

Räumliche Datenverwaltung mit DuckDB

Von den SQL-Grundlagen zur
fortgeschrittenen Geodatenanalyse



Qiusheng Wu

Räumliche Datenverwaltung mit DuckDB

Von den SQL-Grundlagen zur fortgeschrittenen
Geodatenanalyse

Qiusheng Wu
2026

Contents

Vorwort	1
Einführung	3
Für wen dieses Buch ist	4
Was dieses Buch behandelt	5
Das Beste aus diesem Buch herausholen	6
In diesem Buch verwendete Konventionen	7
Herunterladen der Code-Beispiele	8
Video-Tutorials und ergänzende Ressourcen	8
Community und Feedback	9
Danksagungen	9
Über den Autor	10
Lizenzierung und Urheberrecht	10
I: DuckDB-Grundlagen	13
1. Erste Schritte mit DuckDB	15
1.1. Einführung	15
1.2. Lernziele	15
1.3. Was DuckDB von traditionellen Datenbanken unterscheidet	15
1.4. Wann (und wann nicht) DuckDB für räumliche Arbeit verwenden	17
1.5. Installation der DuckDB CLI und Ausführung Ihrer ersten Abfrage	19
1.6. Installation des DuckDB Python-Clients	20
1.7. Installation von Visual Studio Code	24
1.8. Verwendung der DuckDB UI	27
1.9. Installation der DBeaver SQL IDE	28
1.10. Wichtige Erkenntnisse	32
1.11. Übungen	33
2. Grundlegendes SQL für räumliche Analysen	36
2.1. Einleitung	36
2.2. Lernziele	36
2.3. Beispieldatensätze	37
2.4. Einrichten der Umgebung	37
2.5. Verbindung zu DuckDB herstellen	39
2.6. Installieren von Erweiterungen	39
2.7. Lesen von CSV-Dateien von URLs	39
2.8. Erstellen von Tabellen für bessere Leistung	40
2.9. Die SQL SELECT-Anweisung	42
2.10. Filtern von Daten mit der WHERE-Klausel	49
2.11. Mustervergleich mit LIKE	51
2.12. Der IN-Operator	52
2.13. Der BETWEEN-Operator	53
2.14. Kombinieren von Daten mit SQL-Joins	54
2.15. Query Plans und Performance	61
2.16. Aggregieren von Daten für zusammenfassende Statistiken	61
2.17. Bedingte Anweisungen	65
2.18. Ergebnisse speichern	66
2.19. Erweiterte SQL-Features für räumliche Analysen	67
2.20. SQL-Kommentare und Dokumentation	72

2.21. Wichtigste Erkenntnisse	72
2.22. Übungen	74
3. DuckDB Python-Integration	77
3.1. Einführung	77
3.2. Lernziele	77
3.3. Beispieldatensätze	78
3.4. Installation und Setup	79
3.5. Installation und Laden von Erweiterungen	80
3.6. Lesen von Daten aus mehreren Quellen	81
3.7. Nahtlose Integration mit Pandas DataFrames	84
3.8. Polars-Interoperabilität	88
3.9. Ergebniskonvertierung und Ausgabeformate	88
3.10. Schreiben von Daten auf die Festplatte	91
3.11. Persistenter Speicher und Datenbankdateien	91
3.12. Vorbereitete Anweisungen und Parameter	94
3.13. Wichtigste Erkenntnisse	95
3.14. Übungen	96
II: Räumliche Datenoperationen	103
4. Laden räumlicher Datenformate	105
4.1. Einleitung	105
4.2. Lernziele	106
4.3. Beispieldatensätze	107
4.4. Installation und Einrichtung	107
4.5. Installieren und Laden von Extensions	108
4.6. Herunterladen von Beispieldaten	109
4.7. Laden von CSV-Dateien mit Koordinaten	109
4.8. Laden von JSON-Dateien	113
4.9. Pandas DataFrames direkt abfragen	115
4.10. Laden von Parquet-Dateien für Leistung	117
4.11. Laden von GeoJSON-Dateien mit räumlichen Geometrien	119
4.12. Laden von Shapefiles in moderne Workflows	125
4.13. Laden von GeoParquet für cloud-native räumliche Analyse	127
4.14. Strategien für Datenlade-Leistung	130
4.15. Behebung häufiger Probleme beim Laden von Daten	130
4.16. Wichtige Erkenntnisse	131
4.17. Übungen	133
5. Exportieren und Konvertieren räumlicher Daten	137
5.1. Einführung	137
5.2. Lernziele	138
5.3. Beispieldatensätze	139
5.4. Installation und Einrichtung	139
5.5. Installieren und Laden von Erweiterungen	140
5.6. Laden von Beispieldaten	141
5.7. Exportieren in Pandas DataFrames	143
5.8. Exportieren in CSV-Dateien	147
5.9. Exportieren in JSON-Dateien	151
5.10. Exportieren in Excel-Dateien	153
5.11. Exportieren in Parquet-Dateien	156

5.12. Exportieren in das GeoJSON-Format	158
5.13. Exportieren in das Shapefile-Format	162
5.14. Exportieren in das GeoPackage-Format	163
5.15. Wichtigste Erkenntnisse	165
5.16. Übungen	166
6. Geometrieverarbeitung und -funktionen	171
6.1. Einführung	171
6.2. Lernziele	171
6.3. Beispieldatensätze	172
6.4. Installation und Einrichtung	173
6.5. Verbindung zu DuckDB herstellen und Extensions laden	174
6.6. Geometriertypen verstehen	176
6.7. Arbeiten mit Points	179
6.8. Arbeiten mit LineStrings	182
6.9. Arbeiten mit Polygons	184
6.10. Arbeiten mit Collections	186
6.11. Visualisierung von NYC-Raumdaten	187
6.12. Geometrieverarbeitung	190
6.13. Geometriegültigkeit und Robustheit	194
6.14. Wichtigste Erkenntnisse	196
6.15. Übungen	198
7. Räumliche Abfragen und Beziehungen	203
7.1. Einführung	203
7.2. Lernziele	204
7.3. Beispieldatensätze	204
7.4. Installation und Setup	205
7.5. Verbindung zu DuckDB herstellen und Erweiterungen laden	206
7.6. Räumliche Beziehungen verstehen	207
7.7. Testen der geometrischen Identität	207
7.8. Topologische Beziehungen	210
7.9. Entfernungsbasierte Beziehungen	218
7.10. Filtern nach Nähe-Schwellwert	220
7.11. Nächste-Nachbarn-Abfragen	221
7.12. Wichtigste Erkenntnisse	221
7.13. Übungen	223
8. Erweiterte räumliche Joins	226
8.1. Einführung	226
8.2. Lernziele	227
8.3. Beispieldatensätze	228
8.4. Installation	228
8.5. Bibliotheksimport und Konfiguration	228
8.6. Verbindung zu DuckDB herstellen	229
8.7. Intersection Joins	230
8.8. Distance Within Joins	235
8.9. Erweiterte Joins	239
8.10. Koordinatensystem-Transformationen	241
8.11. Referenz für räumliche Beziehungsfunktionen	243
8.12. Wichtige Erkenntnisse	244

8.13. Übungen	246
9. Interaktive Datenvisualisierung	251
9.1. Einführung	251
9.2. Lernziele	252
9.3. Beispieldatensätze	253
9.4. Installation und Einrichtung	253
9.5. Download von Beispieldaten	253
9.6. Verbindung zu DuckDB und Laden von Extensions	254
9.7. Visualisierung von Punktdaten	255
9.8. Visualisierung von Liniendaten	258
9.9. Visualisierung von Polygondaten	261
9.10. Wichtigste Erkenntnisse	265
9.11. Übungen	267
10. Arbeiten mit Vektorkacheln und PMTiles	271
10.1. Einleitung	271
10.2. Lernziele	272
10.3. Beispieldatensätze	273
10.4. Installation und Einrichtung	274
10.5. Visualisierung von Vector Tiles direkt aus Dateien	274
10.6. Konvertierung von Vektordaten zu PMTiles	279
10.7. Visualisierung von PMTiles	281
10.8. Wichtige Erkenntnisse	290
10.9. Übungen	292
III: Praxisnahe Geodatenanalyse	297
11. Analyse des US National Wetlands Inventory	299
11.1. Einführung	299
11.2. Lernziele	300
11.3. In diesem Kapitel verwendete Datensätze	301
11.4. Verständnis der Datensatzquelle	303
11.5. Zugriff auf Feuchtgebietsdaten mit DuckDB	305
11.6. Visualisierung von Feuchtgebietsverteilungen	309
11.7. Nationale Feuchtgebietsanalyse	313
11.8. Wichtige Erkenntnisse	326
11.9. Übungen	329
12. Analyse globaler Gebäude-Footprints	332
12.1. Einführung	332
12.2. Lernziele	333
12.3. Über den Datensatz	334
12.4. Installation und Einrichtung	337
12.5. Installation und Laden von Erweiterungen	337
12.6. Erkundung verfügbarer Daten	337
12.7. Regionale Analyse mit Bounding-Box-Filterung	341
12.8. Multi-Regionen-Vergleich	351
12.9. Filterung nach Datenquelle und Qualitätsbewertung	356
12.10. Grid-basierte räumliche Aggregation	361
12.11. Aggregation von Gebäudedaten mit H3-Hexagonal-Grids	366
12.12. Wichtige Erkenntnisse	378
12.13. Übungen	380

13. Analyse der NYC-Taxidaten	383
13.1. Einführung	383
13.2. Lernziele	384
13.3. Über den Datensatz	385
13.4. Installation	387
13.5. Bibliotheken-Import	387
13.6. Installation und Laden von Extensions	387
13.7. Laden von Taxidaten	388
13.8. Zeitliche Analyse	392
13.9. Laden von Taxi-Zonen-Lookup-Daten	399
13.10. Räumliche Analyse	402
13.11. Analyse von Fahrtströmen	410
13.12. Zahlungs- und Wirtschaftsanalyse	415
13.13. Fahrgastverhalten-Analyse	419
13.14. Multi-Monats-Analyse	420
13.15. Visualisierung	422
13.16. Tipps zur Leistungsoptimierung	428
13.17. Wichtigste Erkenntnisse	429
13.18. Übungen	432
14. Entwicklung interaktiver Dashboards mit Voilà und Solara	434
14.1. Einleitung	434
14.2. Lernziele	436
14.3. Installation von Voilà und Solara	436
14.4. Einführung in Hugging Face Spaces	436
14.5. Erstellen einer einfachen Voilà-Anwendung	438
14.6. Erstellen einer fortgeschrittenen Webanwendung mit Solara	448
14.7. Wichtigste Erkenntnisse	456
14.8. Übungen	457

Vorwort

Einführung

In einer zunehmend datengesteuerten Welt ist die Fähigkeit, räumliche Informationen effektiv zu verwalten und zu analysieren, unerlässlich geworden. Von der Stadtplanung und Umweltüberwachung bis hin zu Logistik und personalisierten standortbasierten Diensten bilden räumliche Daten das Rückgrat zahlreicher Anwendungen, die unser tägliches Leben beeinflussen. Die Arbeit mit räumlichen Daten wurde jedoch oft als spezialisierter und komplexer Bereich betrachtet, der komplizierte Werkzeuge und eine steile Lernkurve erfordert.

Hier kommt DuckDB ins Spiel.

DuckDB ist eine innovative analytische Datenbank, die auf Effizienz und Benutzerfreundlichkeit ausgelegt ist. Als In-Process-OLAP-Datenbank (Online Analytical Processing) arbeitet sie direkt in Ihrer Anwendung und eliminiert die Notwendigkeit separater Server-Deployments und umständlicher Konfigurationen. Diese eingebettete Natur, gepaart mit ihrer spaltenorientierten Architektur und vektorisierten Ausführungs-Engine, macht DuckDB außergewöhnlich schnell für analytische Abfragen, selbst bei großen Datensätzen. Ursprünglich für ihre universellen Datenverarbeitungsfähigkeiten bekannt geworden, stellt DuckDBs schnell wachsendes Ökosystem (insbesondere ihre Erweiterungen für räumliche Daten) einen transformativen Wandel dar.

Dieses Buch, „***Spatial Data Management with DuckDB: From SQL Basics to Advanced Geospatial Analytics***“, zielt darauf ab, räumliche Daten zu entmystifizieren und zu zeigen, wie DuckDB jeden (von Datenanalysten und Wissenschaftlern bis hin zu Entwicklern und GIS-Fachleuten) befähigt, ihre Fähigkeiten mit beispieloser Einfachheit und Geschwindigkeit zu nutzen. Wir glauben, dass robuste räumliche Analysen zugänglich sein sollten und nicht auf teure spezialisierte Software oder komplexe Programmiersprachen beschränkt bleiben dürfen. Mit DuckDB wird diese Zugänglichkeit Realität.

Unsere Reise beginnt mit den grundlegenden Konzepten räumlicher Daten und ihrer Darstellung und legt eine solide Grundlage in SQL für die Arbeit mit Punkten, Linien und Polygonen. Während wir fortschreiten, werden Sie entdecken, wie DuckDBs native räumliche Funktionen (erweitert durch ihre PostGIS-kompatible Erweiterung) anspruchsvolle Operationen wie räumliche Joins, Pufferbildung und Nächster-Nachbar-Suchen durch elegante SQL-Abfragen ermöglichen. Wir werden verschiedene reale Anwendungen erkunden und demonstrieren, wie man räumliche Datensätze lädt, transformiert, analysiert und visualisiert, um Sie zu befähigen, bedeutungsvolle Erkenntnisse aus geografischen Informationen zu extrahieren.

Ob Sie räumliche Analysen in Ihre Daten-Pipelines integrieren, schnelle Ad-hoc-Geoabfragen durchführen oder interaktive standortbewusste Anwendungen entwickeln möchten – dieses Buch dient Ihnen als umfassender Leitfaden. Wir werden Themen behandeln, die von der Einrichtung Ihrer DuckDB-Umgebung und dem Import verschiedener räumlicher Dateiformate (wie Shapefiles, GeoJSON und GeoParquet) bis hin zur Ausführung komplexer analytischer Aufgaben und der Integration mit Visualisierungstools reichen.

Unser Ziel ist es nicht nur, Ihnen Syntax beizubringen, sondern ein Verständnis dafür zu entwickeln, warum diese Werkzeuge und Techniken leistungsstark sind. Am Ende dieses Buches werden Sie versiert darin sein, DuckDB als Ihre bevorzugte Engine für räumliches Datenmanagement und -analyse einzusetzen, neue Möglichkeiten für Ihre Projekte zu erschließen und Sie zu befähigen, informierte, räumlich bewusste Entscheidungen zu treffen.

Begleiten Sie uns, während wir in die aufregende Schnittstelle zwischen DuckDBs analytischen Fähigkeiten und der reichen Welt räumlicher Daten eintauchen. Die Zukunft zugänglicher räumlicher Analysen ist hier, und sie läuft auf DuckDB.

Für wen dieses Buch ist

Dieses Buch ist für jeden konzipiert, der mit den Komplexitäten moderner räumlicher Datenanalyse zu kämpfen hat. Wenn Sie jemals Stunden damit verbracht haben, auf den Abschluss eines räumlichen Joins zu warten, Schwierigkeiten hatten, große geografische Datensätze in den Speicher zu laden, oder sich einen einfacheren Weg gewünscht haben, die Kraft von SQL mit räumlichen Operationen zu kombinieren, ist dieses Buch für Sie.

Sie werden den größten Nutzen daraus ziehen, wenn Sie

Ein GIS-Fachmann sind, der von den Einschränkungen von Desktop-Software beim Umgang mit großen Datensätzen frustriert ist. Sie sind mit QGIS oder ArcGIS vertraut, müssen aber Millionen von Features analysieren, umfangreiche GPS-Tracks verarbeiten oder räumliche Analysen in automatisierte Workflows integrieren.

Ein Data Scientist oder Analyst sind, der häufig auf Standortdaten stößt. Sie sind mit Python und pandas vertraut, aber räumliche Daten fühlen sich oft wie ein Mysterium an. Sie möchten geografische Dimensionen in Ihre Analysen einbeziehen, ohne in komplexe GIS-Software eintauchen zu müssen.

Ein Softwareentwickler sind, der Anwendungen mit räumlichen Features entwickelt. Sie benötigen schnelle räumliche Abfragen, möchten schwere Datenbank-Infrastruktur vermeiden und bevorzugen die Arbeit mit vertrautem SQL gegenüber spezialisierten räumlichen Bibliotheken.

Ein Forscher oder Akademiker in Bereichen wie Geographie, Umweltwissenschaften oder Stadtplanung sind. Ihre Forschung umfasst große räumliche Datensätze, und Sie benötigen reproduzierbare, skalierbare Analysemethoden, die sich an wachsende Datenmengen anpassen können.

Ein Business-Intelligence-Fachmann sind, der mit standortbasierten Geschäftsdaten arbeitet. Ob Filialstandorte, Lieferrouten, Kundengebiete oder Immobilienportfolios – Sie müssen Geschäftskennzahlen mit räumlichen Erkenntnissen zusammenführen.

Wesentliche Voraussetzungen

Sie sollten vertraut sein mit:

- **Python-Programmierung**: Verständnis von Variablen, Funktionen und wie man Bibliotheken importiert (Expertise ist nicht erforderlich)
- **Datenanalyse-Konzepte**: Filtern von Datensätzen, Aggregieren von Daten und Verknüpfen von Tabellen
- **SQL-Grundlagen**: Vertrautheit mit SELECT-, WHERE- und GROUP BY-Klauseln (wir werden die räumlichen Aspekte behandeln)
- **Grundlagen räumlicher Daten**: Verständnis dafür, dass Daten einen Standort haben (Breiten-/Längengrad, Projektionen)

Hilfreicher Hintergrund (aber nicht erforderlich)

- Erfahrung mit pandas, GeoPandas oder Jupyter Notebooks
- Frühere Erfahrung mit Datenbanken oder Data Warehouses
- Vertrautheit mit GIS-Software (QGIS, ArcGIS, PostGIS)
- Kenntnisse räumlicher Dateiformate (GeoJSON, Shapefiles, Parquet)

Wenn Sie neu in der Python-Programmierung sind

Wenn Sie neu in der räumlichen Python-Programmierung sind, bietet das folgende Buch eine hervorragende Einführung sowohl in grundlegende GIS-Konzepte als auch in Python-Programmierung:

Wu, Q. (2025). *Introduction to GIS Programming: A Practical Python Guide to Open Source Geospatial Tools*. Independently published. ISBN 979-8286979455. <https://www.amazon.com/dp/B0FFW34LL3>

Was dieses Buch behandelt

Dieses Buch bietet eine strukturierte Reise von SQL-Grundlagen zu fortgeschrittenen räumlichen Analysen und vermittelt Ihnen praktische Fähigkeiten durch reale Beispiele. Jedes Kapitel schreitet von einfachen Abfragen zu komplexen räumlichen Analysen voran und baut Ihre Expertise im modernen räumlichen Datenmanagement auf.

Teil I: DuckDB-Grundlagen (*Kapitel 1-3*)

Beherrschen Sie die wesentlichen Konzepte, die allen nachfolgenden Inhalten zugrunde liegen:

- **Kapitel 1: Erste Schritte mit DuckDB:** Installation, erste Abfragen und Einblicke, wie DuckDB räumliche Analysen revolutioniert.
- **Kapitel 2: Grundlegendes SQL für räumliche Analysen:** Wichtige SQL-Muster zum Filtern, Aggregieren, Verknüpfen und Optimieren von Abfragen, zugeschnitten auf räumliche Daten.
- **Kapitel 3: DuckDB Python-Integration:** Kombinieren Sie die Kraft von SQL mit der Flexibilität von pandas, um einen nahtlosen räumlichen Analyse-Workflow zu schaffen.

Am Ende von Teil I werden Sie selbstbewusst räumliche Datensätze abfragen und DuckDB in jede Python-basierte Analyse-Pipeline integrieren können.

Teil II: Räumliche Datenoperationen (*Kapitel 4-10*)

Tauchen Sie in das zentrale räumliche Toolkit ein und decken Sie alles ab, vom Laden von Daten bis zu fortgeschrittenen Analysen:

- **Kapitel 4: Laden räumlicher Datenformate:** Importieren Sie verschiedene Formate, einschließlich CSV-Koordinaten, GeoJSON von APIs, massive Shapefiles und Cloud-gehostetes GeoParquet.
- **Kapitel 5: Exportieren und Konvertieren räumlicher Daten:** Transformieren Sie Ihre Ergebnisse in jedes Format, das Ihre Stakeholder benötigen.
- **Kapitel 6: Geometrieoperationen und -funktionen:** Erstellen, messen und transformieren Sie räumliche Features mithilfe von SQL-Funktionen.
- **Kapitel 7: Räumliche Abfragen und Beziehungen:** Beherrschen Sie räumliche Äquivalente von Tabellen-Joins: Enthaltensein, Schnittmenge und Nähe.
- **Kapitel 8: Erweiterte räumliche Joins:** Kombinieren Sie Datensätze nach Standort statt nach IDs, was die Essenz räumlicher Analysen ist.
- **Kapitel 9: Interaktive Datenvisualisierung:** Erstellen Sie überzeugende Karten und Diagramme, die die räumlichen Narrative Ihrer Daten effektiv kommunizieren.
- **Kapitel 10: Arbeiten mit Vektorkacheln und PMTiles:** Implementieren Sie interaktive Karten, die Millionen von Features reibungslos handhaben können.

Am Ende von Teil II werden Sie geschickt darin sein, jedes räumliche Datenformat zu verwalten, komplexe Operationen auszuführen und professionelle Visualisierungen zu erstellen.

Teil III: Reale räumliche Analysen (*Kapitel 11-14*)

Erkunden Sie vier umfassende Fallstudien mit großen, realen Datensätzen:

- **Kapitel 11: Analyse des US National Wetlands Inventory:** Führen Sie Umweltanalysen in allen 50 Bundesstaaten durch und verarbeiten Sie Millionen von Feuchtgebietspolygonen.
- **Kapitel 12: Analyse globaler Gebäude-Footprints:** Analysieren Sie städtische Daten mithilfe globaler Gebäude-Footprints von Overture Maps.
- **Kapitel 13: Analyse der NYC-Taxidaten:** Entdecken Sie räumlich-zeitliche Muster aus Hunderten von Millionen von Taxifahrten und enthüllen Sie Einblicke in urbane Mobilität.
- **Kapitel 14: Entwicklung interaktiver Dashboards mit Voilà und Solara:** Erstellen und implementieren Sie Webanwendungen, die Ihre Analysen für Stakeholder zugänglich machen.

Am Ende von Teil III werden Sie Portfolio-würdige Projekte haben, die Ihre fortgeschrittenen räumlichen Analysefähigkeiten demonstrieren.

Übergreifende Themen

- **Leistung im großen Maßstab:** Techniken, die effektiv sind, ob Sie Tausende oder Millionen räumlicher Features handhaben.
- **Cloud-native Workflows:** Verarbeiten Sie Daten direkt von S3 und integrieren Sie nahtlos mit modernen Daten-Stacks.
- **Reproduzierbare Analysen:** Teilbarer Code und Methoden, die versionskontrolliert und in der Produktion eingesetzt werden können.
- **Reale Datenherausforderungen:** Bewältigen Sie Projektionsprobleme, fehlende Werte und Datenqualitätsprobleme.
- **Integrationsmuster:** Kombinieren Sie DuckDB mit dem breiteren Python-Ökosystem für räumliche Daten für erweiterte Funktionalität.

Was dieses Buch anders macht

Im Gegensatz zu theoretischen Diskussionen oder Tool-spezifischen Tutorials betont dieses Buch *das Lösen realer Probleme*. Jede Technik ist in tatsächlichen analytischen Herausforderungen verwurzelt, mit realen Datensätzen demonstriert und in klaren Begriffen erklärt, wann und warum man sie verwenden sollte.

Das Beste aus diesem Buch herausholen

Um Ihr Lernerlebnis mit diesem Buch zu maximieren, beachten Sie die folgenden Empfehlungen:

Richten Sie eine angemessene Entwicklungsumgebung ein: Installieren Sie Python und die erforderlichen Bibliotheken wie in Kapitel 1 beschrieben. Eine gut konfigurierte Umgebung spart Ihnen Zeit und Frustration während Ihrer Lernreise. Erwägen Sie die Verwendung von conda oder uv zur Verwaltung Ihrer Python-Pakete, da dies die Installation räumlicher Bibliotheken vereinfacht.

Folgen Sie den Code-Beispielen: Dieses Buch ist für die Interaktion konzipiert. Lesen Sie nicht nur den Code; tippen Sie ihn ein, führen Sie ihn aus und experimentieren Sie mit Modifikationen. Verständnis kommt durch Übung, und jedes Beispiel baut Fähigkeiten auf, die Sie später benötigen werden.

Bearbeiten Sie die Übungen: Jedes Kapitel enthält Übungen, die darauf ausgelegt sind, die Konzepte zu festigen, die Sie gelernt haben. Diese sind keine optionalen Extras; sie sind ein integraler Bestandteil des

Lernprozesses. Beginnen Sie mit den geführten Übungen und fordern Sie sich dann mit eigenen Projekten heraus.

Verwenden Sie reale Daten: Während das Buch Datensätze für Beispiele und Übungen bereitstellt, versuchen Sie, die Techniken auf Daten aus Ihrem eigenen Bereich oder Ihren Interessen anzuwenden. Dies wird Ihnen helfen zu verstehen, wie die Konzepte auf reale Szenarien anwendbar sind und Vertrauen in Ihre Fähigkeiten aufzubauen.

Erstellen Sie Projekte: Während Sie durch das Buch fortschreiten, erwägen Sie die Arbeit an einem persönlichen Projekt, das Sie interessiert. Dies könnte die Analyse von Daten aus Ihrer Forschung, das Erstellen von Karten für Ihre Gemeinschaft oder das Lösen eines Problems sein, auf das Sie bei Ihrer Arbeit gestoßen sind.

Seien Sie geduldig mit sich selbst: Programmierung kann frustrierend sein, besonders wenn Sie lernen. Erwarten Sie, auf Fehler zu stoßen, Zeit mit Debugging zu verbringen und sich gelegentlich festzufahren. Dies ist normal und Teil des Lernprozesses. Machen Sie Pausen, wenn nötig, und denken Sie daran, dass Expertise sich allmählich durch konsequente Übung entwickelt. Wenn Sie nicht weiterkommen, zögern Sie nicht, auf dem GitHub-Repository des Buches um Hilfe zu bitten.

Üben Sie weiter: Die Fähigkeiten in diesem Buch erfordern regelmäßige Übung, um sie zu erhalten und zu entwickeln. Nehmen Sie sich regelmäßig Zeit für die Arbeit an räumlichen Programmierprojekten, auch wenn es kleine sind.

In diesem Buch verwendete Konventionen

Dieses Buch verwendet mehrere Konventionen, um Ihnen bei der Navigation durch den Inhalt und beim Verständnis der Code-Beispiele zu helfen:

Code-Formatierung: Aller Python-Code erscheint in Monospace-Schrift innerhalb von Code-Blöcken. Wenn Code innerhalb des regulären Textes erscheint, ist er so formatiert. Datei- und Verzeichnisnamen sind ebenfalls in Monospace-Schrift formatiert.

Code-Beispiele: Die meisten Code-Beispiele sind vollständig und lauffähig. Sie enthalten Kommentare, die die wichtigsten Konzepte und Techniken erklären, die demonstriert werden. Zeilenummern können zur Referenz im begleitenden Text enthalten sein.

```
# Dies ist ein Beispiel für einen Code-Block
import leafmap
m = leafmap.Map()
m.add_basemap("OpenTopoMap") # Basiskarte zur Karte hinzufügen
m
```

SQL-Stilrichtlinie: Aus Gründen der Konsistenz und Lesbarkeit folgen SQL-Beispiele diesen Mustern:

- **Schlüsselwörter in GROSSBUCHSTABEN:** SELECT , FROM , WHERE , JOIN
- **Funktionsnamen erhalten die Groß-/Kleinschreibung:** ST_Area() , read_csv_auto()
- **Tabellen- und Spaltennamen in Kleinbuchstaben:** cities , population
- **Einrückung für Lesbarkeit:** Mehrzeilige Abfragen sind für Klarheit formatiert

```
SELECT name, ST_Area(geometry) as area
FROM neighborhoods
```

```
WHERE borough = 'Manhattan'  
ORDER BY area DESC;
```

Kommandozeilen-Anweisungen: Befehle, die an der Kommandozeile oder im Terminal eingegeben werden sollen, werden mit einem \$ -Prompt angezeigt (geben Sie das \$ -Symbol selbst nicht ein):

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

Herunterladen der Code-Beispiele

Alle Code-Beispiele, Datensätze und ergänzenden Materialien für dieses Buch sind kostenlos auf GitHub verfügbar:

<https://github.com/giswqs/duckdb-spatial>

Um die Materialien herunterzuladen, können Sie eine der folgenden Methoden verwenden:

- **Repository klonen** (wenn Sie Git installiert haben):

```
$ git clone https://github.com/giswqs/duckdb-spatial.git
```

- **Als ZIP herunterladen** (wenn Sie Git nicht verwenden möchten):

- Besuchen Sie die GitHub-Repository-Seite
- Klicken Sie auf die grüne **Code**-Schaltfläche
- Wählen Sie **Download ZIP**
- Extrahieren Sie die Dateien an Ihren bevorzugten Ort

- **Einzelne Dateien online durchsuchen** über die GitHub-Oberfläche, wenn Sie nur bestimmte Beispiele benötigen

Das Repository wird regelmäßig mit Korrekturen, Verbesserungen und zusätzlichen Beispielen aktualisiert. Schauen Sie regelmäßig vorbei, um Updates zu erhalten, oder **beobachten** Sie das Repository auf GitHub, um über Änderungen benachrichtigt zu werden.

Wenn Sie Fehler im Code finden oder Verbesserungsvorschläge haben, öffnen Sie bitte ein Issue oder reichen Sie einen Pull Request auf GitHub ein. Community-Beiträge helfen dabei, diese Ressource für alle besser zu machen.

Video-Tutorials und ergänzende Ressourcen

Ergänzend zum geschriebenen Inhalt wird dieses Buch durch eine umfassende Serie von Video-Tutorials unterstützt, die wichtige Konzepte durchgehen und zusätzliche Beispiele bieten:

<https://tinyurl.com/duckdb-spatial-videos>

Die Videos sind als Ergänzung, nicht als Ersatz für das geschriebene Material konzipiert. Sie sind besonders hilfreich für:

- Visuelle Lerner, die davon profitieren, Code geschrieben und ausgeführt zu sehen
- Verständnis komplexer Konzepte durch mehrere Erklärungen

- Lernen über den Entwicklungs-Workflow und Best Practices
- Sehen, wie man Probleme angeht und Probleme debuggt

Die Playlist ist so organisiert, dass sie der Struktur des Buches folgt. Sie können sie der Reihe nach ansehen, während Sie durch das Buch fortschreiten, oder zu bestimmten Themen nach Bedarf springen.

Die Videos wurden im Herbst 2023 erstellt, als ich den Kurs **Spatial Data Management**¹ an der University of Tennessee unterrichtete. Obwohl der Kurs abgeschlossen ist, bleiben die Videos relevant und können als Referenzen für das Buch verwendet werden. Zusätzliche Videos werden in Zukunft hinzugefügt.

Community und Feedback

Ich freue mich über Feedback, Fragen und Vorschläge von Lesern. Ihr Input hilft dabei, das Buch zu verbessern und es für die räumliche Programmier-Community nützlicher zu machen.

Für buchbezogene Fragen und Diskussionen:

- GitHub Issues: <https://github.com/giswqs/duckdb-spatial/issues>
- GitHub Discussions: <https://github.com/giswqs/duckdb-spatial/discussions>

Arten von Feedback, die besonders hilfreich sind:

- Fehler oder unklare Erklärungen im Text oder Code
- Vorschläge für zusätzliche Beispiele oder Anwendungsfälle
- Ideen für neue Themen oder Kapitel
- Berichte über Kompatibilitätsprobleme mit verschiedenen Betriebssystemen oder Bibliotheksversionen
- Erfolgsgeschichten darüber, wie Sie die Techniken aus dem Buch angewendet haben

Danksagungen

Dieses Buch existiert dank der Beiträge vieler Einzelpersonen und der breiteren Open-Source-Geodaten-Community.

Zunächst möchte ich dem **DuckDB-Entwicklungsteam** dafür danken, dass sie ein so außergewöhnliches Datenbanksystem geschaffen haben. Ihr Engagement für Leistung, Einfachheit und Open-Source-Prinzipien hat räumliche Analysen einem viel breiteren Publikum zugänglich gemacht. Besondere Anerkennung gebührt den Teammitgliedern, die zu DuckDBs räumlichen Fähigkeiten beigetragen haben.

Ich bin meinen **Kollegen und Studenten** an der University of Tennessee dankbar, die Feedback gegeben, Beispiele getestet und dazu beigetragen haben, den Inhalt durch den Kurs **Spatial Data Management** zu verfeinern. Ihre Fragen und Einsichten haben dieses Buch viel stärker gemacht.

Die **Open-Source-Geodaten-Community** verdient besondere Anerkennung. Projekte wie GDAL, GeoPandas, Shapely und unzählige andere bilden die Grundlage, die moderne räumliche Analysen möglich macht. Der kollaborative Geist dieser Community inspiriert weiterhin meine Arbeit.

Danke an die **frühen Leser**, die Feedback zu Entwurfskapiteln gegeben und dabei geholfen haben, Bereiche zu identifizieren, die Klärung oder Verbesserung benötigten. Ihre unterschiedlichen Perspektiven (von erfahrenen GIS-Fachleuten bis hin zu Neulingen in der Datenwissenschaft) haben dazu beigetragen, dass dieses Buch seiner Zielgruppe dient.

¹<https://geog-414.gishub.org>

Ich möchte **meiner Familie** für ihre Geduld und Unterstützung während der vielen Abende und Wochenenden danken, die ich mit dem Schreiben verbracht habe. Ihr Verständnis und ihre Ermutigung haben dieses Projekt möglich gemacht.

Schließlich danke ich **Ihnen**, dem Leser, für Ihr Interesse an räumlicher Datenanalyse und Open-Source-Tools. Es sind Praktiker wie Sie, die Innovation vorantreiben und das Geodatenfeld zu einem so aufregenden Arbeitsplatz machen.

Wenn Ihnen dieses Buch dabei hilft, räumliche Probleme effektiver zu lösen, dann hat sich all die Mühe gelohnt.

Über den Autor

Dr. Qiusheng Wu ist Associate Professor im Department of Geography & Sustainability an der University of Tennessee, Knoxville. Er ist auch Amazon Scholar. Dr. Wus Forschung konzentriert sich auf die Weiterentwicklung von Open-Source-Geodaten-Analysen durch Cloud Computing und GeoAI. Er ist der Schöpfer und Betreuer mehrerer weit verbreiteter Open-Source-Python-Pakete, darunter [Geemap](#)², [Leafmap](#)³, [SAMGeo](#)⁴ und [GeoAI](#)⁵, die Cloud-basierte räumliche Plattformen mit KI-gestützter Analyse und Visualisierung integrieren. Dr. Wus Arbeit verbindet Fernerkundung, Erdbeobachtung und künstliche Intelligenz, um großmaßstäbliche räumliche Daten für Forscher, Pädagogen und Praktiker weltweit zugänglicher, reproduzierbarer und intelligenter zu machen. Seine Open-Source-Projekte finden Sie auf GitHub unter <https://github.com/opengeos>.

Lizenzierung und Urheberrecht

Dieses Buch umfasst die Prinzipien offener Wissenschaft und offener Bildung. Um Transparenz, Lernen und Wiederverwendung zu unterstützen, werden die **Code-Beispiele** in diesem Buch unter einer [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](#)-Lizenz veröffentlicht. Dies bedeutet, dass Sie den Code frei kopieren, modifizieren und verbreiten können, auch für kommerzielle Zwecke, solange eine angemessene Quellenangabe erfolgt.

Bitte geben Sie die Code-Nutzung an, indem Sie das Buch zitieren oder auf das GitHub-Repository verlinken:

Wu, Q. (2025). *Spatial Data Management with DuckDB: From SQL Basics to Advanced Geospatial Analytics*. Independently published. PDF edition ISBN 979-8993859705; Print edition ISBN 979-8274710572. <https://duckdb.gishub.org>

Während der Code frei verfügbar ist, sind der **Text, Abbildungen und Bilder** in diesem Buch **urheberrechtlich geschützt** durch den Autor und dürfen ohne ausdrückliche Genehmigung nicht reproduziert, weiterverbreitet oder modifiziert werden. Dies umfasst alle geschriebenen Inhalte, benutzerdefinierten Diagramme und eingebetteten Visualisierungen, sofern nicht anders angegeben.

Wenn Sie nicht-Code-Material aus dem Buch wiederverwenden oder anpassen möchten (zum Beispiel für Lehre, Präsentationen oder Publikationen), kontaktieren Sie bitte den Autor, um eine Genehmigung zu beantragen.

²<https://geemap.org>

³<https://leafmap.org>

⁴<https://samgeo.gishub.org>

⁵<https://opengeoai.org>

Dieser duale Lizenzierungsansatz hilft dabei, offenen Zugang zu Lernmaterialien mit dem Schutz originärer kreativer Arbeit in Einklang zu bringen. Vielen Dank, dass Sie diese Bedingungen respektieren und die Open-Source-Geodaten-Community unterstützen.