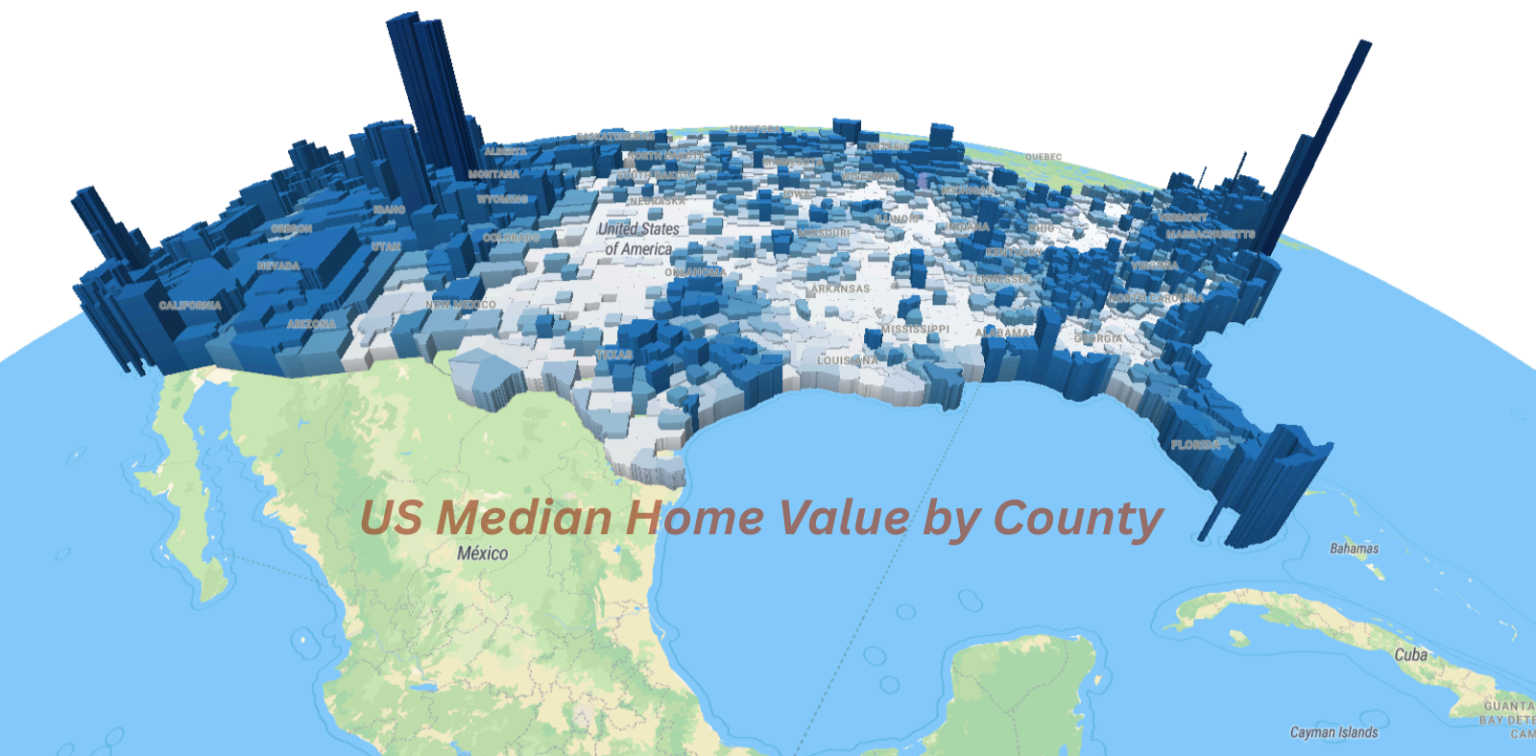


Introduction to GIS Programming

A Practical Python Guide to
Open Source Geospatial Tools

Qiusheng Wu



Introduction to GIS Programming

A Practical Python Guide to Open Source
Geospatial Tools

Qiusheng Wu
2025

Contents

Preface	1
Introduction	3
Who This Book Is For	3
What This Book Covers	4
Getting the Most Out of This Book	5
Conventions Used in This Book	5
Downloading the Code Examples	6
Video Tutorials	6
Get in Touch	7
Acknowledgments	7
About the Author	8
Licensing and Copyright	8
I: Software Setup	9
1. Overview of Software Tools	11
1.1. Introduction	11
1.2. Learning Objectives	11
1.3. Essential Software Tools	11
1.4. Tool Integration and Workflow	13
1.5. Running Code Examples	14
1.6. Key Takeaways	14
2. Introduction to Python Package Management	15
2.1. Introduction	15
2.2. Learning Objectives	15
2.3. Installing Conda (Miniconda)	16
2.4. Understanding Conda Concepts	17
2.5. Creating Your First Geospatial Environment	18
2.6. Troubleshooting Conda	19
2.7. Essential Conda Commands	20
2.8. Introducing uv: The Fast Alternative	24
2.9. Best Practices for Package Management	25
2.10. Key Takeaways	26
2.11. Exercises	26
3. Setting Up Visual Studio Code	28
3.1. Introduction	28
3.2. Learning Objectives	28
3.3. Installing Visual Studio Code	29
3.4. Essential Extensions for Python Programming	29
3.5. Configure VS Code for Python Development	31
3.6. Essential Keyboard Shortcuts	32
3.7. References and Further Learning	34
3.8. Key Takeaways	34
3.9. Exercises	34
4. Version Control with Git	36
4.1. Introduction	36
4.2. Learning Objectives	37
4.3. Setting Up GitHub Account	37

4.4.	Installing Git	37
4.5.	Configuring Git	38
4.6.	Understanding Git Concepts	38
4.7.	Essential Git Commands	39
4.8.	Using GitHub	42
4.9.	Integration with VS Code	43
4.10.	Best Practices for Geospatial Projects	43
4.11.	Key Takeaways	44
4.12.	Exercises	45
5.	<i>Using Google Colab</i>	47
5.1.	Introduction	47
5.2.	Learning Objectives	47
5.3.	Getting Started with Google Colab	47
5.4.	Setting Up Your Geospatial Environment	48
5.5.	Essential Colab Features	50
5.6.	Run Code Examples in Colab	51
5.7.	Key Takeaways	51
5.8.	Exercises	52
6.	<i>Working with JupyterLab</i>	53
6.1.	Introduction	53
6.2.	Learning Objectives	53
6.3.	Installing and Setting Up JupyterLab	53
6.4.	Getting Started with JupyterLab	54
6.5.	Essential Keyboard Shortcuts	57
6.6.	Running Code Examples on MyBinder	59
6.7.	Key Takeaways	59
6.8.	Exercises	59
7.	<i>Using Docker</i>	62
7.1.	Introduction	62
7.2.	Learning Objectives	62
7.3.	Installing Docker Desktop	62
7.4.	Basic Concepts	64
7.5.	Running Code Examples in Docker	64
7.6.	Common Docker Commands	65
7.7.	Key Takeaways	66
7.8.	Exercises	67
II:	Python Programming Fundamentals	69
8.	<i>Variables and Data Types</i>	71
8.1.	Introduction	71
8.2.	Learning Objectives	71
8.3.	Variables in Python	71
8.4.	Naming Variables	72
8.5.	Data Types	73
8.6.	Escape Characters	74
8.7.	Comments in Python	74
8.8.	Working with Variables and Data Types	74
8.9.	Basic String Operations	75
8.10.	Key Takeaways	76

8.11. Exercises	76
9. Python Data Structures	78
9.1. Introduction	78
9.2. Learning Objectives	78
9.3. Tuples	78
9.4. Lists	79
9.5. Sets	81
9.6. Dictionaries	84
9.7. Data Structure Selection Guide	87
9.8. Key Takeaways	87
9.9. Exercises	88
10. String Operations	90
10.1. Introduction	90
10.2. Learning Objectives	90
10.3. Creating and Manipulating Strings	90
10.4. String Methods for Geospatial Data	92
10.5. String Formatting	95
10.6. String Operation Decision Guide	98
10.7. Key Takeaways	98
10.8. Exercises	99
11. Loops and Conditional Statements	101
11.1. Introduction	101
11.2. Learning Objectives	101
11.3. For Loops	101
11.4. While Loops	103
11.5. Control Statements: Making Decisions in Your Code	104
11.6. Combining Loops and Control Statements	105
11.7. Loop and Control Statement Decision Guide	107
11.8. Key Takeaways	108
11.9. Exercises	108
12. Functions and Classes	110
12.1. Introduction	110
12.2. Learning Objectives	110
12.3. Functions: Building Reusable Code Blocks	110
12.4. Classes: Organizing Data and Behavior Together	115
12.5. Combining Functions and Classes	117
12.6. Function and Class Design Guidelines	117
12.7. Key Takeaways	118
12.8. Exercises	118
13. Working with Files	120
13.1. Introduction	120
13.2. Learning Objectives	120
13.3. Creating a Sample File	120
13.4. Reading and Writing Files	121
13.5. Exception Handling	122
13.6. Combining File Handling and Exception Handling	124
13.7. Working with Different File Formats	125
13.8. Key Takeaways	127

13.9. Exercises	128
14. Data Analysis with NumPy and Pandas	130
14.1. Introduction	130
14.2. Learning Objectives	130
14.3. Introduction to NumPy	130
14.4. Introduction to Pandas	141
14.5. Combining NumPy and Pandas	147
14.6. Key Takeaways	148
14.7. Further Reading	149
14.8. Exercises	149
III: Geospatial Programming with Python	151
15. Introduction to Geospatial Python	153
15.1. Introduction	153
15.2. The Geospatial Python Ecosystem	153
15.3. Understanding Library Relationships	154
15.4. Setting Up Your Environment	154
15.5. Verification and First Steps	156
15.6. Learning Path and Chapter Overview	156
15.7. Key Concepts to Remember	157
15.8. Getting Help and Resources	157
15.9. Next Steps	158
15.10. Exercises	158
16. Vector Data Analysis with GeoPandas	159
16.1. Introduction	159
16.2. Learning Objectives	159
16.3. Core Concepts	159
16.4. Installing GeoPandas	160
16.5. Creating GeoDataFrames	160
16.6. Reading and Writing Geospatial Data	161
16.7. Projections and Coordinate Reference Systems (CRS)	162
16.8. Spatial Measurements and Analysis	164
16.9. Visualizing Geospatial Data	166
16.10. Advanced Geometric Operations	170
16.11. Spatial Relationships and Queries	174
16.12. Best Practices and Performance Considerations	174
16.13. Key Takeaways	175
16.14. Exercises	175
17. Working with Raster Data Using Rasterio	177
17.1. Introduction	177
17.2. Learning Objectives	177
17.3. Installing Rasterio	178
17.4. Reading Raster Data	178
17.5. Visualizing Raster Data	181
17.6. Accessing and Manipulating Raster Bands	188
17.7. Writing Raster Data	189
17.8. Clipping Raster Data	190
17.9. Key Takeaways	193
17.10. Exercises	193

18. Multi-dimensional Data Analysis with Xarray	196
18.1. Introduction	196
18.2. Learning Objectives	196
18.3. Understanding Xarray's Data Model	197
18.4. Setting Up Your Environment	197
18.5. Loading and Exploring Real Climate Data	198
18.6. Working with DataArrays	199
18.7. Intuitive Data Selection and Indexing	201
18.8. Performing Operations on Multi-Dimensional Data	202
18.9. Data Visualization with Xarray	204
18.10. Working with Datasets: Multiple Variables	206
18.11. The Power of Label-Based Operations	207
18.12. Advanced Indexing Techniques	209
18.13. High-Level Computational Operations	210
18.14. Data Input and Output	213
18.15. Key Takeaways	214
18.16. Further Reading	215
18.17. Exercises	215
19. Raster Analysis with Rioxarray	217
19.1. Introduction	217
19.2. Learning Objectives	217
19.3. Setting Up Your Rioxarray Environment	217
19.4. Loading and Exploring Georeferenced Raster Data	218
19.5. Fundamental Geospatial Operations	221
19.6. Working with Spatial Dimensions and Resolution	222
19.7. Visualizing Geospatial Raster Data	224
19.8. Data Storage and File Management	228
19.9. Coordinate System Comparisons	230
19.10. Introduction to Band Math	232
19.11. Key Takeaways	235
19.12. Exercises	236
20. Interactive Visualization with Leafmap	238
20.1. Introduction	238
20.2. Learning Objectives	239
20.3. Installing and Setting Up Leafmap	239
20.4. Creating Interactive Maps	240
20.5. Changing Basemaps	243
20.6. Visualizing Vector Data	248
20.7. Creating Choropleth Maps	253
20.8. Visualizing GeoParquet Data	254
20.9. Visualizing PMTiles	256
20.10. Visualizing Raster Data	260
20.11. Accessing and Visualizing Maxar Open Data	267
20.12. Key Takeaways	274
20.13. Exercises	274
21. Geoprocessing with WhiteboxTools	278
21.1. Introduction	278
21.2. Learning Objectives	278

21.3.	Why Whitebox?	278
21.4.	Useful Resources for Whitebox	280
21.5.	Installing Whitebox	280
21.6.	Watershed Analysis	281
21.7.	LiDAR Data Analysis	294
21.8.	Key Takeaways	301
21.9.	Exercises	302
22.	<i>3D Mapping with MapLibre</i>	305
22.1.	Introduction	305
22.2.	Learning Objectives	305
22.3.	Useful Resources	305
22.4.	Installation and Setup	305
22.5.	Creating Interactive Maps	306
22.6.	Adding Map Controls	307
22.7.	Adding Layers	310
22.8.	Using MapTiler	312
22.9.	3D Mapping	313
22.10.	Visualizing Vector Data	320
22.11.	Visualizing Raster Data	330
22.12.	Adding Custom Components	332
22.13.	Visualizing PMTiles	339
22.14.	Adding DeckGL Layers	344
22.15.	Exporting to HTML	347
22.16.	Key Takeaways	347
22.17.	Exercises	347
23.	<i>Cloud Computing with Earth Engine and Geemap</i>	350
23.1.	Introduction	350
23.2.	Learning Objectives	350
23.3.	Introduction to Google Earth Engine	350
23.4.	Introduction to Interactive Maps and Tools	353
23.5.	The Earth Engine Data Catalog	358
23.6.	Earth Engine Data Types	360
23.7.	Earth Engine Raster Data	360
23.8.	Earth Engine Vector Data	362
23.9.	More Tools for Visualizing Earth Engine Data	365
23.10.	Vector Data Processing	373
23.11.	Raster Data Processing	376
23.12.	Exporting Earth Engine Data	382
23.13.	Creating Timelapse Animations	385
23.14.	Charting Earth Engine Data	390
23.15.	Key Takeaways	420
23.16.	Exercises	421
24.	<i>Hyperspectral Data Visualization with HyperCoast</i>	423
24.1.	Introduction	423
24.2.	Learning Objectives	423
24.3.	Environment Setup	424
24.4.	Finding Hyperspectral Data	424
24.5.	Downloading Hyperspectral Data	426

24.6.	Reading Hyperspectral Data	427
24.7.	Visualizing Hyperspectral Data	427
24.8.	Creating Image Cubes	429
24.9.	Interactive Slicing	430
24.10.	Interactive Thresholding	432
24.11.	Key Takeaways	433
24.12.	Exercises	433
25.	<i>High-Performance Geospatial Analytics with DuckDB</i>	435
25.1.	Introduction	435
25.2.	Learning Objectives	435
25.3.	Installation and Setup	436
25.4.	SQL Basics for Spatial Analysis	438
25.5.	Python API Integration	443
25.6.	Data Import	445
25.7.	Data Export	449
25.8.	Working with Geometries	451
25.9.	Spatial Relationships	454
25.10.	Spatial Joins	456
25.11.	Large-Scale Data Analysis	459
25.12.	Key Takeaways	466
25.13.	Exercises	467
26.	<i>Geospatial Data Processing with GDAL and OGR</i>	471
26.1.	Introduction	471
26.2.	Learning Objectives	471
26.3.	Installation and Setup	472
26.4.	Sample Datasets	472
26.5.	Understanding Your Data	472
26.6.	Coordinate Transformation	473
26.7.	Format Conversion	474
26.8.	Clipping and Masking	475
26.9.	Raster Analysis and Calculations	476
26.10.	Converting Between Raster and Vector	477
26.11.	Geometry Processing	478
26.12.	Managing Fields and Layers	479
26.13.	Tiling and Data Management	479
26.14.	Advanced Raster Processing	481
26.15.	Terrain Analysis	482
26.16.	Key Takeaways	488
26.17.	References and Further Reading	489
26.18.	Exercises	489
27.	<i>Building Interactive Dashboards with Voilà and Solara</i>	492
27.1.	Introduction	492
27.2.	Learning Objectives	493
27.3.	Installing Voilà and Solara	493
27.4.	Introduction to Hugging Face Spaces	493
27.5.	Creating a Basic Voilà Application	494
27.6.	Creating an Advanced Web Application with Solara	500
27.7.	Key Takeaways	505

27.8. Exercises	506
28. <i>Distributed Computing with Apache Sedona</i>	507
28.1. Introduction	507
28.2. Learning Objectives	508
28.3. Installing and Setting Up Apache Sedona	508
28.4. Downloading Sample Data	510
28.5. Core Concepts and Data Structures	510
28.6. Spatial Operations and Functions	513
28.7. Spatial Joins and Indexing	516
28.8. Advanced Spatial Analysis	519
28.9. Reading Vector Data	521
28.10. Visualizing Vector Data	524
28.11. Writing Vector Data	527
28.12. Reading Raster Data	527
28.13. Visualizing Raster Data	529
28.14. Raster Map Algebra	529
28.15. Raster Zonal Statistics	531
28.16. Writing Raster Data	532
28.17. Integration with GeoPandas	533
28.18. Real-World Use Cases	536
28.19. Key Takeaways	537
28.20. References and Further Reading	538
28.21. Exercises	539

Preface

Introduction

Geographic Information Systems (GIS) and geospatial analysis have become fundamental tools across numerous disciplines, from environmental science and urban planning to business analytics and public health. As the volume and complexity of geospatial data continue to grow exponentially, the ability to programmatically process, analyze, and visualize this data has become an essential skill for researchers, analysts, and professionals working with spatial information.

Python has emerged as the leading programming language for geospatial analysis, offering a rich ecosystem of libraries and tools that make complex spatial operations accessible to both beginners and experts. However, the path from Python novice to confident geospatial programmer can seem daunting, with numerous libraries to learn and concepts to master.

This book bridges that gap by providing a structured, practical approach to learning geospatial programming with Python. Rather than overwhelming you with advanced techniques from the start, we focus on building a solid foundation of essential skills that will serve you throughout your geospatial programming journey. Each chapter builds upon the previous ones, ensuring you develop both theoretical understanding and practical expertise.

The approach taken in this book is hands-on and example-driven. You'll work with real geospatial datasets, solve practical problems, and build projects that demonstrate the power of Python for geospatial analysis and visualization. By the end of this book, you'll have the confidence and skills to tackle your own geospatial programming challenges.

Who This Book Is For

This book is designed for a diverse audience of learners who want to harness the power of Python for geospatial analysis and visualization:

Students and Researchers in geography, environmental science, urban planning, data science, and related fields who need to analyze spatial data as part of their studies or research. No prior programming experience is assumed, though basic familiarity with computers and data analysis concepts is helpful.

GIS Professionals who currently use desktop GIS software and want to expand their toolkit with programming skills. If you've worked with ArcGIS, QGIS, or similar tools and want to automate workflows or perform analyses that are challenging in traditional GIS software, this book will help you make that transition.

Data Scientists and Analysts who work with location-based data and want to add spatial analysis capabilities to their skillset. If you're comfortable with Python basics but new to geospatial concepts, this book will introduce you to the spatial thinking and tools you need.

Software Developers interested in building applications that work with geospatial data. Whether you're developing web mapping applications, mobile apps with location features, or data processing pipelines, this book provides the foundation you need.

Self-Learners and Career Changers who are interested in the growing field of geospatial data science. The book assumes no prior knowledge of either Python programming or GIS concepts, making it accessible to motivated beginners.

Professionals in Government and Industry who need to incorporate spatial analysis into their work, such as urban planners, environmental consultants, market researchers, logistics coordinators, or public health officials.

The key requirement is curiosity and willingness to learn. While programming experience is helpful, it's not necessary. We start with the fundamentals and build up systematically.

What This Book Covers

This book is organized into three progressive sections that take you from software setup through Python fundamentals to advanced geospatial programming:

Software Setup prepares your development environment with everything you need for geospatial programming. You'll learn to install and configure essential tools including Miniconda for package management, VS Code for development, Git for version control, and cloud-based alternatives like Google Colab and JupyterLab. This section ensures you have a solid foundation before diving into programming.

Python Programming Fundamentals builds your core programming skills through seven comprehensive chapters. Starting with Python basics, you'll master variables and data types, data structures (lists, dictionaries, sets), string operations, control flow with loops and conditionals, functions and classes, file handling, and data analysis with NumPy and Pandas. These skills form the foundation for all geospatial programming tasks.

Geospatial Programming with Python comprises fourteen specialized chapters that transform you into a confident geospatial programmer:

- **Introduction to Geospatial Python** - Core concepts and the Python geospatial ecosystem
- **Vector Data Analysis with GeoPandas** - Working with points, lines, and polygons
- **Raster Data with Rasterio** - Processing satellite imagery and gridded datasets
- **Multi-dimensional Data Analysis with Xarray** - Handling complex scientific datasets
- **Raster Analysis with Rioxarray** - Advanced raster processing and analysis
- **Interactive Visualization with Leafmap** - Creating dynamic, interactive maps
- **Geoprocessing with WhiteboxTools** - Advanced spatial analysis operations
- **3D Mapping with MapLibre** - Building three-dimensional visualizations
- **Cloud Computing with Earth Engine and Geemap** - Leveraging Google Earth Engine for large-scale analysis
- **Hyperspectral Data Visualization with HyperCoast** - Working with hyperspectral data
- **High-Performance Geospatial Analytics with DuckDB** - High-performance spatial data processing
- **Geospatial Data Processing with GDAL and OGR** - Working with various geospatial data formats
- **Building Interactive Dashboards with Solara** - Creating interactive dashboards for geospatial applications
- **Distributed Computing with Apache Sedona** - Processing large geospatial datasets in a distributed environment

Each chapter follows a consistent structure:

- Clear concept explanations with real-world context
- Step-by-step code examples with detailed annotations
- Hands-on exercises using authentic geospatial datasets
- Common pitfalls and troubleshooting guidance
- References to additional resources and further reading

The progression is carefully designed so that each chapter builds upon previous concepts while introducing new capabilities, ensuring you develop both breadth and depth in geospatial programming.

Getting the Most Out of This Book

To maximize your learning experience with this book, consider the following recommendations:

Set Up a Proper Development Environment: Install Python and the required libraries as described in the first section of the book. A well-configured environment will save you time and frustration throughout your learning journey. Consider using conda or uv to manage your Python packages, as this simplifies the installation of geospatial libraries.

Follow Along with Code Examples: This book is designed to be interactive. Don't just read the code—type it out, run it, and experiment with modifications. Understanding comes through practice, and each example builds skills you'll need later.

Work Through the Exercises: Each chapter includes exercises designed to reinforce the concepts you've learned. These are not optional extras—they are an integral part of the learning process. Start with the guided exercises, then challenge yourself with your own projects.

Use Real Data: While the book provides datasets for examples and exercises, try applying the techniques to data from your own field or interests. This will help you understand how the concepts apply to real-world scenarios and build confidence in your abilities.

Build Projects: As you progress through the book, consider working on a personal project that interests you. This could be analyzing data from your research, creating maps for your community, or solving a problem you've encountered in your work.

Be Patient with Yourself: Programming can be frustrating, especially when you're learning. Expect to encounter errors, spend time debugging, and occasionally feel stuck. This is normal and part of the learning process. Take breaks when needed, and remember that expertise develops gradually through consistent practice. If you get stuck, don't hesitate to ask for help on the book's GitHub repository.

Keep Practicing: The skills in this book require regular practice to maintain and develop. Set aside time regularly to work on geospatial programming projects, even if they're small ones.

Conventions Used in This Book

This book uses several conventions to help you navigate the content and understand the code examples:

Code Formatting: All Python code appears in monospaced font within code blocks. When code appears within regular text, it is formatted `like this`. File and directory names are also formatted in monospaced font.

Code Examples: Most code examples are complete and runnable. They include comments explaining the key concepts and techniques being demonstrated. Line numbers may be included for reference in the accompanying text.

```
# This is an example of a code block
import leafmap
m = leafmap.Map()
m.add_basemap("OpenTopoMap") # add a basemap to the map
m
```

Command Line Instructions: Commands to be entered at the command line or terminal are shown with a `$` prompt:

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

Downloading the Code Examples

All code examples, datasets, and supplementary materials for this book are freely available on GitHub:

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

To download the materials, you can use one of the following methods:

- **Clone the repository** (if you have Git installed):

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

- **Download as ZIP** (if you prefer not to use Git):
 - Visit the GitHub repository page
 - Click the green **Code** button
 - Select **Download ZIP**
 - Extract the files to your preferred location
- **Browse individual files** online through the GitHub interface if you only need specific examples

The repository is regularly updated with corrections, improvements, and additional examples. Check back periodically for updates, or **watch** the repository on GitHub to be notified of changes.

If you find errors in the code or have suggestions for improvements, please open an issue or submit a pull request on GitHub. Community contributions help make this resource better for everyone.

Video Tutorials

Complementing the written content, this book is supported by a comprehensive series of video tutorials that walk through key concepts and provide additional examples:

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

The videos are designed to complement, not replace, the written material. They're particularly helpful for:

- Visual learners who benefit from seeing code being written and executed
- Understanding complex concepts through multiple explanations
- Learning about the development workflow and best practices
- Seeing how to approach problems and debug issues

The playlist is organized to follow the book's structure. You can watch them in order as you progress through the book, or jump to specific topics as needed.

The videos were created in Fall 2024 when I was teaching the [Introduction to GIS Programming](https://geog-312.gishub.org)¹ course at the University of Tennessee. Although the course has concluded, the videos remain relevant and can be used as a reference for the book. Additional videos will be added in the future.

¹<https://geog-312.gishub.org>

Get in Touch

I welcome feedback, questions, and suggestions from readers. Your input helps improve the book and makes it more useful for the geospatial programming community.

For book-related questions and discussions:

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

Types of feedback that are particularly helpful:

- Errors or unclear explanations in the text or code
- Suggestions for additional examples or use cases
- Ideas for new topics or chapters
- Reports of compatibility issues with different operating systems or library versions
- Success stories of how you've applied the techniques from the book

Acknowledgments

This book would not have been possible without the contributions and support of many individuals and the broader open-source geospatial community.

The Open-Source Community: This book builds upon the incredible work of countless open-source developers who have created and maintained the Python geospatial ecosystem. Special thanks to the developers and maintainers of NumPy, Pandas, GeoPandas, Rasterio, Xarray, Rioxarray, Folium, ipyleaflet, MapLibre, GDAL, and the many other libraries that make geospatial programming accessible.

Students and Colleagues: The questions, challenges, and insights from students in my geospatial programming courses at the University of Tennessee have shaped the approach and content of this book. Their feedback on what works and what doesn't has been invaluable in creating materials that truly serve learners.

Research Collaborators: Colleagues and collaborators in the geospatial research community have provided real-world use cases, datasets, and problem scenarios that inform the practical examples throughout the book.

Family and Friends: Writing a technical book requires significant time and focus. I'm grateful for the patience and support of family and friends who understood the many evenings and weekends dedicated to this project.

The Broader GIS Community: The geospatial field is built on a foundation of sharing knowledge and tools. This book is part of that tradition, and I'm honored to contribute to the resources available for learning geospatial programming.

This book was written using [MyST Markdown](https://mystmd.org)² and compiled using [Typst](https://github.com/typst/typst)³ with the [min-book](https://github.com/mayconfmelo/min-book)⁴ template. Credits to developers and maintainers of the Typst and MyST Markdown projects. Special thanks to [Maycon F. Melo](https://github.com/mayconfmelo)⁵ for the min-book template and their help with customizing the template for this book.

²<https://mystmd.org>

³<https://github.com/typst/typst>

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

⁵<https://github.com/mayconfmelo>

Any errors or omissions in this book remain my responsibility. I'm committed to addressing issues and improving the content based on reader feedback.

About the Author

Dr. Qiusheng Wu is an Associate Professor and the Director of Graduate Studies in the Department of Geography & Sustainability at the University of Tennessee, Knoxville. He also serves as an Amazon Scholar. Dr. Wu's research focuses on geospatial data science and open-source software development, with an emphasis on leveraging big geospatial data and cloud computing to study environmental change, particularly surface water and wetland inundation dynamics. He is the creator of several widely used open-source Python packages, including [geemap](https://geemap.org)⁶, [leafmap](https://leafmap.org)⁷, [segment-geospatial](https://samgeo.gishub.org)⁸, and [geoai](https://opengeoai.org)⁹, which support advanced geospatial analysis and interactive visualization. His open-source work is available at the [Open Geospatial Solutions](https://github.com/opengeos)¹⁰ on GitHub.

Licensing and Copyright

This book embraces the principles of open science and open education. To support transparency, learning, and reuse, the **code examples** in this book are released under a [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/) license. This means you are free to copy, modify, and distribute the code, even for commercial purposes, as long as appropriate credit is given.

Please attribute code usage by citing the book or linking to the GitHub repository:

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

While the code is freely available, the **text, figures, and images** in this book are **copyrighted** by the author and may not be reproduced, redistributed, or modified without explicit permission. This includes all written content, custom diagrams, and embedded visualizations unless otherwise noted.

If you wish to reuse or adapt any non-code material from the book—for example, for teaching, presentations, or publications—please contact the author to request permission.

This dual licensing approach helps balance open access to learning materials with the protection of original creative work. Thank you for respecting these terms and supporting the open-source geospatial community.

⁶<https://geemap.org>

⁷<https://leafmap.org>

⁸<https://samgeo.gishub.org>

⁹<https://opengeoai.org>

¹⁰<https://github.com/opengeos>

Introduction to GIS Programming

Unlock the Power of Python for Geospatial Analysis and Visualization

This book offers a comprehensive, hands-on introduction to the world of geospatial analysis using open-source Python packages. Designed for learners of all levels, this book breaks down the complexities of Geographic Information Systems (GIS) into clear, actionable steps, making it ideal for students, researchers, professionals, and self-learners interested in mastering spatial data programming.

What sets this book apart is its step-by-step, example-driven approach. Beginning with foundational Python programming skills, you'll build your understanding gradually, progressing to advanced techniques in geospatial analysis. The content is designed to be interactive, with real-world datasets and practical exercises that allow you to apply your skills immediately. You'll work through a variety of projects, from basic spatial data manipulation to building interactive dashboards and cloud-based geospatial applications. Whether you're looking to automate GIS workflows, develop geospatial web applications, or deepen your spatial data science skills, this book will guide you through the entire process with clarity and confidence.

What You'll Learn:

- **How to set up your development environment using conda, VS Code, Git, Docker, and cloud-computing tools**
- **Core Python programming skills, from variables and data structures to file I/O and Pandas**
- **Vector and raster data processing, interactive mapping, 3D visualization, and geoprocessing**
- **Cloud-based geospatial computing with Google Earth Engine and advanced topics like hyperspectral data and spatial SQL**

Dr. Qiusheng Wu is a leading researcher and educator in open geospatial data science. He is the creator of several widely used open-source Python packages, including geemap, leafmap, segment-geospatial, and geoai. Connect with him online:

- <https://x.com/giswqs>
- <https://linkedin.com/in/giswqs>
- <https://youtube.com/@giswqs>
- <https://github.com/giswqs>



Scan the QR code to visit the
book website and download
code examples from
<https://gispro.gishub.org>