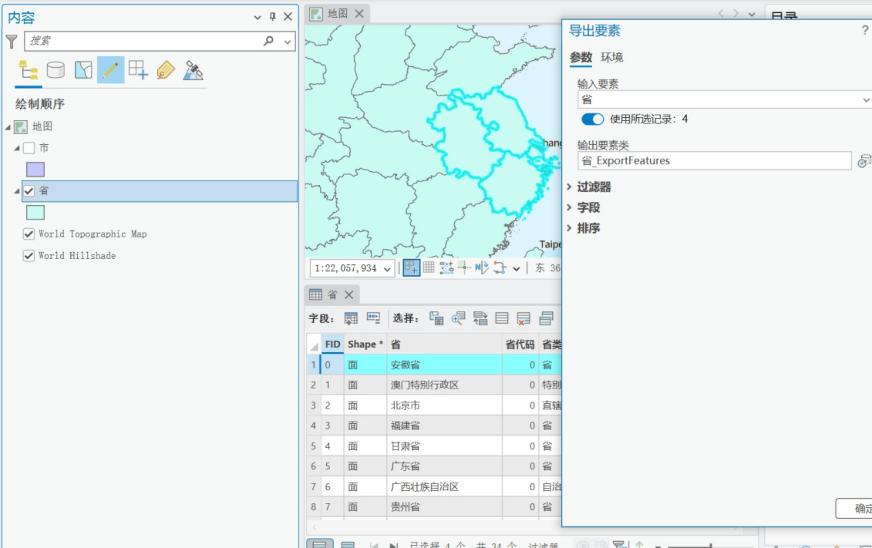
3.1实验项目创建

在ArcGIS Pro中创建“长三角地区土地利用变化及驱动性分析”项目。

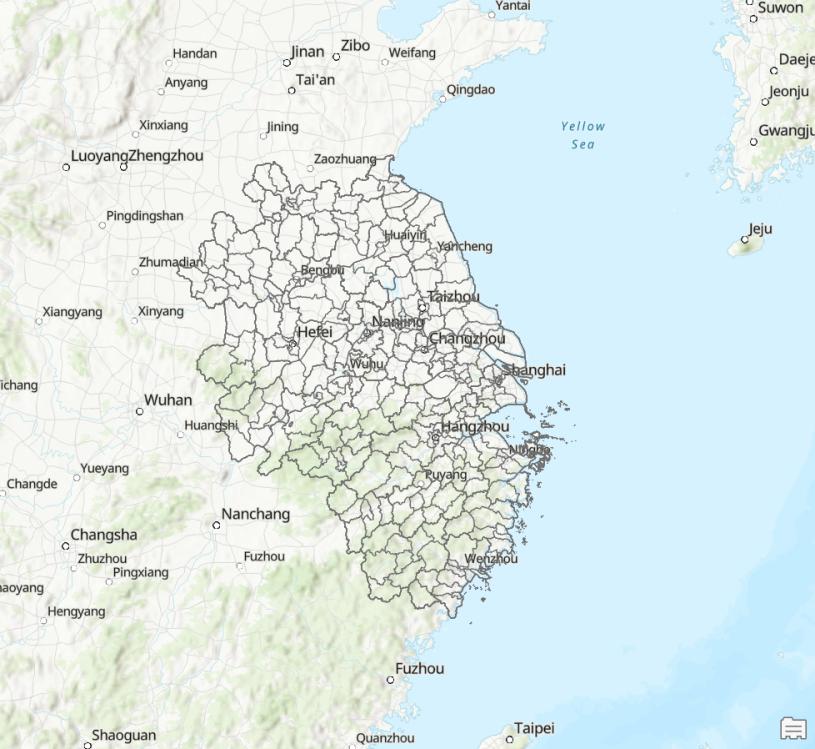


3.2研究区范围提取

在全国省级行政区数据中，通过属性选择工具筛选安徽、江苏、浙江、上海四省（市）边界要素。



运用“导出要素”工具，将筛选后的边界数据导出为“长三角省级边界”要素类，作为研究区范围边界，然后提取市县二级行政区。



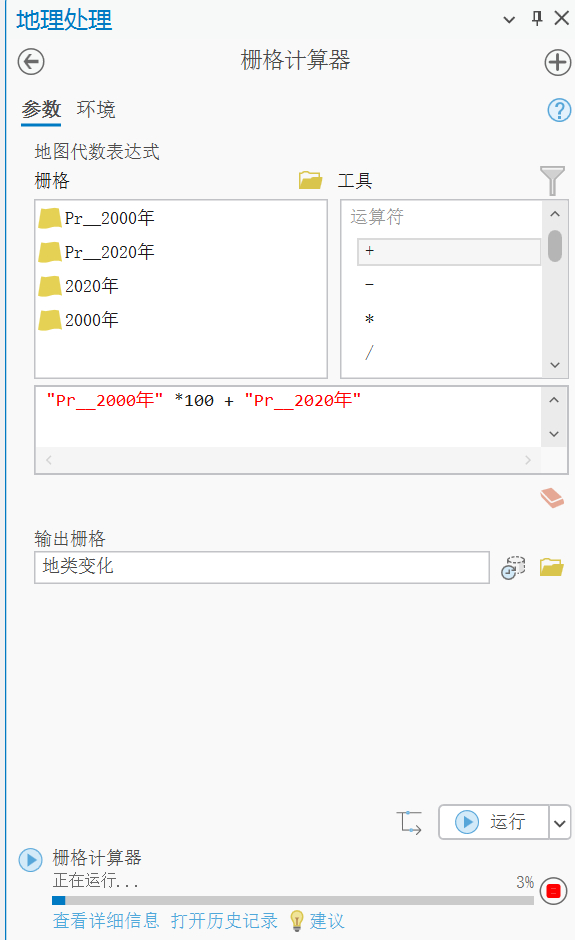
3.3土地利用数据提取

通过掩膜提取，Globeland30地类数据，得到2期地类。

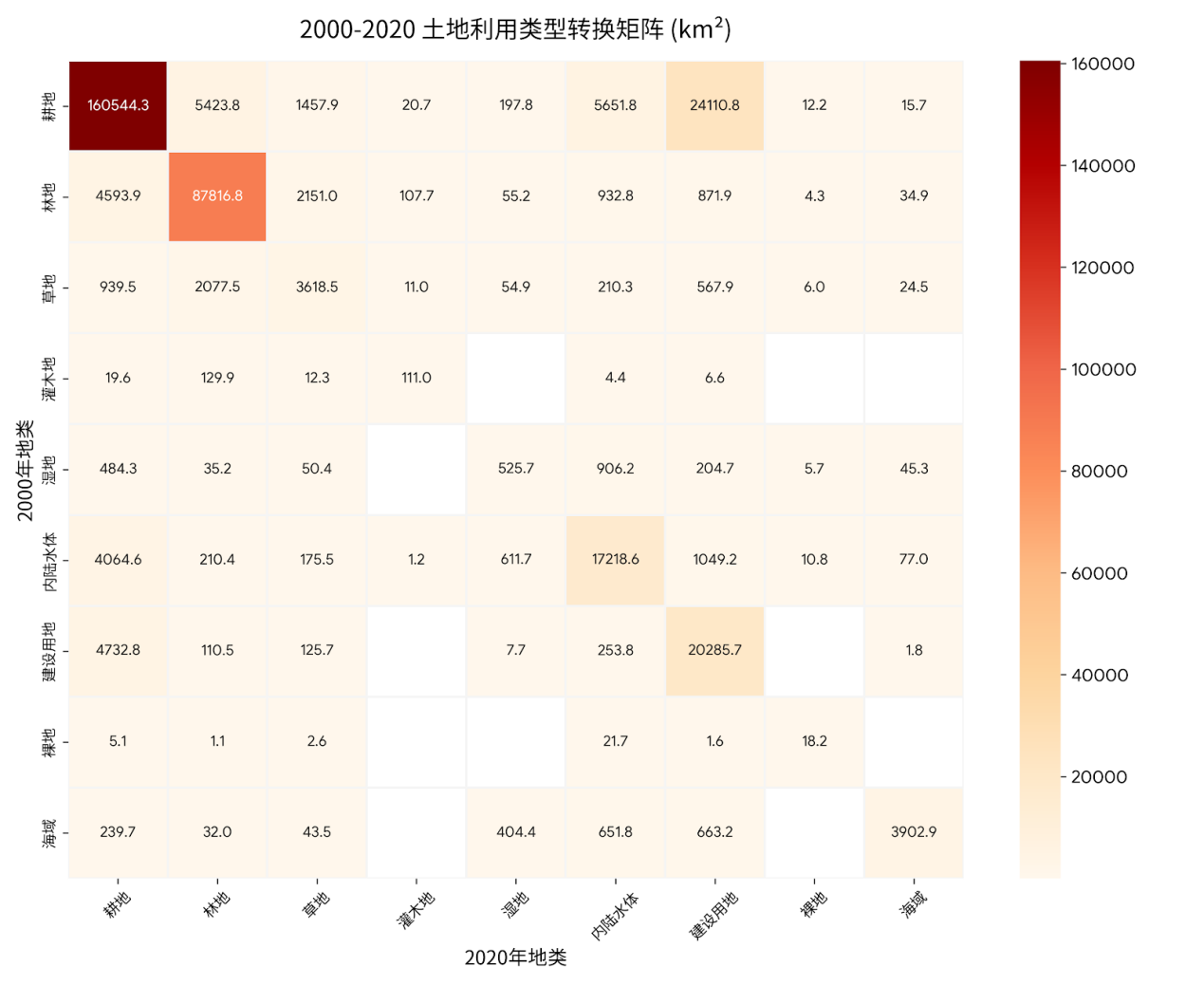


3.4土地利用变化数据计算

运用ArcGISPro栅格计算器工具，对两期土地利用分类数据进行空间叠加，提取土地利用变化信息。



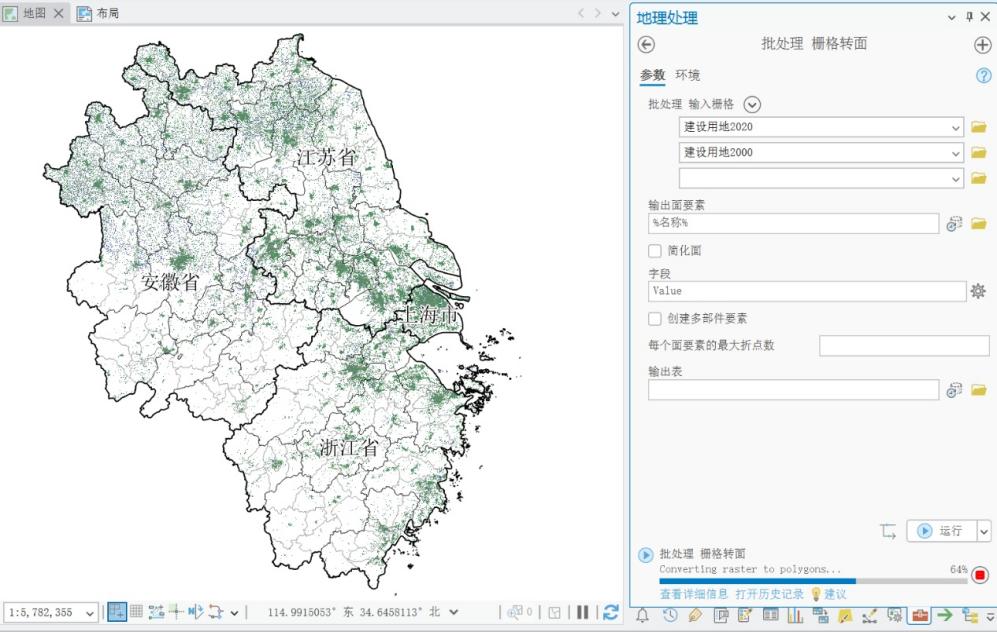
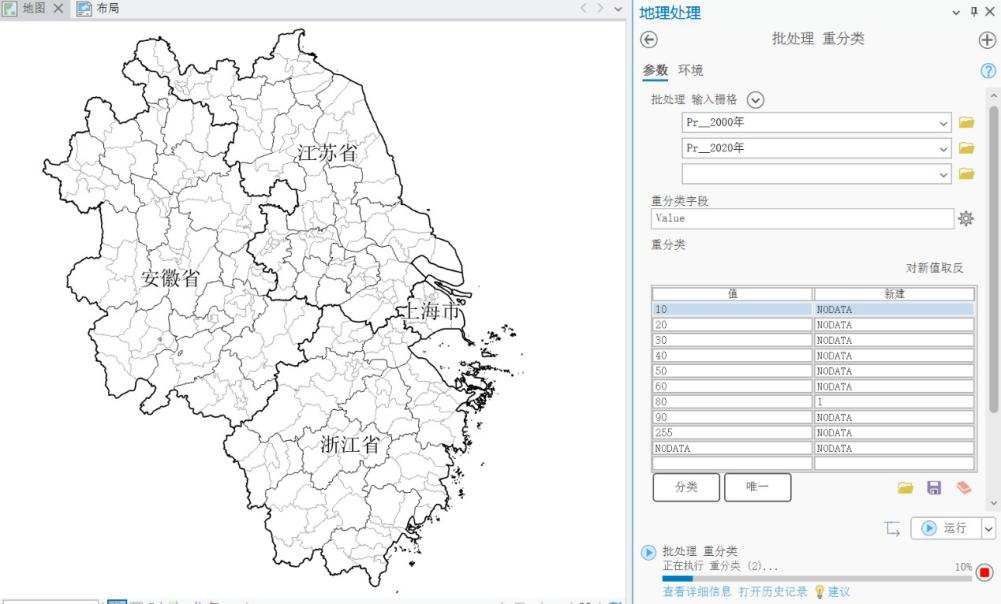
基于叠加结果，统计不同土地利用类型间的转换面积，构建土地利用转移矩阵。

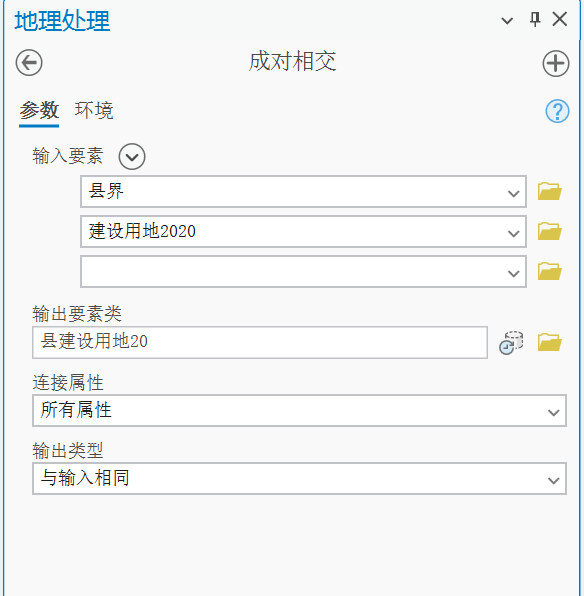


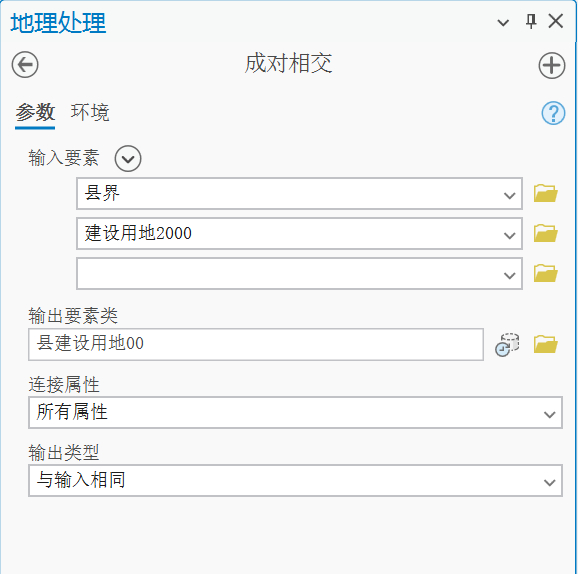
计算各类土地利用类型的面积变化量、变化率及动态度，量化时间维度变化特征。

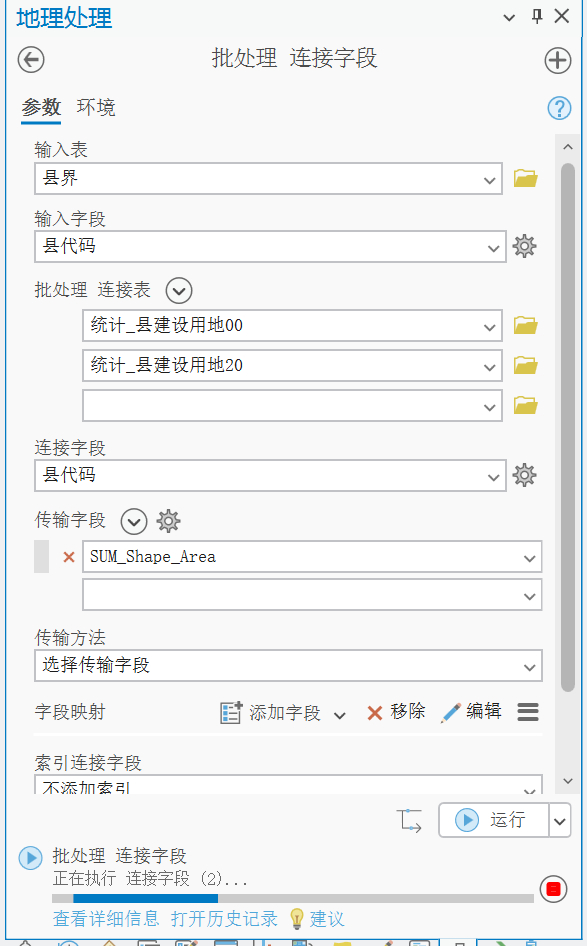
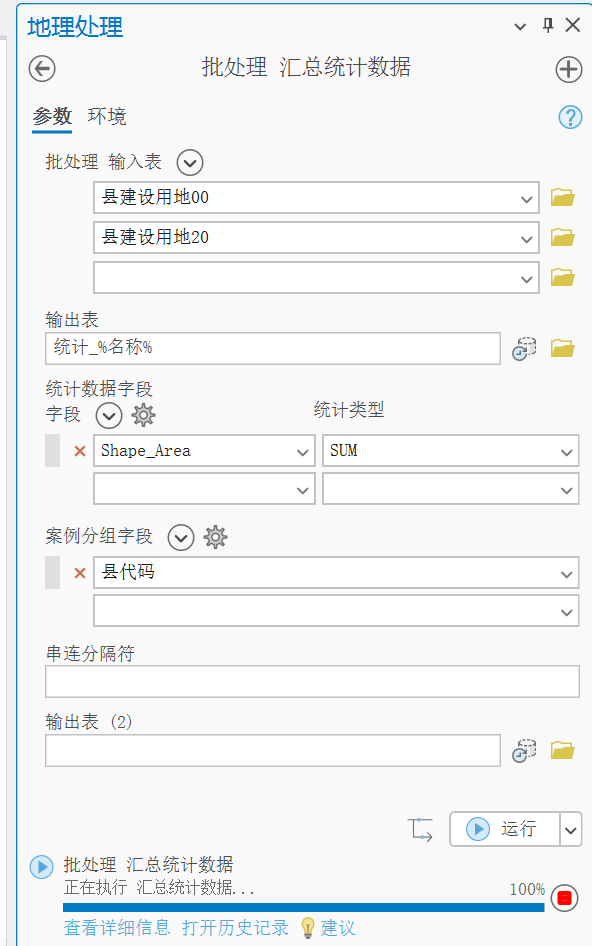
3.5建设用地数据处理

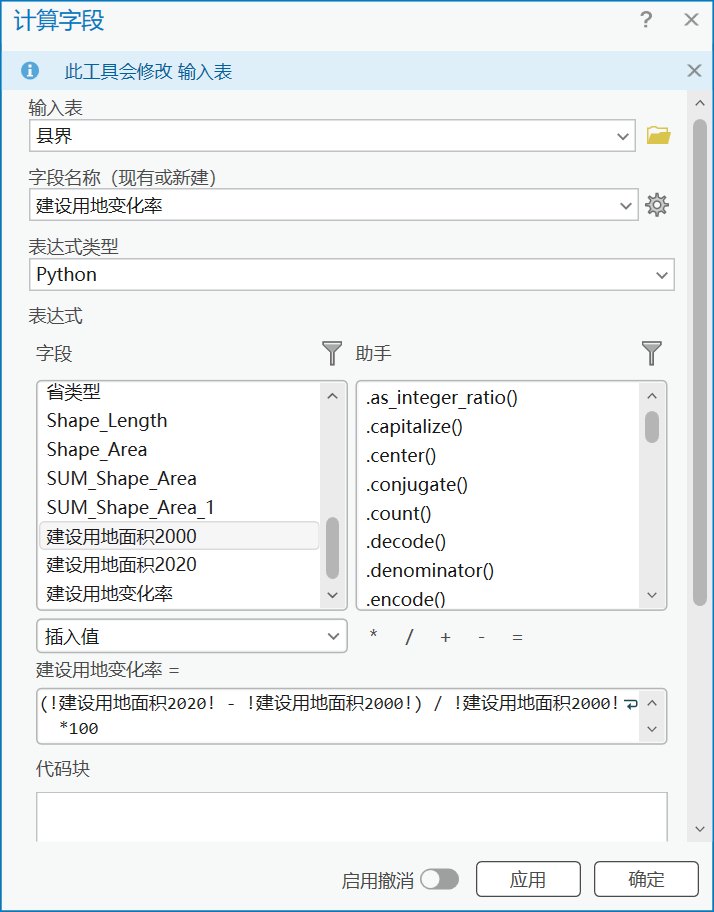
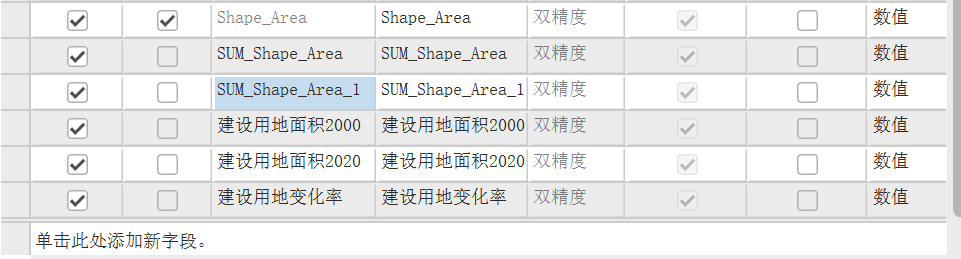
重分类提取建设用地。



转矢量与县界相交然后汇总。

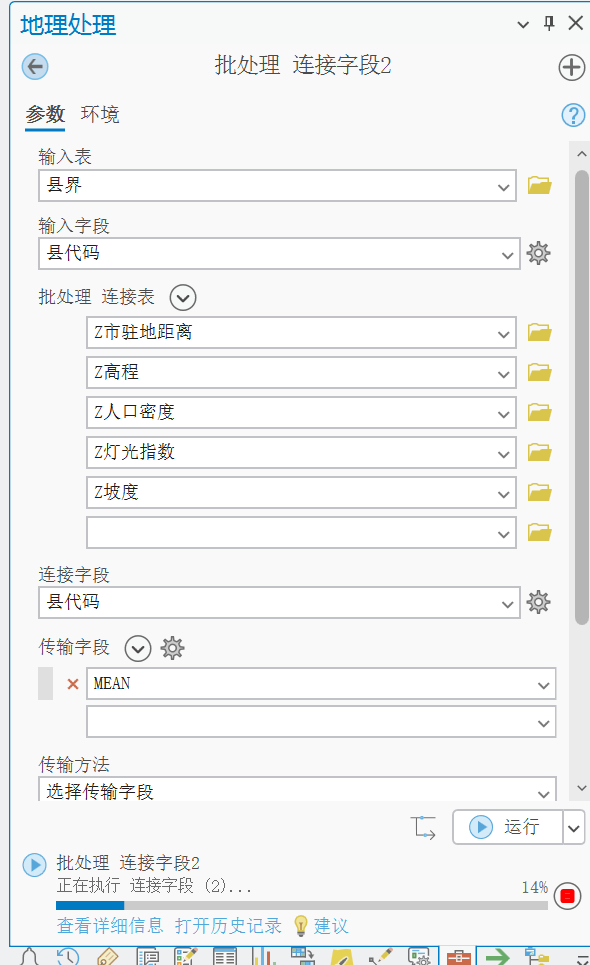
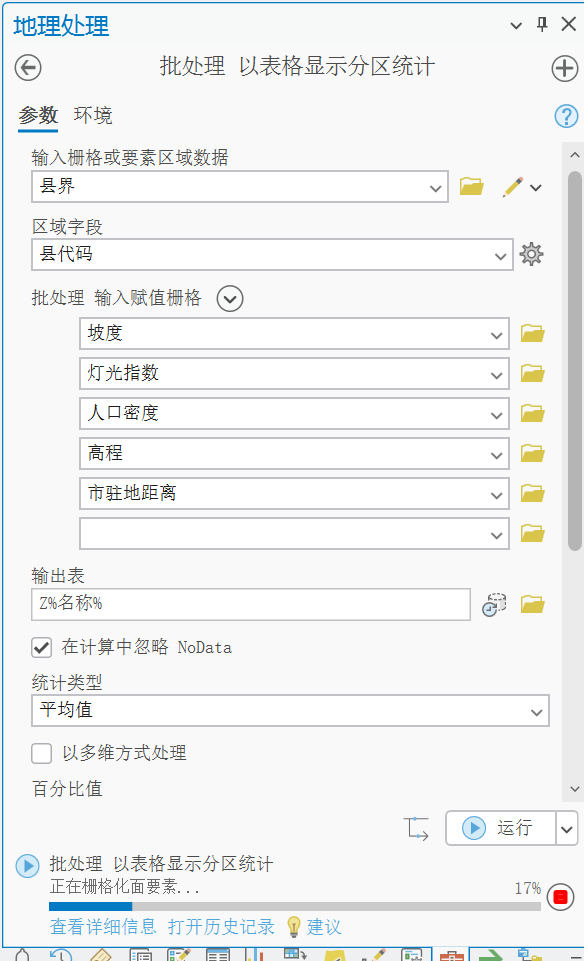






3.6驱动因子数据处理

选取人口密度、夜间灯光指数、距城市中心距离、平均高程以及平均坡度这5个核心驱动因子。对数据进行以表格显示分区统计，然后连接到属性表内。



3.7 地理探测器模型运算

将标准化后的驱动因子栅格数据与建设用地变化栅格数据导入地理探测器模型。设置探测参数，开展因子探测，计算各驱动因子的q统计量，量化单因子对建设用地扩张的解释力。进行交互探测，分析不同驱动因子间的交互作用类型及综合解释力。

****四、研究结果与分析****

4.1土地利用总体变化特征

4.1.1面积变化分析

根据土地利用分类结果统计，研究期初至期末，长三角地区土地利用类型面积变化显著：

建设用地面积持续增加，共增加22242.66km²，变化率为4.358%，是增幅最大的土地利用类型。

耕地面积由19.7万km²缩减至17.5万km²，总减幅约为11.17%。经土地利用动态度模型计算得出，耕地的年均动态度为-0.552%。

这表明在2000-2020 年间，长三角地区的耕地以年均0.5%以上的速度持续向其他地类（主要是建设用地）转换。

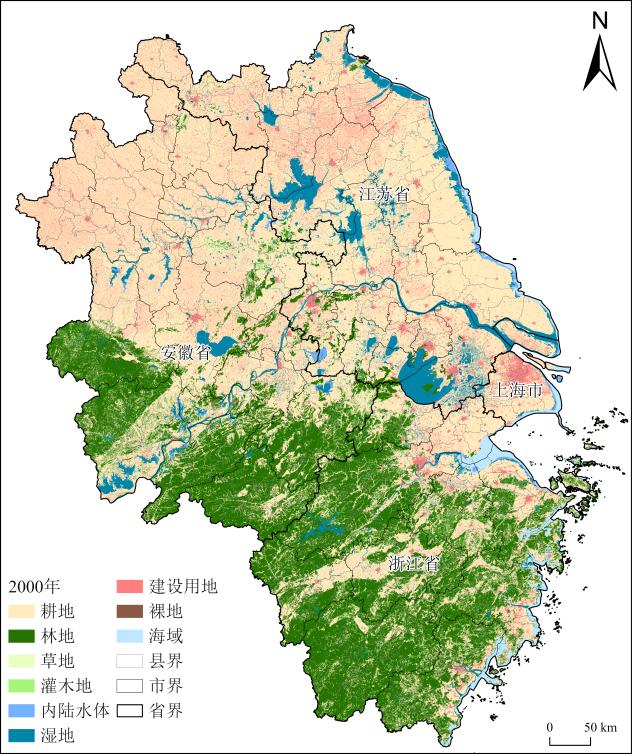
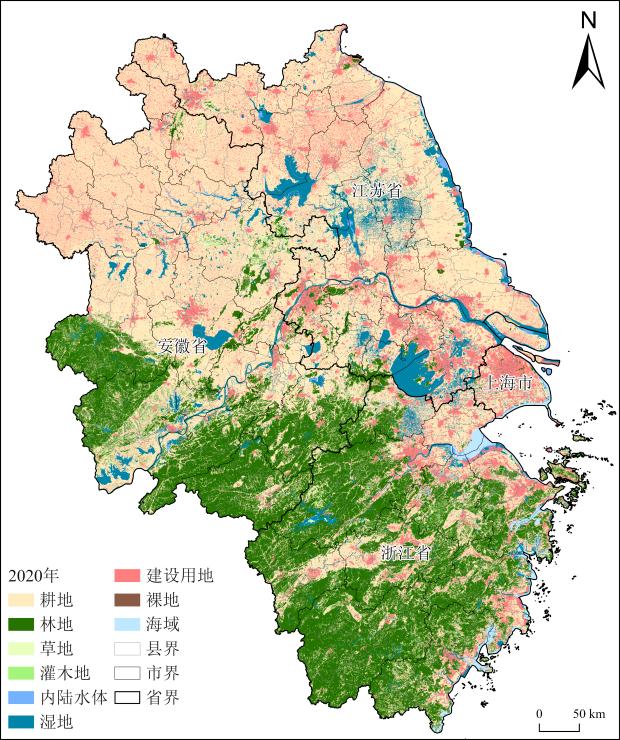
林地面积略有减少，减少731.2km²，变化率为-0.038%；水域面积基本保持稳定，变化率仅为0.519%。

| **地类** | **2000(单位：km²)** | **2020(单位：km²)** | **土地利用变化动态度（%）** |
| --- | --- | --- | --- |
| 耕地 | 197434.85 | 175623.94 | -0.552 |
| 林地 | 96568.4 | 95837.2 | -0.038 |
| 草地 | 7510.06 | 7637.28 | 0.085 |
| 灌木地 | 284.22 | 252.44 | -0.559 |
| 湿地 | 2257.73 | 1857.42 | -0.887 |
| 内陆水体 | 23418.87 | 25851.23 | 0.519 |
| 建设用地 | 25518.98 | 47761.64 | 4.358 |
| 裸地 | 50.37 | 57.81 | 0.738 |
| 海域 | 5937.53 | 4102.06 | -1.546 |
| **总计** | **358981.01** | **358981.01** | **0** |

4.1.2空间格局变化分析

建设用地扩张呈现明显的“中心集聚、廊道延伸”特征：

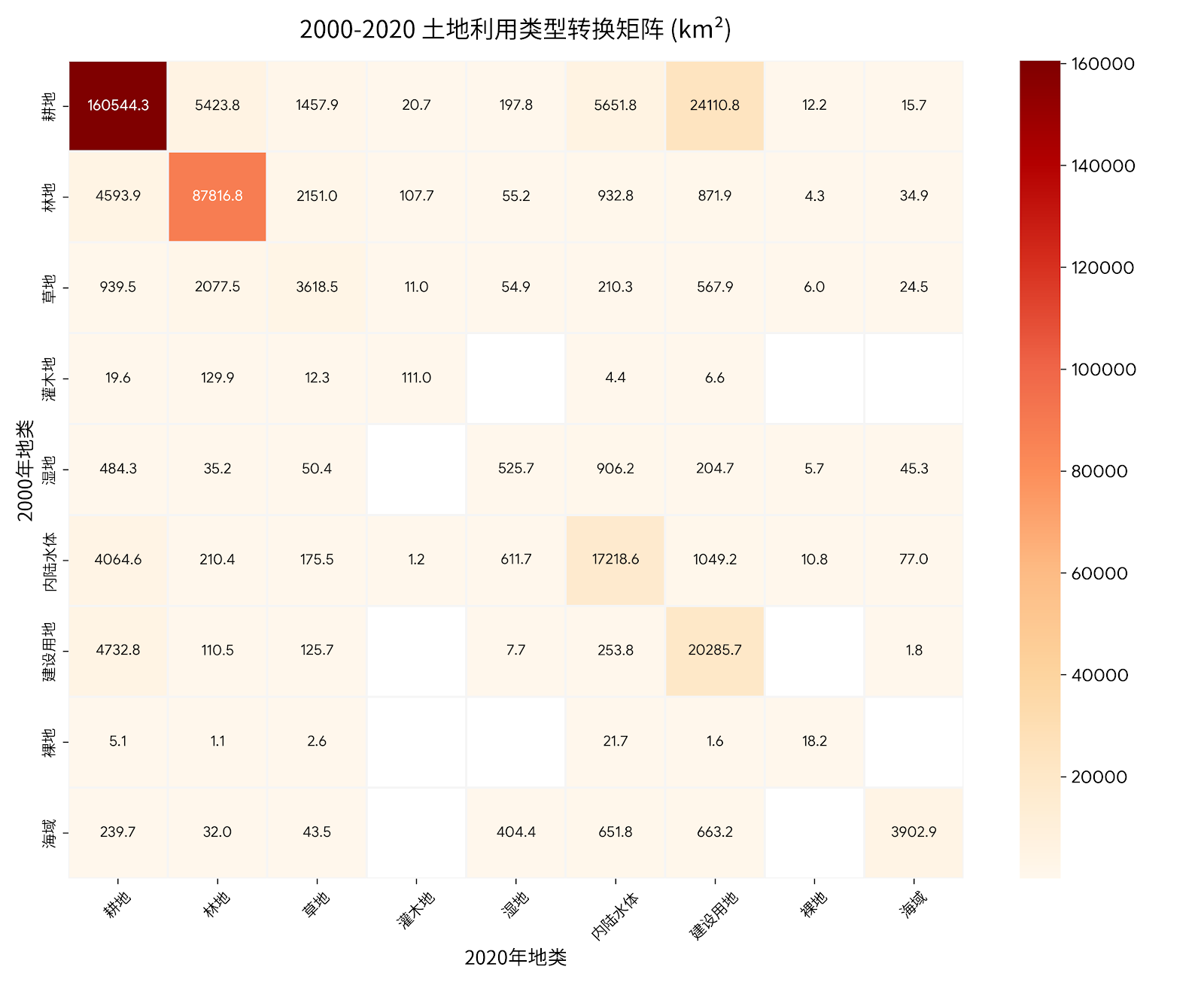
1. 上海、南京、杭州、苏州等核心城市建成区持续向外围扩张，形成连片式建设用地格局。
2. 沿交通干线（如京沪高铁、沪昆高速等）的建设用地呈现廊道式延伸，形成“轴带式”扩张格局。
3. 耕地减少区域主要集中在核心城市外围及交通干线周边，与建设用地扩张区域高度重合。
4. 林地减少主要分布在丘陵边缘区域，多转换为耕地或建设用地。



4.2土地利用转移矩阵分析

土地利用转移矩阵清晰揭示了各类用地的转换关系，核心特征如下：

1. 建设用地的主要来源为耕地，研究时段内共有24110.79km²的耕地转换为建设用地，是建设用地扩张的最主要贡献地类，反映了城市化进程中耕地向建设用地的转换主导模式。
2. 同时，有871.89km²的林地、5.96km²的草地、0.26km²的灌木地、5.75km²的湿地、10.85km²的内陆水体、0.27km²的裸地及1.83km²的海域也转换为建设用地，共同构成了建设用地的增量来源。
3. 耕地除转换为建设用地外，另有5423.8km²转换为林地、5651.76km²转换为内陆水体，反映了生态修复与水域保护等工程对耕地流转的引导作用。
4. 此外，2000–2020年建设用地自身保留面积为20285.73km²，体现了建设用地的较强稳定性；内陆水体也有1049.21km²转换为建设用地，反映了部分水域空间的开发利用。

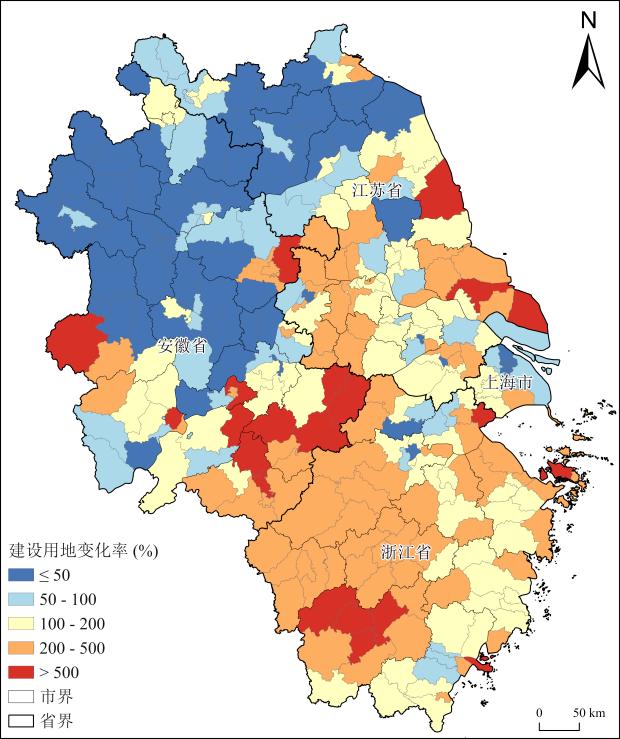


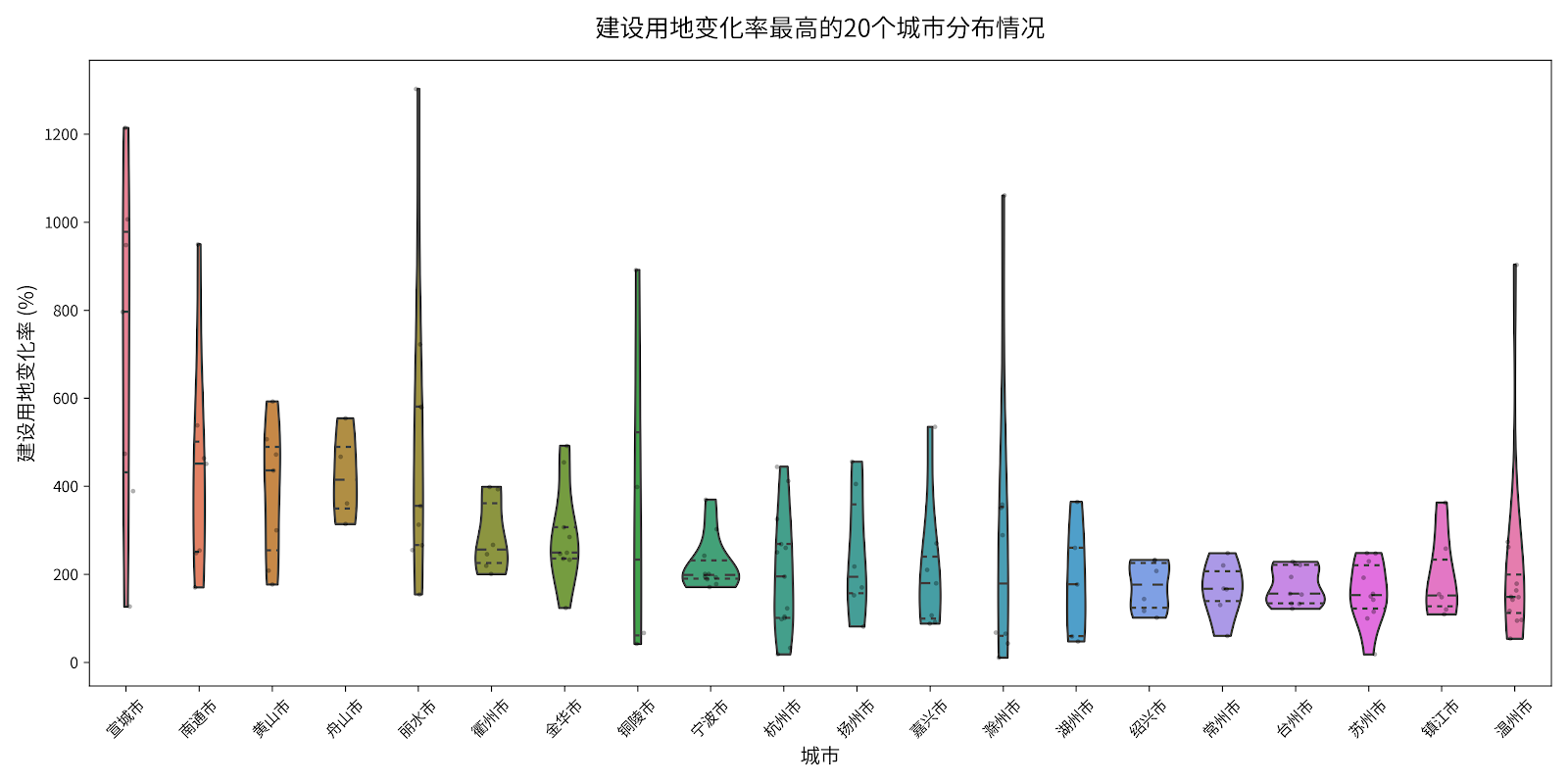
|  |  | 2020 |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 地类 | 耕地 | 林地 | 草地 | 灌木地 | 湿地 | 内陆水体 | 建设用地 | 裸地 | 海域 |
| 2000 | 耕地 | 160544.26 | 5423.8 | 1457.86 | 20.66 | 197.81 | 5651.76 | 24110.79 | 12.25 | 15.67 |
|  | 林地 | 4593.95 | 87816.77 | 2150.95 | 107.69 | 55.19 | 932.78 | 871.89 | 4.31 | 34.87 |
|  | 草地 | 939.55 | 2077.49 | 3618.51 | 10.98 | 54.86 | 210.26 | 567.92 | 5.96 | 24.52 |
|  | 灌木地 | 19.63 | 129.91 | 12.34 | 111.04 |  | 4.44 | 6.6 | 0.26 |  |
|  | 湿地 | 484.25 | 35.23 | 50.35 | 0.21 | 525.72 | 906.19 | 204.71 | 5.75 | 45.33 |
|  | 内陆水体 | 4064.64 | 210.36 | 175.46 | 1.16 | 611.67 | 17218.57 | 1049.21 | 10.85 | 76.95 |
|  | 建设用地 | 4732.82 | 110.48 | 125.66 | 0.69 | 7.74 | 253.76 | 20285.73 | 0.27 | 1.83 |
|  | 裸地 | 5.13 | 1.13 | 2.63 |  | 0.01 | 21.72 | 1.6 | 18.16 |  |
|  | 海域 | 239.72 | 32.04 | 43.52 |  | 404.42 | 651.76 | 663.18 |  | 3902.89 |

4.3建设用地变化的空间差异

从建设用地变化率的空间分布来看，核心城市圈（上海-苏州-无锡、南京-镇江、杭州-嘉兴-绍兴）是建设用地扩张的高值区，呈现连片扩张态势。这些区域对应地图中建设用地变化率较高的橙红色至红色区间，反映出城镇密集地区的建设用地扩张强度与集聚效应更为显著。

皖北、苏北等外围区域建设用地扩张强度相对较低，对应地图中以蓝色、浅黄色为主的区间，呈现以县城及重点镇为中心的点状扩张特征，区域间扩张的连通性较弱。

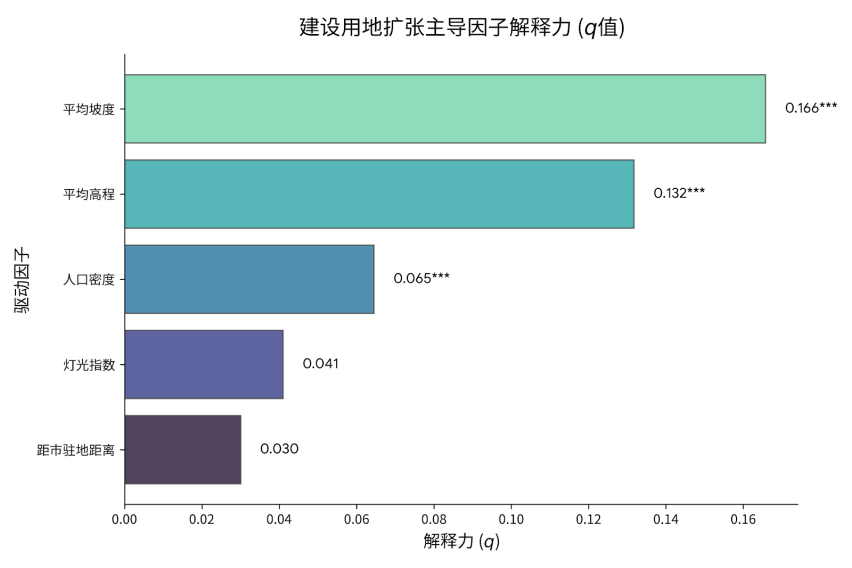
沿长江、京沪高铁、沪昆高铁等交通干线的建设用地扩张特征明显，形成“廊道式”扩张带。这些沿线区域的建设用地变化率较高（多为橙色至红色区间），交通干线的引导作用推动了沿线城镇的联动扩张，强化了区域间的空间关联。



4.4建设用地驱动性分析结果

4.4.1单因子探测结果

地理探测器因子探测结果显示，各驱动因子对建设用地扩张的解释力（q值）排序为：

平均坡度（q=0.1658）>平均高程（q=0.1317）>人口密度（q=0.0645）>灯光指数（q=0.041）>距市驻地距离（q=0.03）

平均坡度与平均高程的q值相对更高，且p值均小于0.01，表明地形坡度与高程条件对建设用地扩张具有显著的解释力，是影响扩张的重要地形因子。

人口密度的q值为0.0645，p值小于0.01，说明人口集聚对建设用地扩张存在显著的驱动作用，人口分布的空间差异会影响建设用地的扩张进程。

灯光指数的q值为0.041，p值显著（p≈0.0181<0.05），反映夜间灯光所代表的人类活动强度对建设用地扩张有一定解释力，人类活动的空间差异会推动建设用地扩张。

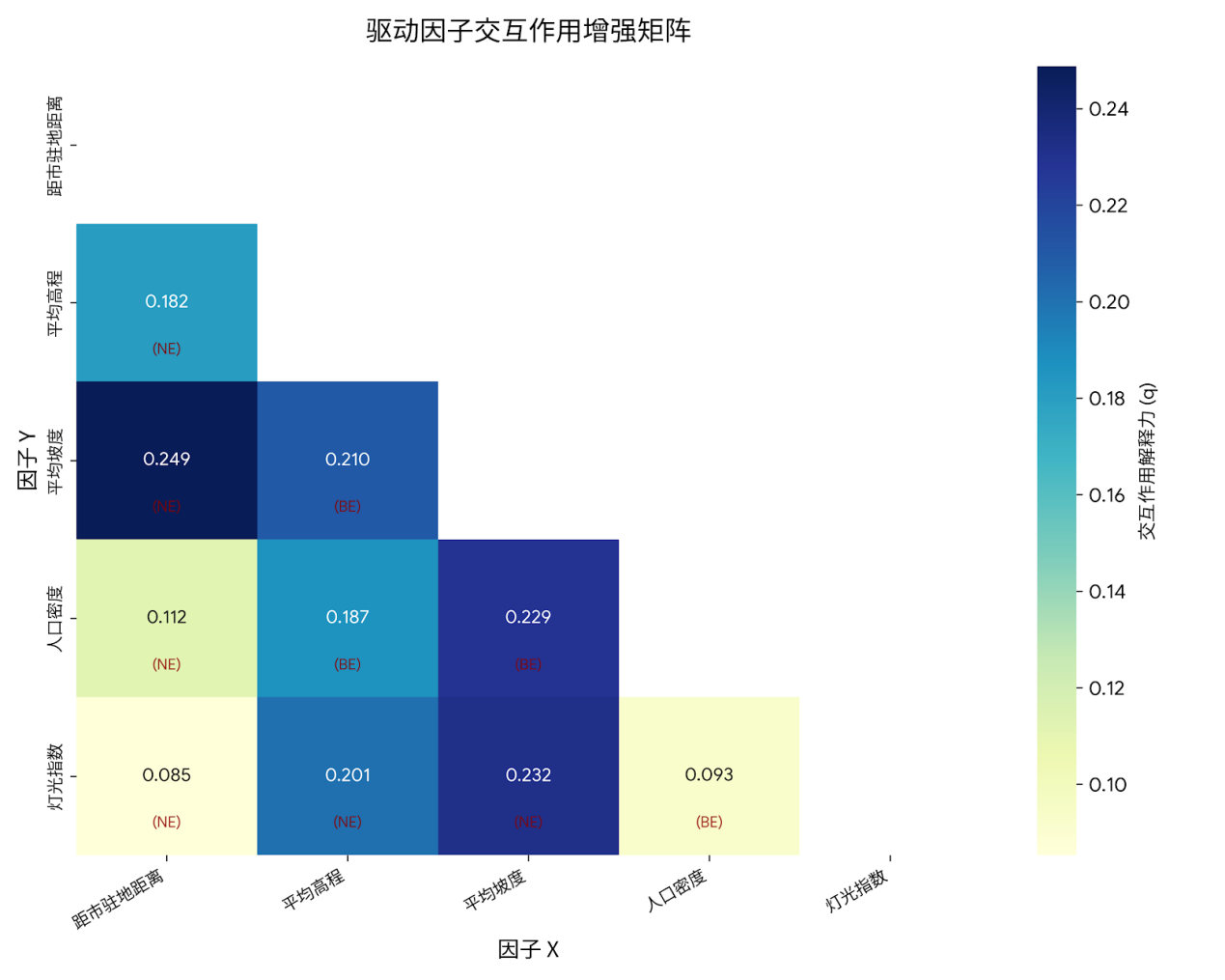
距市驻地距离的q值最低（0.03），且p值未通过0.05水平的显著性检验（p≈0.0849>0.05），表明其对建设用地扩张的解释力较弱且不够显著。

| **Variable** | **q** | **p-value** | **number\_of\_strata** | **CI\_90** | **CI\_95** | **CI\_99** |
| --- | --- | --- | --- | --- | --- | --- |
| 距市驻地距离 | 0.03 | 0.08487746283221809 | 5 | 0.0000-0.0568 | 0.0000-0.0654 | 0.0000-0.0834 |
| **平均高程** | **0.1317** | **<0.01** | **5** | **0.0695-0.1823** | **0.0606-0.1943** | **0.0446-0.2181** |
| **平均坡度** | **0.1658** | **<0.01** | **5** | **0.0983-0.2195** | **0.0881-0.2319** | **0.0694-0.2561** |
| 人口密度 | 0.0645 | <0.01 | 5 | 0.0189-0.1033 | 0.0139-0.1137 | 0.0059-0.1349 |
| 灯光指数 | 0.041 | 0.018087416073086282 | 5 | 0.0049-0.0725 | 0.0019-0.0818 | 0.0000-0.1010 |

4.4.2交互探测结果

交互探测结果表明，各驱动因子间的交互作用均为“双因子增强”或“非线性增强”，无独立作用或减弱作用：

距市驻地距离分别与平均高程、平均坡度、人口密度、灯光指数的交互作用均为非线性增强型，q值依次为0.1816、0.2488、0.112、0.0853，其中距市驻地距离与平均坡度的交互q值最高，表明地形坡度条件与距市驻地距离的协同作用，对建设用地扩张的非线性驱动效应最为突出。

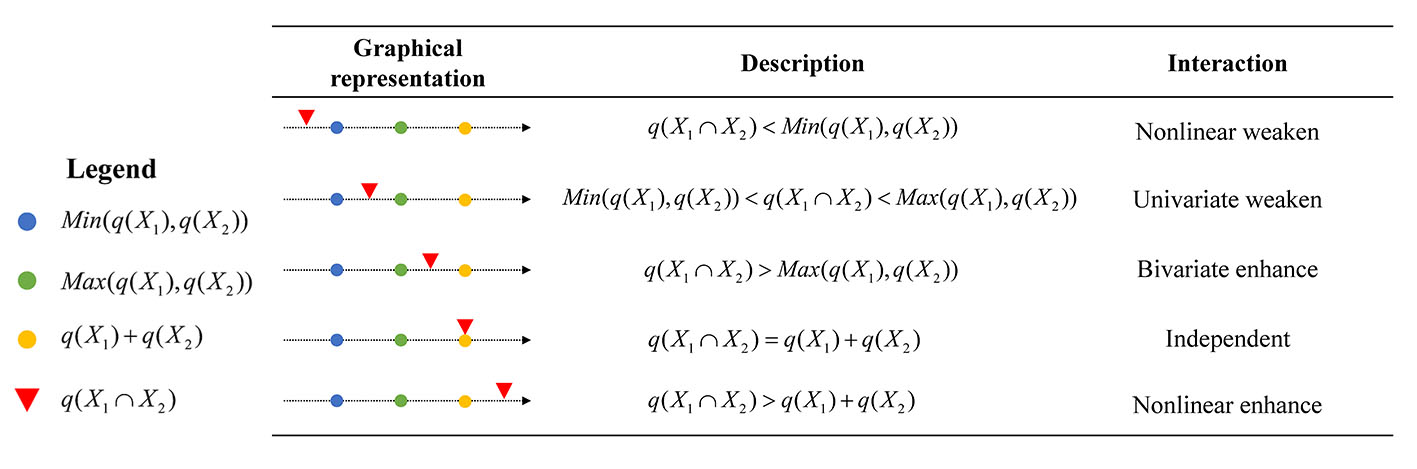


平均高程与平均坡度、人口密度的交互作用为双变量增强型，q值分别为0.2101、0.1866，说明地形条件组合、高程与人口集聚的叠加作用，对建设用地扩张的驱动效应强于单一因子；平均高程与灯光指数的交互作用为非线性增强型（q=0.2009），反映高程条件与人类活动强度的耦合关系更复杂，二者协同会以非线性方式放大对建设用地扩张的驱动作用。

平均坡度与人口密度的交互作用为双变量增强型（q=0.2288），表明坡度条件与人口集聚的协同作用显著增强了对建设用地扩张的驱动；平均坡度与灯光指数的交互作用为非线性增强型（q=0.2322），体现坡度与人类活动强度的耦合存在非线性放大效应。

人口密度与灯光指数的交互作用为双变量增强型（q=0.0929），说明人口集聚与夜间灯光反映的人类活动强度的协同作用，进一步推动了建设用地扩张。

|  |  |  |
| --- | --- | --- |
| VariablesInteraction  变量交互作用 | Inter\_q\_value  交互作用q值 | Interaction\_Type  交互作用类型 |
| 距市驻地距离&平均高程 | 0.1816 | Nonlinear\_enhance  (非线性增强型) |
| 距市驻地距离&平均坡度 | 0.2488 | Nonlinear\_enhance |
| 距市驻地距离&人口密度 | 0.112 | Nonlinear\_enhance |
| 距市驻地距离&灯光指数 | 0.0853 | Nonlinear\_enhance |
| 平均高程&平均坡度 | 0.2101 | Bivariate\_enhance  (双变量增强型) |
| 平均高程&人口密度 | 0.1866 | Bivariate\_enhance |
| 平均高程&灯光指数 | 0.2009 | Nonlinear\_enhance |
| 平均坡度&人口密度 | 0.2288 | Bivariate\_enhance |
| 平均坡度&灯光指数 | 0.2322 | Nonlinear\_enhance |
| 人口密度&灯光指数 | 0.0929 | Bivariate\_enhance |



****五、结论与建议****

****5.1结论****

2000-2020年，长三角地区土地利用呈现显著城乡转型特征，建设用地以4.358%的增幅持续扩张，核心来源为耕地（24110.79km²），空间上形成“中心集聚、廊道延伸”格局。耕地大幅减少且与建设用地扩张区域高度重合，林地、湿地等略有缩减，内陆水体基本稳定。驱动机制上，地形因子（平均坡度、高程）对建设用地扩张解释力最强，人口集聚、人类活动强度（灯光指数）为重要驱动，且各因子间以“双因子增强”“非线性增强”交互，其中距市驻地距离与平均坡度的非线性协同效应最为突出，共同塑造了区域土地利用演变格局。

****5.2建议****

长三角地区的地方政府应沿着交通干线和核心城市外围，划好耕地保护红线，避免优质耕地被随意改成建设用地。建房修路要顺着地形来，平坦的地方合理规划，坡度大、海拔高的区域少搞大规模建设，别过度开发丘陵边缘的林地。

针对上海、苏州等建设用地扩张快的地方，要统筹人口、产业和建设规模，避免盲目扩张；皖北、苏北等外围区域，重点围绕县城适度发展，同时做好湿地、林地的生态修复。政策制定时多考虑地形和城市距离的影响，让建设和保护更贴合实际需求。

****六、致谢****

首先我得感谢本校本院的何青松老师，他提醒我们“上机数据有点大”“学院楼网络很糟糕”“可以带个硬盘”，所以我每节课都带上自己的硬盘。

何青松老师特别亲切随和、耐心细致，每节课都带着我们操作。但由于自身的硬件短板和注意力不集中，我有时会跟不上大家的进度。所以在后两次上机课，我又带了个手机支架，专门录视频，录下老师的操作过程。

我还得感谢江苏苏州的同学邹予嘉、湖南郴州的同学朱念龙，在得知我的电脑死在了安装ArcGIS那一步，他们想方设法、不留余力地尝试帮我解决问题。

我还得感谢江苏南通的同学蒋鹏伟，他总能录下老师的关键步骤，并帮助我跟上进度。

我还得感谢西藏琼结的藏族同胞琼桑，他简直是少数民族中的大师，他对各大品牌、各大平台的数码产品了如指掌。在最后一次上机课，琼桑大师将他的数据线借给我，帮助我顺利完成了“老师的操作.mp4”在各个设备间的有线传输。

值得一提的是，2.2中提到的四项技术，我只会数据预处理、土地利用变化分析，因为那就是“老师的操作.mp4”里面讲到的。但由于自身的硬件短板，具体到本实验的很多操作，还是很让人头疼。所幸，我及时寻求了建规学院一位校友的帮助。他学过GIS，精通计算机，觉得指导我“很没意思”，就建议“上点难度”，加上了“驱动因子选取与数据标准化”“驱动性分析”，源码是用Python写的，因为我只系统学过Python这一门编程语言。

总的来说，“GIS城市大数据”这门课程给我提供了很多水论文的研究方法和手段，我特别喜欢“设计地图”“渲染地图”这个环节，我很害怕“搜集数据”“检查错误”“回溯操作”。

但是，喜欢的东西如果很少派上用场，就会从记忆中淡漠；害怕的东西如果总是得到克服，就会成为自己的强项。这可能就是这门课程带给我的最珍贵的感受吧。

****【附录Python数据分析源码】****

****【附1】****

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
import matplotlib as mpl  
  
# 设置中文字体  
plt.rcParams['font.family'] = ['Noto Sans CJK JP']  
plt.rcParams['axes.unicode\_minus'] = False # 解决负号显示问题  
  
# 1. 因子探测器 (Factor Detector) - 高级学术风格 Lollipop Chart  
factor\_df = pd.read\_csv('探测因子报表.xlsx - Factor detector.csv')  
factor\_df = factor\_df.sort\_values('q', ascending=True)  
  
plt.figure(figsize=(10, 6))  
# 使用更学术的色调  
my\_color = "#34495e"  
plt.hlines(y=factor\_df['Variable'], xmin=0, xmax=factor\_df['q'], color='lightgrey', linewidth=1)  
plt.scatter(factor\_df['q'], factor\_df['Variable'], s=150, color="#e74c3c", edgecolors=my\_color, zorder=3)  
  
# 在点旁边标注 q 值  
for i, q in enumerate(factor\_df['q']):  
 plt.text(q + 0.005, i, f'{q:.4f}', va='center', fontsize=11, fontweight='bold', color=my\_color)  
  
plt.title('建设用地扩张驱动因子影响力排名 ($q$ 值)', fontsize=16, pad=20)  
plt.xlabel('解释力 ($q$)', fontsize=12)  
plt.ylabel('驱动因子', fontsize=12)  
plt.xlim(0, factor\_df['q'].max() \* 1.2)  
plt.grid(axis='x', linestyle='--', alpha=0.5)  
plt.gca().spines['top'].set\_visible(False)  
plt.gca().spines['right'].set\_visible(False)  
plt.tight\_layout()  
plt.savefig('advanced\_factor\_importance.png', dpi=300)  
plt.close()  
  
# 2. 交互探测器 (Interaction Detector) - 高级学术热力图  
# 读取交互数据  
inter\_raw = pd.read\_csv('探测因子报表.xlsx - Interaction detector.csv')  
cols = inter\_raw.iloc[0, 1:6].values.tolist()  
matrix\_data = inter\_raw.iloc[1:6, 1:6].values.astype(float)  
inter\_matrix = pd.DataFrame(matrix\_data, index=cols, columns=cols)  
  
# 获取单因子 q 值用于判断交互类型  
q\_dict = dict(zip(factor\_df['Variable'], factor\_df['q']))  
  
# 构造交互类型标注  
# 定义逻辑：Nonlinear-enhance (NE), Bi-enhance (BE)  
annot\_matrix = np.full(inter\_matrix.shape, "", dtype=object)  
for i in range(len(cols)):  
 for j in range(i):  
 q1 = q\_dict[cols[i]]  
 q2 = q\_dict[cols[j]]  
 q\_inter = inter\_matrix.iloc[i, j]  
 if q\_inter > (q1 + q2):  
 type\_str = f"{q\_inter:.3f}\n(NE)" # Nonlinear Enhance  
 elif q\_inter > max(q1, q2):  
 type\_str = f"{q\_inter:.3f}\n(BE)" # Bi-enhance  
 else:  
 type\_str = f"{q\_inter:.3f}"  
 annot\_matrix[i, j] = type\_str  
  
plt.figure(figsize=(12, 10))  
mask = np.triu(np.ones\_like(inter\_matrix, dtype=bool))  
sns.heatmap(inter\_matrix, mask=mask, annot=annot\_matrix, fmt="", cmap="YlGnBu",   
 square=True, cbar\_kws={"shrink": .8, "label": "交互 $q$ 值"},   
 linewidths=1, linecolor='white', annot\_kws={"size": 10})  
  
plt.title('因子交互作用强度及类型 (NE: 非线性增强, BE: 双因子增强)', fontsize=16, pad=20)  
plt.xticks(rotation=45, ha='right')  
plt.yticks(rotation=0)  
plt.tight\_layout()  
plt.savefig('advanced\_interaction\_heatmap.png', dpi=300)  
plt.close()  
  
# 3. 土地利用转移矩阵 (Land Use Transfer) - 细致热力图  
land\_use\_raw = pd.read\_csv('土地利用报表.xlsx - Sheet1.csv')  
transfer\_data = land\_use\_raw.iloc[13:22, 2:11].values.astype(float)  
categories = land\_use\_raw.iloc[12, 2:11].values.tolist()  
transfer\_df = pd.DataFrame(transfer\_data, index=categories, columns=categories)  
  
# 计算百分比转移矩阵  
transfer\_percent = transfer\_df.div(transfer\_df.sum(axis=1), axis=0) \* 100  
  
plt.figure(figsize=(12, 10))  
sns.heatmap(transfer\_df, annot=True, fmt=".1f", cmap="Reds", cbar\_kws={'label': '转移面积 (km²)'})  
# 覆盖一层数值，如果是重点转移（比如到建设用地），可以特别标注  
plt.title('2000-2020 长三角土地利用空间转移矩阵 (km²)', fontsize=16, pad=20)  
plt.xlabel('2020年地类', fontsize=12)  
plt.ylabel('2000年地类', fontsize=12)  
plt.tight\_layout()  
plt.savefig('advanced\_transfer\_matrix.png', dpi=300)  
plt.close()  
  
# 4. 城市建设用地变化率 - 雨靶图 (Raincloud-like plot: Box + Strip)  
construction\_df = pd.read\_csv('建设用地情况统计.xlsx - 县界.csv')  
city\_order = construction\_df.groupby('市')['建设用地变化率'].median().sort\_values(ascending=False).index  
  
plt.figure(figsize=(16, 8))  
# 绘制箱线图  
sns.boxplot(data=construction\_df, x='市', y='建设用地变化率', order=city\_order,   
 whis=[0, 100], width=.6, palette="vlag", linewidth=1)  
# 叠加散点（抖动）  
sns.stripplot(data=construction\_df, x='市', y='建设用地变化率', order=city\_order,   
 size=3, color=".3", linewidth=0, alpha=0.4)  
  
plt.title('长三角各市县域建设用地变化率分布 (学术风格)', fontsize=16, pad=20)  
plt.ylabel('建设用地变化率 (%)', fontsize=12)  
plt.xlabel('城市', fontsize=12)  
plt.xticks(rotation=90)  
plt.grid(axis='y', linestyle='--', alpha=0.3)  
plt.tight\_layout()  
plt.savefig('advanced\_construction\_variation.png', dpi=300)  
plt.close()

****【附2】****

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
  
# Set global academic style  
plt.rcParams['font.family'] = ['Noto Sans CJK JP']  
plt.rcParams['axes.unicode\_minus'] = False  
plt.rcParams['font.size'] = 12  
plt.rcParams['axes.labelpad'] = 10  
sns.set\_context("paper", font\_scale=1.2)  
plt.style.use('seaborn-v0\_8-white')  
  
# 1. Advanced Factor Importance (Academic Style)  
factor\_df = pd.read\_csv('探测因子报表.xlsx - Factor detector.csv')  
factor\_df = factor\_df.sort\_values('q', ascending=True)  
  
fig, ax = plt.subplots(figsize=(8, 6))  
colors = sns.color\_palette("Blues\_d", n\_colors=len(factor\_df))  
ax.barh(factor\_df['Variable'], factor\_df['q'], color=colors, height=0.6, edgecolor='black', linewidth=0.8)  
  
# Add p-value stars (all seem < 0.01 or ~0.08)  
for i, (q, p) in enumerate(zip(factor\_df['q'], factor\_df['p-value'])):  
 star = ""  
 try:  
 p\_val = float(p) if '<' not in str(p) else 0.0  
 except:  
 p\_val = 1.0  
 if p\_val < 0.01 or '<0.01' in str(p): star = "\*\*\*"  
 elif p\_val < 0.05: star = "\*\*"  
 elif p\_val < 0.1: star = "\*"  
 ax.text(q + 0.002, i, f"{q:.3f}{star}", va='center', fontweight='bold')  
  
ax.set\_title('建设用地扩张驱动因子探测结果 ($q$值)', fontsize=16, pad=20)  
ax.set\_xlabel('解释力 ($q$)', fontsize=13)  
ax.spines['right'].set\_visible(False)  
ax.spines['top'].set\_visible(False)  
plt.tight\_layout()  
plt.savefig('academic\_factor\_detector.png', dpi=300)  
plt.close()  
  
# 2. Advanced Interaction Detector Matrix  
# Load Interaction Data  
inter\_raw = pd.read\_csv('探测因子报表.xlsx - Interaction detector.csv')  
headers = inter\_raw.iloc[0, 1:6].values.tolist()  
matrix\_data = inter\_raw.iloc[1:6, 1:6].values.astype(float)  
inter\_matrix = pd.DataFrame(matrix\_data, index=headers, columns=headers)  
  
# Get individual q values for comparison  
q\_vals = factor\_df.set\_index('Variable')['q'].to\_dict()  
  
# Calculate Interaction Types  
# types: 0: Independent, 1: Weaken, 2: Uni-weaken, 3: Bi-enhance, 4: Nonlinear-enhance  
type\_matrix = np.zeros\_like(matrix\_data, dtype=object)  
for i in range(len(headers)):  
 for j in range(len(headers)):  
 if i > j:  
 q12 = matrix\_data[i, j]  
 q1 = q\_vals.get(headers[i], 0)  
 q2 = q\_vals.get(headers[j], 0)  
 if q12 > (q1 + q2):  
 type\_matrix[i, j] = 'NE' # Nonlinear Enhance  
 elif q12 > max(q1, q2):  
 type\_matrix[i, j] = 'BE' # Bi-variable Enhance  
 else:  
 type\_matrix[i, j] = 'E'  
  
fig, ax = plt.subplots(figsize=(10, 8))  
mask = np.triu(np.ones\_like(inter\_matrix, dtype=bool))  
sns.heatmap(inter\_matrix, annot=True, mask=mask, cmap='YlGnBu', fmt=".3f",   
 linewidths=1, linecolor='white', cbar\_kws={'label': '交互$q$值'},  
 annot\_kws={"size": 12, "weight": "bold"})  
  
# Add Type Labels (NE/BE) in the cells  
for i in range(len(headers)):  
 for j in range(len(headers)):  
 if i > j:  
 label = type\_matrix[i, j]  
 ax.text(j + 0.5, i + 0.8, f"({label})", ha='center', va='center', fontsize=10, color='darkred')  
  
ax.set\_title('驱动因子交互作用探测矩阵 (NE: 非线性增强, BE: 双因子增强)', fontsize=16, pad=20)  
plt.xticks(rotation=45, ha='right')  
plt.yticks(rotation=0)  
plt.tight\_layout()  
plt.savefig('advanced\_interaction\_matrix.png', dpi=300)  
plt.close()  
  
# 3. Land Use Transfer Matrix (Polished Heatmap)  
land\_use\_raw = pd.read\_csv('土地利用报表.xlsx - Sheet1.csv')  
transfer\_data = land\_use\_raw.iloc[13:22, 2:11].values.astype(float)  
categories = land\_use\_raw.iloc[12, 2:11].values.tolist()  
transfer\_df = pd.DataFrame(transfer\_data, index=categories, columns=categories)  
  
# Advanced Heatmap with log scale color for better visibility of small transfers  
from matplotlib.colors import LogNorm  
plt.figure(figsize=(12, 10))  
# Mask zeros or very small values for cleaner look  
annot\_data = transfer\_df.applymap(lambda x: f"{x:.1f}" if x > 10 else "")  
  
sns.heatmap(transfer\_df, annot=annot\_data.values, fmt="", cmap="Reds",   
 linewidths=0.5, linecolor='#eeeeee', cbar\_kws={'label': '转移面积 (km²)'},  
 norm=LogNorm(vmin=1, vmax=transfer\_df.values.max()))  
  
plt.title('2000-2020 长三角土地利用空间转移矩阵 (km²)', fontsize=18, pad=25)  
plt.xlabel('2020年地类', fontsize=14, labelpad=15)  
plt.ylabel('2000年地类', fontsize=14, labelpad=15)  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.savefig('academic\_transfer\_matrix.png', dpi=300)  
plt.close()  
  
# 4. Construction Land Growth (Advanced Distribution - Raincloud Plot style)  
construction\_df = pd.read\_csv('建设用地情况统计.xlsx - 县界.csv')  
# Keep only Top 15 cities or aggregate  
top\_cities = construction\_df.groupby('市')['建设用地变化率'].median().sort\_values(ascending=False).index  
  
plt.figure(figsize=(15, 8))  
# Strip plot + Box plot  
sns.stripplot(data=construction\_df, x='市', y='建设用地变化率', order=top\_cities,   
 size=3, color="gray", alpha=0.4, jitter=0.2)  
sns.boxplot(data=construction\_df, x='市', y='建设用地变化率', order=top\_cities,   
 whis=[0, 100], width=.6, palette="vlag", boxprops=dict(alpha=.7))  
  
plt.title('长三角各市县域建设用地变化率分布特征', fontsize=18, pad=20)  
plt.ylabel('建设用地变化率 (%)', fontsize=14)  
plt.xlabel('城市', fontsize=14)  
plt.xticks(rotation=90)  
plt.grid(axis='y', linestyle='--', alpha=0.3)  
plt.tight\_layout()  
plt.savefig('construction\_growth\_distribution.png', dpi=300)  
plt.close()

****【附3】****

import matplotlib.font\_manager as fm  
import os  
  
# Find the specific font file for Noto Sans CJK JP  
font\_list = fm.findSystemFonts()  
cjk\_font\_path = [f for f in font\_list if 'NotoSansCJK' in f or 'NotoSansCJKJP' in f]  
print(f"Font paths found: {cjk\_font\_path}")  
  
# Set the font properties using the path  
if cjk\_font\_path:  
 my\_font = fm.FontProperties(fname=cjk\_font\_path[0])  
 print(f"Using font: {my\_font.get\_name()}")  
else:  
 print("No CJK font path found.")

****【附4】****

import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
import matplotlib.font\_manager as fm  
  
# Font Setup  
font\_path = '/usr/share/fonts/NotoSansCJK-Regular.ttc'  
prop = fm.FontProperties(fname=font\_path)  
plt.rcParams['axes.unicode\_minus'] = False  
  
# Helper for labels  
def set\_label\_font(ax, title=None, xlabel=None, ylabel=None):  
 if title: ax.set\_title(title, fontproperties=prop, fontsize=16, pad=15)  
 if xlabel: ax.set\_xlabel(xlabel, fontproperties=prop, fontsize=13)  
 if ylabel: ax.set\_ylabel(ylabel, fontproperties=prop, fontsize=13)  
 for label in ax.get\_xticklabels(): label.set\_fontproperties(prop)  
 for label in ax.get\_yticklabels(): label.set\_fontproperties(prop)  
  
# 1. Academic Factor Importance  
factor\_df = pd.read\_csv('探测因子报表.xlsx - Factor detector.csv').sort\_values('q', ascending=True)  
fig, ax = plt.subplots(figsize=(9, 6))  
colors = sns.color\_palette("mako", n\_colors=len(factor\_df))  
bars = ax.barh(factor\_df['Variable'], factor\_df['q'], color=colors, edgecolor='0.3', alpha=0.85)  
  
for i, (q, p) in enumerate(zip(factor\_df['q'], factor\_df['p-value'])):  
 sig = "\*\*\*" if ('<0.01' in str(p) or (isinstance(p, (int, float)) and p < 0.01)) else ("\*\*" if (isinstance(p, (int, float)) and p < 0.05) else "")  
 ax.text(q + 0.005, i, f"{q:.3f}{sig}", va='center', fontsize=11, fontweight='bold')  
  
set\_label\_font(ax, "建设用地扩张主导因子解释力 ($q$值)", "解释力 ($q$)", "驱动因子")  
ax.spines[['top', 'right']].set\_visible(False)  
plt.tight\_layout()  
plt.savefig('academic\_factor\_q.png', dpi=300)  
plt.close()  
  
# 2. Advanced Interaction Matrix (Heatmap)  
inter\_raw = pd.read\_csv('探测因子报表.xlsx - Interaction detector.csv')  
headers = inter\_raw.iloc[0, 1:6].values.tolist()  
matrix\_data = inter\_raw.iloc[1:6, 1:6].values.astype(float)  
q\_dict = factor\_df.set\_index('Variable')['q'].to\_dict()  
  
fig, ax = plt.subplots(figsize=(10, 8))  
mask = np.triu(np.ones\_like(matrix\_data, dtype=bool))  
sns.heatmap(matrix\_data, mask=mask, annot=True, fmt=".3f", cmap="YlGnBu",   
 xticklabels=headers, yticklabels=headers,  
 cbar\_kws={'label': '交互作用解释力 (q)'},  
 annot\_kws={"size": 11, "weight": "bold"})  
  
# Annotate Interaction Types  
for i in range(len(headers)):  
 for j in range(len(headers)):  
 if i > j:  
 q12 = matrix\_data[i, j]  
 q1, q2 = q\_dict[headers[i]], q\_dict[headers[j]]  
 label = "NE" if q12 > (q1 + q2) else "BE"  
 ax.text(j + 0.5, i + 0.8, f"({label})", ha='center', va='center', fontsize=9, color='darkred', alpha=0.8)  
  
set\_label\_font(ax, "驱动因子交互作用增强矩阵", "因子 X", "因子 Y")  
plt.xticks(rotation=30, ha='right')  
plt.tight\_layout()  
plt.savefig('advanced\_interaction\_heatmap.png', dpi=300)  
plt.close()  
  
# 3. Refined Transfer Matrix  
land\_use\_raw = pd.read\_csv('土地利用报表.xlsx - Sheet1.csv')  
trans\_data = land\_use\_raw.iloc[13:22, 2:11].values.astype(float)  
cats = land\_use\_raw.iloc[12, 2:11].values.tolist()  
  
plt.figure(figsize=(11, 9))  
# We use a threshold to simplify the display  
display\_data = np.where(trans\_data > 1.0, trans\_data, 0)  
sns.heatmap(display\_data, annot=True, fmt=".1f", cmap="OrRd",   
 xticklabels=cats, yticklabels=cats, mask=(display\_data == 0),  
 linewidths=1, linecolor='whitesmoke')  
ax = plt.gca()  
set\_label\_font(ax, "2000-2020 土地利用类型转换矩阵 (km²)", "2020年地类", "2000年地类")  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.savefig('advanced\_transfer\_matrix.png', dpi=300)  
plt.close()  
  
# 4. Multi-level City Growth Distribution  
const\_df = pd.read\_csv('建设用地情况统计.xlsx - 县界.csv')  
top\_cities = const\_df.groupby('市')['建设用地变化率'].median().sort\_values(ascending=False).head(20).index  
subset = const\_df[const\_df['市'].isin(top\_cities)]  
  
plt.figure(figsize=(14, 7))  
sns.violinplot(data=subset, x='市', y='建设用地变化率', order=top\_cities,   
 palette="husl", inner="quart", cut=0, alpha=0.6)  
sns.stripplot(data=subset, x='市', y='建设用地变化率', order=top\_cities,   
 color="black", size=3, alpha=0.3, jitter=True)  
ax = plt.gca()  
set\_label\_font(ax, "建设用地变化率最高的20个城市分布情况", "城市", "建设用地变化率 (%)")  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.savefig('city\_growth\_distribution.png', dpi=300)  
plt.close()