

農研機構
NARO 農業・食品産業技術総合研究機構

PostGIS入門の入門

農研機構
 (農業・食品産業技術総合研究機構)
 寺元郁博

NARO

※ 農研機構 (のうけんきこう) は、国立研究開発法人 農業・食品産業技術総合研究機構のコミュニケーションネーム (通称) です。

1

はじめに

- **PostGIS** (<http://postgis.net/>) の**ごく初歩**の話
 - 入門編でも50分では終わらない
- PostGIS/GISは**言うほど恐くない**
- 怖いところ = **GIS特有の話**
 - 地理空間データ
 - 座標参照系
 - 経度緯度データにはGEOGRAPHY型
- **実際に処理をやってみる**
- 参照情報

2

自己紹介

- 農業・食品産業技術総合研究機構 23年目
 - 2000年 農林水産省中国農業試験場
 - 広島県福山市
 - 農研機構は2001年発足
 - 2017年 つくば市へ異動
- 農研機構 - <https://www.naro.go.jp/>

3

GISとは

- PostGIS = PostgreSQL + GIS なので
- GIS=Geographical Information System の略
 - **地理空間データを扱うソフトウェア全般**
 - カテゴリ(「表計算」とか)ぐらいの気持ちで
 - **地理空間データ: 地球上の実際の位置とリンクしたデータ**
- GISは、出自によって**地図系**、**リモセン系**(人工衛星からの写真で分析)、**CAD系**(土木工事の設計)に大別

4

PostGISとは

- 地理空間データ用の**エクステンション**
- **地理空間データを扱う場面**どこでも利用可
 - 文字列をキーに地理空間データを検索
 - 例: 土地テーブルで、地目(文字列)が「田」の土地を抽出
 - 地理空間データの**検索**
 - 例: コンビニテーブルで、指定した位置から100m以内にあるコンビニを探す
 - 地理空間データの**加工**
 - 例: 市区町村ポリゴンを結合して都道府県ポリゴンを作る

5

PostGIS/GISは言うほど恐くない

6

PostGISの内容

- 地理空間データ用の**ユーザ定義型**
- 地理空間データ用**インデックス**
 - 地理空間の検索速度が向上
 - GiSTなど (BRINとかも)
- 多数の関数
 - **計測関数** (ST_Area()など)
 - **テスト関数** (ST_Intersects()など)
 - **重ね合わせ関数** (ST_Union()など)
 - **ジオメトリ処理関数** (ST_Buffer()など)

7

テスト関数

- WHERE節、JOIN節等で使用

ST_Contains(A,B)	×	○	×	×
ST_Covers(A,B)	×	○	×	×
ST_Within(A,B)	×	×	○	×
ST_CoveredBy(A,B)	×	×	○	×
ST_Intersects(A,B)	○	○	○	×
ST_Overlaps(A,B)	○	×	×	×
ST_Disjoint(A,B)	×	×	×	○

他に ST_Touches(), ST_Crosses(), ST_DWithin() など

8

テスト関数の使い方

- テスト関数はWHERE節、JOIN節で使用
 - (135E,35N)から**40km圏内**の駅を探す

```
SELECT * FROM 駅 WHERE
ST_DWithin('POINT(135 35)', geom, 40000);
```

- AとBを**結合**し、A.geomがB.geomを**含む**ダブル抽出

```
SELECT A.*, B.* FROM A INNER JOIN B
ON ST_Contains(A.geom, B.geom);
```

9

重ね合わせ関数

- 重ね合わせ関数は**論理演算に近い**
 - **ST_Intersection(A,B)** - 交差 → **A & B**
 - **ST_Union(A,B)** - 結合 → **A | B**
 - **ST_SymDifference(A,B)** - 対称差 → **A XOR B**
 - A-B (差) は A & (A XOR B)

10

重ね合わせの集約関数

- 一部の重ね合わせ関数は**集約関数版**もある
 - 県ごとの人口総数 (数値の足し算)

```
SELECT 県コード, Sum(人口) FROM 市町村
GROUP BY 県コード;
```

- 県ごとのポリゴン (ポリゴンの足し算)

```
SELECT 県コード, ST_Union(ポリゴン)
FROM 市町村
GROUP BY 県コード;
```

11

図形はGISで見る

- 「**QGIS**」 (<https://qgis.org/>)等を使用すれば

出典: 政府統計の総合窓口(e-Stat)
 出典: 地理院タイル
 出典: 基盤地図情報DEM (10B)

12

それでも「怖い」ところがある

- PostGIS, GIS独特の概念
- **地理空間データ型** (詳細情報が多くない)
- **座標参照系**の話は結構面倒
 - CRS, Coordinate Reference System
 - **地図投影法**とかの話
 - 使われるCRS数が多い
 - でも「**平面直角座標系**」「**UTM**」「**Webメルカトル**」でだいたいOK
- 地理座標系の長さや面積等の計測

13

13

地理空間データの話

14

14

地理空間データ(ベクタデータ)

- 地理空間データ(ベクタデータ)
= **ジオメトリ (geometry)**+**属性 (attribute)**
- ジオメトリ: 位置情報を持つ図形データ
- 属性: 図形以外のデータ

属性

市町村名: △市

大字名: 大字×

地番: 1234番地

面積(m²): 2000.0

ジオメトリ

15

15

PostGISでは1行1地理空間データ

- ジオメトリは1個 (1カラム)の場合が多い (必須ではない)

市町村名	大字名	地番	面積	geom
△市	大字×	1234番地	2000.0	...

属性

市町村名: △市

大字名: 大字×

地番: 1234番地

面積(m²): 2000.0

ジオメトリ

16

16

例: 農業集落境界 (農林業センサス)

属性データ	値
agri202034	
pref_name	広島県
city_name	福山市
s_name	米屋
kcity	01
agri	040
pref_n	広島県
city_n	福山市
kcity_n	米屋
agn_n	米屋

17

17

ジオメトリのタイプ - 単純ジオメトリ

- ジオメトリは**各頂点に位置情報**を持つ図形データ
 - 頂点は**座標値** (X=○, Y=△)
- **単純ジオメトリ**: 点、線、面のうち一つ
 - **ポイント**、**ラインストリング**、**ポリゴン**

ポイント (点)

ラインストリング (線)

ポリゴン (面)

●: 頂点。位置情報 (経度・緯度等) が入る

18

18

ジオメトリのタイプ - マルチ系ジオメトリ

- **マルチ系ジオメトリ**: 同じタイプの単純ジオメトリを**複数集めたもの**
- **マルチポイント**、**マルチラインストリング**、**マルチポリゴン**

マルチポイント

マルチラインストリング

マルチポリゴン

19

19

ジオメトリコレクション

- 前2ページのジオメトリの**コレクション**
- あまり使わない
- これで**一つのジオメトリ**扱い

ジオメトリコレクション

20

20

参考: ラスタデータ

- 地理空間データの種類
- ビットマップ (TIFF多い) + 位置情報
 - ピクセルごとにデータを持たせる
 - 位置情報はアフィン変換で付与
 - $x' = Ax + b$ (x' : 地図座標値, x : 画像座標値)
 - パラメータ数: A (2行×2列)と b (2行) → 計6個

出典: 基盤地図情報DEM (10B)

21

21

座標参照系の話

22

座標参照系/空間参照系

- 「**座標参照系**」 (Coordinate Reference System) **の方が優勢**
- 「空間参照系」 (Spatial Reference System) と同じ (SRSの方が古い表現)
- 座標値の決め方ごとに「系」がある
 - <https://epsg.org/> にレポジトリ
 - 数字で指定する
- **測地系数 × 投影法数 × 投影パラメータ数** だけ座標参照系がある

23

測地系

- 地球(**回転楕円体**) **サイズ・扁平率等が違う**
 - **Tokyo** (1918) - Bessel 1841
 - **JGD2000, JGD2011** – GRS80
 - **WGS84** - WGS84 (GRS80の精度落とした版)
- JGD2000, JGD2011 と WGS84 は**結構近い**
 - PostGISでは全く同じと捉えてる
- Tokyo だけ結構ずれる
 - 「**日本のへそ**」が**二つ**ある

24

測地系の差でできた二つの「へそ」

大正のへそ
(Tokyoの 35N 135E)

約437m

日本へそ公園
(WGS84の 35N 135E)

出典：地理院タイル

25

25

地理座標系と投影座標系

- 座標系は**地理座標系**と**投影座標系**に大別
- 地理座標系** - 緯度経度のまま使用
- 投影座標系** - 緯度経度を地図平面上の位置に変換したもの
 - 投影すると**必ず歪む**
 - 許せる投影法 (歪ませ方) を採用
- 「**ガウス=クリューゲル図法**」と「**メルカトル図法**」ぐらいしか出てこない

26

26

投影法

- ガウス=クリューゲル図法**
 - 戦後の国内陸域で広く利用
 - 平面直角座標系**、**UTM**で利用する投影法
 - 平面直角: **1:2,500**地図用、日本を19分割
 - UTM: **1:25,000**地図用、地球を6度毎分割 → 60ゾーン
- メルカトル図法**
 - 海図に利用
 - Googleマップなどで使われる
 - Webメルカトル** - 地球を**球面**として計算

27

27

パラメータ

- 同じガウス=クリューゲル図法でも**パラメータが違うと別の座標参照系**
 - 拡大率: 平面直角: 0.9999, UTM: 0.9994
- 同じ平面直角/UTMでも**原点位置が違うと別の座標参照系**

出典: 国土地理院

出典: 地理院タイル

28

28

パラメータ

日本の平面直角座標系
この図は、国土地理院の「Webメルカトル」図法をベースに、平面直角座標系を各都道府県ごとに設定したものです。一部は正確な可視化ができません。正確な座標系が必要な場合は、国土地理院の「Webメルカトル」図法を参照してください。

出典: 国土地理院

29

29

国内陸域で使われる空間参照系の種類

- 地理座標系:**
(Tokyo, JGD 2000, JGD2011, WGS84) = 4種
- 平面直角座標系:**
(Tokyo, JGD 2000, JGD2011) × (19区画) = 57種
- UTM座標系**(日本近く):
 - (Tokyo, JGD 2000, JGD2011) × (5ゾーン (51...55)) = 15種
 - UTMは地球を6度ずつのゾーンに分割
- Webメルカトル** (半径6378137m球面) = 1種

→ 合計 77種

30

30

座標参照系コード



- 世界中の**座標参照系**に**コード**が与えられる
- “(作成機関):(識別数字)”
- **EPSGコード**が**一般的**
 - レポジトリ: <https://epsg.org/>
 - EPSG = European Petroleum Survey Group (欧州石油調査グループ)
 - 現: International Association of Oil & Gas Producers (国際石油ガス生産者協会)

31

31

国内陸域でよく使われるSRID一覧



SRID	測地系	座標系
4612		地理座標系
2443~2461	JGD2000	平面直角座標系1系~19系
3097~3101		UTMゾーン51~55
6668		地理座標系
6669~6687	JGD2011	平面直角座標系1系から19系まで
6688~6692		UTMゾーン51から55まで
4301		地理座標系
30161~30179	Tokyo	平面直角座標系1系から19系まで
3092~3096		UTMゾーン51から55まで
4326		地理座標系
3857	WGS84	Webメルカトル

32

32

座標参照系を間違えると



- 座標値同じでも座標参照系によって全く違う位置を示す
- 例: (135, 35)
 - 地理座標系 (35N 135E): **兵庫県**西脇市
 - WGS84とTokyoで**400m以上ズレる**
 - 投影座標系: 原点から東135m, 北35mの地点
 - Webメルカトル: **ギニア湾**
 - 平面直角9系: **千葉県**野田市のゴルフ場
- 地理座標系とWebメルカトルを取違えると**ギニア湾の海上**(ヌル島)に**建物**が立つ

33

33

なのでエラーにする



```

SELECT ST_Union(
  SRID=6671のポリゴンの
  EWKT表現 (説明しません)
  'SRID=6671;POLYGON((...))':GEOMETRY,
  'SRID=3857;POLYGON((...))':GEOMETRY
);
SRID=3857のポリゴン
  
```



```

ERROR: lwgeom_union_prec: Operation on
mixed SRID geometries (6671 != 3857)
  
```

34

34

経度緯度データにはGEOGRAPHY型

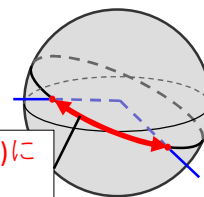
35

35

経度緯度ベースで計測関数が使えない



- **経度緯度ベース**での**長さ**の計算は**複雑**
 - 経度緯度ベースの長さは
 $\sqrt{\text{緯度差}^2 + \text{経度差}^2}$ では**計算できない**
→ ST_Length(geom) が出る長さがコレ
- **経度緯度ベース**の長さ、面積を**計測できる方法**が欲しい



大円コース(曲線)に沿って長さ計算

36

36

経度緯度の測定にはGEOGRAPHY



- ジオメトリは普通は **GEOMETRY** 型
- **経度緯度ベースの計算はGEOGRAPHY型にキャストするだけで測れる**
 - ST_Area(geom::GEOGRAPHY)
 - ST_Length(geom::GEOGRAPHY)
 - 単位はメートル
 - ただし計算量多い
- **GEOGRAPHY型は「経度緯度の計測計算をしてください」と伝えるためのもの**

37

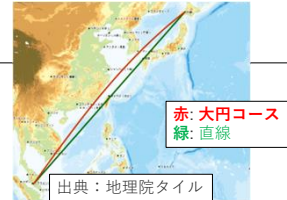
37

計測以外もGEOGRAPHY



- ST_Segmentize(geog)
 - **線分の分割。GEOGRAPHYでは分割するほど大円コースに近くなる。**

```
ST_Segmentize(
  'LINESTRING(102 3,141.4 43.1)::GEOGRAPHY,
  100000
)
```



38

38

GEOMETRY対GEOGRAPHY



- GEOGRAPHYでは複雑な計算を行ってくれる
 - **GEOGRAPHYの方が計算量が多い**
 - 曲線にそった距離計算 OR 投影→本処理→元座標系 を実行
- あらかじめ **投影座標系に変換しておく**と **GEOMETRY型だけ**でいいことになる
- **地理座標系**でも普段はGEOMETRY型で **要る時だけGEOGRAPHY型**を使う方がいい
 - GEOGRAPHY, GEOMETRYは相互キャスト可

39

39

ためしにデータをいじってみよう



40

40

例題



- ある市で **市街化区域内にある田畑**を抽出して **面積の合計**を求める
 - 農地転用は、**市街化区域**では**届出制**、それ以外では**許可制**
- データ所在
 - 都市計画区域
 - 「**国土数値情報(都市地域データ)**」
 - 田畑ポリゴン
 - 「**筆ポリゴン**」
 - 他に「**農業集落境界**」

41

41

筆ポリゴン



- 田畑のポリゴン (テーブル名: fude)



筆ポリゴンデータ(農林水産省)

42

42

都市計画区域

- テーブル名: planing

■ 市街化区域
■ 市街化調整区域

国土数値情報 (都市地域データ)

43

都市計画区域

- 白いところが気になる

白色の領域は
・ 他市領域
・ 都市計画区域外
どれ?

市区町村ポリゴン
と重ねたら分かる
はず

国土数値情報 (都市地域データ)

44

市区町村ポリゴンを作ろう!

- 市区町村ポリゴンがない!
- 本当は「国土数値情報(行政区域)」使う
- 「農業集落境界」から作ってみよう

出典: 政府統計の総合窓口(e-Stat)

45

テーブルを作る

- ジオメトリカラムを **GEOMETRY型** とする
- **タイプ** と **SRID** の修飾をしましょう
- **GiST** で **インデックス** を貼りましょう

```
CREATE TABLE mncpl (
  gid SERIAL PRIMARY KEY,
  mcode TEXT,
  mname TEXT,
  geom GEOMETRY (MULTIPOLYGON, 4326)
);
CREATE INDEX ON mncpl USING GiST (geom);
```

座標参照系=4326の
ポリゴンのみ格納

46

市コードごとにグループ化して結合

- **ST_Union()** 集約関数版
- **GROUP BY** でグループ化されたポリゴン全てを結合
- **離島** があると **マルチポリゴン** 返す
- 離島ないとポリゴン返す (要注意、後述)

```
INSERT INTO mncpl (mcode, mname, geom)
SELECT pref||city, city_name, ST_Union(geom)
FROM agri202034
GROUP BY pref||city, city_name; エラーあり
```

47

INSERT実行したらエラーが出た

```
INSERT INTO mncpl (mcode, mname, geom)
SELECT pref||city, city_name, ST_Union(geom)
FROM agri202034
GROUP BY pref||city, city_name;
```

ERROR: Geometry SRID (4612) does not match column SRID (4326)

48

SRIDを確かめる！



- “**geometry_columns**”が**必ずある**ので確認
- agri202034.geom が SRID=4612
→ できたポリゴンも 4612
- mncpl.geom が SRID=4326
→ **4612のポリゴン**を入れられない

```
test=# SELECT * FROM geometry_columns;
 f_table_catalog | f_table_schema | f_table_name | f_geometry_column | coord_dimension | srid | type
-----
 test           | public         | agri202034   | geom              | 2               | 4612 | MULTIPOLYGON
 test           | public         | fude        | geom              | 2               | 4326 | POLYGON
 test           | public         | tbl         | geom              | 2               | 4612 | POINT
 test           | public         | vl          | geom              | 2               | 4612 | MULTIPOLYGON
 test           | public         | mncpl       | geom              | 2               | 4326 | MULTIPOLYGON
 test           | public         | planing     | geom              | 2               | 6668 | POLYGON
 test           | public         | ufieid      | geom              | 2               | 4326 | POLYGON
(7 rows)
```

49

49

エラーへの対応



- SRIDの**変更方法は二つ**
- ST_Transform() - **座標変換**を行う
 - 適用例: 6668 から 6672
 - 6668: JGD2011 地理座標系
 - 6672: JGD2011 平面直角座標系III系
- ST_SetSRID() - **SRIDを付替え**を行う
 - 適用例: 6668 から 4326
 - 6668: JGD2011 地理座標系
 - 4326: WGS84 地理座標系
- 今回は **SRIDの付け替え**でいい、と判断

50

50

できた！（ただし 3.4.0以降）



- **3.4.0以降**だとエラーが出ない

```
INSERT INTO mncpl (mcode, mname, geom)
SELECT pref||city, city_name,
ST_SetSRID(ST_Union(geom), 4326)
FROM agri202034
GROUP BY pref||city, city_name;
INSERT O 30
```

51

51

3.4.0より前のバージョンは注意



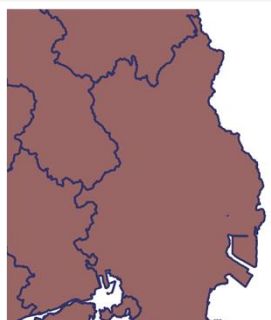
- 離島の有無でマルチポリゴン、ポリゴンどちらも返しうる
- 3.4.0より前は**マルチポリゴンのカラムにポリゴン**を入れようとすると**エラー発生**
→ **ST_Multi()**で**全部マルチポリゴンに強制**

```
INSERT INTO mncpl (mcode, mname, geom)
SELECT pref||city, city_name,
ST_Multi(ST_SetSRID(ST_Union(geom), 4326))
FROM agri202034
GROUP BY pref||city, city_name;
```

52

52

市区町村ポリゴンが完成



政府統計の総合窓口(e-Stat)を加工して作成

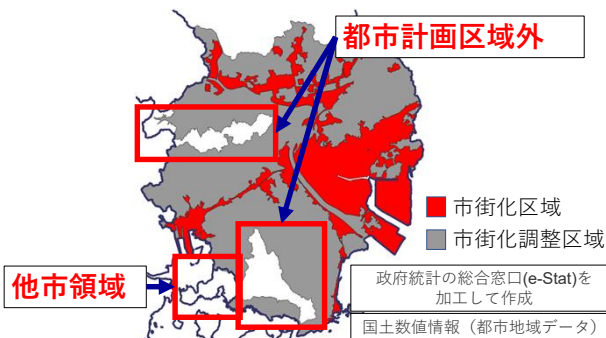
53

53

重ね合わせる



- 都市計画区域外も他市領域もあった



54

54

市街化区域内の田畑のテーブルを作る

- あまりVIEWは使わない (重い)

```
CREATE TABLE ufield (
  gid SERIAL PRIMARY KEY,
  geom GEOMETRY(POLYGON, 4326)
);
CREATE INDEX ON ufield USING GiST (geom);
```

55

55

挿入してみる

- 市街化区域に該当する planing.geom とインタセクトする fude.geom を抽出 → 挿入
- planing.geomのSRIDは6668 → 4326に揃える

```
INSERT INTO ufield (geom)
SELECT fude.geom FROM fude INNER JOIN
planing ON ST_Intersects(
  ST_SetSRID(planing.geom,4326),
  fude.geom
) AND planing.layer_no=1;
```

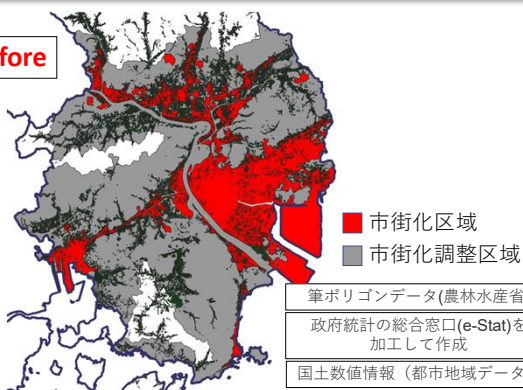
```
INSERT O 17628
```

56

56

市街化区域の田畑のみ抽出

Before

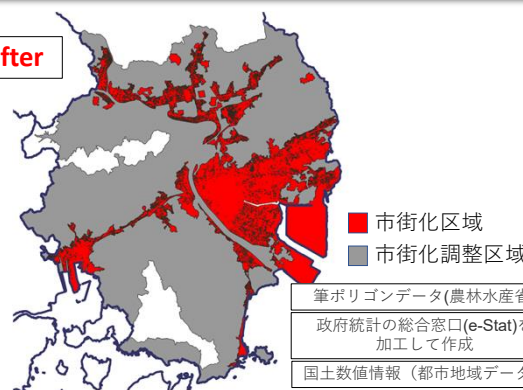


57

57

市街化区域の田畑のみ抽出

After



58

58

面積はGEOGRAPHYにすると測れる

- 約810haと出ました

```
SELECT Sum(ST_Area(geom::GEOGRAPHY)) FROM
ufield;
```

sum

8095010.63859276
(1 row)

59

59

じゃあ筆当たり面積は?

- 約4.59a/筆 と出ました

```
SELECT
Sum(ST_Area(geom::GEOGRAPHY))/COUNT(*)
FROM ufield;
```

?column?

459.21321979763786
(1 row)

60

60

アップにしてみると

- はみ出してる...

筆ポリゴンデータ(農林水産省)
国土数値情報(都市地域データ)

61

アップにしてみると問題発見

- 一部の田畑ポリゴンが市街化地域からはみ出してる！

筆ポリゴンデータ(農林水産省)
国土数値情報(都市地域データ)

62

なぜそうなるか？

- そう指示しているので仕方ない

```
INSERT INTO ufild (geom)
SELECT ... JOIN
ST_Intersects( ... , ... ) AND planing.layer_no=1;
```

「ほんの少しでもかかると」

筆ポリゴンデータ(農林水産省)
国土数値情報(都市地域データ)

63

JOINの条件を変えるといいはず

- `ST_Covers(g1, g2)` にすると、はみ出した田畑はヒットしない
→ 総面積 約733ha, 筆あたり 約4.49a/筆

```
ST_Covers( planning.geom, fude.geom ) AND ...
```

ST_Intersectsの場合

64

おわりに

65

いかがだったでしょうか？

- PostGISを入れるとGEOMETRY / GEOGRAPHY型が使える
- インデックスも使える
- 空間参照系はなんか恐いが表があればなんとかなる
- 地理座標系の長さ/面積にはGEOGRAPHY型
- ためしにデータをいじってみた

66

各種情報

- 公式サイト <https://postgis.net/>
 - マニュアル、「PostGIS入門」(後述)
- 「交通情報学特論」伊藤昌毅准教授@東大
 - <https://www.docswell.com/tag/交通情報学特論>
- Let's Postgres 記事
 - <https://lets.postgresql.jp/documents/tutorial/PostGIS>
- Zenn 記事
 - <https://zenn.dev/boiledorange73?tab=books>

67

マニュアル日本語訳の所在 1/2

- <https://postgis.net/documentation/manual/>

68

マニュアル日本語訳の所在 1/2

- 英語版のURLの末尾に「ja/」追加

<https://postgis.net/docs/manual-3.4/> → <https://postgis.net/docs/manual-3.4/ja/>

69

PostGIS入門

- <https://postgis.net/workshops/ja/postgis-intro/>
- ニューヨークのデータを扱う (翻訳70%)

70

書籍

- PostGISの和書は非常に少ない
- 「地域研究のための空間データ分析入門: QGISとPostGISを用いて」
- 外国語版はいくつか
 - 英 「PostGIS in Action」
 - 英 「Mastering PostGIS and OpenStreetMap」
 - 英 「PostGIS Cookbook」
 - 西 「PostGIS: Análisis Espacial Avanzado」
 - 仏 「PostGIS - Tous les ingrédients pour concocter un SIG sur de bonnes bases」

71

ありがとうございました

72