

GIS编程入门

开源地理空间工具的Python实用指南

吴秋生



GIS 编程入门

开源地理空间工具的 Python 实用指南

吴秋生

2025

Contents

序言	1
简介	3
本书的目标读者	3
本书涵盖的内容	3
充分利用本书	4
本书使用的约定	5
下载代码示例	5
视频教程	6
联系我们	6
致谢	6
关于作者	7
许可和版权	7
I: 软件设置	9
1. 软件工具概览	11
1.1. 简介	11
1.2. 学习目标	11
1.3. 基本软件工具	11
1.4. 工具集成和工作流程	13
1.5. 运行代码示例	14
1.6. 要点总结	14
2. Python 包管理入门	15
2.1. 简介	15
2.2. 学习目标	15
2.3. 安装 Conda (Miniconda)	16
2.4. 理解 Conda 概念	17
2.5. 创建您的第一个地理空间环境	18
2.6. Conda 故障排除	18
2.7. 基本 Conda 命令	19
2.8. 介绍 uv: 快速替代方案	23
2.9. 包管理最佳实践	25
2.10. 要点总结	26
2.11. 练习	26
3. 设置 Visual Studio Code	28
3.1. 简介	28
3.2. 学习目标	28
3.3. 安装 Visual Studio Code	28
3.4. Python 编程的基本扩展	29
3.5. 为 Python 开发配置 VS Code	31
3.6. 基本键盘快捷键	32
3.7. 参考资料和进一步学习	34
3.8. 要点总结	34
3.9. 练习	34
4. 使用 Git 进行版本控制	36

4.1. 简介	36
4.2. 学习目标	37
4.3. 设置 GitHub 账户	37
4.4. 安装 Git	37
4.5. 配置 Git	38
4.6. 理解 Git 概念	38
4.7. 基本 Git 命令	39
4.8. 使用 GitHub	42
4.9. 与 VS Code 集成	43
4.10. 地理空间项目的最佳实践	43
4.11. 要点总结	44
4.12. 练习	45
5. 使用 Google Colab	47
5.1. 简介	47
5.2. 学习目标	47
5.3. Google Colab 入门	47
5.4. 设置您的地理空间环境	48
5.5. Colab 的基本功能	49
5.6. 在 Colab 中运行代码示例	51
5.7. 要点总结	51
5.8. 练习	51
6. 使用 JupyterLab	53
6.1. 简介	53
6.2. 学习目标	53
6.3. 安装和设置 JupyterLab	53
6.4. JupyterLab 入门	54
6.5. 基本键盘快捷键	57
6.6. 在 MyBinder 上运行代码示例	59
6.7. 要点总结	59
6.8. 练习	59
7. 使用 Docker	61
7.1. 简介	61
7.2. 学习目标	61
7.3. 安装 Docker Desktop	61
7.4. 基本概念	62
7.5. 在 Docker 中运行代码示例	63
7.6. 常用 Docker 命令	64
7.7. 要点总结	65
7.8. 练习	66
II: Python 编程基础	69
8. 变量和数据类型	71
8.1. 简介	71
8.2. 学习目标	71
8.3. Python 中的变量	71
8.4. 变量命名	72

8.5. 数据类型	73
8.6. 转义字符	74
8.7. Python 中的注释	74
8.8. 使用变量和数据类型	74
8.9. 基本字符串操作	75
8.10. 要点总结	76
8.11. 练习	76
9. Python 数据结构	78
9.1. 简介	78
9.2. 学习目标	78
9.3. 元组	78
9.4. 列表	79
9.5. 集合	81
9.6. 字典	83
9.7. 数据结构选择指南	86
9.8. 要点总结	87
9.9. 练习	87
10. 字符串操作	90
10.1. 简介	90
10.2. 学习目标	90
10.3. 创建和操作字符串	90
10.4. 地理空间数据的字符串方法	92
10.5. 字符串格式化	95
10.6. 字符串操作决策指南	98
10.7. 要点总结	98
10.8. 练习	98
11. 循环和条件语句	100
11.1. 简介	100
11.2. 学习目标	100
11.3. For 循环	100
11.4. While 循环	101
11.5. 控制语句：在代码中做出决策	103
11.6. 结合循环和控制语句	104
11.7. 循环和控制语句决策指南	106
11.8. 要点总结	106
11.9. 练习	107
12. 函数和类	108
12.1. 简介	108
12.2. 学习目标	108
12.3. 函数：构建可重用的代码块	108
12.4. 类：将数据和行为组织在一起	113
12.5. 结合函数和类	115
12.6. 函数和类设计指南	115
12.7. 要点总结	116
12.8. 练习	116

13. 文件操作	117
13.1. 简介	117
13.2. 学习目标	117
13.3. 创建示例文件	117
13.4. 读取和写入文件	118
13.5. 异常处理	119
13.6. 结合文件处理和异常处理	121
13.7. 处理不同的文件格式	122
13.8. 要点总结	124
13.9. 练习	124
14. 使用 NumPy 和 Pandas 进行数据分析	127
14.1. 简介	127
14.2. 学习目标	127
14.3. NumPy 简介	127
14.4. Pandas 简介	137
14.5. 结合 NumPy 和 Pandas	144
14.6. 要点总结	145
14.7. 进一步阅读	145
14.8. 练习	146
III: Python 地理空间编程	149
15. 地理空间 Python 入门	151
15.1. 简介	151
15.2. 地理空间 Python 生态系统	151
15.3. 理解库之间的关系	152
15.4. 设置您的环境	152
15.5. 验证和首次步骤	153
15.6. 学习路径和章节概览	154
15.7. 需要记住的关键概念	155
15.8. 获取帮助和资源	155
15.9. 下一步	156
15.10. 练习	156
16. 使用 GeoPandas 进行矢量数据分析	157
16.1. 简介	157
16.2. 学习目标	157
16.3. 核心概念	157
16.4. 安装 GeoPandas	158
16.5. 创建 GeoDataFrames	158
16.6. 读写地理空间数据	159
16.7. 投影和坐标参考系统 (CRS)	160
16.8. 空间测量和分析	162
16.9. 可视化地理空间数据	164
16.10. 高级几何操作	168
16.11. 空间关系和查询	171
16.12. 最佳实践和性能考虑	172
16.13. 要点总结	173

16.14. 练习	173
17. 使用 Rasterio 处理栅格数据	174
17.1. 简介	174
17.2. 学习目标	174
17.3. 安装 Rasterio	175
17.4. 读取栅格数据	175
17.5. 可视化栅格数据	178
17.6. 访问和操作栅格波段	186
17.7. 写入栅格数据	187
17.8. 裁剪栅格数据	188
17.9. 要点总结	191
17.10. 练习	191
18. 使用 Xarray 进行多维数据分析	194
18.1. 简介	194
18.2. 学习目标	194
18.3. 理解 Xarray 的数据模型	195
18.4. 设置您的环境	195
18.5. 加载和探索真实气候数据	196
18.6. 使用 DataArrays	197
18.7. 直观的数据选择和索引	199
18.8. 对多维数据执行操作	200
18.9. 使用 Xarray 进行数据可视化	202
18.10. 使用 Datasets：多个变量	204
18.11. 基于标签操作的威力	205
18.12. 高级索引技术	207
18.13. 高级计算操作	208
18.14. 数据输入和输出	211
18.15. 要点总结	212
18.16. 延伸阅读	212
18.17. 练习	213
19. 使用 Rioxarray 进行栅格分析	215
19.1. 简介	215
19.2. 学习目标	215
19.3. 设置您的 Rioxarray 环境	215
19.4. 加载和探索地理参考栅格数据	216
19.5. 基本地理空间操作	218
19.6. 处理空间维度和分辨率	220
19.7. 可视化地理空间栅格数据	222
19.8. 数据存储和文件管理	226
19.9. 坐标系统比较	227
19.10. 波段数学简介	229
19.11. 要点总结	233
19.12. 练习	233
20. 使用 Leafmap 进行交互式可视化	235
20.1. 简介	235

20.2.	学习目标	235
20.3.	安装和设置 Leafmap	236
20.4.	创建交互式地图	237
20.5.	更改底图	240
20.6.	可视化矢量数据	245
20.7.	创建分级统计图	249
20.8.	可视化 GeoParquet 数据	251
20.9.	可视化 PMTiles	252
20.10.	可视化栅格数据	256
20.11.	访问和可视化 Maxar 开放数据	263
20.12.	要点总结	269
20.13.	练习	270
21.	使用 WhiteboxTools 进行地理处理	273
21.1.	简介	273
21.2.	学习目标	273
21.3.	为什么选择 Whitebox?	273
21.4.	Whitebox 的有用资源	274
21.5.	安装 Whitebox	275
21.6.	流域分析	275
21.7.	LiDAR 数据分析	288
21.8.	要点总结	294
21.9.	练习	295
22.	使用 MapLibre 进行三维地图制作	298
22.1.	简介	298
22.2.	学习目标	298
22.3.	有用资源	298
22.4.	安装和设置	298
22.5.	创建交互式地图	299
22.6.	添加地图控件	300
22.7.	添加图层	303
22.8.	使用 MapTiler	305
22.9.	三维地图	306
22.10.	可视化矢量数据	313
22.11.	可视化栅格数据	323
22.12.	添加自定义组件	324
22.13.	可视化 PMTiles	332
22.14.	添加 DeckGL 图层	337
22.15.	导出为 HTML	340
22.16.	要点总结	340
22.17.	练习	340
23.	使用 Earth Engine 和 Geemap 进行云计算	343
23.1.	简介	343
23.2.	学习目标	343
23.3.	Google Earth Engine 简介	343
23.4.	交互式地图和工具简介	346

23.5. Earth Engine 数据目录	351
23.6. Earth Engine 数据类型	352
23.7. Earth Engine 栅格数据	352
23.8. Earth Engine 矢量数据	355
23.9. 更多用于可视化 Earth Engine 数据的工具	357
23.10. 矢量数据处理	366
23.11. 栅格数据处理	368
23.12. 导出 Earth Engine 数据	375
23.13. 创建延时动画	377
23.14. 绘制 Earth Engine 数据图表	382
23.15. 要点总结	411
23.16. 练习	412
24. 使用 <i>HyperCoast</i> 进行高光谱数据可视化	413
24.1. 简介	413
24.2. 学习目标	413
24.3. 环境设置	413
24.4. 查找高光谱数据	414
24.5. 下载高光谱数据	416
24.6. 读取高光谱数据	417
24.7. 可视化高光谱数据	417
24.8. 创建图像立方体	419
24.9. 交互式切片	420
24.10. 交互式阈值处理	421
24.11. 要点总结	422
24.12. 练习	423
25. 使用 <i>DuckDB</i> 进行高性能地理空间分析	424
25.1. 简介	424
25.2. 学习目标	424
25.3. 安装与设置	425
25.4. 面向空间分析的 SQL 基础	427
25.5. Python API 集成	432
25.6. 数据导入	434
25.7. 数据导出	438
25.8. 使用几何图形	440
25.9. 空间关系	443
25.10. 空间连接	445
25.11. 大规模数据分析	448
25.12. 要点总结	455
25.13. 练习	455
26. 使用 <i>GDAL</i> 和 <i>OGR</i> 进行地理空间数据处理	460
26.1. 简介	460
26.2. 学习目标	460
26.3. 安装与设置	461
26.4. 示例数据集	461
26.5. 理解你的数据	461

26.6.	坐标变换	462
26.7.	格式转换	462
26.8.	裁剪和掩膜	463
26.9.	栅格分析与计算	464
26.10.	在栅格和矢量之间转换	465
26.11.	几何处理	466
26.12.	管理字段和图层	467
26.13.	切片和数据管理	468
26.14.	高级栅格处理	469
26.15.	地形分析	470
26.16.	要点总结	476
26.17.	参考文献和进一步阅读	477
26.18.	练习	477
27.	使用 Voilà 和 Solara 构建交互式仪表板	480
27.1.	简介	480
27.2.	学习目标	480
27.3.	安装 Voilà 和 Solara	481
27.4.	Hugging Face Spaces 简介	481
27.5.	创建一个基本的 Voilà 应用程序	482
27.6.	使用 Solara 创建高级 Web 应用程序	487
27.7.	要点总结	493
27.8.	练习	493
28.	使用 Apache Sedona 进行分布式计算	495
28.1.	简介	495
28.2.	学习目标	495
28.3.	安装和设置 Apache Sedona	496
28.4.	下载示例数据	498
28.5.	核心概念和数据结构	498
28.6.	空间操作和函数	500
28.7.	空间连接和索引	503
28.8.	高级空间分析	505
28.9.	读取矢量数据	507
28.10.	可视化矢量数据	510
28.11.	写入矢量数据	513
28.12.	读取栅格数据	514
28.13.	可视化栅格数据	515
28.14.	栅格地图代数	516
28.15.	栅格区域统计	518
28.16.	写入栅格数据	518
28.17.	与 GeoPandas 集成	519
28.18.	真实世界用例	522
28.19.	要点总结	524
28.20.	参考文献和进一步阅读	525
28.21.	练习	525

序言

简介

地理信息系统 (GIS) 和地理空间分析已成为众多学科的基础工具，从环境科学和城市规划到商业分析和公共卫生。随着地理空间数据量和复杂性的指数级增长，以编程方式处理、分析和可视化这些数据的能力已成为处理空间信息的研究人员、分析师和专业人士的必备技能。

Python 已成为地理空间分析的主导编程语言，提供了丰富的库和工具生态系统，使复杂的空间操作对初学者和专家都变得易于访问。然而，从 Python 新手到自信的地理空间程序员的道路可能看起来令人望而生畏，需要学习众多库和掌握各种概念。

本书通过提供结构化、实用的 Python 地理空间编程学习方法来弥合这一差距。我们不是从一开始就用高级技术让您感到压倒，而是专注于建立坚实的基础技能，这些技能将在您的整个地理空间编程之旅中为您服务。每一章都建立在前一章的基础上，确保您既能发展理论理解又能获得实践专长。

本书采用的方法是实践性和示例驱动的。您将使用真实的地理空间数据集，解决实际问题，并构建展示 Python 在地理空间分析和可视化方面强大功能的项目。在本书结束时，您将有信心和技能来应对自己的地理空间编程挑战。

本书的目标读者

本书专为希望利用 Python 强大功能进行地理空间分析和可视化的多样化学习者群体而设计：

学生和研究人員 在地理学、环境科学、城市规划、数据科学和相关领域，需要在学习或研究中分析空间数据。不假设有编程经验，尽管对计算机和数据分析概念的基本熟悉是有帮助的。

GIS 专业人士 目前使用桌面 GIS 软件并希望通过编程技能扩展其工具包。如果您曾使用过 ArcGIS、QGIS 或类似工具，并希望自动化工作流程或执行在传统 GIS 软件中具有挑战性的分析，本书将帮助您实现这一转变。

数据科学家和分析师 处理基于位置的数据并希望将空间分析能力添加到他们的技能组合中。如果您对 Python 基础知识很熟悉但对地理空间概念较新，本书将为您介绍所需的思维空间和工具。

软件开发人员 对构建处理地理空间数据的应用程序感兴趣。无论您是在开发网络地图应用程序、具有位置功能的移动应用程序，还是数据处理管道，本书都提供了您所需的基础。

自学者和职业转换者 对不断增长的地理空间数据科学领域感兴趣。本书不假设有 Python 编程或 GIS 概念的先验知识，使有动力的初学者能够轻松入门。

政府和行业专业人士 需要将空间分析纳入其工作中，如城市规划师、环境顾问、市场研究人员、物流协调员或公共卫生官员。

关键要求是好奇心和意愿。虽然编程经验有帮助，但不是必需的。我们从基础知识开始，系统地建立起来。

本书涵盖的内容

本书分为三个渐进的部分，带您从软件设置到 Python 基础再到高级地理空间编程：

软件设置 为您准备地理空间编程所需的一切开发环境。您将学习安装和配置基本工具，包括用于包管理的 Miniconda、用于开发的 VS Code、用于版本控制的 Git，以及像 Google Colab 和 JupyterLab 这样的基于云的替代方案。本部分确保您在深入编程之前有一个坚实的基础。

Python 编程基础 通过七个综合章节构建您的核心编程技能。从 Python 基础开始，您将掌握变量和数据类型、数据结构（列表、字典、集合）、字符串操作、使用循环和条件的控制流、函数和类、文件处理，以及使用 NumPy 和 Pandas 进行数据分析。这些技能构成了所有地理空间编程任务的基础。

Python 地理空间编程 包含十四个专业章节，将您转变为自信的地理空间程序员：

- **地理空间 Python 简介** - 核心概念和 Python 地理空间生态系统
- **使用 GeoPandas 进行矢量数据分析** - 处理点、线和面
- **使用 Rasterio 处理栅格数据** - 处理卫星图像和网格数据集
- **使用 Xarray 进行多维数据分析** - 处理复杂的科学数据集
- **使用 Rioxarray 进行栅格分析** - 高级栅格处理和分析
- **使用 Leafmap 进行交互式可视化** - 创建动态交互式地图
- **使用 WhiteboxTools 进行地理处理** - 高级空间分析操作
- **使用 MapLibre 进行 3D 地图制作** - 构建三维可视化
- **使用 Earth Engine 和 Geemap 进行云计算** - 利用 Google Earth Engine 进行大规模分析
- **使用 HyperCoast 进行高光谱数据可视化** - 处理高光谱数据
- **使用 DuckDB 进行高性能地理空间分析** - 高性能空间数据处理
- **使用 GDAL 和 OGR 进行地理空间数据处理** - 处理各种地理空间数据格式
- **使用 Solara 构建交互式仪表板** - 为地理空间应用程序创建交互式仪表板
- **使用 Apache Sedona 进行分布式计算** - 在分布式环境中处理大型地理空间数据集

每章都遵循一致的结构：

- 具有真实世界背景的清晰概念解释
- 带有详细注释的逐步代码示例
- 使用真实地理空间数据集的实践练习
- 常见陷阱和故障排除指导
- 附加资源和进一步阅读的参考

这种进展是精心设计的，使每章都建立在前面的概念基础上，同时引入新的功能，确保您在地理空间编程方面既有广度又有深度。

充分利用本书

为了最大化您的学习体验，请考虑以下建议：

建立适当的开发环境：按照本书第一部分的描述安装 Python 和所需的库。一个配置良好的环境将在您的整个学习过程中为您节省时间和挫折。考虑使用 conda 或 uv 来管理您的 Python 包，因为这简化了地理空间库的安装。

跟随代码示例：本书旨在是交互式的。不要只是阅读代码——输入它，运行它，并尝试修改。理解来自于实践，每个示例都构建了您稍后需要的技能。

完成练习：每章都包含旨在加强您所学概念的练习。这些不是可选的额外内容——它们是学习过程的组成部分。从指导练习开始，然后用您自己的项目挑战自己。

使用真实数据：虽然本书为示例和练习提供了数据集，但尝试将技术应用于您自己领域或兴趣的数据。这将帮助您理解概念如何应用于真实世界的场景，并建立对您能力的信心。

构建项目：当您在本书中进步时，考虑从事一个您感兴趣的个人项目。这可能是分析您研究中的数据、为您的社区创建地图，或解决您在工作中遇到的问题。

对自己有耐心：编程可能令人沮丧，特别是当您在学习时。期望遇到错误、花时间调试，偶尔感到困惑。这是正常的，是学习过程的一部分。在需要时休息，记住专业知识是通过持续实践逐渐发展的。如果您遇到困难，不要犹豫在本书的 GitHub 仓库上寻求帮助。

继续练习：本书中的技能需要定期练习来维持和发展。定期留出时间进行地理空间编程项目，即使是小项目。

本书使用的约定

本书使用几种约定来帮助您浏览内容和理解代码示例：

代码格式：所有 Python 代码都出现在代码块内的等宽字体中。当代码出现在常规文本中时，它被格式化为 像这样。文件和目录名也以等宽字体格式化。

代码示例：大多数代码示例都是完整的和可运行的。它们包含解释正在演示的关键概念和技术的注释。可能包含行号以供配套文本中的参考。

```
# 这是代码块的示例
import leafmap
m = leafmap.Map()
m.add_basemap("OpenTopoMap") # 向地图添加底图
m
```

命令行指令：要在命令行或终端输入的命令用 \$ 提示符显示：

```
$ pip install leafmap
$ python script.py
```

下载代码示例

本书的所有代码示例、数据集和补充材料都在 GitHub 上免费提供：

<https://github.com/giswqs/intro-gispro>

要下载材料，您可以使用以下方法之一：

- **克隆仓库**（如果您安装了 Git）：

```
$ git clone https://github.com/giswqs/intro-gispro.git
```

- **下载为 ZIP**（如果您不想使用 Git）：
 - 访问 GitHub 仓库页面
 - 点击绿色的 **Code** 按钮
 - 选择 **Download ZIP**
 - 将文件解压到您首选的位置
- **在线浏览单个文件**，如果您只需要特定示例，可通过 GitHub 界面

仓库定期更新，包含更正、改进和额外示例。定期检查更新，或在 GitHub 上[关注](#)仓库以获得更改通知。

如果您在代码中发现错误或有改进建议，请在 GitHub 上开启 issue 或提交 pull request。社区贡献有助于为每个人改善这一资源。

视频教程

为了补充书面内容，本书由一系列全面的视频教程支持，这些教程讲解关键概念并提供额外示例：

<https://tinyurl.com/intro-gispro-videos>

视频旨在补充而不是替代书面材料。它们特别有助于：

- 受益于看到代码被编写和执行的视觉学习者
- 通过多种解释理解复杂概念
- 了解开发工作流程和最佳实践
- 了解如何处理问题和调试问题

播放列表组织得遵循本书的结构。您可以在阅读本书时按顺序观看，或根据需要跳转到特定主题。

视频是在 2024 年秋季我在田纳西大学教授 [GIS 编程入门](#)¹ 课程时创建的。虽然课程已经结束，但视频仍然相关，可以用作本书的参考。未来将添加更多视频。

联系我们

我欢迎读者的反馈、问题和建议。您的意见有助于改进本书，使其对地理空间编程社区更有用。

对于与书籍相关的问题和讨论：

- GitHub Issues: <https://github.com/giswqs/intro-gispro/issues>
- GitHub Discussions: <https://github.com/giswqs/intro-gispro/discussions>

特别有用的反馈类型：

- 文本或代码中的错误或不清楚的解释
- 对额外示例或用例的建议
- 新主题或章节的想法
- 不同操作系统或库版本的兼容性问题报告
- 您如何应用本书技术的成功故事

致谢

如果没有许多个人的贡献和支持以及更广泛的开源地理空间社区，本书就不可能完成。

开源社区：本书建立在无数开源开发人员的惊人工作基础上，他们创建和维护了 Python 地理空间生态系统。特别感谢 NumPy、Pandas、GeoPandas、Rasterio、Xarray、Rioxarray、Folium、ipyleaflet、MapLibre、GDAL 以及许多其他使地理空间编程变得易于访问的库的开发人员和维护者。

学生和同事：我在田纳西大学地理空间编程课程中学生的的问题、挑战和见解塑造了本书的方法和内 容。他们对什么有效、什么无效的反馈在创建真正服务于学习者的材料方面非常宝贵。

¹<https://geog-312.gishub.org>

研究合作者：地理空间研究社区的同事和合作者提供了真实世界的用例、数据集和问题场景，为本书中的实际示例提供了信息。

家人和朋友：撰写技术书籍需要大量的时间和专注。我感谢家人和朋友的耐心和支持，他们理解许多晚上和周末都致力于这个项目。

更广泛的 GIS 社区：地理空间领域建立在分享知识和工具的基础上。本书是这一传统的一部分，我很荣幸能为学习地理空间编程的可用资源做出贡献。

本书使用 [MyST Markdown](#)² 编写，并使用 [Typst](#)³ 和 [min-book](#)⁴ 模板编译。感谢 Typst 和 MyST Markdown 项目的开发人员和维护者。特别感谢 [Maycon F. Melo](#)⁵ 提供 min-book 模板以及他们在为本书定制模板方面的帮助。

本书中的任何错误或遗漏仍然是我的责任。我致力于根据读者反馈解决问题和改进内容。

关于作者

吴秋生博士是田纳西大学诺克斯维尔分校地理系副教授。他还担任亚马逊学者。吴博士的研究专注于地理空间数据科学和开源软件开发，重点是利用大地理空间数据和云计算来研究环境变化，特别是地表水和湿地淹没动态。他是几个广泛使用的开源 Python 包的创造者，包括 [geemap](#)⁶、[leafmap](#)⁷、[segment-geospatial](#)⁸ 和 [geoai](#)⁹，这些支持高级地理空间分析和交互式可视化。他的开源工作可在 GitHub 的 [Open Geospatial Solutions](#)¹⁰ 上获得。

许可和版权

本书拥抱开放科学和开放教育的原则。为了支持透明度、学习和重用，本书中的**代码示例**在 [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](#) 许可下发布。这意味着您可以自由复制、修改和分发代码，甚至用于商业目的，只要给出适当的信用。

请通过引用本书或链接到 GitHub 仓库来归属代码使用：

Wu, Q. (2025). *Introduction to GIS Programming: A Practical Python Guide to Open Source Geospatial Tools*. <https://gispro.gishub.org>

虽然代码是免费提供的，但本书中的**文本、图表和图像**是作者**受版权保护的**，未经明确许可不得复制、重新分发或修改。这包括所有书面内容、自定义图表和嵌入式可视化，除非另有说明。

如果您希望重用或改编本书中的任何非代码材料——例如，用于教学、演示或出版物——请联系作者请求许可。

这种双重许可方法有助于平衡对学习材料的开放访问与对原创性工作的保护。感谢您遵守这些条款并支持开源地理空间社区。

²<https://mystmd.org>

³<https://github.com/typst/typst>

⁴<https://github.com/mayconfmelo/min-book>

⁵<https://github.com/mayconfmelo>

⁶<https://geemap.org>

⁷<https://leafmap.org>

⁸<https://samgeo.gishub.org>

⁹<https://opengeoai.org>

¹⁰<https://github.com/opengeos>

GIS编程入门

开源地理空间工具的 Python 实用指南

《GIS编程入门》是一本全面、实践性强的书籍，旨在帮助读者掌握使用开源Python包进行地理空间分析。本书适合各个层次的学习者，将地理信息系统（GIS）的复杂性分解为清晰、可操作的步骤，是学生、研究人员、专业人士和自学者掌握空间数据编程的理想选择。

地理空间数据已成为环境科学、城市规划、公共卫生和商业分析等众多领域的关键要素。随着这类数据的体量和复杂性不断增长，对可访问工具来分析、处理和可视化地理空间数据的需求从未如此迫切。Python凭借其丰富的库生态系统，成为处理地理空间数据的首选编程语言——然而面对广泛的库和概念，可能会让人感到不知所措。本书提供了从Python新手成长为自信的地理空间程序员所需的结构和清晰指导。

本书的独特之处在于其循序渐进、以实例驱动的方法。从Python编程基础技能开始，您将逐步建立理解，进而掌握地理空间分析的高级技术。内容设计为交互式，配有真实世界的数据集和实践练习，让您能够立即应用所学技能。您将完成各种项目，从基础的空间数据操作到构建交互式仪表板和基于云的地理空间应用程序。

无论您是希望自动化GIS工作流程、开发地理空间Web应用程序，还是深化空间数据科学技能，本书都将以清晰和自信的方式指导您完成整个过程。

你将学到的内容：

- **搭建开发环境：**用于地理空间编程的工具，如 Miniconda、VS Code、Git 和 Colab。
- **核心Python编程：**包括数据类型、控制流、函数、类、文件处理，以及用于数据操作的 NumPy 和 Pandas 等库。
- **地理空间编程：**使用 GeoPandas、Rasterio、和 Leafmap 等库进行矢量和栅格数据处理、地理空间分析和创建交互式可视化的实践指导。
- **高级主题：**使用 Google Earth Engine 进行云计算、高光谱数据分析、高性能地理空间分析，以及使用 Apache Sedona 进行分布式计算。

关于作者：

吴秋生博士是田纳西大学副教授和亚马逊学者。他的研究专注于地理空间数据科学，重点使用开源Python工具和云计算来研究环境变化。他是多个广泛使用的Python包的创建者，包括geemap、leafmap和geoai，致力于推进地理空间分析和可视化。吴博士将其深厚的专业知识带入本书，指导您学习和掌握Python地理空间编程的过程。



扫描二维码访问图书网站，并从
<https://gispro.gishub.org>
下载代码示例。