

QGIS Training Manual

QGIS Project

2026 年 03 月 14 日

目 次

第 1 章　コースの紹介 **3**
1.1 序文 . . . 3
1.1.1 なぜ QGIS？ . . . 3
1.1.2 背景 . . . 4
1.1.3 ライセンス . . . 4
1.1.4 協賛の節 . . . 5
1.1.5 作者 . . . 5
1.1.6 個人の貢献者 . . . 6
1.1.7 スポンサー . . . 6
1.1.8 ソースファイルと問題報告 . . . 6
1.1.9 最新バージョン . . . 6
1.2 演習について . . . 6
1.2.1 このチュートリアルの使い方 . . . 7
1.2.2 段階的なコース目標 . . . 7
1.2.3 データ . . . 8

第 2 章　モジュール: 基本的な地図を作成して探索する **9**
2.1 レッスン: インタフェースの概要 . . . 9
2.1.1 ★☆☆ （初級レベル） 自分でやってみよう: 基本 . . . 9
2.1.2 ★☆☆ （初級レベル） 自分でやってみよう: 1 . . . 12
2.1.3 ★☆☆ （初級レベル） 自分でやってみよう: 2 . . . 12
2.1.4 次は? . . . 13
2.2 レッスン: 最初のレイヤを追加する . . . 13
2.2.1 ★☆☆ （初級レベル） 理解しよう: 地図を準備する . . . 13
2.2.2 ★☆☆ （初級レベル） 自分でやってみよう: . . . 16
2.2.3 ★☆☆ （初級レベル） 理解しよう: GeoPackage データベースからベクタデータを読み込む . . . 17
2.2.4 ★☆☆ （初級レベル） 理解しよう: ブラウザで SpatiaLite データベースからベクタデータを読み込む . . . 18
2.2.5 ★★☆ （中級レベル） 自分でやってみよう: さらにベクタデータを読み込む . . . 20
2.2.6 理解する: レイヤを並び替える . . . 20
2.2.7 結論 . . . 21
2.2.8 次は? . . . 22
2.3 レッスン: マップキャンバスを見てまわる . . . 22
2.3.1 ★☆☆ （初級レベル） 理解しよう: 基本の画面操作ツール . . . 22
2.3.2 結論 . . . 26
2.4 レッスン: シンボロジ . . . 26
2.4.1 ★☆☆ （初級レベル） 理解しよう: 色を変更する . . . 27
2.4.2 ★☆☆ （初級レベル） 自分でやってみよう: . . . 28
2.4.3 ★☆☆ （初級レベル） 理解しよう: シンボル構造を変更する . . . 29

2.4.4 ★☆☆（初級レベル）自分でやってみよう: . . . 30
2.4.5 ★★☆（中級レベル）理解する: 縮尺に基づく表示 . . . 32
2.4.6 ★★☆（中級レベル）理解しよう: シンボルレイヤを追加する . . . 33
2.4.7 ★★☆（中級レベル）自分でやってみよう: . . . 37
2.4.8 ★★☆（中級レベル）理解しよう: シンボルレベルを並び替える . . . 37
2.4.9 ★★☆（中級レベル）自分でやってみよう: . . . 41
2.4.10 ★★★（上級レベル）自分でやってみよう: . . . 43
2.4.11 ★★☆（中級レベル）理解しよう: シンボルレイヤタイプ . . . 45
2.4.12 ★★☆（中級レベル）自分でやってみよう: . . . 52
2.4.13 ★★★（上級レベル）理解しよう: ジオメトリジェネレータシンボロジ . . . 54
2.4.14 ★★★（上級レベル）理解しよう: カスタム SVG 塗りつぶしを作る . . . 59
2.4.15 ★★☆（中級レベル）理解しよう: マスキング . . . 64
2.4.16 ★★☆（中級レベル）理解しよう: ポイントのシンボルを使ってマスクする . . . 66
2.4.17 結論 . . . 68
2.4.18 参考文献 . . . 69
2.4.19 次は? . . . 69

第 3 章 モジュール: ベクタデータを分類する 71
3.1 レッスン: ベクタ属性データ . . . 71
3.1.1 ★☆☆（初級レベル）理解しよう: レイヤ属性を表示する . . . 71
3.1.2 ★☆☆（初級レベル）自分でやってみよう: ベクタデータ属性を探索する . . . 75
3.1.3 結論 . . . 76
3.1.4 次は? . . . 76
3.2 レッスン: ラベル . . . 76
3.2.1 ★☆☆（初級レベル）理解しよう: ラベルを使う . . . 76
3.2.2 ★☆☆（初級レベル）理解しよう: ラベルオプションを変更する . . . 78
3.2.3 ★★☆（中級レベル）理解しよう: レイヤシンボルの代わりにラベルを使う . . . 82
3.2.4 ★★☆（中級レベル）自分でやってみよう: ラベルのカスタマイズ . . . 85
3.2.5 ★★☆（中級レベル）理解しよう: ラインにラベルを付ける . . . 89
3.2.6 ★★★（上級レベル）理解しよう: データ定義による設定 . . . 93
3.2.7 ★★★（上級レベル）自分でやってみよう: データ定義による設定を使う . . . 96
3.2.8 ★★★（上級レベル）ラベル付けのさらなる可能性 . . . 98
3.2.9 結論 . . . 99
3.2.10 次は? . . . 99
3.3 レッスン: 分類 . . . 99
3.3.1 ★☆☆（初級レベル）理解しよう: 名義データを分類する . . . 100
3.3.2 ★☆☆（初級レベル）自分でやってみよう: その他の分類 . . . 104
3.3.3 ★★☆（中級レベル）理解しよう: 比率分類 . . . 104
3.3.4 ★★☆（中級レベル）自分でやってみよう: 分類の絞り込み . . . 113
3.3.5 ★★★（上級レベル）理解しよう: 規則に基づく分類 . . . 113
3.3.6 結論 . . . 117
3.3.7 次は? . . . 117

第 4 章 モジュール: 地図をレイアウトする 119
4.1 レッスン: 印刷レイアウトを使う . . . 119
4.1.1 ★☆☆（初級レベル）理解しよう: レイアウトマネージャ . . . 119
4.1.2 ★☆☆（初級レベル）理解しよう: 基本地図の構図 . . . 121

4.1.3 ★☆☆（初級レベル）理解しよう: タイトルを追加する . . . 124
4.1.4 ★☆☆（初級レベル）理解しよう: 凡例を追加する . . . 126
4.1.5 ★★☆（中級レベル）理解しよう: 凡例をカスタマイズする . . . 127
4.1.6 ★☆☆（初級レベル）理解しよう: 地図をエクスポートする . . . 129
4.1.7 結論 . . . 131
4.2 レッスン: ダイナミック印刷レイアウトを作る . . . 131
4.2.1 ★★☆（中級レベル）理解しよう: 動的マップキャンバスを作る . . . 131
4.2.2 ★★☆（中級レベル）理解しよう: 動的ヘッダを作る . . . 132
4.2.3 ★★☆（中級レベル）理解しよう: 動的ヘッダのラベルを作る . . . 133
4.2.4 ★★☆（中級レベル）理解しよう: 動的ヘッダに画像を追加する . . . 135
4.2.5 ★★☆（中級レベル）理解しよう: 動的ヘッダのスケールバーを作る . . . 136
4.2.6 次は? . . . 137
4.3 課題 1 . . . 137
4.3.1 結論 . . . 138

第 5 章 モジュール: ベクタデータを作る 139
5.1 レッスン: 新しいベクタデータセットを作る . . . 139
5.1.1 ★☆☆（初級レベル）理解しよう: レイヤ作成ダイアログ . . . 139
5.1.2 ★☆☆（初級レベル）理解しよう: データソース . . . 143
5.1.3 ★☆☆（初級レベル）自分でやってみよう: ポリゴンをデジタイズする . . . 155
5.1.4 ★★☆（中級レベル）理解しよう: 頂点編集テーブルを使う . . . 157
5.1.5 ★☆☆（初級レベル）自分でやってみよう: 線をデジタイズする . . . 160
5.1.6 結論 . . . 163
5.1.7 次は? . . . 164
5.2 レッスン: 地物のトポロジ . . . 164
5.2.1 ★★☆（中級レベル）理解しよう: スナップする . . . 164
5.2.2 ★★☆（中級レベル）理解しよう: トポロジ的地物を修正 . . . 166
5.2.3 ★★☆（中級レベル）理解しよう: ツール: 地物の簡素化 . . . 169
5.2.4 ★★☆（中級レベル）自分でやってみよう: ツール: リングの追加 . . . 171
5.2.5 ★★☆（中級レベル）自分でやってみよう: ツール: 部分の追加 . . . 172
5.2.6 ★★☆（中級レベル）理解しよう: ツール: 地物の変形 . . . 174
5.2.7 ★★☆（中級レベル）自分でやってみよう: ツール: 地物の分割 . . . 176
5.2.8 ★★★（上級レベル）自分でやってみよう: ツール: 地物のマージ . . . 178
5.2.9 結論 . . . 178
5.2.10 次は? . . . 179
5.3 レッスン: フォーム . . . 179
5.3.1 ★☆☆（初級レベル）理解しよう: QGIS のフォームデザイン機能を使う . . . 179
5.3.2 ★☆☆（初級レベル）自分でやってみよう: フォームを使って値を編集する . . . 180
5.3.3 ★★☆（中級レベル）理解しよう: フォームのフィールドタイプを設定する . . . 181
5.3.4 ★★★（上級レベル）自分でやってみよう: . . . 183
5.3.5 ★★★（上級レベル）自分でやってみよう: テストデータを作る . . . 185
5.3.6 ★★★（上級レベル）理解しよう: 新しいフォームを作る . . . 186
5.3.7 ★★★（上級レベル）理解しよう: レイヤをフォームに関連付ける . . . 188
5.3.8 結論 . . . 189
5.3.9 参考文献 . . . 189
5.3.10 次は? . . . 189
5.4 レッスン: アクション . . . 189

5.4.1 ★☆☆（初級レベル）理解しよう: 画像のためのフィールドを追加 190
5.4.2 ★☆☆（初級レベル）理解しよう: アクションを作る 192
5.4.3 ★★☆（中級レベル）理解しよう: インターネットを検索する 196
5.4.4 ★★★（上級レベル）理解しよう: QGIS で直接 Web ページを開く 199
5.4.5 結論 201
5.4.6 次は? 201

第 6 章 モジュール: ベクタ解析 **203**
6.1 レッスン: データを再投影および変換する 203
6.1.1 ★☆☆（初級レベル）理解しよう: 投影法 203
6.1.2 ★☆☆（初級レベル）理解しよう:「オンザフライ」再投影 204
6.1.3 ★★☆（中級レベル）理解しよう: 他の CRS へデータセットを保存する 205
6.1.4 ★★★（上級レベル）理解しよう: 独自の投影法を作る 208
6.1.5 結論 210
6.1.6 参考文献 210
6.1.7 次は? 210
6.2 レッスン: ベクタ解析 211
6.2.1 ★☆☆（初級レベル）GIS プロセス 211
6.2.2 ★☆☆（初級レベル）問題 211
6.2.3 ★☆☆（初級レベル）データ 211
6.2.4 ★☆☆（初級レベル）理解しよう: プロジェクトを開始してデータを取得する . . 212
6.2.5 ★☆☆（初級レベル）自分でやってみよう: レイヤ CRS の変換 214
6.2.6 ★☆☆（初級レベル）理解しよう: 学校と道路からの距離 216
6.2.7 ★☆☆（初級レベル）自分でやってみよう: 学校からの距離 223
6.2.8 ★☆☆（初級レベル）理解しよう: 重複している地域 226
6.2.9 ★☆☆（初級レベル）理解しよう: 建物を抽出する 229
6.2.10 ★★☆（中級レベル）自分でやってみよう: さらに建物をフィルタ 230
6.2.11 ★☆☆（初級レベル）理解しよう: 正しいサイズの建物の選択 233
6.2.12 ★☆☆（初級レベル）自分でやってみよう: 234
6.2.13 結論 234
6.2.14 次は? 235
6.3 レッスン: ネットワーク分析 235
6.3.1 ★☆☆（初級レベル）理解しよう: ツールとデータ 235
6.3.2 ★☆☆（初級レベル）最短経路を計算する（2 地点間） 236
6.3.3 ★★☆（中級レベル）自分でやってみよう: 最速経路 238
6.3.4 ★★☆（中級レベル）理解しよう: 高度なオプション 240
6.3.5 ★★★（上級レベル）速度制限のある最短パス 242
6.3.6 ★★☆（中級レベル）サービスエリア（レイヤから） 245
6.3.7 結論 247
6.3.8 次は? 247
6.4 レッスン: 空間統計 247
6.4.1 ★☆☆（初級レベル）理解しよう: テストデータセットの作成 248
6.4.2 ★☆☆（初級レベル）理解しよう: 基本統計 252
6.4.3 ★☆☆（初級レベル）理解しよう: 点間の距離についての統計を計算 255
6.4.4 ★☆☆（初級レベル）理解しよう: 最近傍解析（レイヤ内） 256
6.4.5 ★☆☆（初級レベル）理解しよう: 加重平均座標 258
6.4.6 ★☆☆（初級レベル）理解しよう: 画像ヒストグラム 259

6.4.7 ★☆☆ （初級レベル） 理解しよう: 空間補間 . . . 262
6.4.8 ★★☆ （中級レベル） 自分でやってみよう: 補間方法の違い . . . 262
6.4.9 結論 . . . 263
6.4.10 次は? . . . 263

第 7 章 モジュール: ラスタ **265**
7.1 レッスン: ラスタデータの操作 . . . 265
7.1.1 ★☆☆ （初級レベル） 理解しよう: ラスタデータを読み込む . . . 265
7.1.2 ★☆☆ （初級レベル） 理解しよう: 仮想ラスタの作成 . . . 266
7.1.3 ★★★ （上級レベル） ラスタデータの変換 . . . 268
7.1.4 結論 . . . 271
7.1.5 次は? . . . 271
7.2 レッスン: ラスタのシンボロジを変更する . . . 271
7.2.1 ★☆☆ （初級レベル） 自分でやってみよう: . . . 271
7.2.2 ★☆☆ （初級レベル） 理解しよう: ラスタレイヤのシンボロジを変更する . . . 272
7.2.3 ★☆☆ （初級レベル） 理解しよう: 単バンドグレー . . . 273
7.2.4 ★☆☆ （初級レベル） 理解しよう: 単バンド疑似カラー . . . 274
7.2.5 理解しよう: 透明度を変える . . . 276
7.2.6 結論 . . . 279
7.2.7 参照 . . . 279
7.2.8 次は? . . . 279
7.3 レッスン: 地形解析 . . . 279
7.3.1 ★☆☆ （初級レベル） 理解しよう: 陰影図を計算する . . . 279
7.3.2 ★☆☆ （初級レベル） 理解しよう: 陰影図をオーバーレイとして使う . . . 281
7.3.3 理解しよう: 最適な地域を見つける . . . 282
7.3.4 ★★☆ （中級レベル） 理解しよう: 傾斜を計算する . . . 283
7.3.5 ★★☆ （中級レベル） 自分でやってみよう: 傾斜方位を計算する . . . 284
7.3.6 ★★☆ （中級レベル） 理解しよう: 北向きの傾斜方位を見つける . . . 285
7.3.7 ★★☆ （中級レベル） 自分でやってみよう: その他の条件 . . . 287
7.3.8 ★★☆ （中級レベル） 理解しよう: ラスタ解析結果を組み合わせる . . . 290
7.3.9 ★★☆ （中級レベル） 理解しよう: ラスタを簡素化する . . . 292
7.3.10 ★★☆ （中級レベル） 理解しよう: ラスタを再分類する . . . 295
7.3.11 ★☆☆ （初級レベル） 理解しよう: ラスタを問い合わせる . . . 299
7.3.12 結論 . . . 303
7.3.13 次は? . . . 303

第 8 章 モジュール: 分析の完了 **305**
8.1 レッスン: ラスタからベクタへの変換 . . . 305
8.1.1 ★★☆ （中級レベル） 理解しよう: **ラスタのベクタ化** ツール . . . 305
8.1.2 ★★☆ （中級レベル） 自分でやってみよう: . . . 306
8.1.3 ★★☆ （中級レベル） 理解しよう: **ベクタのラスタ化** ツール . . . 308
8.1.4 結論 . . . 309
8.1.5 次は? . . . 309
8.2 レッスン: 分析を組み合わせる . . . 309
8.2.1 ★★☆ （中級レベル） 自分でやってみよう: . . . 309
8.2.2 ★★☆ （中級レベル） やってみよう: 結果を検査する . . . 310
8.2.3 ★★☆ （中級レベル） やってみよう: 解析の改良 . . . 311

8.2.4 結論 . . . 316
8.2.5 次は? . . . 316
8.3 課題 . . . 316
8.4 レッスン: 補足実習 . . . 316
8.4.1 問題文 . . . 317
8.4.2 解決策の概要 . . . 317
8.4.3 理解しよう: 地図を設定する . . . 317
8.4.4 地図へデータを読み込む . . . 318
8.4.5 レイヤ順序を変更する . . . 320
8.4.6 正しい地区の検索 . . . 320
8.4.7 ラスタのクリップ . . . 321
8.4.8 ベクタレイヤのシンボロジを変更する . . . 323
8.4.9 ラスタレイヤのシンボロジを変更する . . . 323
8.4.10 地図をクリーンアップします . . . 324
8.4.11 陰影図の作成 . . . 324
8.4.12 傾斜 . . . 325
8.4.13 やってみよう: 傾斜方位 . . . 326
8.4.14 ラスターを再分類する . . . 326
8.4.15 ラスタを組み合わせる . . . 327
8.4.16 農村地域を検索する . . . 327
8.4.17 負のバッファを作成する . . . 328
8.4.18 ラスタをベクタ化する . . . 329
8.4.19 ジオメトリを修復する . . . 331
8.4.20 ベクタの交点を求める . . . 331
8.4.21 各ポリゴンの面積を計算する . . . 331
8.4.22 与えられたサイズの面積を選択する . . . 333
8.4.23 ケープタウン大学のデジタイズ . . . 334
8.4.24 ケープタウン大学から最寄りの場所を検索します . . . 335

第 9 章 モジュール: プラグイン **337**
9.1 レッスン: プラグインのインストールと管理 . . . 337
9.1.1 ★☆☆ （初級レベル） 理解しよう: プラグインを管理する . . . 337
9.1.2 ★☆☆ （初級レベル） 理解しよう: 新しいプラグインをインストールする . . . 338
9.1.3 ★☆☆ （初級レベル） 理解しよう: 追加のプラグインリポジトリを設定する . . . 339
9.1.4 結論 . . . 341
9.1.5 次は? . . . 341
9.2 レッスン: 便利な QGIS プラグイン . . . 342
9.2.1 ★☆☆ （初級レベル） 理解しよう: QuickMapServices プラグイン . . . 342
9.2.2 ★☆☆ （初級レベル） 理解しよう: QuickOSM プラグイン . . . 344
9.2.3 ★☆☆ （初級レベル） 理解しよう: DataPlotly プラグイン . . . 347
9.2.4 結論 . . . 351
9.2.5 次は? . . . 351

第 10 章 モジュール: オンラインリソース **353**
10.1 レッスン: Web Mapping Service . . . 353
10.1.1 ★☆☆ （初級レベル） 理解しよう: WMS レイヤを読み込む . . . 353
10.1.2 ★☆☆ （初級レベル） 自分でやってみよう: . . . 364

10.1.3 ★★★（上級レベル）自分でやってみよう: . . . 365
10.1.4 ★★★（上級レベル）自分でやってみよう: . . . 366
10.1.5 結論 . . . 367
10.1.6 参考文献 . . . 367
10.1.7 次は? . . . 367
10.2 レッスン: Web Feature Service . . . 368
10.2.1 ★☆☆（初級レベル）理解しよう: WFS レイヤを読み込む . . . 368
10.2.2 ★★☆（中級レベル）理解しよう: WFS レイヤをクエリする . . . 372
10.2.3 結論 . . . 376
10.2.4 次は? . . . 376

第 11 章 モジュール: QGIS Server **377**
11.1 レッスン: QGIS Server をインストールする . . . 377
11.1.1 ★★☆（中級レベル）理解しよう: パッケージからインストールする . . . 377
11.1.2 ★★☆（中級レベル）理解しよう: QGIS Server 実行可能ファイル . . . 378
11.1.3 ★★★（上級レベル）HTTP Server の構築 . . . 378
11.1.4 ★★☆（中級レベル）理解しよう: 別の仮想ホストを作成 . . . 378
11.1.5 結論 . . . 379
11.1.6 次は? . . . 379
11.2 レッスン: WMS サーバーを運用する . . . 380
11.2.1 ログ出力 . . . 382
11.2.2 GetMap リクエスト . . . 383
11.2.3 ★★☆（中級レベル）自分でやってみよう: 画像とレイヤのパラメータを変更する 384
11.2.4 ★★☆（中級レベル）理解しよう: フィルタ、不透明度、スタイルのパラメータを使う . . . 385
11.2.5 ★★☆（中級レベル）理解しよう: レッドラインを使う . . . 386
11.2.6 GetPrint リクエスト . . . 388
11.2.7 結論 . . . 390
11.2.8 次は? . . . 390

第 12 章 モジュール: GRASS **391**
12.1 レッスン: GRASS のセットアップ . . . 391
12.1.1 ★☆☆（初級レベル）理解しよう: 新しい GRASS セッションを始める . . . 391
12.1.2 ★☆☆（初級レベル）理解しよう: 新しい GRASS プロジェクトを始める . . . 393
12.1.3 ★☆☆（初級レベル）理解しよう: GRASS にベクタデータを読み込む . . . 397
12.1.4 ★☆☆（初級レベル）理解しよう: GRASS にラスタデータを読み込む . . . 401
12.1.5 ★☆☆（初級レベル）自分でやってみよう: マップセットにレイヤを追加する . . . 403
12.1.6 ★☆☆（初級レベル）既存の GRASS マップセットを開く . . . 403
12.1.7 結論 . . . 406
12.1.8 次は? . . . 406
12.2 レッスン: GRASS ツール . . . 406
12.2.1 ★☆☆（初級レベル）理解しよう: 傾斜方位地図を作る . . . 406
12.2.2 ★☆☆（初級レベル）理解しよう: ラスタレイヤの基本的な統計情報を取得する . . . 408
12.2.3 ★★☆（中級レベル）理解しよう: Reclass ツール . . . 410
12.2.4 ★★☆（中級レベル）自分でやってみよう: 自分の規則で再分類する . . . 413
12.2.5 ★★☆（中級レベル）理解しよう: Mapcalc ツール . . . 414
12.2.6 結論 . . . 417

第 13 章 モジュール: 学習評価 **419**
13.1 基図を作る . . . 419
13.1.1 ポイントレイヤを追加する . . . 419
13.1.2 ラインレイヤを追加する . . . 420
13.1.3 ポリゴンレイヤを追加する . . . 421
13.1.4 ラスタ背景を作成する . . . 421
13.1.5 基図を完成させる . . . 421
13.2 データを分析する . . . 422
13.2.1 ★★☆ （中級レベル） / ★★★ （上級レベル） . . . 422
13.3 最終的な地図 . . . 422

第 14 章 モジュール: 林業への応用 **423**
14.1 レッスン: 林業モジュールの紹介 . . . 423
14.1.1 林業のサンプルデータ . . . 423
14.2 レッスン: 地図をジオリファレンスする . . . 424
14.2.1 ★☆☆ （初級レベル） 地図をスキャンする . . . 424
14.2.2 ★☆☆ （初級レベル） 理解しよう: スキャンした地図をジオリファレンスする . . . 425
14.2.3 結論 . . . 431
14.2.4 次は? . . . 431
14.3 レッスン: 林分をデジタイズする . . . 431
14.3.1 ★☆☆ （初級レベル） 理解しよう: 林分境界を抽出する . . . 431
14.3.2 ★☆☆ （初級レベル） 自分でやってみよう: 緑色画素の画像をジオリファレンス . . . 434
14.3.3 ★☆☆ （初級レベル） 理解しよう: デジタイズを助けるポイントを作る . . . 435
14.3.4 ★☆☆ （初級レベル） 理解しよう: 林分をデジタイズする . . . 437
14.3.5 ★☆☆ （初級レベル） 自分でやってみよう: 林分のデジタイズを完了する . . . 442
14.3.6 ★☆☆ （初級レベル） 理解しよう: 林分データを結合する . . . 443
14.3.7 ★☆☆ （初級レベル） 自分でやってみよう: 面積と周囲の長さを追加する . . . 446
14.3.8 結論 . . . 447
14.3.9 次は? . . . 447
14.4 レッスン: 林分を更新する . . . 447
14.4.1 ★☆☆ （初級レベル） 古い林分を現在の航空写真と比較する . . . 447
14.4.2 ★☆☆ （初級レベル） CIR 画像を解釈する . . . 448
14.4.3 ★☆☆ （初級レベル） 自分でやってみよう: CIR 画像から林分をデジタイズする . . . 450
14.4.4 ★☆☆ （初級レベル） 理解しよう: 保全情報で林分を更新する . . . 452
14.4.5 ★☆☆ （初級レベル） 自分でやってみよう: 流域への距離で林分を更新する . . . 457
14.4.6 結論 . . . 458
14.4.7 次は? . . . 458
14.5 レッスン: 体系的なサンプリングの設計 . . . 458
14.5.1 森林の目録を作成する . . . 458
14.5.2 ★☆☆ （初級レベル） 理解しよう: 体系的サンプリングプロット設計を実装する . . . 459
14.5.3 ★☆☆ （初級レベル） 理解しよう: GPX 形式としてサンプルプロットを書き出す . . . 463
14.5.4 結論 . . . 464
14.5.5 次は? . . . 464
14.6 レッスン: 地図帳ツールで詳細な地図を作る . . . 464
14.6.1 ★☆☆ （初級レベル） 理解しよう: 印刷レイアウトを準備する . . . 465
14.6.2 ★☆☆ （初級レベル） 理解しよう: 背景地図を追加する . . . 466
14.6.3 ★☆☆ （初級レベル） 自分でやってみよう: レイヤのシンボロジを変更する . . . 467

14.6.4 ★☆☆ （初級レベル） 自分でやってみよう: 基本地図テンプレートを作る 469
14.6.5 ★☆☆ （初級レベル） 理解しよう: 印刷レイアウトにさらに要素を追加する . . . 470
14.6.6 ★☆☆ （初級レベル） 理解しよう: 地図帳カバレッジを作る 472
14.6.7 ★☆☆ （初級レベル） 理解しよう: 地図帳ツールを設定する 474
14.6.8 ★☆☆ （初級レベル） 理解しよう: カバレッジレイヤを編集する 476
14.6.9 ★☆☆ （初級レベル） 理解しよう: 地図を印刷する 479
14.6.10 結論 480
14.6.11 次は? 480
14.7 レッスン: 森林パラメータを計算する 480
14.7.1 ★☆☆ （初級レベル） 理解しよう: 調査結果を追加する 481
14.7.2 ★☆☆ （初級レベル） 理解しよう: 森林全体のパラメータの推定 481
14.7.3 ★☆☆ （初級レベル） 理解しよう: 林分のパラメータを推定する 482
14.7.4 結論 487
14.7.5 次は? 487
14.8 レッスン: レーザー測量データからの DEM 487
14.8.1 ★☆☆ （初級レベル） 理解しよう: Lastools をインストールする 487
14.8.2 ★☆☆ （初級レベル） 理解しよう: LAStools で DEM を計算する 489
14.8.3 ★☆☆ （初級レベル） 理解しよう: 地形陰影起伏を作る 495
14.8.4 結論 496
14.8.5 次は? 496
14.9 レッスン: 地図プレゼンテーション 497
14.9.1 ★☆☆ （初級レベル） 理解しよう: 地図データを準備する 497
14.9.2 ★☆☆ （初級レベル） 自分でやってみよう: さまざまなブレンドモードを試す . . 499
14.9.3 ★☆☆ （初級レベル） 自分でやってみよう: レイアウトテンプレートを使って地図の結果を作る 500
14.9.4 結論 503

第 15 章 モジュール: PostgreSQL でのデータベース概念 **505**
15.1 レッスン: データベースの概要 505
15.1.1 データベースとは何ですか？ 505
15.1.2 表 505
15.1.3 列/フィールド 506
15.1.4 レコード 506
15.1.5 データ型 506
15.1.6 住所データベースをモデル化 507
15.1.7 データベース理論 508
15.1.8 正規化 508
15.1.9 ★☆☆ （初級レベル） 自分でやってみよう: 509
15.1.10 索引 509
15.1.11 シーケンス 510
15.1.12 エンティティ・リレーションシップ図の作成 510
15.1.13 制約、主キーと外部キー 512
15.1.14 トランザクション 513
15.1.15 結論 513
15.1.16 次は? 513
15.2 レッスン: データモデルの実装 513
15.2.1 PostgreSQL のインストール 514

15.2.2 ヘルプ . . . 514
15.2.3 データベースユーザーの作成 . . . 514
15.2.4 新しいアカウントの確認 . . . 515
15.2.5 データベースの作成 . . . 516
15.2.6 データベースのシェルセッションの開始 . . . 516
15.2.7 SQL でテーブルを作成する . . . 517
15.2.8 SQL でキーを作成する . . . 519
15.2.9 SQL でインデックスを作成する . . . 520
15.2.10 SQL でテーブルを削除する . . . 520
15.2.11 pgAdmin III について一言 . . . 521
15.2.12 結論 . . . 521
15.2.13 次は? . . . 521
15.3 レッスン: モデルにデータを追加する . . . 521
15.3.1 insert 文 . . . 522
15.3.2 制約に従ってデータの追加を順序付けする . . . 523
15.3.3 やってみよう: ★★☆ （中級レベル） . . . 523
15.3.4 データを選択 . . . 524
15.3.5 データを更新 . . . 524
15.3.6 データを削除 . . . 524
15.3.7 やってみよう: ★★★ （上級レベル） . . . 525
15.3.8 結論 . . . 525
15.3.9 次は? . . . 525
15.4 レッスン: クエリ . . . 525
15.4.1 結果を並べ替える . . . 526
15.4.2 フィルタリング . . . 526
15.4.3 結合 . . . 528
15.4.4 副選択 . . . 528
15.4.5 クエリの集約 . . . 529
15.4.6 結論 . . . 531
15.4.7 次は? . . . 531
15.5 レッスン: ビュー . . . 531
15.5.1 ビューの作成 . . . 531
15.5.2 ビューの変更 . . . 532
15.5.3 ビューの削除 . . . 532
15.5.4 結論 . . . 532
15.5.5 次は? . . . 533
15.6 レッスン: ルール . . . 533
15.6.1 ログに記録するルールを作る . . . 533
15.6.2 結論 . . . 534
15.6.3 次は? . . . 534

第 16 章 Module: Spatial Database Concepts with PostGIS / PostgreSQL 535
16.1 レッスン: PostGIS の設定 . . . 535
16.1.1 Ubuntu でのインストール . . . 536
16.1.2 Windows でのインストール . . . 536
16.1.3 その他のプラットフォームへのインストール . . . 536
16.1.4 PostGIS を使うためにデータベースを設定する . . . 537

16.1.5 インストールされた PostGIS 関数を見る . . . 537
16.1.6 空間参照系 . . . 538
16.1.7 結論 . . . 539
16.1.8 次は? . . . 539
16.2 レッスン: 単純地物モデル . . . 540
16.2.1 OGC とは . . . 540
16.2.2 SFS モデルとは . . . 540
16.2.3 ジオメトリフィールドをテーブルに追加する . . . 541
16.2.4 ジオメトリタイプに基づく制約を追加する . . . 541
16.2.5 ★★★ （上級レベル） 自分でやってみよう: . . . 541
16.2.6 geometry_columns テーブルの設定 . . . 541
16.2.7 SQL を使用してテーブルにジオメトリレコードを追加する . . . 542
16.2.8 結論 . . . 545
16.2.9 次は? . . . 545
16.3 レッスン: インポートとエクスポート . . . 546
16.3.1 shp2pgsql . . . 546
16.3.2 pgsql2shp . . . 546
16.3.3 ogr2ogr . . . 547
16.3.4 DB Manager . . . 547
16.3.5 結論 . . . 547
16.3.6 次は? . . . 547
16.4 レッスン: 空間クエリ . . . 547
16.4.1 空間演算子 . . . 548
16.4.2 空間索引 . . . 548
16.4.3 やってみよう: ★★☆ （中級レベル） . . . 549
16.4.4 PostGIS 空間関数デモ . . . 549
16.4.5 結論 . . . 557
16.4.6 次は? . . . 557
16.5 レッスン: ジオメトリの構成 . . . 557
16.5.1 ラインストリングの作成 . . . 557
16.5.2 やってみよう: ★★☆ （中級レベル） . . . 557
16.5.3 ポリゴンの作成 . . . 558
16.5.4 練習:Cities を People にリンクする . . . 559
16.5.5 スキーマに着目する . . . 560
16.5.6 やってみよう: ★★★ （上級レベル） . . . 561
16.5.7 サブオブジェクトへのアクセス . . . 562
16.5.8 データプロセシング . . . 562
16.5.9 クリッピング . . . 562
16.5.10 ジオメトリを他のジオメトリから構築する . . . 564
16.5.11 ジオメトリクリーニング . . . 565
16.5.12 テーブル間の差 . . . 565
16.5.13 表領域 . . . 566
16.5.14 結論 . . . 566

第 17 章 QGIS プロセシングガイド **567**
17.1 はじめに . . . 567
17.2 始める前の重要な警告 . . . 568

17.3 プロセシングフレームワークの準備をする . . . 569
17.4 最初のアルゴリズムを実行する・ツールボックス . . . 572
17.5 さらなるアルゴリズムとデータタイプ . . . 575
17.6 CRS・再投影 . . . 583
17.7 選択 . . . 587
17.8 外部アルゴリズムを実行する . . . 589
17.9 プロセシングログ . . . 595
17.9.1 ★☆☆（初級レベル）理解しよう . . . 596
17.9.2 ★★☆（中級レベル）理解しよう . . . 597
17.10 ラスタ計算機。No-data 値 . . . 597
17.11 ベクター計算機 . . . 602
17.12 範囲を定義する . . . 607
17.13 HTML 出力 . . . 611
17.14 最初の分析例 . . . 614
17.15 ラスタレイヤをクリップしてマージする . . . 622
17.16 水文解析 . . . 632
17.17 モデルデザイナーから始める . . . 645
17.18 より複雑なモデル . . . 659
17.19 モデラーでの数値計算 . . . 665
17.20 モデル内のモデル . . . 669
17.21 モデルを作成するためにモデラー専用ツールを使用する . . . 671
17.22 内挿 . . . 676
17.23 補間（続） . . . 686
17.24 アルゴリズムの反復実行 . . . 694
17.25 アルゴリズムの反復実行（続） . . . 699
17.26 バッチ処理インターフェイス . . . 701
17.27 バッチ プロセシング インタフェースのモデル . . . 705
17.28 実行前後のスクリプトのフック . . . 706
17.29 その他のプログラム . . . 708
17.29.1 GRASS: Geographic Resources Analysis Support System、地理的資源分析支援システム 708
17.29.2 R . . . 708
17.29.3 他 . . . 708
17.29.4 バックエンドの間での比較 . . . 709
17.30 補間と等高線作成 . . . 710
17.30.1 内挿 . . . 710
17.30.2 等高線 . . . 711
17.31 ベクターの単純化と平滑化 . . . 711
17.32 太陽光発電所を計画する . . . 712
17.33 プロセシングで R スクリプトを使用する . . . 712
17.33.1 スクリプトを追加する . . . 713
17.33.2 プロットを作成する . . . 714
17.33.3 ベクタを作成する . . . 718
17.33.4 R からのテキスト・グラフ出力 - 文法 . . . 721
17.34 地滑りを予測する . . . 721

第 18 章 モジュール: QGIS で空間データベースを使う **723**
18.1 レッスン: QGIS ブラウザでデータベースを操作する . . . 723

18.1.1 ★☆☆（初級レベル）理解しよう: ブラウザを使って QGIS にデータベーステーブルを追加する . . . 723
18.1.2 ★☆☆（初級レベル）理解しよう: レコードのフィルタセットをレイヤとして追加する . . . 725
18.1.3 結論 . . . 727
18.1.4 次は? . . . 727
18.2 レッスン: DB マネージャを使用して QGIS で空間データベースと連携する . . . 727
18.2.1 ★☆☆ (Basic level) Follow Along: Managing PostgreSQL Databases with DB Manager 727
18.2.2 ★☆☆（初級レベル）理解しよう: 新しいテーブルを作る . . . 734
18.2.3 ★☆☆（初級レベル）理解しよう: 基本的なデータベース管理 . . . 735
18.2.4 ★☆☆（初級レベル）理解しよう: DB マネージャで SQL クエリを実行する . . . 736
18.2.5 DB マネージャを使用したデータベースへのデータのインポート . . . 738
18.2.6 DB マネージャを使用したデータベースからのデータの書き出し . . . 740
18.2.7 結論 . . . 742
18.2.8 次は? . . . 742
18.3 レッスン：QGIS で SpatiaLite データベースを操作する . . . 742
18.3.1 ★☆☆（初級レベル）理解しよう: ブラウザで SpatiaLite データベースを作る . . . 743
18.3.2 結論 . . . 745

第 19 章 付録：このマニュアルに貢献する **747**
19.1 リソースのダウンロード . . . 747
19.2 マニュアルの形式 . . . 747
19.3 モジュールを追加する . . . 747
19.4 レッスンを追加する . . . 749
19.5 セクションを追加する . . . 749
19.5.1 「理解しよう」セクションを追加する . . . 750
19.5.2 「自分でやってみよう」セクションを追加する . . . 750
19.6 結論を追加 . . . 751
19.7 参考文献セクションを追加 . . . 751
19.8 [次は] セクションを追加 . . . 751
19.9 マークアップを使用する . . . 751
19.9.1 新しい概念 . . . 751
19.9.2 強調 . . . 752
19.9.3 画像 . . . 752
19.9.4 内部リンク . . . 752
19.9.5 外部リンク . . . 752
19.9.6 等幅テキストを使う . . . 753
19.9.7 ラベルする GUI 項目 . . . 753
19.9.8 メニューの選択 . . . 753
19.9.9 注を追加する . . . 753
19.9.10 後援/原作者注を追加する . . . 754
19.10 ありがとうございました！ . . . 754

第 20 章 練習データを準備する **755**
20.1 OSM を基にしたベクタファイルを作る . . . 755
20.2 SRTM DEM tiff ファイルを作る . . . 762
20.3 画像の tiff ファイルを作る . . . 762

20.4 トークンを置き換える 763

第1章　コースの紹介

1.1 序文

ようこそ当講座へ! QGIS の簡単で効率的な使い方をお伝えしていきます。GIS が初めての方には、使い始めるために必要なことをお伝えします。経験者の方には、QGIS が GIS プログラムに期待されるすべての機能、そしてそれ以上のものをどのように満たしているかをご覧いただきます。

1.1.1 なぜ QGIS？

情報がますます空間的に認識されるようになり、一般的に使用される GIS 機能の一部または全部を実現するツールには事欠かなくなっています。なぜ、他の GIS ソフトウェアパッケージではなく、QGIS を使用する必要があるのでしょうか？

ここでは、その理由の一部をご紹介します:

- **ランチのように無料です。** QGIS のインストールと使用には、全く費用がかかりません。初期費用も、月額費用もまったくありません。
- **解放されたように自由です。** QGIS に追加機能が必要な場合、次のリリースに含まれることを期待するだけでなく、それ以上のことができます。機能の開発を支援することもできますし、プログラミングに慣れていれば自分で追加することもできます。
- **常に開発が進められています。** 誰でも新しい機能を追加したり、既存の機能を改良することができるので、QGIS は決して停滞することがありません。新しいツールの開発は、必要な限り迅速に行うことができます。
- **豊富なヘルプとドキュメントがあります。** 何か困ったことがあれば、豊富なドキュメント、先輩の QGIS ユーザー、あるいは開発者にも頼ることができます。
- **クロスプラットフォームです。** QGIS は、macOS、Windows そして Linux にインストールできます。

QGIS を使いたくなる理由がわかったところで、これらの演習で使い方がわかるようになります。

1.1.2 背景

2008 年に私たちは やさしい GIS 入門 、専門用語や新しい用語に煩わされることなく GIS について学びたい人のための完全にフリーのオープンコンテンツリソースを立ち上げました。南アフリカ政府が援助したこの教材は驚異的な成功を収め、世界中の人々がこの教材を使って大学の訓練コースを運営したり、GIS を自習しているか、など、お便りをいただいています。『やさしい入門』はソフトウェアのチュートリアルではなく、GIS について学ぶ人のための一般的な教科書を目指しています（すべての例で QGIS を使用してはいますが）。QGIS アプリケーションの詳細な機能の概要を提供する QGIS マニュアルもあります。しかしそれはチュートリアルとしてというよりは、リファレンスガイドとして構成されています。私たちは Linfiniti Consulting CC. で頻繁に訓練コースを実行しており、QGIS の重要な側面をトレーナー・訓練生形式で順次学習していくという、第三の材料が必要であることに気づき、それがこの作品を作るきっかけになりました。

このトレーニングマニュアルは、 QGIS、PostgreSQL、PostGIS に関する 5 日間のコースを実施するために必要なすべての資料を提供することを目的としています。このコースは、初心者から中級者、上級者のすべてに適した内容で構成されており、テキスト全体に注釈付きの解答を含む多くの演習が用意されています。

1.1.3 ライセンス



Linfiniti Consulting CC. による無料の Quantum GIS トレーニングマニュアルは、Linfiniti による以前のバージョンに基づいており、Creative Commons Attribution 4.0 International の下でライセンスされています。このライセンスの範囲を超える許可は、下記で入手できるかもしれません。

私たちはこの QGIS トレーニングマニュアルを自由なライセンス化で公開しており、あなたはこの著作物を自由にコピーしたり、修正したり、再配布したりすることができます。ライセンスの完全なコピーが、この文書の末尾にあります。 簡単に言うと、使用ガイドラインは以下の通りです。

- この著作物を、あなた自身の著作物として表現したり、この著作物から著作者を表すテキストやクレジットを削除したりすることはできません。
- この著作物を、提供されたときよりも制限された許諾条件で再頒布することはできません。
- あなたがこの著作物に実質的な部分（少なくとも 1 つの完全なモジュール）を追加し、プロジェクトに反映することで貢献した場合、このドキュメントの著者リストの最後にあなたの名前を加えることができます（トップページに表示されます）。
- マイナーチェンジや校正によって貢献した場合は、下記の貢献者リストに、あなたを追加することができます。
- このドキュメントを全体的に翻訳した場合は、著者リストに「Translated by Joe Bloggs」という形で名前を追加することができます。
- モジュールまたはレッスンを資金提供された方は、提供したレッスンの先頭に謝辞を置くように要求することができます。例：

注釈

このレッスンは MegaCorp が提供しました。

- このライセンスの下で何をしてよいのかがはっきりわからないという方は、office@linfiniti.com を通して私たちに連絡してください。そうすれば、私たちはあなたがするつもりであることが許容されるかどうかについて助言いたします。
- この著作物を https://www.lulu.com などのセルフパブリッシングサイトで公開する場合、その利益を QGIS プロジェクトに寄付していただくようお願いしています。
- 作者の表現された許可があるときを除いて、この著作物を商業化してはいけません。明確には、商業化とは、営利目的で販売したり、商業的な二次的著作物を作成したりしてはならないことを意味します（例えば、雑誌の記事として使うためにコンテンツを販売するなど）。ただし、すべての利益を QGIS プロジェクトに寄附する場合は例外です。トレーニングコース自体が営利目的であっても、コースを実施するときに、教科書としてこの作品を使用できます（そして我々はそうすることをお勧めします）。言い換えれば、教科書としてこの作品を使ったトレーニングコースを実施して利益を得ることは歓迎されますが、この本自体の販売で利益を得ることはできません - このようなすべての利益は QGIS に還元されなければなりません。

1.1.4 協賛の節

この著作物は、QGIS でできる全てのことを網羅したものでは決してなく、他の方が新しい資料を追加して隙間を埋めることをお勧めします。 Linfiniti Consulting CC. は、商業サービスとして追加資料を作ることもできますが、作られたそのような著作物の全てはコアコンテンツの一部となり、同じライセンスのもとで公開すべきであることをご理解ください。

1.1.5 作者

- Rüdiger Thiede - Rudi は QGIS の指導的な教材と PostGIS の教材の一部を執筆しました。
- Tim Sutton (tim@kartoza.com) - Tim はこのプロジェクトの監督と指導を行い、PostgreSQL と PostGIS の部分を共同執筆しました。
- Horst Düster (horst.duester@kappasys.ch) -Horst は、PostgreSQL と PostGIS の部分を共同執筆
- Marcelle Sutton - Marcelle は、本作品の制作にあたり、校正と編集のアドバイスを行いました。

1.1.6 個人の貢献者

あなたの名前をここに！

1.1.7 スポンサー

- Cape Peninsula University of Technology

1.1.8 ソースファイルと問題報告

この文書のソースは GitHub の QGIS 文書リポジトリ で提供されています。Git のバージョン管理システムを使用する方法については、 GitHub.com を参照。

私たちの努力にもかかわらず、このトレーニングの中で間違いを見つけたり情報を見逃したりしている可能性があります。それらは https://github.com/qgis/QGIS-Documentation/issues でご報告ください。

1.1.9 最新バージョン

このドキュメントの最新版は、QGIS ドキュメントウェブサイト (https://www.qgis.org/resources/hub/#documentation) の一部であるオンライン版にアクセスすれば、いつでも入手することができます。

注釈

ドキュメントウェブサイトには、トレーニングマニュアルやその他の QGIS ドキュメントのオンライン版と PDF 版の両方へのリンクがあります。

1.2 演習について

QGIS を使いたい理由がわかったところで、使い方を紹介します。

警告

このコースには GIS データセットを追加、削除、変更する指示があります。このためにトレーニングデータセットを用意しています。ここで説明する手法を自分のデータで使用する前に、常に適切なバックアップがあることを確認してください。

1.2.1 このチュートリアルの使い方

このように見える テキストは、 QGIS のユーザーインターフェースで見ることができるものを指します。

looks like this と書かれたテキストは、メニューの案内をします。

この種のテキスト は、コマンドのような入力できるものを指します。

`This/kind/of.text` はパスまたはファイル名を指します。

`This+That` は 2 つのボタンからなるキーボードショートカットを指します。

1.2.2 段階的なコース目標

このコースは、さまざまなユーザの経験レベルに対応しています。自分がどのカテゴリに属していると考えるかによって、コースの結果は異なります。各カテゴリには次のカテゴリに不可欠な情報が含まれているため、経験レベル以下のすべての演習を行うことが重要です。

★☆☆ （初級レベル） 初級

このカテゴリのコースでは、GIS の理論的な知識や GIS ソフトウェアの操作に関する経験がほとんどないか、まったくないことを前提としています。

プログラムで実行するアクションの目的を説明するために、限られた理論的背景が提供されますが、重点は実行による学習にあります。

コースを完了すると、GIS の可能性をより良く把握し、QGIS を通じてそれらの力を利用する方法がわかります。

★★☆ （中級レベル） 中級

このカテゴリでは、GIS ソフトウェアの日常的な使用に関する実用的な知識と経験があることを前提としています。

初心者レベルの手順に基づいて構築すると、慣れ親しんだ基礎が得られるだけでなく、QGIS が慣れている他のソフトウェアとは少し異なる動作をするケースを認識することができます。また、QGIS で分析機能を使用する方法も学習します。

コースを完了すると、日常の使用に通常必要なすべての機能に QGIS を使用できるようになります。

★★★ （上級レベル） 上級

このカテゴリでは、GIS ソフトウェアの経験、空間データベースの知識と経験、リモートサーバ上のデータの使用、おそらく分析目的のスクリプトの記述、などを前提としています。

他の 2 つのレベルの指示に基づいて、QGIS インターフェイスが従うアプローチに慣れ、必要な基本機能にアクセスする方法を確実に理解できます。また、QGIS プラグインシステム、データベースアクセスなどを利用する方法も示されます。

コースを完了すれば、QGIS の日常的な操作とより高度な機能について十分に理解しているでしょう。

1.2.3 データ

このリソースに付属するサンプルデータは無料で入手でき、以下のソースから入手することができます。

- OpenStreetMap の Streets and Places データセット
- NGI の敷地境界（都市部、農村部）、水域
- CGIAR CSI の SRTM DEM

準備済のデータセットを トレーニングデータリポジトリ からダウンロードし、ファイルを展開してください。必要なデータはすべて `exercise_data` フォルダにあります。

もしあなたが講師で、より関連性の高いデータを使用したい場合は、付録の **練習データを準備する** にローカルデータを作成する手順が記載されています。

第2章　モジュール: 基本的な地図を作成して探索する

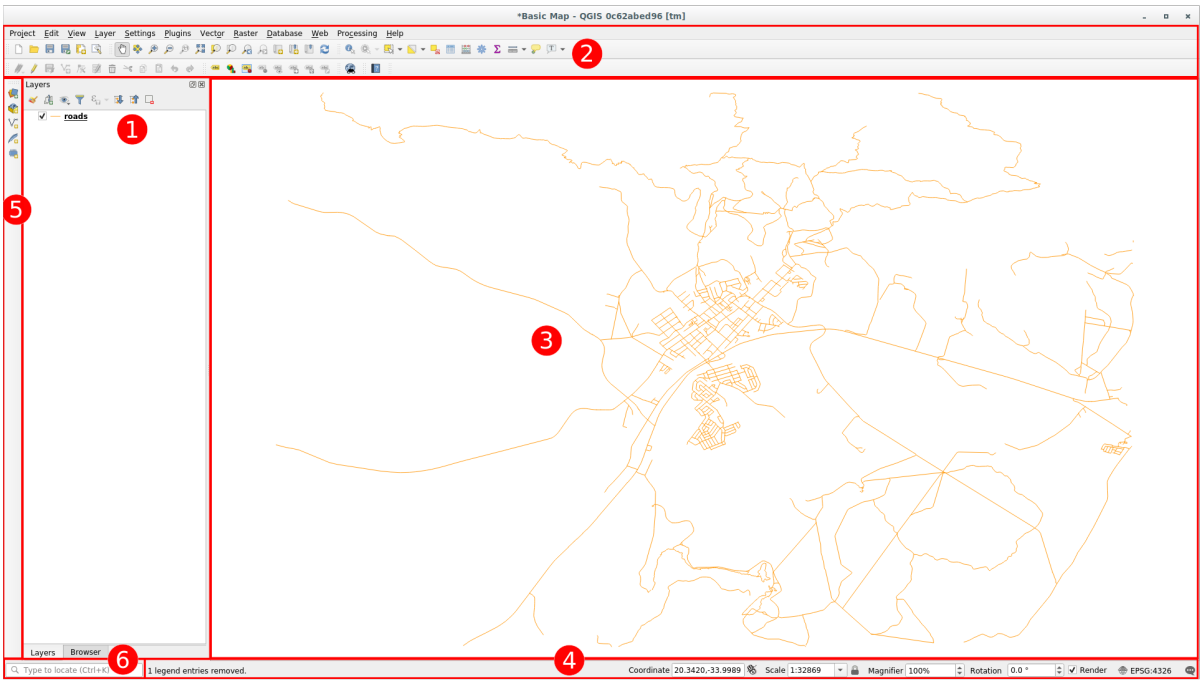
このモジュールでは、QGIS の機能性のさらなるデモンストレーションに基いて、後で使用される基本的な地図を作成します。

2.1 レッスン: インタフェースの概要

QGIS のユーザーインタフェースを探って、インタフェースの基本的な構造を形成するメニューやツールバー、マップキャンバス、レイヤリストに慣れましょう。

このレッスンの目標: QGIS のユーザーインタフェースの基礎を理解すること。

2.1.1 ★☆☆ （初級レベル） 自分でやってみよう: 基本



上の図で確認できる要素は以下の通りです。

1. レイヤリスト/ブラウザパネル

2. ツールバー
3. マップキャンバス
4. ステータスバー
5. サイドツールバー
6. ロケータバー

★☆☆ （初級レベル） レイヤリスト

レイヤリストでは、いつでも、利用可能なすべてのレイヤのリストを見ることができます。

（横にある矢印やプラス記号をクリックして）折りたたまれた項目を展開すると、そのレイヤの現在の外観についての追加情報が表示されます。

レイヤにカーソルを合わせると、レイヤ名、ジオメトリの種類、座標参照系、デバイス上の位置の完全なパスなどの基本情報が表示されます。

レイヤを右クリックすると、たくさんの追加オプションがあるメニューが表示されます。これらのオプションは、すぐにでも使用することになるので、一読しておきましょう。

注釈

ベクタレイヤは、道路、樹木のような一般的に特定の種類のオブジェクトのデータの集合体です。ベクタレイヤはポイント、ライン、ポリゴンのいずれかで構成されます。

★☆☆ （初級レベル） ブラウザパネル

The QGIS Browser is a panel in QGIS that lets you easily navigate in your database. You can have access to common vector files (e.g. ESRI Shapefile or MapInfo files), databases (e.g. PostgreSQL, Oracle, SpatiaLite, GeoPackage or MS SQL Server) and WMS/WFS connections. You can also view your GRASS data.

プロジェクトを保存している場合、ブラウザパネルでは、  **プロジェクトホーム** 項目の下にあるプロジェクトファイルと同じパスに保存されているすべてのレイヤに素早くアクセスすることができます。

Moreover, you can set one or more folder as **Favorites**: search under your path and once you have found the folder, right click on it and click on *Add as a Favorite*. You should then be able to see your folder in the  *Favorites* item.

Tip

It can happen that the folders added to Favorite item have a really long name: don't worry right-click on the path and choose *Rename Favorite...* to set another name.

★☆☆ （初級レベル） ツールバー

最もよく使うツールのセットをツールバーにして、基本的なアクセスを可能にすることができます。例えば、 **プロジェクト** ツールバーでは、保存、読み込み、印刷、新しいプロジェクトの開始ができます。メニューの **ビュー　ツールバー** から、必要に応じてツールバーを追加・削除し、よく使うツールだけを表示するようにインターフェイスを簡単にカスタマイズすることができます。

ツールバーに表示されていない場合でも、すべてのツールはメニューからアクセス可能なままです。例えば、 **プロジェクト** ツールバー（ **保存** ボタンを含む）を削除しても、 **プロジェクト** メニューをクリックし、 **保存** をクリックすれば、マップを保存できます。

★☆☆ （初級レベル） マップキャンバス

ここは、地図自体が表示され、レイヤが読み込まれる場所です。マップキャンバスでは、表示されているレイヤを操作できます: 拡大／縮小、地図移動、地物の選択、その他、次のセクションで詳しく説明する多くの操作を行うことができます。

★☆☆ （初級レベル） ステータスバー

現在の地図に関する情報を表示します。また、地図の縮尺や地図の回転を調節したり、地図上のマウスカーソルの座標を確認することができます。

★☆☆ （初級レベル） サイドツールバー

デフォルトでは、サイドツールバーには、レイヤを読み込むためのボタンと、新しいレイヤを作成するための全てのボタンが含まれています。しかし、全てのツールバーはどこでもあなたにとって快適な場所に移動させることができることを覚えておきましょう。

★☆☆ （初級レベル） ロケータバー

このバーから QGIS のほぼ全てのオブジェクトにアクセスすることができます：レイヤ、レイヤ機能、アルゴリズム、空間ブックマークなどです。QGIS ユーザーマニュアルの locator_options セクションで様々なオプションを確認することができます。

Tip

ショートカットキー `Ctrl+K` で、簡単にそのバーにアクセスできます。

2.1.2 ★☆☆ （初級レベル） 自分でやってみよう: 1

上の図を参照せずに、自分の画面で上記の 4 つの要素を確認してください。それらの名前と機能が分かるかどうか見てください。今後それらの要素を使うにつれて、より詳しく理解できるようになります。

答え

インターフェイスのレイアウトを示す画像を参照し、画面要素の名前と機能を覚えているかどうか確認してください。

2.1.3 ★☆☆ （初級レベル） 自分でやってみよう: 2

画面上でこれらの各ツールを探してみてください。それらの目的は何ですか？

1. 
2. 
3. 
4.  **レンダ**
5. 

注釈

これらのツールのいずれかが画面に表示されていない場合は、現在隠されているいくつかのツールバーを有効にしてみてください。また、画面上に十分なスペースがない場合は、ツールバーがそのツールの一部を隠すことで短縮されることがあることに注意してください。そのように折りたたまれたツールバーでは右二重矢印ボタンをクリックすると非表示のツールを表示できます。ツールの上でしばらくの間マウスを保持すると、そのツールの名前のツールチップが確認できます。

答え

1. **名前をつけて保存**
2. **レイヤの領域にズーム**
3. **地物選択を反転**
4. **レンダリング** *on/off*
5. **線の長さを測る**

2.1.4 次は?

QGIS インターフェイスの基本を理解したところで、次のレッスンでは、いくつかの一般的なデータ型を読み込む方法を紹介します。

2.2 レッスン: 最初のレイヤを追加する

アプリケーションを起動し、その例と演習で使用する基本的な地図を作成します。

このレッスンの目標: 例題の地図を使って始める

> **注釈**
>
> この練習を始める前にあなたのコンピュータに QGIS をインストールする必要があります。また、使用する *sample data* をダウンロードしておいてください。

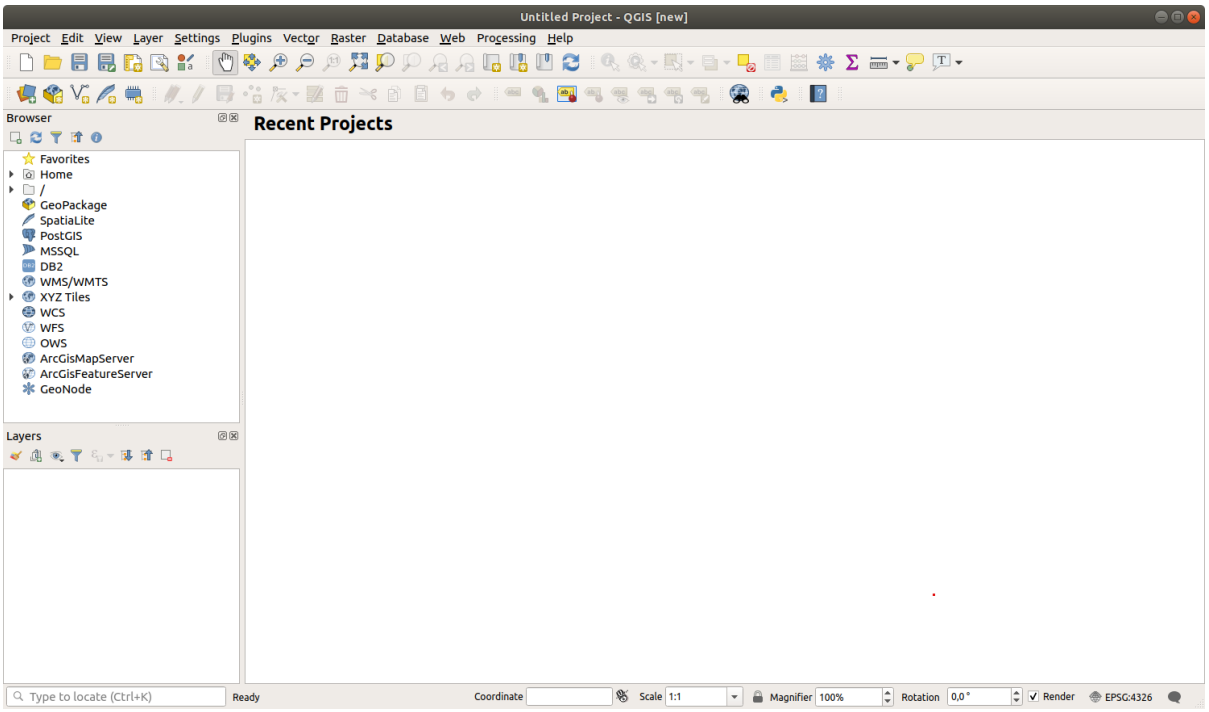
デスクトップのショートカット、メニューアイテムなどから QGIS を起動します。この設定はインストール時の設定に依存します。

> **注釈**
>
> このコースのスクリーンショットは、Linux 上で動作する QGIS 3.4 で撮影されています。あなたのセットアップによっては、表示される画面が多少異なるかもしれません。しかし、どの OS でも同じボタンが使用可能で、指示も機能します。このコースを利用するには、QGIS 3.4（執筆時点の最新版）が必要です。

すぐにはじめましょう！

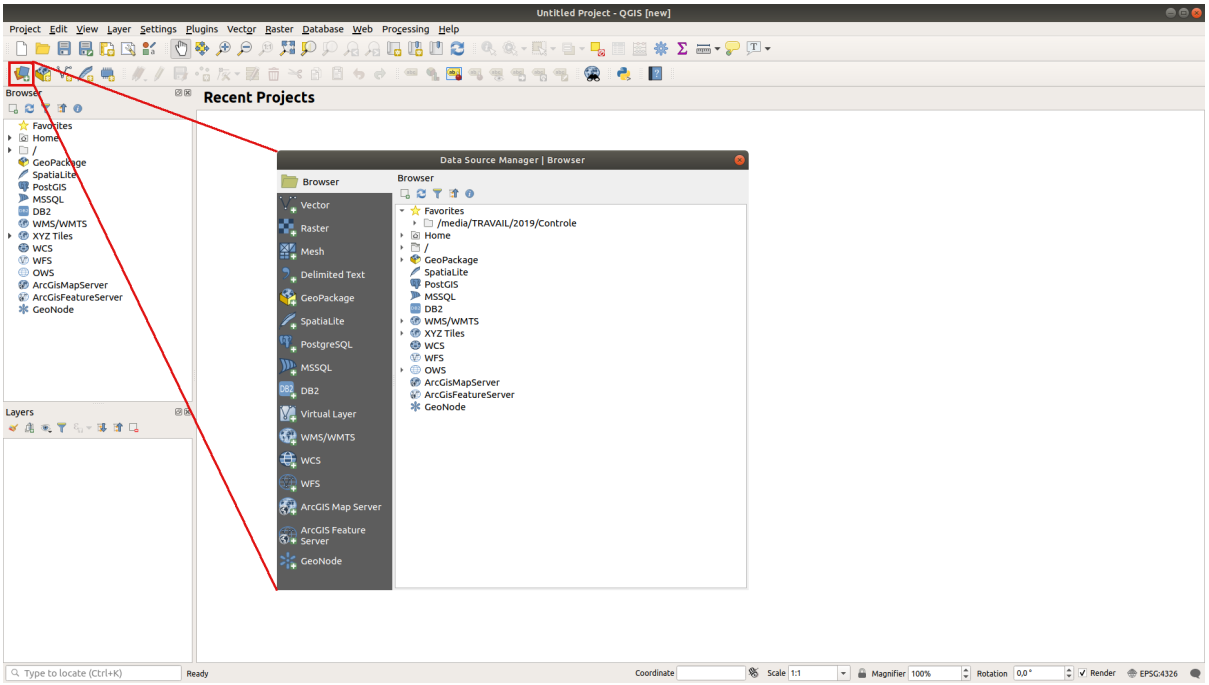
2.2.1 ★☆☆ （初級レベル） 理解しよう: 地図を準備する

1. QGIS を開きます。新しい白紙の地図ができます。



2. **データソースマネージャ** ダイアログでは、データ型に応じて読み込むデータを選択することができます。ここではデータセットを読み込むために使います:  データソースマネージャを開く ボタンをクリックします。

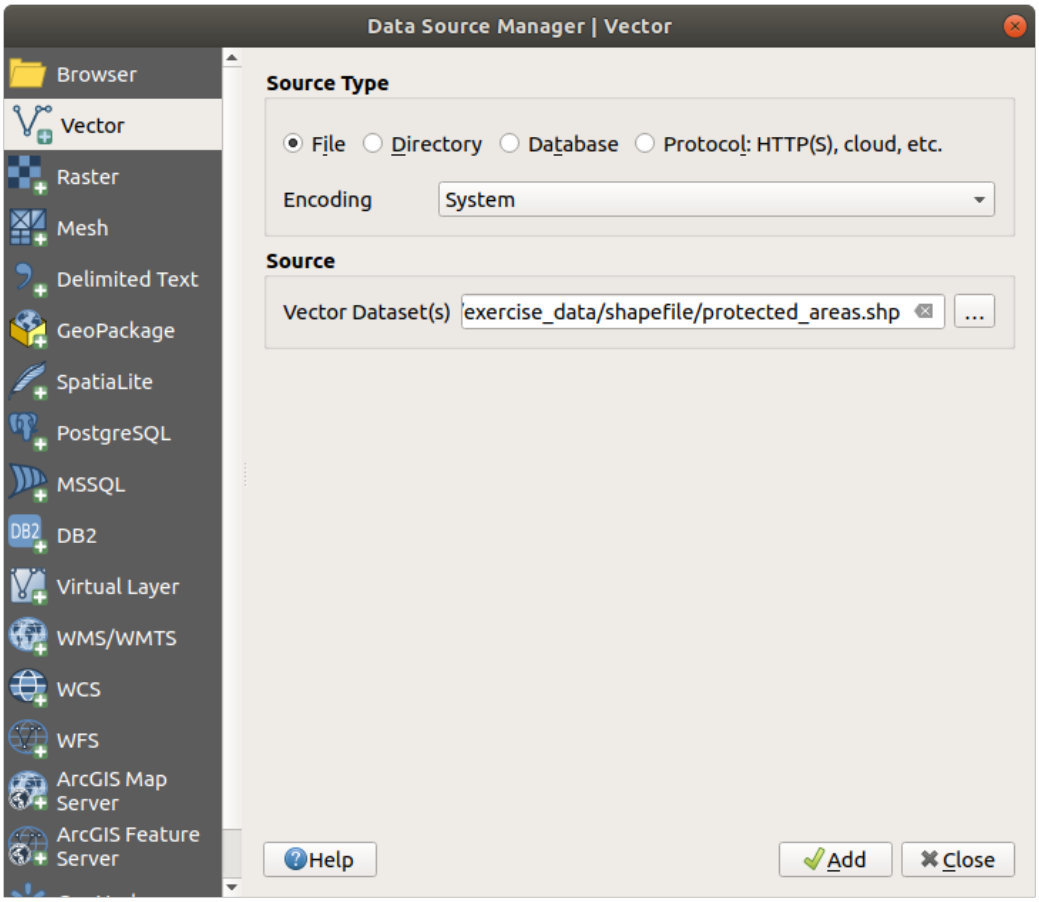
   アイコンが見つからないときは、**ビュー　ツールバー** メニューで **データソースマネージャ** ツールバーが有効にされていることを確認してください。



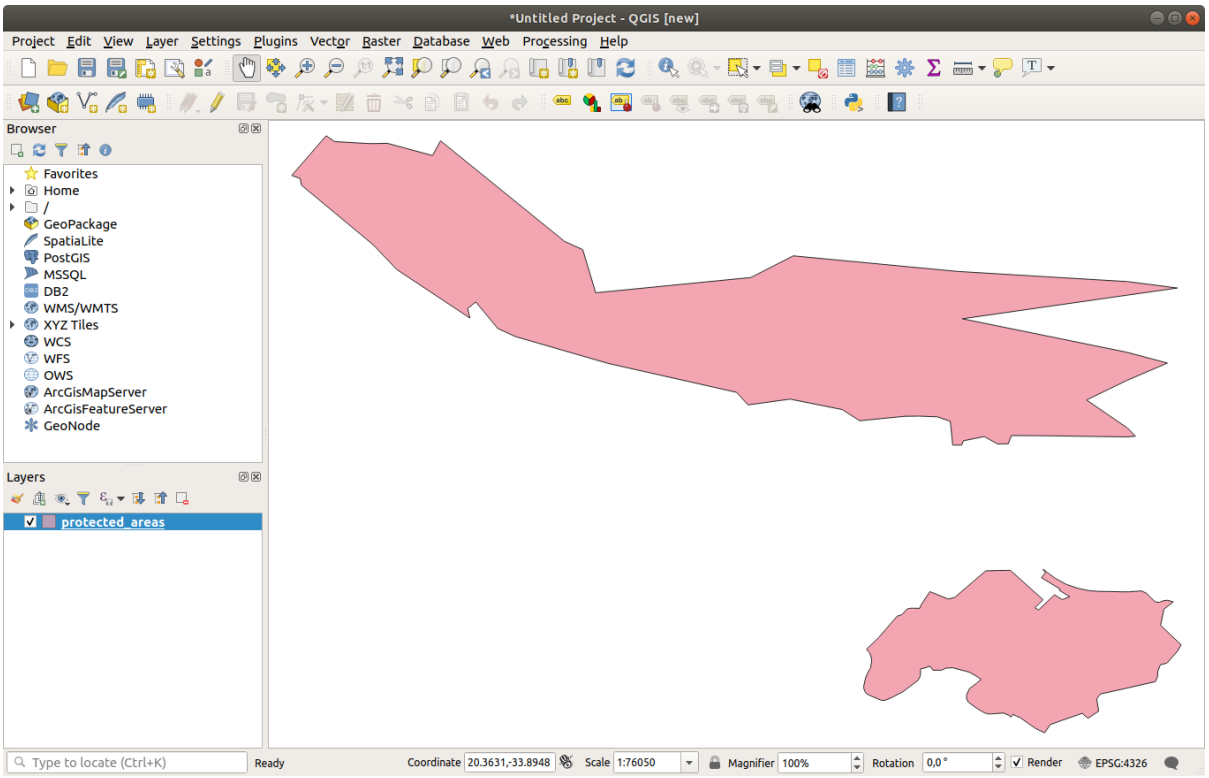
3. `protected_areas.shp` ベクタデータセットを読み込みます:

   1. **ベクタ** タブをクリックします。

2. **ファイル** ソースタイプを有効にします。
3. **ベクタデータセット** の隣にある ... ボタンを押します。
4. 訓練ディレクトリにある `exercise_data/shapefile/protected_areas.shp` ファイルを選択します。
5. **開く** をクリックします。元のダイアログにファイルパスが入力されています。



6. ここでも **追加** をクリックします。指定したデータが読み込まれます: **レイヤ** パネル（左下）に `protected_areas` アイテムが表示され、その地物がメインマップキャンバスに表示されているのが確認できます。



おめでとうございます！これで基本的な地図ができました。作業を保存するにはよいタイミングでしょう。

1. **名前を付けて保存** ボタンをクリックしてください：

2. 地図を `exercise_data` の隣にある `solution` フォルダの下に `basic_map.qgz` という名前で保存します。

2.2.2 ★☆☆ （初級レベル） 自分でやってみよう:

上記のステップを繰り返し、同じフォルダ (`exercise_data/shapefile`) から `places.shp` と `rivers.shp` レイヤを地図に追加してください。

答え

ダイアログのメインエリアに色の異なるたくさんのシェープが表示されているはずです。それぞれのシェープは、左側のパネルに表示されている色で識別できるレイヤに属しています（あなたのものは下の 図 2.1 の色とは異なるかもしれません）。

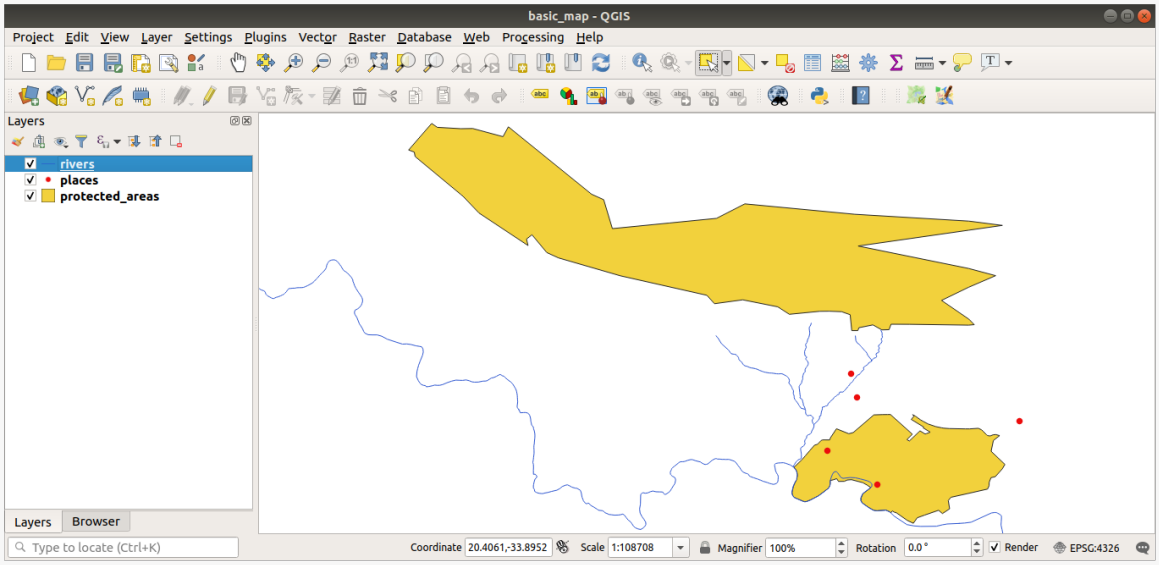


図 2.1: 基本的な地図

2.2.3 ★☆☆ （初級レベル） 理解しよう: GeoPackage データベースからベクタデータを読み込む

データベースを使用すると、大量の関連データを 1 つのファイルに保存することができます。Libreoffice Base や MS Access などのデータベース管理システム (DBMS) にはすでに馴染みがあるかもしれません。GIS アプリケーションもデータベースを利用することができます。GIS に特化した DBMS（PostGIS など）は、空間データを扱う必要があるため、特別な機能を備えています。

GeoPackage オープンフォーマットは、GIS データ（レイヤ）を 1 つのファイルに格納するためのコンテナです。 ESRI シェープファイルフォーマット（例 先ほど読み込んだ `protected_areas.shp` データセット）とは異なり、1 つの GeoPackage ファイルには異なる座標参照系の様々なデータ（ベクタおよびラスタデータ）、および空間情報のないテーブルを格納できます。これらの機能により、データを簡単に共有し、ファイルの重複を回避することができます。

GeoPackage からレイヤを読み込むには、まず、そのレイヤへの接続を作成する必要があります。

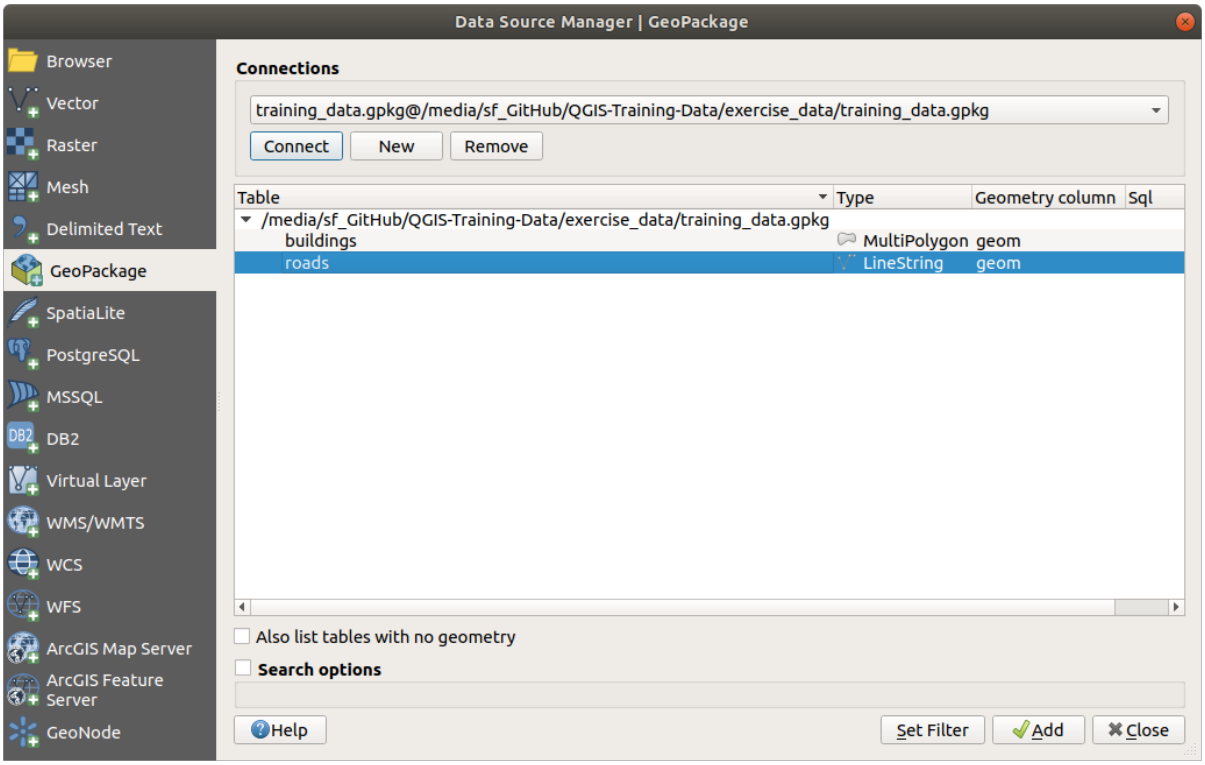
1. 

ボタンをクリックします。
2. 左側で  *GeoPackage* タブをクリックします。
3. **新規** ボタンをクリックし、前にダウンロードした `exercise_data` フォルダにある `training_data.gpkg` ファイルをブラウズします。
4. ファイルを選択し、 *Open* を押してください。ファイルパスが Geopackage 接続リストに追加され、ドロップダウンメニューに表示されます。

これで、この GeoPackage から任意のレイヤを QGIS に追加する準備が整いました。

1. **接続** ボタンをクリックします。ウィンドウの中央部分に、GeoPackage ファイルに含まれるすべてのレイヤのリストが表示されます。

2. Select the `roads` layer and click on the *Add* button.



A `roads` layer is added to the *Layers* panel with features displayed on the map canvas.

3. **閉じる** をクリックします。

おめでとうございます。GeoPackage から最初のレイヤを読み込みました。

2.2.4 ★☆☆（初級レベル）理解しよう: ブラウザで SpatiaLite データベースからベクタデータを読み込む

QGIS は他にも多くのデータベースフォーマットへのアクセスを提供しています。GeoPackage と同様に、SpatiaLite データベースフォーマットは SQLite ライブラリの拡張機能です。そして、SpatiaLite プロバイダからレイヤを追加することは、上記と同じルールに従います：接続を作成 --> 有効化 --> レイヤを追加。

これは SpatiaLite データをマップに追加する 1 つの方法ですが、データを追加するもう 1 つの強力な方法である **ブラウザ** を調べてみましょう。

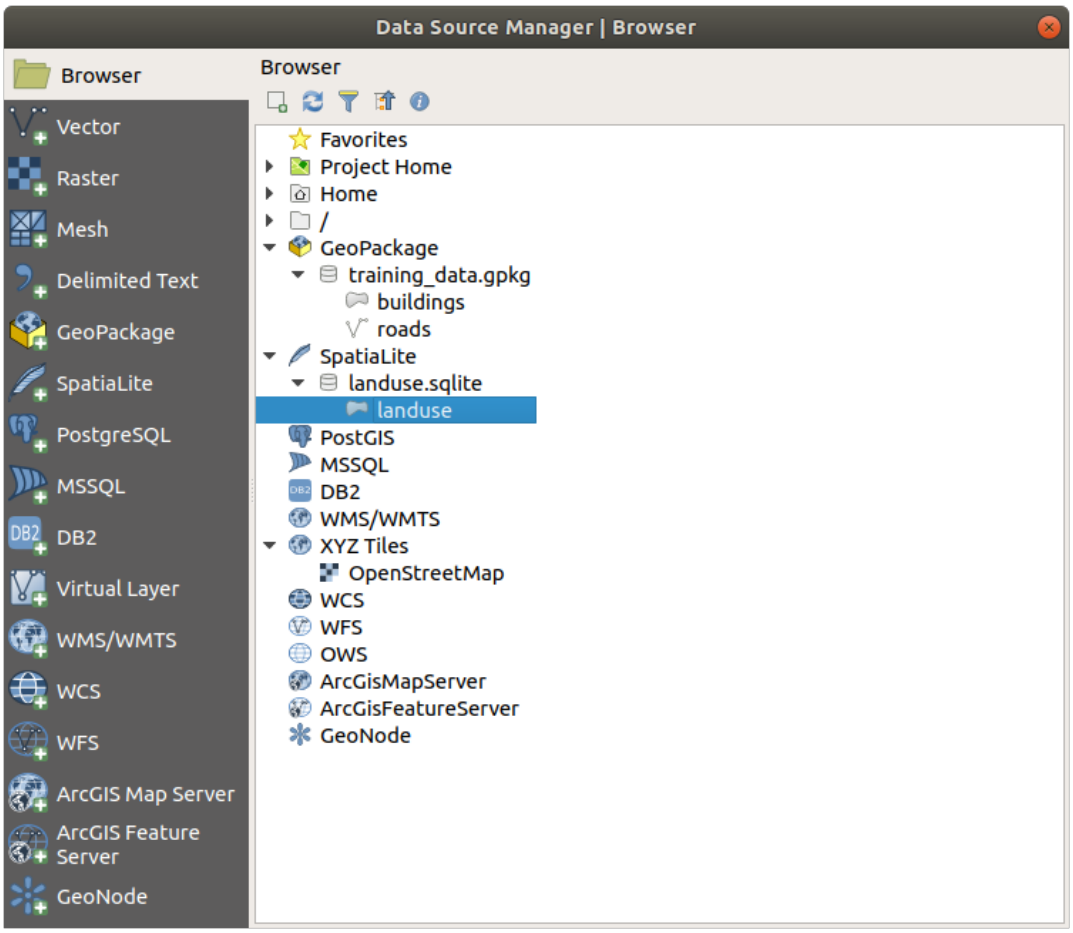
1. アイコンをクリックし、 **データソースマネージャ** ウィンドウを開きます。
2. **ブラウザ** タブをクリックします。
3. このタブには、コンピュータに接続されているすべてのストレージディスクと、左側のほとんどのタブのエントリが表示されます。これらにより、接続されているデータベースやフォルダに素早くアクセスすることができます。

   例えば、 *GeoPackage* エントリの横にあるドロップダウンのアイコンをクリックします。以前接続した `training-data.gpkg` ファイルが表示されます（展開されている場合はそのレイヤも表示さ

れます)。

4. *SpatiaLite* エントリを右クリックし、 **新規接続**... を選択します。
5. `exercise_data` フォルダに移動し、`landuse.sqlite` ファイルを選択し、**開く** をクリックします。

   SpatiaLite の下に *landuse.sqlite* というエントリが追加されていることに注意してください。
6. *landuse.sqlite* のエントリを展開します。
7. Double-click the `landuse` layer or select and drag-and-drop it onto the map canvas. A new layer is added to the *Layers* panel and its features are displayed on the map canvas.



Tip

ビュー　パネル　で **ブラウザ** パネルを有効にし、データを追加するのに使用します。**データソースマネージャ　ブラウザ** タブは同じ機能を備えた便利なショートカットです。

注釈

プロジェクトを頻繁に保存することを忘れないでください。プロジェクトファイルにはデータ自体は含まれていませんが、どのレイヤを地図に読み込んだかは記憶されています。

2.2.5 ★★☆ （中級レベル） 自分でやってみよう: さらにベクタデータを読み込む

上で説明した方法のどれかを使い、 `exercise_data` フォルダから次のデータセットを地図に読み込んでください：

- `buildings`
- `water`

答え

あなたの地図には七つのレイヤがある筈です。

- `protected_areas`
- `places`
- `rivers`
- `roads`
- `landuse`
- `buildings` (taken from `training_data.gpkg`) and
- `water` (taken from `exercise_data/shapefile`).

2.2.6 理解する: レイヤを並び替える

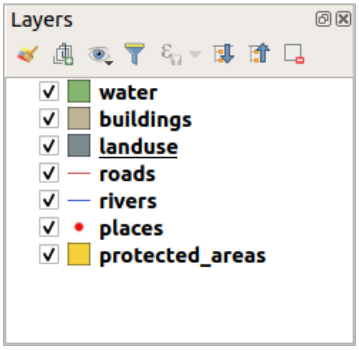
レイヤリストにあるレイヤは、特定の順序で地図上に描かれます。リストの一番下にあるレイヤが最初に描かれ、そして一番上のレイヤは最後に描かれます。リストに表示される順を変更することで、それらが描かれる順序を変更できます。

注釈

レイヤ順序 パネルの下にある **描画順序の制御** チェックボックスを使用して、この動作を変更することができます。しかし、この機能についてはまだ説明しません。

レイヤが地図にロードされた順序は、おそらくこの段階では論理的ではありません。他のレイヤがその上にあることで道路レイヤが完全に隠れている可能性があります。

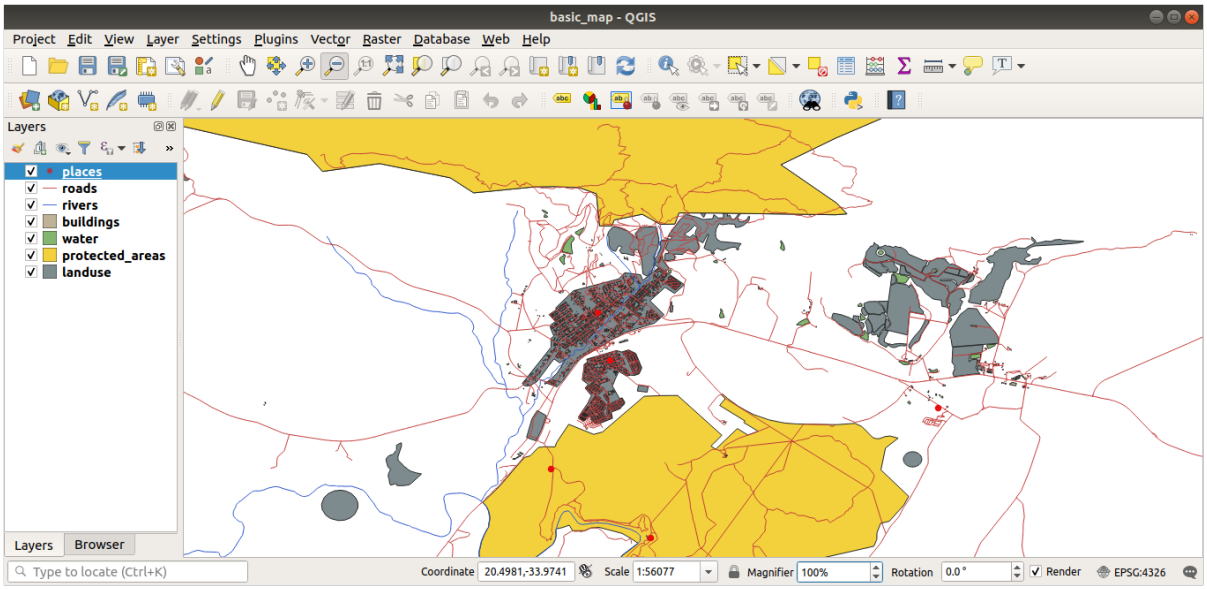
たとえば、このレイヤ順では・・・



・・・道路や地名が土地利用レイヤのポリゴンの **下** を通っているため、隠れてしまいます。

問題を解決するために:

1. レイヤリスト中でレイヤをクリックしてドラッグします。
2. こう見えるようにそれらの順番を入れ替えます:



地図は現在、土地利用領域の上に表示される道路や建物で、視覚的により理にかなっていることがわかります。

2.2.7 結論

これで、いくつかの異なるソースから必要なレイヤをすべて追加し、基本的な地図を作成することができました！

2.2.8 次は?

データソースマネージャを開く ボタンの基本的な機能は理解できたと思いますが、他の機能についてはどうでしょう？このインターフェイスはどのように機能するのでしょうか？先に進む前に、QGIS のインターフェイスの基本的な操作について見てみましょう。これは次のレッスンのトピックです。

2.3 レッスン: マップキャンバスを見てまわる

このセクションでは、マップキャンバス内の操作に使用される基本的な QGIS ナビゲーションツールに焦点を当てます。これらのツールは、異なる縮尺でレイヤを視覚的に探索することを可能にします。

このレッスンの目標 ： QGIS 内のパンとズームツールの使用方法を学び、地図の縮尺について学ぶ。

2.3.1 ★☆☆ （初級レベル） 理解しよう: 基本の画面操作ツール

マップキャンバスの操作方法を学ぶ前に、このチュートリアルで探索することができるいくつかのレイヤを追加してみましょう。

1. 新しい空白のプロジェクトを開き、 **地図を準備する** で学んだ手順で、以前見た `protected_areas` 、 `roads` 、 `buildings` レイヤをプロジェクトに読み込みます。結果は 図 2.2 のような表示になるはずです（色は関係ありません）：

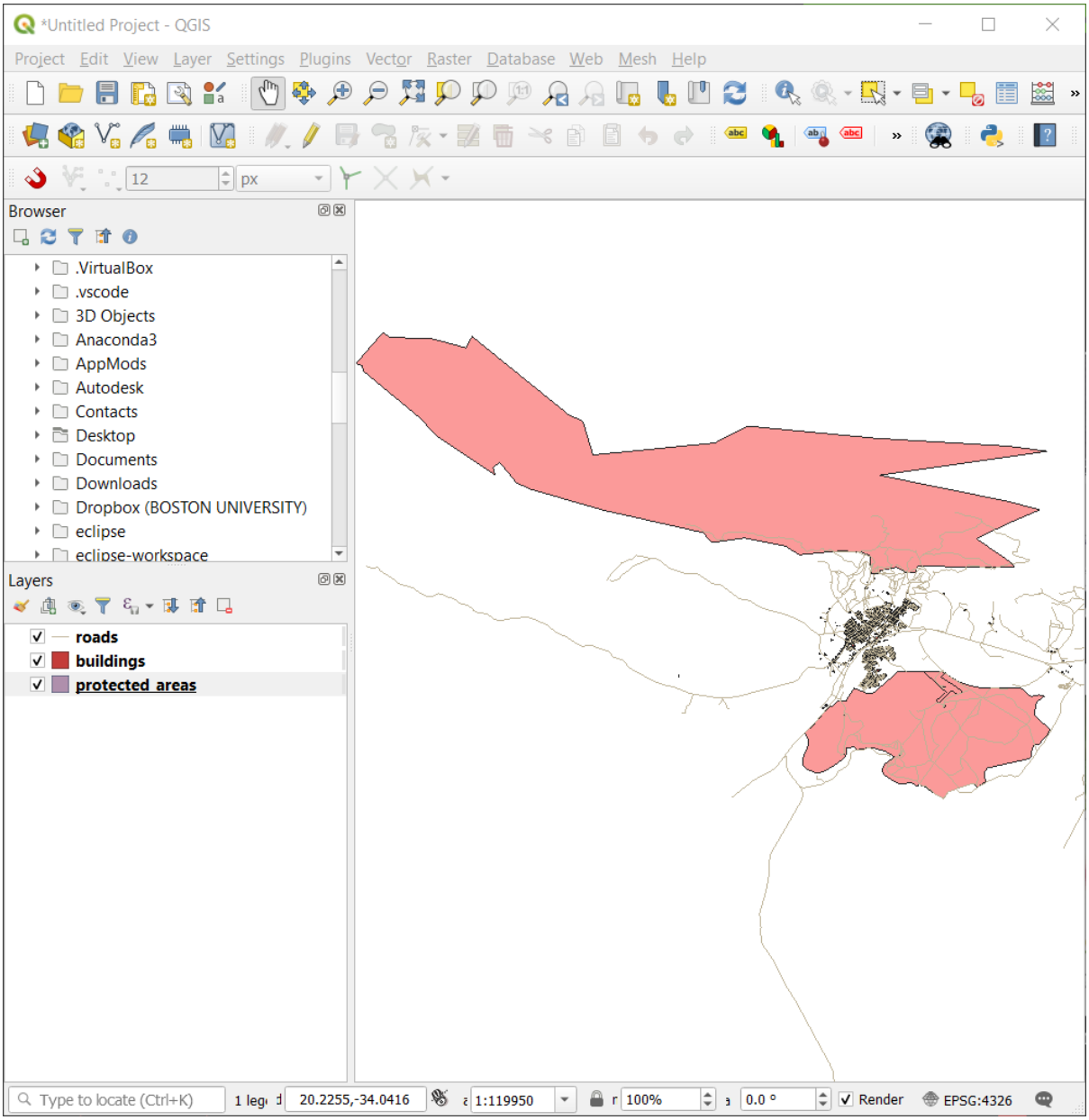


図 2.2: Protected areas、roads、buildings が追加された

まず、パンツールの使い方を学びましょう。

1. **ナビゲーションツールバー** で、 地図を移動 ボタンが有効になっていることを確認します。
2. マウスをマップキャンバスの中央に移動させます。
3. 左クリックしたまま、マウスを任意の方向にドラッグすると、地図がパンされます。

次に、拡大して、インポートしたレイヤを詳しく見てみましょう。

1. **ナビゲーションツールバー** で、 拡大 ボタンをクリックします。
2. 建物や道路が密集している左上付近までマウスを移動します。
3. 左クリックしたままにしておきます。

4. マウスをドラッグすると矩形が作られ、建物や道路の密集したエリアを覆います（ 図 2.3 ）。

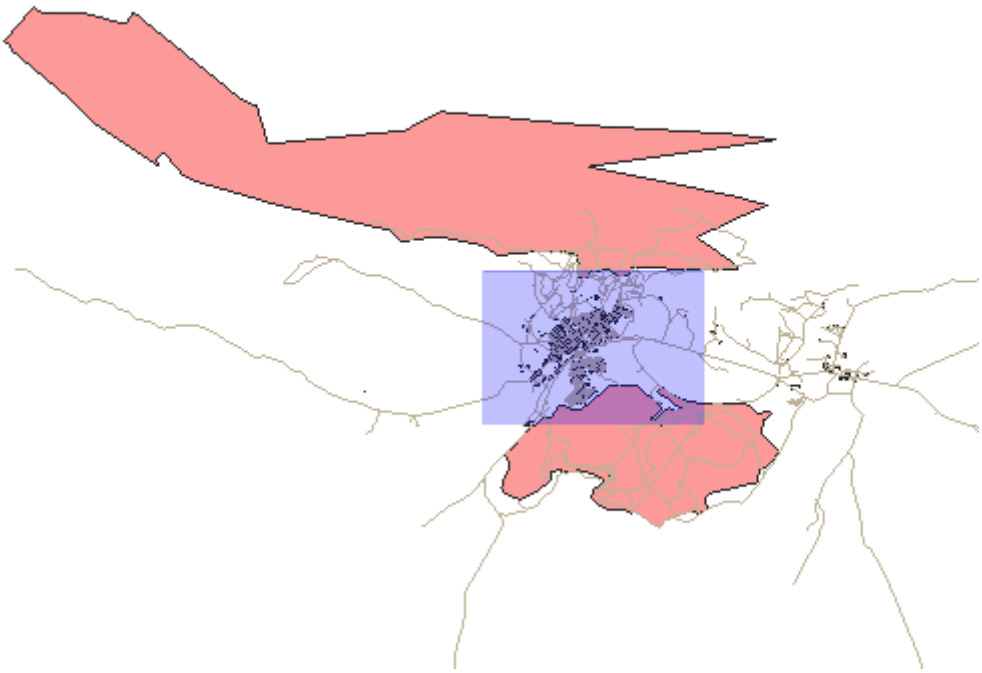
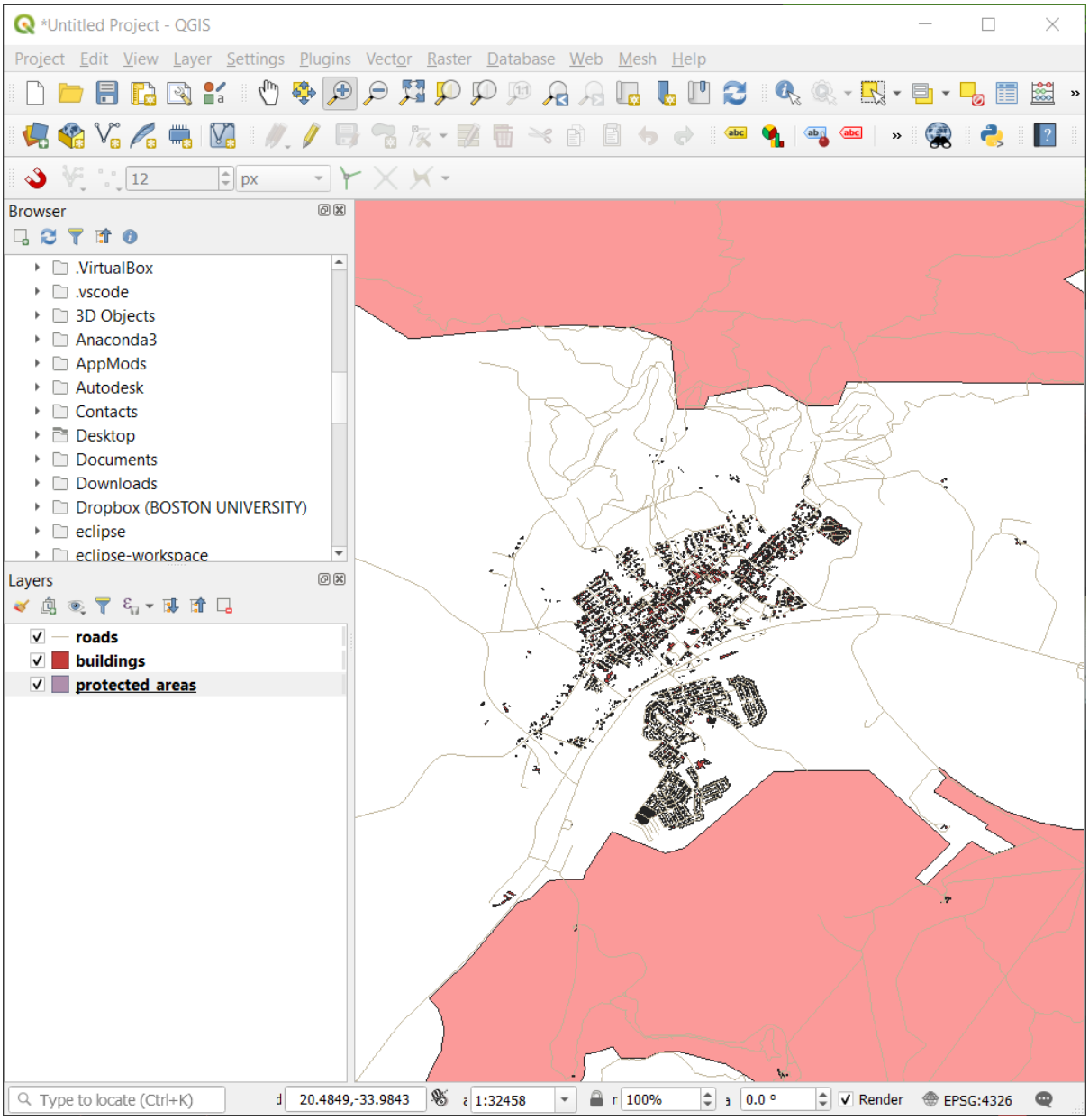


図 2.3: 拡大

5. 左クリックを離します。これで、矩形で選択した領域が含まれるように拡大されます。



6. 縮小するには、 

ボタンを選択し、拡大したときと同じ操作を実行します。

パン、拡大または縮小すると、QGIS はこれらのビューを履歴に保存します。これにより、前のビューに戻ることができます。

1. **ナビゲーションツールバー** で、 

ボタンをクリックすると、前のビューに移動します。
2. 

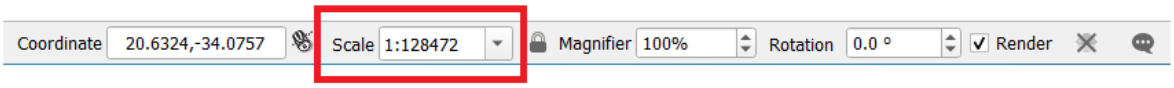
ボタンをクリックすると、履歴を先に辿ることができます。

データを探索した後、全てのレイヤの範囲にビューをリセットする必要がある場合があります。縮小ツールを何度も使う代わりに、QGIS はそのアクションを行うためのボタンを提供しています。

1. 

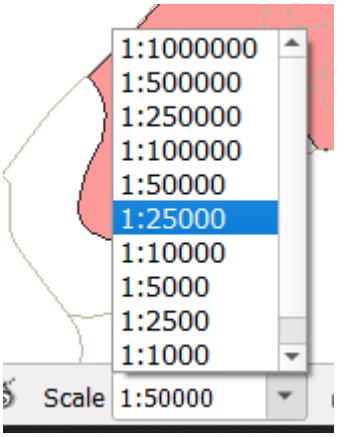
ボタンをクリックします。

拡大および縮小すると、ステータスバーの **縮尺** 値が変化することに注意してください。 **縮尺** 値は、地図の縮尺を表します。一般に、 **：** の右側の数字は、マップキャンバスに表示されている対象物が、実世界の実際の対象物よりも何倍小さいかを表します。



このフィールドを使用して、地図の縮尺を手動で設定することもできます。

1. ステータスバーの **縮尺** テキストフィールドをクリックします。
2. `50000` を入力し、 `Enter` キーを押します。これにより、マップキャンバスの地物が、入力された縮尺を反映して再描画されます。
3. または、 **縮尺** フィールドのオプション矢印をクリックすると、事前に設定された地図の縮尺が表示されます。



4. *1:5000* を選択します。これにより、マップキャンバスの地図の縮尺も更新されます。

これで、マップキャンバスを操作するための基本的な方法がわかりました。マップキャンバスを操作する別の方法については、ユーザーマニュアルの ズームとパン を参照してください 。

2.3.2 結論

マップキャンバスの操作方法を理解することは、レイヤの探索と視覚的な検査を可能にするために重要です。これは、最初のデータ調査や、空間解析のアウトプットを検証するために行うことができます。

2.4 レッスン: シンボロジ

レイヤのシンボロジとは、地図上におけるその外観です。空間的な側面を持つデータを表現する他の方法と比較した GIS の基本的な強みは、GIS では扱うデータを動的な視覚表現で把握できることにあります。

したがって、（個々のレイヤのシンボロジに依存する）地図の外観は非常に重要です。作成した地図のエンドユーザーに地図が何を表現するかが簡単に分かるようにする必要があります。同様に重要なのは、作業中にデータを探索できることであり、優れたシンボロジはこれを大いに助けます。

つまり、適切なシンボロジを持つことは、贅沢でもなければ単にあると助かることでもありません。実のところ、適切に GIS を使用して、地図や、人々が使用できる情報を生成するために不可欠です。

このレッスンの目標: ベクタレイヤに対して、望むとおりのシンボロジを作成できるようになる

2.4.1 ★☆☆ （初級レベル） 理解しよう: 色を変更する

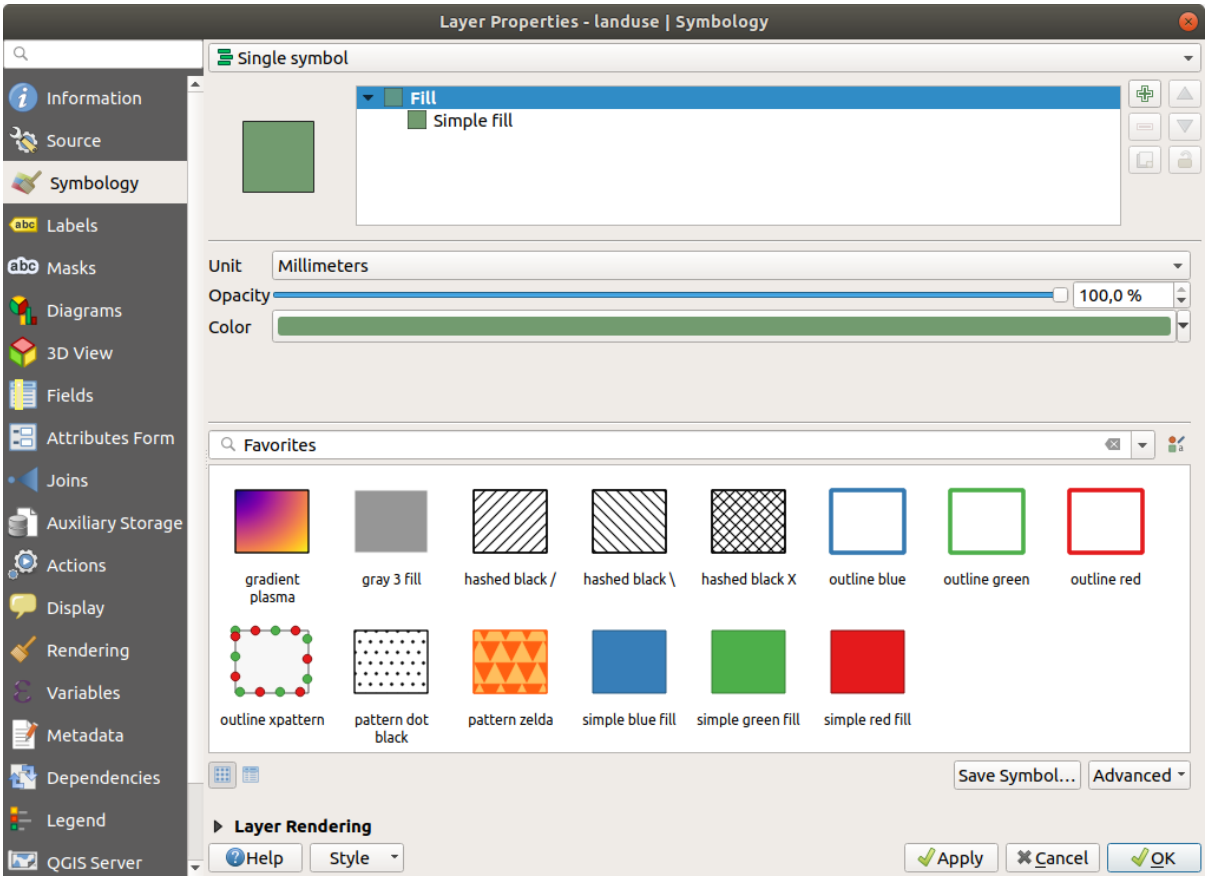
レイヤのシンボロジを変更するには、そのレイヤの **レイヤのプロパティ** を表示します。 `landuse` レイヤの色を変更することから始めましょう。

1. レイヤリストから `landuse` レイヤを右クリックします。
2. 表示されたメニューの中から **プロパティ...** メニュー項目を選択します。

> **注釈**
>
> デフォルトでは、レイヤリストにあるレイヤをダブルクリックすることで、レイヤのプロパティにアクセスできます。

> **Tip**
>
> **レイヤ** パネルの上部にある  ボタンをクリックすると、 **レイヤスタイリング** パネルが表示されます。このパネルを使って、レイヤのいくつかのプロパティを変更することができます。デフォルトでは、変更はすぐに適用されます！

3. **レイヤプロパティ** ウィンドウで、  **シンボロジ** タブを選択します。



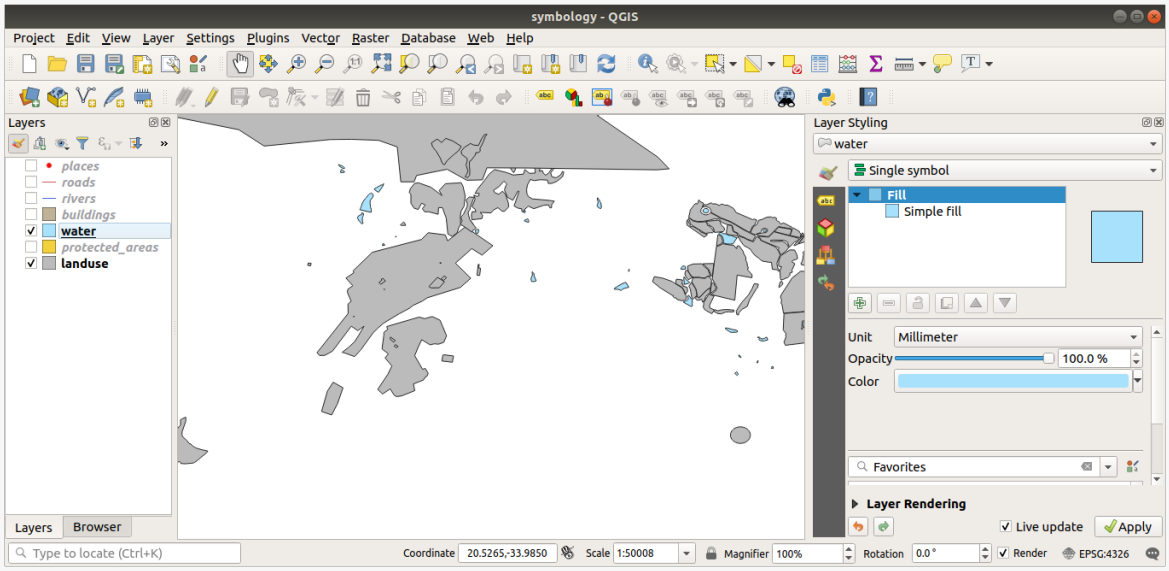
4. **色** ラベルの横にあるカラー選択ボタンをクリックします。標準的なカラーダイアログが表示されます。
5. グレーを選択して、 *OK* をクリックします。
6. **レイヤプロパティ** ウィンドウで再度 *OK* をクリックし、色の変更がレイヤーに適用されたことを確認するでしょう。

2.4.2 ★☆☆ （初級レベル） 自分でやってみよう:

`water` レイヤの色を水色に変更してください。**レイヤプロパティ** メニューではなく、 **レイヤスタイル** パネルを使ってみてください。

Solution

- 色が期待通りに変化していることを確認します。
- 凡例で `water` レイヤを選択し、  レイヤスタイルパネルを開く ボタンをクリックするだけです。色を水のレイヤに合ったものに変更します。



一度に 1 つのレイヤのみで作業を行い、他のレイヤに気を取られたくない場合、レイヤリストでその名前の横にあるチェックボックスをクリックすることで、レイヤを非表示にすることができます。ボックスが空白の場合、そのレイヤは非表示になっています。

2.4.3 ★☆☆ （初級レベル） 理解しよう: シンボル構造を変更する

ここまでは良いのですが、レイヤのシンボロジは色だけでなく、もっと重要なことがあります。次に、異なる土地利用区域の間の線を取り除いて、地図の視覚的な乱雑さを減らしたいと思います。

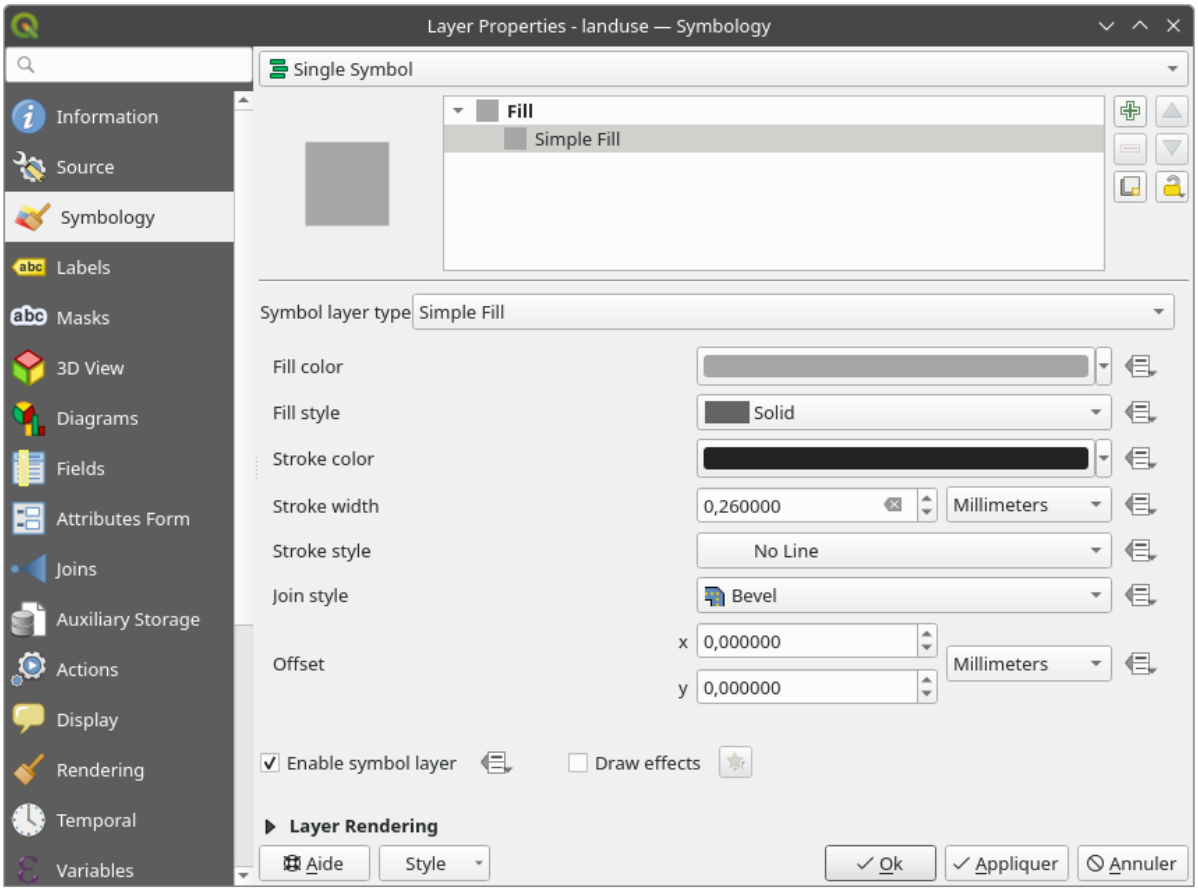
1. landuse レイヤの **レイヤプロパティ** ウィンドウを開きます。

   シンボロジ タブの下に、以前と同じようなダイアログが表示されます。しかし今回は、単に色を素早く変更するだけではありません。

2. シンボルレイヤツリーで、**塗りつぶし** ドロップダウンを展開し、**シンプル塗りつぶし** オプションを選択します。

3. **ストロークスタイル** のドロップダウンをクリックします。その時点では、短い線と **実線** という文字が表示されているはずです。

4. これを **線なし** に変更します。



5. *OK* をクリックします。

これで `landuse` レイヤは区域の間に線がなくなります。

2.4.4 ★☆☆ （初級レベル） 自分でやってみよう:

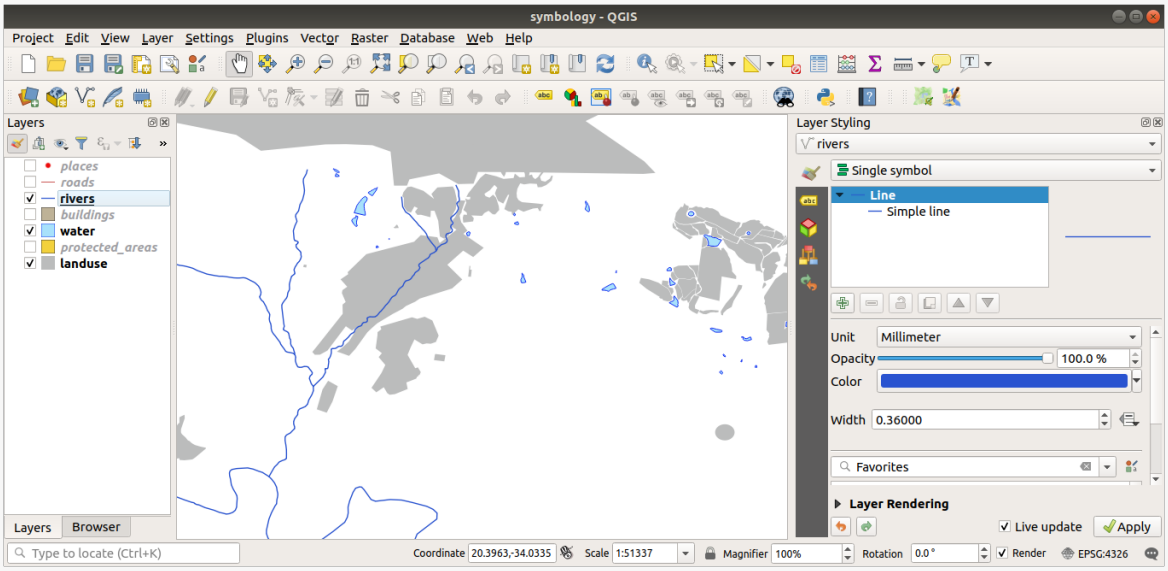
- 再び `water` レイヤのシンボロジを変更して暗い青の輪郭を持つようにします。
- `rivers` レイヤのシンボロジを水路を明確に表現するものに変更します。



ボタンを使うと、すべての変更を即座に確認することができます。そのパネルでは、レイヤのシンボル化中に個々の変更を元に戻すこともできます。

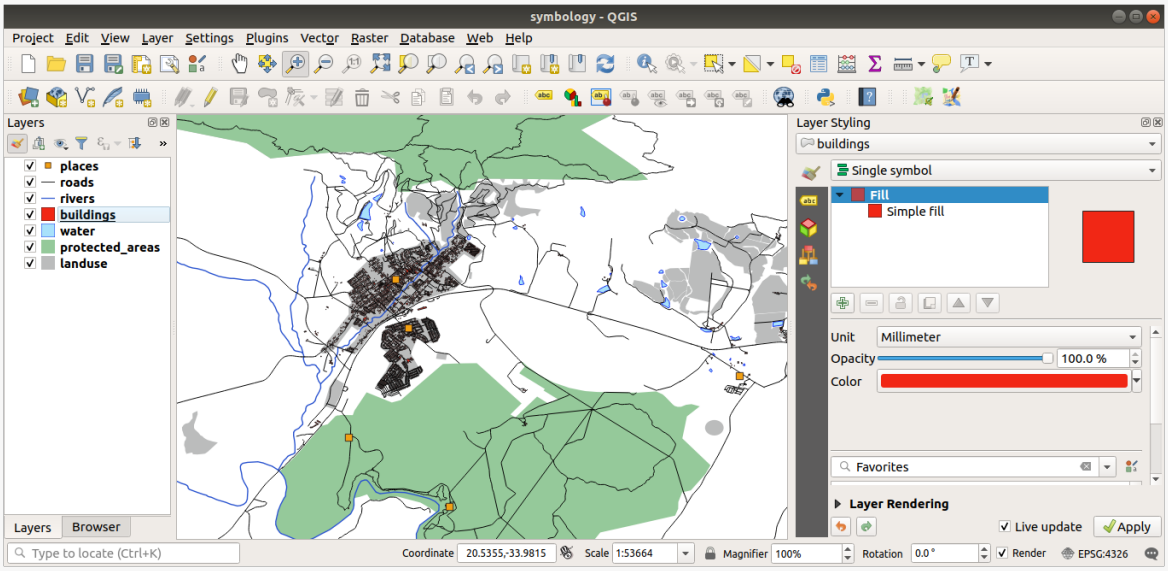
答え

これで地図は次のようになります：



初級者レベルの方は、ここでストップしていただいて結構です。

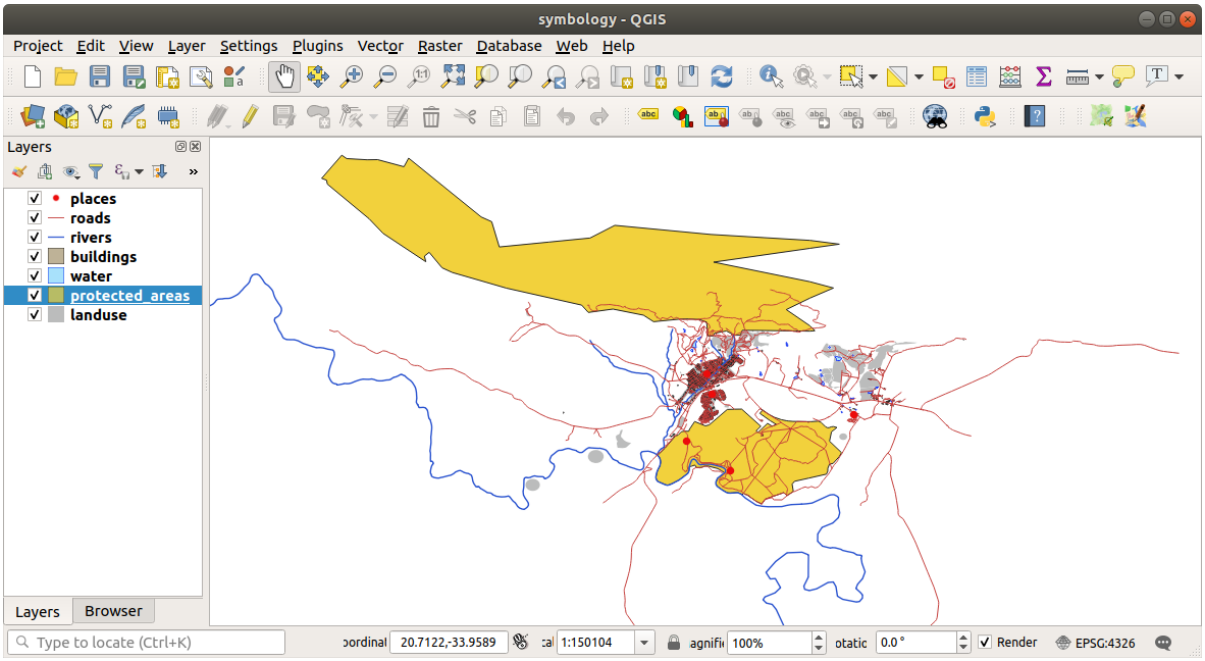
- 上記の方法で、残りのすべてのレイヤの色とスタイルを変更します。
- オブジェクトの色は自然な色を使ってみてください。例えば、道路は赤や青でなく、グレーや黒でもいいのです。
- また、ポリゴンの **塗りつぶしスタイル** や **ストロークスタイル** の設定も自由に変えてみてください。



2.4.5 ★★☆ （中級レベル） 理解する: 縮尺に基づく表示

時には、レイヤが所与の縮尺に適していないことがあります。例えば、全大陸のデータセットは、詳細さが低く、街区レベルではあまり正確ではないかもしれません。それが起こるときには、不適切な縮尺ではそのデータセットを非表示できるようにしたくなります。

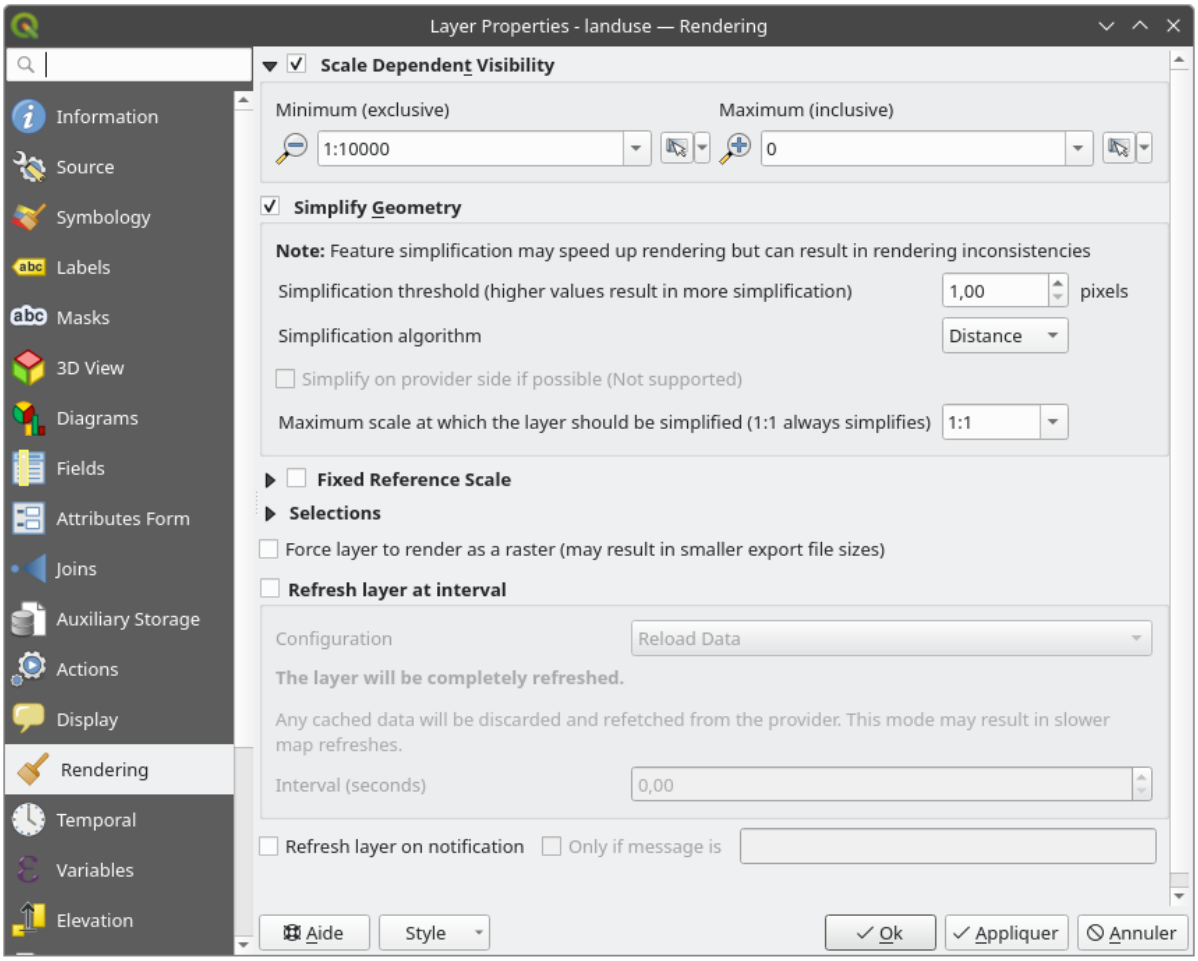
私たちの場合、小縮尺ではビューから建物を見えないようにすることにするかもしれません。このマップは、例えば...



…あまり便利ではありません。その縮尺で建物を区別するのは難しいです。

縮尺に基づくレンダリングを有効にするには：

1. buildings レイヤの **レイヤプロパティ** ダイアログを開きます。
2. **レンダリング** タブをアクティブにします。
3. **縮尺に応じた表示設定** というラベルのチェックボックスをクリックして、縮尺に基づいたレンダリングを有効にします。
4. **最小** 値を 1:10000 に変更します。



5. *OK* をクリックします。

``buildings` レイヤがいつ現れたり消えるかに注目しながら、地図で拡大・縮小する効果をテストします。

注釈

徐々に拡大するために、マウスホイールを使用できます。あるいは、ウインドウにズームするために、ズームツールを使用してください：

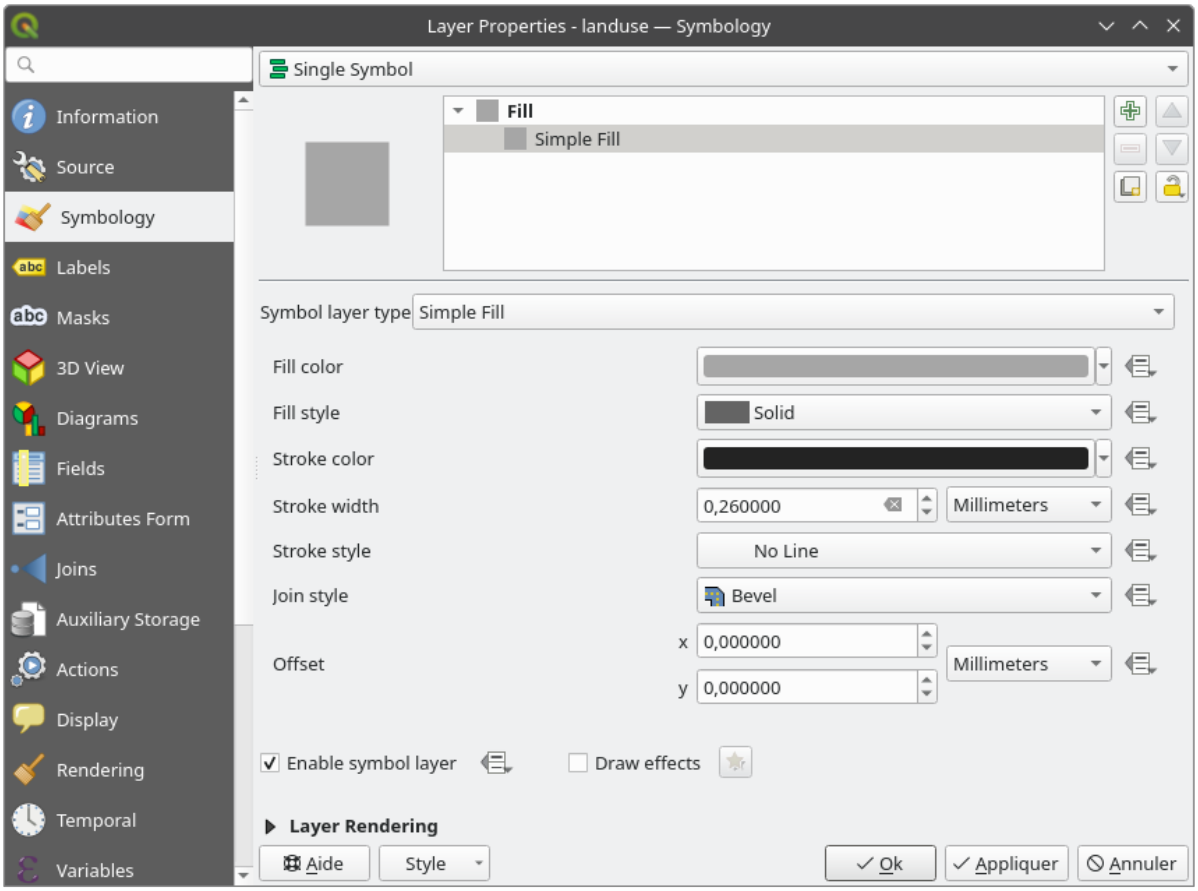


2.4.6 ★★☆ （中級レベル） 理解しよう: シンボルレイヤを追加する

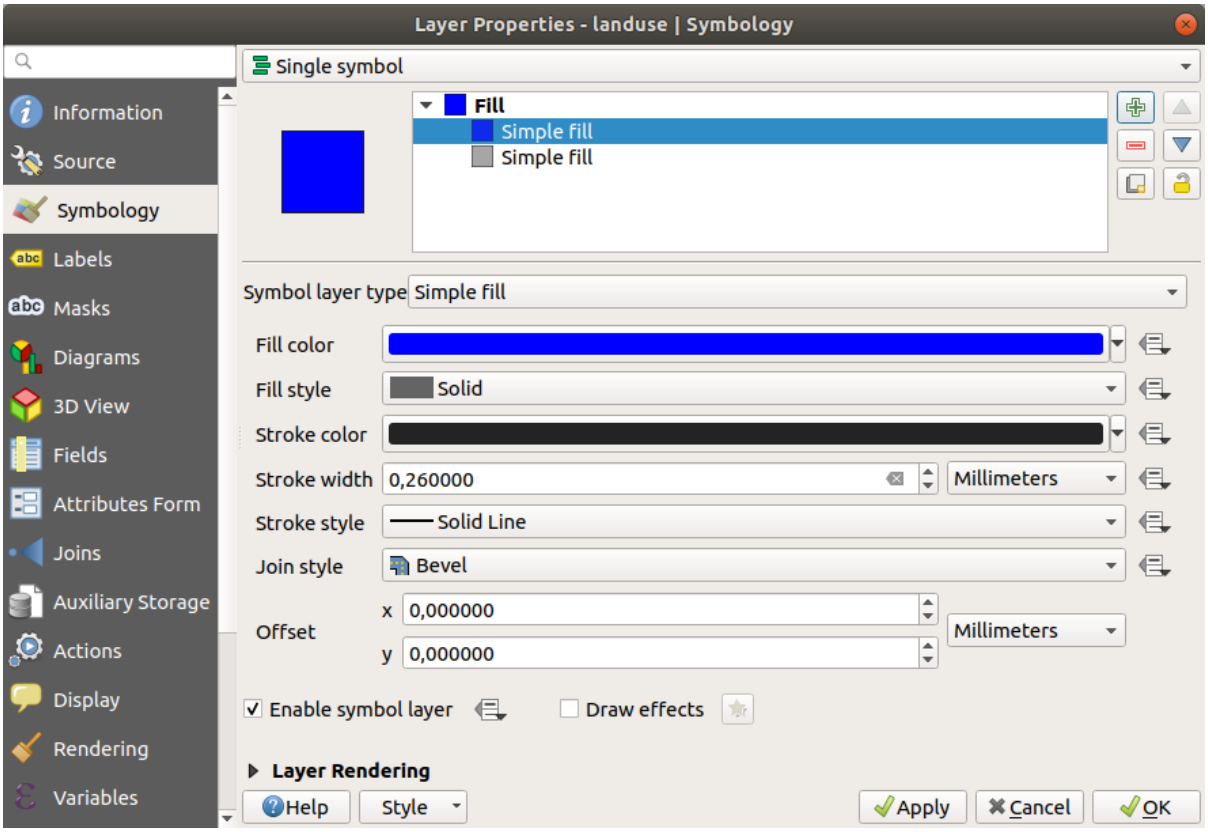
これでレイヤに対する単純なシンボロジを変更する方法は分かりましたので、次のステップは、より複雑なシンボロジを作成することです。QGIS では、シンボルレイヤを使用して、これを行うことができます。

1. （シンボルレイヤツリーの **シンプル塗りつぶし** をクリックして） landuse レイヤのシンボルプロパティパネルに戻ります。

   この例では、現在のシンボルは何のアウトラインを持っていません（すなわち、それは **線なし** 境界スタイルを使用しています）。



2. ツリーで **塗りつぶし** レベルを選択し、 シンボルレイヤを追加 ボタンをクリックします。ダイアログは次のようなものに変わり、新しいシンボルレイヤが追加されます。



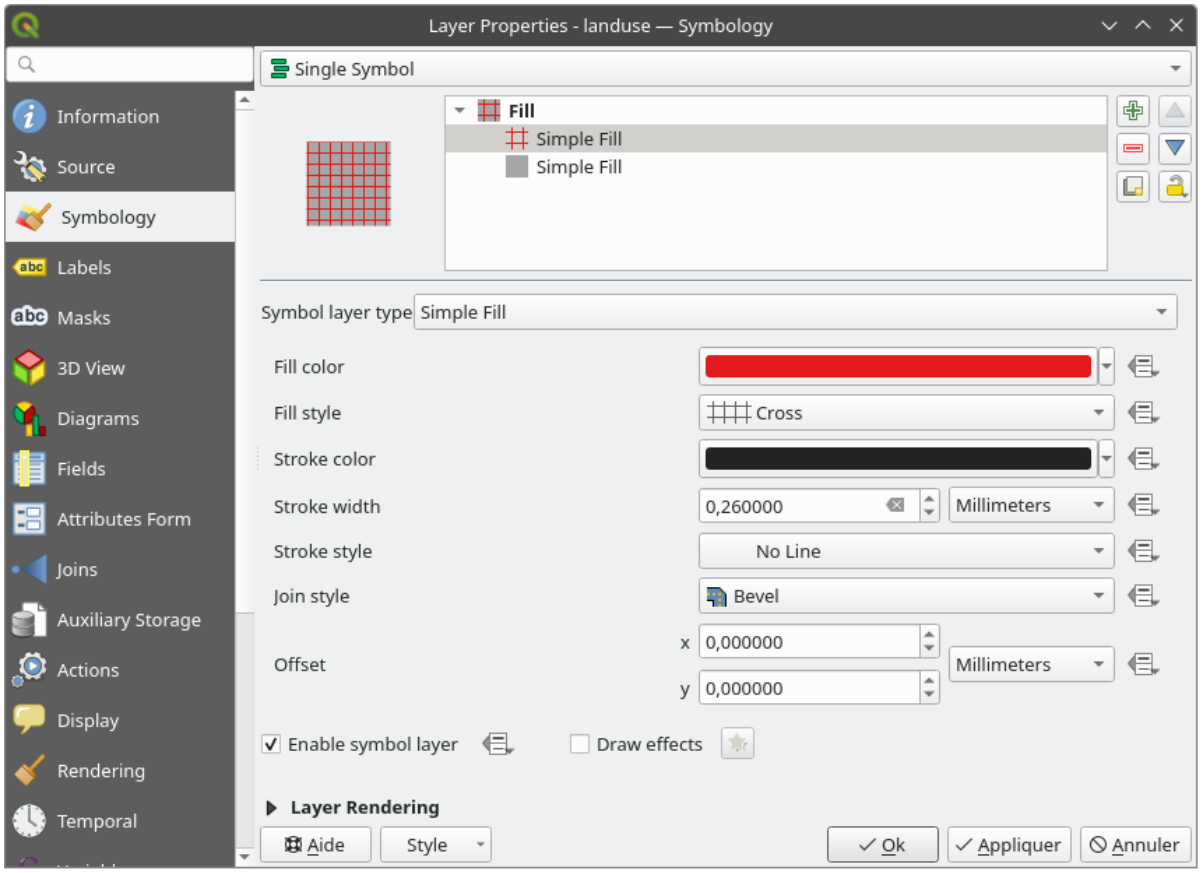
例えば色が多少違って見えるかもしれませんが、それはいずれ変えることになります。

今、第 2 のシンボルレイヤがあります。ソリッドカラーなので、もちろん前の種類のシンボルは完全に隠れます。さらに、これは **実線** の境界スタイルを持っていますが、これは私たちには必要ありません。明らかに、このシンボルを変更する必要があります。

> **注釈**
>
> 地図レイヤとシンボルレイヤを混同しないことが重要です。地図レイヤは地図に読み込まれているベクタ（またはラスタ）です。シンボルレイヤは、地図レイヤを表すために使用されるシンボルの一部です。このコースでは、たいていは地図レイヤを単にレイヤと言いますが、シンボルレイヤは、混乱を防ぐため常に、シンボルレイヤと呼びます。

選択された新しい **シンプル塗りつぶし** シンボルレイヤで:

1. 前のように、境界スタイルを **線なし** へ設定します。
2. 塗りつぶしスタイルを **ソリッド** か **ブラシなし** 以外に変更します。例えば：



3. *OK* をクリックします。

これで、結果を確認し、必要に応じて微調整することができます。複数の追加のシンボルレイヤを加えて、そのようにレイヤのテクスチャのようなものを作成することもできます。



これは楽しい！しかし、実際の地図に使うには色が多すぎるでしょう...

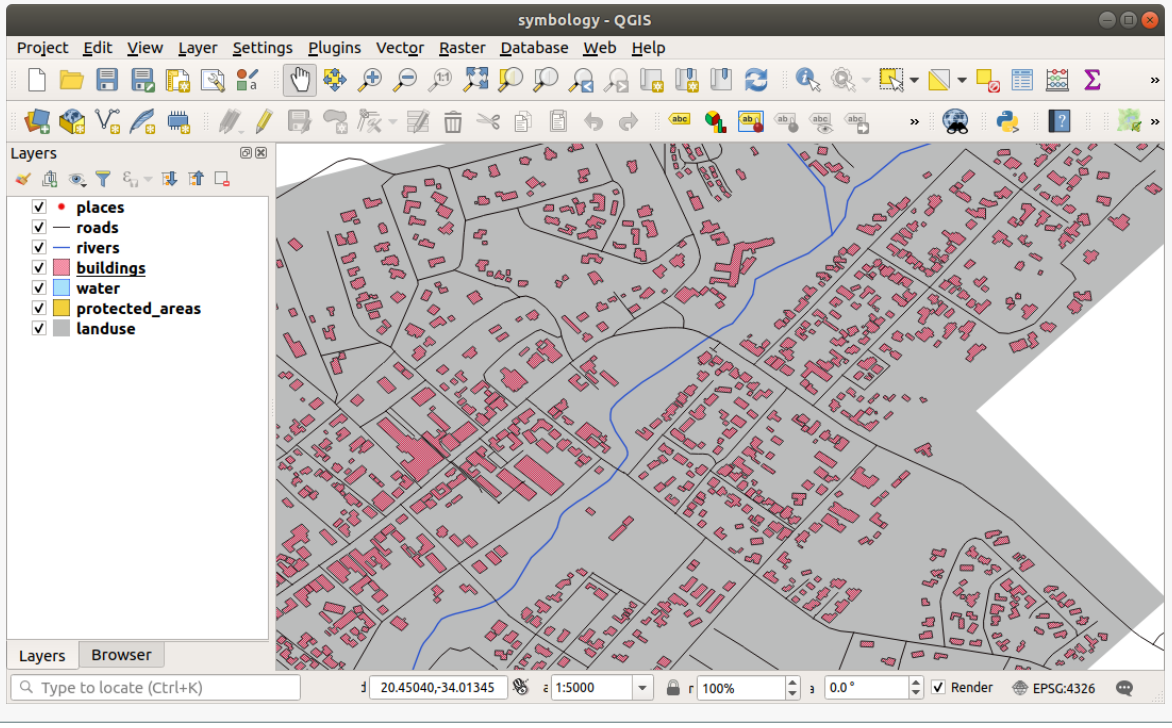
2.4.7 ★★☆ （中級レベル） 自分でやってみよう:

必要に応じて拡大することを思い出し、上記の方法を使用してシンプルだが邪魔にならないテクスチャを buildings レイヤに作りなさい。

答え

buildings レイヤは好きなようにカスタマイズできますが、地図上で異なるレイヤを簡単に見分けられるようにする必要があることを忘れないでください。

こちらが例です：



2.4.8 ★★☆ （中級レベル） 理解しよう: シンボルレベルを並び替える

シンボルレイヤがレンダリングされるとき、それらは、異なる地図レイヤがレンダリングされる方法と同様の順序でレンダリングされます。これは、場合によっては、一つのシンボルに多くのシンボルレイヤを有することは予期しない結果を引き起こす可能性があることを意味します。

1. （上で説明したシンボルレイヤを追加する方法を使って） roads レイヤに追加のシンボルレイヤを与えます。
2. ベースラインは **ストローク幅** を 1.5 にし、色は黒にします。
3. 新しい、一番上のレイヤの幅を 0.8 にし、色は白にします。

このようになることにお気づきでしょう。

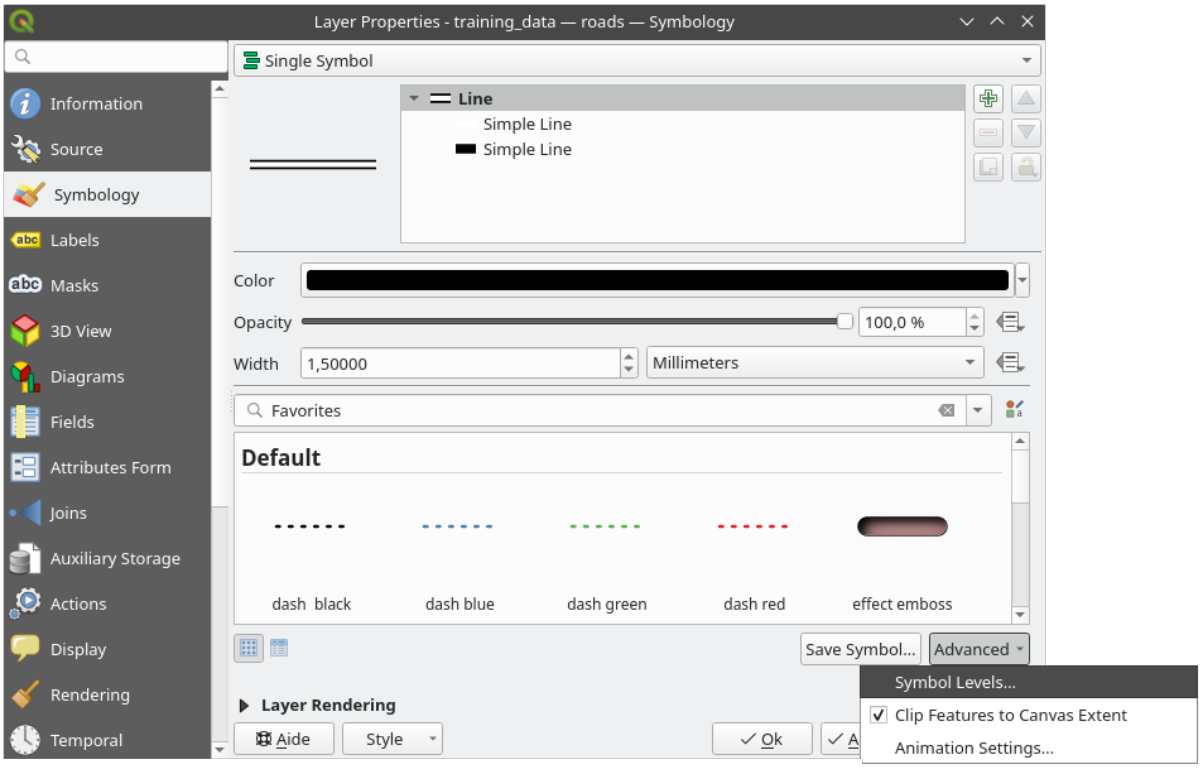


さて、道路は **道路** らしいシンボロジになっていますが、各交差点で線が互いに重なっているのがわかります。これは私たちが望むところではありません！

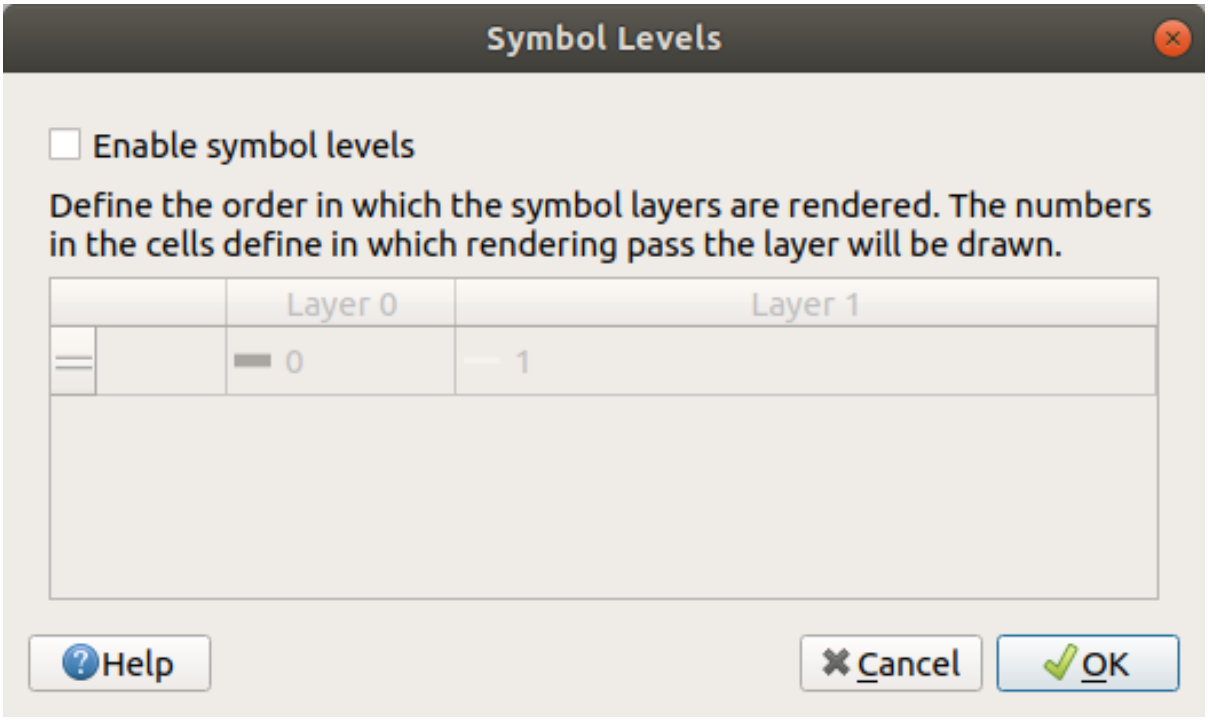
こうならないように、シンボルレベルを並べ替えることにより、異なるシンボルレイヤがレンダリングされる順序を制御することができます。

シンボルレイヤの順番を変更するには：

1. Select the topmost `Line` layer in the symbol layers tree.
2. ウィンドウの右下にある **詳細設定** **描画順序**... をクリックします。

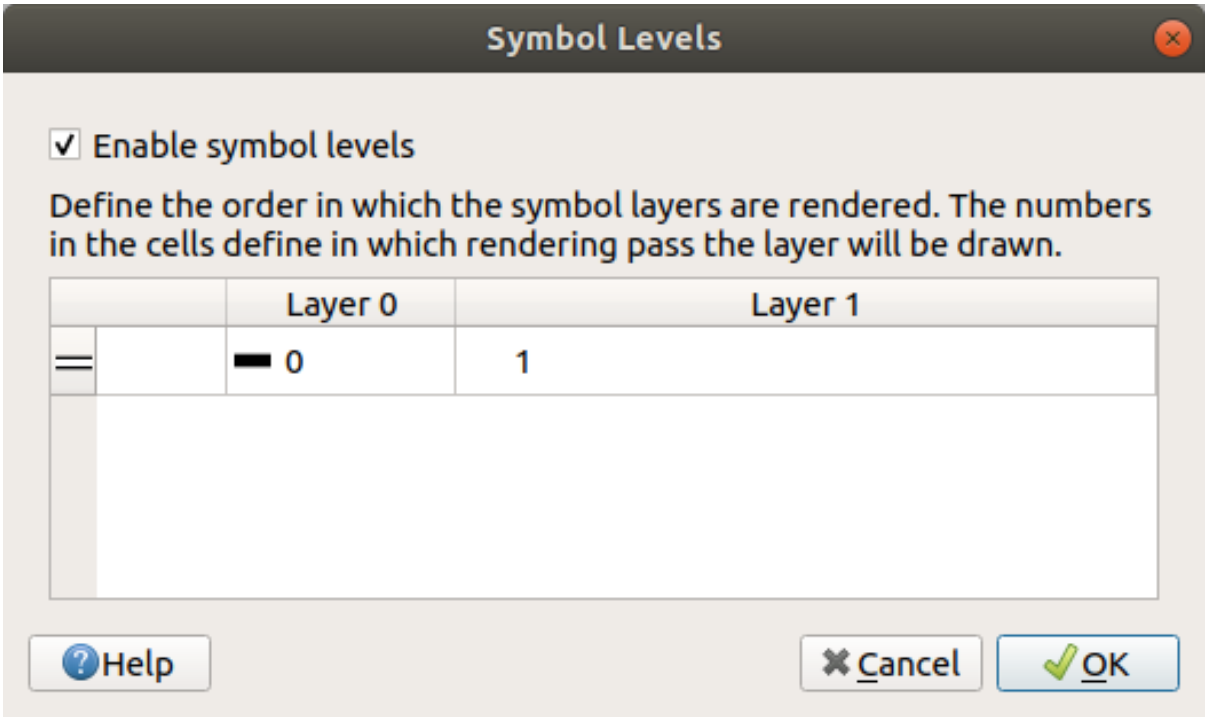


すると、このようなダイアログが表示されます：



3. **描画順序を有効にする** にチェックを入れます。そして、対応するレベル番号を入力することで、各シンボルのレイヤ順序を設定することができます。0 は最下層です。

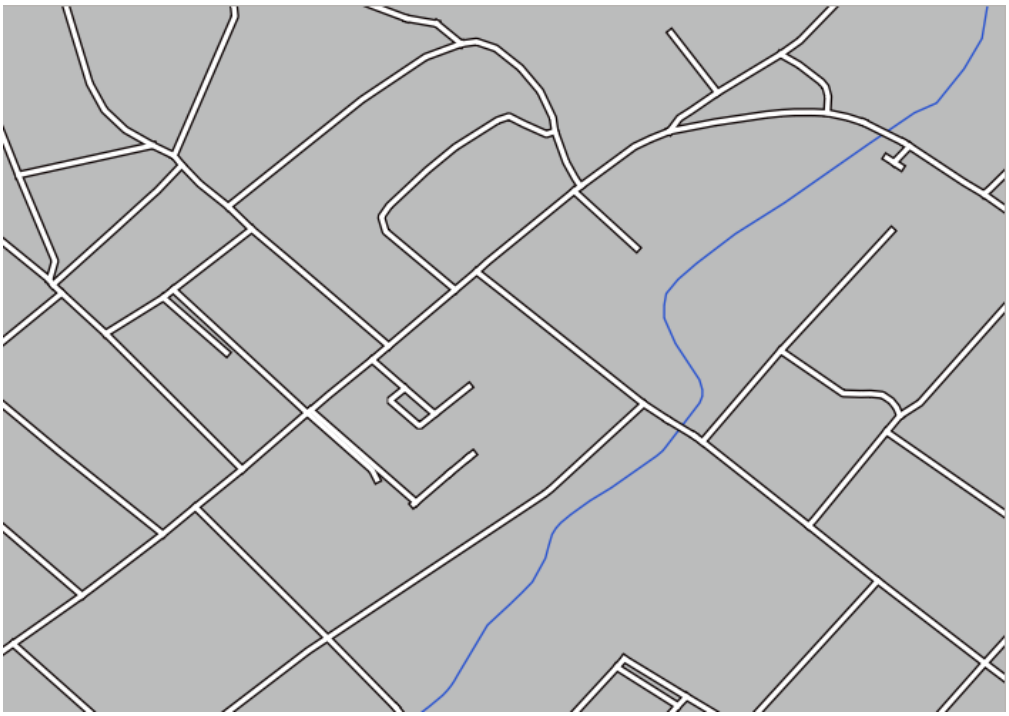
   今回の場合は、このようにオプションを有効にするだけです：



これにより、太い黒線の境界の上に白線が描画されます。

4. *OK* を 2 回クリックし、地図に戻ります。

   地図はこのように見えるでしょう:



終了したら、将来シンボルを再び変更するときに作業が失われないように、シンボル自体を保存することを忘れないでください。現在のシンボルスタイルは、 **レイヤプロパティ** ダイアログの下部にある **スタイルの保存**... ボタンをクリックすることで保存することができます。ここでは、 *QGIS* の *QML* **スタイルファ**

イル フォーマットを使用します。

スタイルは、 `solution/styles/better_roads.qml` フォルダに保存します。以前に保存したスタイルは、**スタイルの読み込み**... ボタンをクリックすることで、いつでも読み込むことができます。スタイルを変更する前に、置き換えようとしている未保存のスタイルが失われることに留意してください。

2.4.9 ★★☆ （中級レベル） 自分でやってみよう:

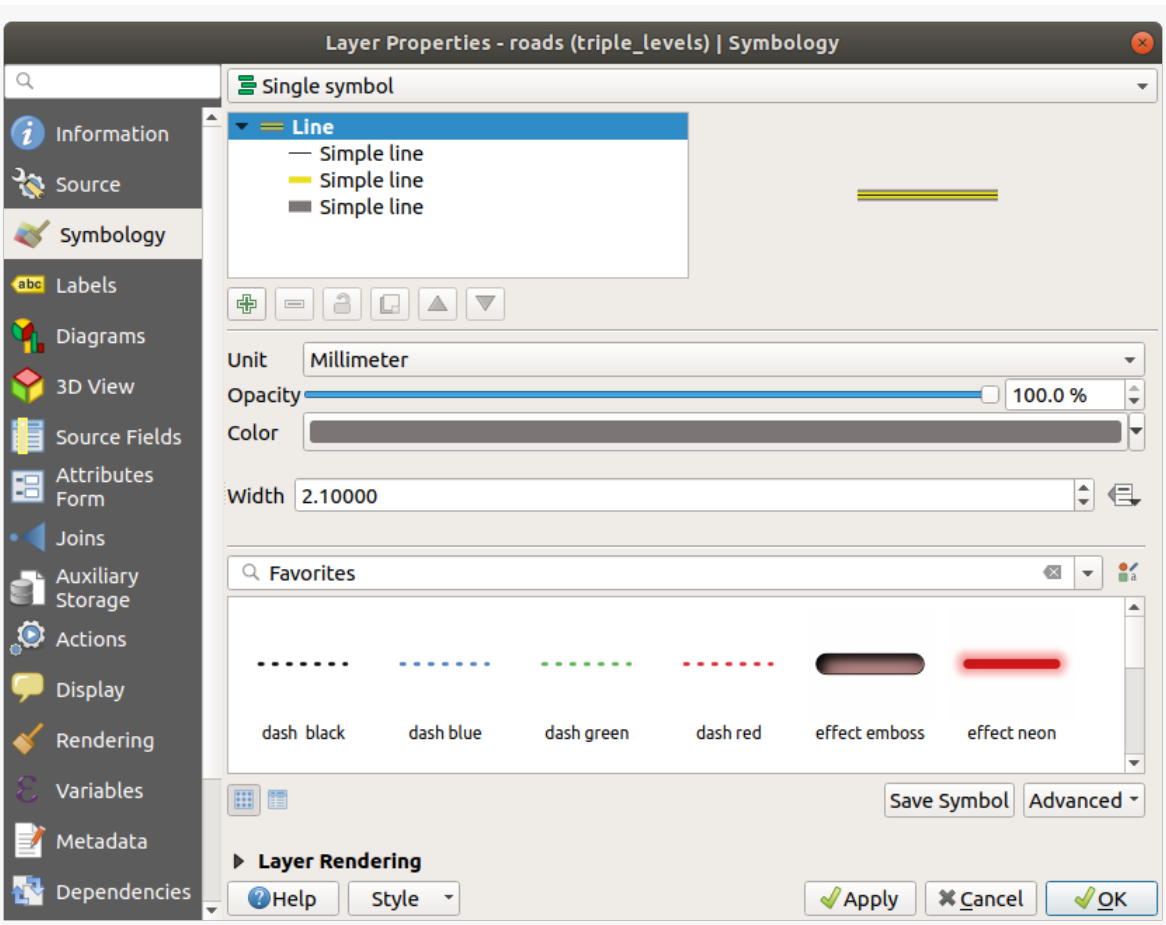
再び `roads` レイヤの外観を変更してください。

道路を狭く、黄色にし、薄いグレーの輪郭と真ん中に細い黒の線を入れます。レイヤのレンダリング順序を **詳細設定** **描画順序**... ダイアログで変更する必要があることを忘れないでください。



答え

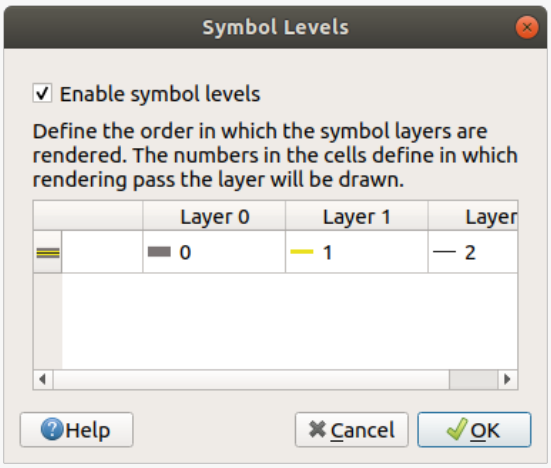
要求されたシンボルを作るには三つのシンボルレイヤが必要です。



最下層のシンボルレイヤは広く、単色のグレーな線です。その上に僅かに細い単色の黄色の線と最後に別の細い単色の黒線があります。

上記のようなシンボルレイヤで、思うような結果が得られない場合：

1. シンボルレベルが以下のようになっていることを確認してください：



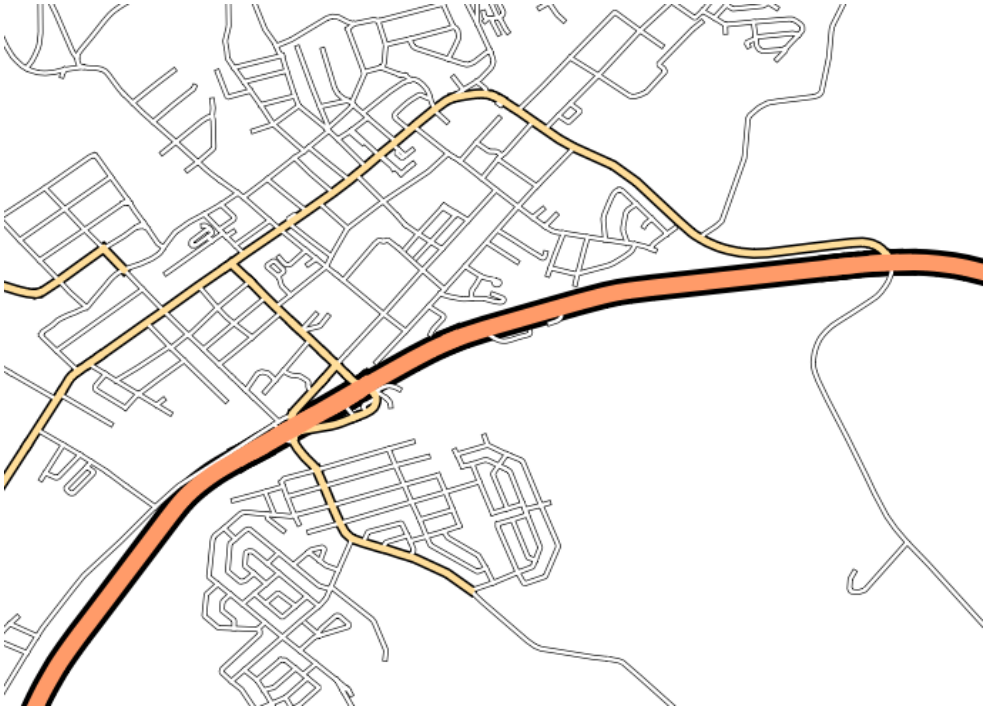
2. これで、地図は次のようになります：



2.4.10 ★★★ （上級レベル） 自分でやってみよう:

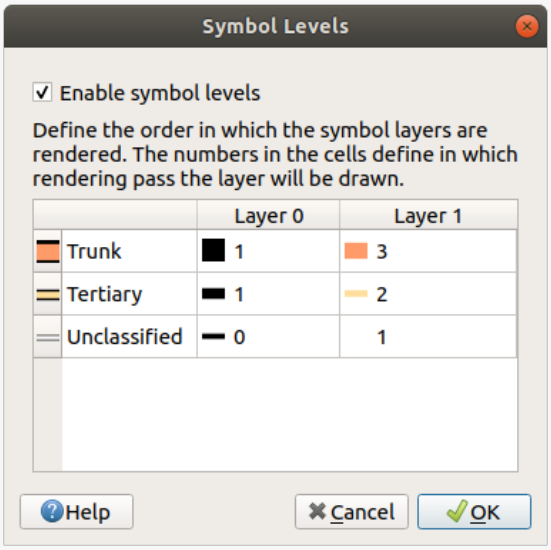
シンボルレベルは、分類されたレイヤ（複数のシンボルを持つレイヤ）にも適用されます。分類についてはまだ説明していませんので、分類前の初歩的なデータで作業します。

1. 新しいプロジェクトを作成し、 `roads` データセットだけを追加します。
2. `exercise_data/styles` にあるスタイルファイル `advanced_levels_demo.qml` をレイヤに適用します。これは、 **レイヤプロパティ** ダイアログの下部にある **スタイル　　スタイルを読み込む**... のコンボボックスから行うことができます。
3. Swellendam のエリアにズームします。
4. シンボルレイヤを使用して、レイヤの輪郭線は、以下の画像に従って互いに流入することを確認してください：



答え

1. シンボルレベルを次の値に調整します：



1. 異なる結果を得るために、異なる値で実験してください。
2. 次の課題に進む前に、元のマップをもう一度開いてください。

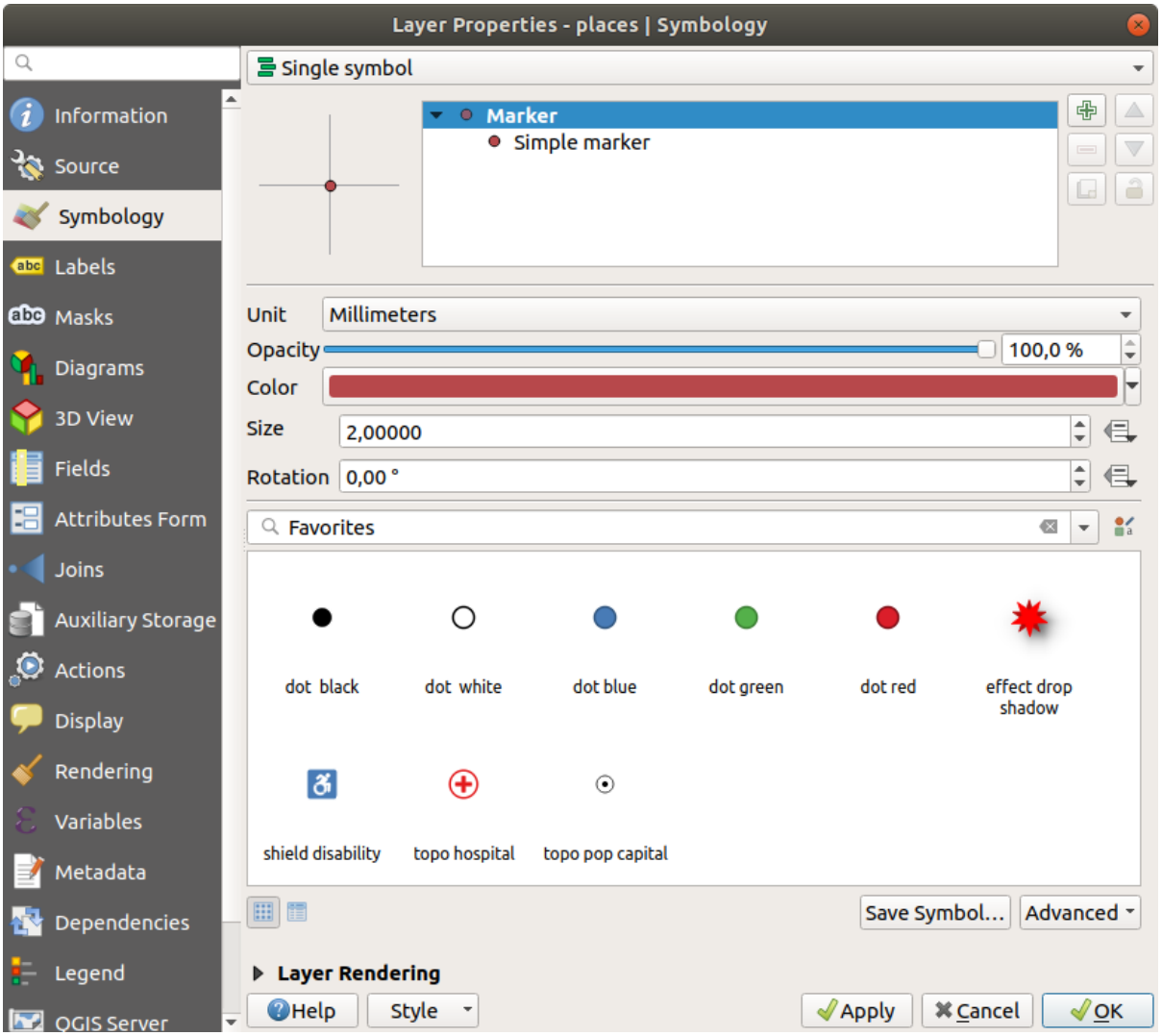
2.4.11 ★★☆ （中級レベル） 理解しよう: シンボルレイヤタイプ

塗りつぶしの色を設定したり、あらかじめ定義されたパターンを使用するだけでなく、完全に異なるシンボルレイヤタイプを使用することができます。今まで使っていたのは **シンプル塗りつぶし** タイプだけでした。より高度なシンボルレイヤタイプを使えば、シンボルをさらにカスタマイズすることができます。

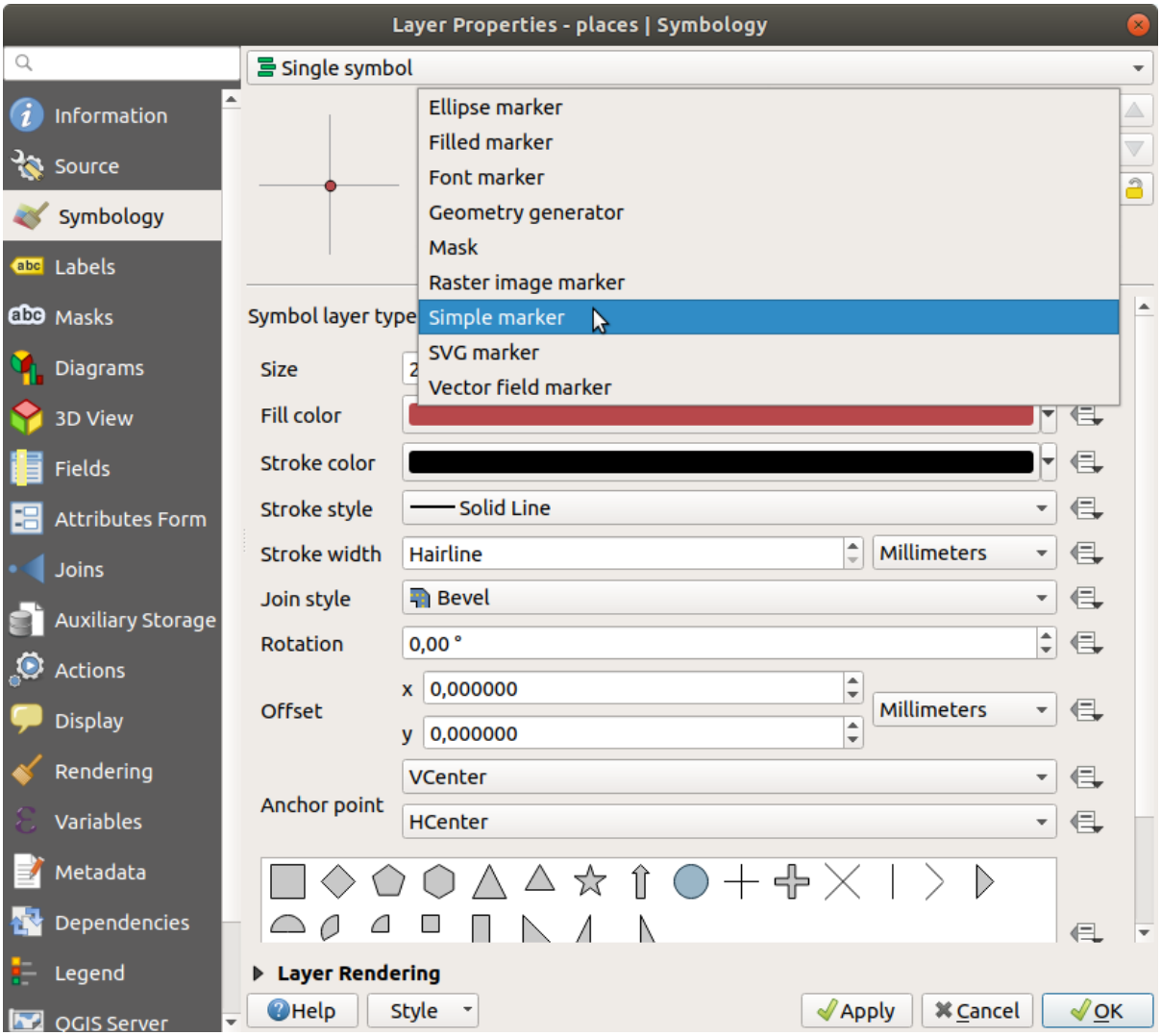
ベクタの各種類（ポイント、ライン、ポリゴン）は、シンボルレイヤタイプの独自のセットを有しています。まず、ポイントのために利用可能なタイプを見ていきます。

ポイントシンボルレイヤタイプ

1. `places` 以外のレイヤのチェックを外します。
2. `places` レイヤのシンボルプロパティを変更します。



3. シンボルレイヤツリーで **シンプルマーカー** レイヤを選択し、**シンボルレイヤタイプ** ドロップダウンをクリックすると、さまざまなシンボルレイヤタイプにアクセスできます。

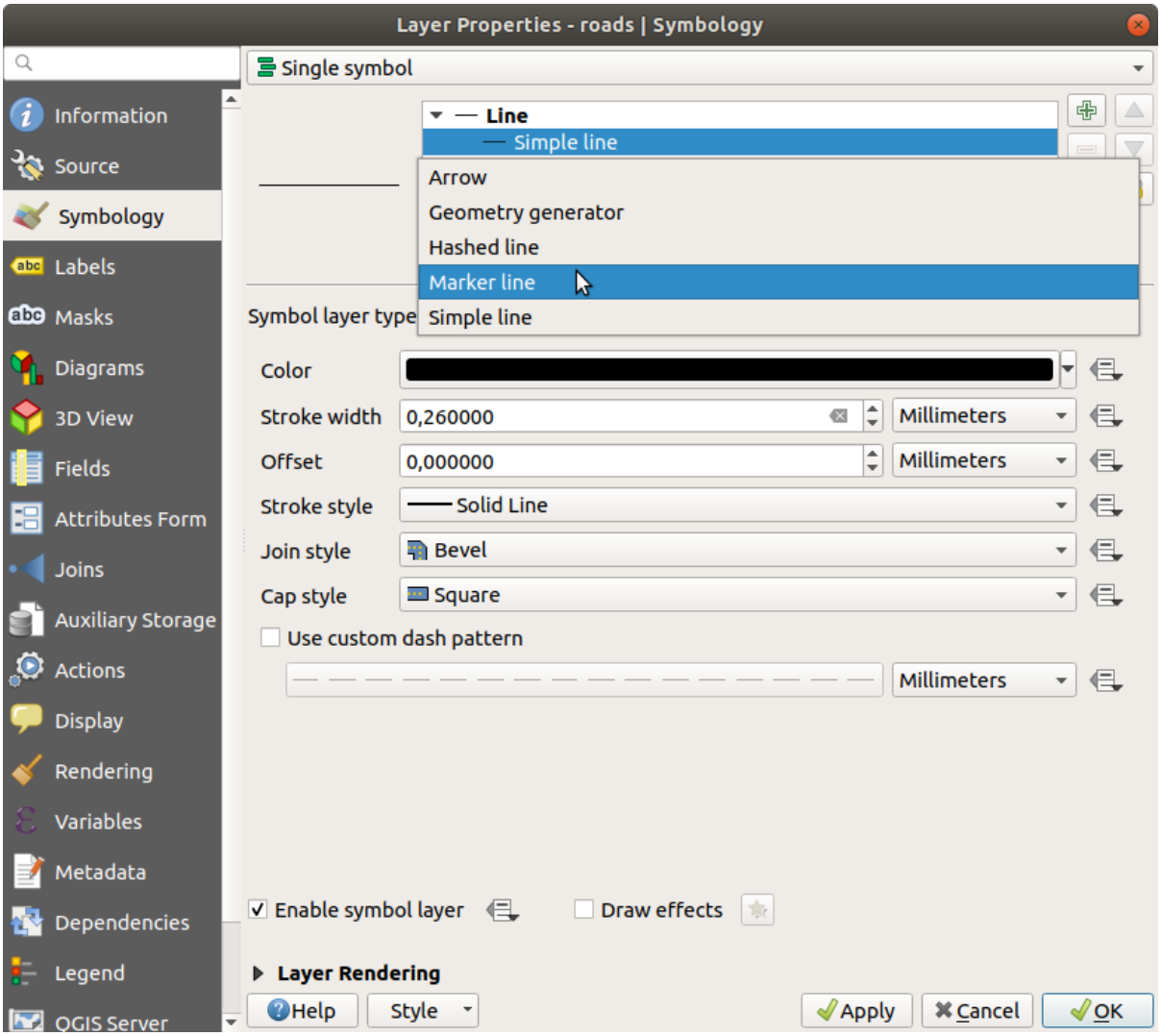


4. 利用できるさまざまなオプションを調べ、適切だと思ったスタイルのシンボルを選択してください。
5. 迷ったら、白の境界と淡いグリーンの塗りつぶし、**大きさ** `3.00`、 **ストローク幅** `0.5` の丸い **シンプルマーカー** を使いましょう。

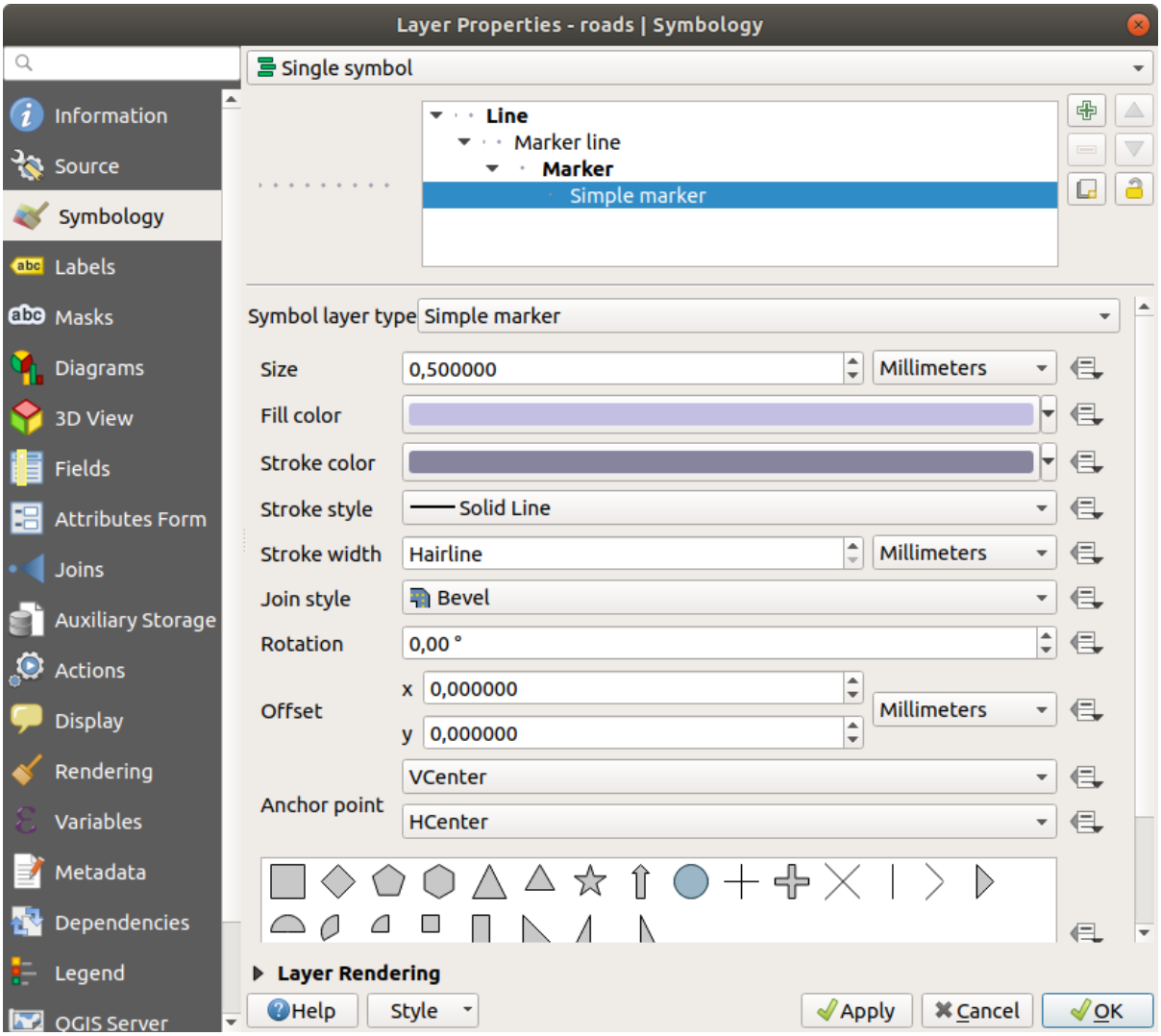
ラインシンボルレイヤ型

ラインデータのために利用できるさまざまなオプションを表示するには：

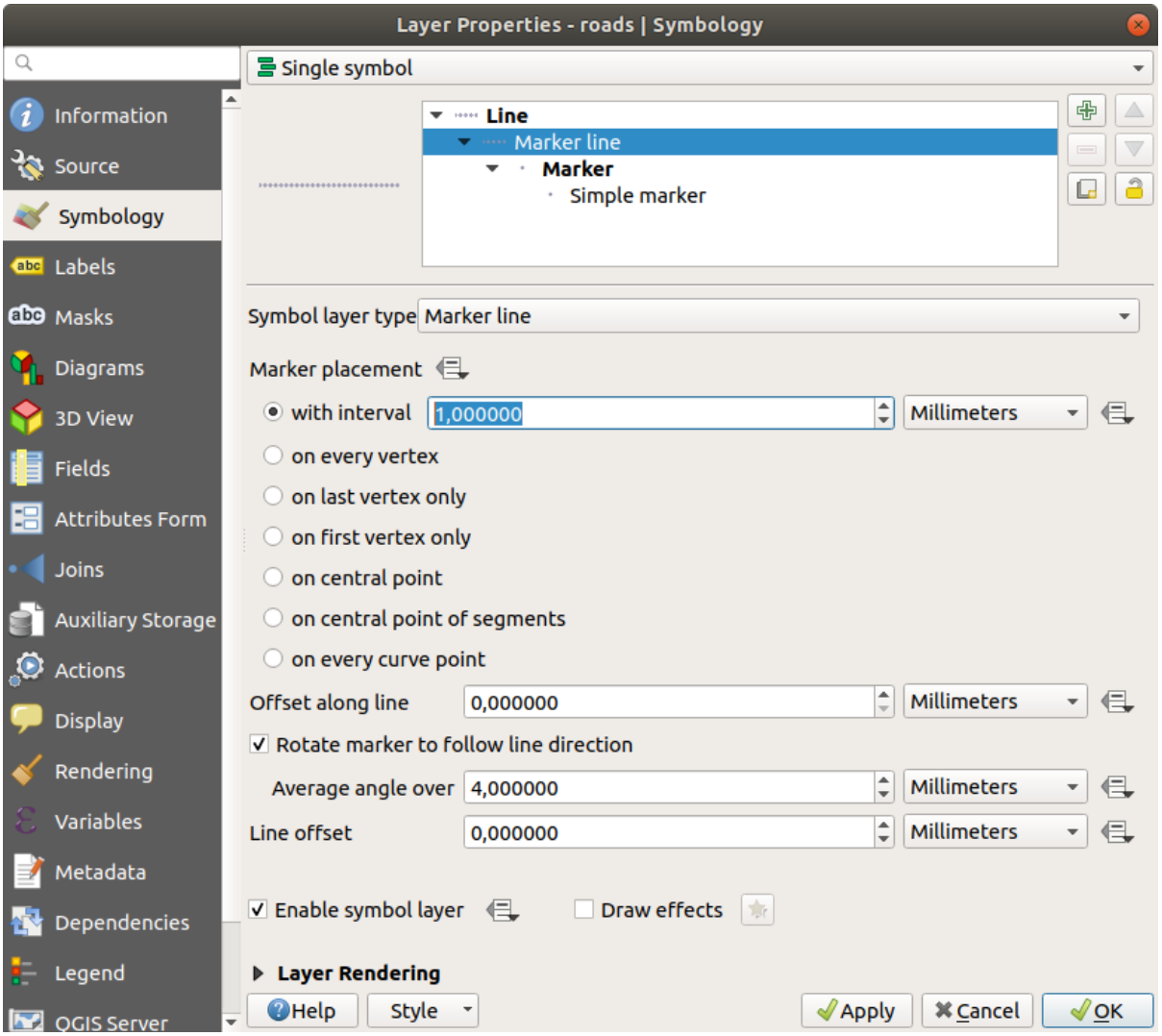
1. roads レイヤの最上段のシンボルレイヤの **シンボルレイヤ型** を **マーカー線** に変更します：



2. シンボルレイヤツリーで **シンプルマーカー** レイヤを選択します。このダイアログに合わせてシンボルのプロパティを変更します：



3. **マーカー線** レイヤを選択し、間隔を `1.00` に変更します：



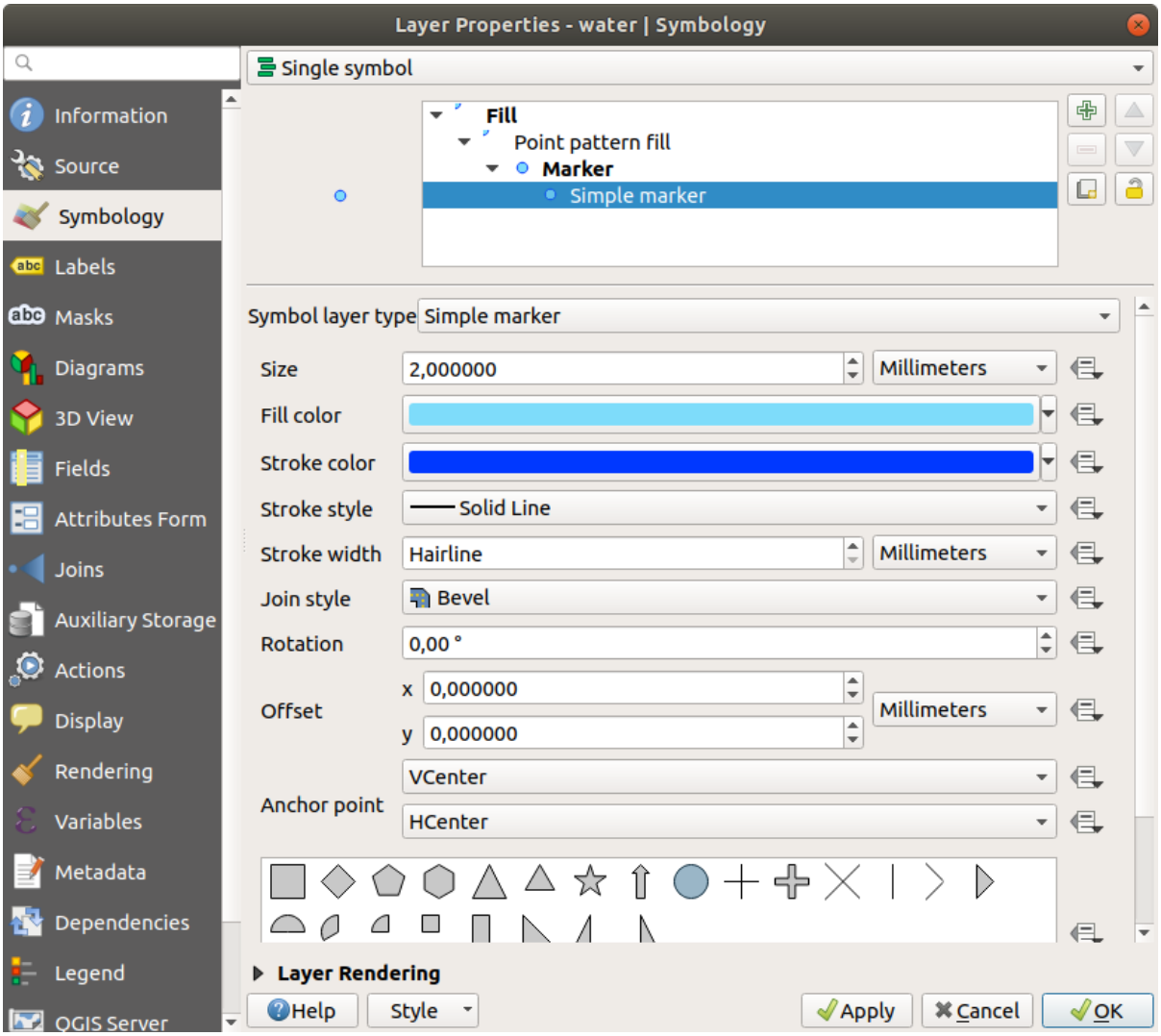
4. スタイルを適用する前に、シンボルレベルが正しいことを（以前に使用した **詳細設定** **描画順序** ダイアログ経由で）確認してください。

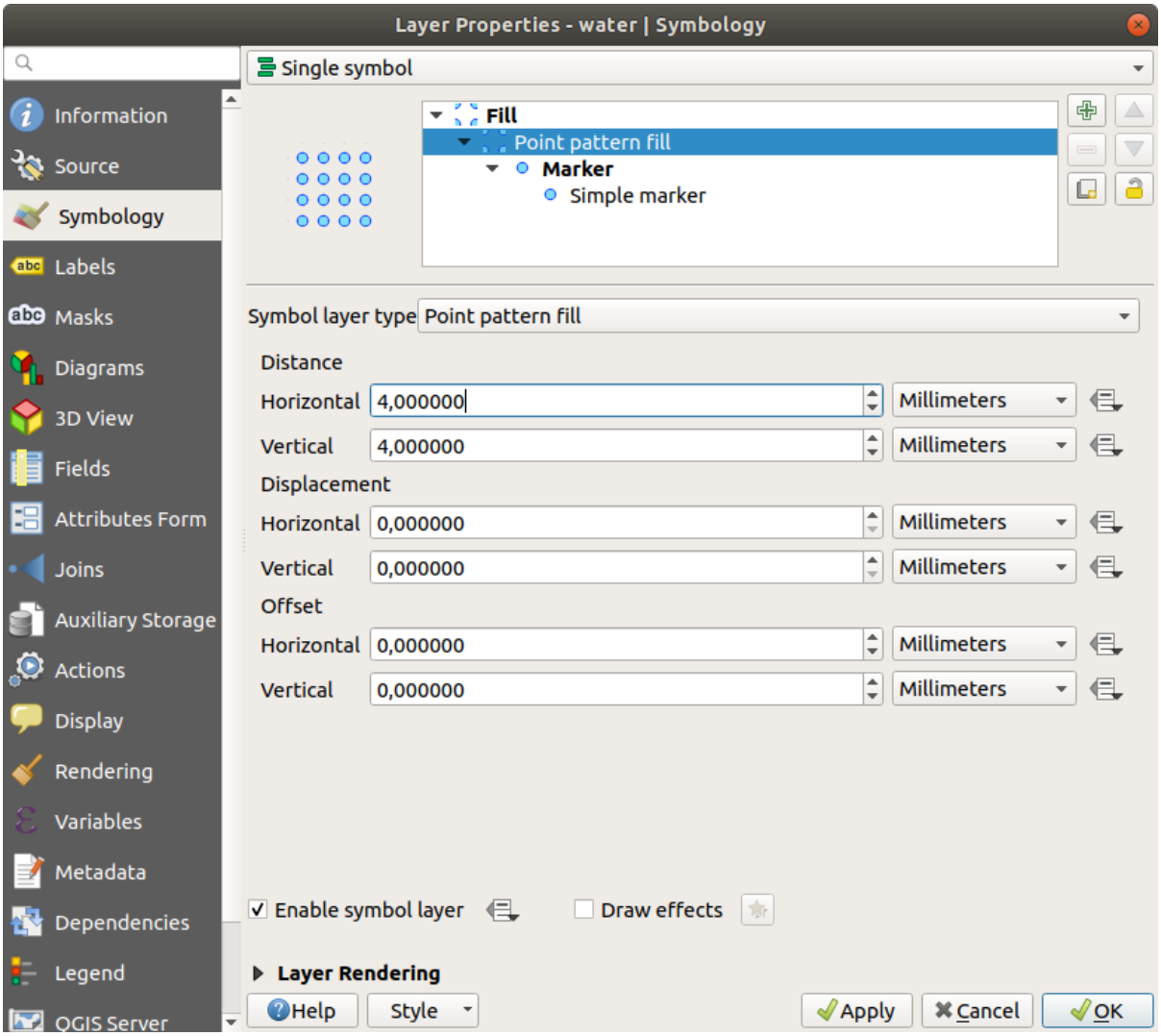
スタイルを適用したら、地図上でその結果を見てみましょう。おわかりのように、これらのシンボルは道路と一緒に向きを変えるが、常にそれに沿って曲げないでください。これは、いくつかの目的には有用ですが、他の目的には有用ではありません。ご希望の場合は、前にした方法に戻って、問題のシンボルレイヤーを変更できます。

ポリゴンシンボルレイヤタイプ

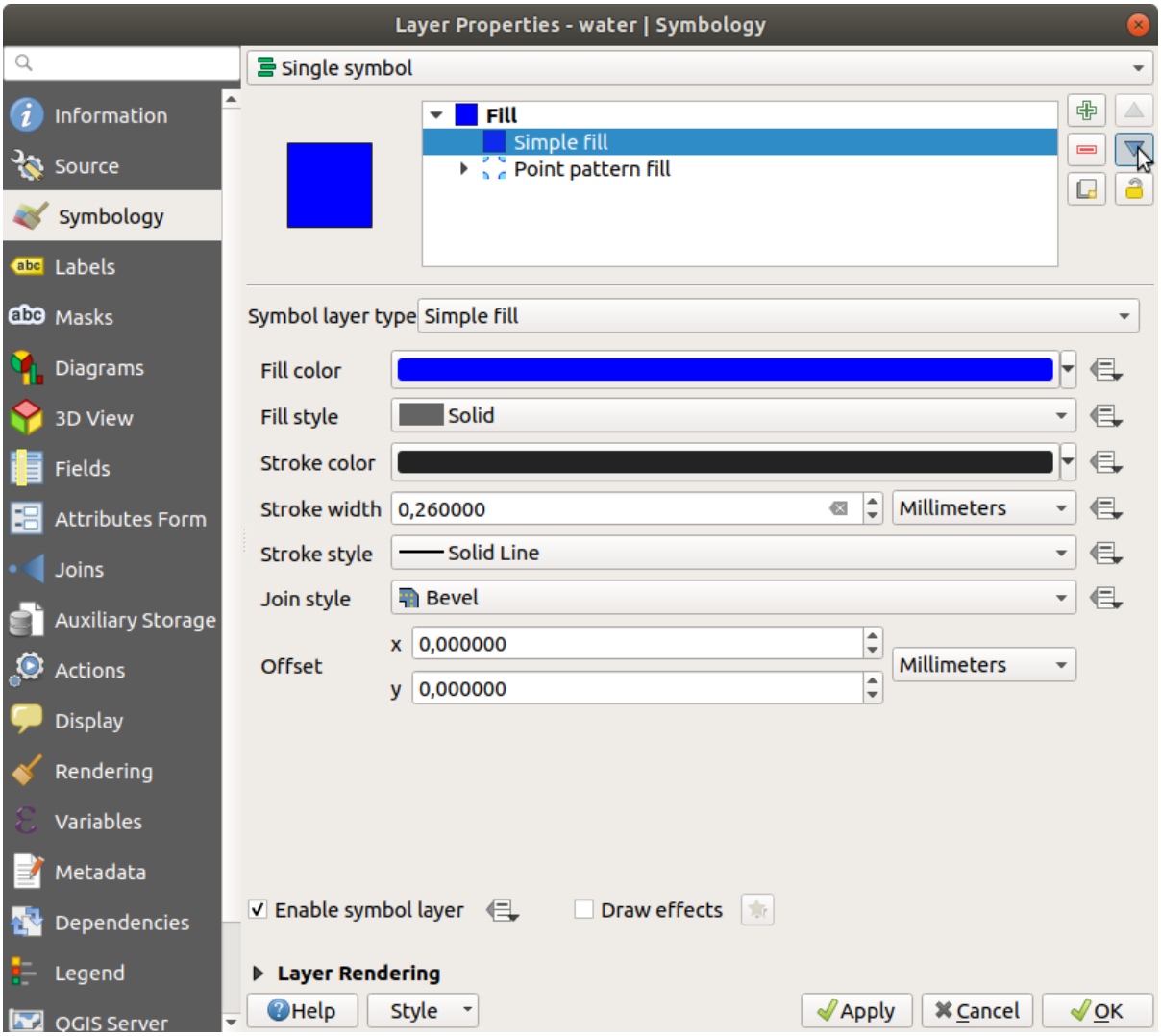
ポリゴンデータのために利用できるさまざまなオプションを表示するには：

1. 他のレイヤと同様に、`water` レイヤの **シンボルレイヤ型** を変更します。
2. リスト上の異なるオプションが何ができるかを調べます。
3. これらのうち、適した 1 つを選択します。
4. 迷ったときは、次のオプションを使用して、 **ポイントパターン塗りつぶし** を使用します。


Layer Properties - water | Symbology
Information
Source
Symbology
Labels
Masks
Diagrams
3D View
Fields
Attributes Form
Joins
Auxiliary Storage
Actions
Display
Rendering
Variables
Metadata
Dependencies
Legend
QGIS Server
Single symbol
Fill
Point pattern fill
Marker
Simple marker
Symbol layer type
Simple marker
Size
2,000000
Millimeters
Fill color
Stroke color
Stroke style
Solid Line
Stroke width
Hairline
Millimeters
Join style
Bevel
Rotation
0,00 °
Offset
x
0,000000
y
0,000000
Millimeters
Anchor point
VCenter
HCenter
Layer Rendering
Help
Style
Apply
Cancel
OK



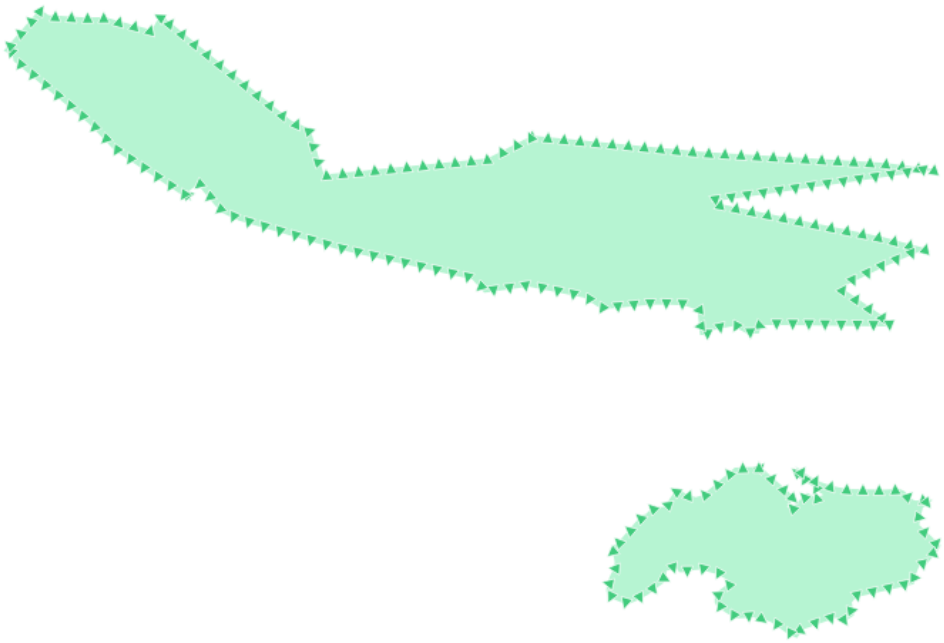
5. 通常の **シンプル塗りつぶし** で新しいシンボルレイヤを追加します。
6. それを暗青の境界を持つ水色にします。
7. **下に移動** ボタンでポイントパターンシンボルレイヤの下に移動させます。



その結果、テクスチャを構成する個々のドットの大きさ、形状や距離を変えることができるという利点があり、水のレイヤのためのテクスチャシンボルを持っています。

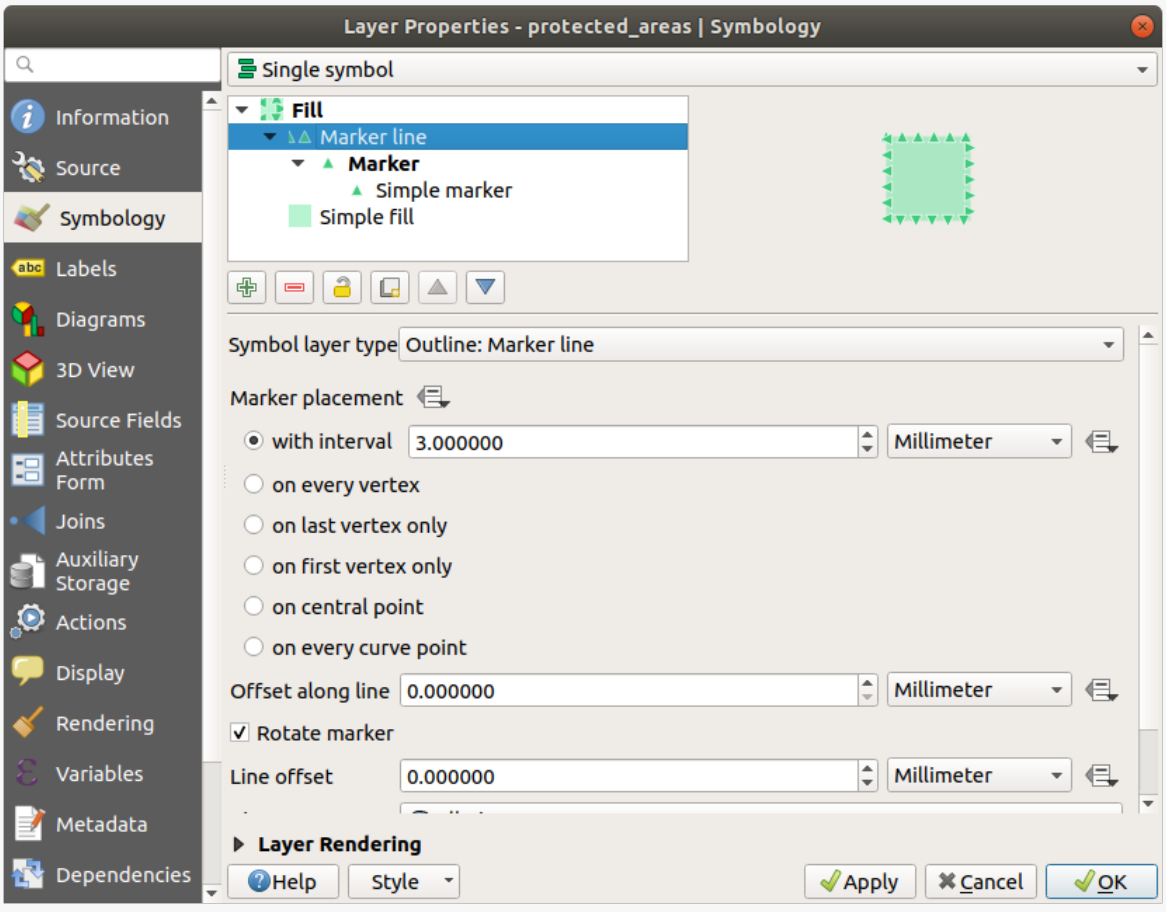
2.4.12 ★★☆ （中級レベル） 自分でやってみよう:

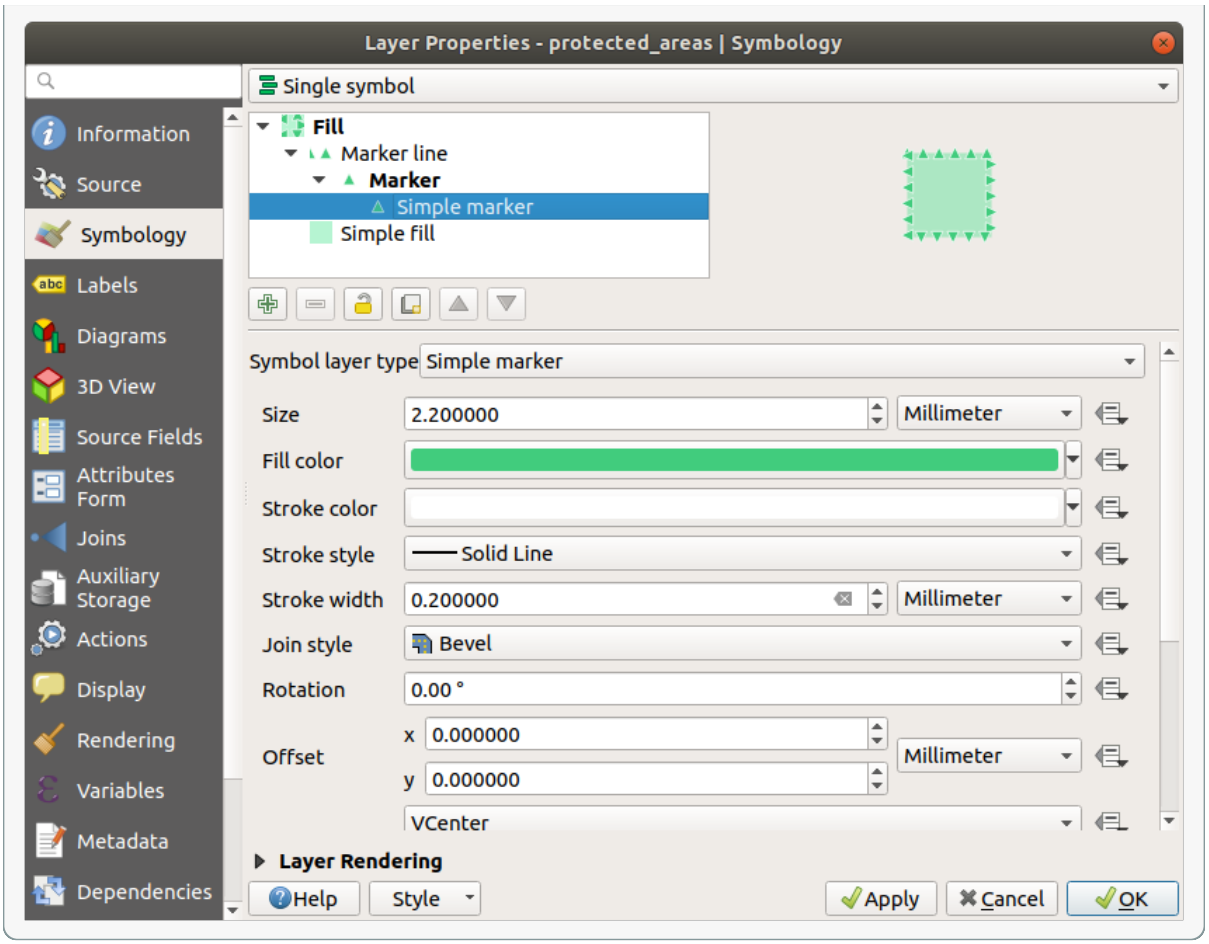
`protected_areas` レイヤに透き通った緑の塗りつぶし色を適用し、輪郭を以下のように変更します:



答え

以下は、シンボル構造の例です：





2.4.13 ★★★ （上級レベル） 理解しよう: ジオメトリジェネレータシンボロジ

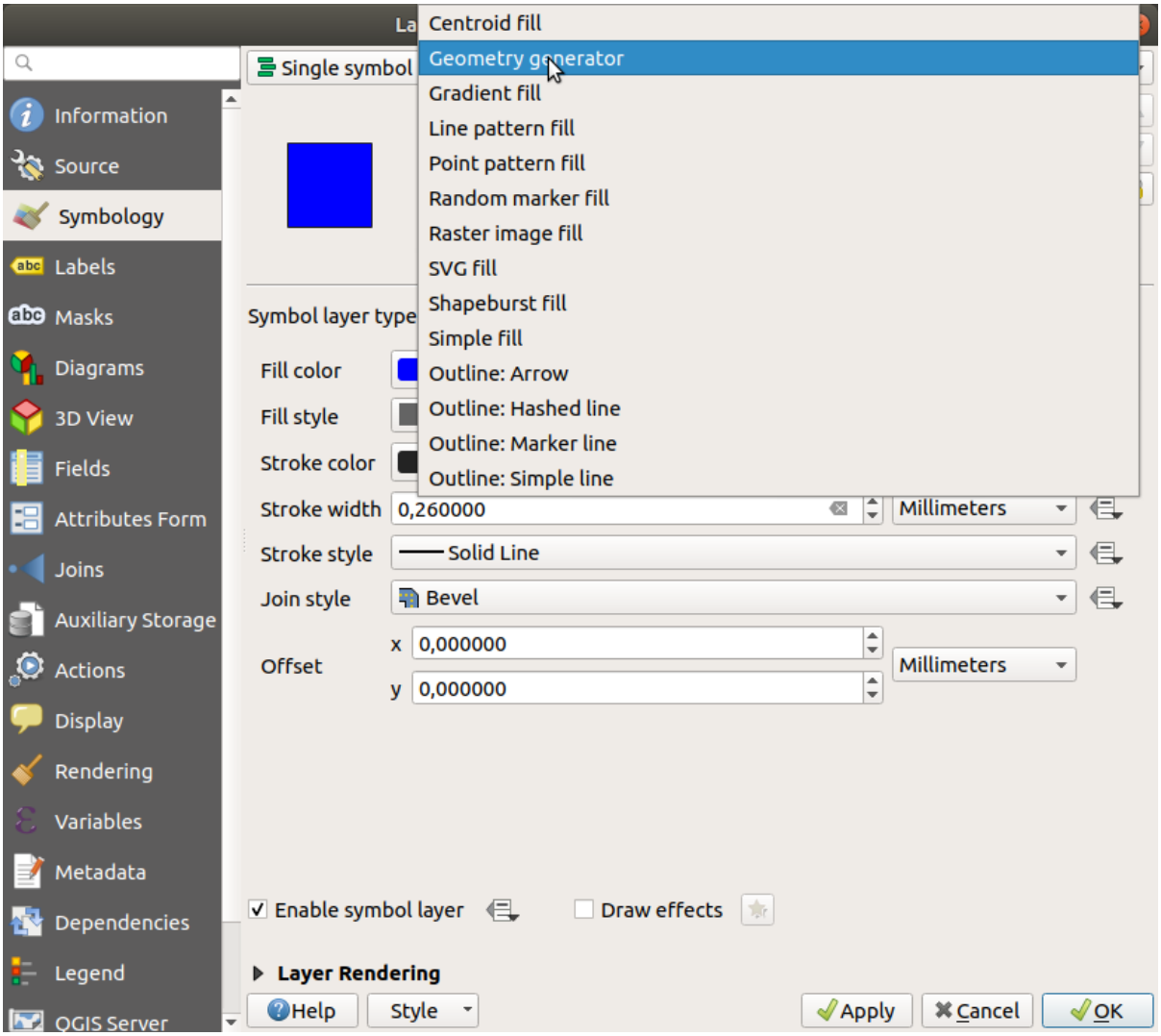
ジオメトリジェネレータシンボロジは、すべてのレイヤタイプ（ポイント、ライン、ポリゴン）で使用することができます。生成されるシンボルは、レイヤタイプに直接依存します。

非常に簡単に説明すると、ジオメトリジェネレータシンボロジでは、シンボロジ自体の中でいくつかの空間演算を実行することができます。例えば、ポイントレイヤを作成せずに、ポリゴンレイヤ上で重心の空間演算を実行することができます。

さらに、結果として得られるシンボルの外観を変更するためのすべてのスタイリングオプションがあります。

試してみましょう！

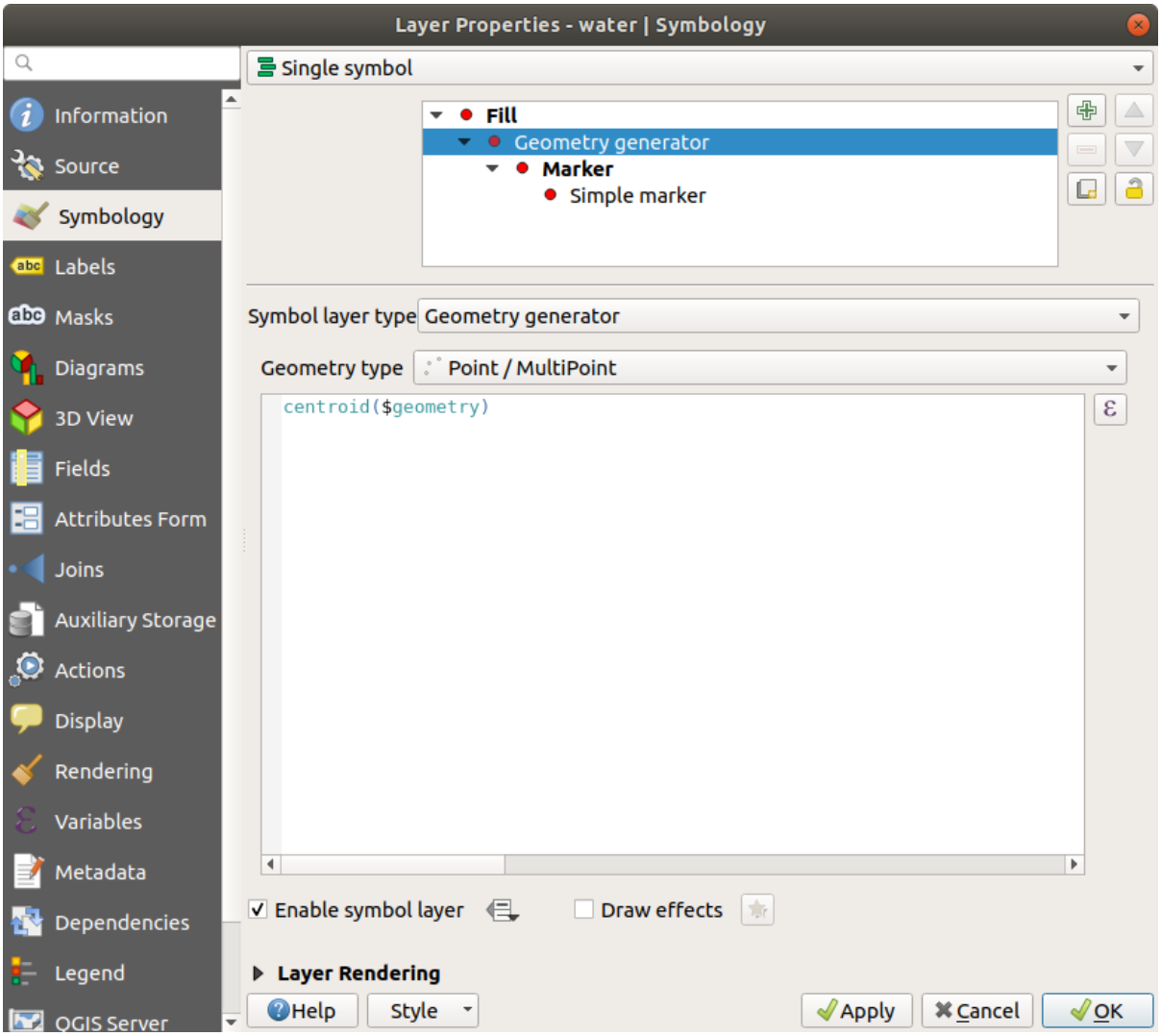
1. `water` レイヤを選択します。
2. **シンプル塗りつぶし** をクリックし、**シンボルレイヤタイプ** を **ジオメトリジェネレータ** に変更します。



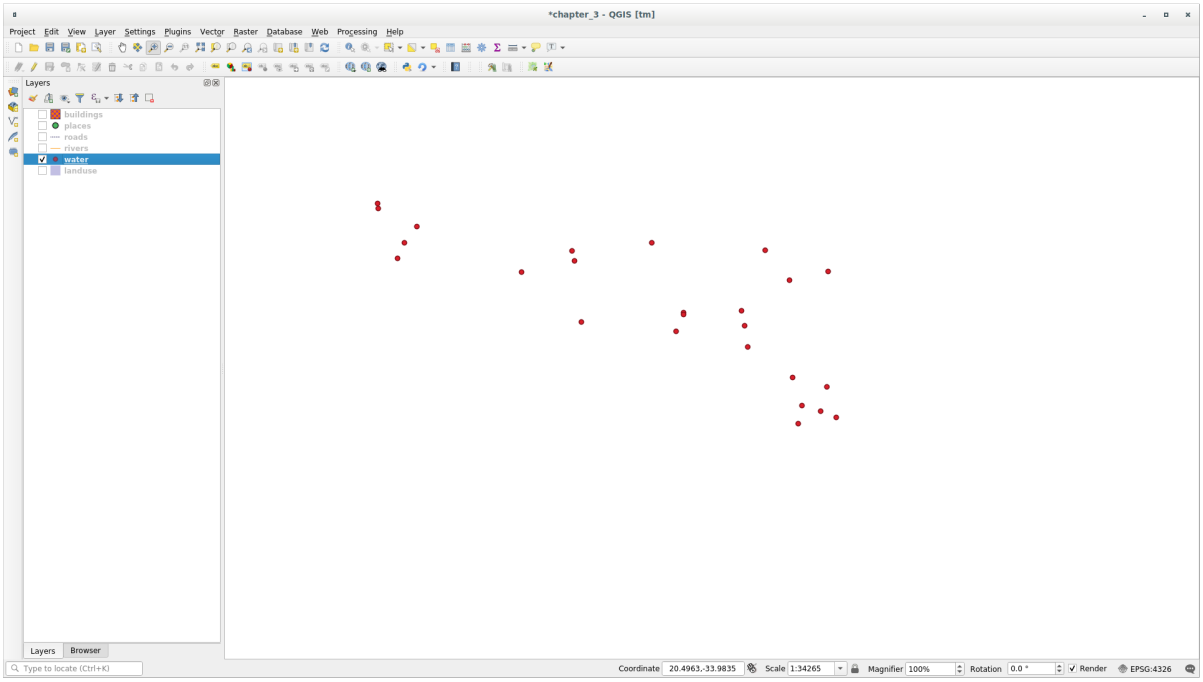
3. 空間クエリを書き始める前に、出力のジオメトリタイプを選択する必要があります。この例では、各地物に重心を作成するため、ジオメトリタイプを **ポイント／マルチポイント** に変更します。

4. クエリパネルにクエリを書いてみましょう：：

```
centroid($geometry)
```



5. *OK* をクリックすると、`water` レイヤがポイントレイヤとしてレンダリングされるのがわかります！レイヤシンボロジの中で空間演算を実行したことになるのです、すごいでしょう？



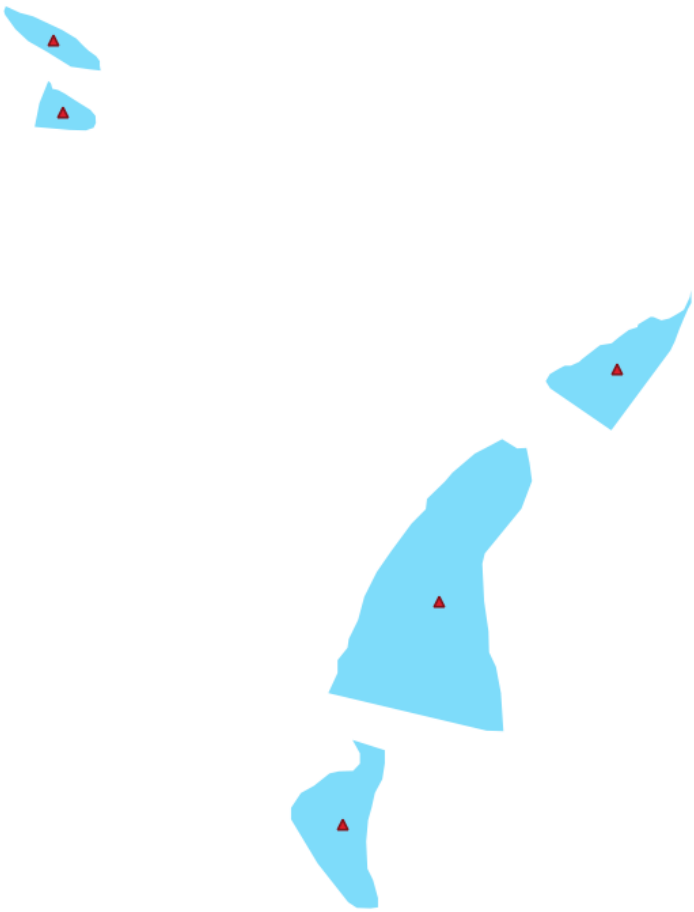
ジオメトリジェネレータシンボロジでは、 **通常** のシンボロジとは一線を画した表現が可能です。

★★★ （上級レベル） 自分でやってみよう:

ジオメトリジェネレータは、単なるシンボルレベルの一つです。 **ジオメトリジェネレータ** の下に、別の **シンプル塗りつぶし** を追加してみてください。

ジオメトリジェネレータシンボロジのシンプルマーカーの外観も変えてみてください。

最終的にはこのようになるはずです：



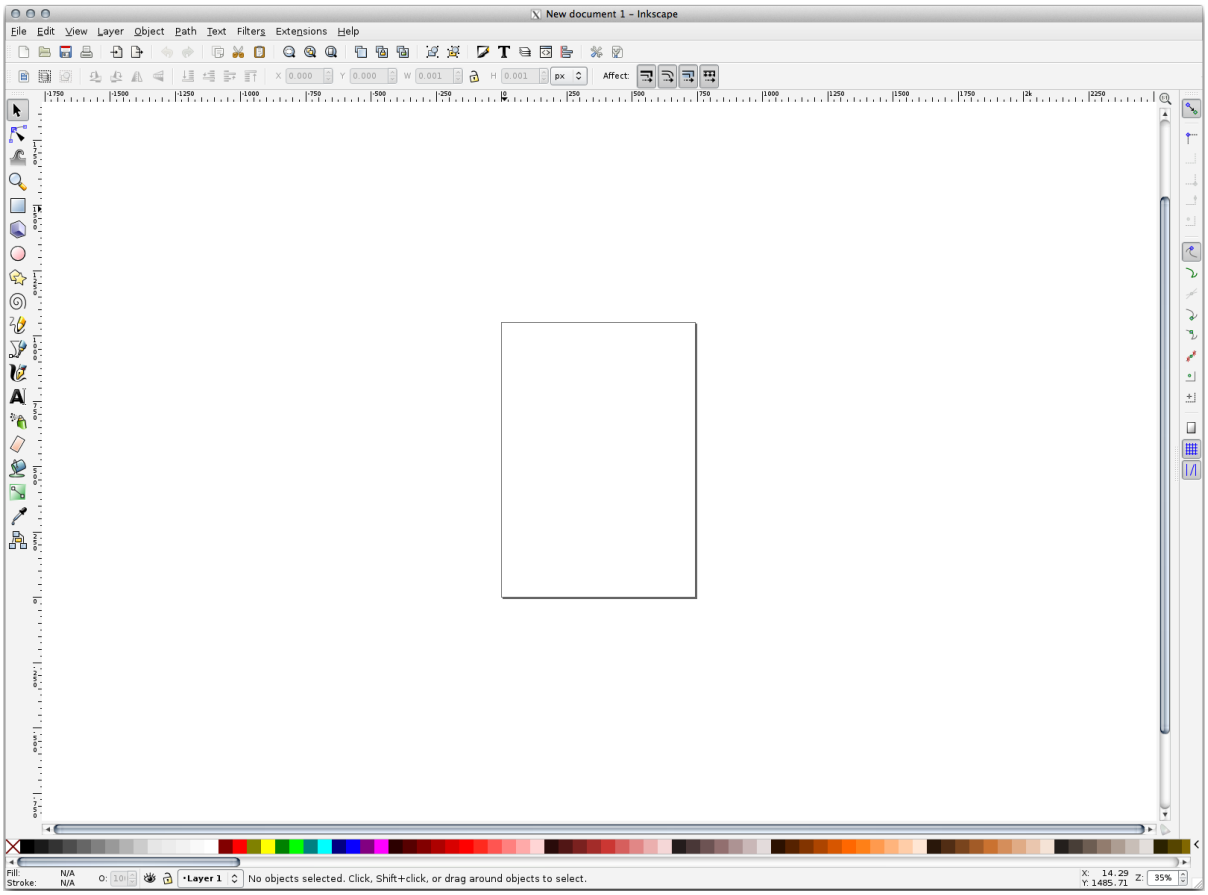
答え

1. 別のシンボルレベルを追加するには、 ボタンをクリックします。
2. ボタンをクリックして、リストの一番下に新しいシンボルを移動します。
3. 水のポリゴンを塗りつぶすために良い色を選択します。
4. ジオメトリジェネレータシンボロジの **マーカー** をクリックし、好きなように円を別の形状に変更します。
5. より有用な結果を得るために、他のオプションも試してみてください。

2.4.14 ★★★ （上級レベル） 理解しよう: カスタム SVG 塗りつぶしを作る

> **注釈**
>
> この演習を行うには、無料のベクタ編集ソフトウェア Inkscape がインストールされている必要があります。

1. Inkscape プログラムを起動します。以下のようなインタフェースが表示されます：

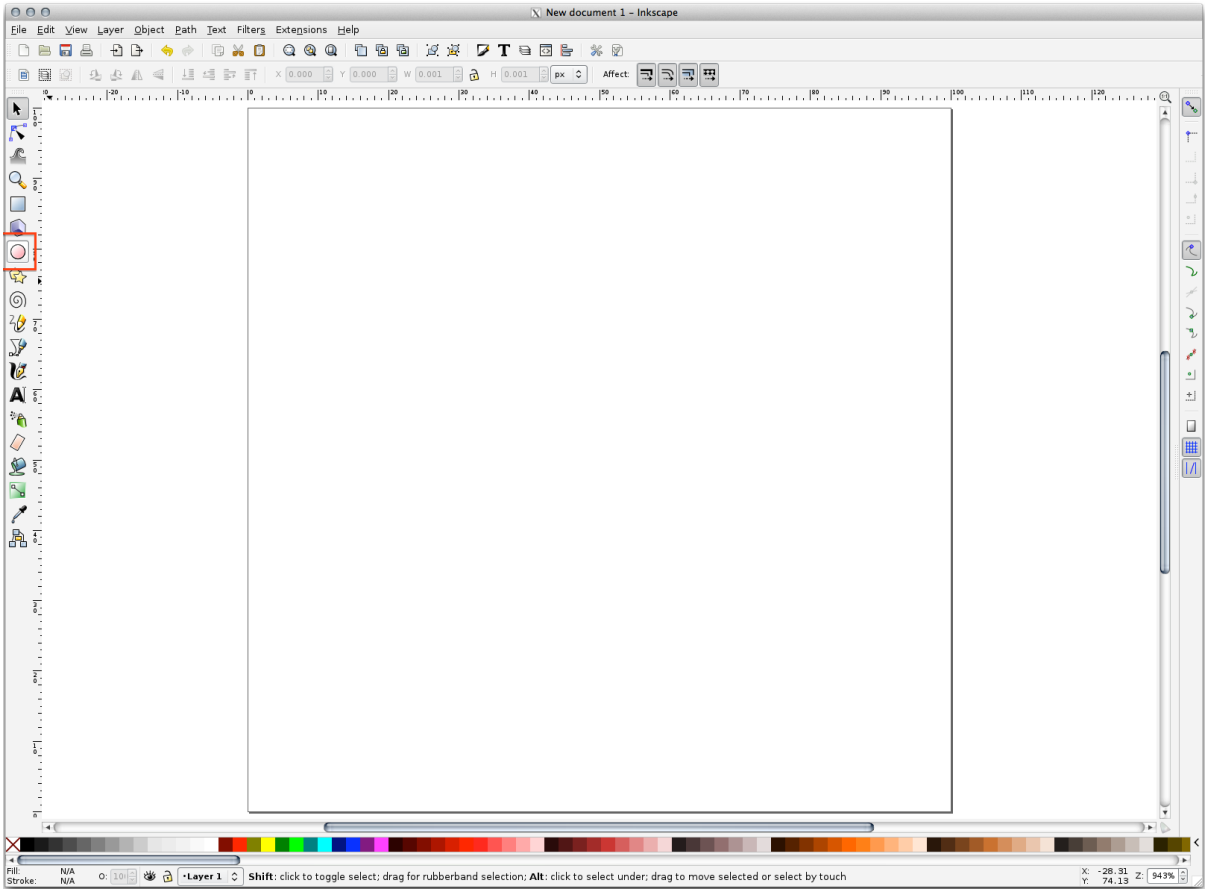


   コーレルのような、他のベクタ画像編集プログラムを使ったことがある方は、これを親しみ易いと思うはずです。

   まず、小さなテクスチャに適したサイズにキャンバスを変更します。

2. **ファイル** **文書のプロパティ** メニューをクリックします。 **文書のプロパティ** ダイアログが表示されます。
3. **単位** を *px* に変更します。
4. **幅** と **高さ** を `100` に変更します。
5. 実行後はダイアログを閉じます。
6. **ビュー** **ズーム** **ページ** メニューをクリックして、作業しているページを参照してください。

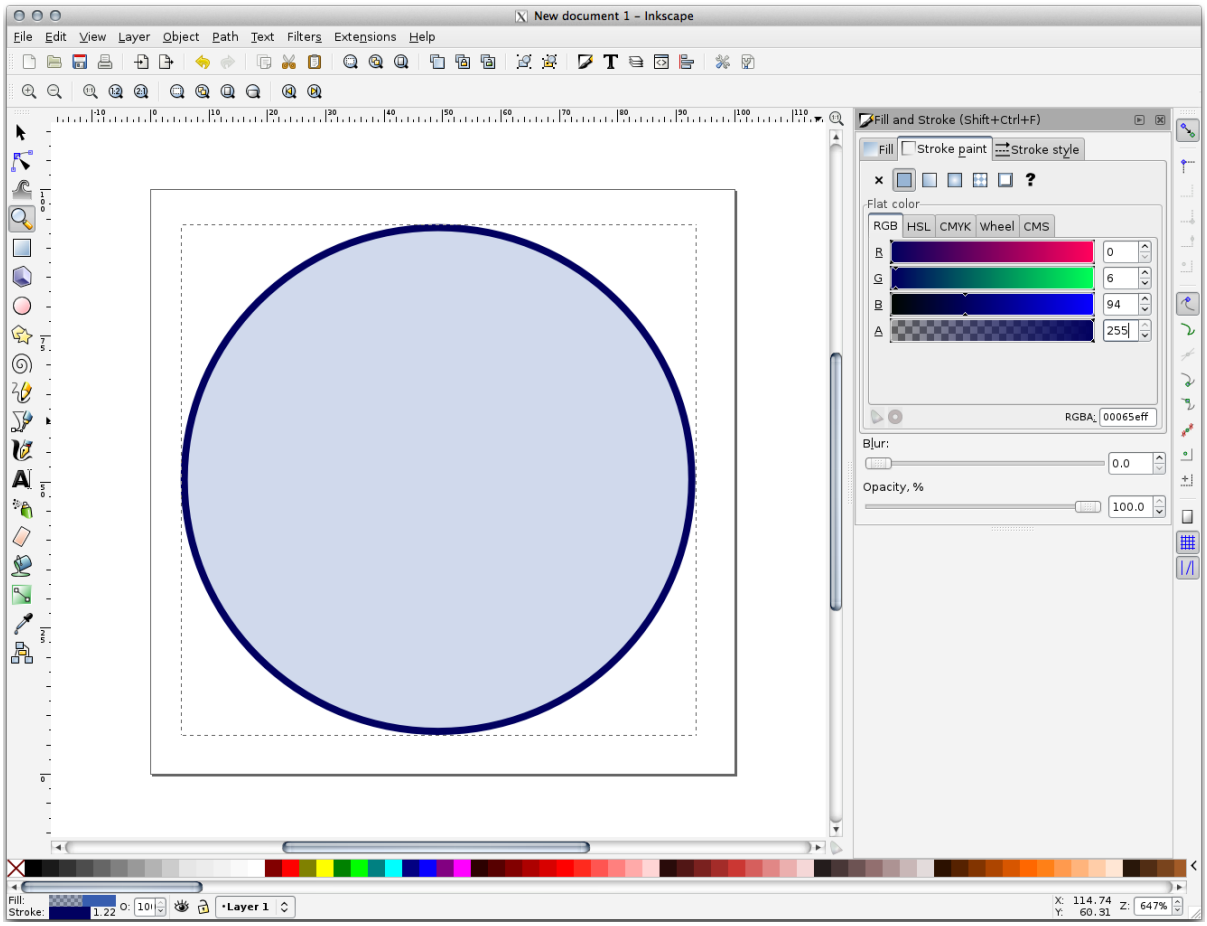
7. **円** ツールを選択します。



8. 楕円を描画するためにクリックして、ページ上でドラッグします。楕円が円に変わるようにするには、描いている間 `Ctrl` ボタンを押し続けます。

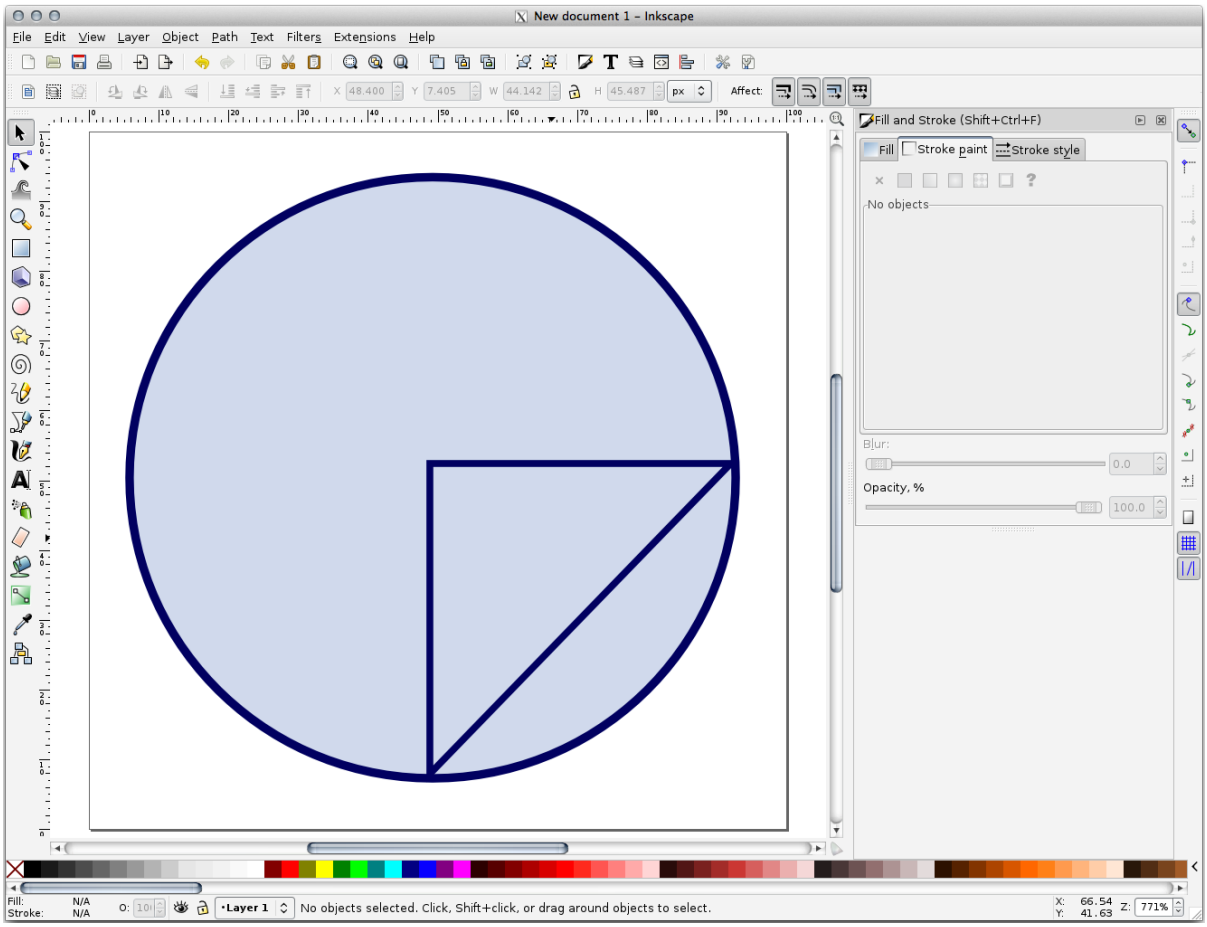
9. 先ほど作成した円を右クリックし、その **塗りとストローク** オプションを開きます。このように、そのレンダリングを変更できます：

    1. **塗り** 色を何とか淡灰青に変更し、
    2. **ストロークの塗り** タブの中で境界に暗い色を割り当て、
    3. **ストロークのスタイル** タブの下で境界の太さを減らします。



10. **鉛筆** ツール使ってラインを引きます：

    1. 一回クリックするとラインが始まります。 `Ctrl` キーを押しながらだと、15 度単位でスナップします。
    2. 水平方向にポインタを移動し、クリックしてポイントを置きます。
    3. クリックして、ラインの頂点にスナップし、垂直線をトレースしてクリックで終わります。
    4. 二つの端の頂点を繋げます。
    5. このようなシンボルで終わるように、円のストロークに合わせ、必要に応じて、それを周りに移動するには三角形のシンボルの色と幅を変更します。

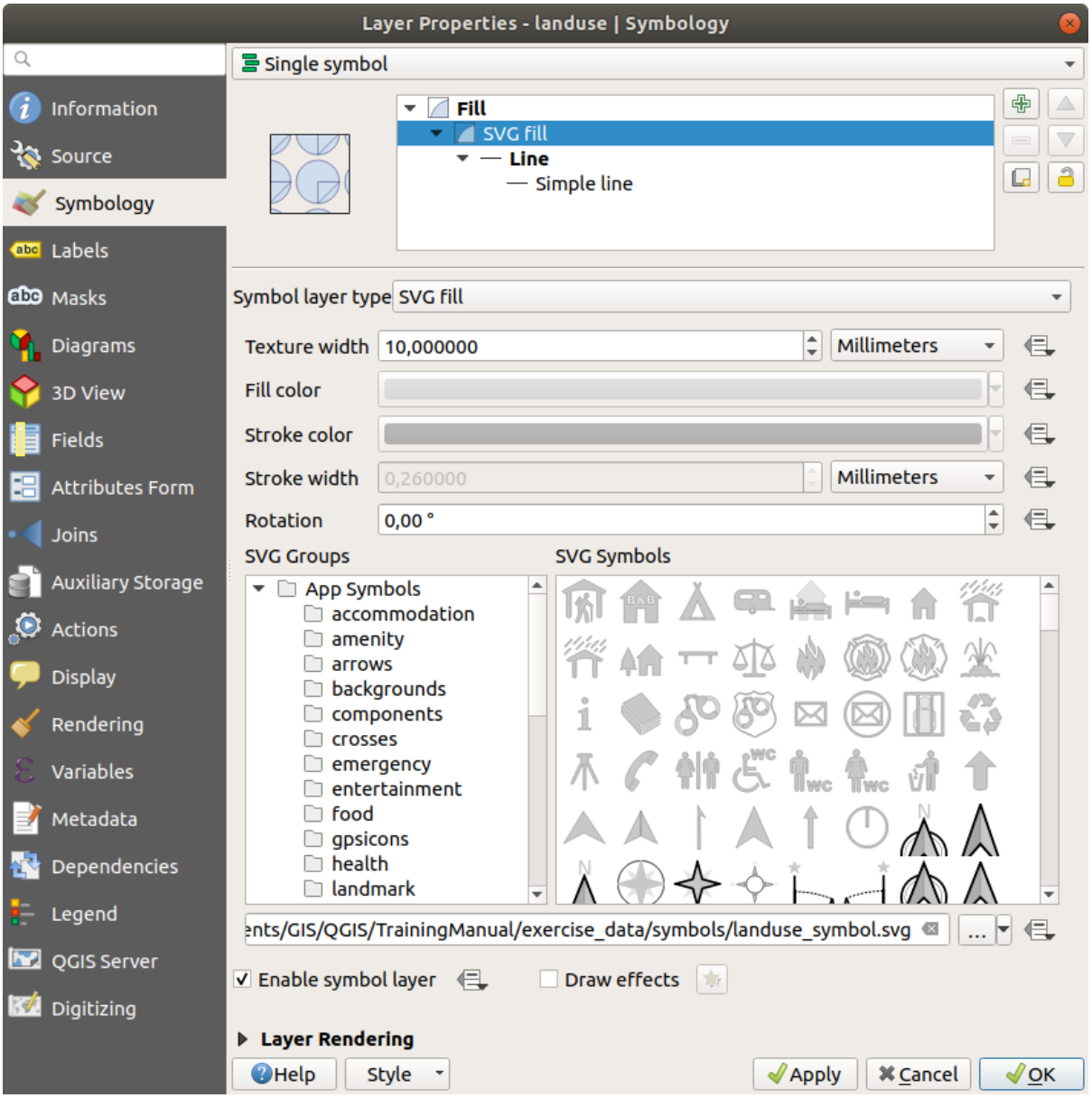


11. 満足のゆくシンボルが得られたら、コースのあるディレクトリの下の `exercise_data/symbols` に `landuse_symbol` として、SVG ファイルで保存してください。

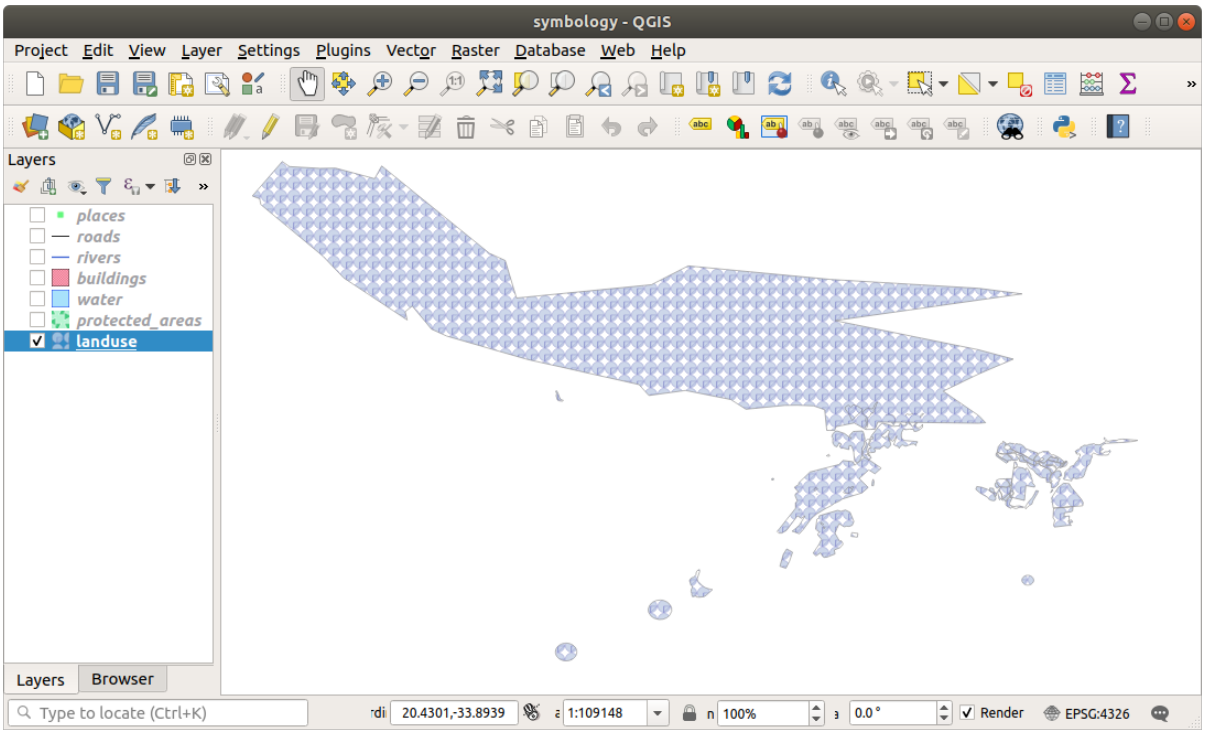
QGIS では:

1. `landuse` レイヤの **レイヤプロパティ** を開きます。
2. **シンボロジ** タブで、 **シンボルレイヤ型** を以下に示す *SVG* **塗りつぶし** に変更して、シンボル構造を変更します。
3. ... ボタンをクリックし、 **ファイルを選択**... をクリックして、SVG 画像を選択します。

   シンボルツリーに追加され、さまざまな特性（色、角度、効果、単位...）をカスタマイズできるようになりました。



ダイアログを有効にすると、 landuse レイヤの地物がシンボルで覆われ、次のマップのようなテクスチャが表示されるはずです。テクスチャが見えない場合は、マップキャンバスを拡大するか、レイヤのプロパティで **テクスチャ幅** を大きく設定する必要があるかもしれません。



2.4.15 ★★☆ （中級レベル） 理解しよう: マスキング

マスキング機能により、あるシンボルレイヤを別のシンボルレイヤまたはラベルでマスクすることができます。

次の演習では、ポリゴンのラベルがラインレイヤの黒い部分をマスクするようにレンダリングを設定します。

1. ディレクトリ `exercise_data/masking` 内のデータセットから `selective_masking.qgs` を読み込みます。
2. `polys` レイヤの **レイヤプロパティ** ウィンドウを開きます。

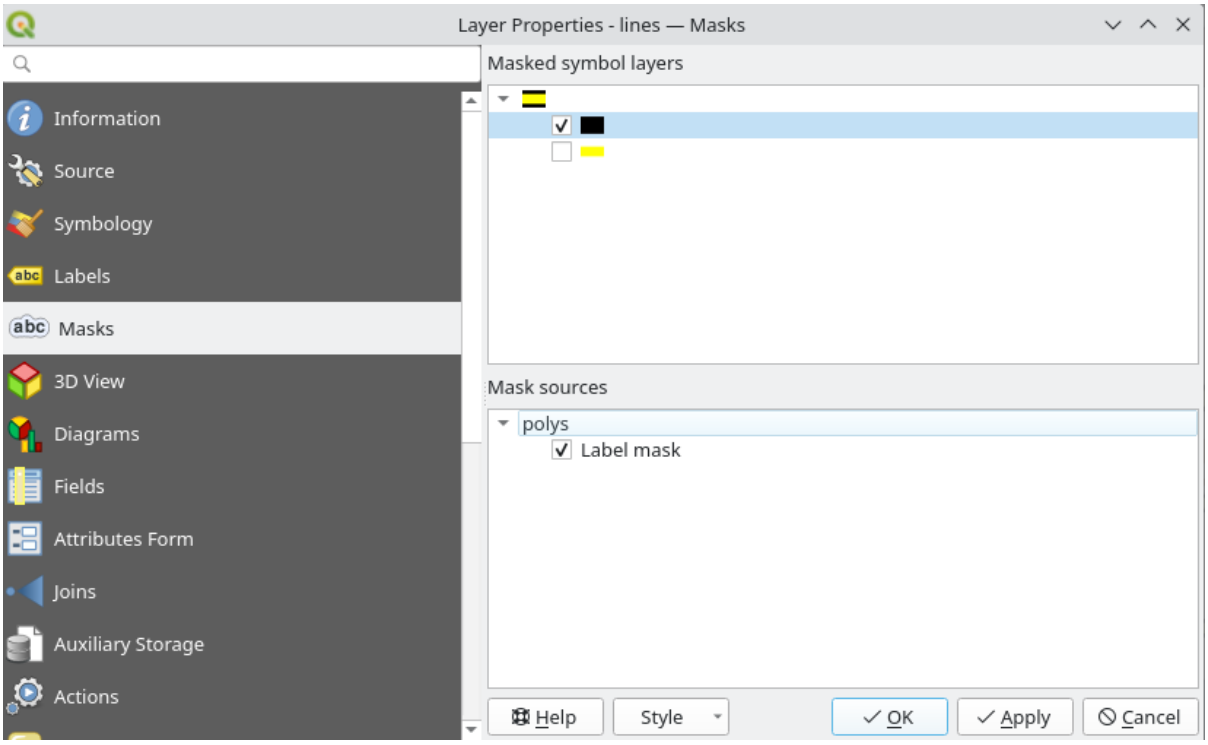
   ラベル タブで、**単一定義** がすでに定義されていることが分かります。
3. **ラベル** タブから、**マスク** を選択します（一般カテゴリリストの **マスク** ではありません）。
4. **マスクを有効化** オプションをチェックして、ラベルのマスクを有効にします。

   現在定義されたラベルマスクの形状は、他のシンボルレイヤをマスクするマスクソースとして後で選択できるというメッセージがあります。それをしてみましょう。
5. *OK* をクリックします
6. `lines` レイヤの **レイヤプロパティ** ウィンドウを開きます。
7. **マスク** タブに移動します
8. マスクするシンボルレイヤから、線の黒線部分のチェックボックスをクリックします。

   マスク設定を保存するには、マスクレイヤとマスクするシンボルレイヤの両方を選択しなければならないという警告メッセージが表示されます。再びそれをしてみましょう。

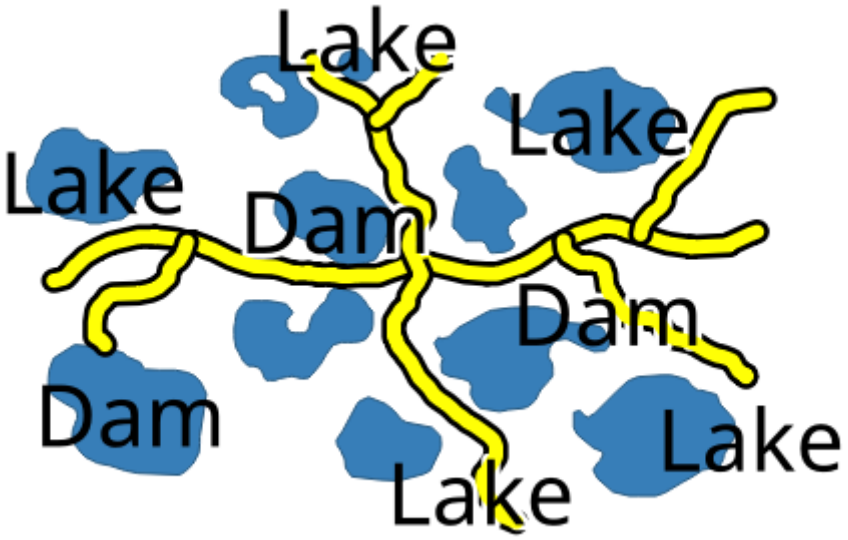
9. polys の **ラベルマスク** チェックボックスをクリックします

   先程のメッセージが消え、設定を安全に保存できます。



10. *OK* をクリックします

するとラベルが線の黒い部分をマスクしていて、黄色の部分だけが残っているのが分かると思います。



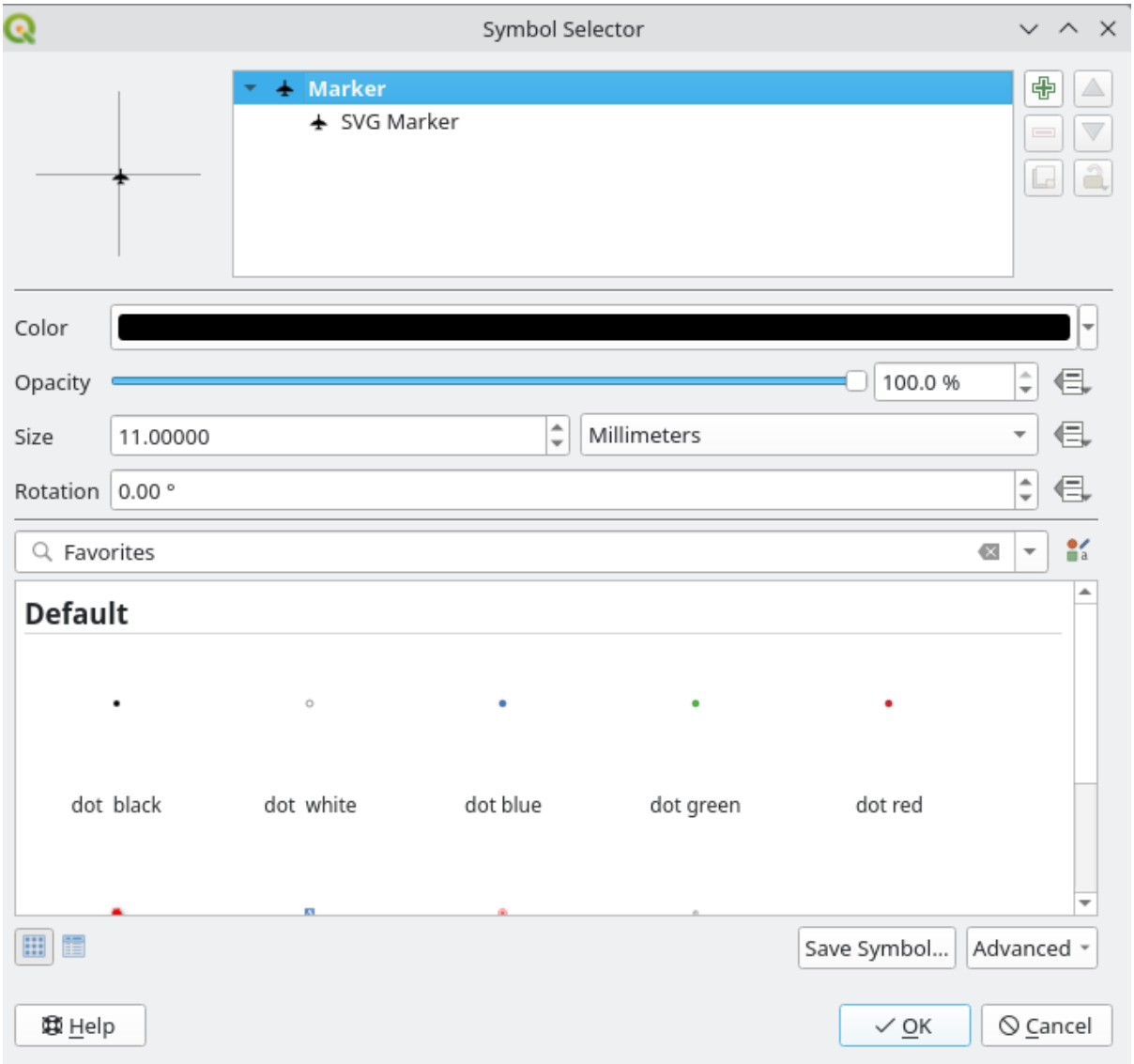
2.4.16 ★★☆ （中級レベル） 理解しよう: ポイントのシンボルを使ってマスクする

今度は、線の黒い部分を ポイント レイヤ（すなわち `B52` と `Jet` のポイントカテゴリ）を表す黒い飛行機だけを使ってマスクしたいと思います。見えるように `polys` レイヤの表示をオフにして構いません。

1. `points` レイヤの **レイヤプロパティ** ウィンドウを開きます。

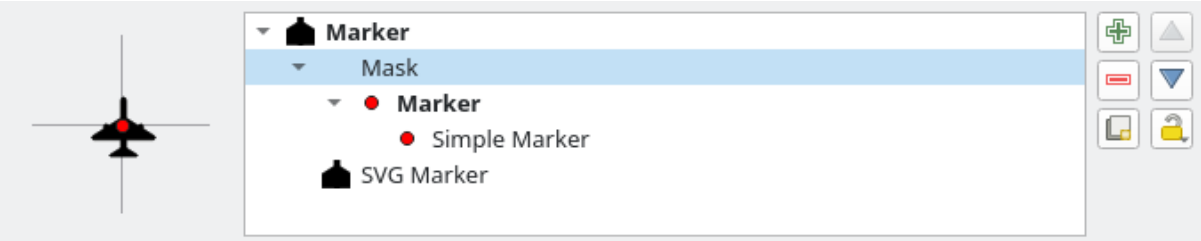
   シンボロジ タブで、**カテゴリ値による定義** レンダラが `Class` フィールドの値に応じて 3 つの異なるシンボルで定義されていることが分かります。

2. **シンボル** 列の中の `B52` 飛行機アイコンをダブルクリックします



3. シンボルレイヤを追加 ボタンをクリックし、マスク **シンボルレイヤ型** を選択します

   ルートシンボル マーカー の下に、新しい階層 マスク （**シンボルレイヤ**） > マーカー （**シンボル**） > シンプルマーカー （**シンボルレイヤ**）が現れます。



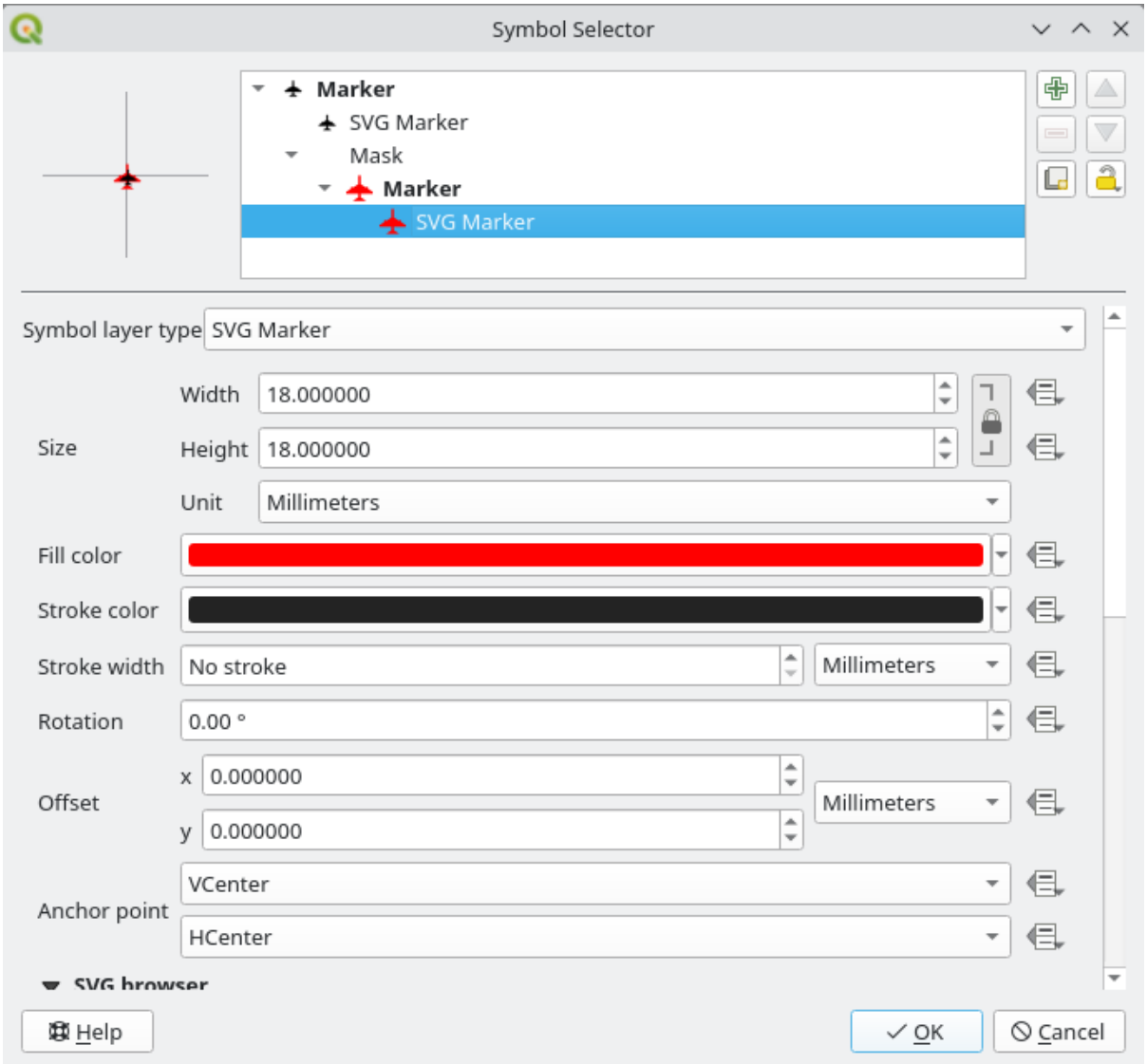
次に、飛行機マーカーと同じで少し大きなマスクを定義しましょう。

4. 新しく作成された シンプルマーカー シンボルレイヤを選択し、その **シンボルレイヤ型** を SVG マーカー に変更します

5. SVG ファイルのパスを既存の SVG マーカーと同じにし、サイズを大きくします（このユースケースでは 24 ミリメートルが適当でしょう）。

   マスクの色はここでは重要ではなく、マスクには形状のみが使用されます。この状況では、シンボルのレイヤの順序も関係ありません。

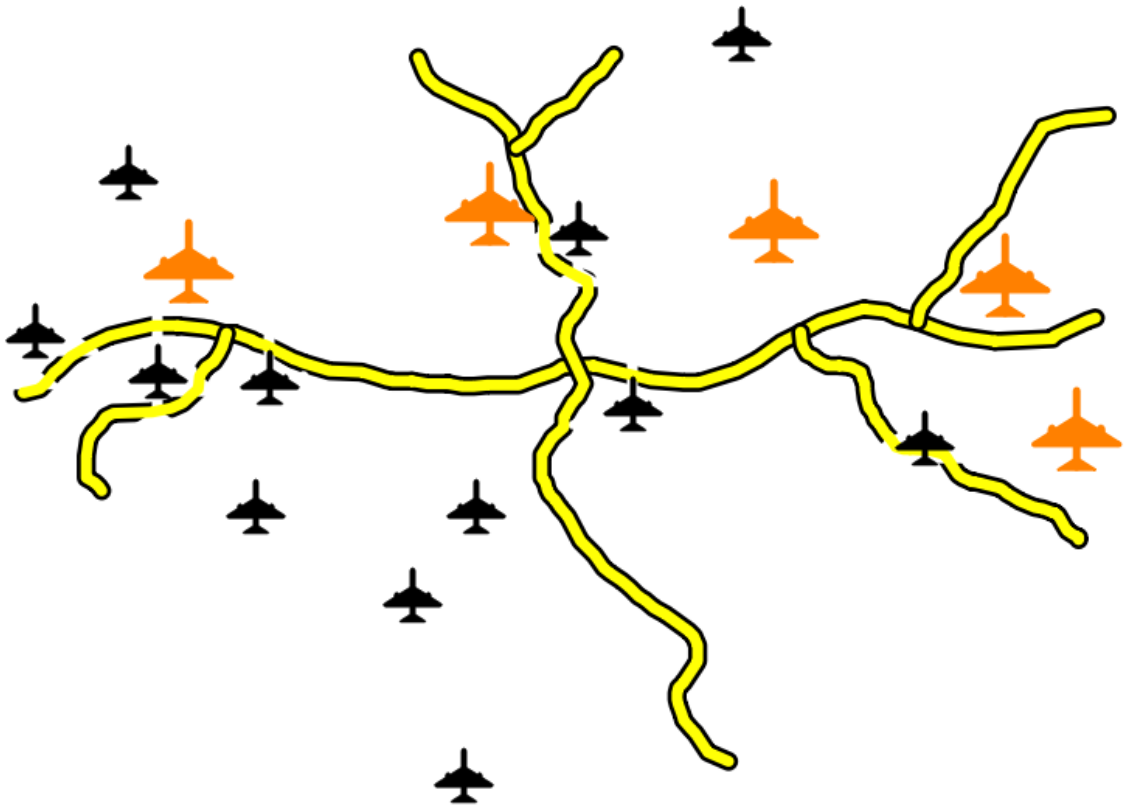
   このようなものが得られる筈です。



6. Jet シンボルについても同じ操作を繰り返します。
7. *OK* をクリックします
8. lines レイヤの **レイヤプロパティ** ウィンドウを開きます。
9. **マスク** タブに移動します。

   マスクするシンボルレイヤの線の黒線部分のチェックボックスは、以前の設定からすでにチェックされています。
10. **ラベルマスク** polys のチェックボックスのチェックを外し、**マスクシンボルレイヤ** B52 と Jet の両方にチェックを入れます。
11. *OK* をクリックします

黒い飛行機が線の黒い部分をマスクし、黄色い部分だけが見えるようになっていることが分かるでしょう。



2.4.17 結論

異なるレイヤのためにシンボルを変更することで、ベクタファイルの集まりを読みやすい地図に変えてきました。何が起こっているか見ることができるだけでなく、それを見るのはうれしくもあります！

2.4.18 参考文献

美しい地図の例

2.4.19 次は?

レイヤ全体のためのシンボルを変更することは便利ですが、それぞれのレイヤの中に含まれている情報は、まだこれらの地図を読んでいる誰かには利用できません。街路が何と呼ばれていますか？ある領域がどの行政地域に属していますか？農場の相対的な表面積は何ですか？この情報のすべてがまだ隠されています。次のレッスンでは、地図上にデータを表現する方法を説明します。

注釈

最近の忘れずに地図を保存しましたか？

第3章　モジュール: ベクタデータを分類する

ベクタデータを分類することで、その属性に応じ、地物（同一レイヤ内の異なるオブジェクト）に異なるシンボルを割り当てることができます。これは、地図を使う人が、様々な地物の属性を簡単に表示することを可能にします。

3.1 レッスン: ベクタ属性データ

ベクタデータは、GIS の日常的な使用において間違いなく最も一般的な種類のデータです。ベクタモデルは、ポイント、ライン、ポリゴン（および 3D データの場合はサーフェスとボリューム）を使用して地理的地物の位置と形状を表し、その他のプロパティは属性として含まれます（QGIS ではよくテーブルとして表示されます）。

今までは、地図に行った変更はどれも、表示されているオブジェクトに影響されませんでした。言い換えれば、全ての土地利用エリアは同じように見え、そして全ての道路は同じように見えます。地図を見ている人は、見ている道路については何も分かっていません。分かっているのは、その地域にその形状の道があるということだけです。

しかし、GIS の全体強みは、地図上に表示される全てのオブジェクトが属性も持っていることです。GIS での地図はただの絵ではありません。それらは場所内のオブジェクトだけでなく、それらのオブジェクトに関する情報も表しています。

このレッスンの目標： ベクタデータの構造について学び、オブジェクトの属性データについて探ります

3.1.1 ★☆☆ （初級レベル） 理解しよう: レイヤ属性を表示する

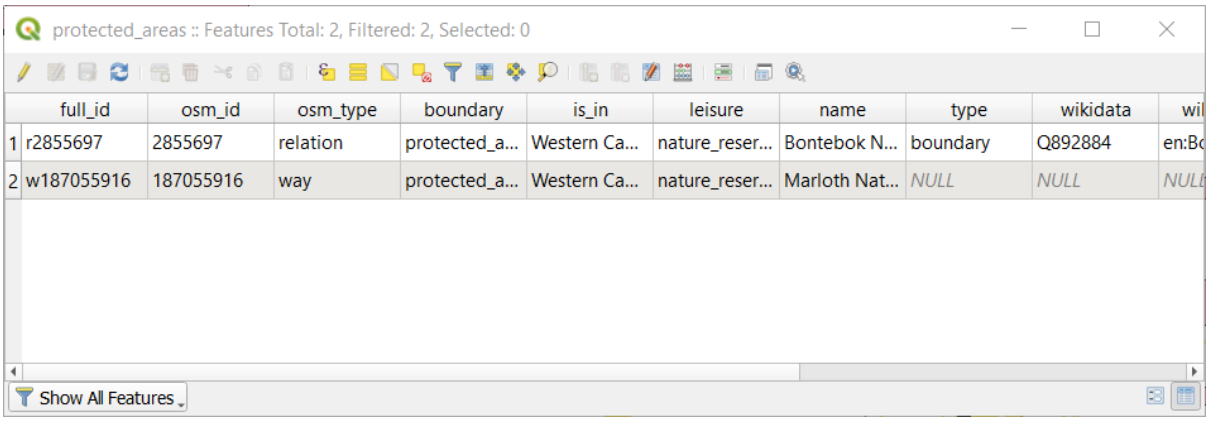
作業しているデータは、オブジェクトが空間の **どこ** にあるかを表すだけでなく、それらのオブジェクトが **何** であるかを語っていることも、知っておくことが重要です。

前の演習から、地図に `protected_areas` レイヤが読み込まれているはずです。読み込まれていない場合は、ディレクトリ `exercise_data/shapefile` に `protected_areas.shp` という *ESRI Shapefile* 形式のデータセットがあります。

保護地域を表すポリゴンが **空間データ** を構成していますが、 **属性テーブル** を調べることで、保護地域についてより詳しく知ることができます。

1. **レイヤ** パネルで、 `protected_areas` レイヤをクリックして選択します。

2. **レイヤ** メニューで、 属性テーブルを開く ボタンをクリックします（トップにあるツールバーのボタンからもアクセスできます）。これにより、protected_areas レイヤの属性テーブルが表示された新しいウィンドウが開かれます。



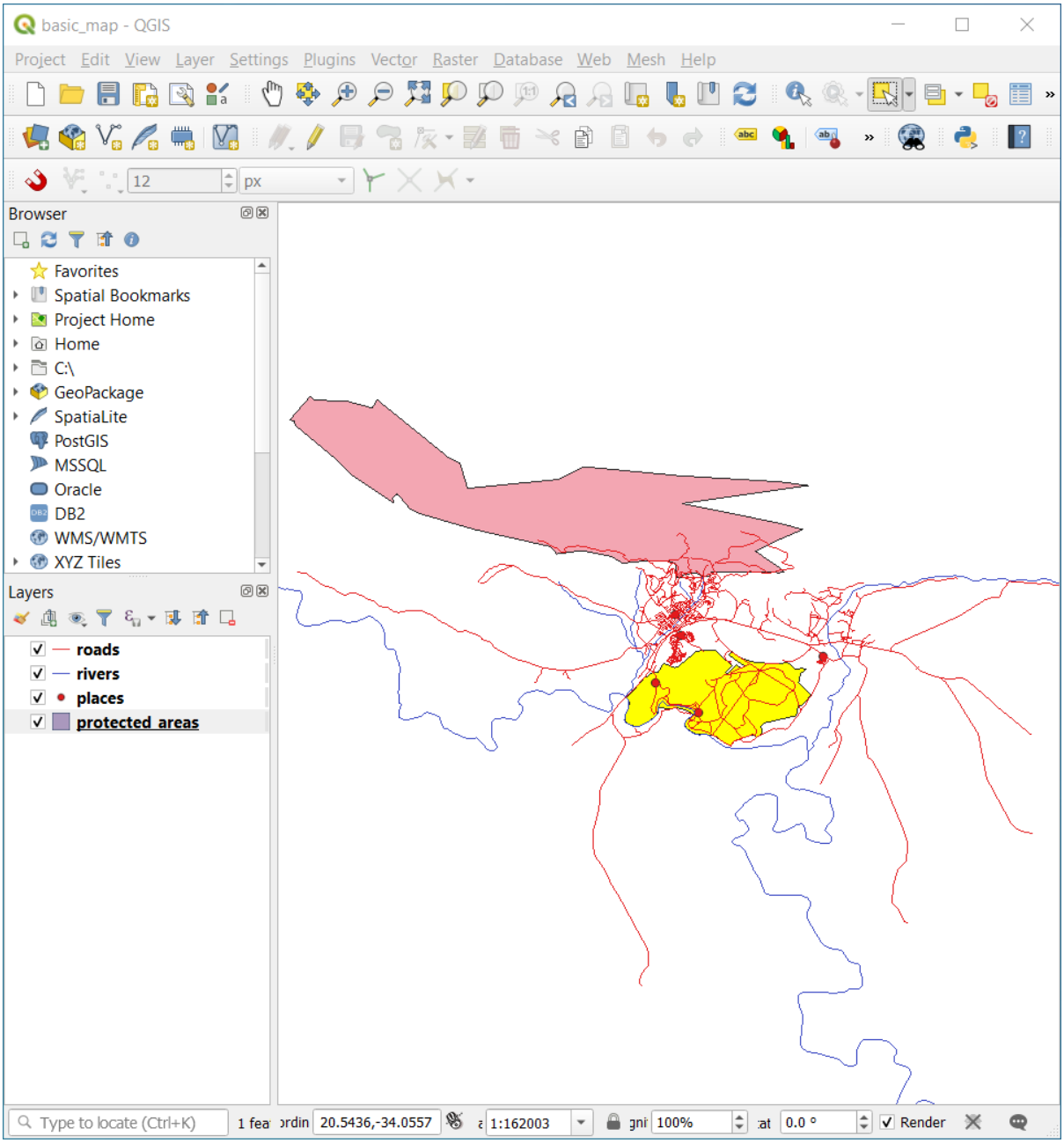
行は **レコード** と呼ばれ、ポリゴンなど、キャンバスマップ中の **地物** に関連付けられています。列は **フィールド**（または **属性**）と呼ばれ、「name」や「id」など、列を説明するのに役立つ名前が付いています。セル内の値は **属性値** と呼ばれます。これらの定義は GIS で一般的に使用されているため、よく理解しておくとよいでしょう。

protected_areas レイヤには 2 つの **地物** があり、マップキャンバスに表示されている 2 つのポリゴンで表現されています。

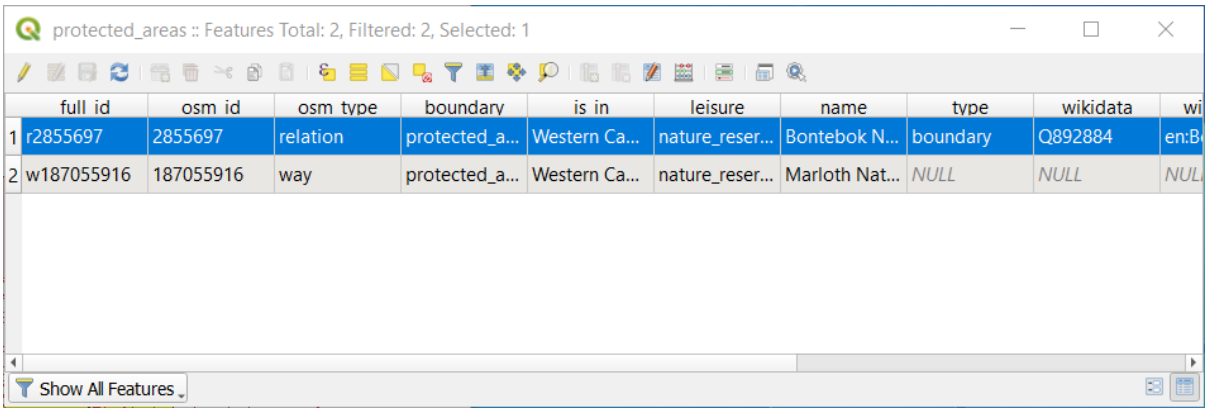
> **注釈**
>
> **フィールド** と **属性値** が何を表しているかを理解するには、属性値の意味を説明するドキュメント（またはメタデータ）を見つける必要がある場合があります。これは通常、データセットの作成者から入手できます。

次に、属性テーブルのレコードが、マップキャンバス上に表示されるポリゴン地物にどのようにリンクされるかを見てみましょう。

1. QGIS のメインウィンドウに戻ります。
2. **編集** 　 **選択** メニューで、 **地物を選択** ボタンをクリックします。
3. **レイヤ** パネルで protected_areas レイヤがまだ選択されていることを確認します。
4. マップキャンバスにマウスを移動し、2 つのポリゴンのうち小さいほうを左クリックします。ポリゴンが黄色に変わり、選択されたことを示します。

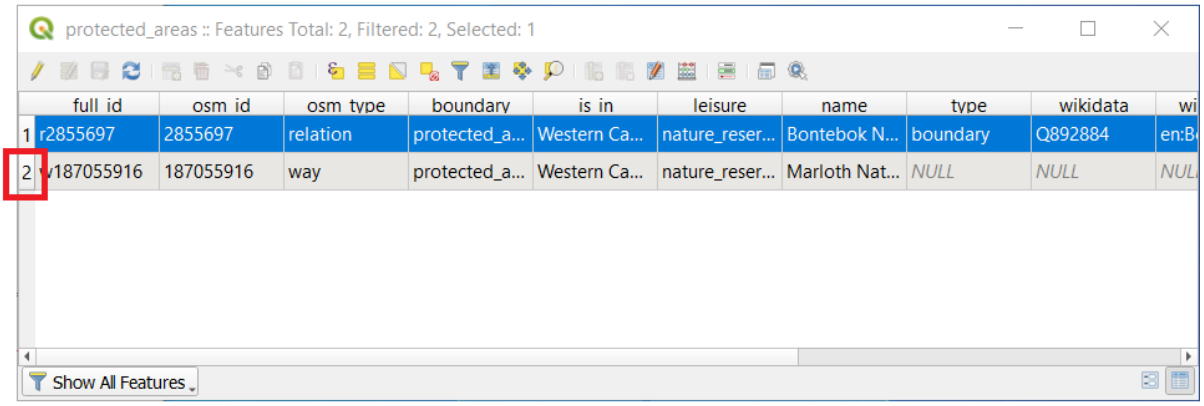


5. **属性テーブル** ウィンドウに戻ると、レコード（行）がハイライトされているのが見えるはずです。これが、選択したポリゴンの属性値です。



また、属性テーブルを使用して地物を選択することもできます。

1. **属性テーブル** ウィンドウの左端にある、現在選択されていないレコードの行番号をクリックします。



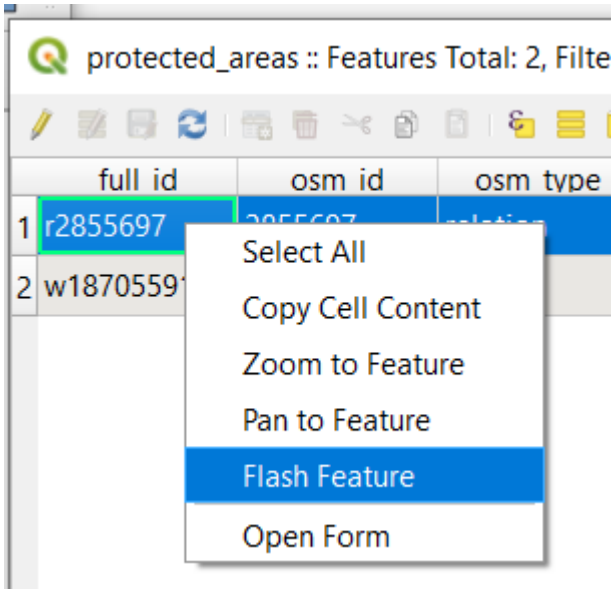
2. QGIS のメインウィンドウに戻り、マップキャンバスを見ます。2 つのポリゴンのうち大きい方が黄色に着色されているのが見えるはずです。

3. この地物の選択を解除するには、**属性テーブル** ウィンドウに行き、 レイヤ内の全地物を選択解除 ボタンをクリックします。

マップキャンバスに表示される地物が多数あり、属性テーブルからどの地物が選択されているかを確認するのが難しい場合があります。地物の場所を特定する別の方法は、**地物をフラッシュ** ツールを使用することです。

1. **属性テーブル** で、フィールド `full_id` の属性値 `r2855697` を持つ行の任意のセルを右クリックします。

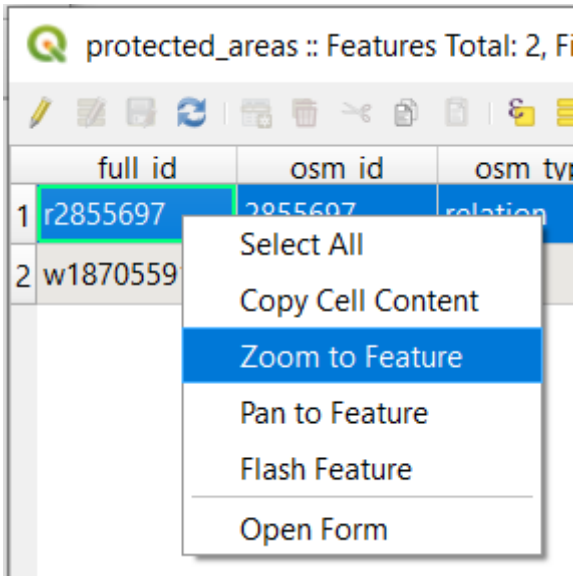
2. コンテキスト メニューの **地物をフラッシュ** をクリックし、マップキャンバスを見ます。



ポリゴンが赤く点滅するのが数回確認できるはずです。もし、見逃した場合は、もう一度試してみてください。

もう一つの便利なツールは **地物にズーム** ツールで、これは QGIS に興味のある地物にズームするように指示します。

1. **属性テーブル** で、`full_id` フィールドの属性値 `r2855697` を持つ行のセルを右クリックします。
2. コンテキスト メニューで、**地物にズーム** をクリックします。



マップキャンバスを見てください。ポリゴンはマップキャンバスの範囲を占めるようになりました。ここで属性テーブルを閉じてください.

3.1.2 ★☆☆ （初級レベル） 自分でやってみよう: ベクタデータ属性を探索する

1. How many fields are available in the `rivers` layer?
2. あなたのデータセットにある 町 の場所について少し教えてください。
3. Open the attribute table for the `places` layer. Which field would be the most useful to represent in label form, and why?

答え

- There should be 9 fields in the `rivers` layer:
  1. **レイヤ** パネルでそのレイヤを選択します。
  2. 右クリックして **属性テーブルを開く** を選択するか、 **属性ツールバー** （ **ビュー** **ツールバー** メニューから有効にすることができます）の ボタンを押します。
  3. 列の数を数えます。

  A quicker approach could be to double-click the `rivers` layer, open the *Layer properties* *Fields* tab, where you will find a numbered list of the table's fields.

- Information about towns is available in the `places` layer. Open its attribute table as you did with the `rivers` layer: there are two features whose *place* attribute is set to `town`: *Swellendam* and *Buffeljagsrivier*. You can add comment on other fields from these two records, if you like.
- `name` フィールドは、ラベルとして表示するのに最も有用なフィールドです。これは、そのすべての値がすべてのオブジェクトに対して一意であり、*NULL* 値を含む可能性が非常に低いからです。データに *NULL* 値が含まれていても、ほとんどの場所に名前があれば心配はありません。

3.1.3 結論

これで使用しているデータ中に実際に何があるかを見るために属性テーブルを使用する方法がわかりましたね。どんなデータセットでも有用なのは気になる属性を持っている場合だけでしょう。どんな属性が必要かを知っていれば、与えられたデータセットが使用できるかどうか、あるいは必要な属性データを持つ別のデータセットを探す必要があるか、すぐに判断できます。

3.1.4 次は?

異なる属性は異なる目的のために有用です。それらのいくつかは、地図のユーザーが見るテキストとして直接表現できます。次のレッスンでこれを行う方法を学びます。

3.2 レッスン: ラベル

オブジェクトに関する情報を表示するためにラベルを地図に追加することができます。ベクタレイヤはそれに関連するラベルを持つことができます。ラベルの内容はレイヤの属性データに依存します。

このレッスンの目標: 役に立ち見栄えの良いラベルをレイヤに適用します。

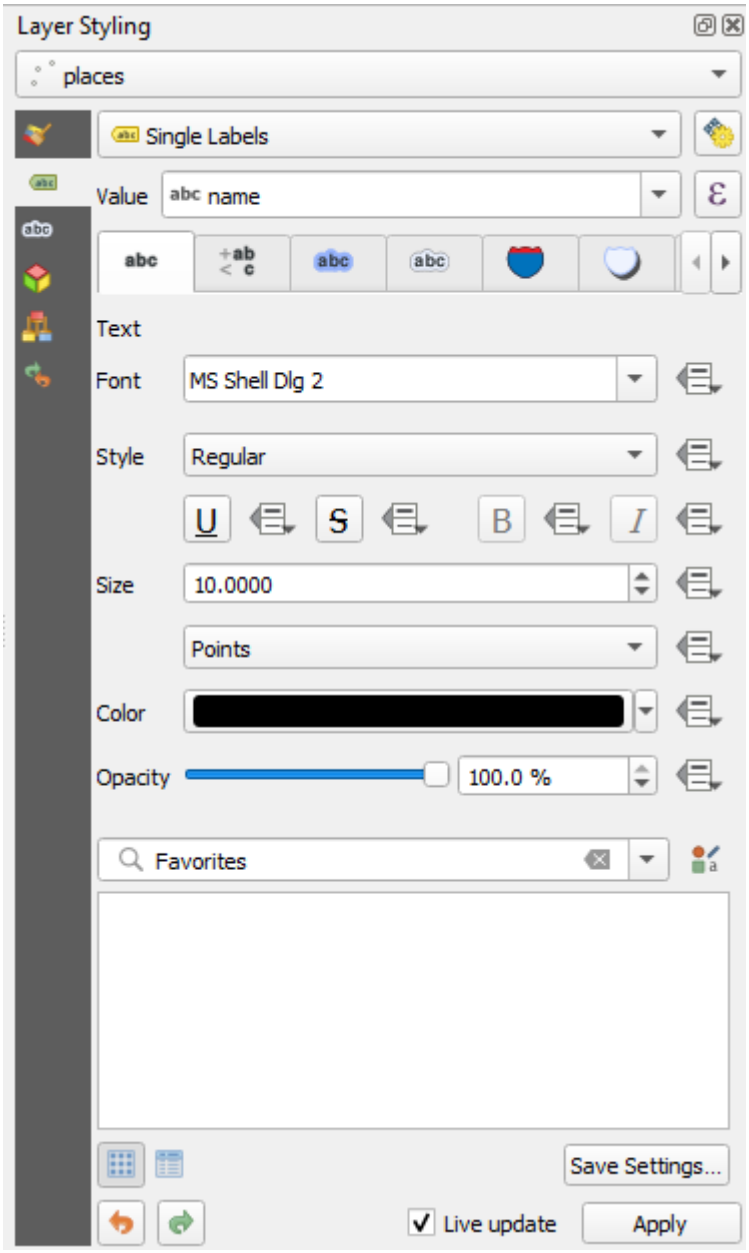
3.2.1 ★☆☆ （初級レベル） 理解しよう: ラベルを使う

まず、 ボタンが GUI に表示されていることを確認します:

1. **ビュー** **ツールバー** メニューに進みます
2. **ラベルツールバー** にチェックマークが付いていることを確認します。チェックマークがない場合は、**ラベルツールバー** をクリックし、アクティブにします。
3. **レイヤ** パネルの `places` レイヤをクリックし、ハイライト表示させます。
4. ツールバーボタンをクリックして、 **レイヤスタイル** パネルの **ラベル** タブを開きます。
5. **ラベルなし** から **単一定義** に切り替えます

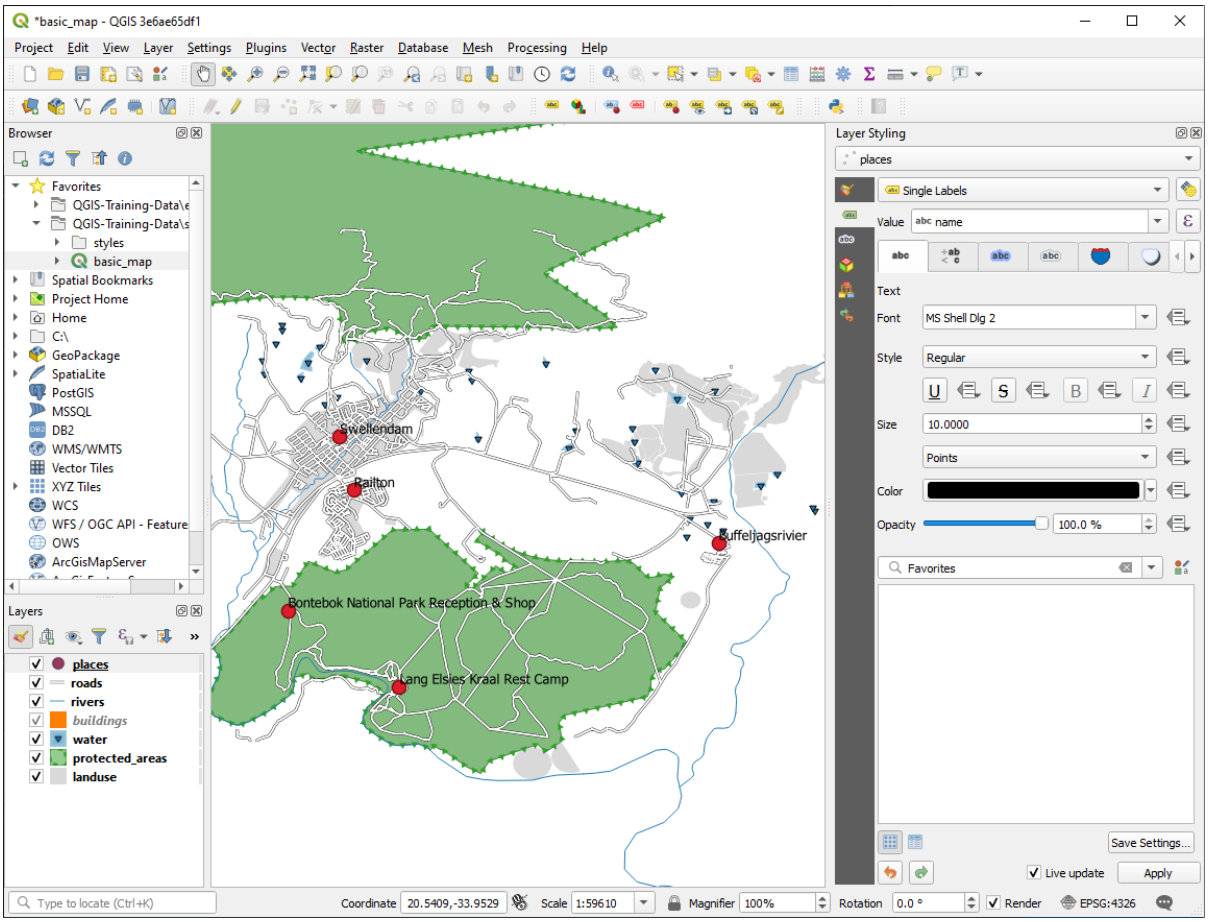
   属性のどのフィールドをラベルに使用するかを選択する必要があります。前のレッスンで、あなたは `name` フィールドがこの目的に最も適していると判断しました。

6. 値リストから name を選択します。



7. **適用** をクリックします。

今、地図にはこのようなラベルが表示されるはずです:

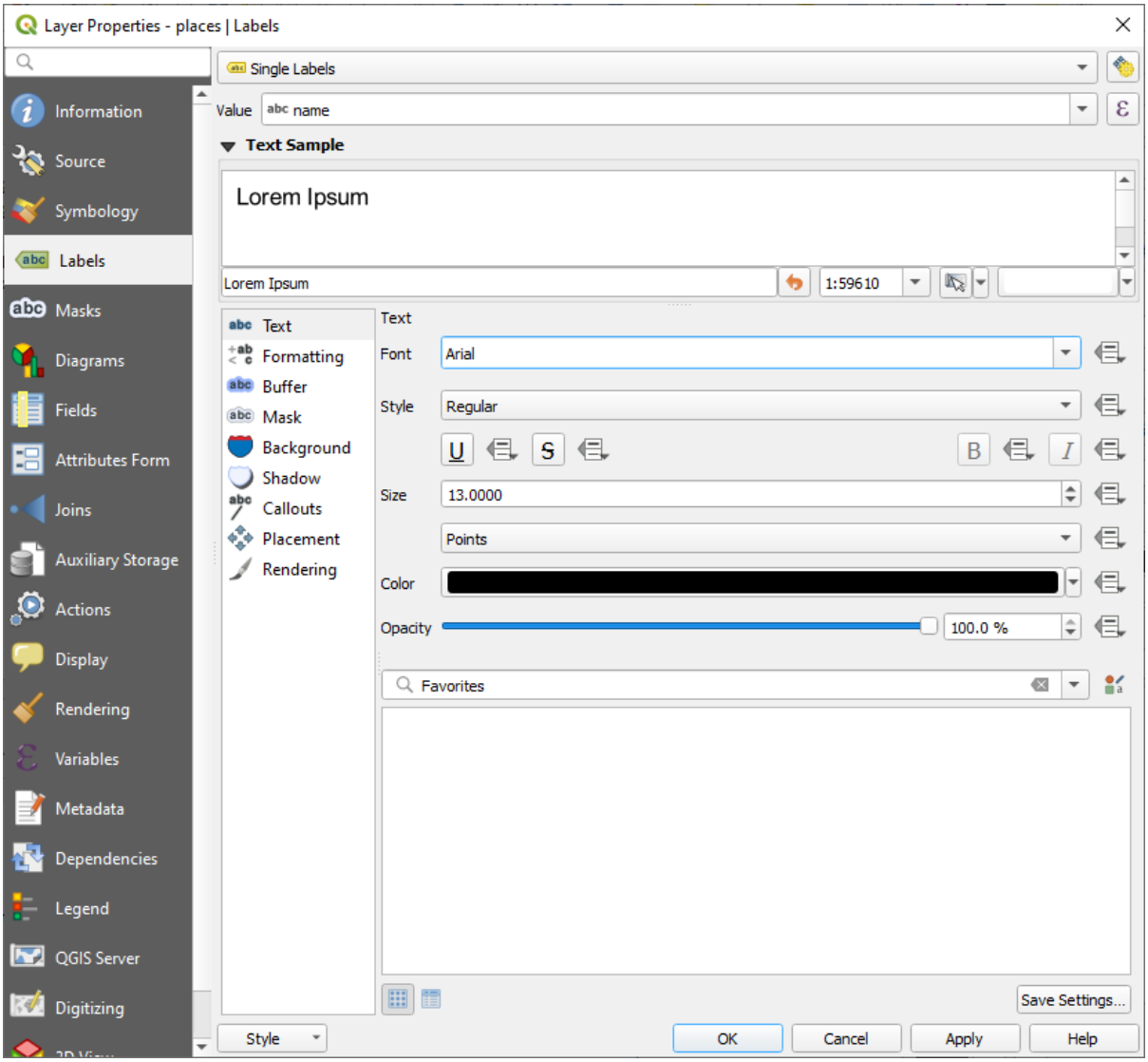


3.2.2 ★☆☆ （初級レベル） 理解しよう: ラベルオプションを変更する

以前のレッスンで地図に選んだスタイルによっては、ラベルが適切にフォーマットされておらず、ポイントマーカーと重なっていたり、遠すぎたりしたのにお気づきかもしれません。

> **注釈**
>
> 上記では、 **ラベルツールバー** の abc ボタンを使って、 **レイヤスタイル** パネルを開きました。 **シンボロジ** と同じように、 **レイヤスタイル** パネルと **レイヤプロパティ** ダイアログの両方で同じラベルオプションが利用できます。ここでは、 **レイヤプロパティ** ダイアログを使用します。

1. places レイヤをダブルクリックし、 **レイヤプロパティ** ダイアログを開きます
2. **ラベル** タブを選びます
3. 左側のオプションリストで **テキスト** が選択されていることを確認し、ここに示されているものと同じになるようにテキスト書式設定を更新します:

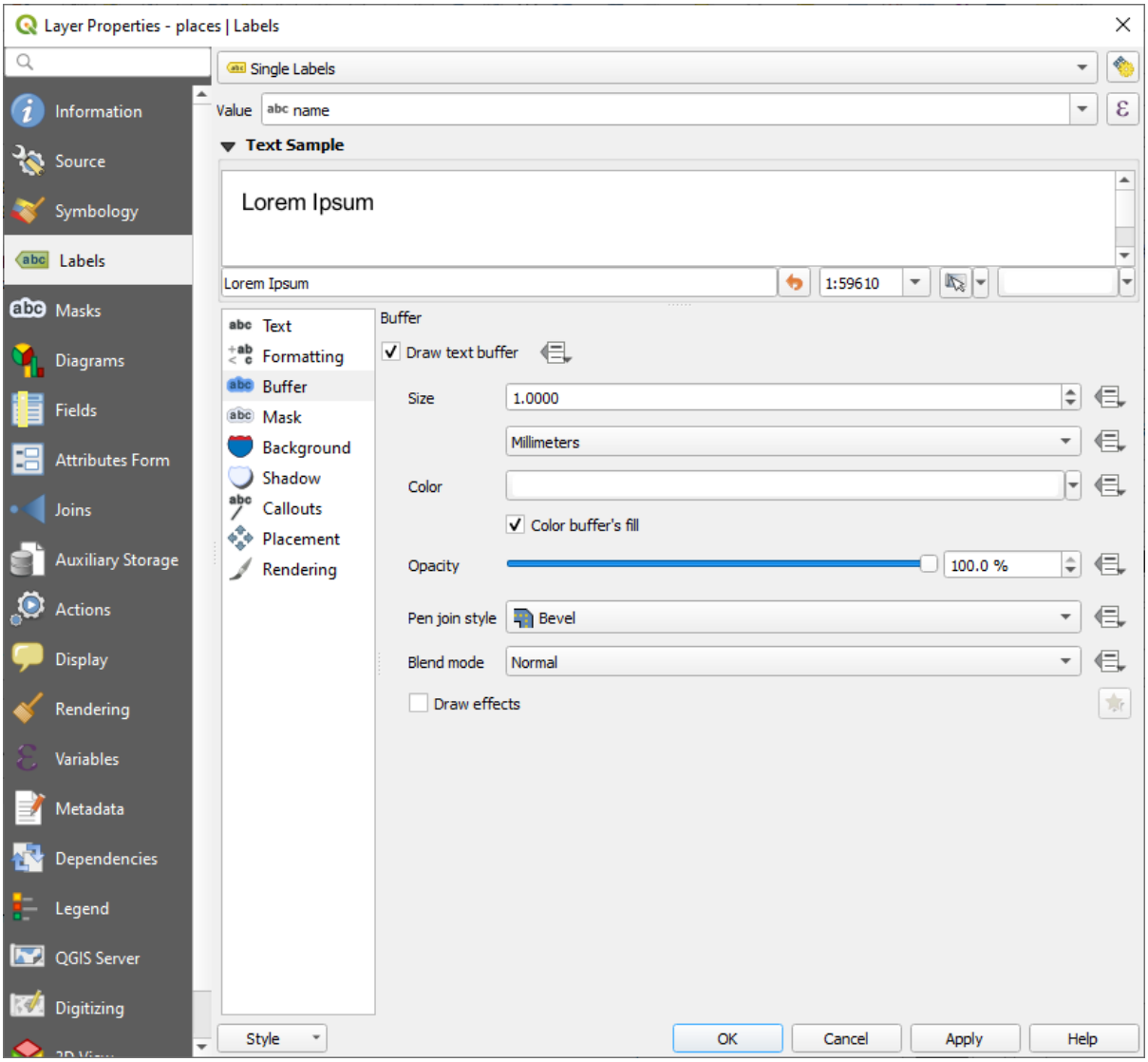


4. **適用** をクリックします。

   そのフォントは、ユーザーにとってより大きく、より馴染みやすいかもしれませんが、その読みやすさは、その下にレンダリングされるレイヤーに依存していることに変わりはありません。これを解決するために **バッファ** オプションを見てみましょう。

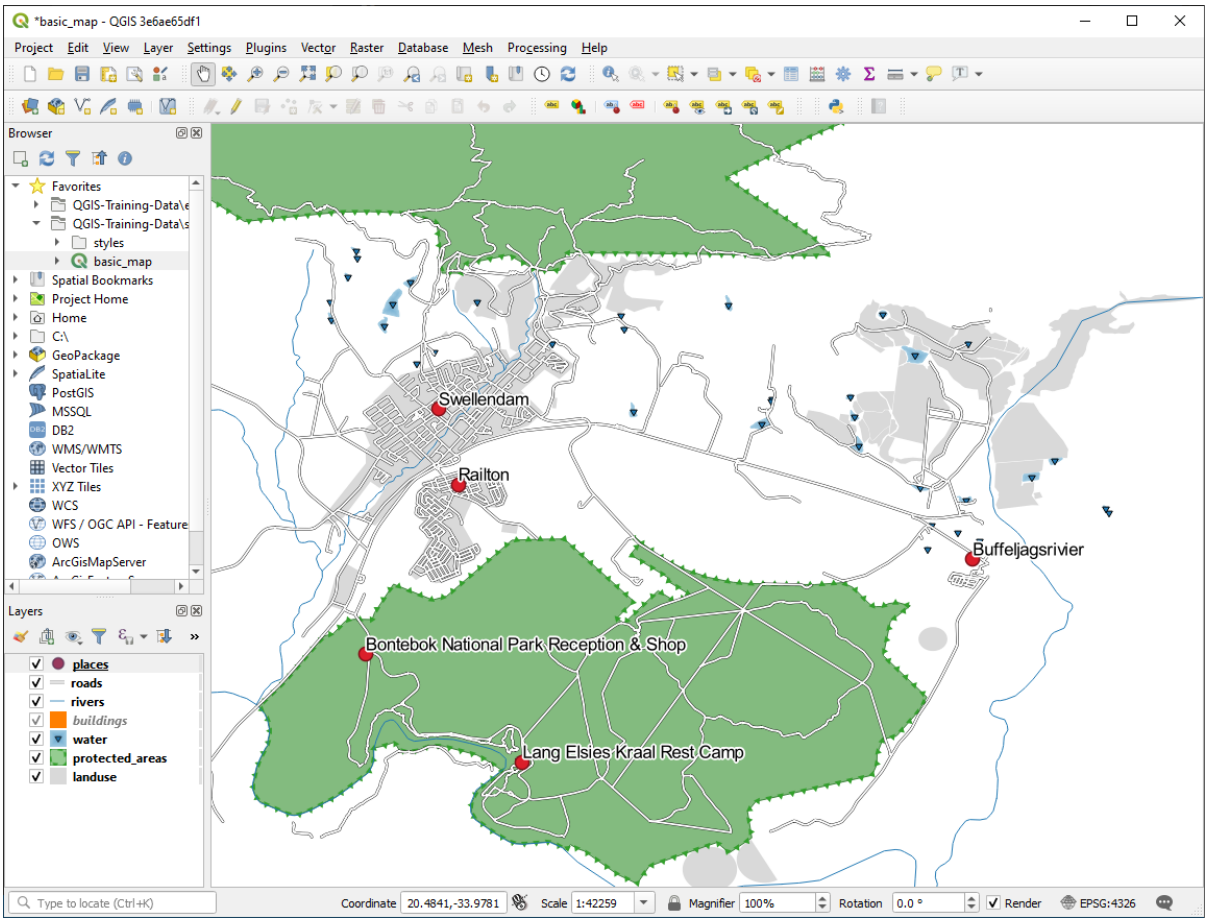
5. 左側のオプションリストから **バッファ** を選びます

6. **テキストバッファを描画** の隣のチェックボックスをチェックし、ここに示されているものと同じになるようにオプションを選びます:



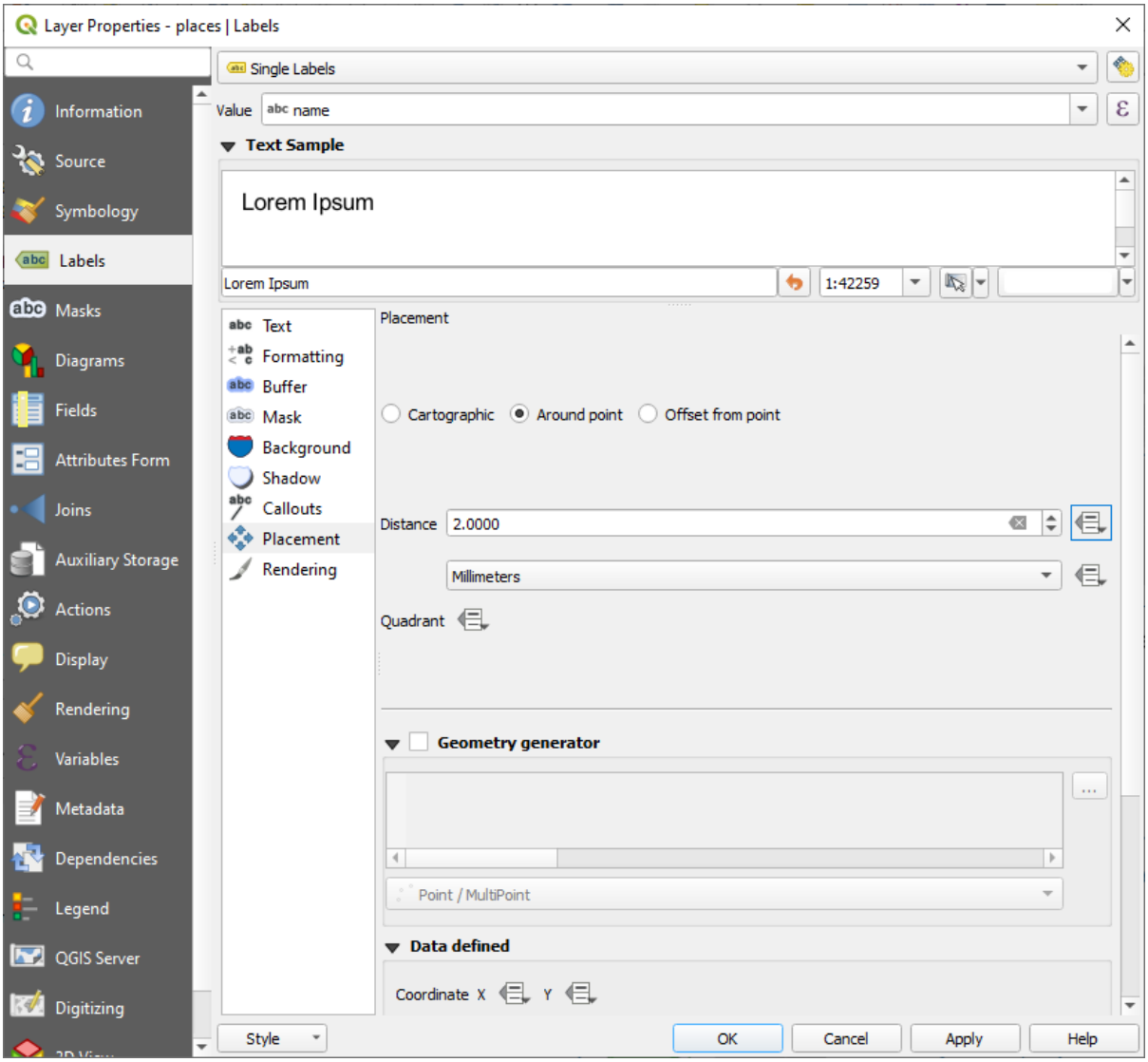
7. **適用** をクリックします。

   着色されたバッファまたは境界線が場所ラベルに追加されて地図上で見分けやすくなりました:



ポイントマーカーに相対的なラベルの配置に取りかかります。

8. 左側のオプションリストから **配置** を選びます
9. **ポイントの周り** を選び、 **距離** の値を `2.0` ミリメートルに変更します:



10. **適用** をクリックします。

    ラベルはもはやポイントマーカーに重なっていません。

3.2.3 ★★☆ （中級レベル） 理解しよう: レイヤシンボルの代わりにラベルを使う

多くの場合、ポイントの位置はそれほど特定する必要はありません。例えば、 `places` レイヤのポイントのほとんどは、町全体や郊外を指しており、そのような地物に関連する特定のポイントは、大きな縮尺ではそれほど特定的ではありません。実際、あまりに具体的なポイントを与えると、地図を読んでいる人がしばしば混乱します。

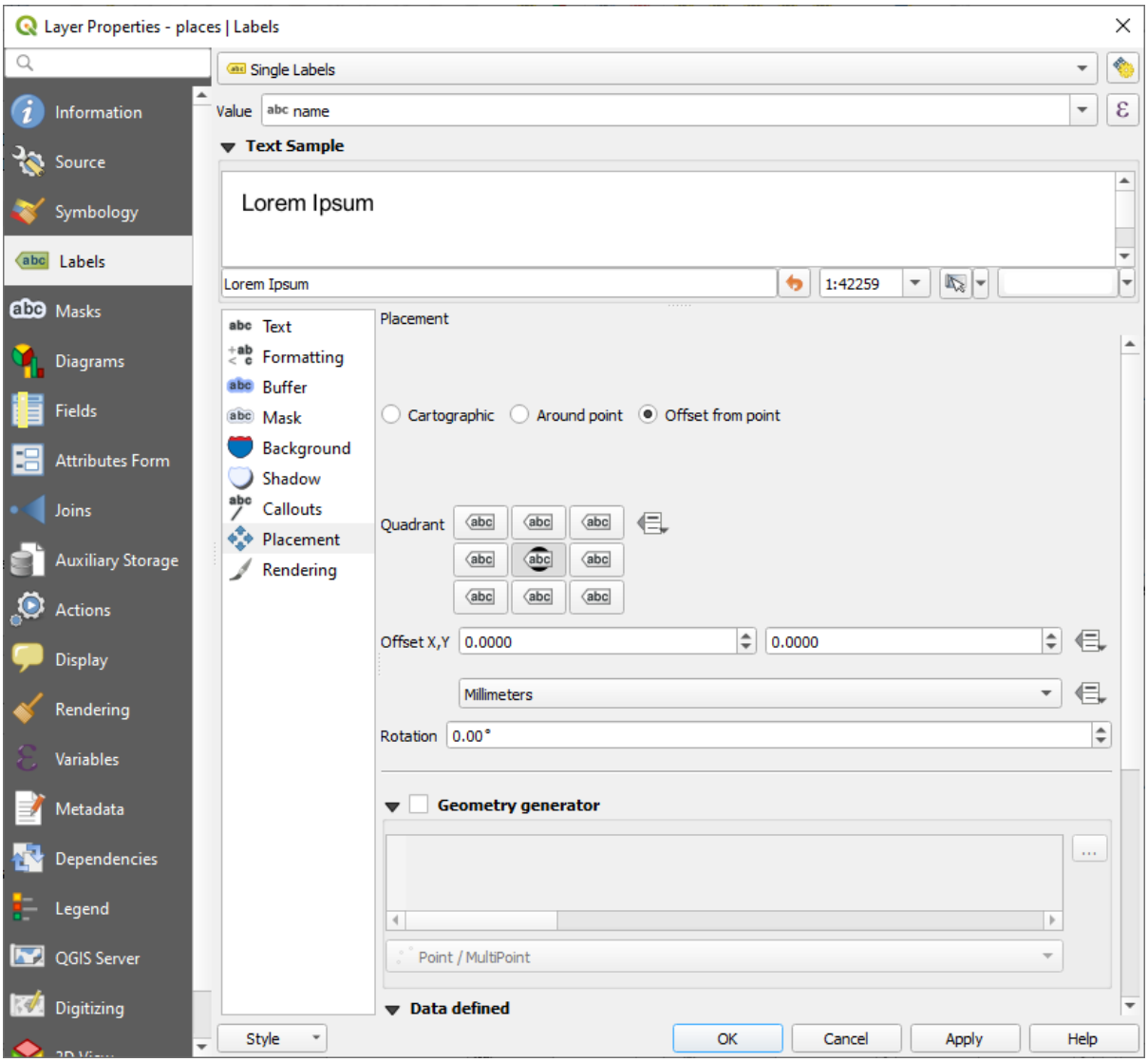
例を挙げると: 例えば、世界地図上では欧州連合のために与えられた点はポーランドのどこかにあります。*European Union* のラベルの付いたポイントがポーランドにあるので、地図を読んでいる人には欧州連合の首都はポーランドにあるように見える場合があります。

ですから、この種の誤解を防ぐためにポイントシンボルを非アクティブ化してラベルに完全に置き換えるのがよい場合があります。

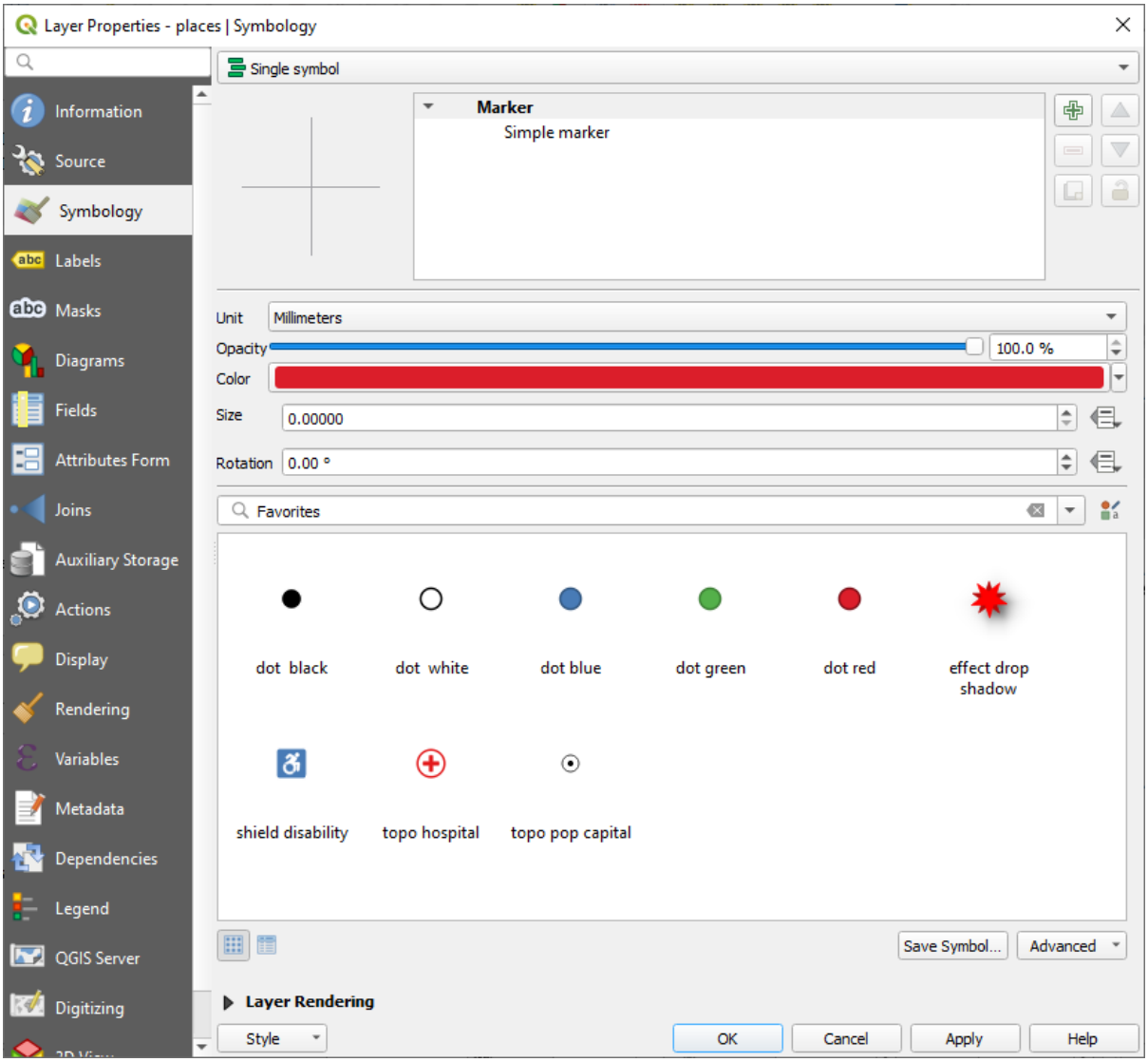
QGIS ではラベルが参照するポイントの直上にラベルの位置を変更することによってこれをすることができます。

1. `places` レイヤの **レイヤプロパティ** ダイアログの **ラベル** タブを開きます
2. オプションリストから **配置** オプションを選びます
3. **点からのオフセット** ボタンをクリックします

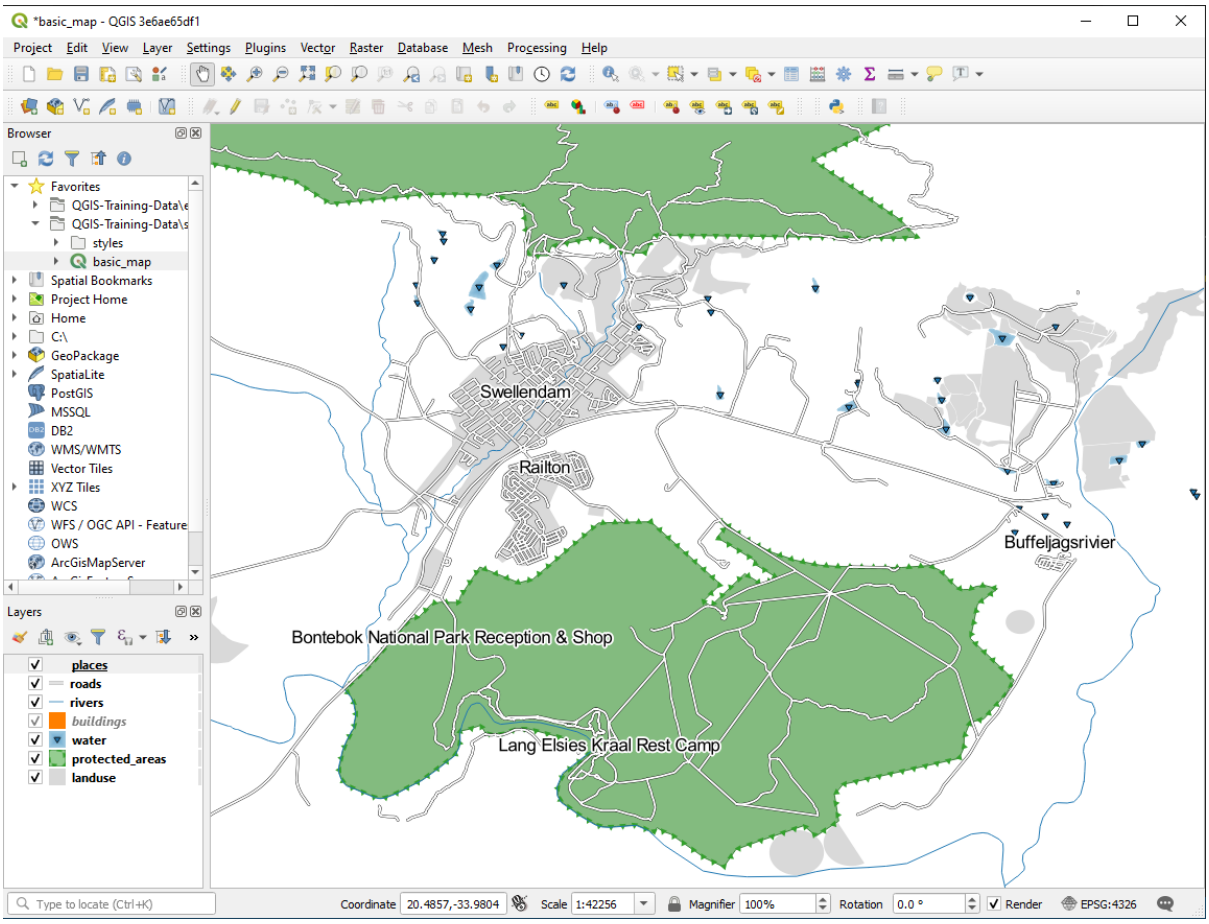
   ポイントマーカーとの相対位置でラベルの位置を設定することのできる **象限** オプションが現れます。この場合、ラベルは点を中心に配置したいので中央の象限を選択します:



4. 通常通りレイヤの **シンボロジ** を編集し、 **マーカー** の大きさを 0.0 にしてポイントシンボルを隠します：



5. **適用** をクリックすると、このような結果が表示されます:



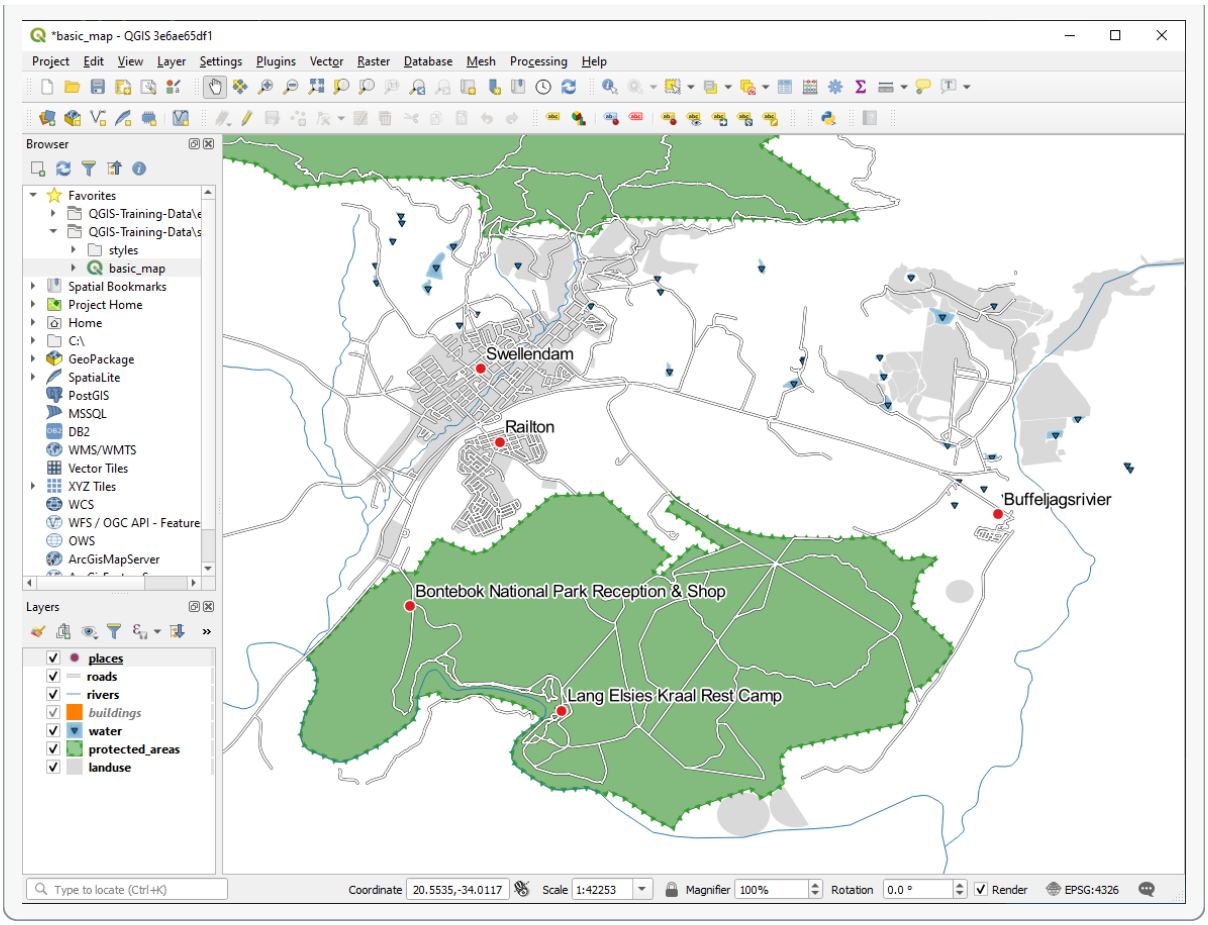
地図を縮小すると、大きな縮尺では重なりを避けるためにいくつかのラベルが消えているのがわかるでしょう。多くの点を持つデータセットを扱う場合、これが望ましい場合もありますが、この方法では有用な情報が失われてしまう場合もあります。このような場合を扱うもう一つの可能性があり、このレッスンの後の練習で取り上げます。とりあえず縮小して、ツールバーにある ボタンをクリックし、何が起こるか見てみましょう。

3.2.4 ★★☆ （中級レベル） 自分でやってみよう: ラベルのカスタマイズ

- ラベルとシンボルの設定をリセットして、ポイントマーカーとラベルオフセットを `2.0` ミリメートルにします。

答え

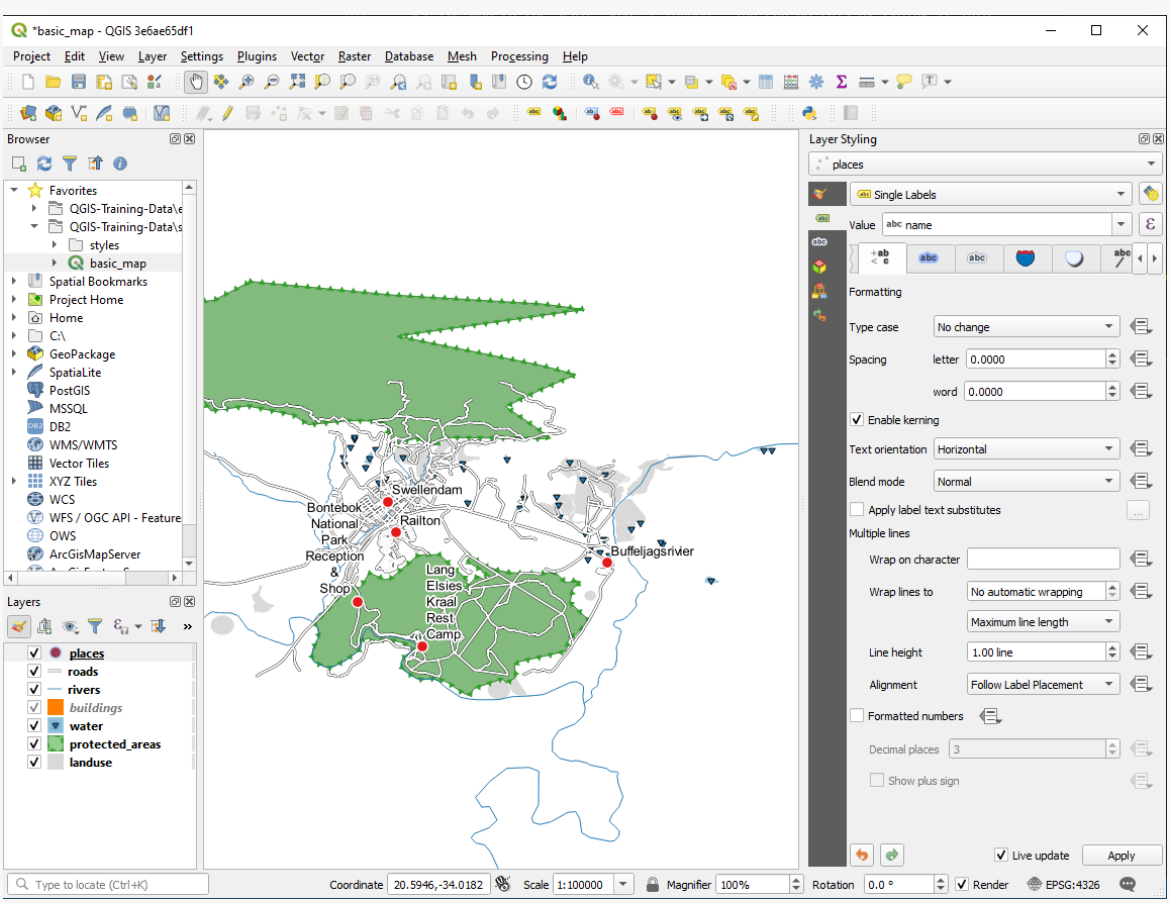
これで地図はマーカーポイントを表示し、ラベルは 2mm ずれているはずです。マーカーとラベルのスタイルは、両方が地図上ではっきり見えるようにする必要があります。



- 地図の縮尺を `1:100000` に設定します。これは、 **ステータスバー** の **縮尺** ボックスに入力することで行うことができます。この縮尺で見るのに適したラベルに変更します。

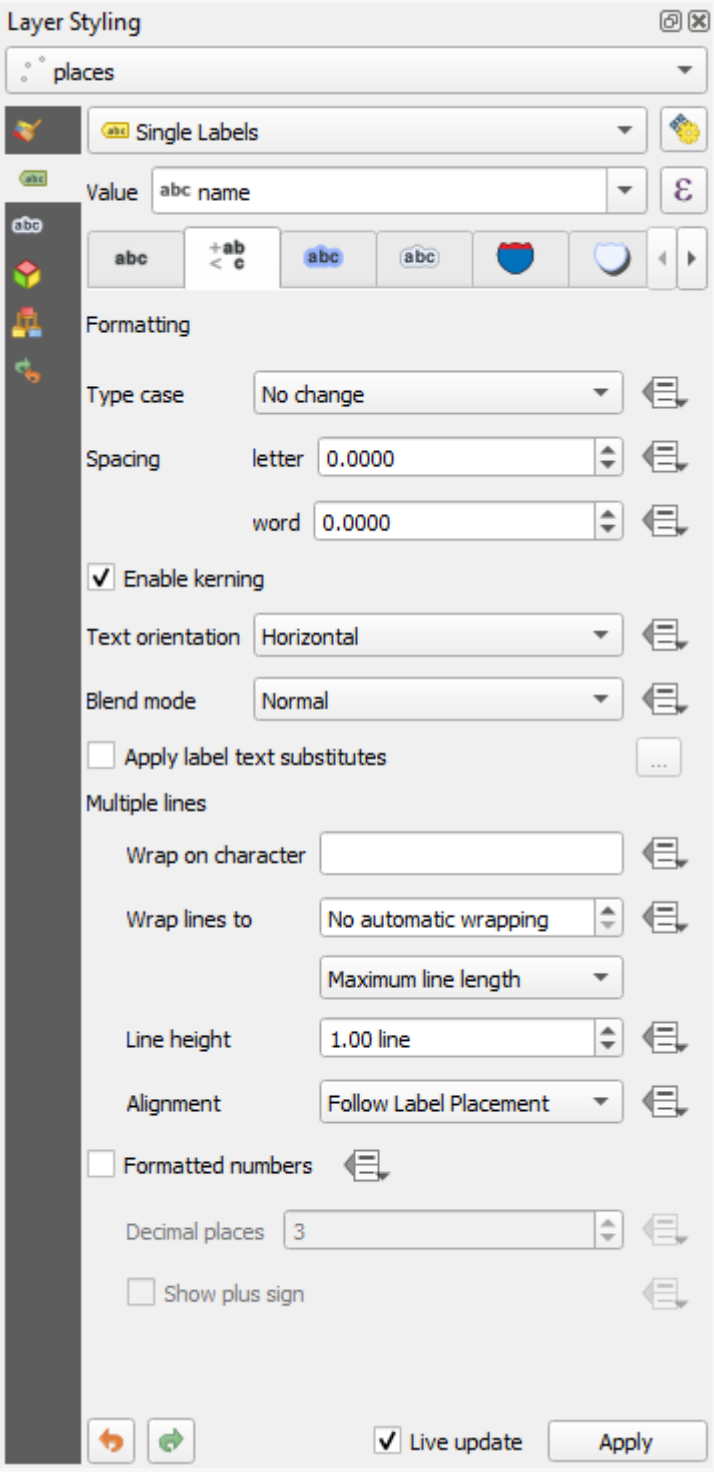
答え

一つの解として、最終的にこのような結果が得られます：



この結果に到達するためには：

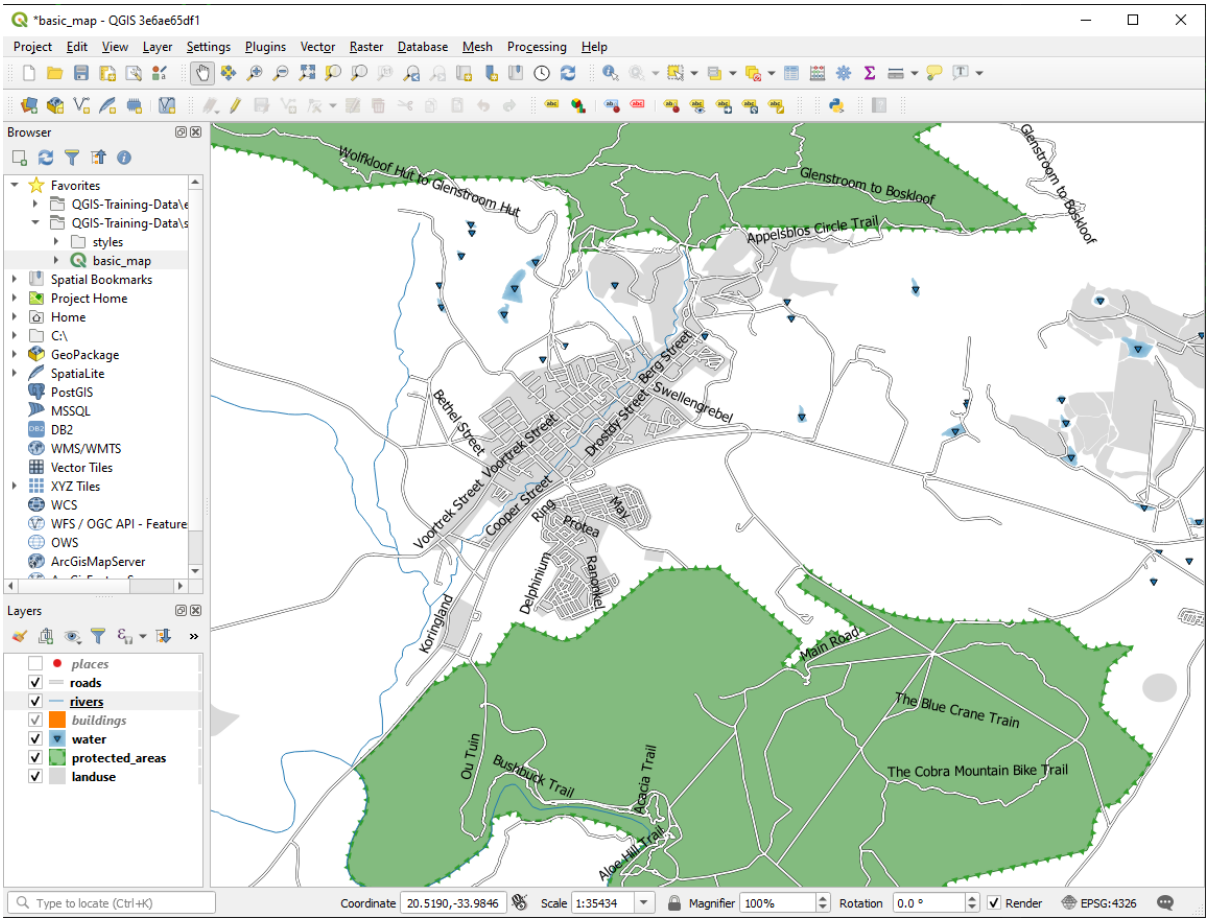
- フォントサイズ 10 を使用
- ポイントの周りの距離に 1.5 mm を使用
- マーカーサイズに 3.0 mm を使用
- さらに、この例では **この文字でラップ処理** オプションを使用しています:



- このフィールドに 空白 を入力し、 **適用** をクリックすると同じ効果が得られます。この例では、地名の一部が非常に長いため、名前が何行にもなってしまい、使い勝手が悪くなっています。この設定は、あなたの地図にもっとふさわしいかもしれません。

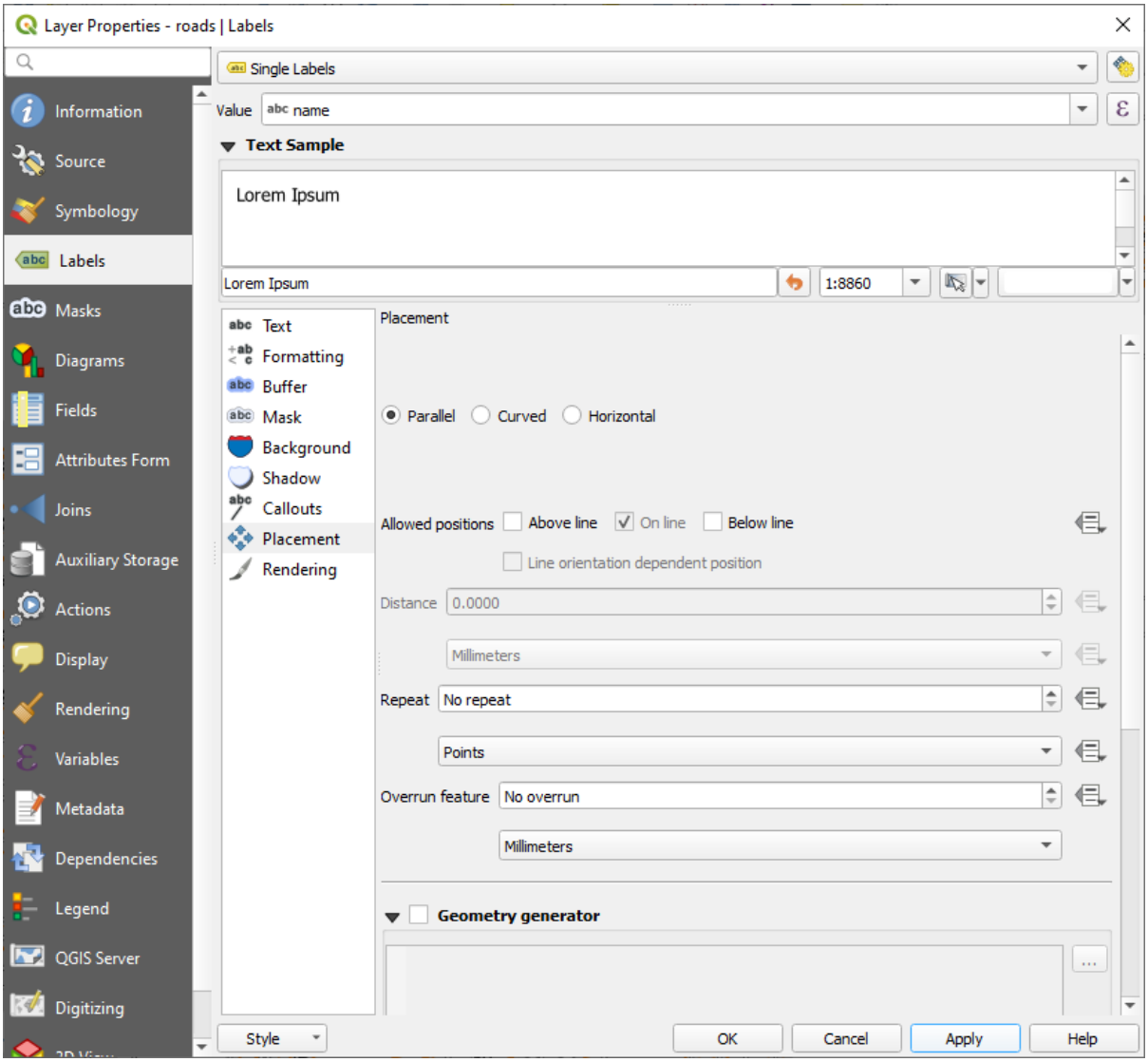
3.2.5 ★★☆ （中級レベル） 理解しよう: ラインにラベルを付ける

これでラベルの作成がどのように動作するかがわかりましたが、まだ別の問題があります。ポイントとポリゴンにラベルを付けるのは簡単ですが、ラインはどうでしょうか? ポイントと同じようにラベルを付ける場合、このようになります:



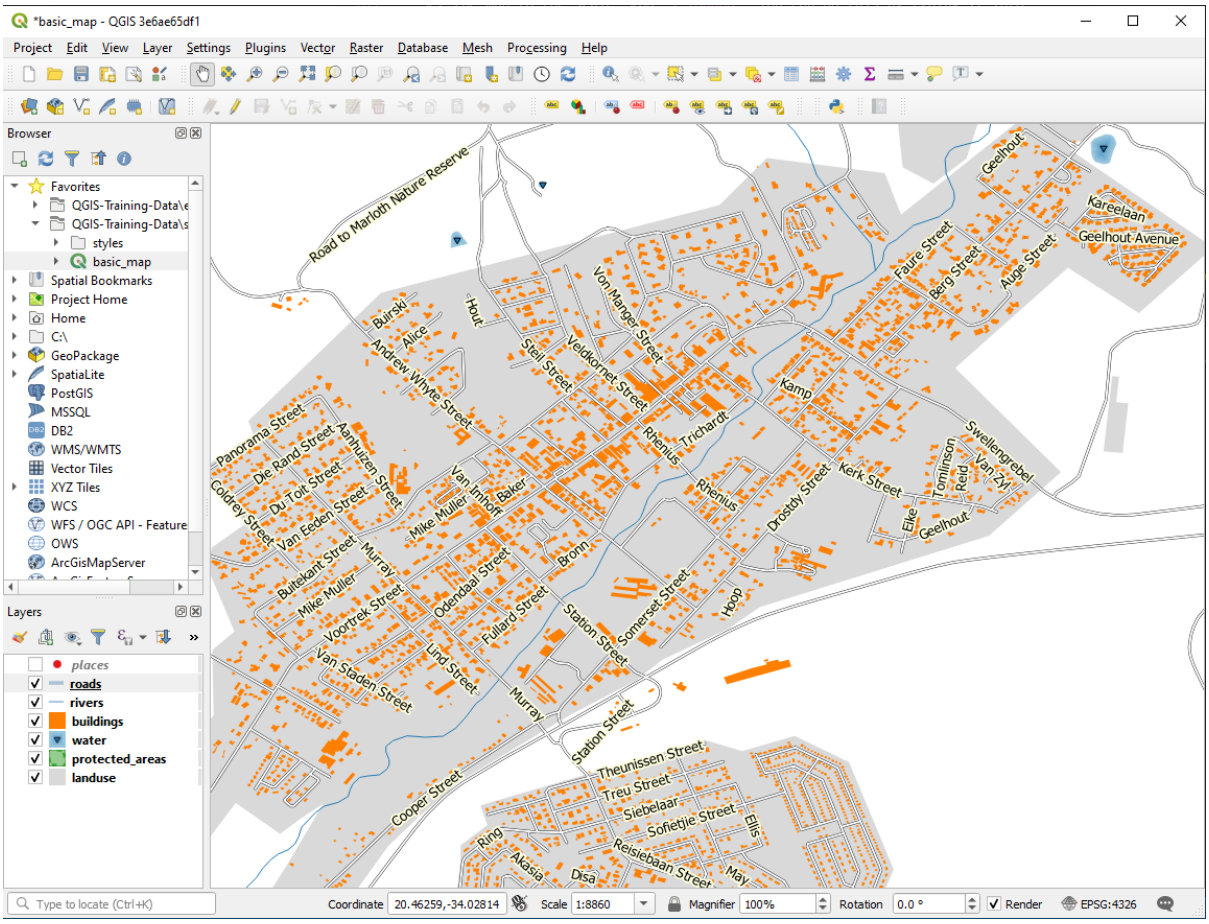
次に、roads レイヤの体裁を整え直してラベルを分かりやすくします。

1. places レイヤを非表示にして、邪魔にならないようにします
2. places のラベルと同じように roads レイヤのラベルの **単一定義** をアクティブにします
3. フォントの **大きさ** を 10 にして、より多くのラベルを表示できるようにします
4. Swellendam 町域を拡大します。
5. **ラベル** タブの **配置** タブで、次の設定を選択します



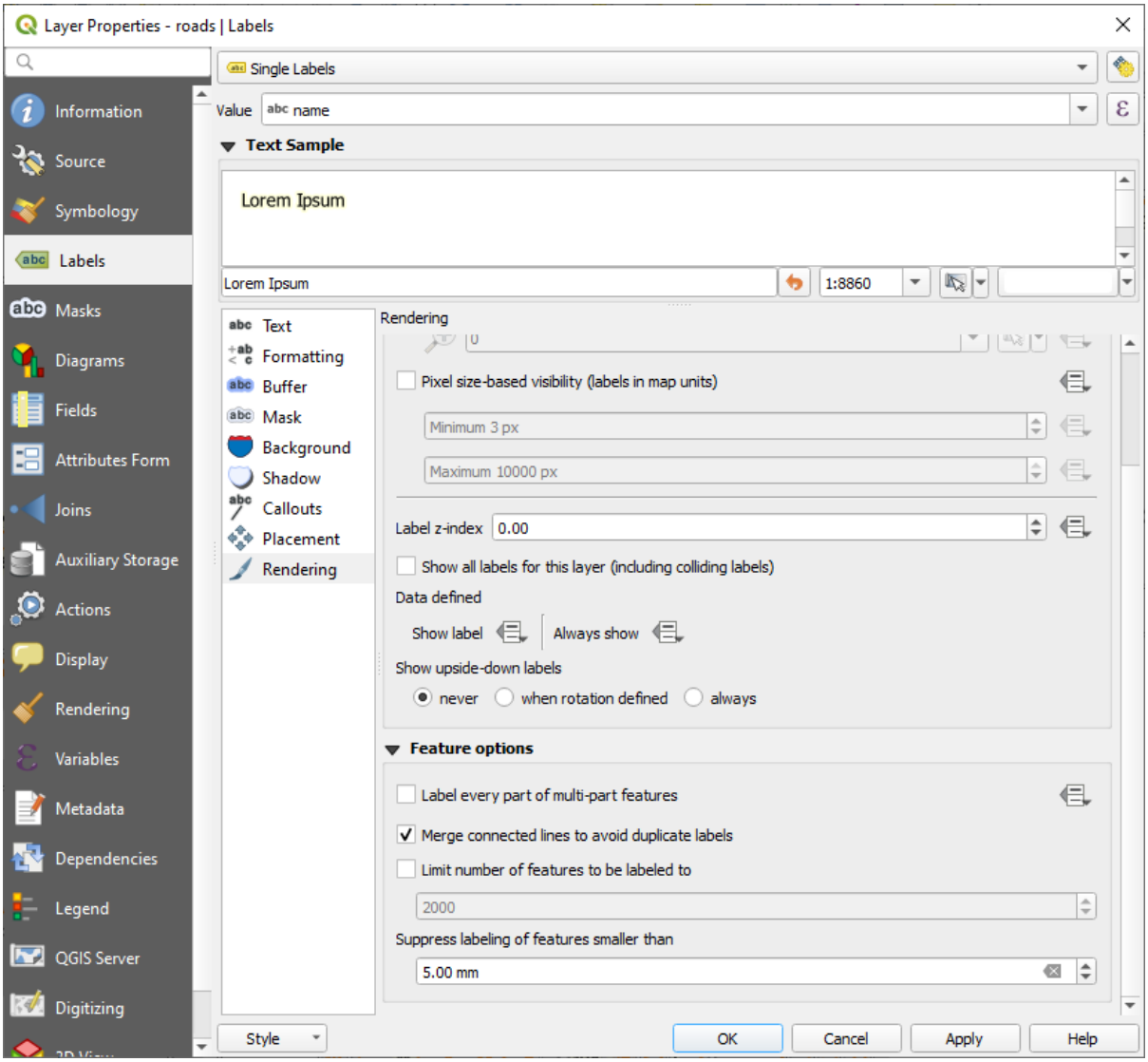
テキストのスタイリングにデフォルト値が使用されているため、ラベルが非常に読みにくいことが分かると思います。 **テキスト** の **色** を濃い灰色か黒に、 **バッファ** の **色** を薄い黄色に更新してください。

地図は縮尺に応じてこのように見えます:



いくつかの道路名が複数回表示されますがいつも必要だとは限りません。これを防ぐには:

6. **レイヤプロパティ** ダイアログの **ラベル** タブで **レンダリング** オプションを選び、図のように **重複ラベルを除去するために接続する線を結合** を選びます。



7. *OK* をクリックします

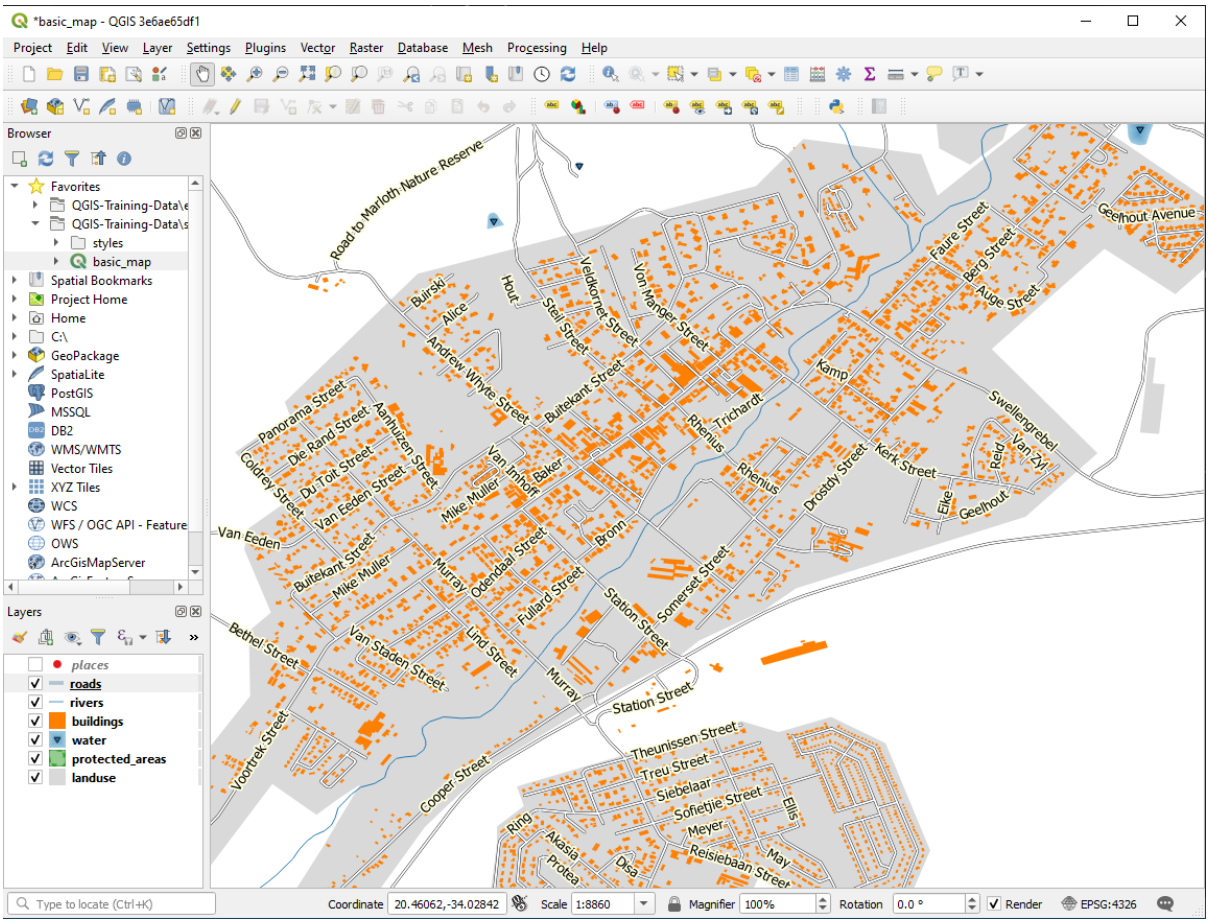
   もう 1 つの有用な機能はラベルを付けるには短すぎる地物にラベルが描画されないようにするものです。

8. 同じ **レンダリング** パネルで、 **これより地物が小さい場合は省略** の値を `5.00 mm` に設定し、 **適用** をクリックしたときの結果に注意してください。

   別の **配置** の設定も試してみてください。前に見たように、この場合、 **水平** オプションは良いアイデアではないので、代わりに **線に沿って湾曲** オプションを試してみましょう。

9. **ラベル** タブの **配置** パネルで **線に沿って湾曲** オプションを選びます

これが結果です:

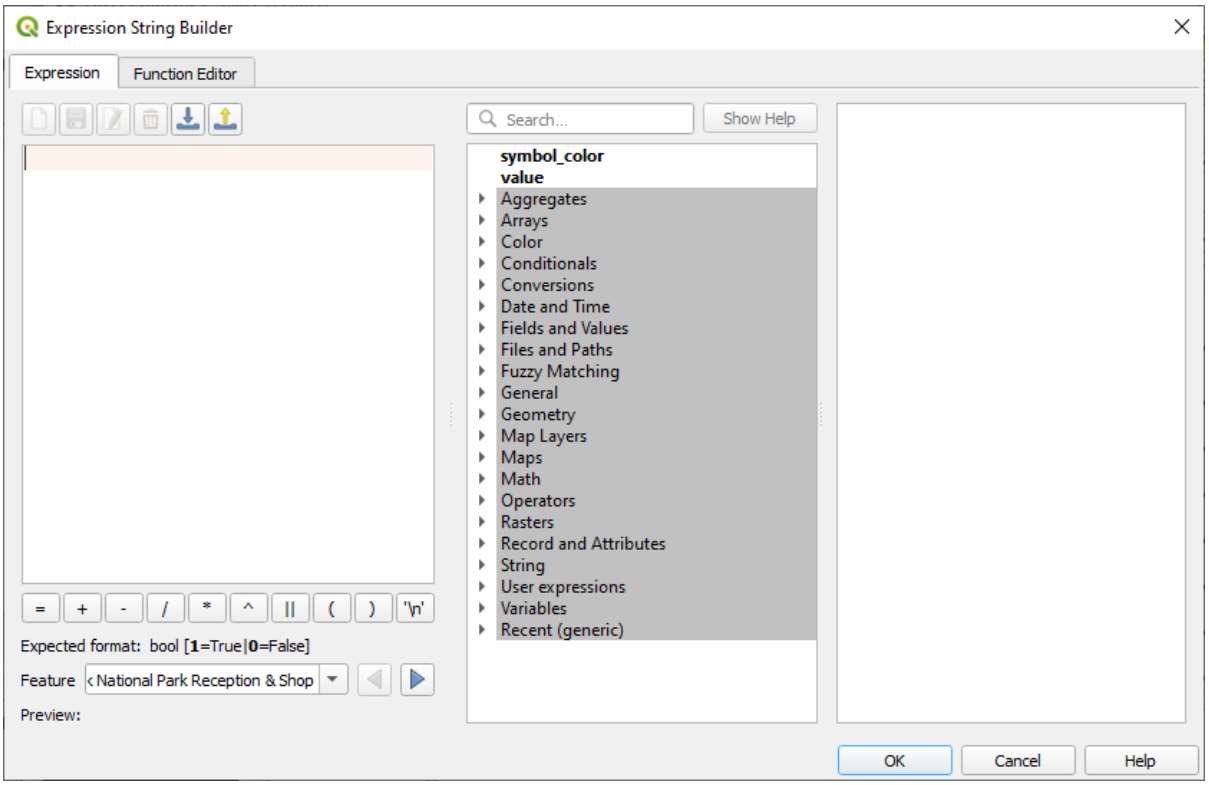


このように、以前は表示されていたいくつかのラベルが非表示になっています。これは、曲がりくねった道路の線に沿わせながら、読みやすくするのが難しいからです。また、他のラベルは、道路と道路の間の空間に浮かんでいるのではなく、道路を追跡するため、より有用になります。どちらのオプションを使うかは、より有用なもの、より見栄えの良いものを選ぶことができます。

3.2.6 ★★★ （上級レベル） 理解しよう: データ定義による設定

1. roads レイヤのラベリングを無効にします
2. places レイヤのラベリングを再び有効にします
3.  ボタンで places の属性テーブルを開きます

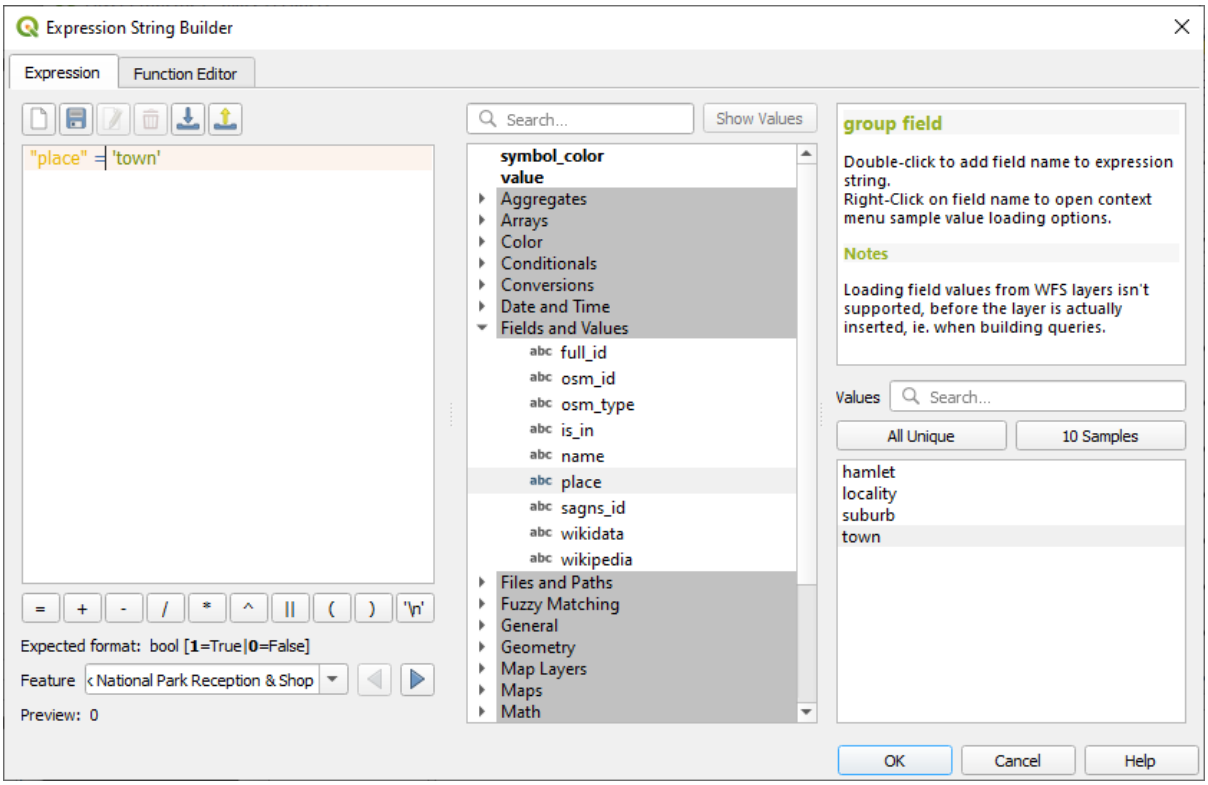
   それには各 **レコード** の都市部のタイプを定義する興味深い places という **フィールド** があります。このデータを使って、ラベルのスタイルに影響を与えることができます。
4. places **ラベル** パネルで **テキスト** パネルに移動します
5. **スタイル** の下の イタリックテキスト ボタンの隣にある  ボタンをクリックして、 **編集**... を選び、 **式文字列ビルダ** を開きます：



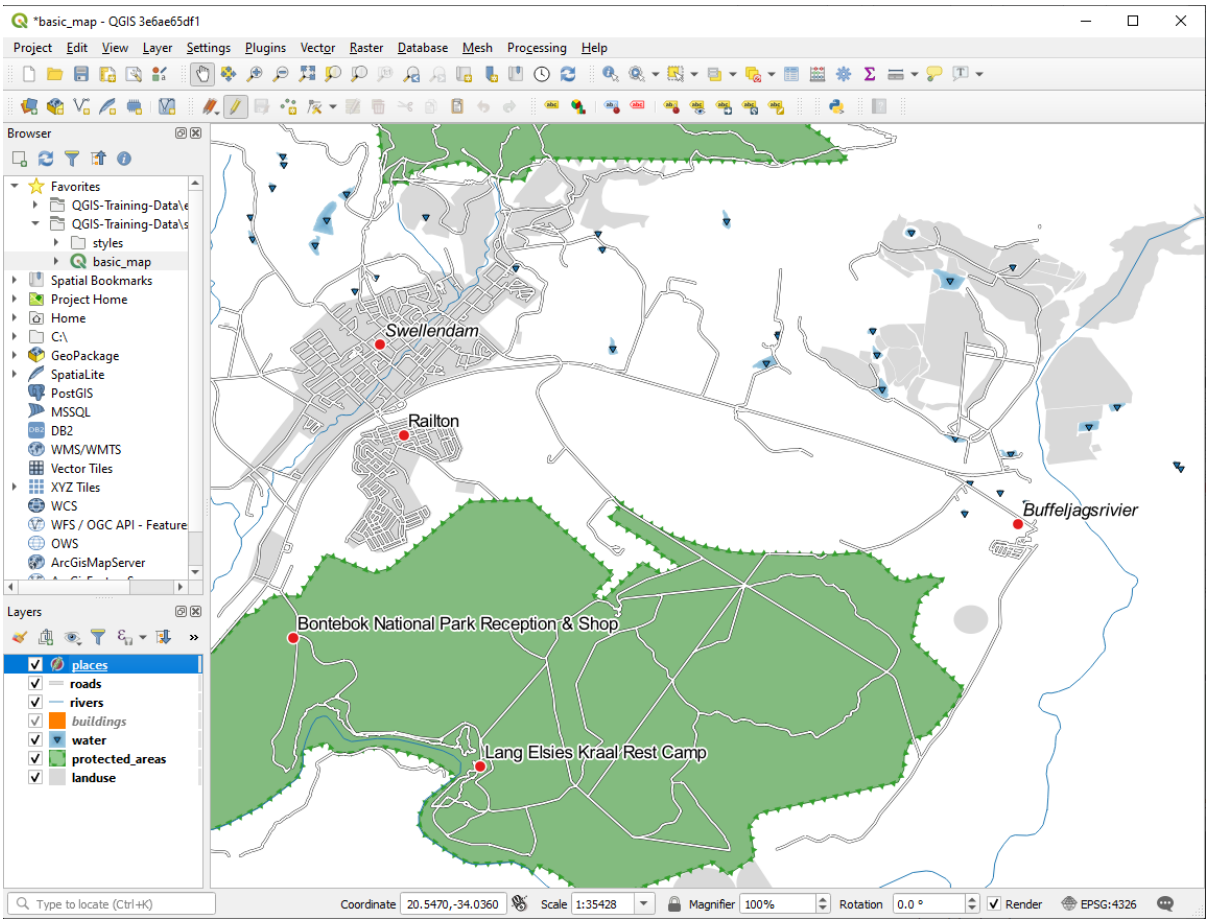
6. **フィールドと値** の下にある `place` をダブルクリックし、 **全ユニーク** をクリックします。これにより、このレイヤーの `place` フィールドのすべての固有値が一覧表示されます。テキストエディタで = を追加し、 `town` をダブルクリックします。

   または、次のようにテキストエディターで `"place" = 'town'` と直接入力することもできます。

7. *OK* を 2 回クリックします。



place フィールドが town と一致するすべての場所のラベルがイタリック体で表示されていることに注目してください。

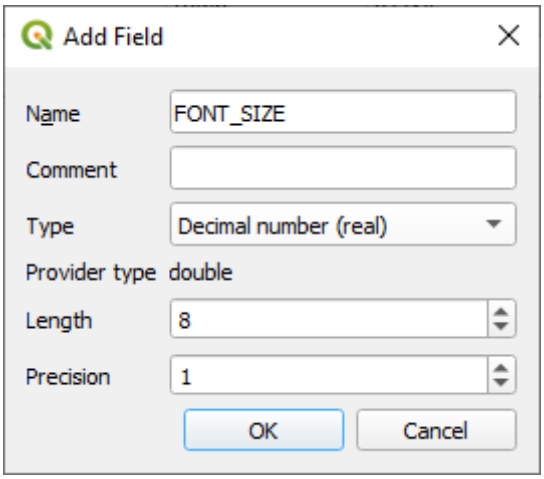


3.2.7 ★★★ （上級レベル） 自分でやってみよう: データ定義による設定を使う

注釈

私たちはいくつかの高度なラベル付け設定を示すために、ここで少し先に飛んでいます。上級レベルでは以下のことが何を意味するか知っていることを想定しています。そうでなければこのセクションは跳ばし、必要な題材をカバーしてから戻って来て下さい。

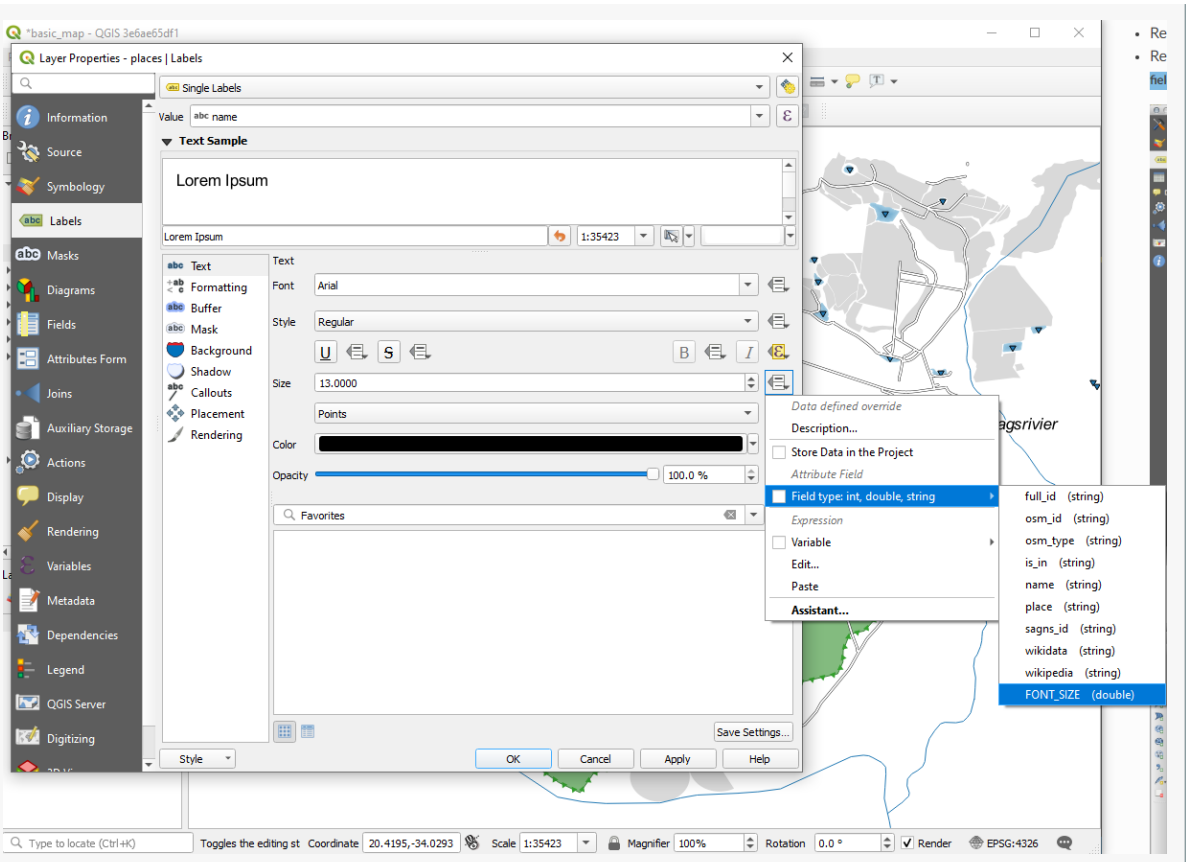
1. `place` の属性テーブルを開く
2.  ボタンをクリックして、編集モードにします
3.  ボタンを使用して新しい列を追加します
4. このように構成します:



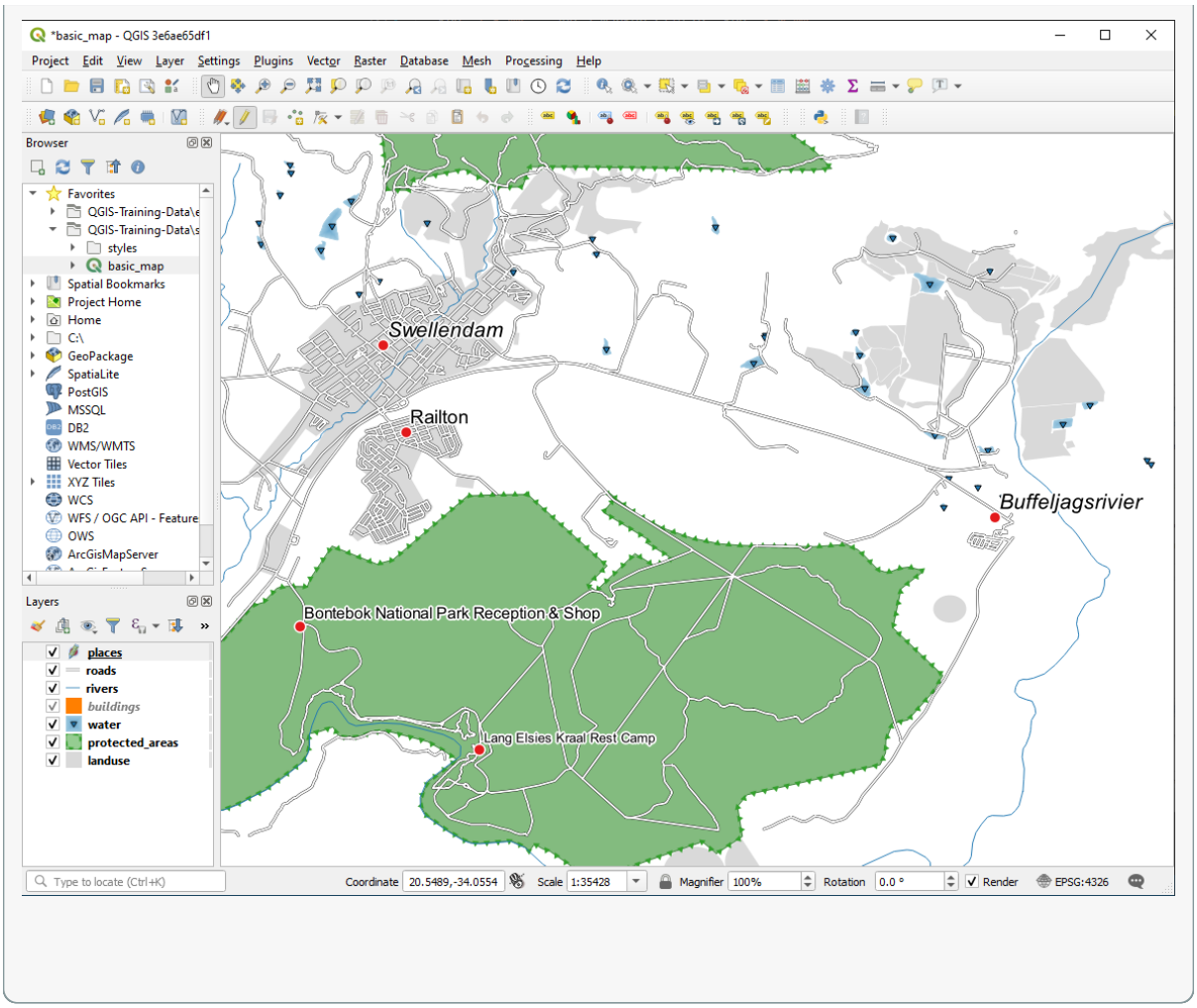
5. これを使用して、異なるタイプの場所（ `place` フィールドの各キー）ごとにカスタムフォントサイズを設定します

答え

1. 編集モードのまま、 `FONT_SIZE` の値を好きなものに設定します。この例では、町 (towns) は `16` 、郊外 (suburbs) は `14` 、地方 (localities) は `12` 、集落 (hamlets) は `10` を使用しています。
2. 変更を保存して編集モードを終了することを忘れないでください
3. `places` レイヤの **テキスト** フォーマットオプションに戻り、フォントサイズ  データ定義オーバーライドドロップダウンの **属性フィールド** で `FONT_SIZE` を選びます：



上記の値を使用した場合、結果はこのようになるはずです：



3.2.8 ★★★ （上級レベル） ラベル付けのさらなる可能性

このコースですべてのオプションを網羅することはできませんが、**ラベル** タブには他にも多くの有用な機能があることを知っておいてください。スケールベース・レンダリングの設定、レイヤ内のラベルのレンダリング優先度の変更、レイヤ属性を使ったあらゆるラベルオプションの設定などが可能です。また、ラベルの回転や XY 位置などのプロパティを設定し（属性フィールドがある場合）、メインの **レイヤーラベルオプション** ボタンに隣接するツールでこれらの属性を編集することも可能です：



(これらのツールは必要な属性フィールドが存在して、編集モードの場合にアクティブになります。)

ラベル付けのシステムの可能性をもっと探ってみて下さい。

3.2.9 結論

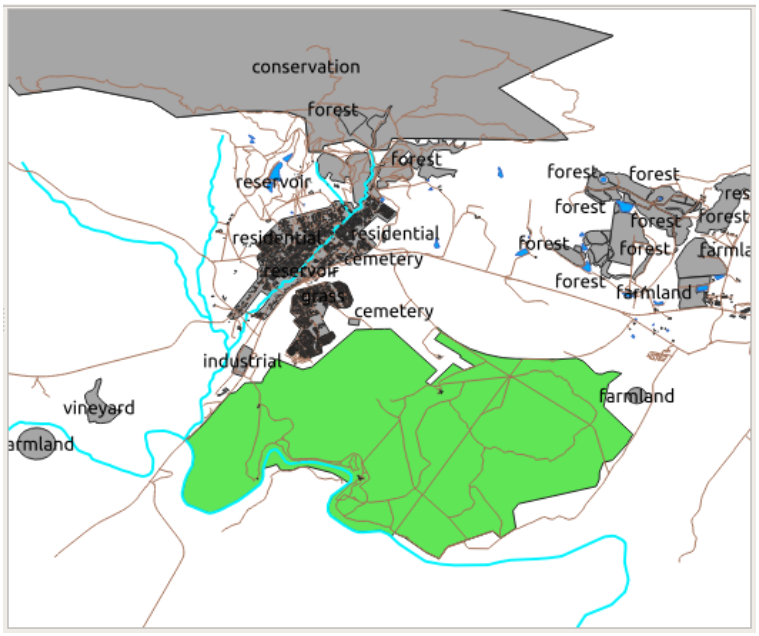
動的なラベル作成のためにレイヤの属性を使用する方法を学びました。これによりあなたの地図の情報量を増やし、地図をスタイリッシュにすることができます!

3.2.10 次は?

属性によって地図に視覚的な違いを生じさせる方法がわかりました。オブジェクトそれ自体のシンボロジを変更するのに属性を使用することはどうでしょうか? 次のレッスンのトピックです!

3.3 レッスン: 分類

ラベルは、個々の地名などの情報を伝えるのに適していますが、何にでも使えるわけではありません。例えば、それぞれの 土地利用 の領域が何に使われているのか知りたい人がいるとします。ラベルを使うと、こうなります：

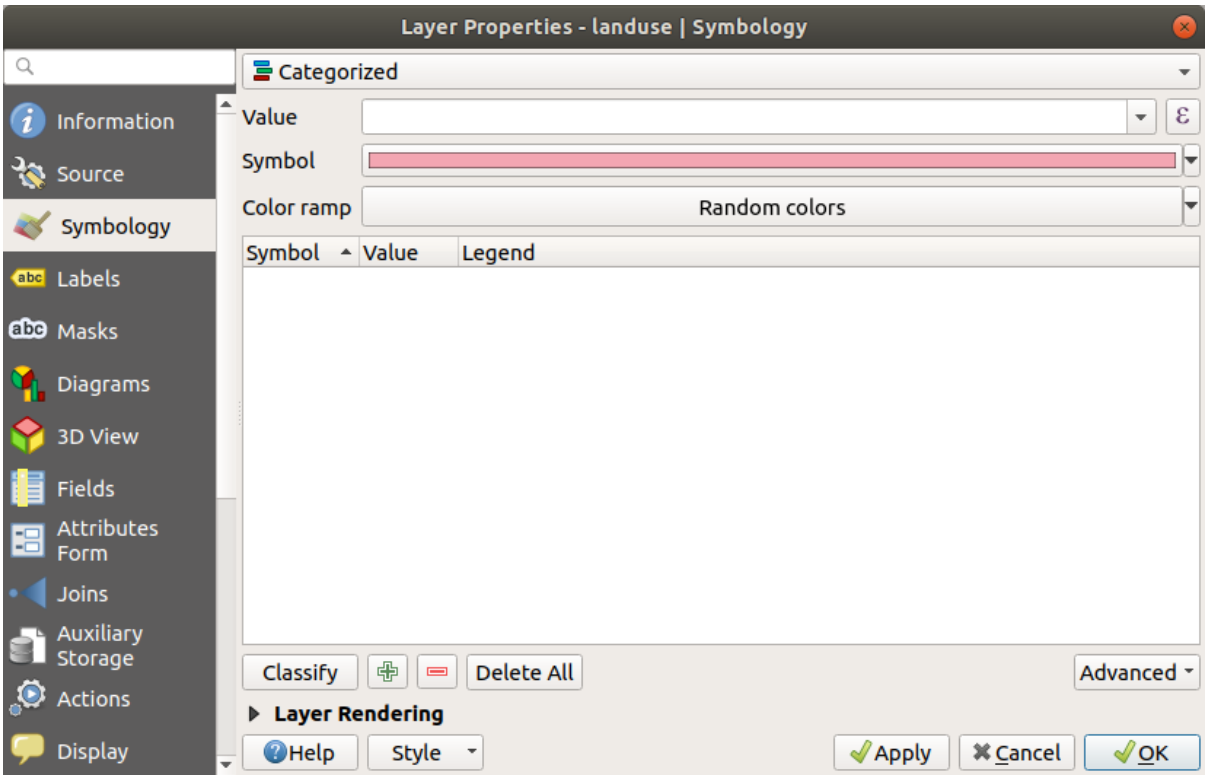


これでは地図のラベルが読みにくくなり、地図上に多数の異なる土地利用の領域がある場合は過剰にさえなります。

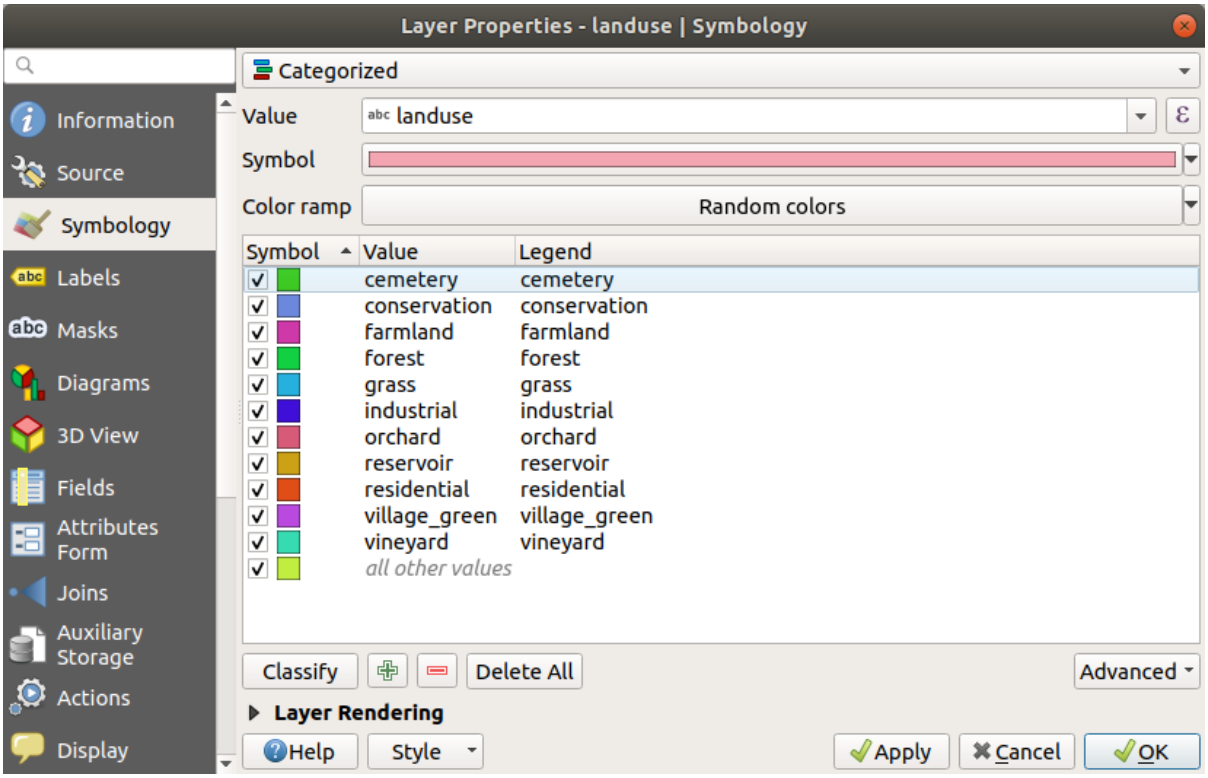
このレッスンの目標： 効果的にベクタデータを分類する方法を学習します。

3.3.1 ★☆☆ （初級レベル） 理解しよう: 名義データを分類する

1. landuse レイヤの **レイヤプロパティ** ダイアログを開きます
2. **シンボロジ** タブをクリックします
3. **単一定義** と言うドロップダウンをクリックし、それを **カテゴリ値による定義** に変更します：

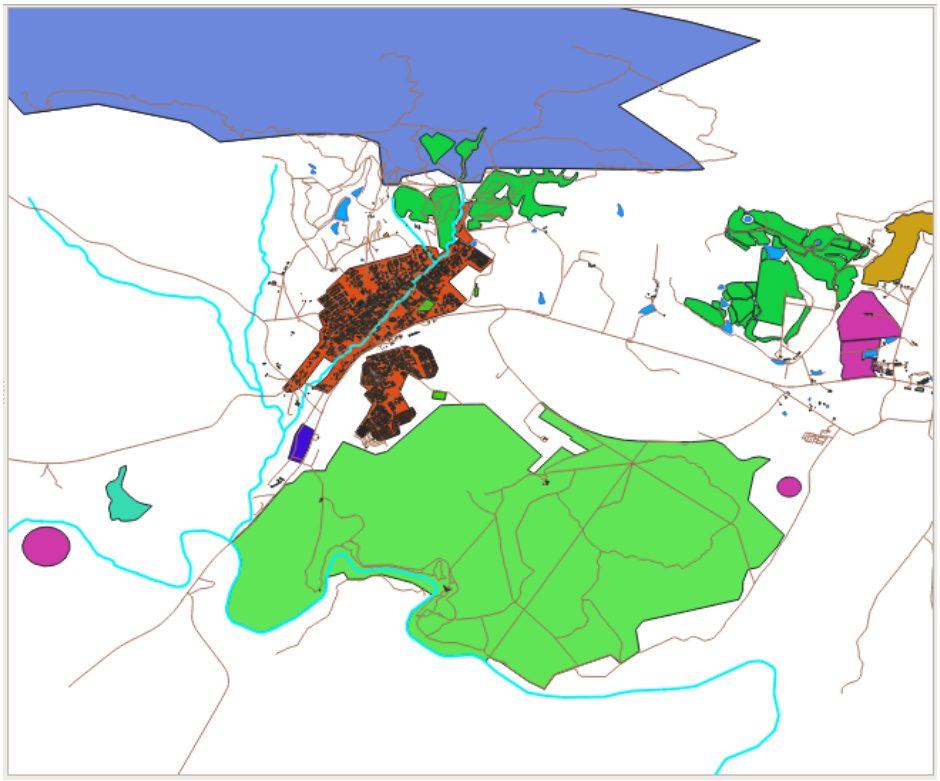


4. 新しいパネルで、 **値** を landuse に、 **カラーランプ** を *Random colors* に変更します
5. **分類** ラベルの付いたボタンをクリックします

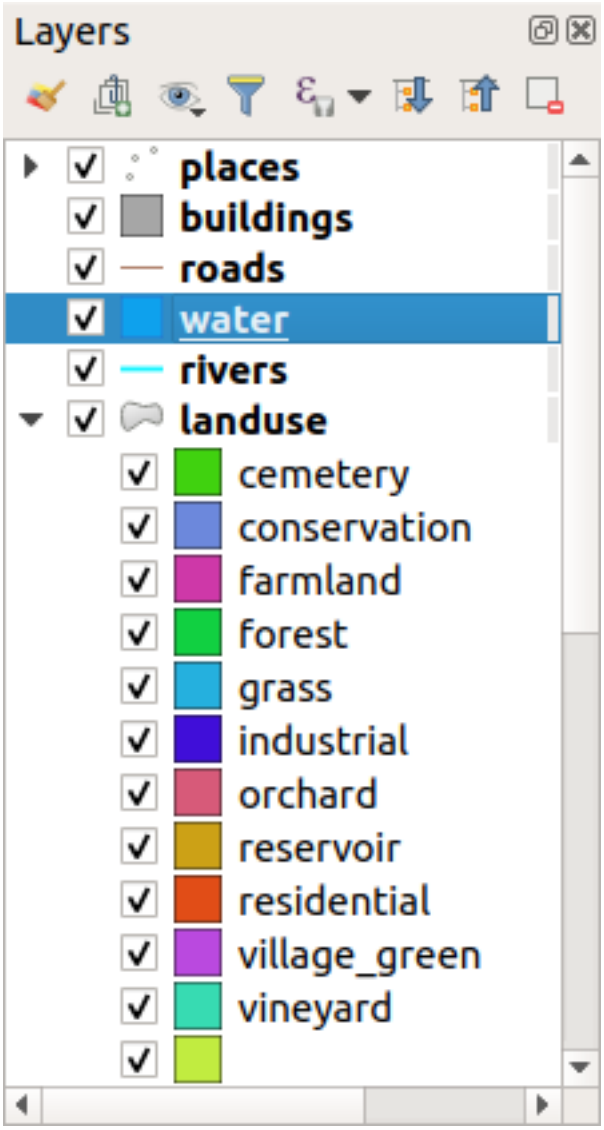


6. *OK* をクリックします。

   このように表示されるはずです:

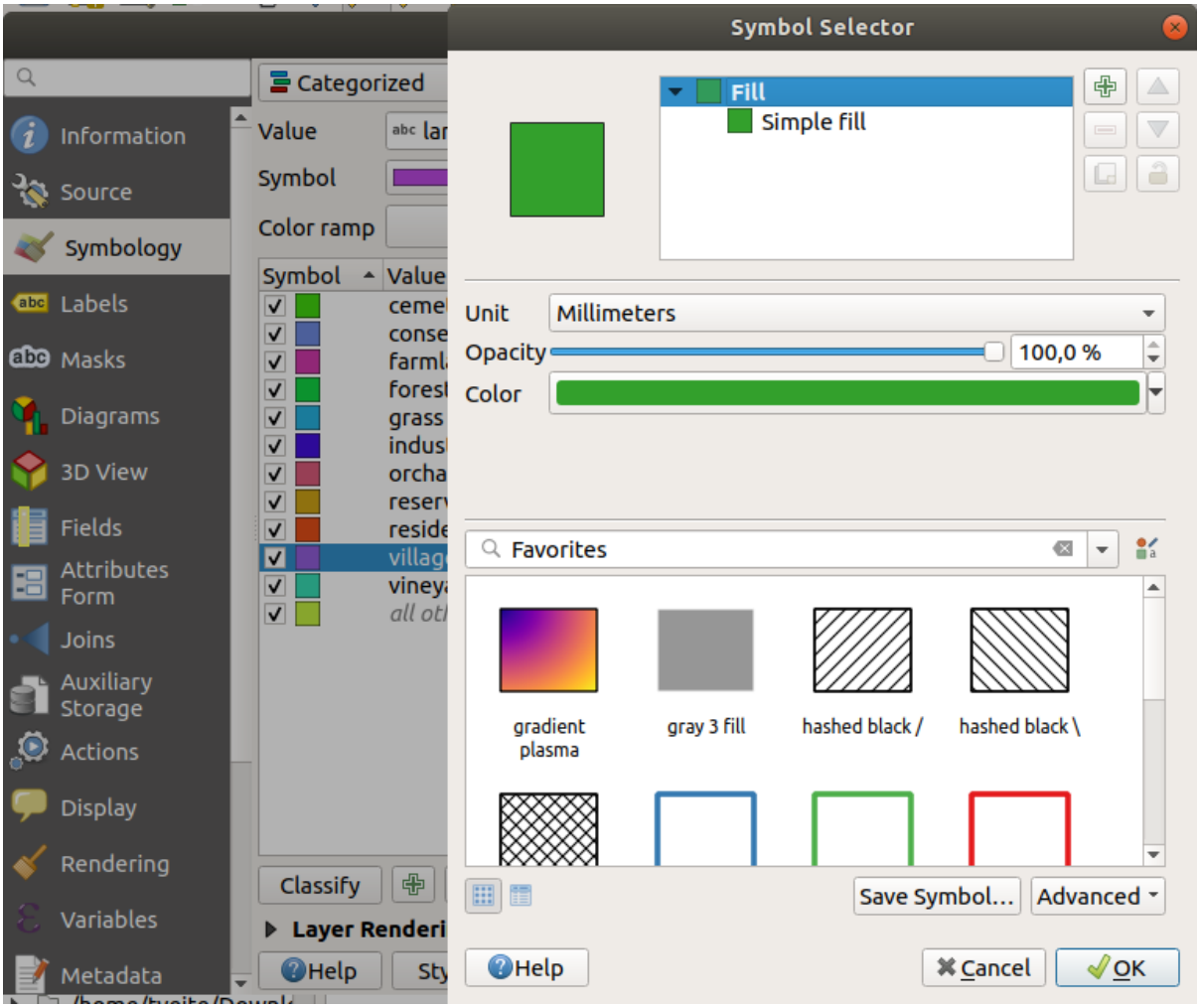


7. **レイヤ** パネルの landuse の横にある矢印（またはプラス記号）をクリックすると、分類の説明が表示されます:

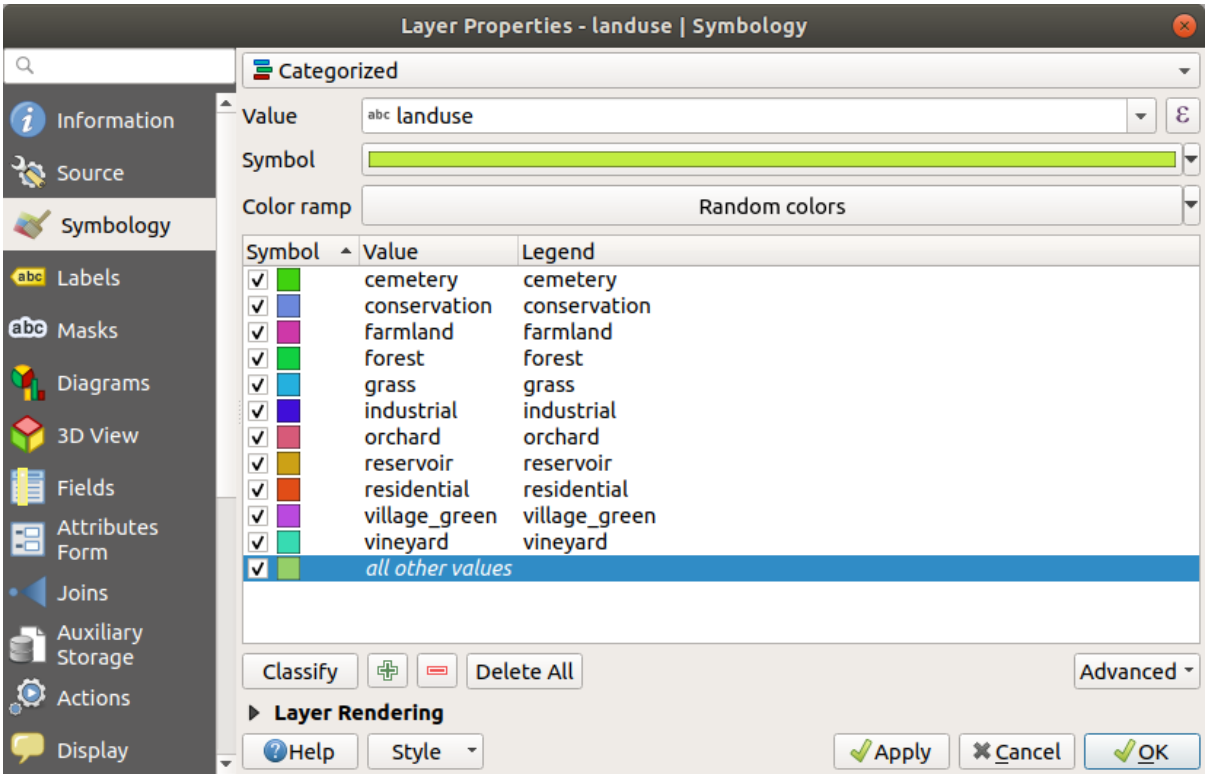


これで土地利用ポリゴンに色が付き、同じ土地利用をするエリアは同じ色になるように分類されました。

8. 必要であれば、**レイヤ** パネルまたは **レイヤプロパティ** ダイアログで該当するカラーブロックをダブルクリックして、各土地利用区域のシンボルを変更することができます:



空の分類が一つあることに注意してください：



この空の分類は、landuse 値が定義されていない、あるいは *NULL* 値を持つオブジェクトに色を付けるために使用されます。この空の分類は、 *NULL* 値を持つ区域が地図上で表現されるようにするために有用です。空白または *NULL* 値をより明確に表現するために色を変更することもできます。

すべての苦労して稼いだ変更を失わないように、今、地図を保存することを忘れないでください！

3.3.2 ★☆☆ （初級レベル） 自分でやってみよう: その他の分類

上記で得た知識を使って `buildings` レイヤを分類します。 `building` フィールドに対して分類を設定し、*Spectral* カラーランプを使用します。

> **注釈**
>
> 忘れずに、都市エリアを拡大して結果を確認してください。

3.3.3 ★★☆ （中級レベル） 理解しよう: 比率分類

分類には 4 つのタイプがあります： **名義** 、 **順序** 、 **間隔** 、 **比率** 。

名義 分類では、対象を分類するカテゴリは名前ベースであり、順序はありません。例えば、町名、地区コードなど。名義データに使用する記号は、順序や大きさを意味するものであってはなりません。

- ポイントには、さまざまな形の記号を使うことができます。

- ポリゴンには、異なるタイプのハッチングや異なる色（明るい色と暗い色の混在を避ける）を使用することができます。
- ラインには、異なるダッシュパターン、異なる色（明るい色と暗い色の混在を避ける）、線に沿った異なる記号を使用することができます。

順序 分類では、カテゴリは一定の順序で並べられます。例えば、世界の都市は、世界貿易、旅行、文化などに対する重要性に応じてランク付けされます。順序データに使用する記号は、順序を意味するものでなければなりませんが、大きさを意味するものであってはなりません。

- ポイントには、明るい色から暗い色の記号を使うことができます。
- ポリゴンには、段階的な色（明るい色から暗い色）を使うことができます。
- ラインには、段階的な色（明るい色から暗い色）を使うことができます。

間隔 分類では、数値は正、負、ゼロの値を持つ目盛りになっています。例：海抜の高さ/海抜の低さ、気温（摂氏）。区間データに使用する記号は、順序と大きさを意味するものでなければなりません。

- ポイントには、さまざまな大きさ（小さいものから大きいものまで）の記号を使うことができます。
- ポリゴンには、段階的な色（明るい色から暗い色）を使ったり、さまざまな大きさの図を追加することができます。
- ラインには、太さ（細いから太い）を使うことができます。

比率 分類では、数値は正とゼロの値だけを持つスケールになっています。例：絶対零度（0 度ケルビン）以上の温度、ある地点からの距離、ある通りの 1 ヶ月の平均交通量など。比率データに用いる記号は、順序や大きさを意味するものでなければなりません。

- ポイントには、さまざまな大きさ（小さいものから大きいものまで）の記号を使うことができます。
- ポリゴンには、段階的な色（明るい色から暗い色）を使ったり、さまざまな大きさの図を追加することができます。
- ラインには、太さ（細いから太い）を使うことができます。

上記の例では、名義分類を使用して、 `landuse`（土地利用）レイヤの各レコードを `landuse` 属性に基づいて色付けしました。今度は、比率分類を使って、レコードを面積で分類してみます。

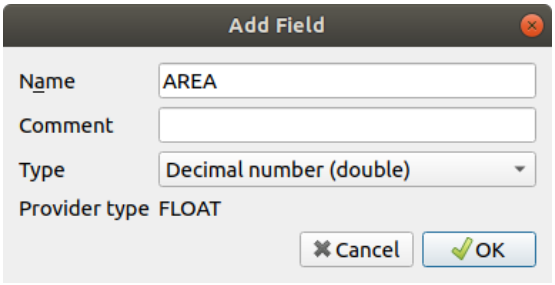
レイヤを再分類するので、既存の分類は保存しないと失われます。現在の分類を保存するには:

1. レイヤのプロパティダイアログを開きます
2. **スタイル** ドロップダウン メニューの **スタイルを保存** ... ボタンをクリックします。
3. **現在の名前を変更**... を選び、 `land usage` を入力し、 *OK* を押します。

   カテゴリとその記号がレイヤのプロパティに保存されました。
4. **スタイル** ドロップダウンメニューの **追加…** エントリをクリックし、`ratio` という名前の新しいスタイルを作成します。これで、新しい分類が保存されます。
5. **レイヤプロパティ** ダイアログを閉じます

土地利用区域を大きさで分類したいのですが、問題があります：大きさのフィールドがないので、作らなければなりません。

1. landuse レイヤの属性テーブルを開きます。
2.  トグル編集 ボタンをクリックして編集モードにします
3.  新規フィールド ボタンを使って、 AREA という倍精度の新しい列を追加します。



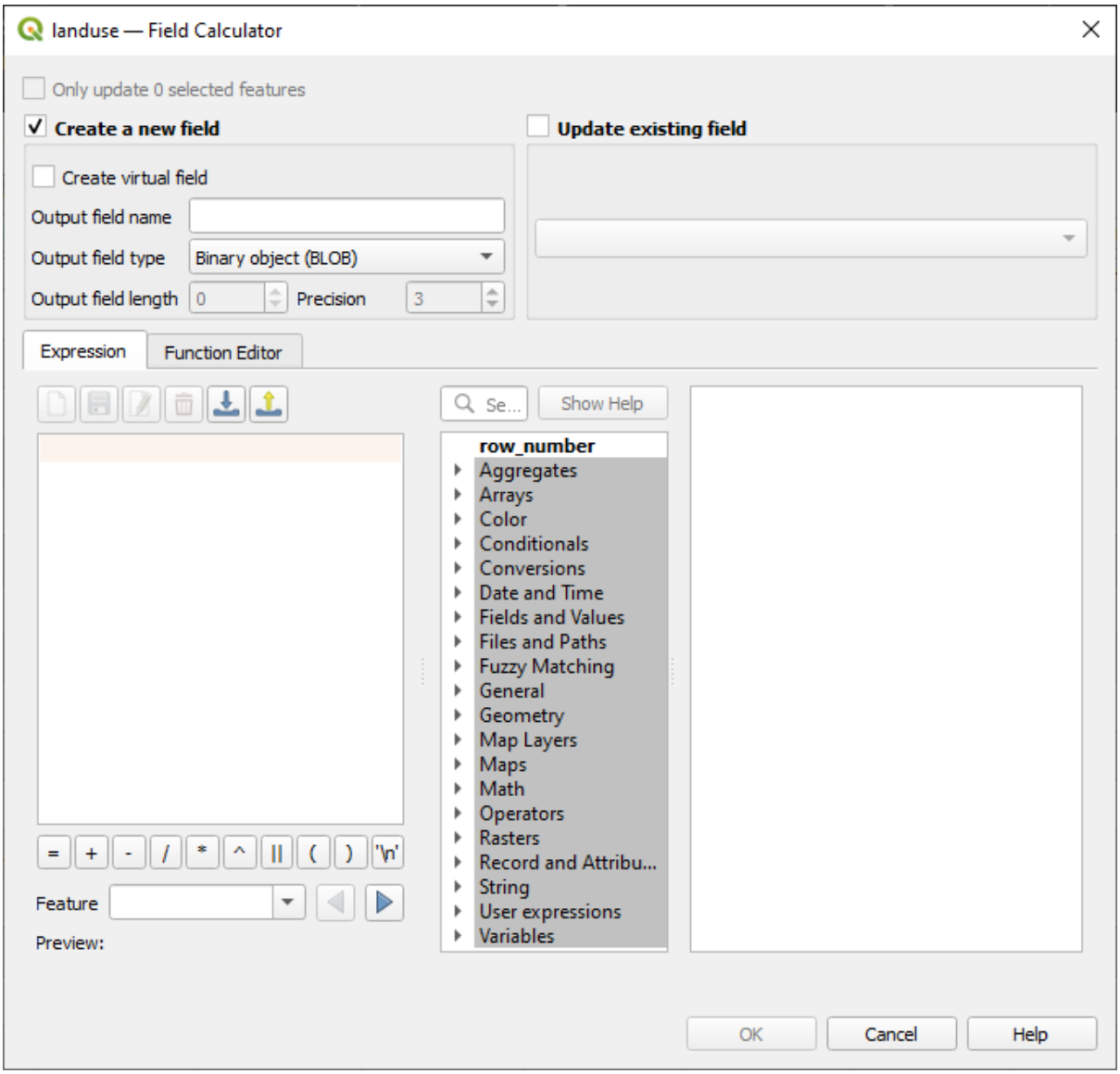
4. *OK* をクリックします。

   新しいフィールドが追加されます (テーブルの右端; 見るには水平方向にスクロールする必要があるかもしれません)。しかし、現時点では、このフィールドには何も入力されておらず、たくさんの *NULL* 値があります。

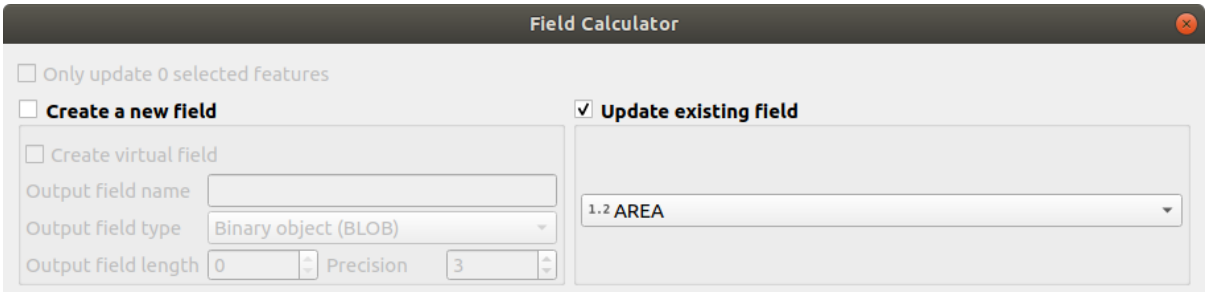
   この問題を解決するために、面積を計算する必要があります。

   1.  ボタンでフィールド計算機を開きます。

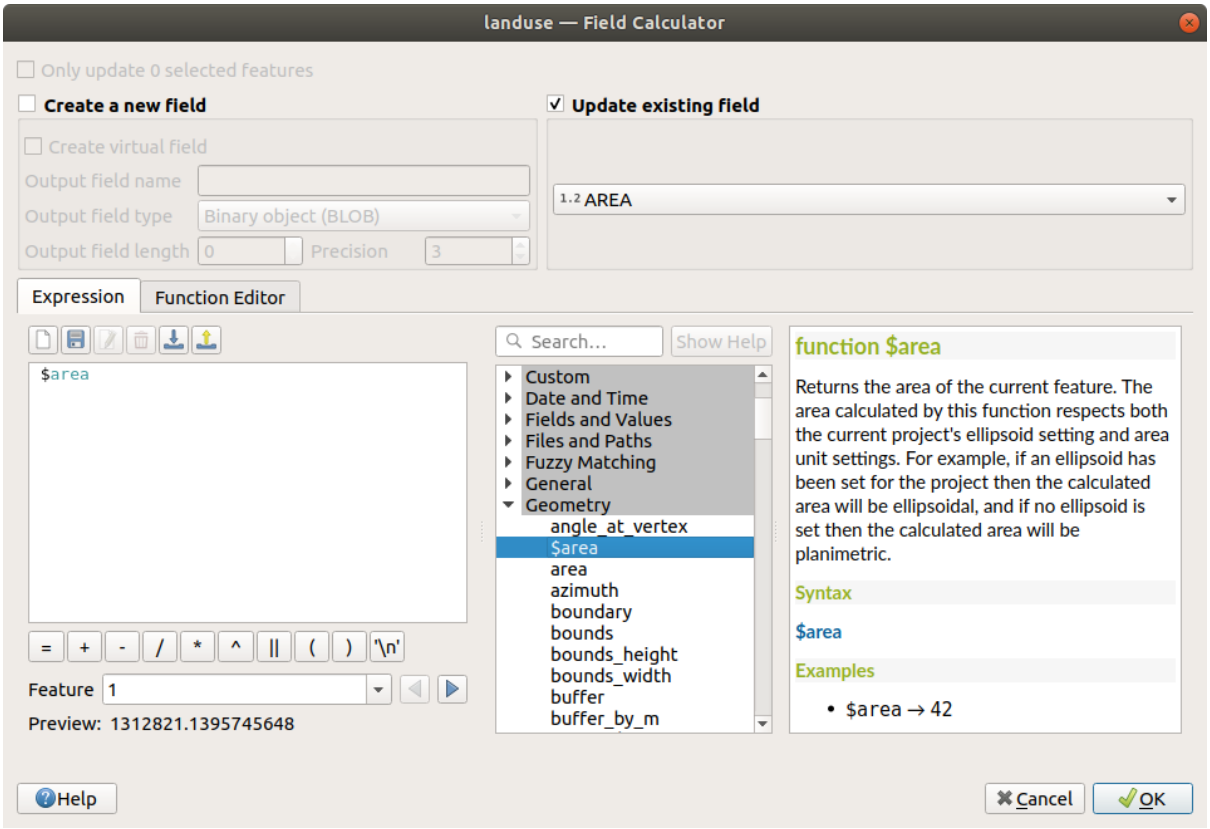
      このようなダイアログが表示されます：



2. **既存のフィールドを更新する** をチェックします
3. フィールドのドロップダウンメニューから、 *AREA* を選びます



4. **式** タブで、リスト内の **ジオメトリ** 関数グループを展開し、 *$area* を見つけます
5. それをダブルクリックして、 **式** フィールドに表示させます



6. *OK* をクリックします。

7. 属性テーブルの AREA フィールドまでスクロールすると、値が入力されていることがわかります（データを更新するために列のヘッダーをクリックする必要がある場合があります）。

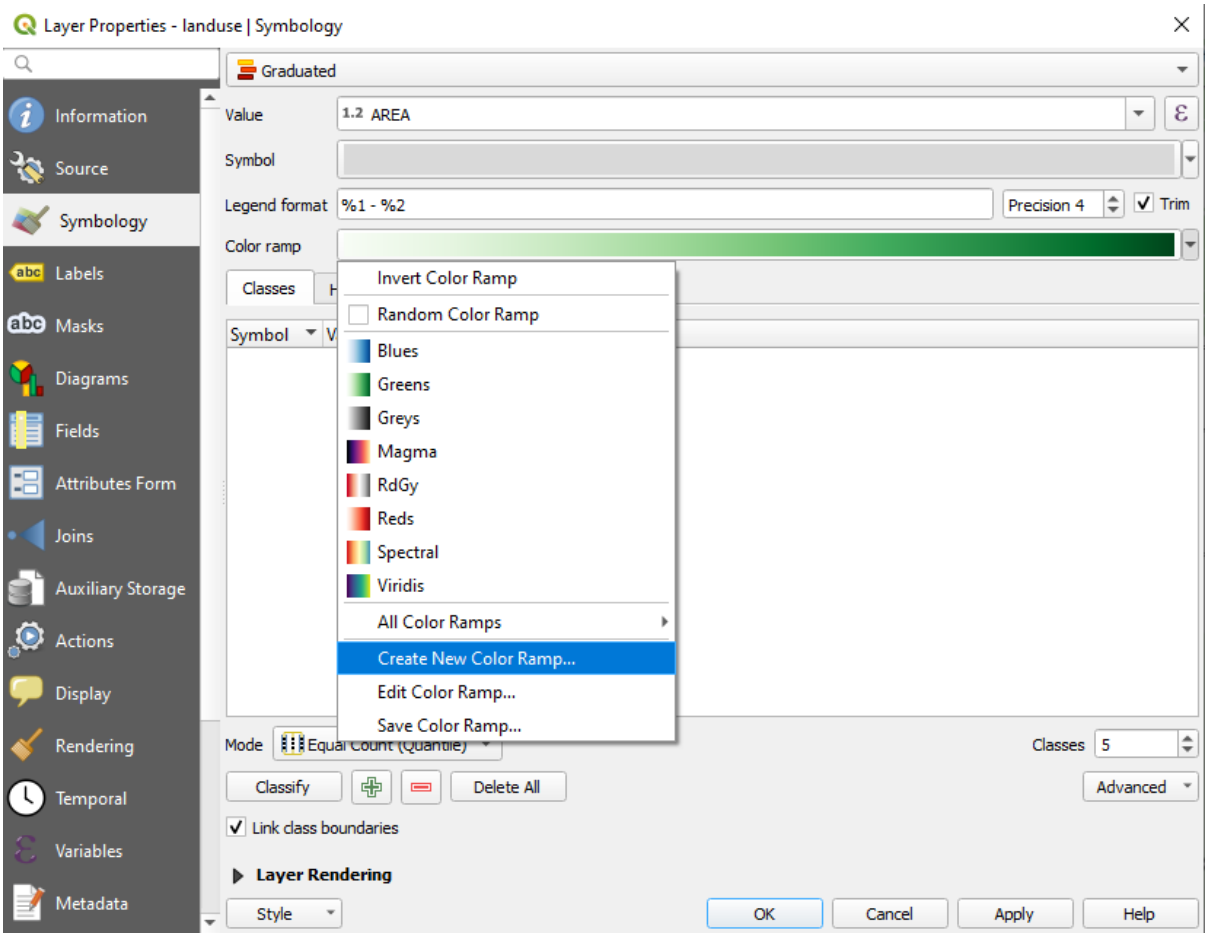
> **注釈**
>
> これらの面積はプロジェクトの面積単位の設定に従いますので、平方メートルまたは平方度の単位を使用できます。

5. を押して編集内容を保存し、 トグル編集 で編集モードを終了します

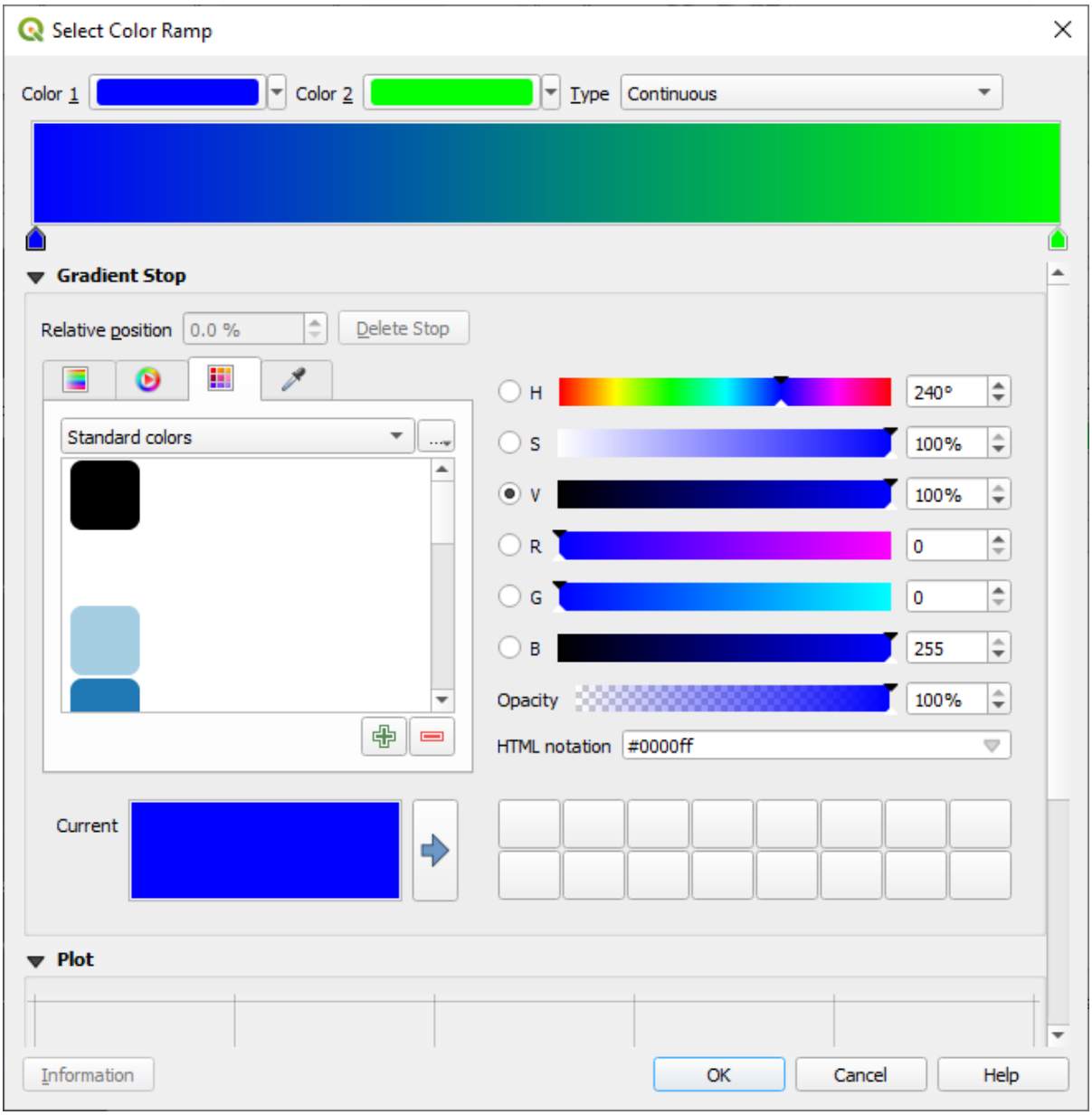
6. 属性テーブルを閉じます

これでデータが揃ったので、それを使って 土地利用 レイヤをレンダリングしてみましょう。

1. landuse レイヤの **レイヤプロパティ** ダイアログの **シンボロジ** タブを開きます
2. 分類スタイルを **カテゴリ値による定義** から **連続値による定義** に変更します
3. **値** を AREA に変更します。
4. **カラーランプ** の下にある **カラーランプを新規作成…** オプションを選びます



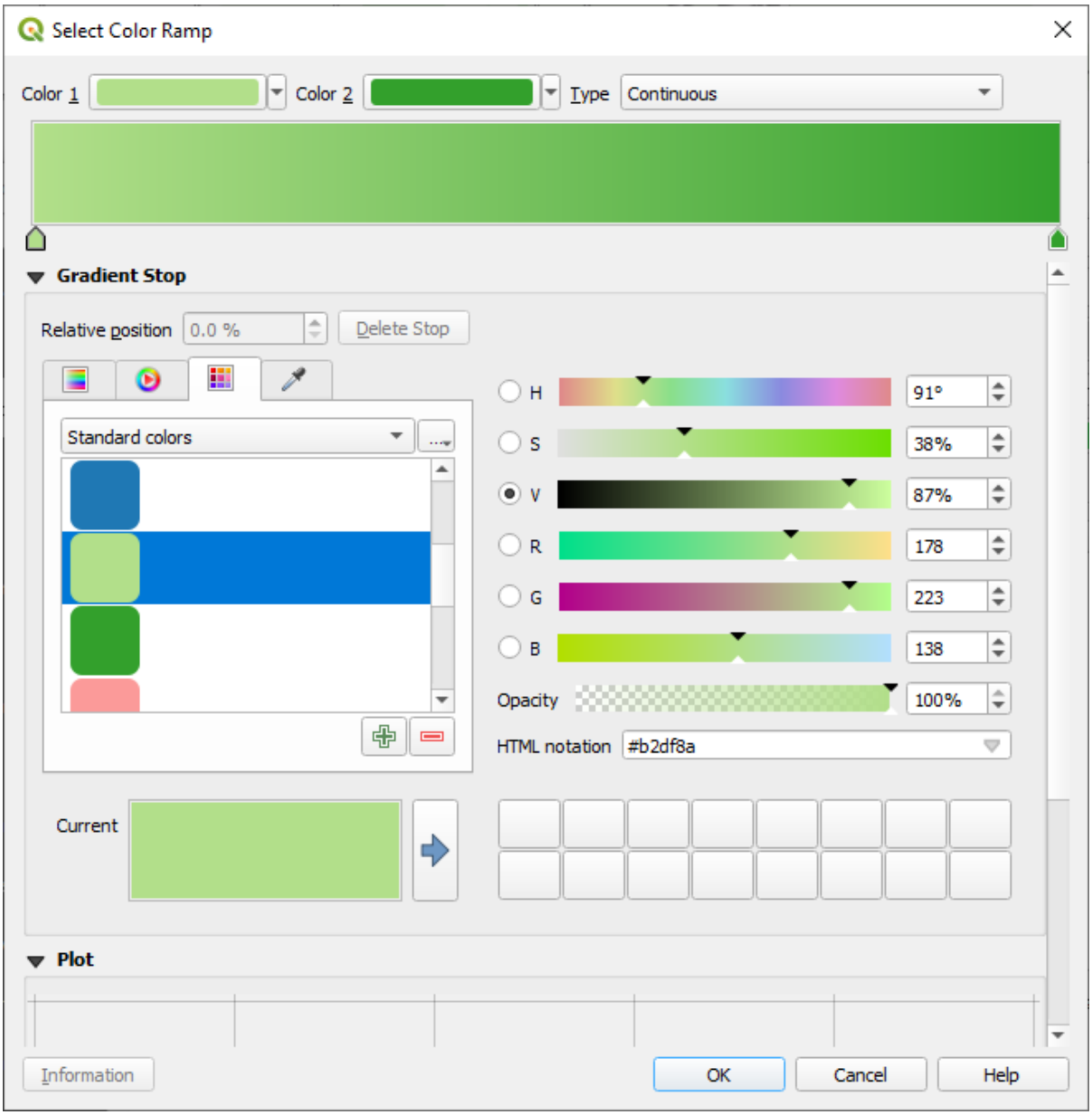
5. **勾配グリッド** を選択し（まだ選択されていない場合）、 *OK* をクリックします。このように表示されます：



これを使用して小さな地域は **色** *1*、大きな地域は **色** *2* で表示します。

6. 適切な色を選びます

   例では、結果は次のようになります。



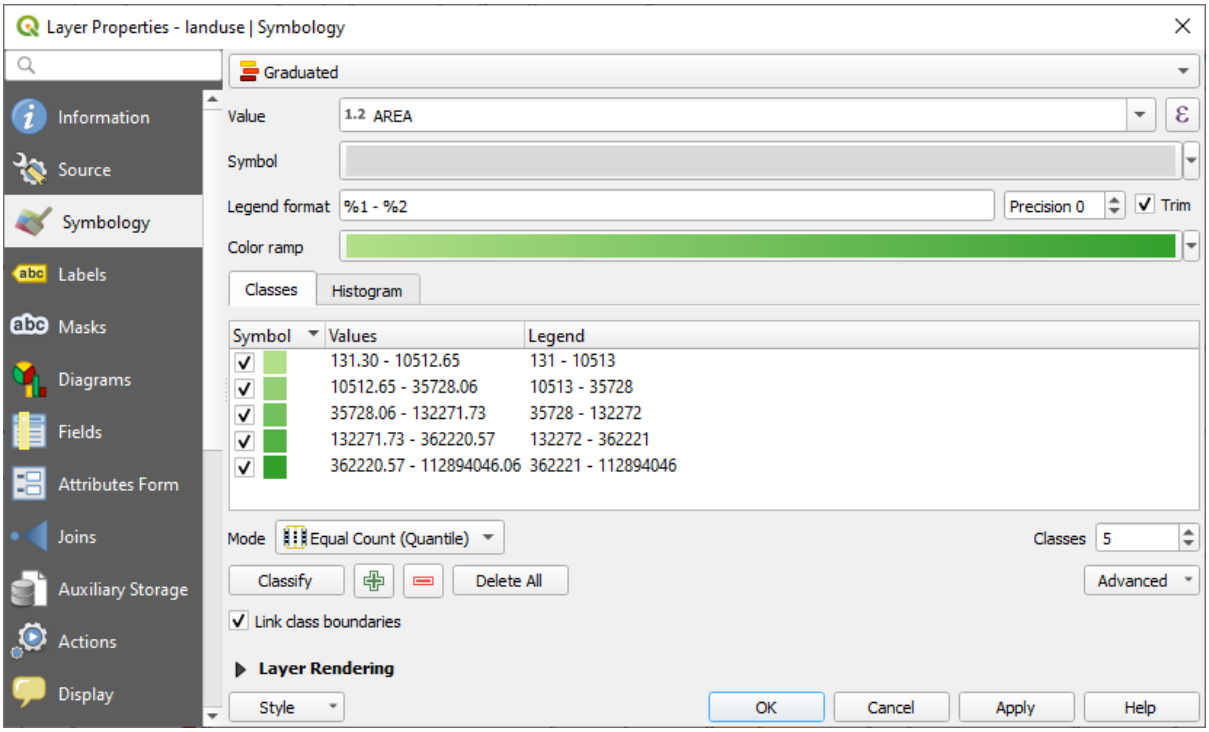
7. *OK* をクリックします。

8. **カラーランプ** タブの **カラーランプを保存**... を選択すると、カラーランプを保存することができます。カラーランプに適切な名前を選び、 **保存** をクリックします。これで、同じカラーランプを **全カラーランプ** で簡単に選択できるようになります。

9. **モード** の下から  棟梁分類 (Quantile) を選びます。

10. **分類** をクリックします。

    これで、次のようなものができあがります:



そのまま他のすべてを残します。

11. *OK* をクリックします:

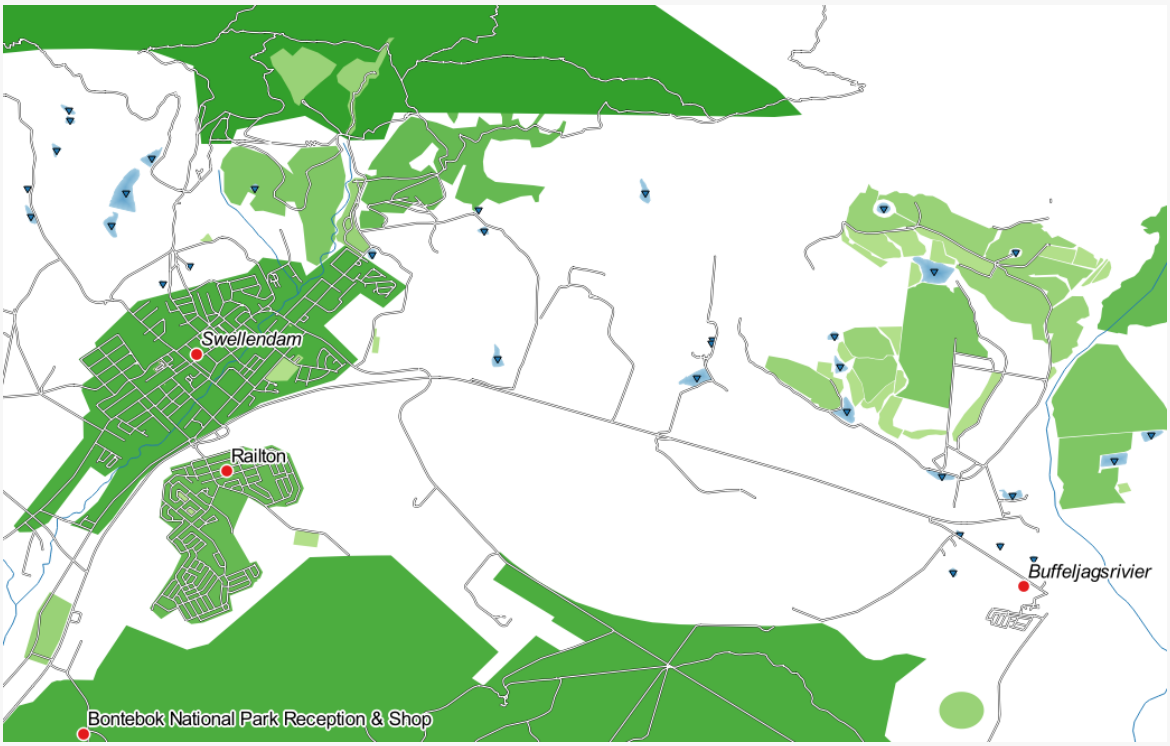


3.3.4 ★★☆ （中級レベル） 自分でやってみよう: 分類の絞り込み

- 理にかなった分類を得るまで **モード** と **クラス** の値を変更します。

答え

使用した設定は同じではないかもしれませんが、 *Classes* = 6 と *Mode* = *Natural Breaks (Jenks)*（そしてもちろん同じ色を使用）では、地図は次のようになります：



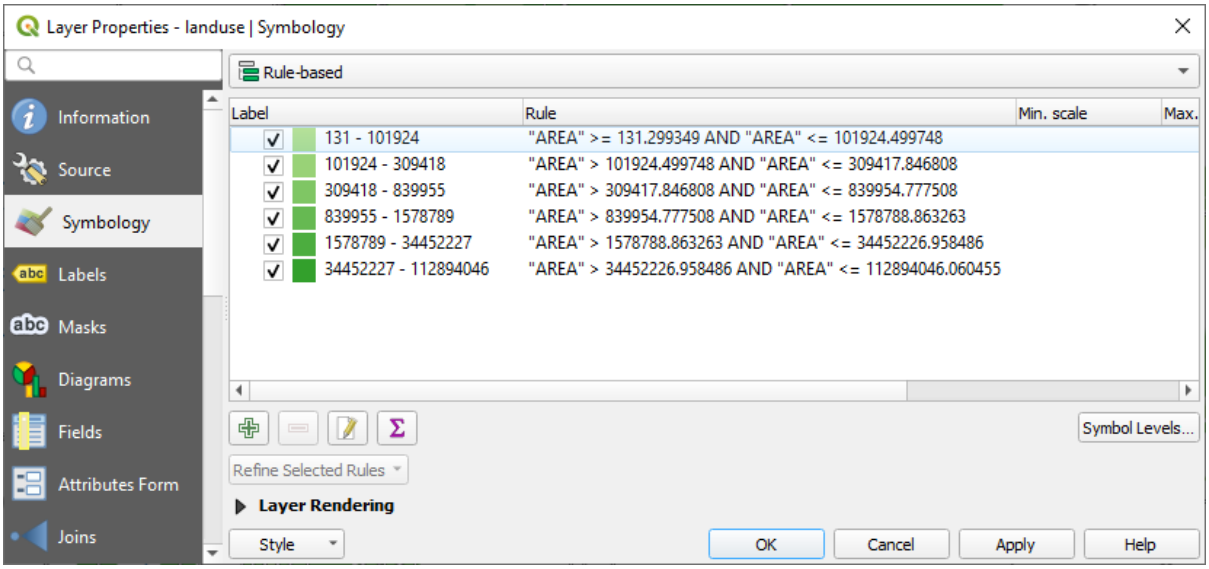
3.3.5 ★★★ （上級レベル） 理解しよう: 規則に基づく分類

分類のための複数の条件を組み合わせると便利ですが、残念ながら通常の分類では 1 つの属性だけを考慮に入れます。このときルールによる分類が有用になります。

このレッスンでは、Swellendam 市をその他の住宅地、そしてその他の種類の土地利用（面積に基づく）から簡単に識別できるように、 `landuse` レイヤを表現してみます。

1. `landuse` レイヤの **レイヤプロパティ** ダイアログを開きます
2. **シンボロジ** タブに切り替えます
3. 分類スタイルを **ルールによる定義** に切り替えます

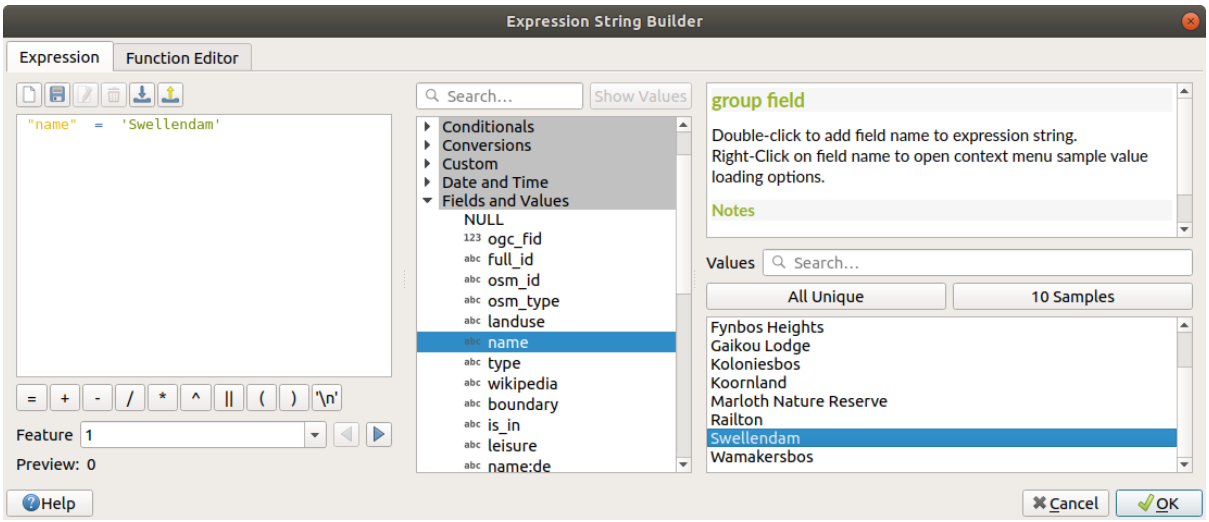
   QGIS はこのレイヤに現在実装されている分類を表すルールを自動的に表示します。例えば、上記の演習を終えた後、以下のように表示されるかもしれません：



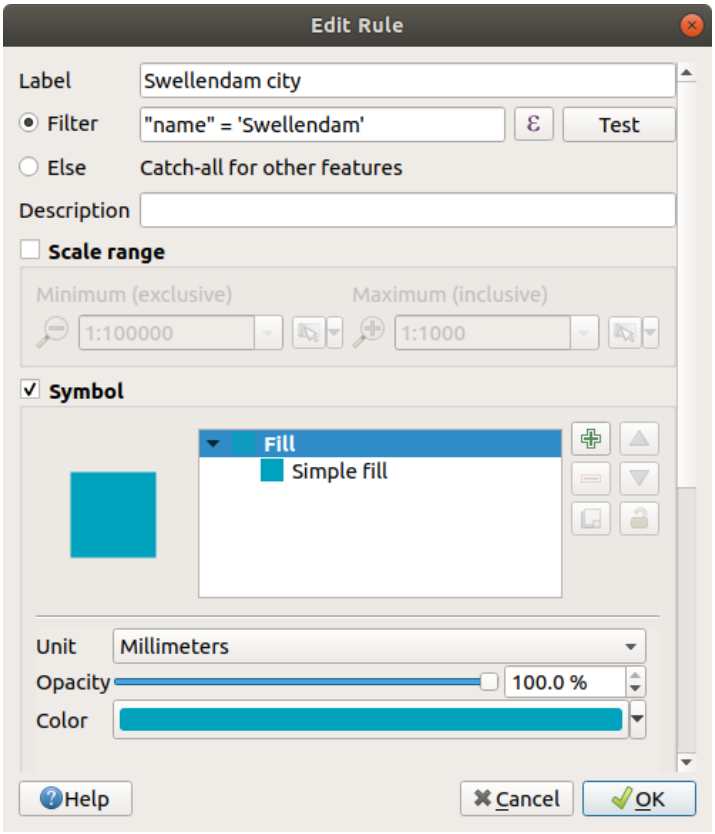
4. 全てのルールを選択するようクリックしドラッグします
5. 選択したルールを削除 ボタンを使って、既存のルールをすべて削除します

では、カスタムルールを追加してみましょう。

1. ルールを追加 ボタンをクリックします
2. *Edit rule* ダイアログが現れます
3. ラベルに `Swellendam city` を入力します
4. **フィルタ** テキストエリアの隣にある ε ボタンをクリックして **式文字列ビルダ** を開きます
5. 条件 `"name" = 'Swellendam'` を入力し検証します



6. **ルールの編集** ダイアログに戻り、その地域での町の重要性を示すために濃いグレーブルーの色を割り当て、境界を削除します。

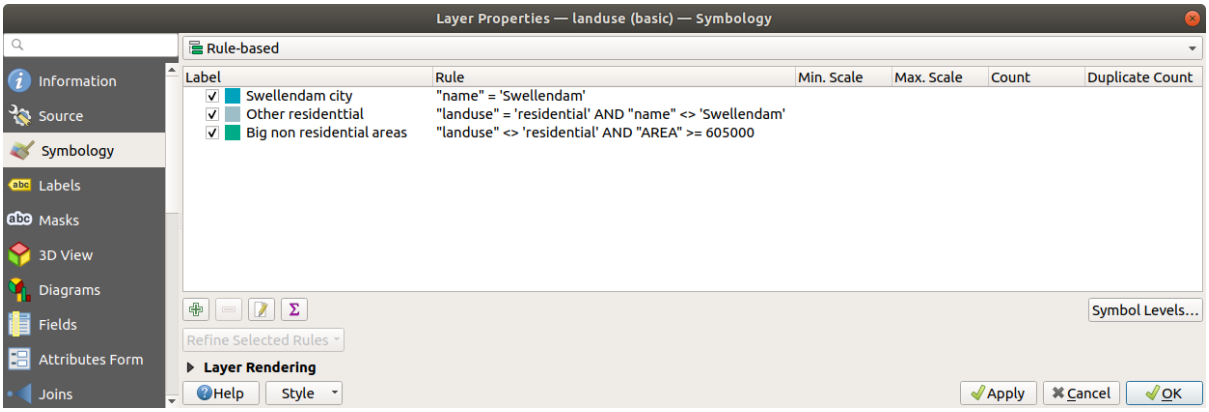


7. *OK* を押します

8. 次のルールを追加するために上記のステップを繰り返します：

   1. **Other residential** ラベルは、条件 `"landuse" = 'residentail' AND "name" <> 'Swellendam'`。淡いブルーグレーの **塗りつぶし** を選びます。

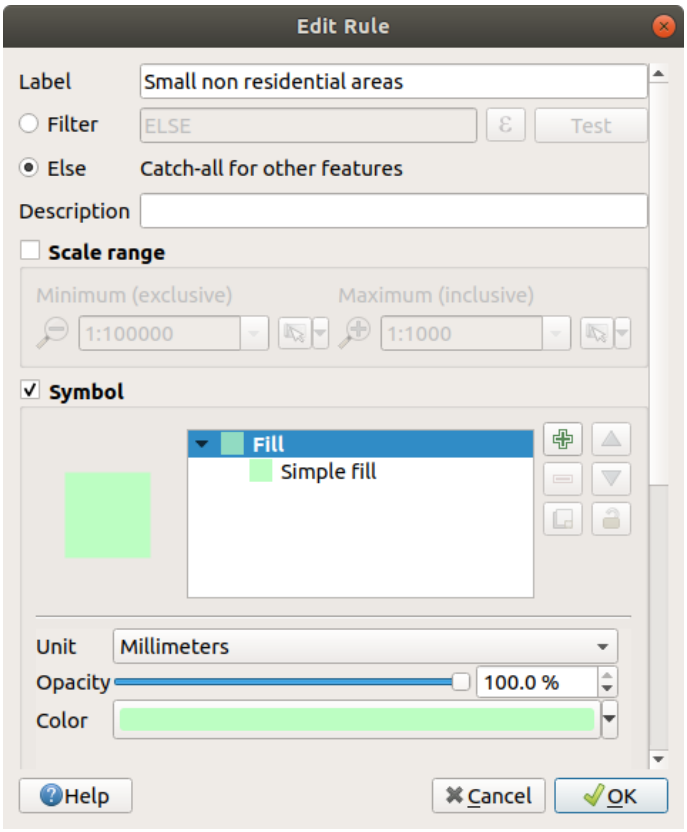
   2. **Big non residential areas** ラベルは、条件 `"landuse" <> 'residential' AND "AREA" >= 605000`。色は中緑色を選びます。



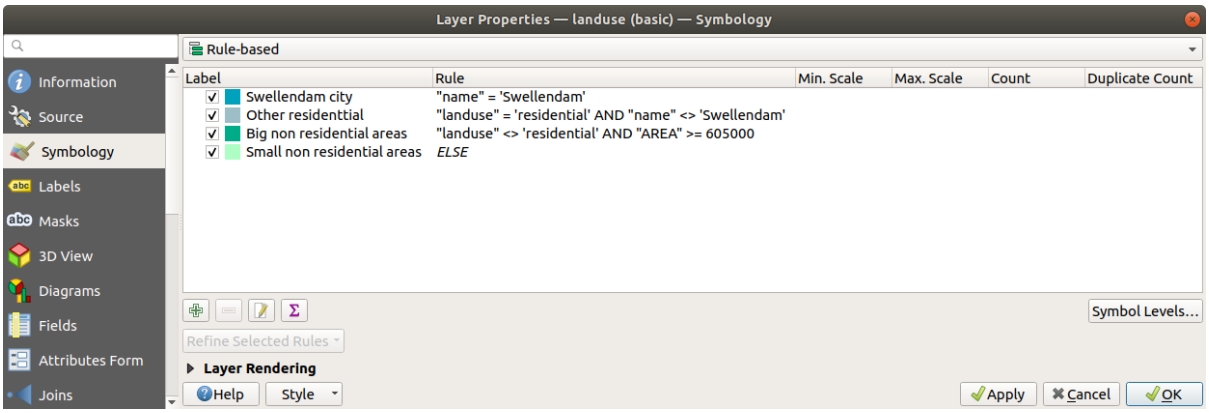
   これらのフィルターは排他的なものであり、地図上の地域を除外します（605000（平方メートル）より小さい非居住地域は、いずれのルールにも含まれません）。

3. 残りの地物は **Small non residential areas** というラベルの付いた新しいルールを使って捕えます。フィルター式の代わりに、 **もしくは** をチェックします。このカテゴリに適切な淡い緑

色を与えます。



ルールは次のようになります：



9. このシンボル体系を適用します

地図は次のようになります。



今、 最も顕著な住宅地 Swellendam とその他の非住宅地をその大きさに応じた色の地図を持っています。

3.3.6 結論

シンボロジにより、レイヤの属性を読みとりやすく表現できます。それは選択した任意の関連する属性を使用して、私たちだけでなく地図読者も地物の重要性を理解できます。直面している問題に応じて、それらを解決するために異なる分類技法を適用するでしょう。

3.3.7 次は?

これで見栄えの良い地図になりましたが、どのようにしてそれを QGIS からプリントアウトできる形式に、または画像や PDF にしようとしていますか？それは、次のレッスンの話題です！

第4章　モジュール: 地図をレイアウトする

本モジュールでは、QGIS の印刷レイアウトを使用して、必要な全ての地図構成要素を含んだ高品質な地図の作製方法を学びます。

4.1 レッスン: 印刷レイアウトを使う

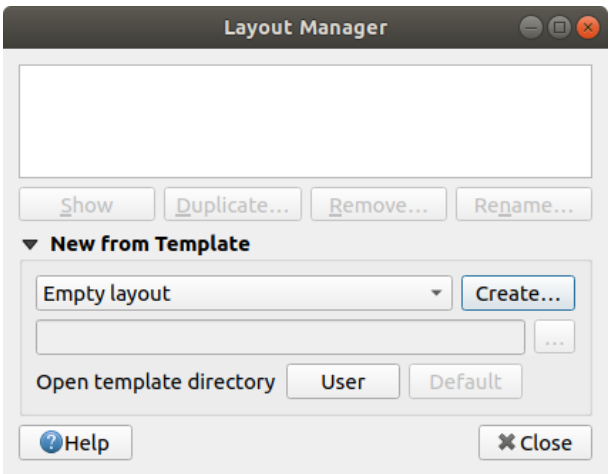
さて、地図を手に入れたら、それを印刷したり、文書に書き出したりする必要があります。なぜなら、GIS 地図ファイルは画像ではないからです。GIS 地図ファイルは画像ではなく、GIS プログラムの状態を保存したもので、全てのレイヤの参照、ラベル、色などが含まれています。そのため、同じデータや同じ GIS プログラム（QGIS など）を持っていない人にとっては、地図ファイルは役に立たないことになります。幸いなことに、QGIS は地図ファイルを誰でも読める形式にエクスポートすることができますし、プリンタを接続すれば地図をプリントアウトすることもできます。エクスポートも印刷も、**印刷レイアウト** で処理されます。

このレッスンの目標： QGIS **印刷レイアウト** を使い、必要な設定が全て行われた基本的な地図を作ること。

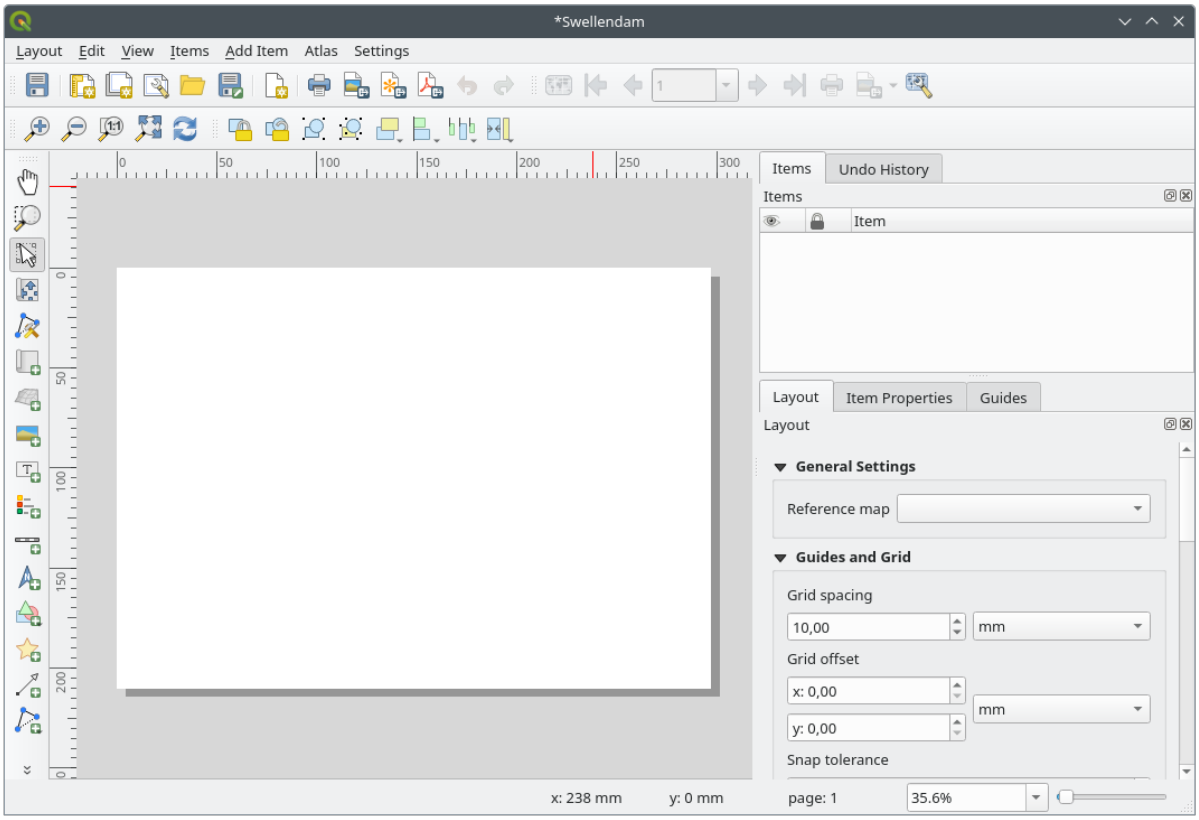
4.1.1 ★☆☆ （初級レベル） 理解しよう: レイアウトマネージャ

QGIS では、同じマップファイルを使って複数の地図を作成することができます。そのため、**レイアウトマネージャ** と呼ばれるツールが用意されています。

1. **プロジェクト　レイアウトマネージャ...** メニューをクリックして、ツールを起動します。すると、空の **レイアウトマネージャ** ダイアログが表示されます。

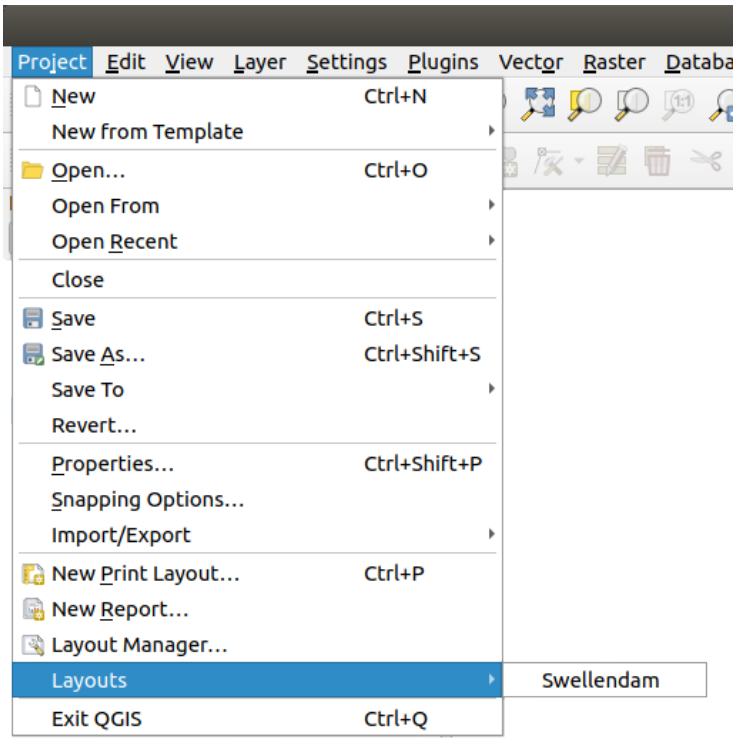


2. **テンプレートから新規作成** の下にある **空のレイアウト** を選び、 **作成**... ボタンを押します。
3. 新しいレイアウトに Swellendam という名前を付けて、 *OK* をクリックします。
4. これで、**印刷レイアウト** ウィンドウが表示されます:



また、この新しいレイアウトは **プロジェクト** **新規印刷レイアウト**... メニューで作成することもできます。

どちらの方法をとったとしても、新規印刷レイアウトは、下の画像のように **プロジェクト** **レイアウト** メニューからアクセスできるようになりました。



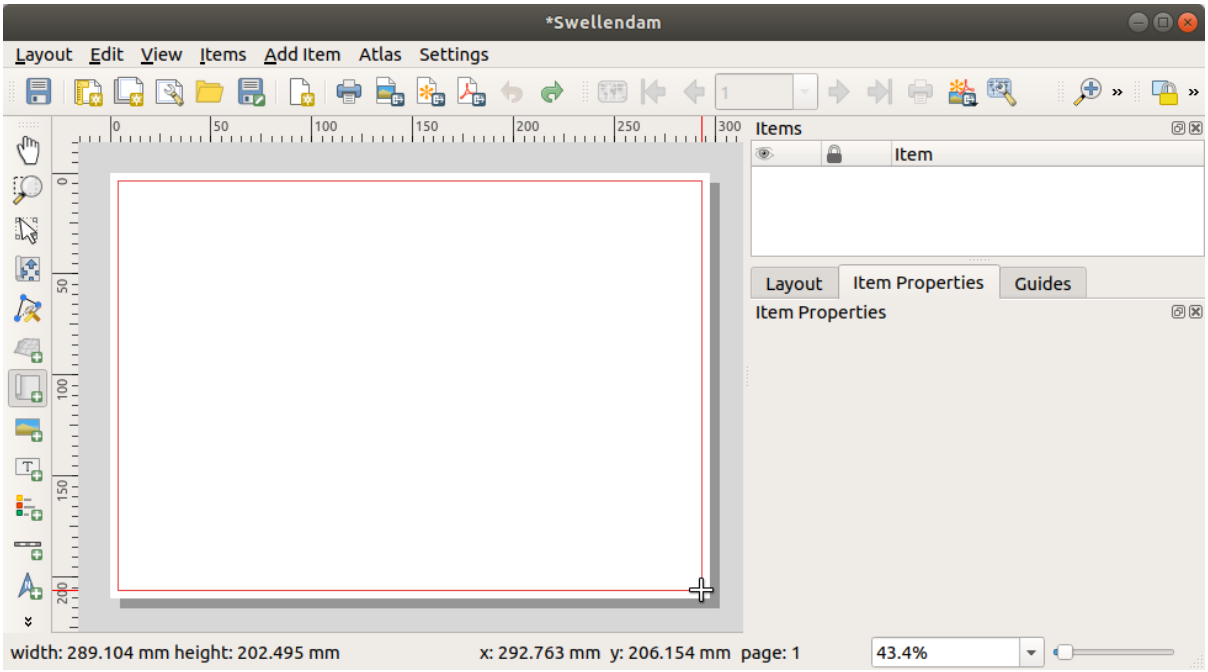
4.1.2 ★☆☆ （初級レベル） 理解しよう: 基本地図の構図

この例では、構図はすでに望んだ形になりました。あなたのものが同様であることを確認してください。

1. レイアウトウィンドウの中央でシートを右クリックし、コンテキストメニューから **ページのプロパティ**... を選択します。
2. **アイテムプロパティ** タブの値が、以下のように設定されていることを確認します:
   - **大きさ**: A4
   - **方向**: 横

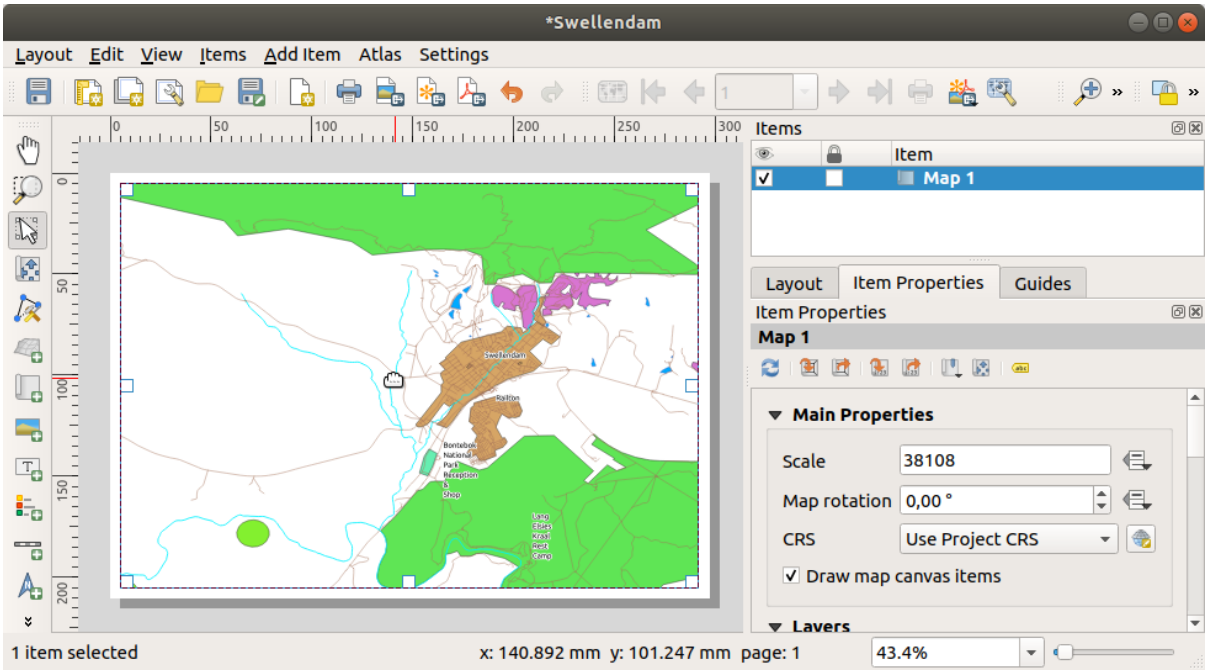
   今、ページレイアウトは望んでいたようになっていますが、このページはまだ空白です。これは明らかに地図を欠いています。その問題を修正しましょう！
3.  地図を追加 ボタンをクリックします。

   このツールを有効にすると、ページ上に地図を配置することができるようになります。
4. クリックして、空白のページにボックスをドラッグします。

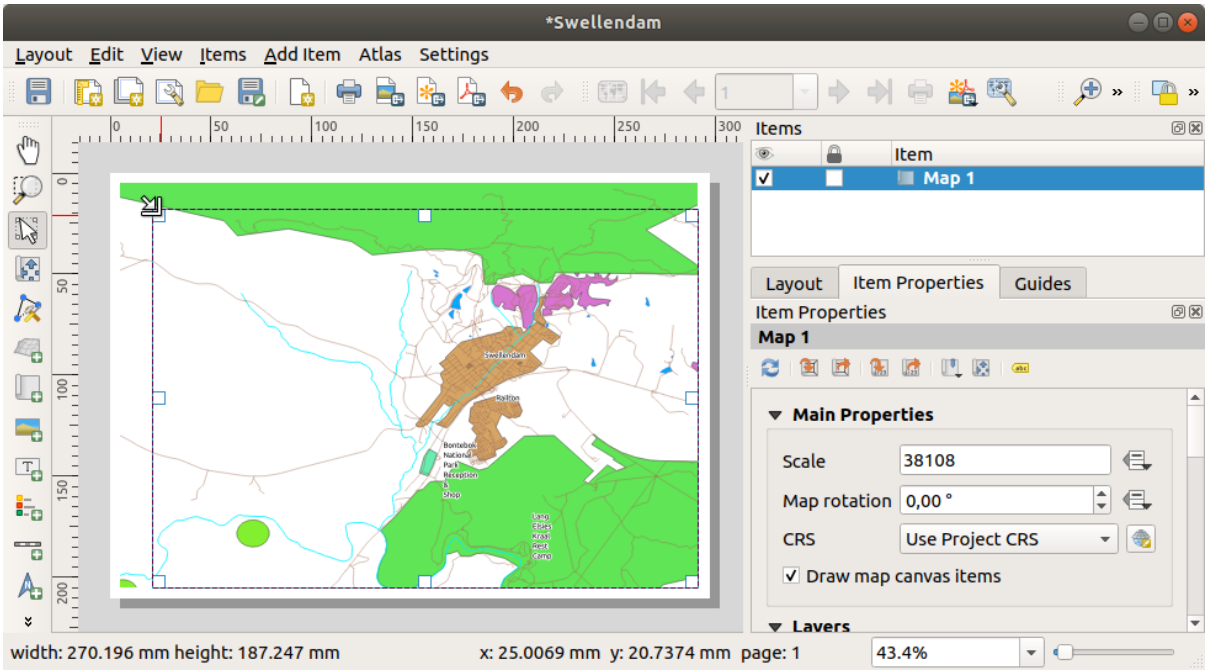


地図がページに表示されます。

5. クリックしてドラッグすることで地図を移動します。



6. 端にあるボックスをクリックしてドラッグすることで、サイズを変更できます:



注釈

地図はもちろん全く違って見えてもよいです！これは、自身のプロジェクトが設定されている方法によって異なります。しかし、心配しないように！これらの命令は一般的なので、地図自体がどのように見えるかに関係なく、同じ動作をします。

7. 縁に沿って余白、および上辺に沿ってタイトル用に空白を残すようにしてください。
8. これらのボタンを使って（地図でなく！）ページ上で拡大、縮小します：

9. QGIS のメイン ウィンドウでマップをズームおよびパンします。また、  アイテムのコンテンツを移動 ツールを使用してマップをパンすることができます。

   拡大または縮小に応じて地図ビューは更新されます

10. 何らかの理由で地図ビューが正しく更新されない場合、  ビューを更新 ボタンをクリックすることで強制的にマップを更新することができます。

    地図の大きさと位置は、最終的なものでなくてもかまいません。もし満足できなければ、いつでも後で戻って変更することができます。とりあえず、このマップに関する作業を確実に保存しておく必要があります。QGIS の **印刷レイアウト** はメイン マップ ファイルの一部であるため、プロジェクトを保存する必要があります。

11. **レイアウト**  **プロジェクトを保存** に移動します。これは、メインダイアログにある便利なショートカットです。

4.1.3 ★☆☆ （初級レベル） 理解しよう: タイトルを追加する

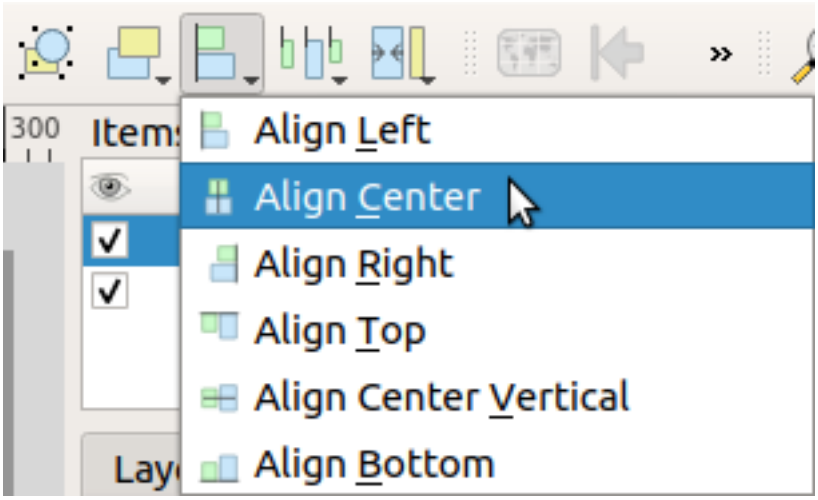
さて、地図はページ上で見栄えがしますが、読者/ユーザーはまだ何が起こっているのか知らされていません。読者やユーザーには文脈が必要です。地図要素を追加することで、読者やユーザーに文脈を提供することができます。まず、タイトルを追加してみましょう。

1.  ラベルを追加 ボタンをクリックします
2. 地図の上でページをクリックし、 **新規アイテムのプロパティ** ダイアログで提案された値を受け入れると、地図の上にラベルが表示されます。
3. サイズを変更し、ページの上部中央に配置します。それは地図の大きさを変更したり移動するのと同じ方法で大きさを変更したり移動できます。

   タイトルを移動すると、ページの中央にタイトルの配置を助けるガイドラインが現れることに気づくでしょう。

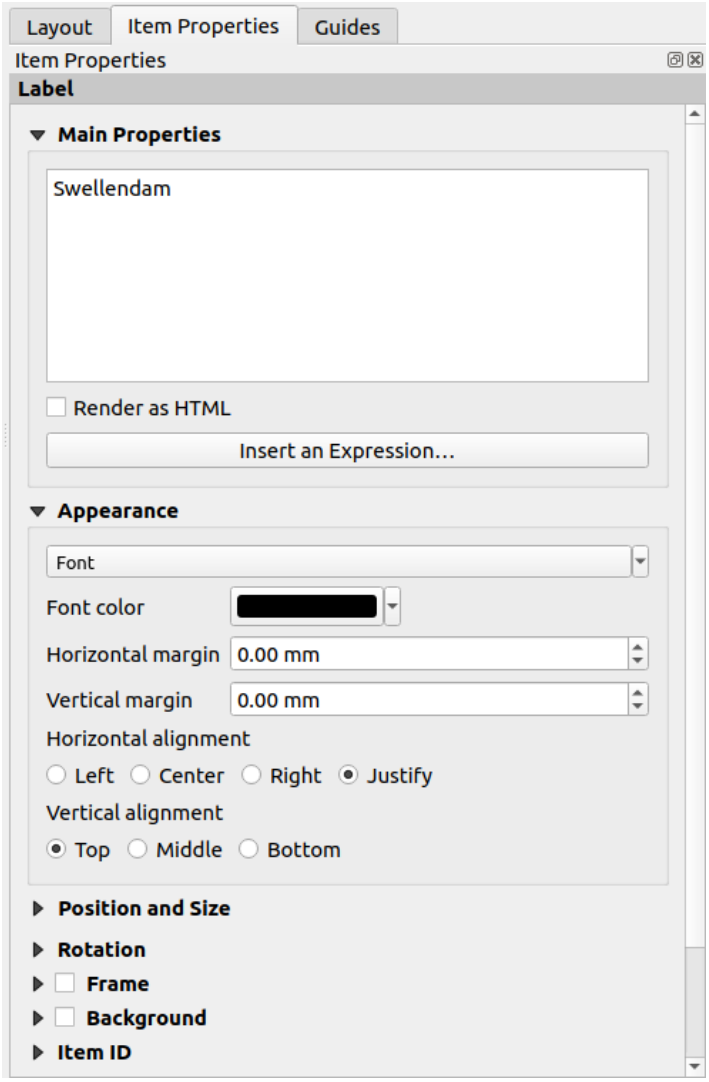
   ただし、アクションツールバーには、タイトルを（ページではなく）マップに相対的に配置するためのツールもあります:

   
4. マップをクリックしてそれを選びます
5. キーボードの `Shift` を押したまま、地図とラベルの両方が選択されるようにラベルをクリックしてください。
6.  選択を左寄せ整列する ボタンを探して、その横のドロップダウン矢印をクリックし、位置決めオプションを表示し、  **中央揃え** をクリックします:

   これで、ラベルの内容ではなくフレームが地図の中央に配置されました。ラベルの内容を中央に表示するには：

   1. クリックしてラベルを選択します。
   2. レイアウトウィンドウのサイドパネルにある **アイテムプロパティ** タブをクリックします。
   3. ラベルの文字を"Swellendam"に変更します:



4. このインターフェイスは **外観** セクションのフォントと配置のオプションを設定するために使用します：

    1. 大きめの、しかし実用的なフォントを選びます (この例では、サイズ 36 のデフォルトフォントを使用します)

    2. **水平方向配置** を **中央** に設定します。

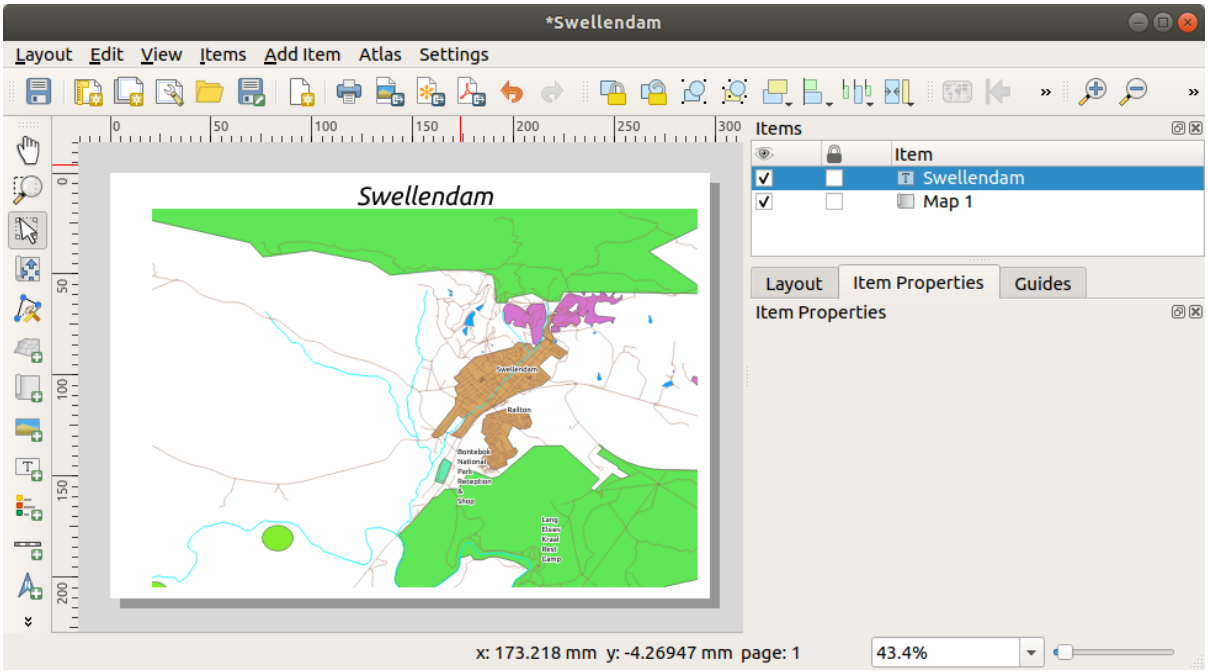
    フォントの色も変更できますが、デフォルトのとおりに黒のままにしておくのがおそらく最善です。

5. デフォルトの設定では、タイトルのテキストボックスにフレームを追加しません。フレームを追加したい場合は、そうすることができます：

    1. **アイテムプロパティ** タブで、**フレーム** オプションが表示されるまで、スクロールダウンします。

    2. **フレーム** チェックボックスをクリックしてフレームを有効にしてください。フレームの色や幅も変更できます。

この例ではフレームを有効にしていないので、これまでのところ私たちのページはこうなっています：



せっかく整列させたのに、間違って動かしてしまわないように、アイテムを固定することができます:

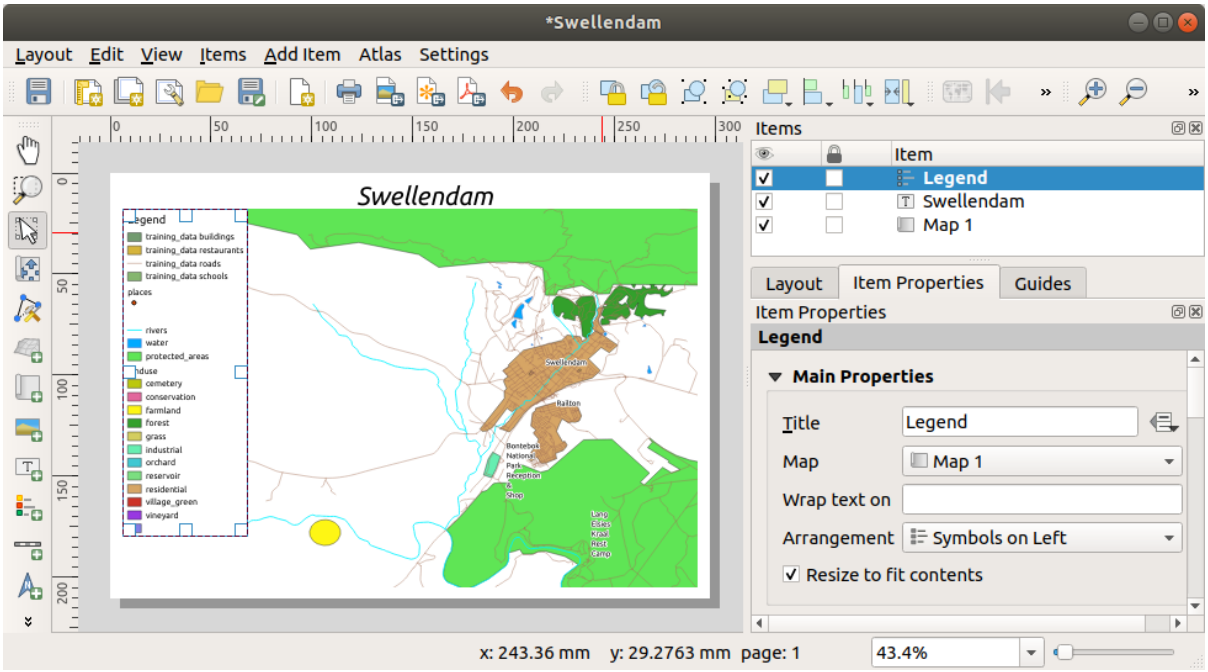
1. ラベルと地図の両方を選択します
2. **アクション** ツールバーの 選択アイテムを固定する ボタンをクリックします。

> **注釈**
>
> **アクション** ツールバーの すべてのアイテムの固定を解除 ボタンをクリックすると、アイテムの編集が再び可能になります。

4.1.4 ★☆☆ （初級レベル） 理解しよう: 凡例を追加する

また、地図の読者は、地図上のさまざまな事柄が実際に何を意味しているのかを理解する必要があります。地名のように一目瞭然の場合もあれば、森の色のように推測が難しい場合もあります。では新しい凡例を追加してみましょう。

1. 凡例を追加 ボタンをクリックします
2. 凡例を配置するページの上でクリックし、 **新規アイテムのプロパティ** ダイアログで提案された値を受け入れます。
3. レイアウトページに凡例が追加され、メインダイアログで設定されたレイヤシンボル体系が表示されます。
4. いつものように、アイテムをクリックして好きな場所に移動させることができます:



4.1.5 ★★☆ （中級レベル） 理解しよう: 凡例をカスタマイズする

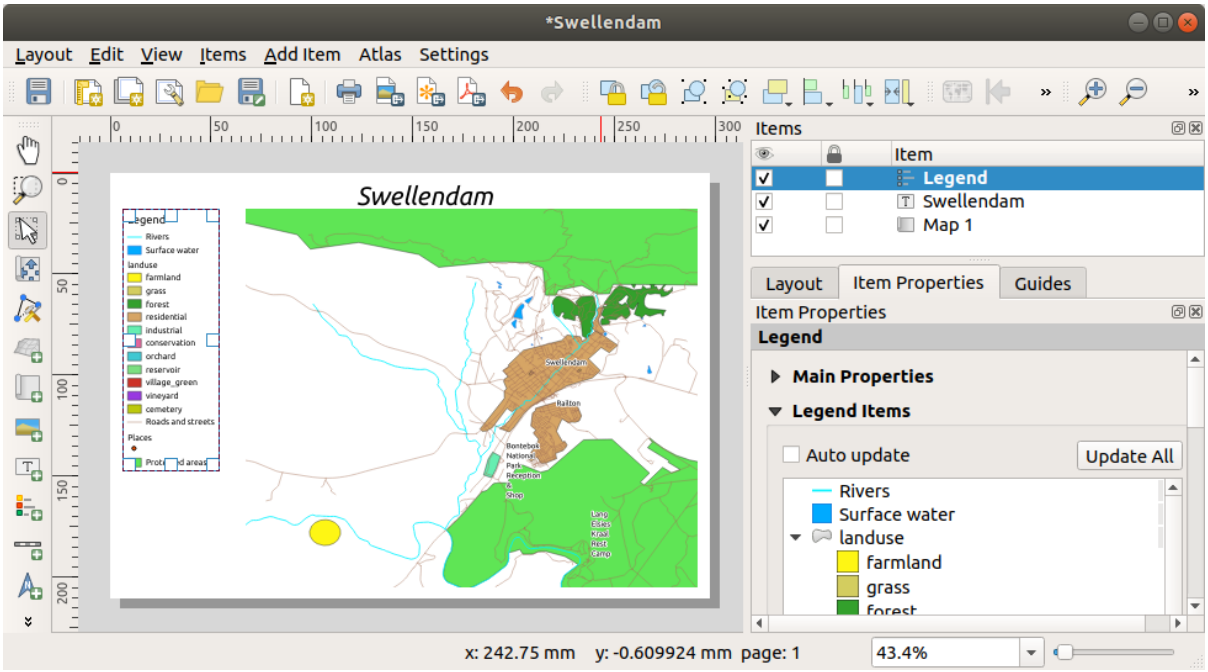
凡例上のすべてが必要ではありませんので、いくつかの不要な項目を削除しましょう。

1. **アイテムプロパティ** タブの中に、 **凡例アイテム** グループが表示されます。
2. **自動更新** ボックスをオフにすると、凡例の項目を直接変更できるようになります
3. *buildings* を持つエントリを選択します
4. ボタンをクリックして、それを凡例から削除します

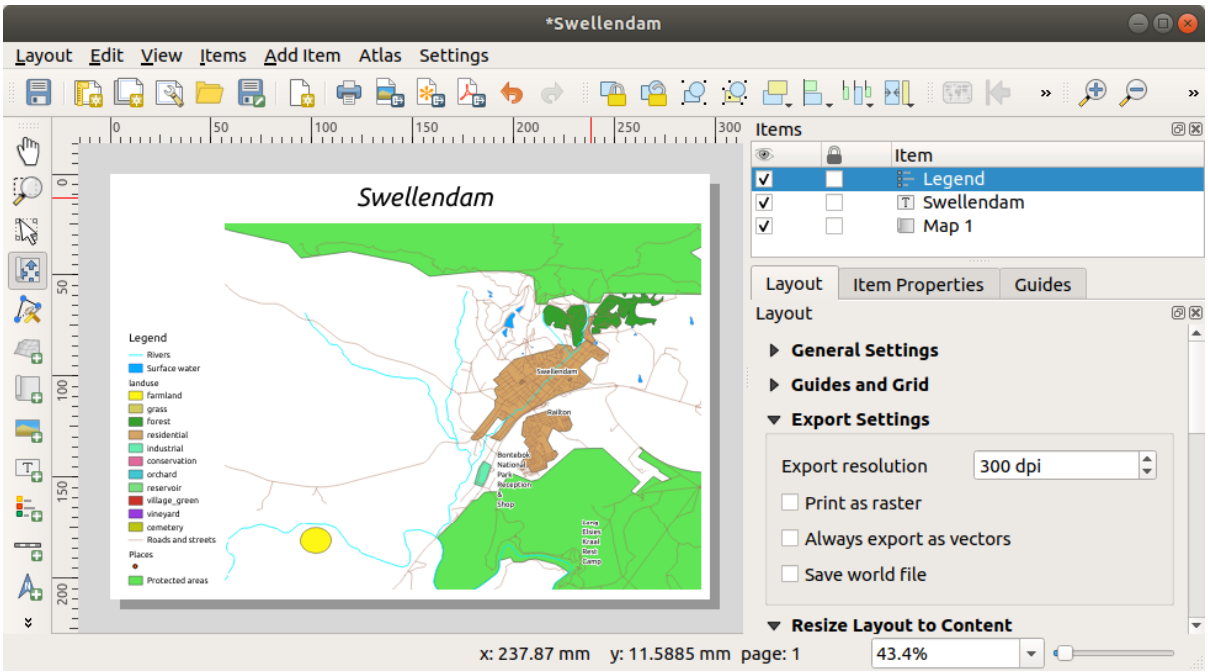
また、アイテムの名前を変更できます。

1. 同じリストからレイヤを選択します。
2. アイテムのプロパティを編集 ボタンをクリックします。
3. レイヤの名前を `Places`, `Roads and Streets`, `Surface Water`, `Rivers` に変更します。

また、アイテムを並び替えることもできます:



凡例はおそらく新しいレイヤ名によって広がることになるので、凡例または地図を移動したり大きさを変更したい場合があります。これがその結果です：



4.1.6 ★☆☆ （初級レベル） 理解しよう: 地図をエクスポートする

> **注釈**
>
> しばしば作業を保存することを覚えていましたか？

ようやく地図をエクスポートする準備ができました。レイアウトウィンドウの左上隅にエクスポートボタンが見えます:

- 印刷レイアウト: プリンタとのインタフェースを提供します。プリンタのオプションは、使用するプリンタのモデルによって異なるので、このトピックに関する詳細な情報は、プリンタのマニュアルや印刷に関する一般的なガイドを参照する方がよいでしょう。

  そのほかのボタンは地図ページをファイルにエクスポートするのに使います。

- 画像としてエクスポート: 様々な一般的な画像フォーマットから選択することができます。これはおそらく最もシンプルなオプションですが、作成される画像は「死んで」おり、編集するのが困難です。

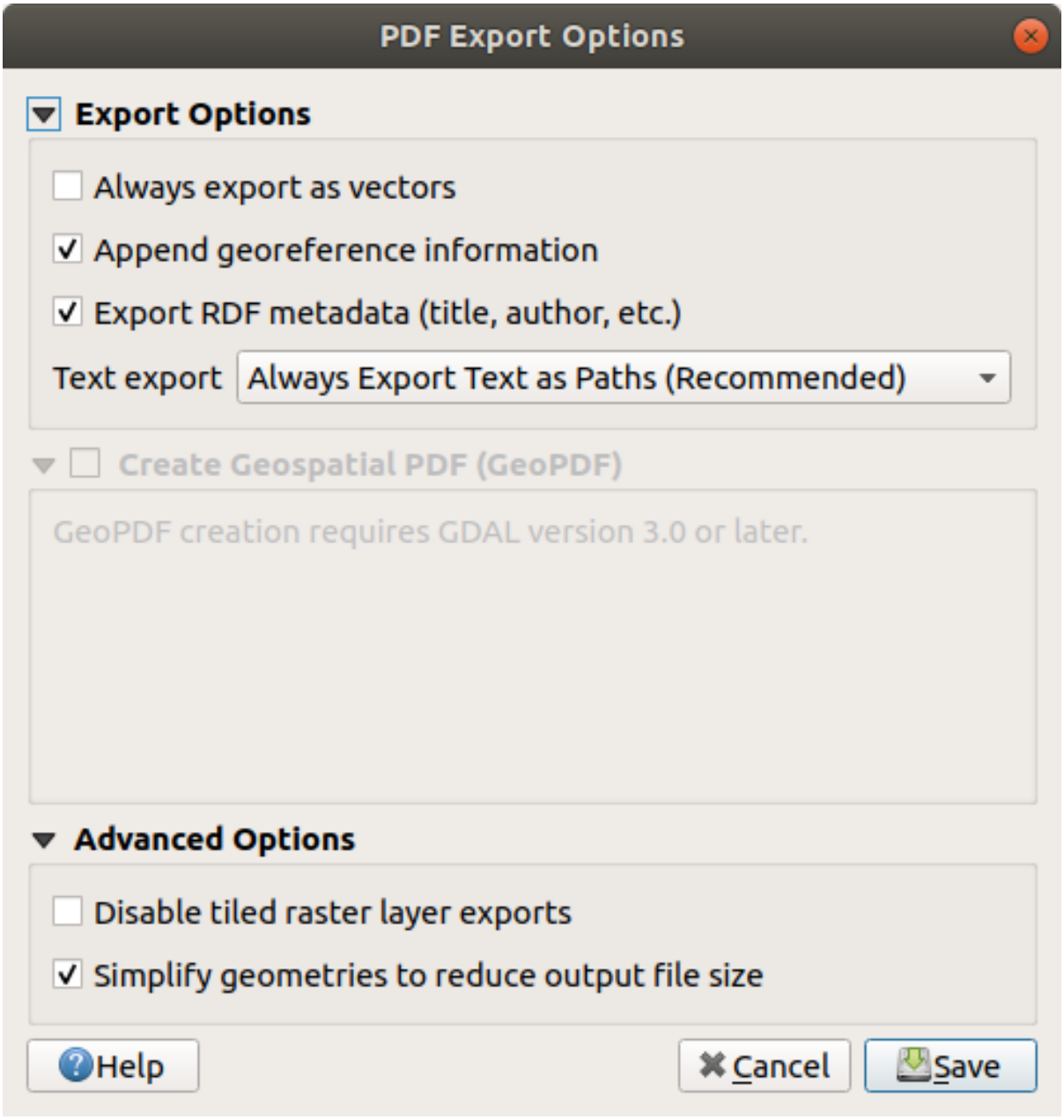
- SVG としてエクスポート: もしあなたが地図を地図製作者に送るなら (その人は出版用に地図を編集したいかもしれません)、SVG としてエクスポートするのが一番よいでしょう。SVG は "Scalable Vector Graphic" の略で、Inkscape やその他のベクター画像編集ソフトウェアにインポートすることができます。

- PDF としてエクスポート: クライアントに地図を送る必要がある場合、PDF を使うのが最も一般的です。なぜなら、PDF では印刷オプションを設定するのが簡単だからです。地図製作者の中には、このフォーマットをインポートしたり編集したりできるプログラムを持っていれば、PDF を好む人もいるでしょう。

ここでは PDF を使用します。

1. :sup:`PDF としてエクスポート ’ ボタンをクリックします。

2. 通常通り、保存場所とファイル名を選択します。以下のようなダイアログが表示されます。



3. これで安全にデフォルト値を使用することができますので、 **保存** をクリックします。

   QGIS は地図のエクスポートを進め、終了すると同時に印刷レイアウトダイアログの上にメッセージを表示します。

4. メッセージ内のハイパーリンクをクリックし、PDF が保存されているフォルダをシステムのファイルマネージャで開きます。

5. 開いてみて、レイアウトがどのように見えるかを確認してください。

   すべて OK ですか？初めての QGIS マッププロジェクト完成おめでとうございます！

6. 何か不満がありますか？ QGIS のウィンドウに戻り、適切な修正を行って再度エクスポートしてください。

7. プロジェクトファイルを保存することを忘れないでください。

4.1.7 結論

これで、基本的な静的マップレイアウトを作成する方法はお分かりいただけたと思います。さらに一歩進んで、より多くのレイアウト項目を持つ、動的に適応するマップレイアウトを作成することができます。

4.2 レッスン: ダイナミック印刷レイアウトを作る

基本的な地図レイアウトの作成方法を学んだので、さらに一歩進んで、地図の範囲やページのプロパティ（例えば、ページのサイズを変更したとき）に動的に適応する地図レイアウトを作成しましょう。また、作成日付も動的に適応します。

4.2.1 ★★☆ （中級レベル） 理解しよう: 動的マップキャンバスを作る

1. ESRI シェープファイル形式のデータセット `protected_areas.shp` 、 `places.shp` 、 `rivers.shp` および `water.shp` をマップキャンバスに読み込み、そのプロパティを自分の良いように修正します。
2. 全てがあなたの好みに合わせてレンダリングされ、シンボル化されたら、 **プロジェクト** ツールバーの 新規印刷レイアウト アイコンをクリックするか **プロジェクト** :menuselection:`新規印刷レイアウト`を選択します。新しい印刷レイアウトのタイトルを選択するようプロンプトが表示されます。
3. ヘッダと南アフリカのスウェレンダム近くの地域の地図で構成される地図レイアウトを作成します。レイアウトのマージンは 7.5 mm で、ヘッダの高さは 36mm でなければなりません。
4. キャンバス上に `main map` という地図アイテムを作成し、 **レイアウト** パネルに移動します。 **変数** セクションまでスクロールして、 **レイアウト** の部分を探します。ここでは、動的印刷レイアウト全体で使用することができるいくつかの変数を設定します。 **レイアウト** パネルに移動して、 **変数** セクションにスクロールダウンしてください。最初の変数がマージンを定義します。 ボタンを押し、 `sw_layout_margin` という名前を入力します。値を `7.5` に設定します。もう一度 ボタンを押して、 `sw_layout_height_header` という名前を入力します。値を `36` に設定します。
5. これで、マップキャンバスの位置とサイズを変数によって自動的に作成する準備が整いました。地図アイテムが選択されていることを確認して、 **アイテムプロパティ** パネルに移動し、スクロールダウンして **位置とサイズ** セクションを開いてください。 *X* の データによって定義された上書き をクリックし、 **変数** の項目から `@sw_layout_margin` を選択します。
6. *Y* の データによって定義された上書き をクリックし、 **編集...** を選択して数式を入力します：

```
to_real(@sw_layout_margin) + to_real(@sw_layout_height_header)
```

7. 地図アイテムのサイズは、 **幅** と **高さ** の変数を使用して作成することができます。 **幅** の データによって定義された上書き をクリックし、 **編集...** を再度選択します。数式を記入します：

```
@layout_pagewidth - @sw_layout_margin * 2
```

高さ の データによって定義された上書き をクリックし、 **編集...** を選択します。ここで数式を入力します:

```
@layout_pageheight - @sw_layout_height_header - @sw_layout_margin * 2
```

8. また、メインキャンバスの地図範囲の座標を含むグリッドを作成します。再び **アイテムプロパティ** に移動して、**グリッド** セクションを選択します。 ボタンをクリックして、グリッドを挿入します。**グリッドの修正**... をクリックして、QGIS メインキャンバスで選択した地図の縮尺に従って、*X*, *Y* と **オフセット** に **間隔** を設定します。**グリッドタイプ クロス** は、私たちの目的にとてもよく合っています。

4.2.2 ★★☆ （中級レベル） 理解しよう: 動的ヘッダを作る

1. 図形追加 ボタンでヘッダを含む長方形を挿入します。**アイテム** パネルに header という名前を入力します。
2. 再び **アイテムプロパティ** に移動し、**位置とサイズ** セクションを開きます。データ定義によって定義された上書き を使用して、*X* と *Y* に sw_layout_margin 変数を選択します。**幅** は式で定義されるものとします:

```
@layout_pagewidth - @sw_layout_margin * 2
```

そして **高さ** を sw_layout_height_header 変数で指定します。

3. ここでは、ノートアイテムを追加 を使って、ヘッダーを分割するための 1 本の横線と 2 本の縦線を挿入します。1 本の横線と 2 本の縦線を作成し、それぞれを Horizontal line, Vertical line 1, Vertical line 2 という名前にします。
    1. 横線：
        1. 変数 sw_layout_margin を *X* にします
        2. *Y* の式に次を設定します:

```
@sw_layout_margin + 8
```

        3. **幅** の式に次を設定します:

```
@layout_pagewidth - @sw_layout_margin * 3 - 53.5
```

    2. 最初の縦線:
        1. *X* の式に次を設定します:

```
@layout_pagewidth - @sw_layout_margin * 2 - 53.5
```

        2. *Y* に変数 sw_layout_margin を設定します
        3. 高さは作ったヘッダと同じでなければならないので、**高さ** には変数 sw_layout_height_header を設定します。
    3. 2 本目の縦線は、1 本目の縦線の左側に配置されます。

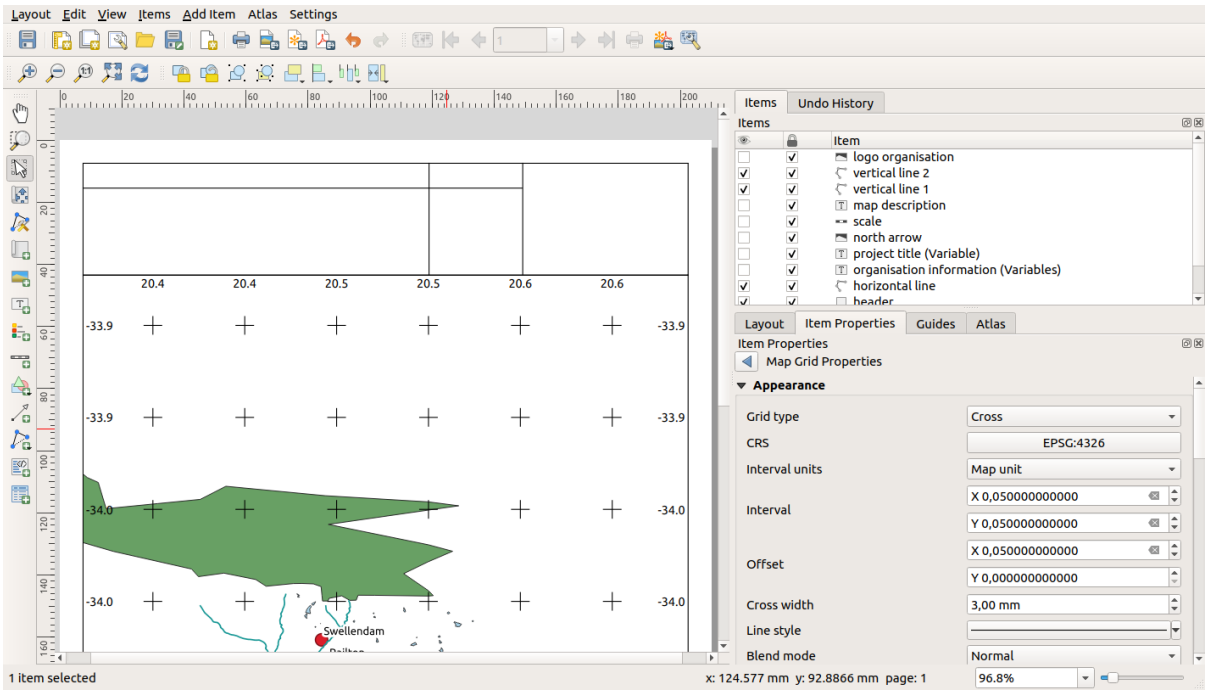
1. *X* の式に次を設定します:

```
@layout_pagewidth - @sw_layout_margin * 2 - 83.5
```

2. *Y* に変数 `sw_layout_margin` を設定します

3. 高さはもう一本の縦線と同じでなければならないので、変数 `sw_layout_height_header` を **高さ** に設定します。

以下の図は、動的レイアウトの構造を示しています。線によって作成された領域をいくつかの要素で埋めます。



4.2.3 ★★☆ （中級レベル） 理解しよう: 動的ヘッダのラベルを作る

1. あなたの QGIS プロジェクトのタイトルを自動的に含めることができます。タイトルは **プロジェクトプロパティ** で設定します。 ラベルを追加 ボタンでラベルを挿入し、 `project title (variable)` という名前を入力します。 **アイテムプロパティ** パネルの **メインプロパティ** に式 を入力します:

```
[%@project_title%]
```

ラベルの位置を設定します。

1. *X* には次の式を使います:

```
@sw_layout_margin + 3
```

2. *Y* には次の式を使います:

```
@sw_layout_margin + 0.25
```

3. **幅** には次の式を使います:

```
@layout_pagewidth - @sw_layout_margin *2 - 90
```

4. **高さ** には 11.25 を入力します

外観 にあるフォントサイズに 16 pt を設定します。

2. 二つ目のラベルには、作成した地図の説明を入れます。ここでもラベルを挿入し、名前を map description とします。 **メインプロパティ** に map description というテキストを入力します。また、 **メインプロパティ** には、次のように入力します:

```
printed on: [%format_date(now(),'dd.MM.yyyy')%]
```

ここでは、2 つの 日付と時刻 関数 (now と format_date) を使用しました。

ラベルの位置を設定します。

1. *X* には次の式を使います:

```
@sw_layout_margin + 3
```

2. *Y* には次の式を使います:

```
@sw_layout_margin + 11.5
```

3. 3 つ目のラベルには、あなたの組織に関する情報を記載します。まず、 **アイテムプロパティ** の **変数** メニューで、いくつかの変数を作成します。**レイアウト** メニューから  ボタンをクリックして、o_department, o_name, o_adress, o_postcode という名前を入力してください。2 行目には、あなたの組織に関する情報を入力してください。これらの変数は **メインプロパティ** セクションで使用します。

メインプロパティ で次を入力します:

```
[% @o_name %]
[% @o_department %]
[% @o_adress %]
[% @o_postcode %]
```

ラベルの位置を設定します。

1. *X* には次の式を使います:

```
@layout_pagewidth - @sw_layout_margin - 49.5
```

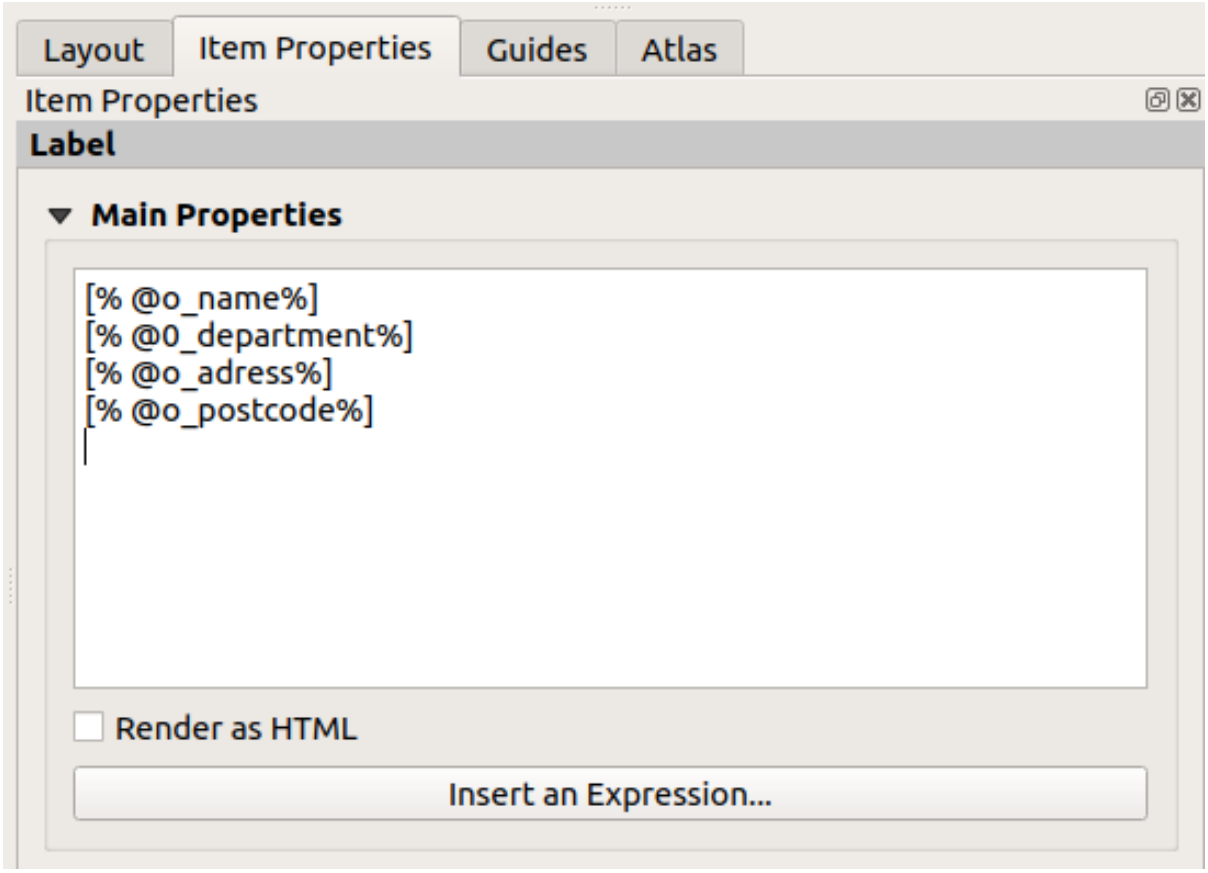
2. *Y* には次の式を使います:

```
@sw_layout_margin + 15.5
```

3. **幅** を 49.00 にします

4. **高さ** には次の式を使います:

```
@sw_layout_height_header - 15.5
```



4.2.4 ★★☆ （中級レベル） 理解しよう: 動的ヘッダに画像を追加する

1. 画像を追加 ボタンを使って、ラベル `organisation information` の上に画像を配置することができます。 `organisation logo` という名前を入力した後、ロゴの位置とサイズを決定します:

   1. *X* には次の式を使います:

      ```
      @layout_pagewidth - @sw_layout_margin - 49.5
      ```

   2. *Y* には次の式を使います:

      ```
      @sw_layout_margin + 3.5
      ```

   3. **幅** を `39.292` にします

   4. **高さ** を `9.583` にします

   あなたの組織のロゴを入れるには、ロゴをホームディレクトリに保存し、**メインプロパティ** *Image Source* にパスを入力する必要があります。

2. 私たちのレイアウトにはまだ方位記号が必要です。これも 方位記号を追加 を使って挿入します。ここでは既定の方位記号を使用します。位置を定義します：

1. *X* には次の式を使います:

```
@layout_pagewidth - @sw_layout_margin * 2 - 78
```

2. *Y* には次の式を使います:

```
@sw_layout_margin + 9
```

3. **幅** を 21.027 にします
4. **高さ** を 21.157 にします

4.2.5 ★★☆ （中級レベル） 理解しよう: 動的ヘッダのスケールバーを作る

1. ヘッダーにスケールバーを挿入するには、  スケールバーを追加 をクリックして、方位記号の上の矩形に配置します。 **メインプロパティ** の **地図** で Main Map(Map 1) を選択します。これは、QGIS のメインキャンバスで選択した範囲に応じて縮尺が自動的に変更されることを意味します。 **スタイル** は **数値** を選択します。これは、スケールバーのないシンプルなスケールを挿入することを意味します。このスケールにはまだ位置とサイズが必要です。

   1. *X* には次の式を使います:

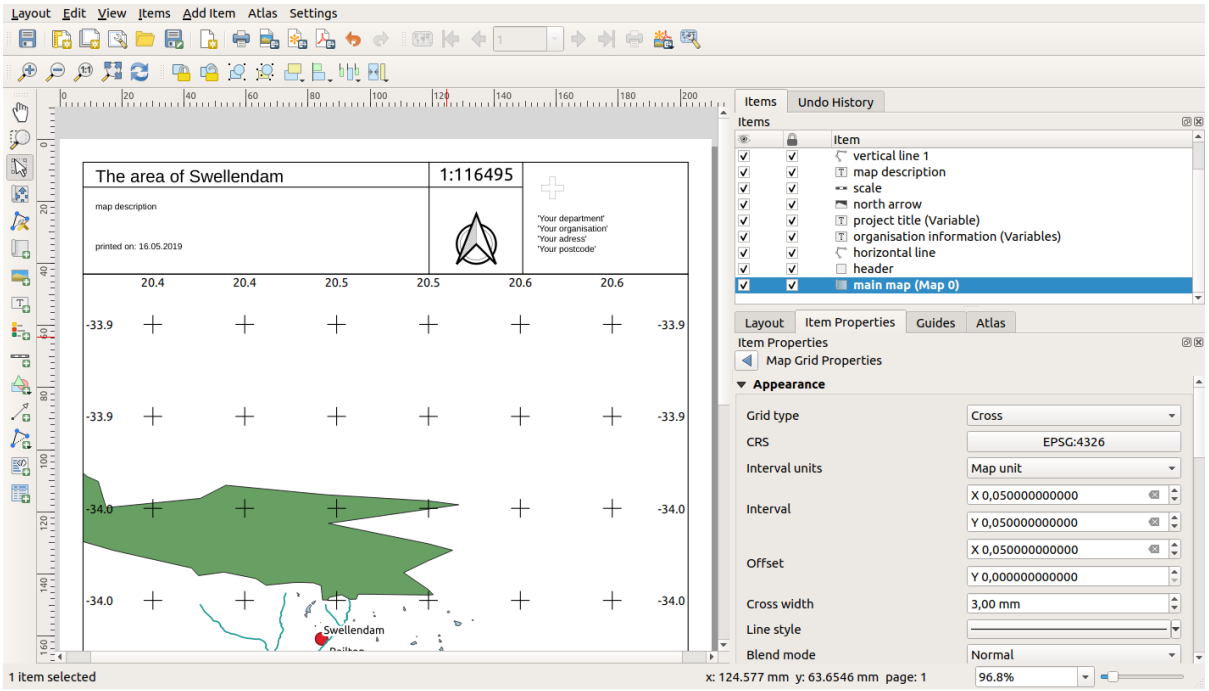
```
@layout_pagewidth - @sw_layout_margin * 2 - 78
```

   2. *Y* には次の式を使います:

```
@sw_layout_margin + 1
```

   3. **幅** を 25 にします
   4. **高さ** を 8 にします
   5. 基準点 を中央に置きます。

Congratulations! You have created your first dynamic map layout. Take a look at the layout and check if everything looks the way you want it! The dynamic map layout reacts automatically when you change the *Page Properties*. For example, if you change the page size from DIN A4 to DIN A3, click the  Refresh view button and the page design is adapted.



4.2.6 次は?

次のページでは、完成すべき課題が与えられます。これによって、これまでに学んだテクニックを実践できます。

4.3 課題 1

あなたの既存の地図プロジェクトを開き、徹底的にそれを修正します。あなたが以前に修正したかったと思う小さな誤りや物事に気づいた場合は、ここでやります。

地図をカスタマイズしながら、自分自身に問い続けてください。この地図は、データに不慣れな人でも読みやすく理解しやすいでしょうか？この地図をインターネットで、またはポスターで、または雑誌で見た場合、注意を惹かれるでしょうか？自分の地図でなかったとしたら、この地図を読みたいでしょうか？

このコースを ★☆☆ 初級または ★★☆ 中級レベルでやっている場合、より高度なセクションからのテクニックをよく読んでください。自分の地図で行いたいものがあったら、実装しようと試みてください。

このコースは、あなたに提示されている場合は、コースのプレゼンターが評価のために地図の最終版を、PDF にエクスポートして、提出するよう求めるかもしれません。自分でこのコースをやっている場合、同じ基準を使用して、ご自身の地図を評価することをお勧めします。地図は地図自体だけでなく、地図ページと要素の外観やレイアウトの全体的な外観と記号で評価されます。地図の外観を評価するための重点は、常に **使いやすさ** になることを覚えておいてください。見た目が良いほど、一目で簡単に理解できるほど、良い地図です。

ハッピーカスタマイズ！

4.3.1 結論

最初の 4 つのモジュールでは、ベクタ地図を作成し、スタイルを付ける方法を教えてきました。次の 4 つのモジュールでは、完全な GIS 分析に QGIS を使用する方法を学びます。内容は、ベクタデータを作成および編集する；ベクタデータを分析する；ラスタデータを使用および分析する；ラスタとベクタの両方のデータソースを使用して、GIS を使用して最初から最後まで問題を解決する。

第5章　モジュール: ベクタデータを作る

既存のデータを使用して地図を作成することはまだ始まったばかりです。このモジュールでは、既存のベクタデータを変更して、まったく新しいデータセットを作成する方法を学びます。

5.1 レッスン: 新しいベクタデータセットを作る

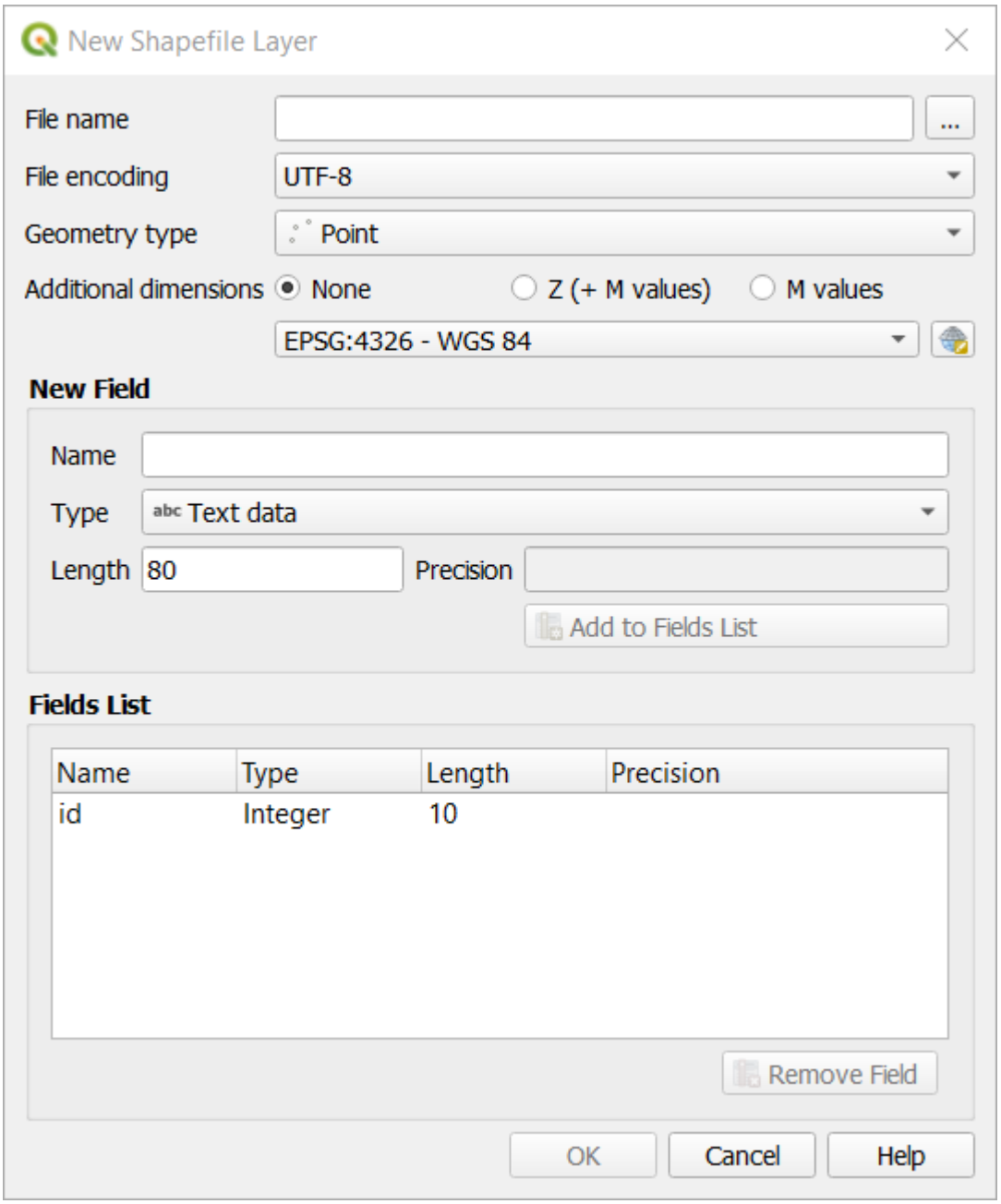
使用するデータはどこかから持ってこなければなりません。最も一般的なアプリケーションでは、データがすでに存在しています。しかしプロジェクトがより特化し専門的になるほど、データが既に利用可能であるという可能性が低くなります。このような場合は、自身の新しいデータを作成する必要があります。

このレッスンの目標： 新しいベクタデータセットを作成します。

5.1.1 ★☆☆ （初級レベル） 理解しよう: レイヤ作成ダイアログ

新しいベクタデータを追加するには、まずそれを追加するためのベクタデータセットが必要です。現在の場合は、既存のデータセットを編集するのではなく、完全に新しいデータを作成して始めましょう。それゆえ、まず自分自身の新しいデータセットを定義する必要があります。

1. QGIS を開き新しい無地のプロジェクトを作ります。
2. メニューから **レイヤ　　レイヤを作成　　新しいシェープファイルレイヤ** を選択し、クリックします。新しいレイヤを定義するための **新規シェープファイルレイヤ** ダイアログが表示されます。



3. **ファイル名** フィールドの ... をクリックします。保存ダイアログが現れます。
4. `exercise_data` ディレクトリに移動します。
5. 新しいレイヤを `school_property.shp` として保存します。

   この段階で欲しいデータセットの種類を決定することが重要です。それぞれの異なるベクタレイヤタイプは、バックグラウンドで「別々に構築」されているので、一度レイヤを作成したらそのタイプは変更できません。

   次の練習では、区域を表す新しい地物を作成します。このような地物には、ポリゴンデータセットを作成する必要があります。
6. **ジオメトリタイプ** ではドロップダウンメニューから **ポリゴン** を選びます：



これは、ダイアログの残りの部分には影響しませんが、それは、ベクタデータセットが作成されたときにジオメトリの正しいタイプが使用されるようになります。

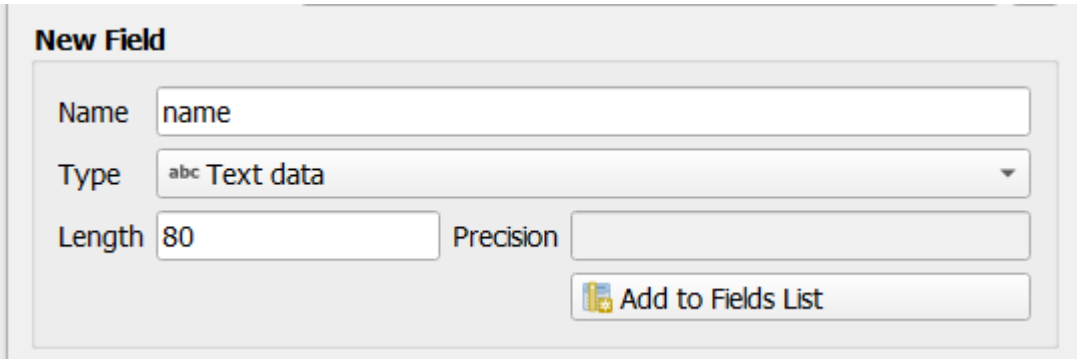
次のフィールドでは、座標参照系、または CRS、を指定します。CRS は数値座標と地球表面の位置とを関連付ける方法です。詳しくはユーザーズマニュアルの 投影法の操作 を参照してください。

この例では、プロジェクトに関連付けられた既定の CRS である WGS84 を使います。

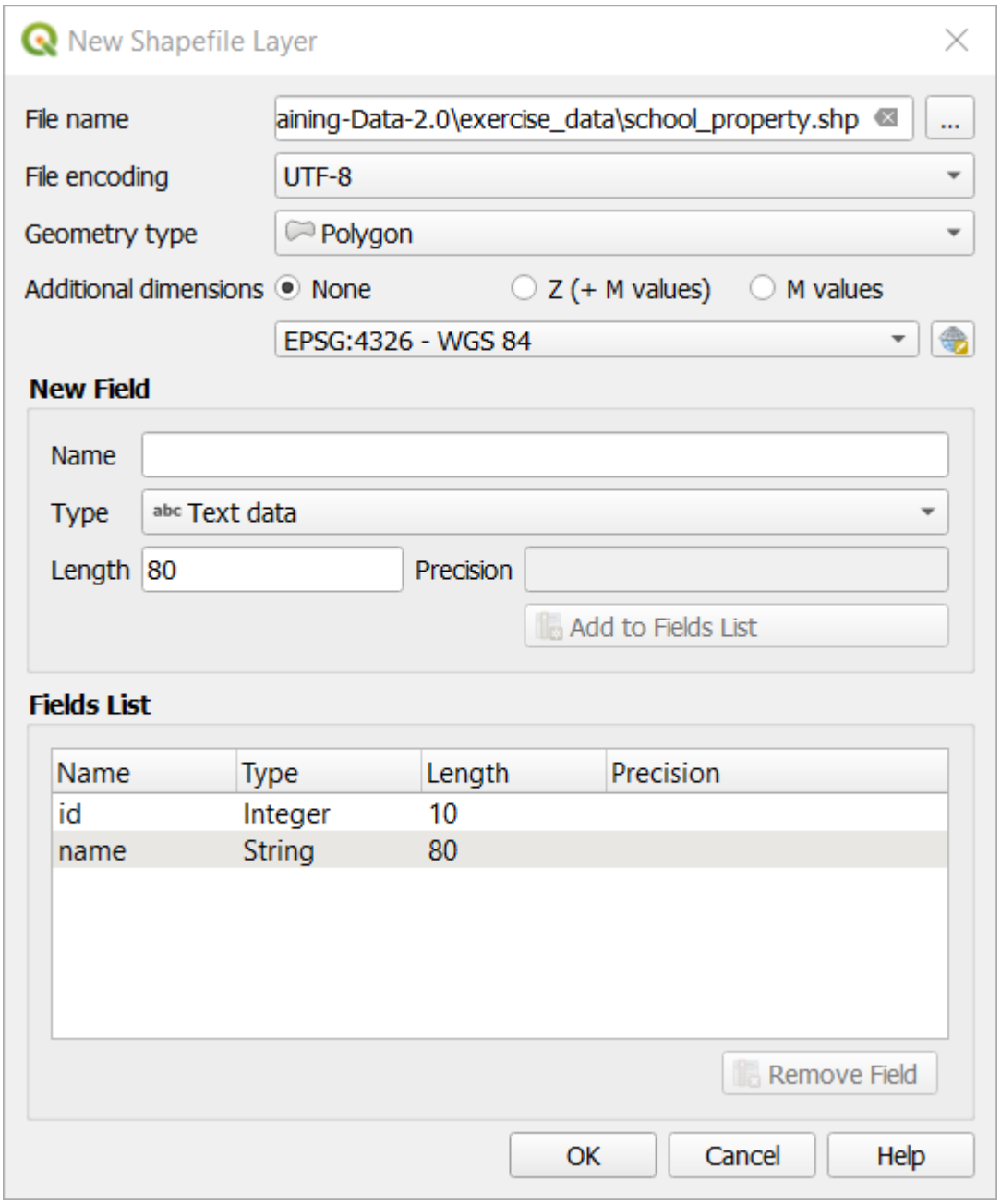


次に、 **新規フィールド** の下にグループ化されたフィールドのコレクションがあります。既定では、新しいレイヤは 1 つの属性、 `id` フィールド（下の **フィールドリスト** に表示されています）しか持ちません。しかし、作成したデータを有効に活用するためには、実際にこの新しいレイヤに作成する地物について、何か記述する必要があります。今のところ、 `name` というフィールドをひとつ追加して、テキスト長を `80` 文字に制限された テキストデータ を格納することで十分でしょう。

7. 以下の設定を再現し、**フィールドリストに追加** ボタンをクリックします：



8. ダイアログが次のようになることを確認します。



9. *OK* をクリックします

新しいレイヤが **レイヤ** パネルに表示されるはずです。

5.1.2 ★☆☆ （初級レベル） 理解しよう: データソース

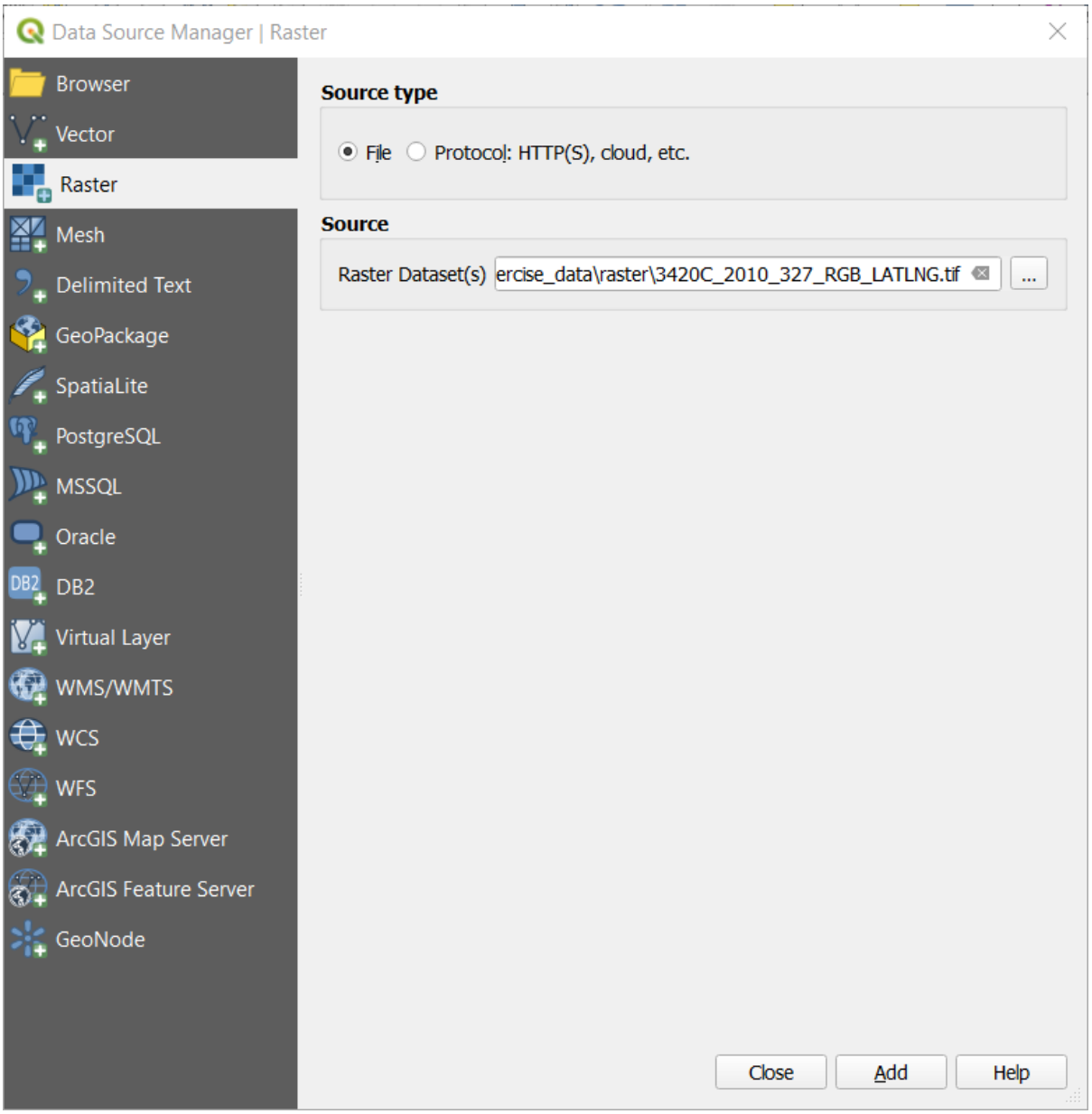
新しいデータを作成するとき、それは明らかに地上に現実に存在するオブジェクトに関するものである必要があります。そのため、どこかから情報を取得する必要があります。

オブジェクトに関するデータを取得するにはさまざまな方法があります。たとえば、GPS を使用して現実の世界でのポイントをキャプチャし、それから QGIS にデータをインポートできます。あるいは、セオドライトを使用してポイントを調査し、新しい地物を作成するために、手動で座標を入力できます。あるいは、デジタイズプロセスを使用して、衛星画像や航空写真などのリモートセンシングデータからオブジェクトをトレースできます。

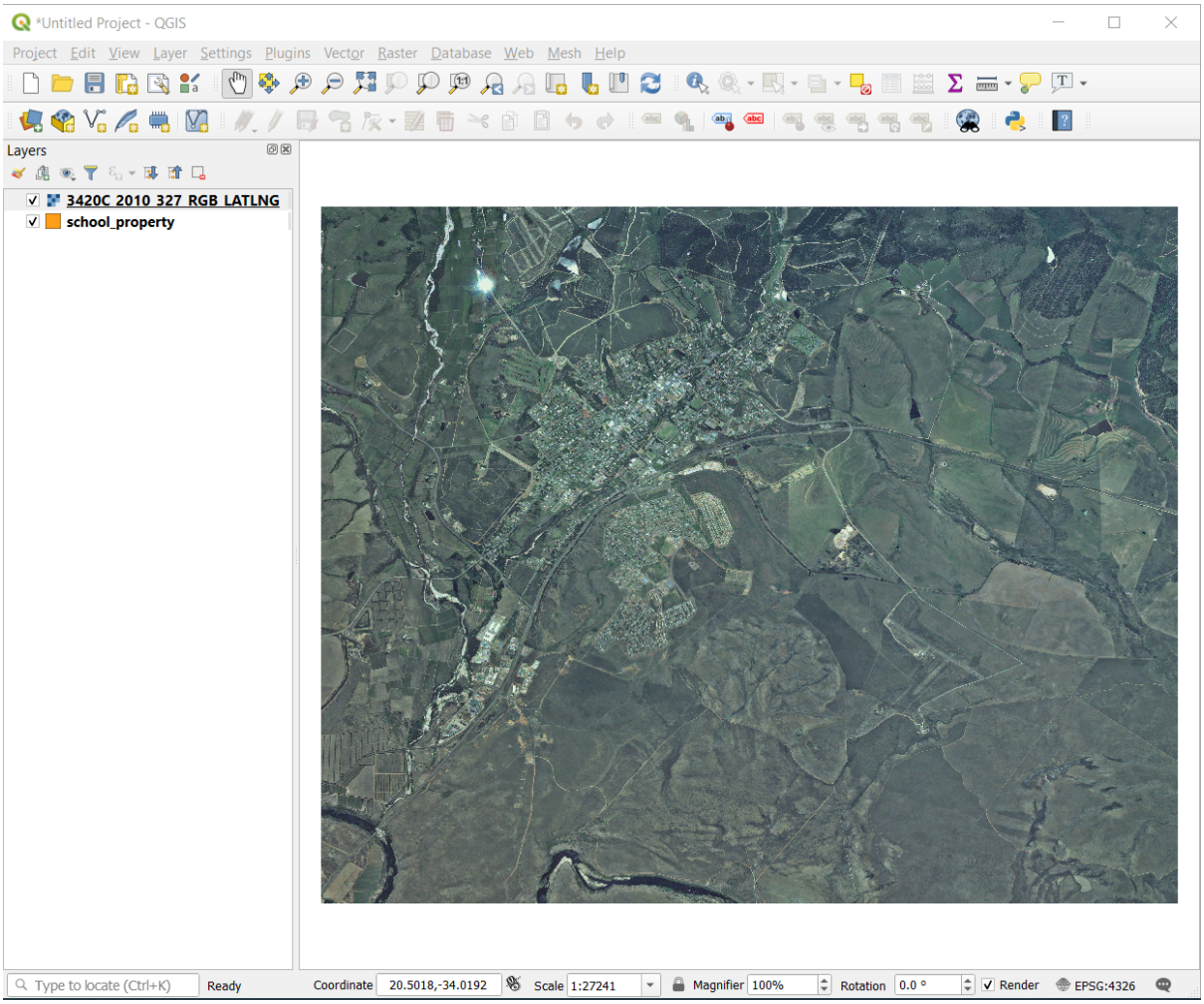
この例では、デジタイズのアプローチを使用します。サンプルラスタデータセットが提供されているので、必要に応じてそれらをインポートする必要があります。

1. 

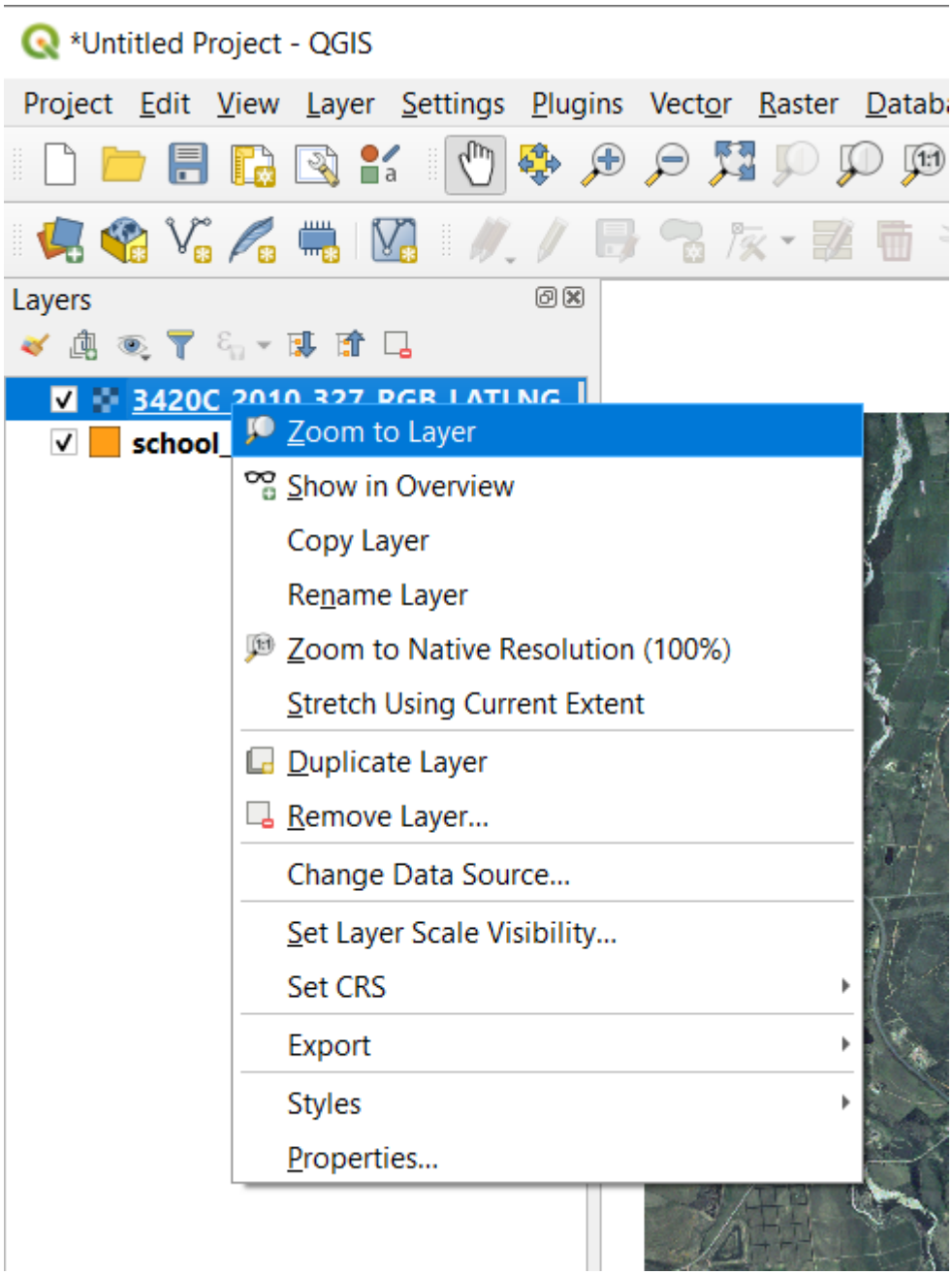
ボタンをクリックします。
2. 左にある  **ラスタ** を選びます。
3. **ソース** パネルにある ... ボタンをクリックします:
4. `exercise_data/raster/` に移動します。
5. ファイル `3420C_2010_327_RGB_LATLNG.tif` を選びます。
6. **開く** をクリックしてダイアログウィンドウを閉じます。



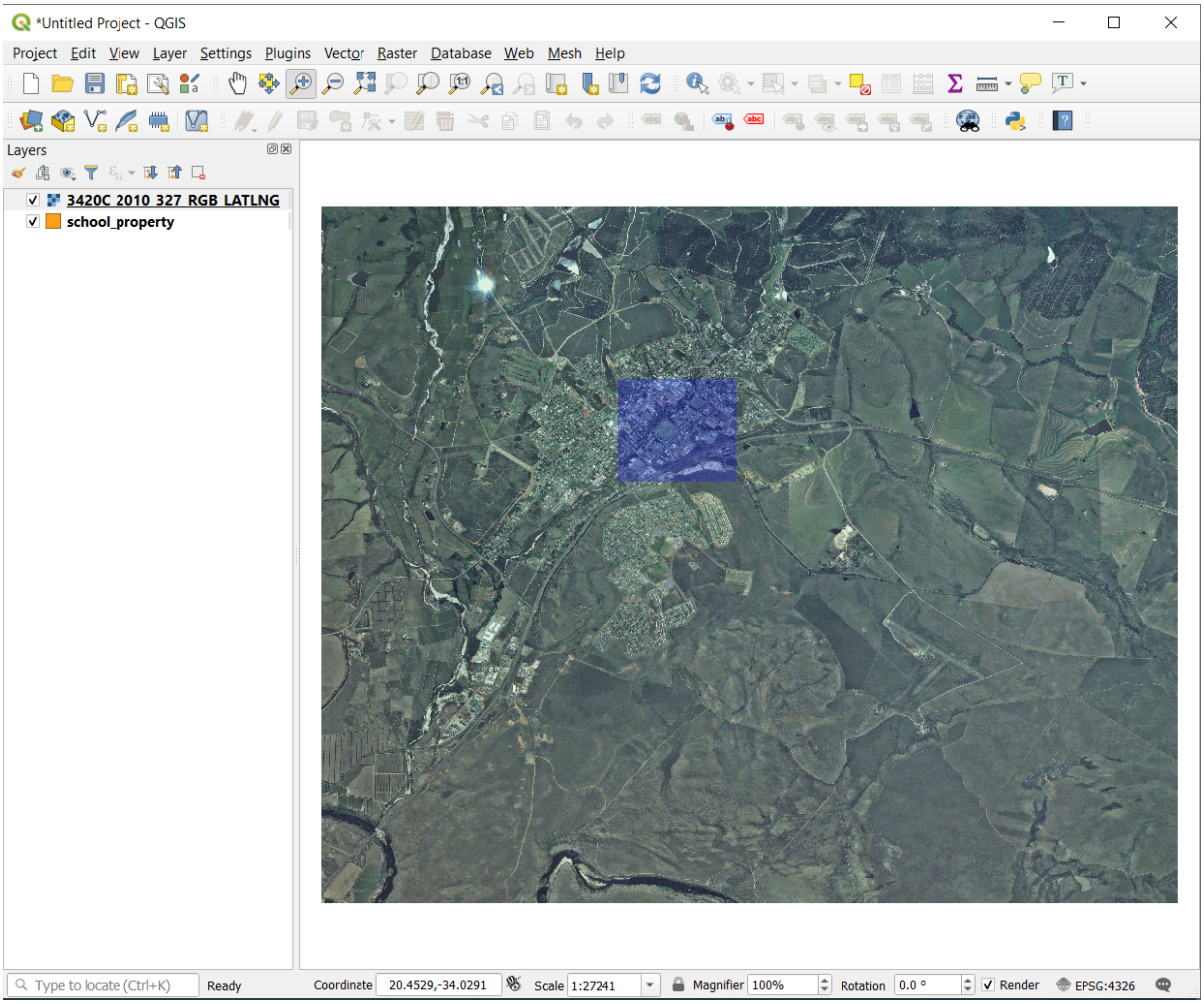
7. **追加** と **閉じる** をクリックします。画像が地図に読み込まれます。



8. 航空写真が表示されない場合は、新規レイヤを選択して右クリックし、コンテキストメニューから **レイヤの領域にズーム** を選択してください。



9. 拡大 ボタンをクリックし、下の青くハイライトされている部分にズームインします：

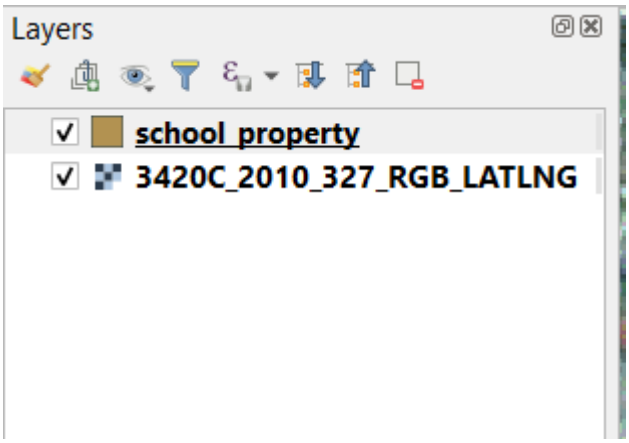


これでこの三つの運動場をデジタイズする準備ができました：



デジタイズする前に、 school_property レイヤを空中写真の上に移しましょう。

1. **レイヤ** ペーンにある school_property レイヤを選び、一番上にドラッグします。



デジタイズを開始するためには、**編集モード** に入る必要があります。GIS ソフトウェアでは一般的に、重要なデータを誤って編集したり削除することを防ぐために、これが必要とされます。編集モードは、レイヤごとに個別にオンまたはオフに切り替えられます。

school_property レイヤで編集モードに入るには:

1. **レイヤ** パネルで school_property レイヤをクリックして選択します。
2.  編集モード切替 ボタンをクリックします。

   このボタンを見つけることができない場合は、**デジタイズ** ツールバーが有効になっているか確認してください。**ビュー　ツールバー　デジタイジング** メニューエントリ の横にチェックマークがあるはずです。

   編集モードに入るとすぐに、いくつかのデジタイズツールが有効になっているのがわかります：

   -  ポリゴン地物を追加
   -  頂点ツール

   他の関連ボタンはまだ無効ですが、新しいデータに触れると有効になります。

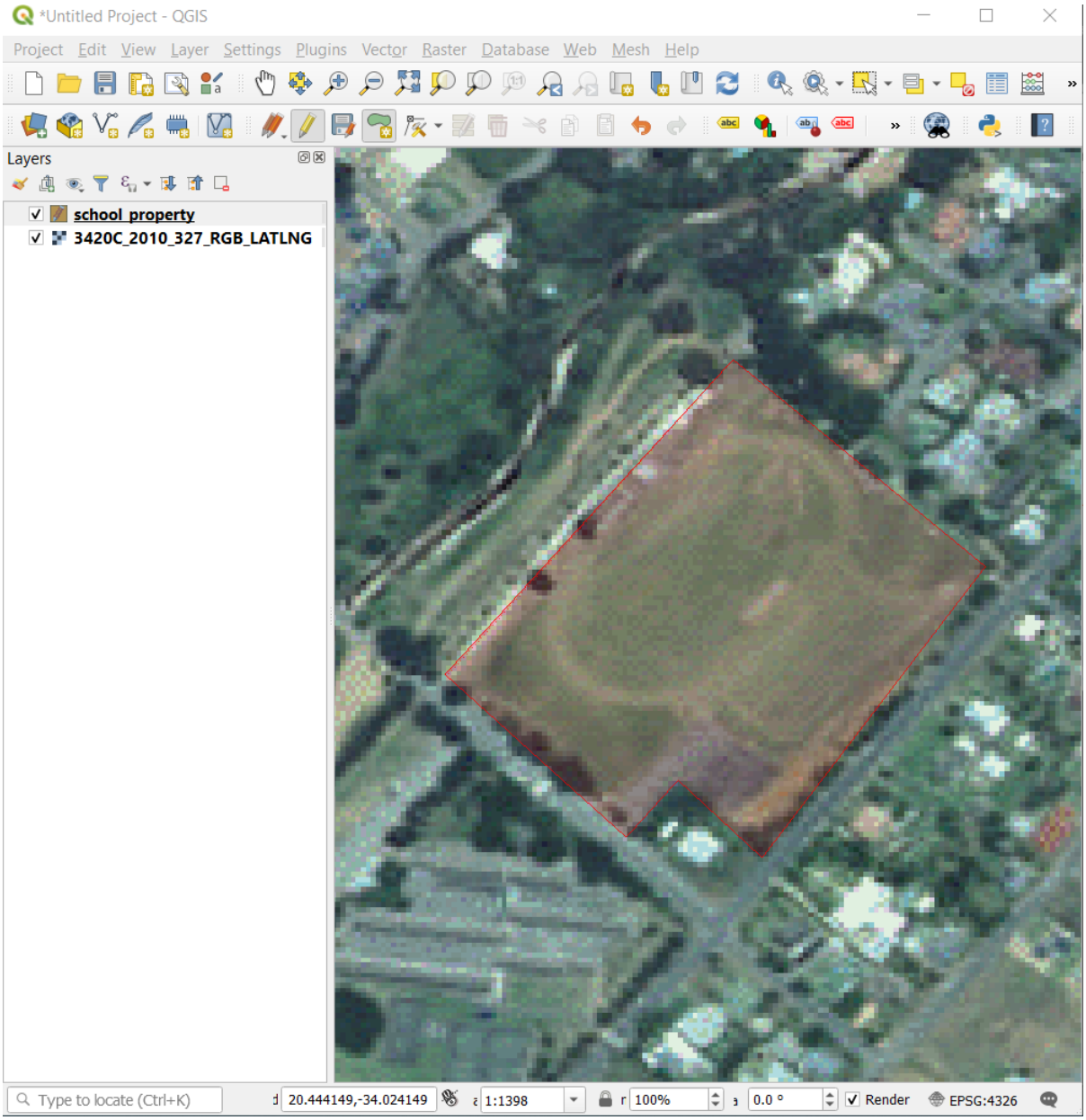
   レイヤ パネルの school_property レイヤ に鉛筆のアイコンが表示され、編集モードになっていることを示していることに注意してください。
3.  ポリゴン地物を追加 ボタンをクリックし、学校の運動場のデジタイズを始めます。

   マウスカーソルが十字になったのがわかると思います。これにより、デジタイズする点をより正確に配置することができます。デジタイズツールを使っているときでも、マウスホイールを回して地図を拡大・縮小したり、マウスホイールを押したまま地図内をドラッグしてパンできることを覚えておいてください。

   デジタイズしている第 1 の地物は athletics field です：

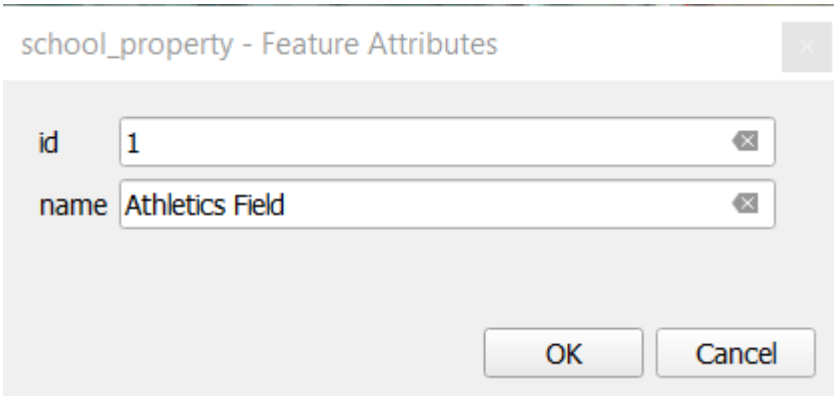


4. 運動場の縁のどこかの点でクリックすることでデジタイズを開始します。
5. さらに縁に沿ってクリックして点を置いてゆき、描画している図形が運動場を完全に覆うようにします。

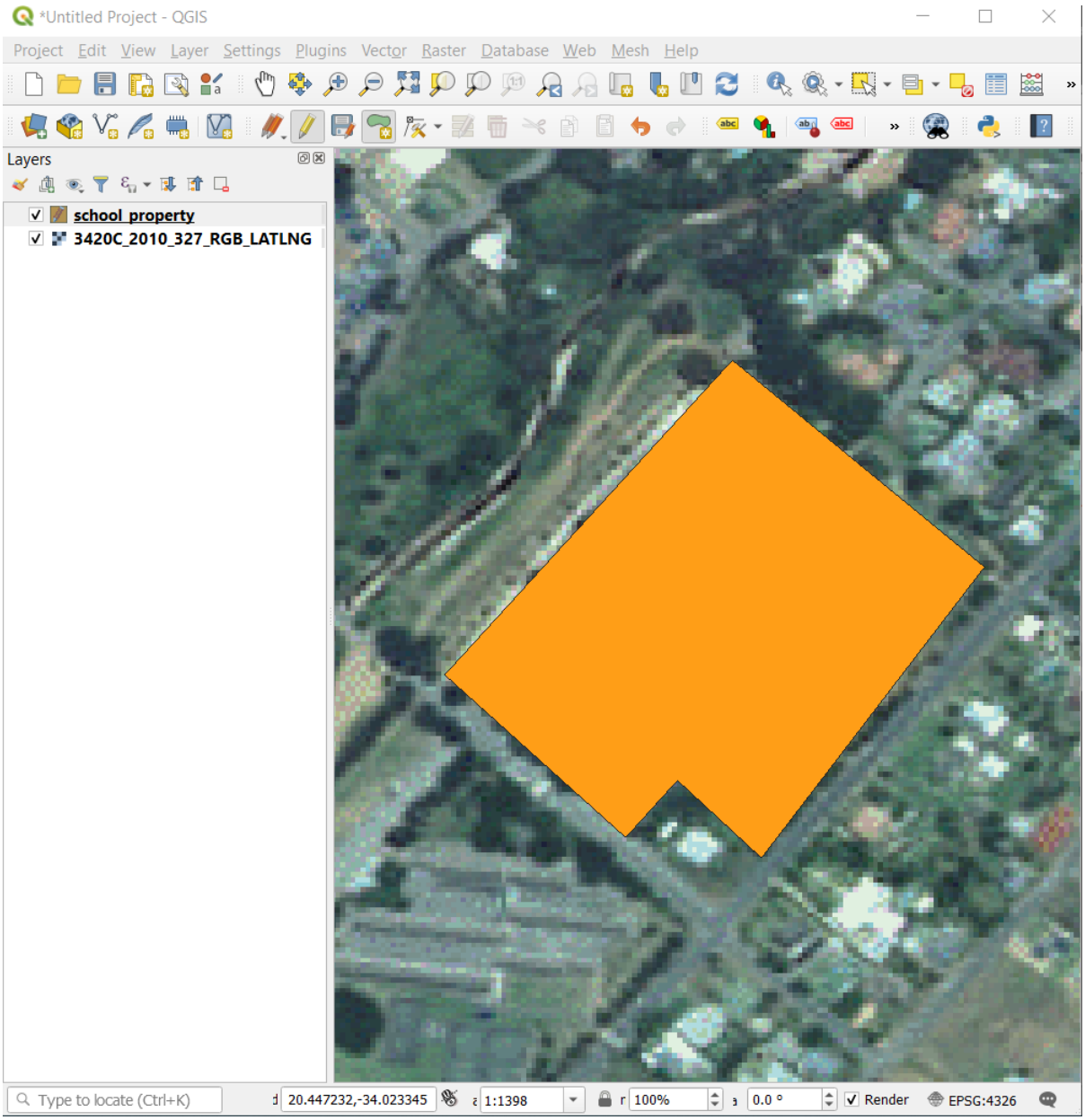


6. 最後の点を配置した後、右クリックしてポリゴンの描画を終了します。これで地物が確定し、 **属性** ダイアログが表示されます。

7. 値を以下のように埋めます:

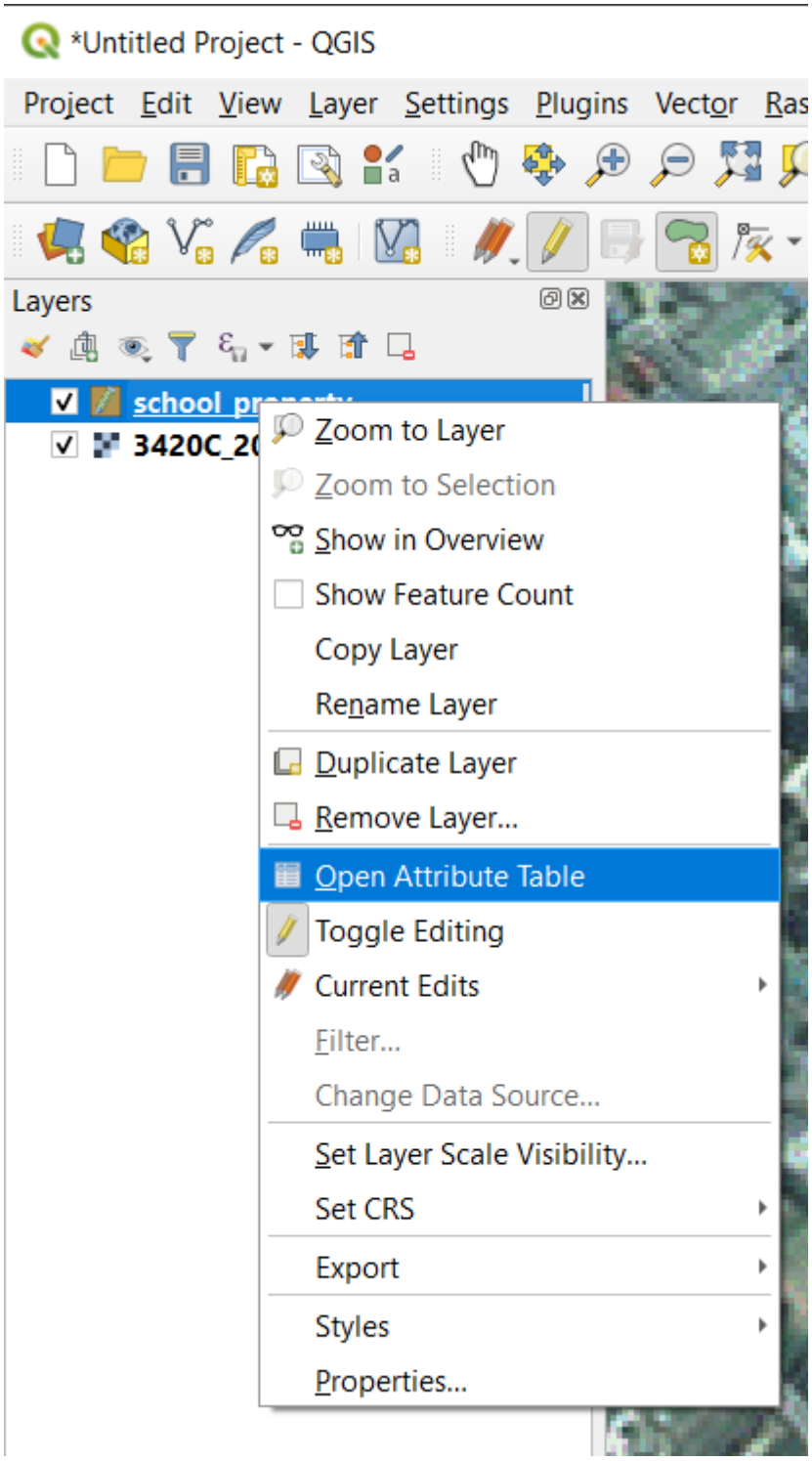


8. *OK* をクリックすると、新しい地物が完成しました！

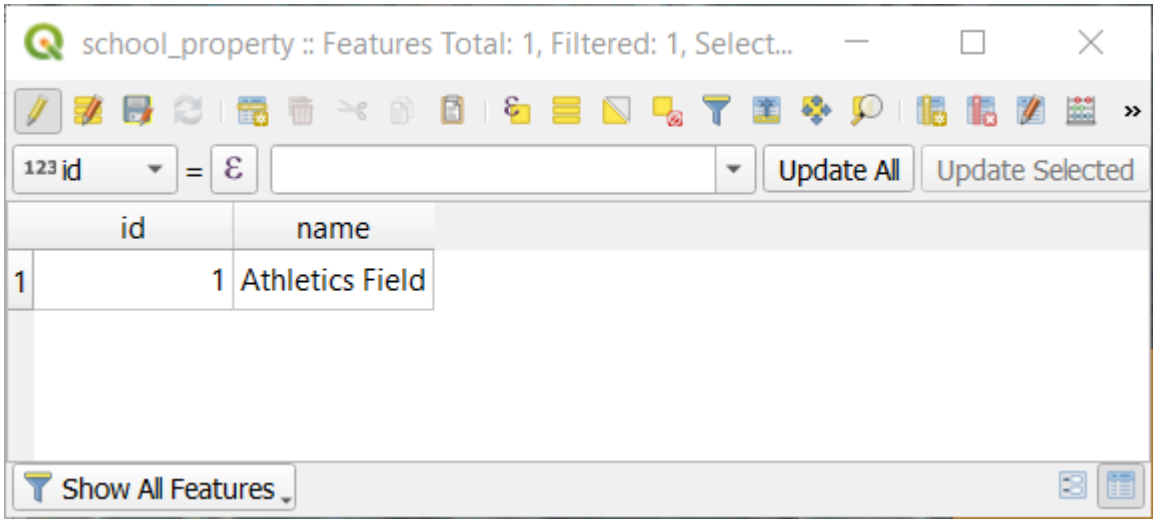


9. **レイヤ** パネルで `school_property` レイヤを選択します。

10. 右クリックしてコンテキストメニューから **属性テーブルを開く** を選びます。



表には、追加したばかりの地物が表示されます。編集モードでは、更新したいセルをダブルクリックすることで、属性データを更新することができます。



11. 属性テーブルを閉じます。
12. 作成したばかりの新しい地物を保存するには、編集内容の保存 ボタンをクリックします。

地物をデジタイズしているときに間違えた場合、作成が終わった後で常にそれを編集できることを覚えておいてください。間違えた場合は、前述のように地物を作成し終わるまで、デジタイズを継続します。その後：

1. 頂点ツール ボタンをクリックします。
2. 移動したい頂点の上にマウスを置き、左クリックします。
3. 頂点の正しい位置にマウスを移動し、左クリックします。これにより、頂点が新しい位置に移動します。





線分の移動も同じ要領で行えますが、線分の中点にカーソルを合わせる必要があります。

変更を取り消す場合は、 元に戻す ボタンか Ctrl+Z を押します。

4. レイヤ編集内容を保存 ボタンをクリックして、変更内容を保存するのを忘れないようにしてください。
5. 編集が終わったら、 編集モード切替 ボタンをクリックして、編集モードから出ます。

5.1.3 ★☆☆ （初級レベル） 自分でやってみよう: ポリゴンをデジタイズする

学校自体と上のフィールドをデジタイズします。デジタイズを支援するためにこの画像を使用します：



それぞれの新しい地物は、一意な id 値を持つ必要があることを覚えておきましょう!

注釈

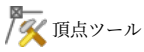
レイヤーに地物を追加し終わったら、編集内容を保存して、編集モードを終了することを忘れないでください。

注釈

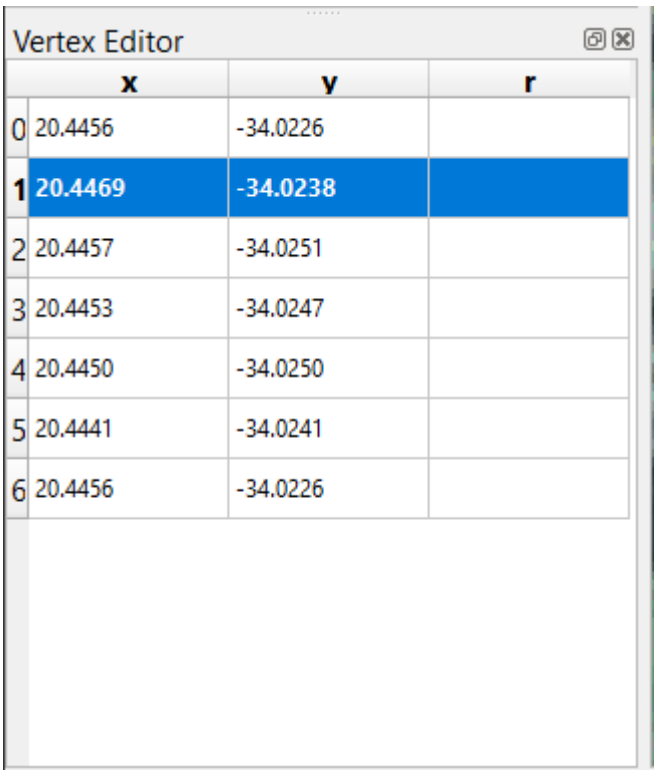
以前のレッスンで学んだ技法を使って、school_property の塗りつぶし、輪郭、ラベルの配置や形式の体裁を整えることができます。

5.1.4 ★★☆ （中級レベル） 理解しよう: 頂点編集テーブルを使う

地物を編集するもう一つの方法は、**頂点編集** テーブルを使って各頂点の実際の座標値を手動で入力することです。

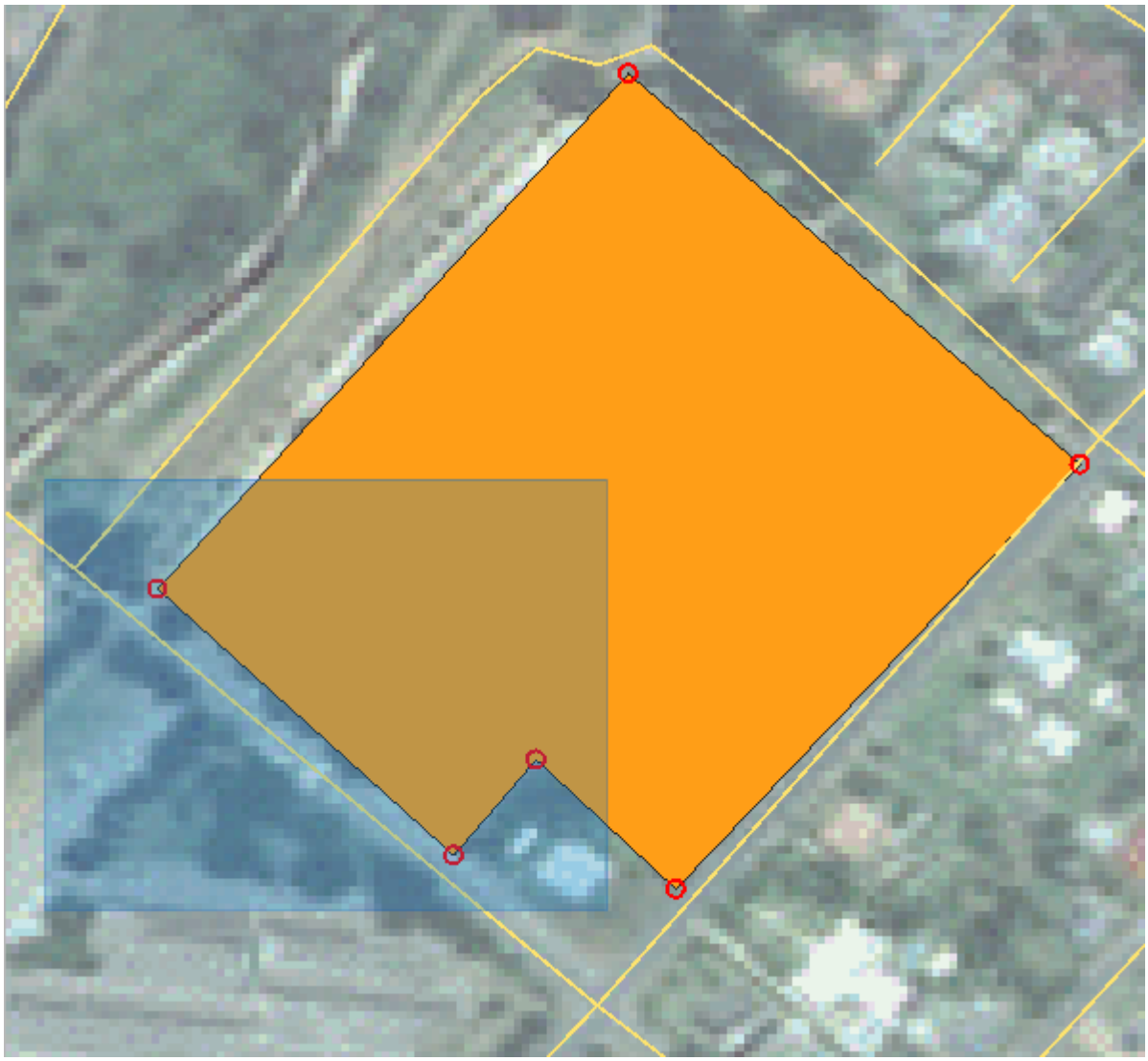
1. school_property レイヤの編集モードになっていることを確認してください。
2. まだアクティブになっていない場合は、 

ボタンをクリックします。
3. school_property レイヤに作成したポリゴン地物の上にマウスを移動し、右クリックをします。すると、その地物が選択され、 **頂点エディタ** ペインが表示されます。

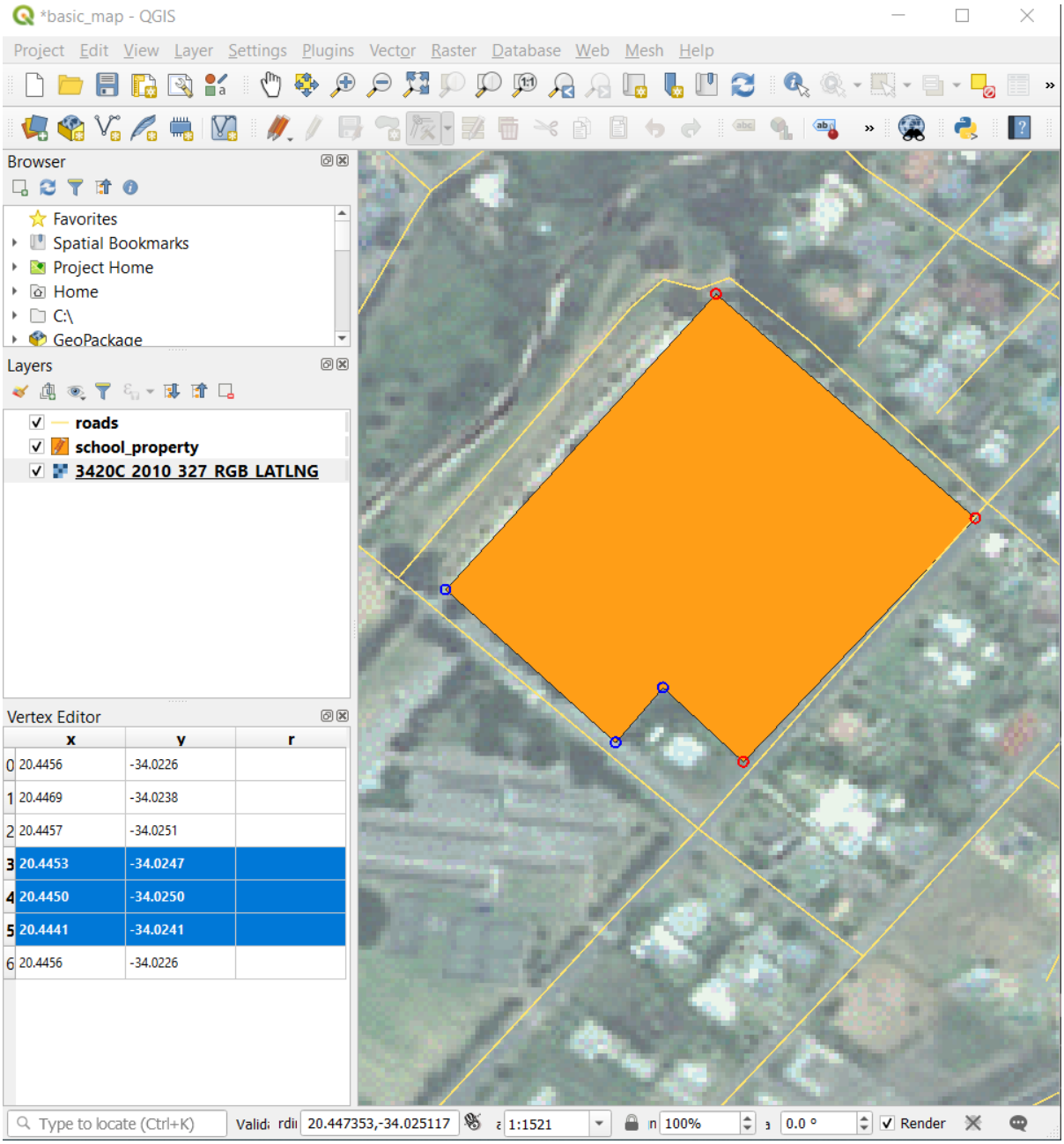


> **注釈**
>
> このテーブルは、その地物の頂点の座標を含んでいます。この地物には 7 つの頂点がありますが、地図上で確認できるのは 6 つだけであることに注意してください。よく見ると、0 行目と 6 行目が同じ座標であることに気づくでしょう。これらは地物ジオメトリの始点と終点であり、閉じたポリゴン地物を作成するために必要です。

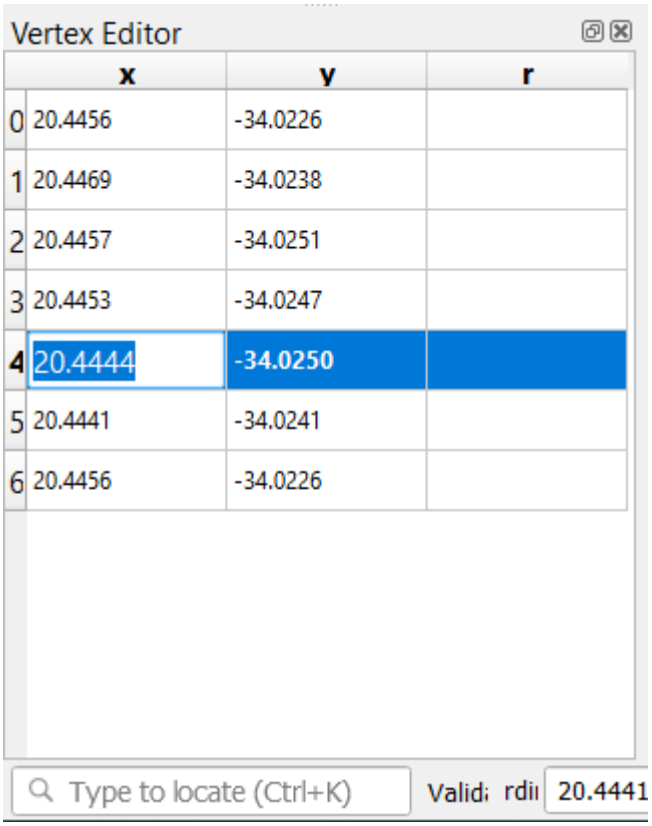
4. 選択した地物のひとつまたは複数の頂点のうえでボックスをクリック＆ドラッグします。



選択された頂点の色が青に変わり、頂点の座標を含んでいる **頂点エディタ** テーブルの対応する行がハイライトされます。



5. 座標を更新するには、編集したいテーブルのセルを左ダブルクリックし、更新した値を入力します。この例では、行 4 の x 座標が `20.4450` から `20.4444` に更新されます。

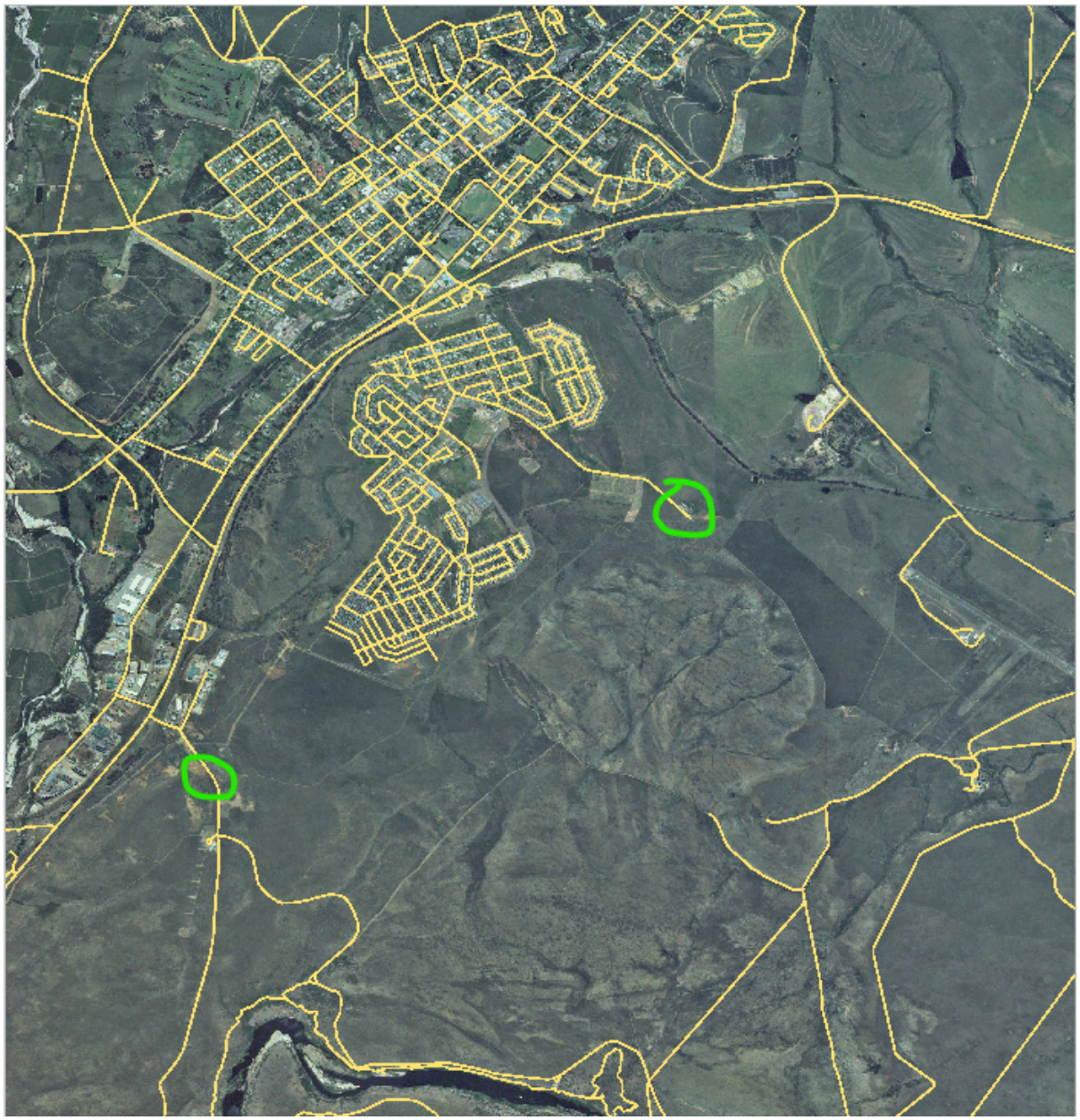


6. 更新した値を入力したら、エンターキーを押して変更を適用します。マップウィンドウで頂点が新しい位置に移動するのが確認できます。

7. 編集が終わったら、 編集モード切替 ボタンをクリックして編集モードを解除し、編集内容を保存します。

5.1.5 ★☆☆ （初級レベル） 自分でやってみよう: 線をデジタイズする

ここでは、道路レイヤにまだ記されていない 2 つのルートをデジタイズします。一つは細道、もう一つは小道です。細道はレイルトン郊外の南端を走り、印を付けた道路を起点と終点としています：



小道はさらにもう少し南です:



1. **道路** レイヤがまだマップにない場合は、ダウンロードした訓練データの `exercise_data` フォルダに含まれる GeoPackage ファイル `training-data.gpkg` から `roads` レイヤを追加してください。やり方については **★☆☆（初級レベル）理解しよう**: *GeoPackage* **データベースからベクタデータを読み込む** を参照してください。

2. `exercise_data` ディレクトリに `routes.shp` という ESRI シェイプファイルのラインデータセットを新規に作成し、 `id` と `type` という属性を設定します（上記の方法を参考にしてください）

3. Activate edit mode on the `routes` layer.

4. ライン機能を使用しているので、 線の地物を追加 ボタンをクリックし、ラインデジタイズモードを開始します。

5. 一度に 1 つずつ、 `routes` レイヤにある細道と小道をデジタイズします。できるだけ正確にルートをたどるようにし、角やカーブに沿って点を追加します。

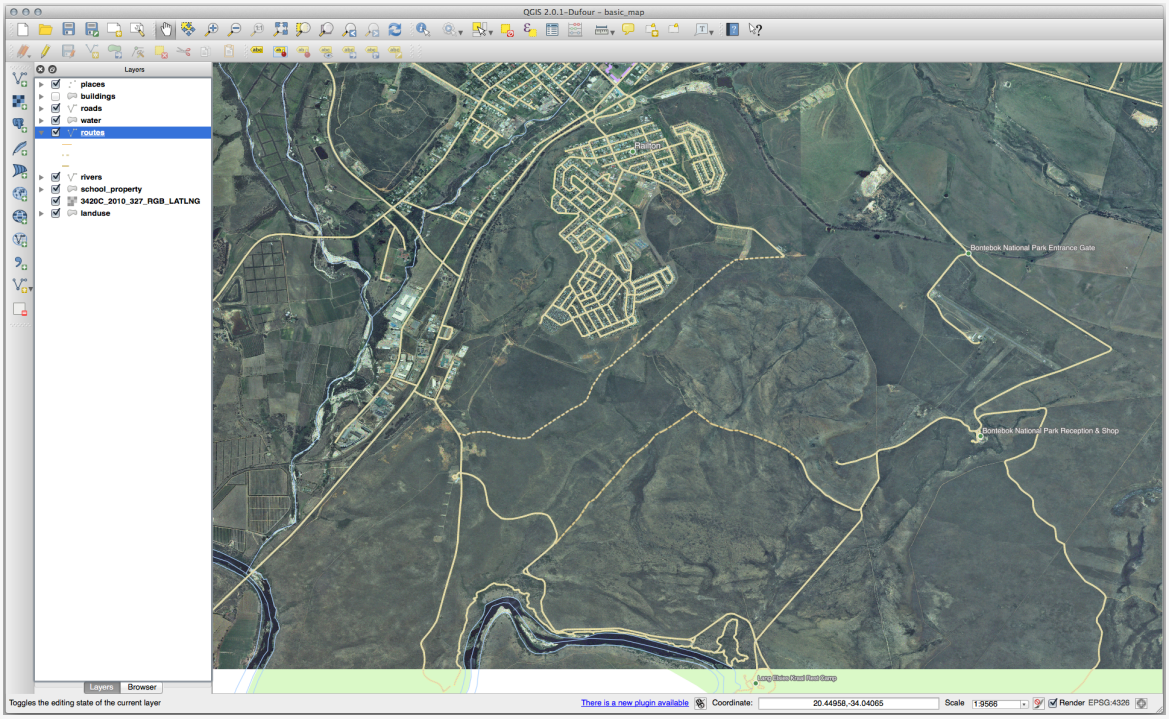
6. `type` 属性の値を `path` または `track` に設定します。

7. **レイヤプロパティ** ダイアログを使って、ルートにスタイルを追加します。細道と小道で異なるスタイルを自由に使ってください。

8. 編集内容を保存し、  編集モードを切り替え ボタンを押して、編集モードをオフにします。

答え

シンボロジは重要ではありませんが、結果は大体このようになるはずです：



5.1.6 結論

これで地物を作成する方法がわかりましたね！このコースではポイント地物の追加についてカバーしていませんが、より複雑な地物（ラインとポリゴン）で作業してきましたから実際に必要ありません。ポイントを置きたいところを 1 回クリックし、いつものように属性を与え、その後地物が作成されること以外、まったく同じ動作です。

デジタイズは GIS プログラムでは非常に一般的な活動ですので、方法を知ることは重要です。

5.1.7 次は?

GIS レイヤーの地物は、単なる画像ではなく、空間内のオブジェクトです。例えば、隣接するポリゴンは、互いの位置関係を知っています。これは、 **トポロジー** と呼ばれています。次のレッスンでは、これがなぜ便利なのか、その例を見てみましょう。

5.2 レッスン: 地物のトポロジ

オーバーラップやギャップなどのエラーを最小限に抑えるため、トポロジはベクタデータレイヤの有用な側面です。

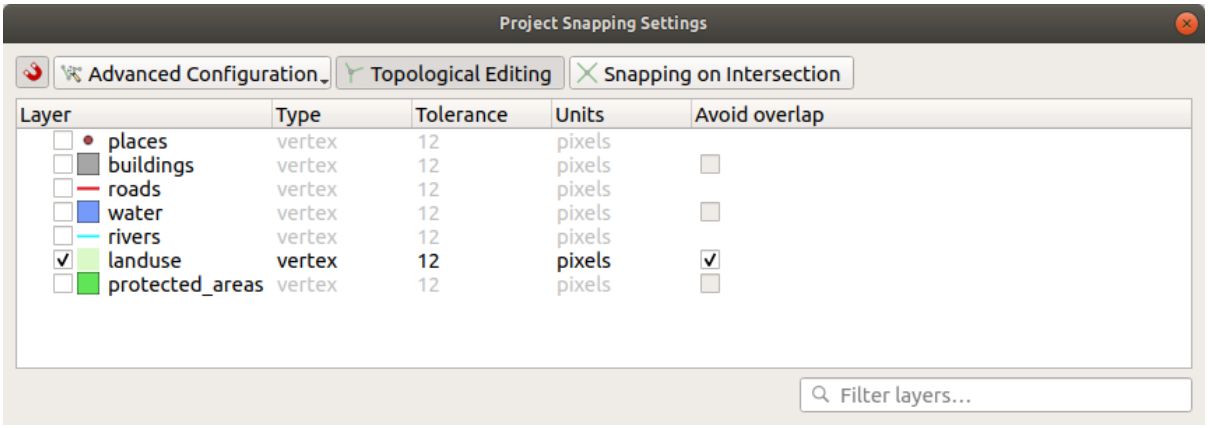
たとえば、2 つの地物が境界を共有し、トポロジを使用してその境界を編集する場合、はじめの地物を編集し、そしてもう 1 つの地物を編集して、それらが合うように慎重に境界を描く必要はありません。それらの共有された境界を編集でき、両方の地物は同時に変化します。

このレッスンの目標: 例を用いてトポロジを理解します。

5.2.1 ★★☆ （中級レベル） 理解しよう: スナップする

スナップはトポロジ編集を容易にします。これによりデジタイズ中にマウスカーソルを他のオブジェクトにスナップさせることができます。スナップオプションを設定するには：

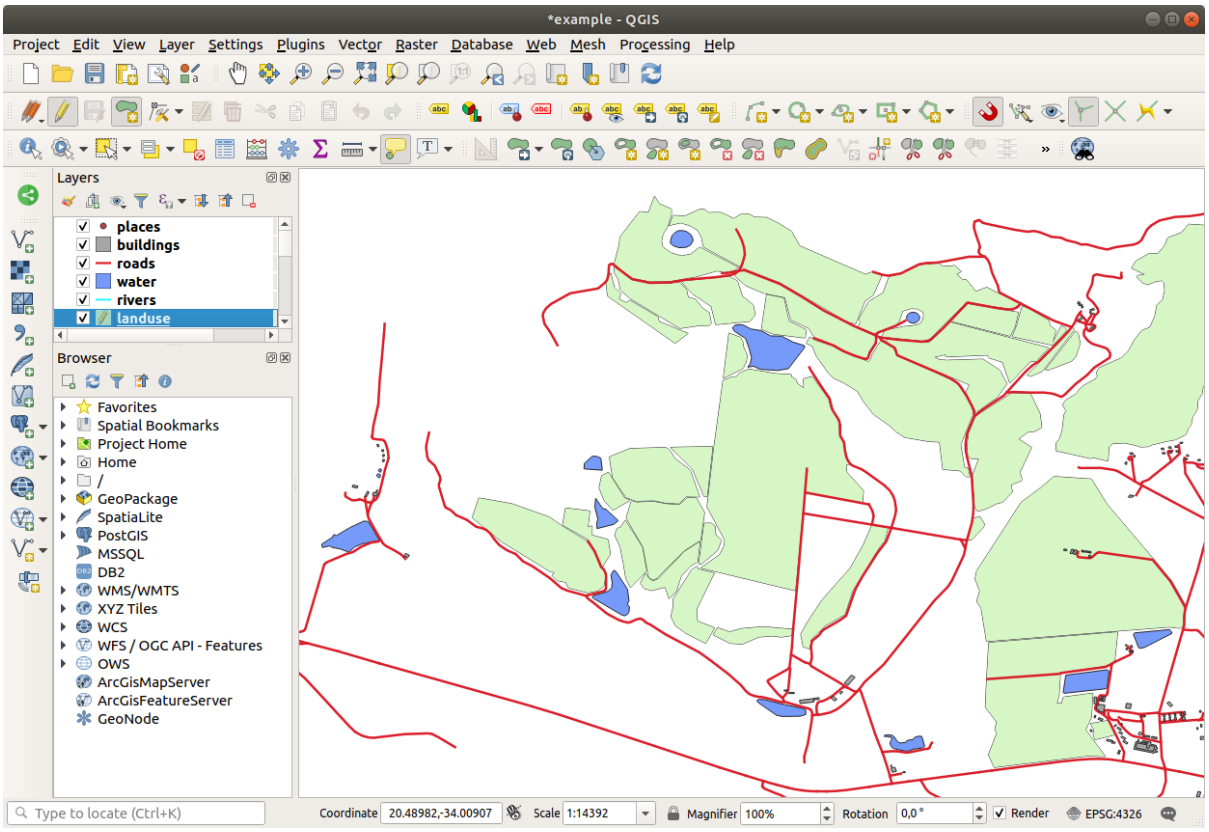
1. メニュー項目 **プロジェクト** **スナップオプション**... に移動します。
2. **スナップ設定** を設定します:  スナップを有効化 ボタンを使ってスナップを有効にし、**詳細構成** を選択、*landuse* レイヤを **型** **頂点** と許容範囲 12 ピクセル で有効にします:



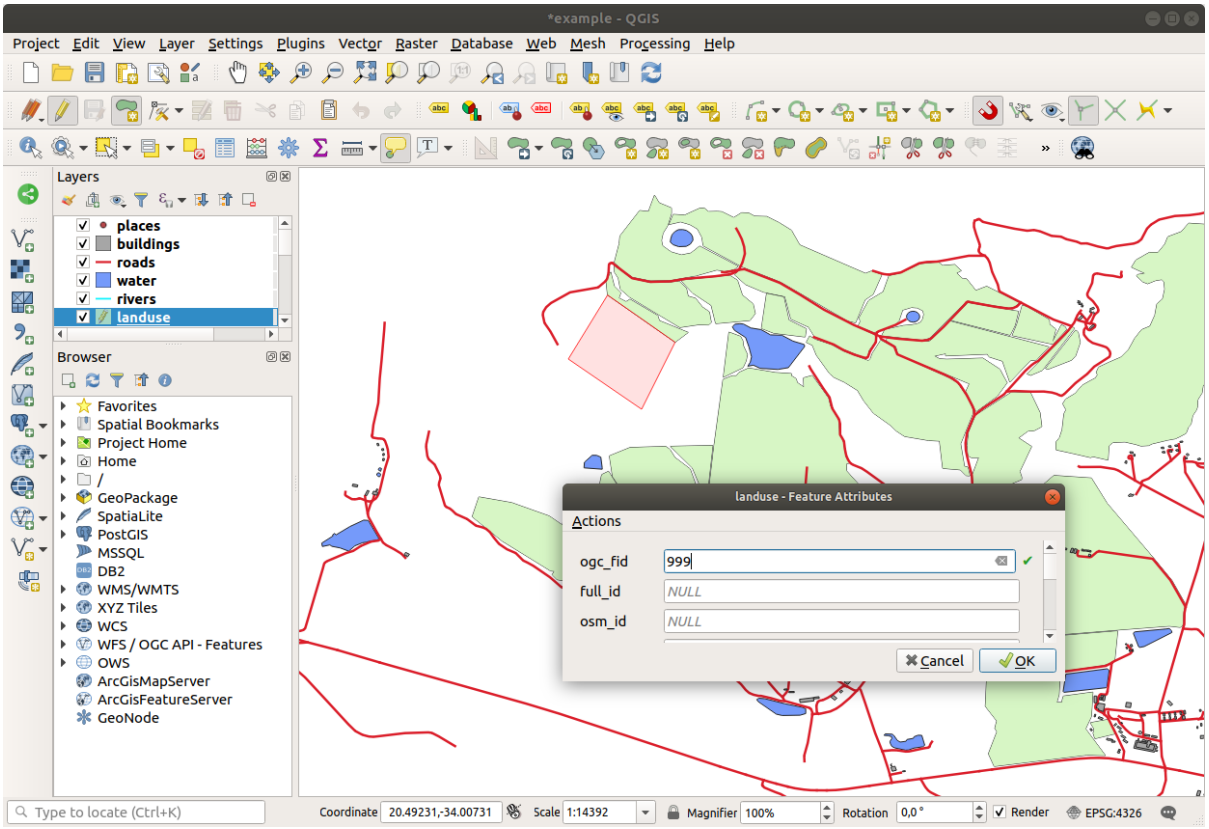
3. **重なりを避ける** 欄のボックスがチェックされていることを確認します。
4. ダイアログを閉じます。
5. Select the `landuse` layer and enter edit mode ()

6. ツールバーの **高度なデジタイズ** が有効になっていることを確認します（ **ビュー　ツールバー** の項を参照)。

7. このエリアにズームします (必要に応じてレイヤとラベルを有効にします):



8. この赤で示した新しい（架空の）区域をデジタイズします:



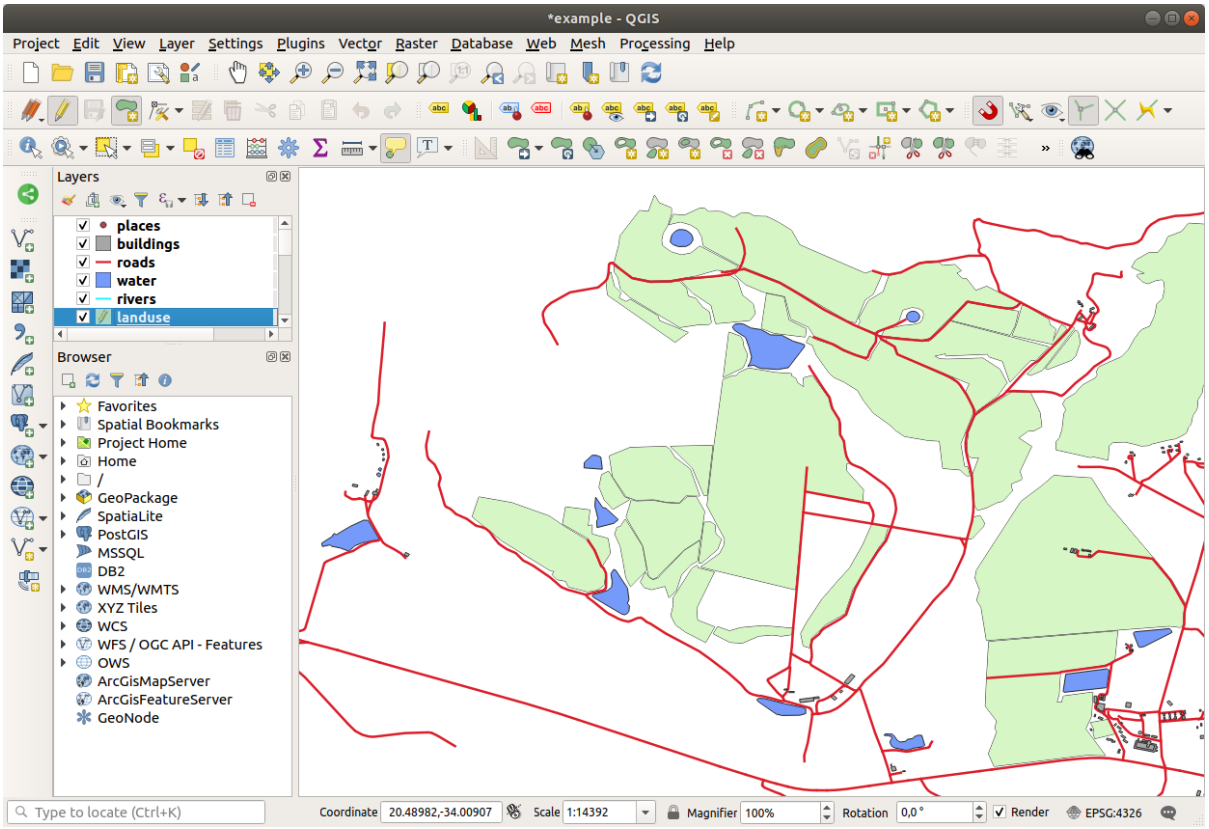
9. プロンプトが表示されたら、*OGC_FID* に 999 を与えます。しかし、他の値は変更しなくても大丈夫です。

   注意深くデジタイズしていて、カーソルを隣接する領域の頂点にスナップさせるようにすると、既存の隣接区域との間に隙間ができないことに気がつくはずです。

10. **高度なデジタイズ** ツールバーの  元に戻す と  やり直す ツールに注目してください。

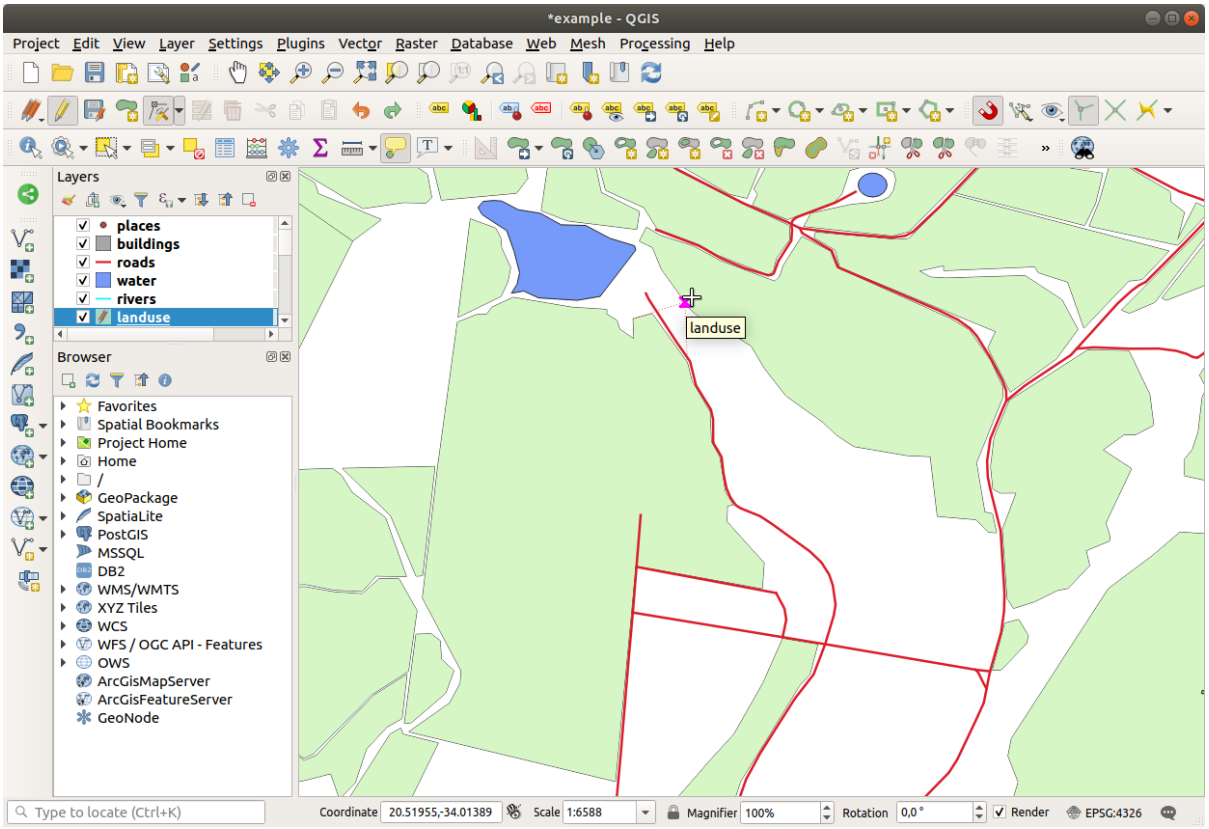
5.2.2 ★★☆ （中級レベル） 理解しよう: トポロジ的地物を修正

トポロジ地物の更新が必要な場合があります。今回の調査区域では、ある区域が森林になったので、`landuse` レイヤを更新する必要があります。そこで、この区域の森林地物を拡大し、結合することにします。



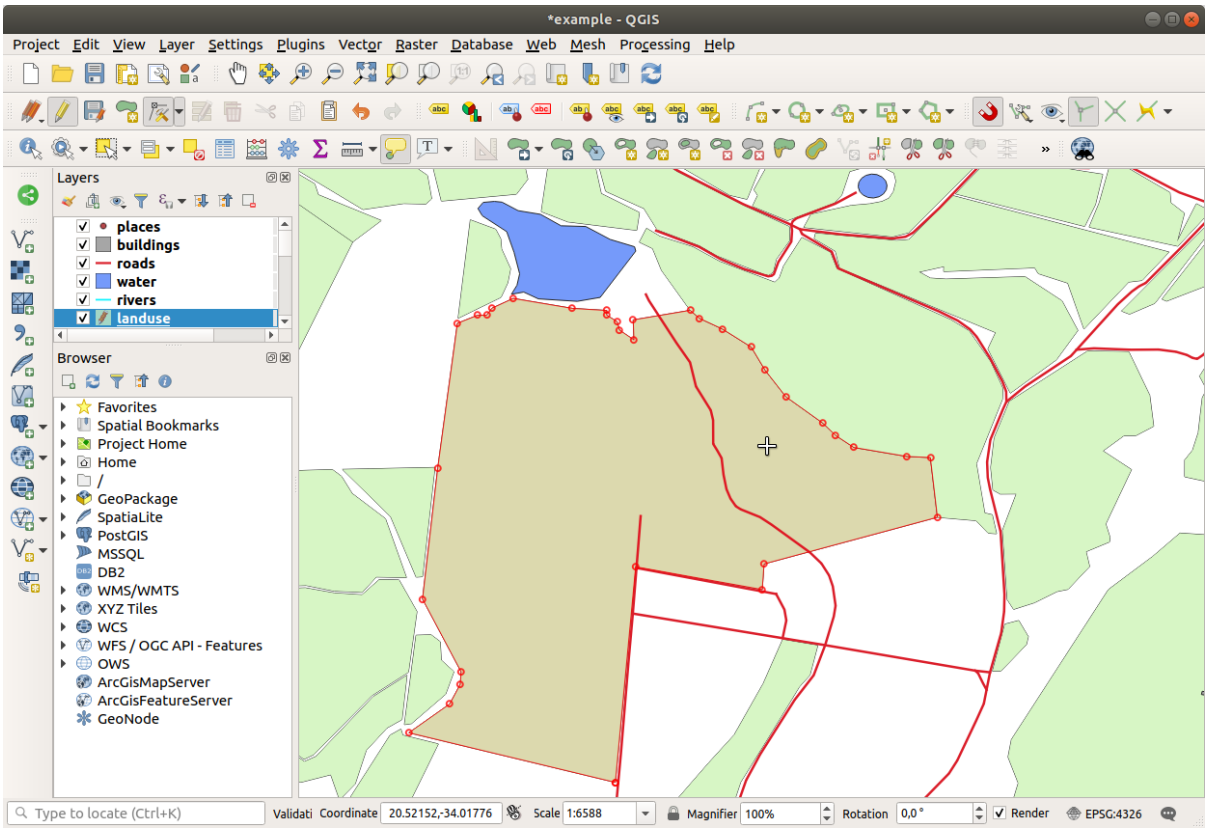
森林区域を結合するために新しいポリゴンを作成するのではなく、**頂点ツール** を使って既存のポリゴンを編集して結合します。

1. 編集モードに入ります（まだアクティブになっていない場合
2.  頂点ツール ツールを選択します。
3. 森林の区域を選択し、頂点を選択し、隣接する頂点に移動して、2 つの森林地物が出会うようにします：



4. 他の頂点をクリックし、所定の位置にスナップさせます。

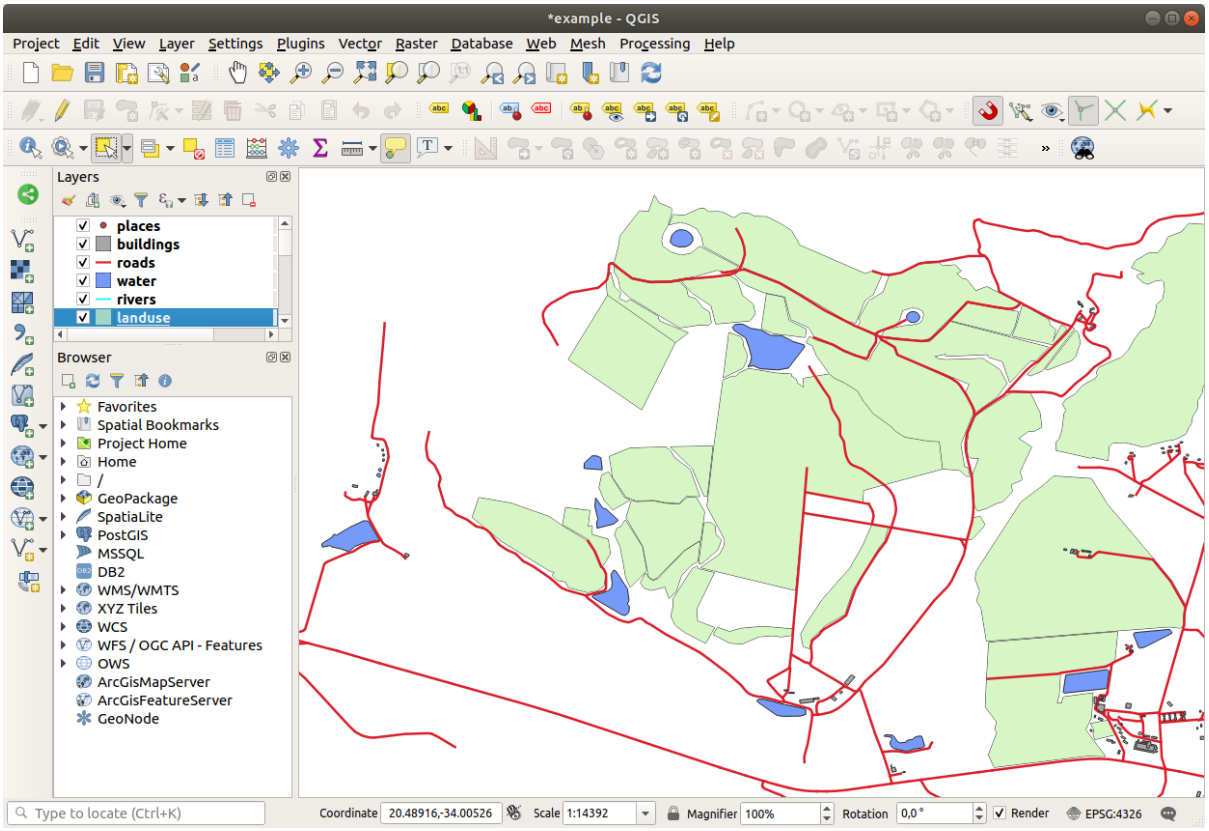
   トポロジとして正しい境界線はこのようになります:



続けて、**頂点ツール** を使って、さらにいくつかの区域を結合します。

また、 ポリゴン地物を追加 ツールを使用して、2 つの森林ポリゴンの間のギャップを埋めることができます。 **重なりを避ける** を有効にすると、頂点を一つ一つ追加する必要はありません。新しいポリゴンが既存のポリゴンに重なる場合、自動的に追加されます。

例題のデータを使用する場合、次のような森林区域があるはずです:

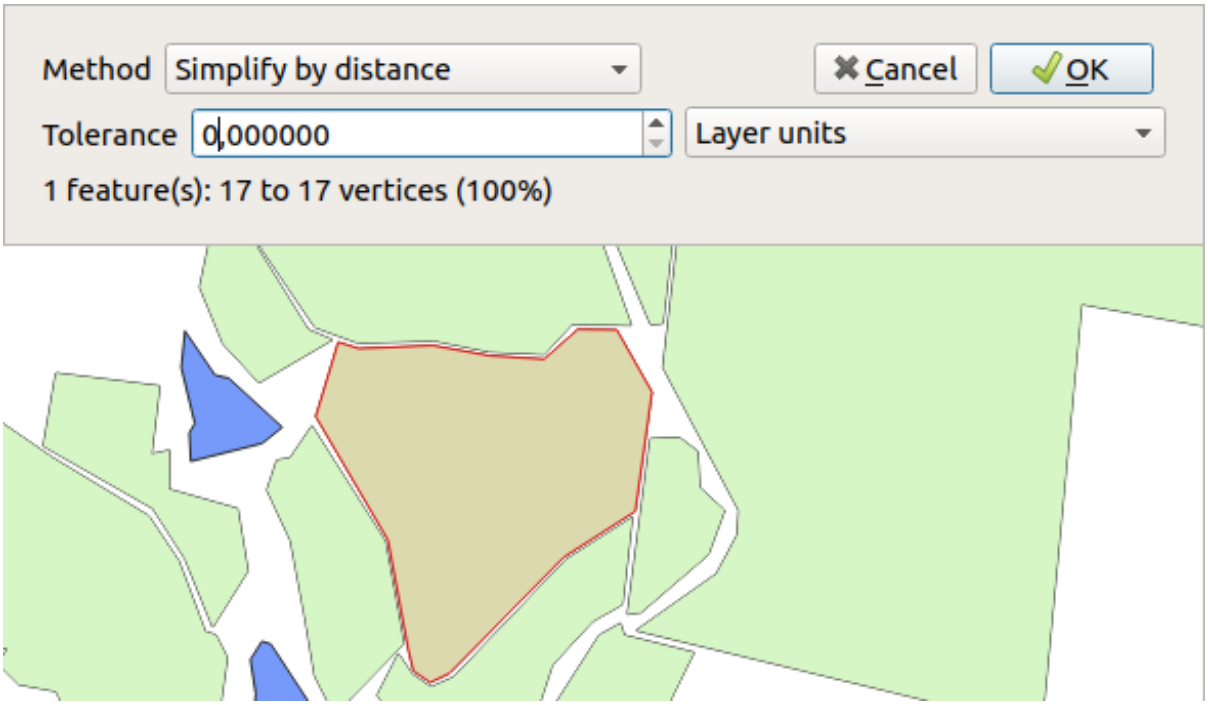


あなたが結合した森林の区域がより多くても、より少なくても、違う区域であったとしても気にしないで下さい。

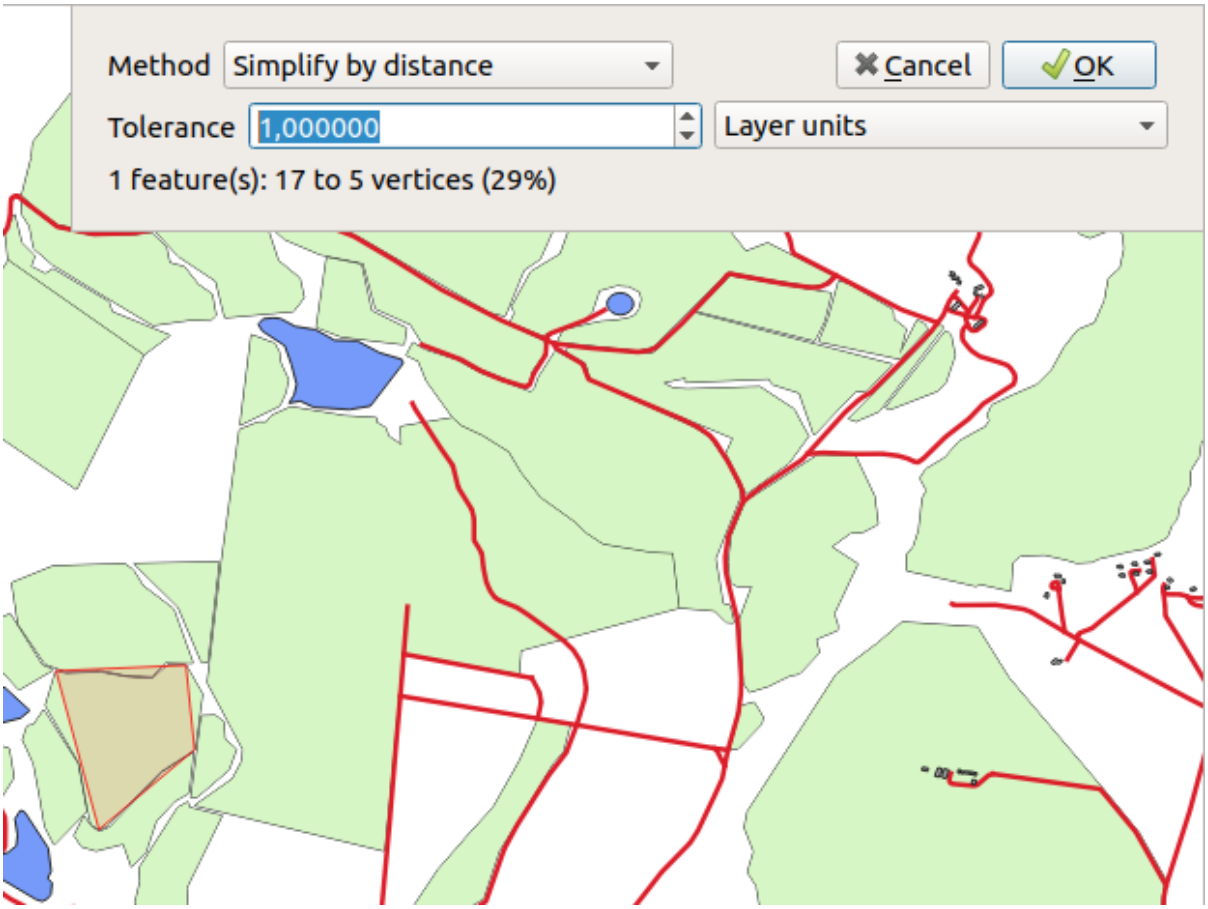
5.2.3 ★★☆ （中級レベル） 理解しよう: ツール: 地物の簡素化

同じレイヤで続けて、 地物を簡素化 ツールをテストしてみましょう：

1. それをクリックしてアクティブにします。
2. Click on one of the areas which you joined using either the Vertex Tool or Add Polygon Feature tool. You will see this dialog:



3. **許容度** を変えて何が起きるか見てみましょう:



これにより、頂点の数を減らすことができます。

4. *OK* をクリックします

このツールの利点は、一般化するためのシンプルで直感的なインターフェイスを提供していることです。しかし、このツールはトポロジを台無しにしていることに注意してください。単純化されたポリゴンは、本来あるべき隣接するポリゴンとの境界を共有しなくなるのです。そのため、このツールは独立した地物に適しています。

次へ進む前に、最後の変更を元に戻すことでポリゴンを元の状態に戻します。

5.2.4 ★★☆ （中級レベル） 自分でやってみよう: ツール: リングの追加



ツールを使用すると、ポリゴン地物に内部リングを追加（ポリゴンに穴を開ける）することができます。その穴はポリゴン内に完全に含まれている必要がありますが、境界に触れていても問題ありません。例えば、南アフリカの外周をデジタイズし、レソトの穴を追加する必要がある場合、このツールを使用することになります。

このツールを使って実験してみると、スナップオプションがポリゴン内にリングを作るのを邪魔していることに気づくかもしれません。そのため、穴を開ける前にスナップをオフにすることをお勧めします。

1. 

ボタン（またはショートカット `s`）を使って `landuse` レイヤのスナップを無効化します。
2. 

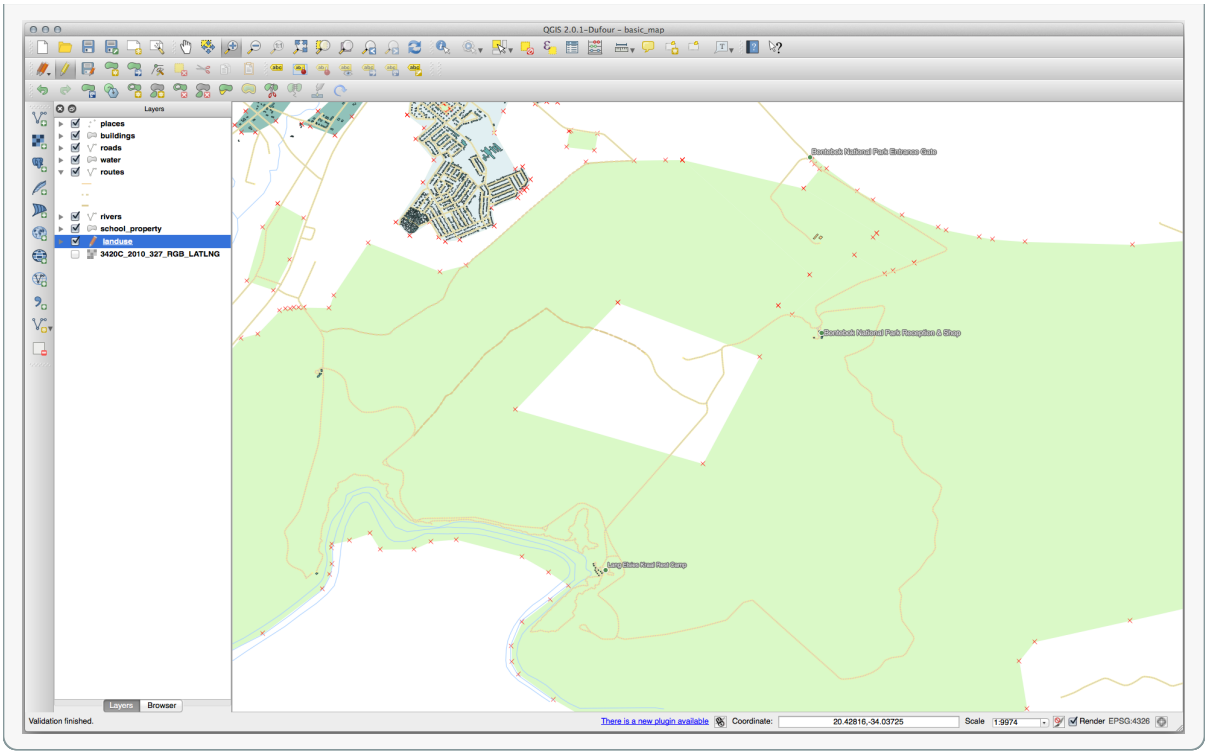
ツールを使ってポリゴンジオメトリの中央に穴を開けます。
3. 

ツールを使っているかのように、対象の地物の上にポリゴンを描画します。
4. 右クリックすると穴が見えるようになります。
5. 

ツールを使って、先ほど作成した穴を削除します。穴の内側をクリックすると、穴が削除されます。

答え

正確な形状は重要ではありませんが、このように地物の中央に穴が開いているはずです：



- 編集を元に戻してから、次のツールの演習を続けてください。

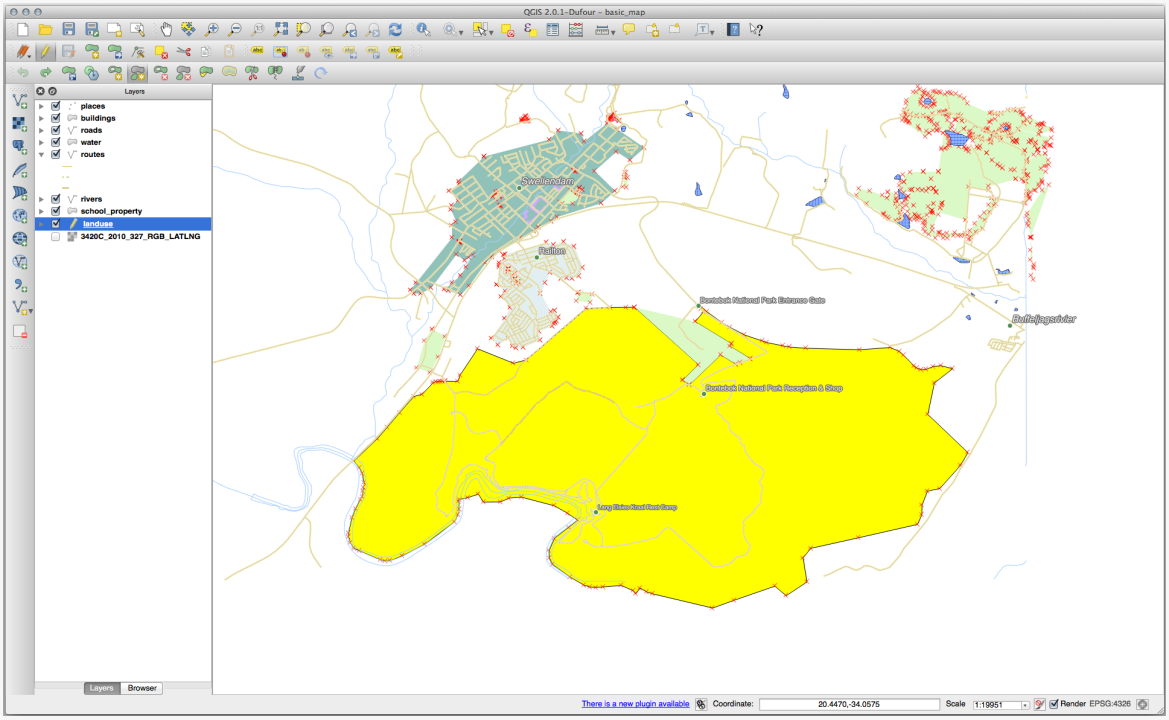
5.2.5 ★★☆ （中級レベル） 自分でやってみよう: ツール: 部分の追加

部分を追加 ツールを使用すると、主地物に直接接続されていない新しい部分を地物に追加することができます。例えば、南アフリカ共和国本土の境界線をデジタル化したが、プリンスエドワード諸島をまだ追加していない場合、このツールを使用して作成します。

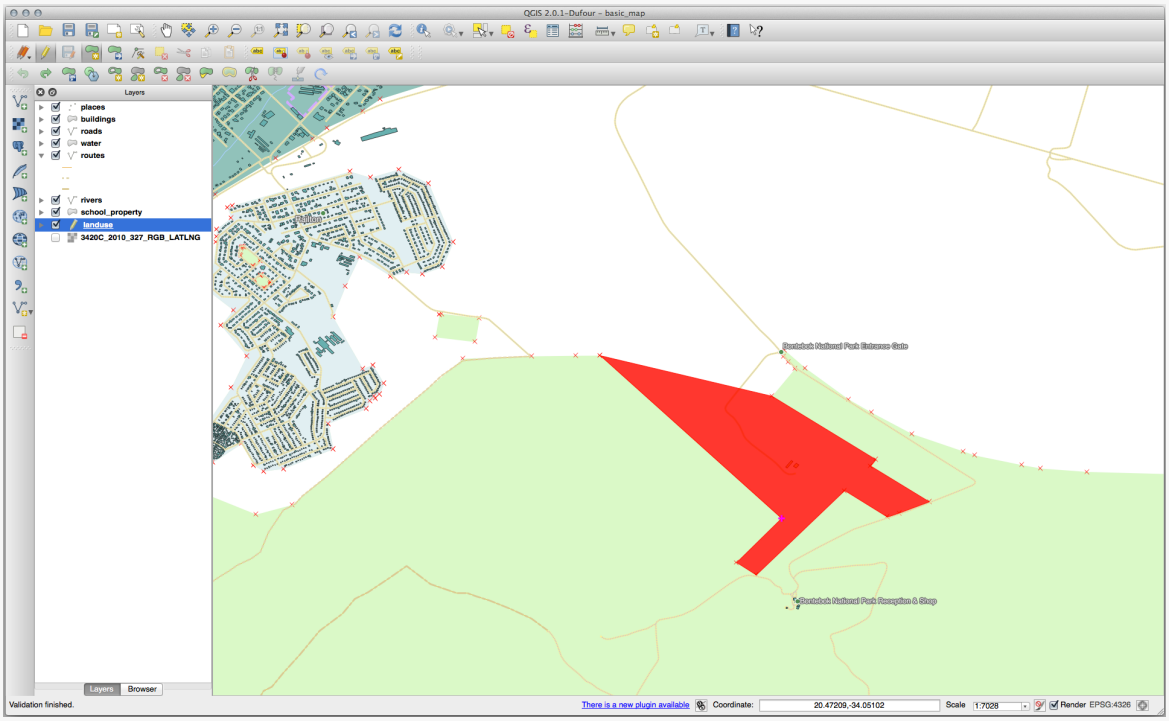
1. 地物を選択 ツールを使って、部分を追加したいポリゴンを選択します。
2. Use the Add Part tool to add an outlying area.
3. 部分の削除 ツールを使って、今作成した部分を削除します。

> **注釈**
>
> 部分の内側をクリックして削除します。

> **答え**
>
> 1. まず、Bontebok National Park を選択します:



2. 新しい部分を追加します：



3. 編集を元に戻してから、次のツールの演習を続けてください。

5.2.6 ★★☆ （中級レベル） 理解しよう: ツール: 地物の変形

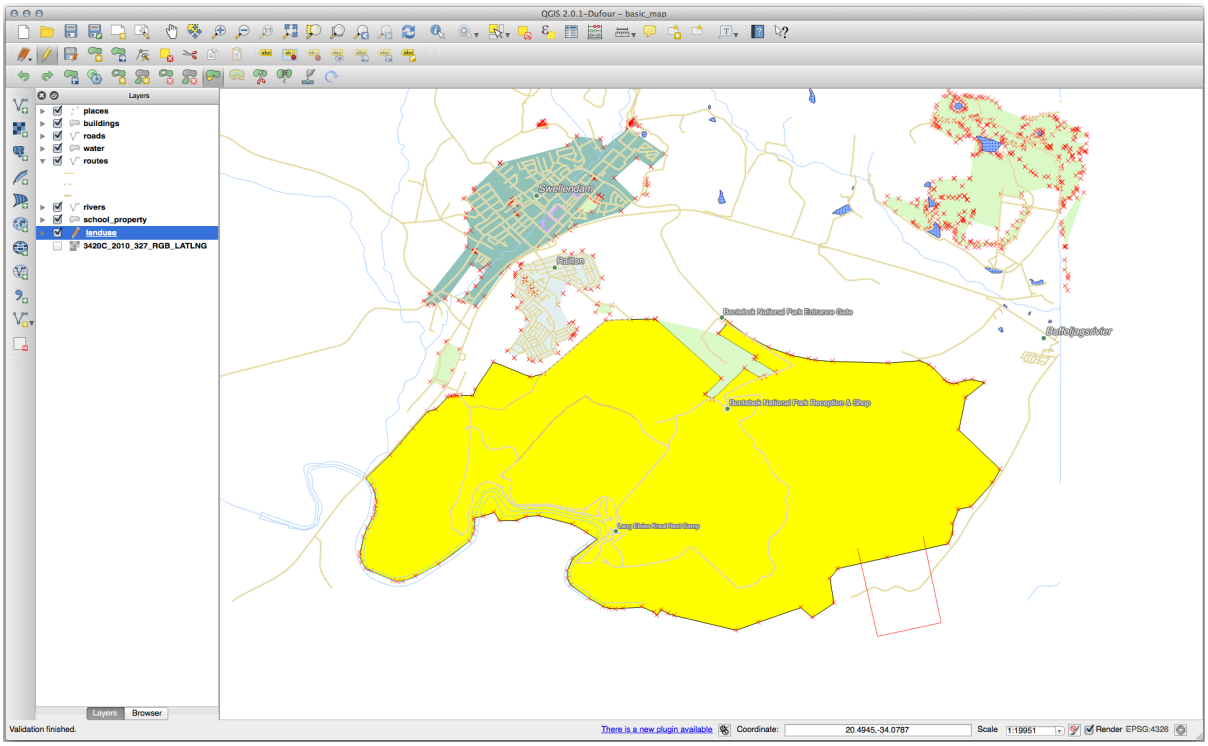


ツールは、ポリゴン地物を拡張したり、その一部を（境界線に沿って）切り取るために使用します。

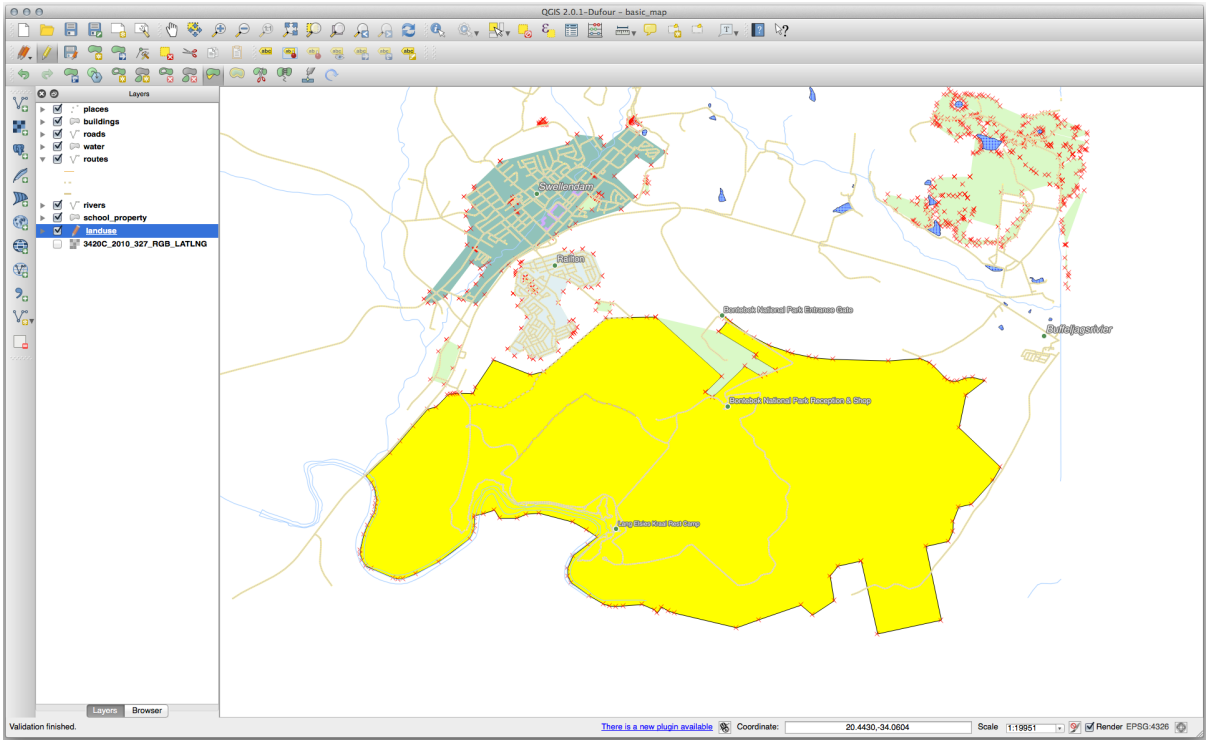
拡張:

1. 

ツールを使ってポリゴンを選択します。
2. ポリゴンの中を左クリックして、描画を開始します。
3. ポリゴンの外側に図形を描きます。最後の頂点はポリゴンの内側にしてください。
4. 右クリックして形状を完成させます：

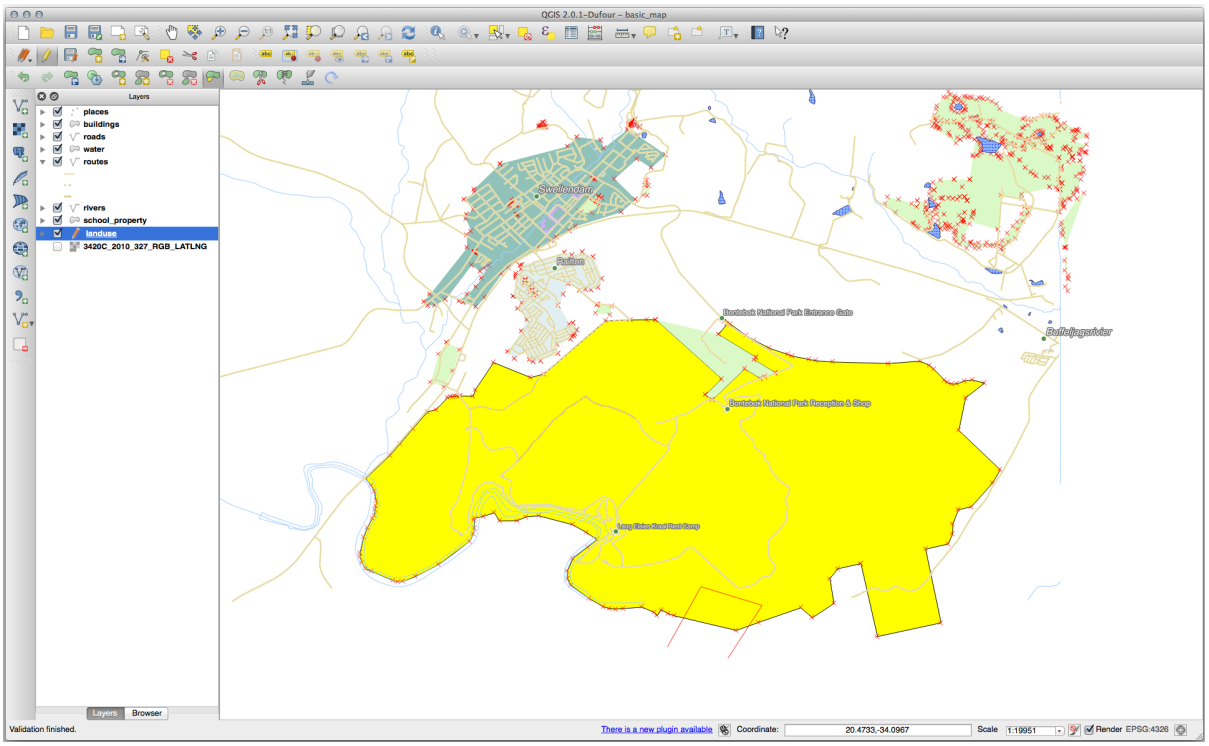


これは次のような結果になります:

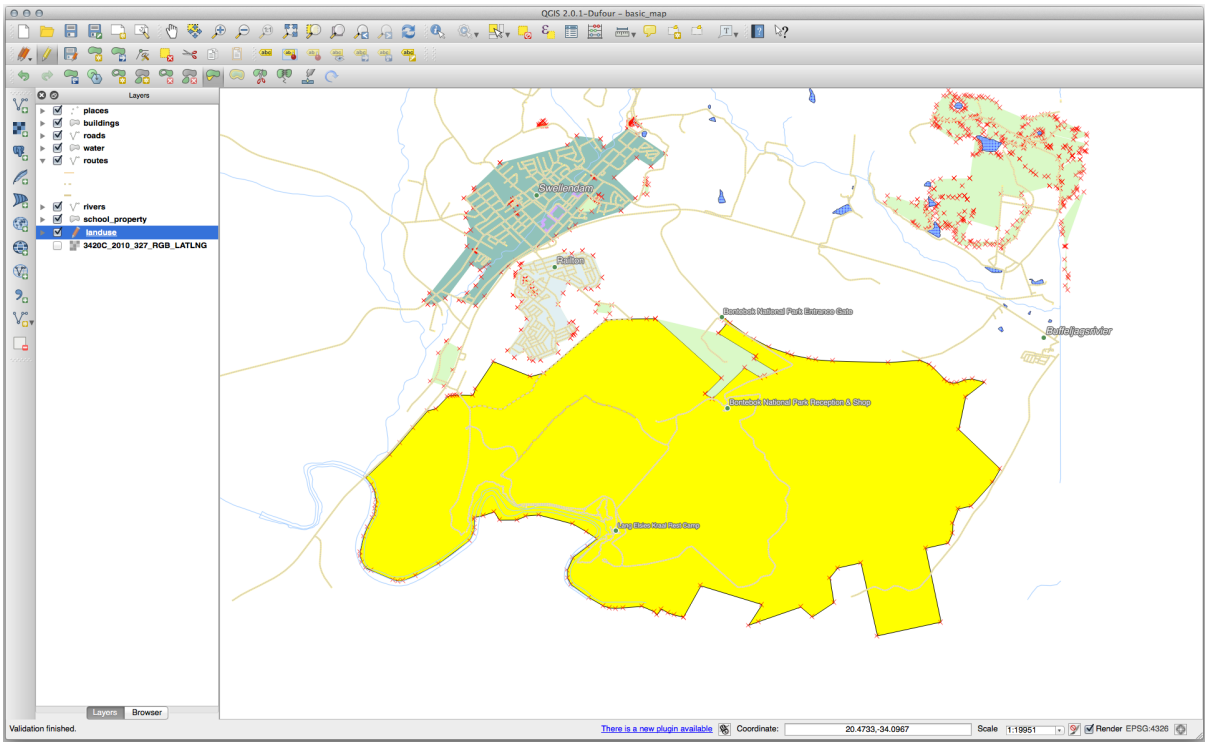


部分を切り取る：

1.  地物を選択 ツールを使ってポリゴンを選択します。
2. ポリゴンの外側をクリックします。
3. ポリゴンの内側に図形を描きます。最後の頂点はポリゴンの外に戻っている必要があります。
4. ポリゴンの外で右クリックします：



以上の結果:

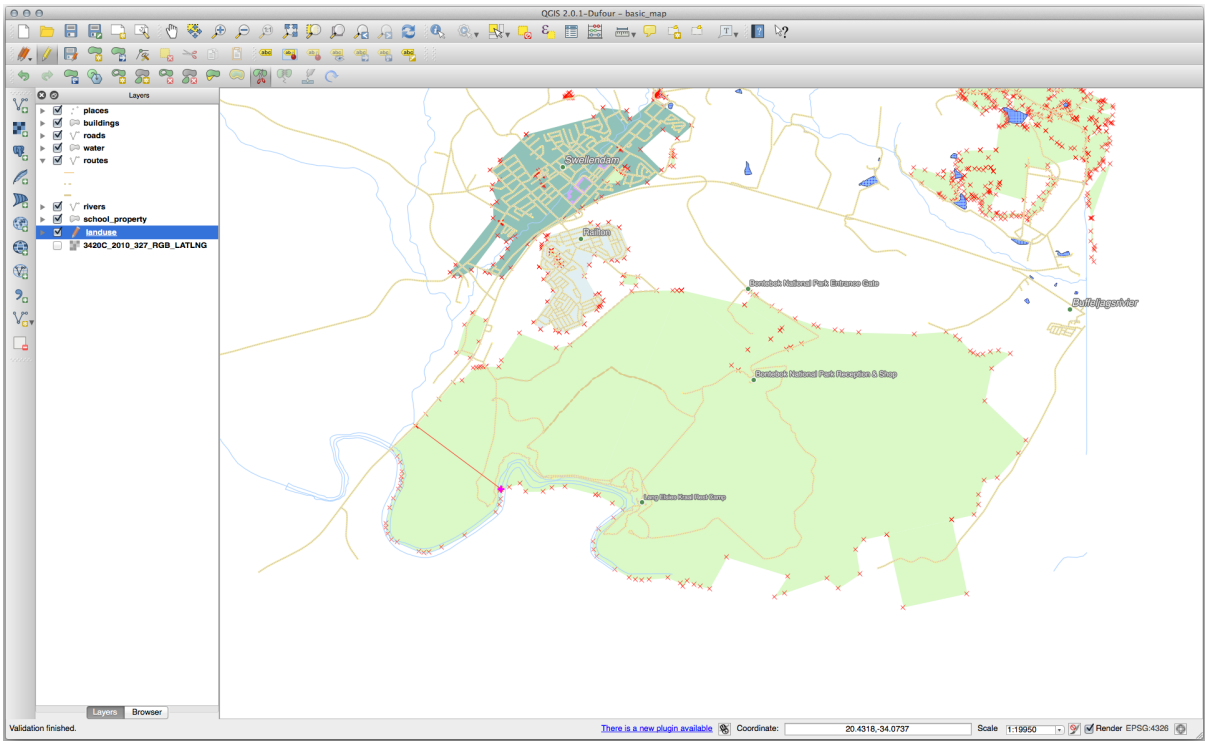


5.2.7 ★★☆ （中級レベル） 自分でやってみよう: ツール: 地物の分割

地物を分割 ツールは 地物を変型 ツールと似ていますが、2 つの部分のどちらかを削除しない点が異なります。そのかわり、その両方を保持します。

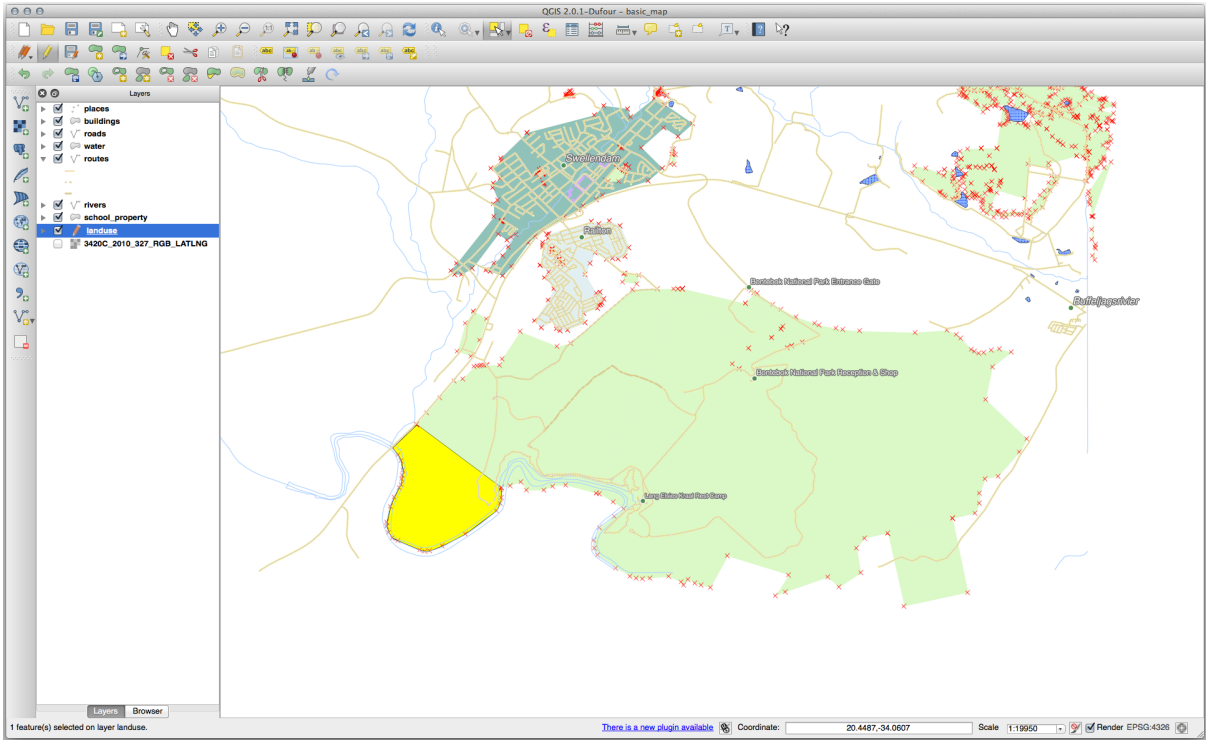
このツールを使って、ポリゴンから角を分割してみます。

1. まず、 landuse レイヤを選択し、スナップを再度有効にします。
2. 地物を分割 ツールを選択し、頂点をクリックして線を描き始めます。
3. 境界線を引きます。
4. 分割したいポリゴンの「反対側」にある頂点をクリックし、右クリックで線を完成させます：



5. この時点では、何も起こっていないように見えるかもしれません。しかし、 landuse レイヤは境界線なしでレンダリングされるため、新しい分割線は表示されないことを覚えておいてください。

6.  地物を選択 ツールを使って、分割した部分を選択すると、新しい地物がハイライトされます:



5.2.8 ★★★ （上級レベル） 自分でやってみよう: ツール: 地物のマージ

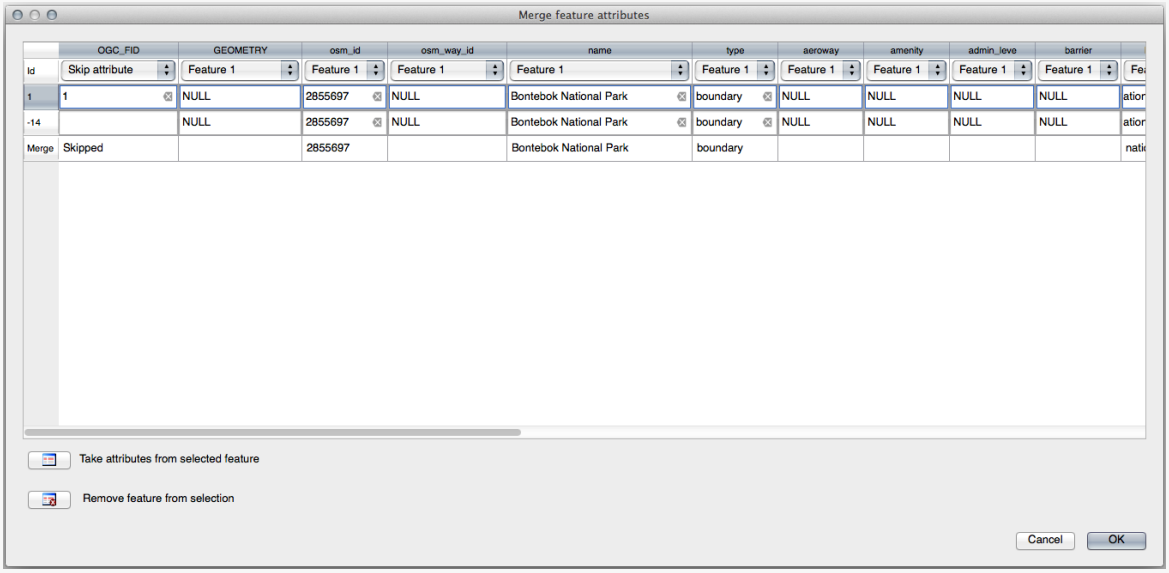
ここで、先ほど分割した地物をポリゴンの残りの部分に再度結合します：

1.  選択した地物を結合 と  選択地物の属性結合 ツールを使って実験してみましょう。
2. 相違点に注目してください。

答え

- Use the  Merge Selected Features tool, making sure to first select both of the polygons you wish to merge.
- 属性のソースとして、*OGC_FID* が 1 の機能を使用します（ダイアログでそのエントリーをクリックし、 **選択した地物から属性を取得する** ボタンをクリックします）：

別のデータセットを使用している場合、元のポリゴンの *OGC_FID* が 1 でない可能性が高くなります。そのような場合には、 *OGC_FID* が指定されている地物を選択します。



Using the  Merge Attributes of Selected Features tool will keep the geometries distinct, but give them the same attributes.

5.2.9 結論

トポロジ編集はトポロジの観点からの正しさを維持しながら迅速かつ容易にオブジェクトの作成や変更ができる強力なツールです。

5.2.10 次は?

オブジェクトの形状を簡単にデジタイズする方法はわかりましたが、属性の追加にはまだ頭を悩ませているようですね！ 次は、フォームを使って、属性編集をよりシンプルに、より効果的に行う方法を紹介します。

5.3 レッスン: フォーム

デジタイズで新しいデータを追加する場合、その地物の属性を入力するダイアログが表示されます。ただし、このダイアログボックスは既定ではあまり見た目がよくありません。これは特に大規模なデータセットを作成する場合や他の人にデジタイズを手伝って貰うときに既定のフォームではわかりにくい場合に、有用性の問題を引き起こす可能性があります。

幸いにも、QGIS ではレイヤに独自のカスタムダイアログを作成できます。このレッスンではその方法について説明します。

このレッスンの目標: レイヤのフォームを作成します。

5.3.1 ★☆☆ （初級レベル） 理解しよう: QGIS のフォームデザイン機能を使う

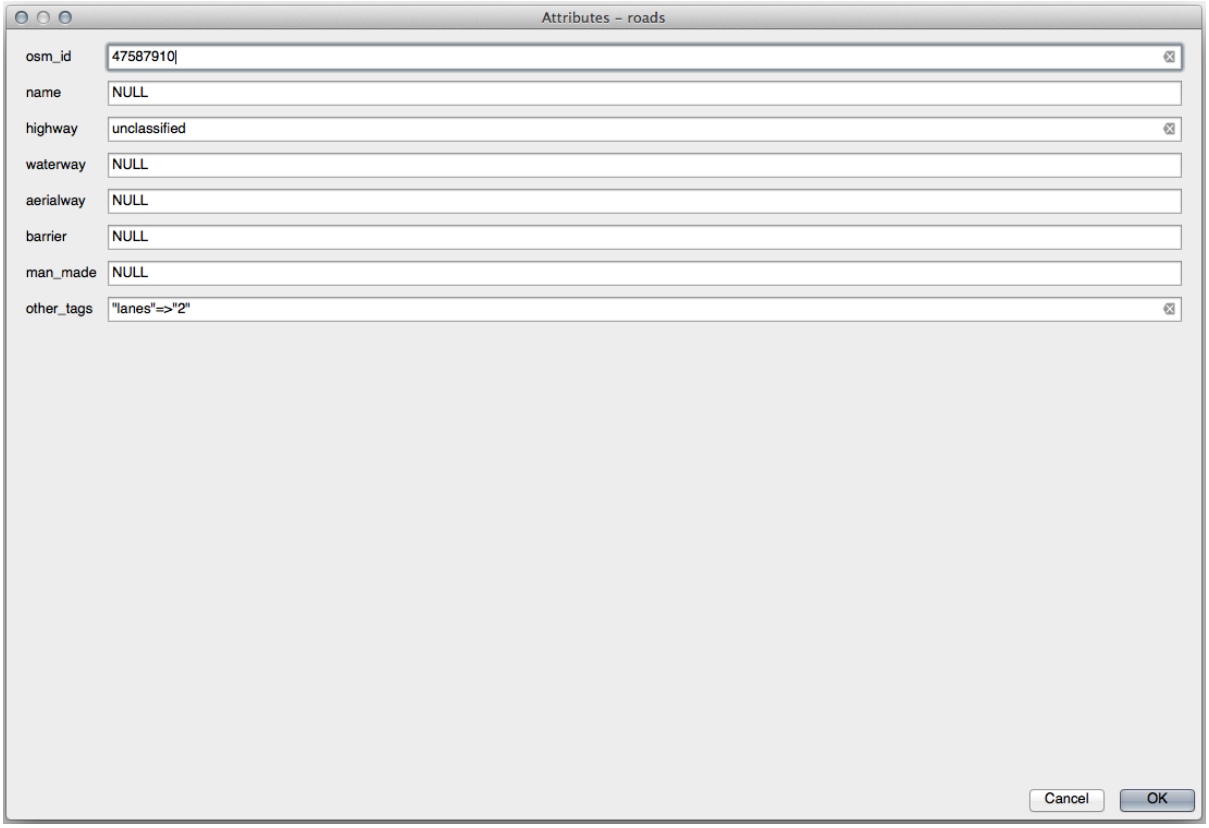
1. **レイヤ** パネルで roads レイヤを選びます
2. 前にやったように **編集モード** に入ります
3. roads レイヤの属性テーブルを開きます
4. テーブルのどこかのセルで右クリックします。 **フォームを開く** を含んだ短いメニューが現れます。
5. それをクリックして、QGIS がこのレイヤのために生成するフォームを見てみましょう

明らかに、毎回 **属性テーブル** で特定の街路を検索するのではなく、地図を見ながらこれを行うことができるといいですね。

1. **レイヤ** パネルで roads レイヤを選びます
2. 

ツールを使って、地図上の任意の街路をクリックします。
3. **地物情報** パネルが開き、フィールドの値やクリックした地物に関する一般的な情報をツリー表示で確認することができます。
4. パネル上部の 

メニューから **単一地物の場合、自動でフォームを開く** チェックボックスをオンにします。
5. さて、地図内のどれかの街路をもう一度クリックします。前の **地物情報** ダイアログに沿って今やおなじみのフォームが表示されます：

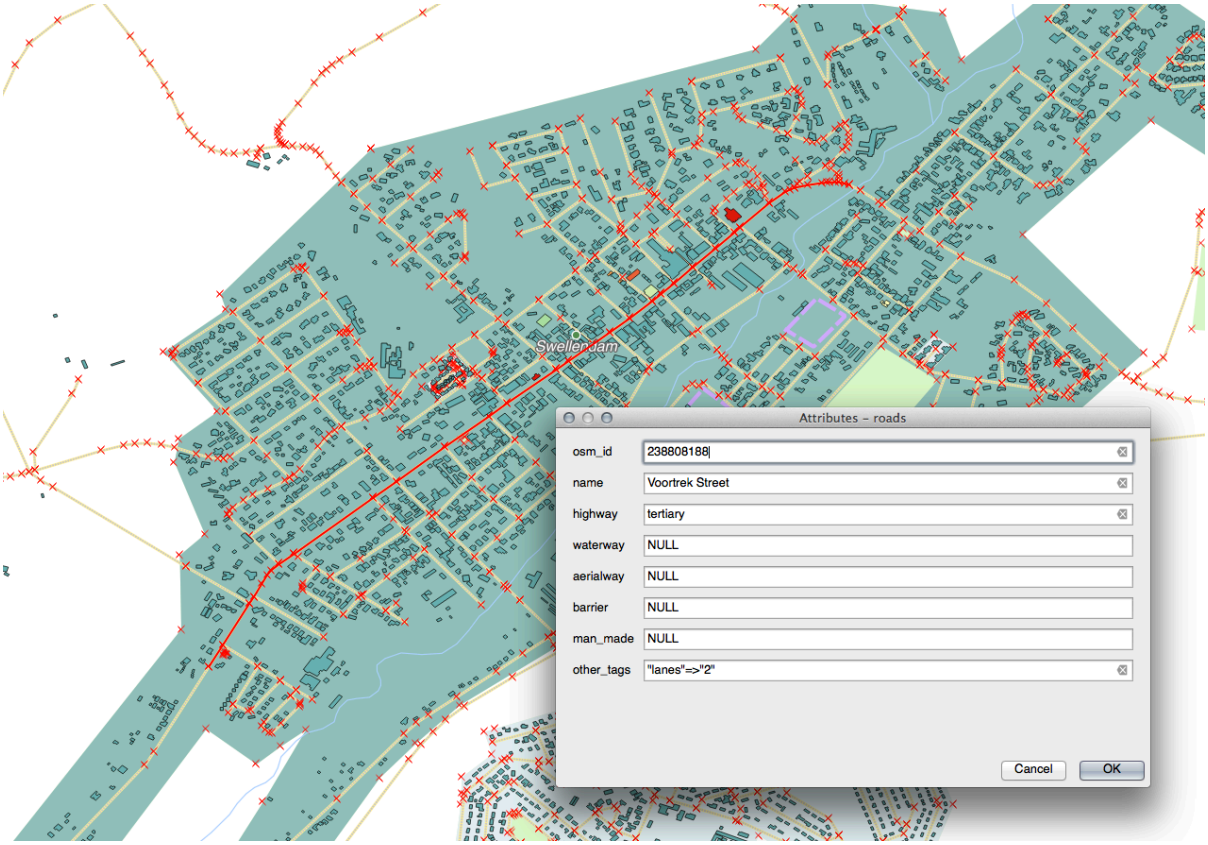


6. Each time you click on a single feature with the Identify Features tool, its form pops up as long as the *Auto open form* is checked.

5.3.2 ★☆☆ （初級レベル） 自分でやってみよう: フォームを使って値を編集する

編集モードの場合は、このフォームを使用して地物の属性を編集できます。

1. 編集モードをアクティブにします (まだアクティブになっていない場合)。
2. 地物情報表示 ツールを使って Swellendam を通る大通りをクリックします：

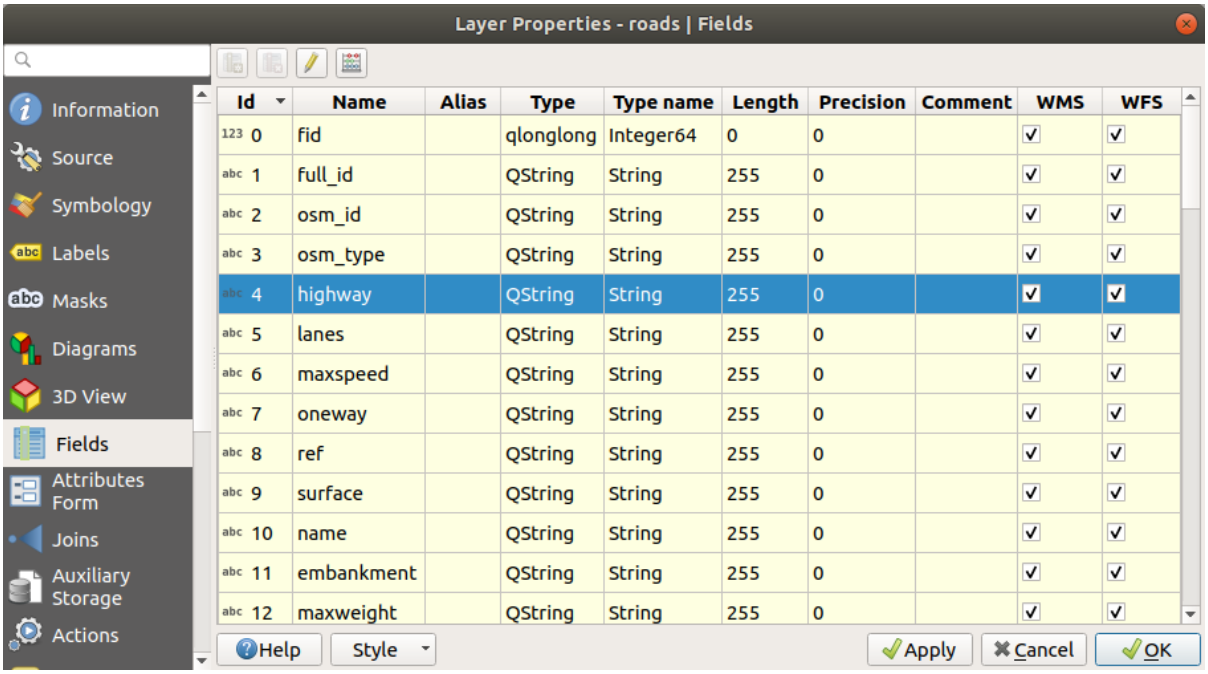


3. その *highway* の値を `secondary` に編集します
4. 編集モードを終了し、編集内容を保存します
5. **属性テーブル** を開くと、属性テーブルの値が更新されたことがわかります。つまり、ソースデータも更新されています。

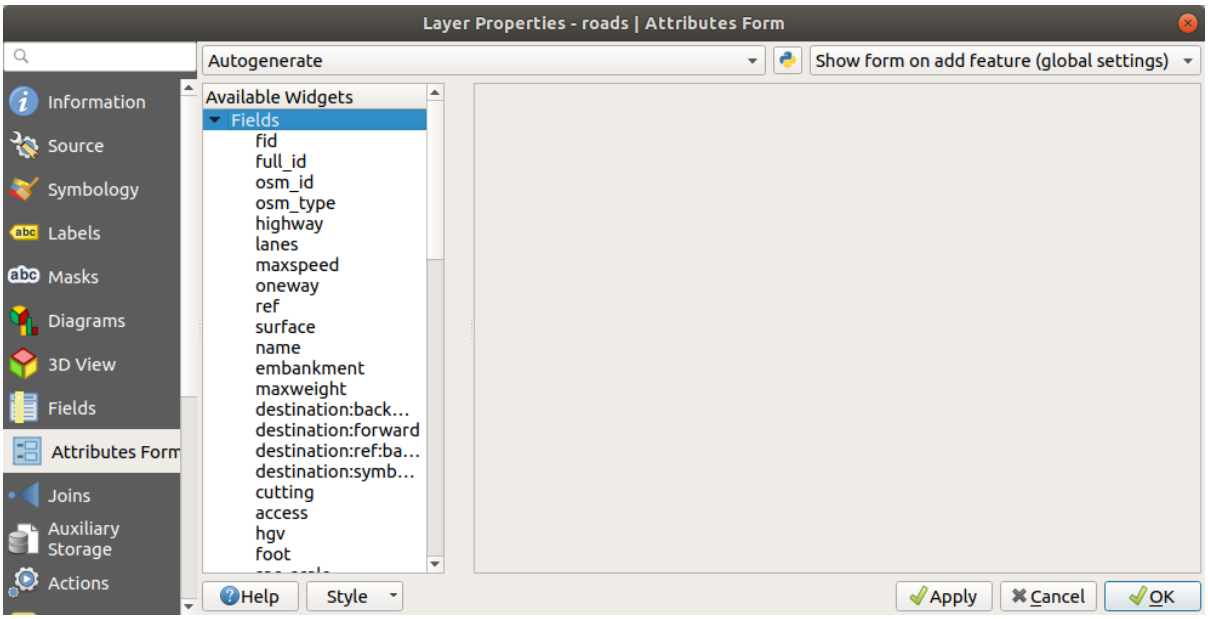
5.3.3 ★★☆ （中級レベル） 理解しよう: フォームのフィールドタイプを設定する

フォームを使用して編集するのはよいのですが、まだ何もかも手で入力しなければいけません。幸いにも、フォームには様々な方法でデータの編集ができる様々な種類の、いわゆる **ウィジェット** を持ちます。

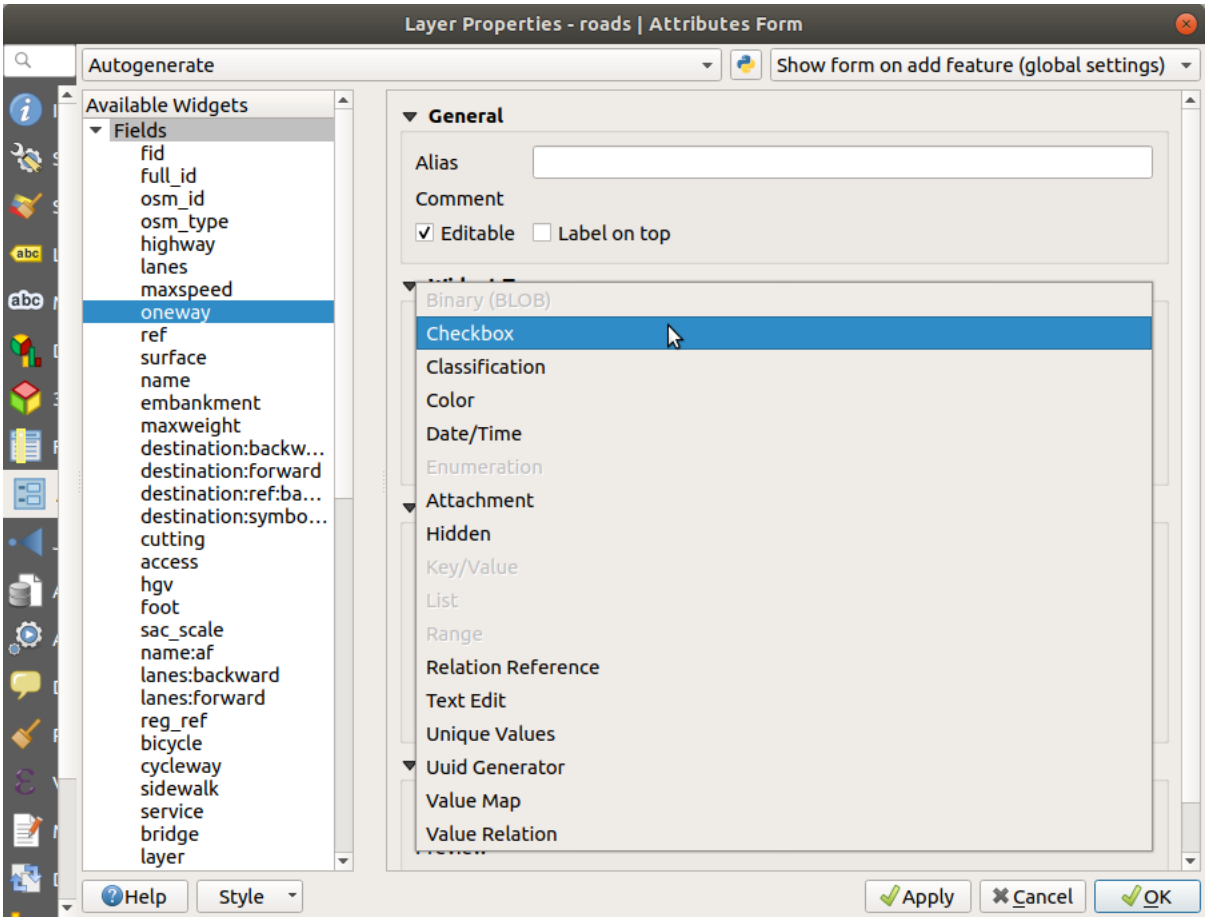
1. Open the `roads` layer's *Properties...*
2. **属性** タブに切り替えます。次が表示されます:



3. **属性フォーム** タブに切り替えます。このように表示されます：



4. oneway の行をクリックし、オプションのリストから **ウィジェットタイプ** として **チェックボックス** を選択します:



5. *OK* をクリックします
6. （もし `roads` レイヤが編集モードになっていなければ）編集モードに入ります
7.  地物情報表示 ツールをクリックします
8. 前に選んだのと同じ大通りをクリックします

これで *oneway* 属性の隣に `True` (チェック済み) または `False` (チェックなし) を示すチェックボックスが表示されることがわかるでしょう。

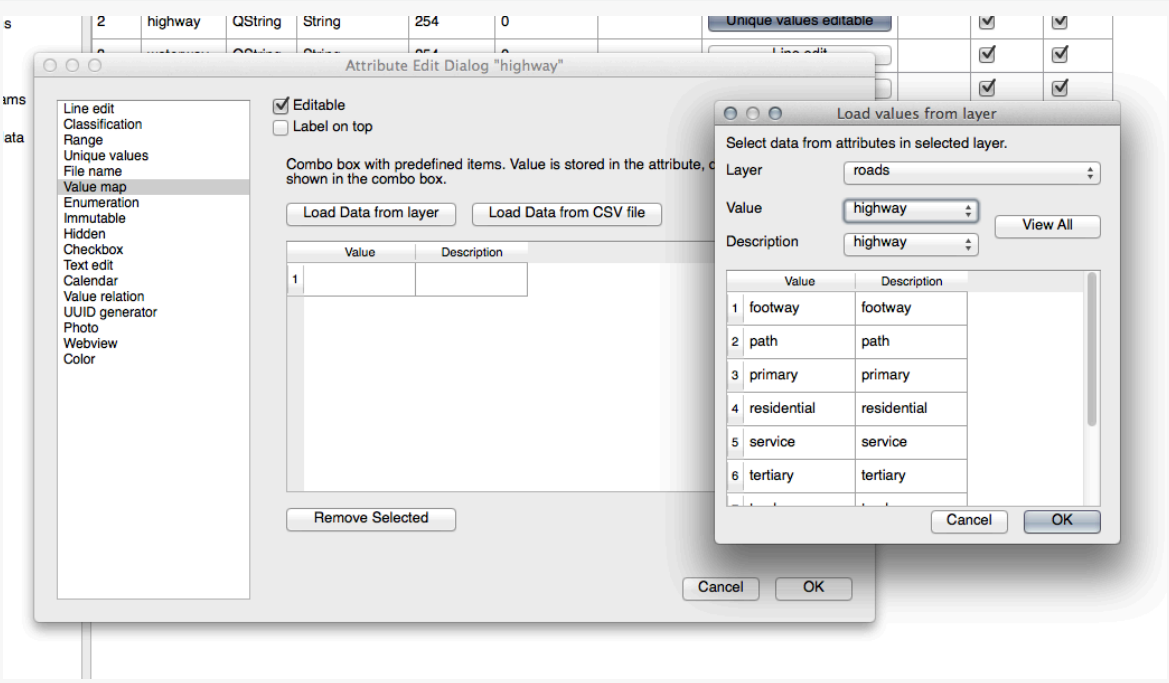
5.3.4 ★★★ （上級レベル） 自分でやってみよう:

highway フィールドに、より適切なフォームウィジェットを設定します。

答え

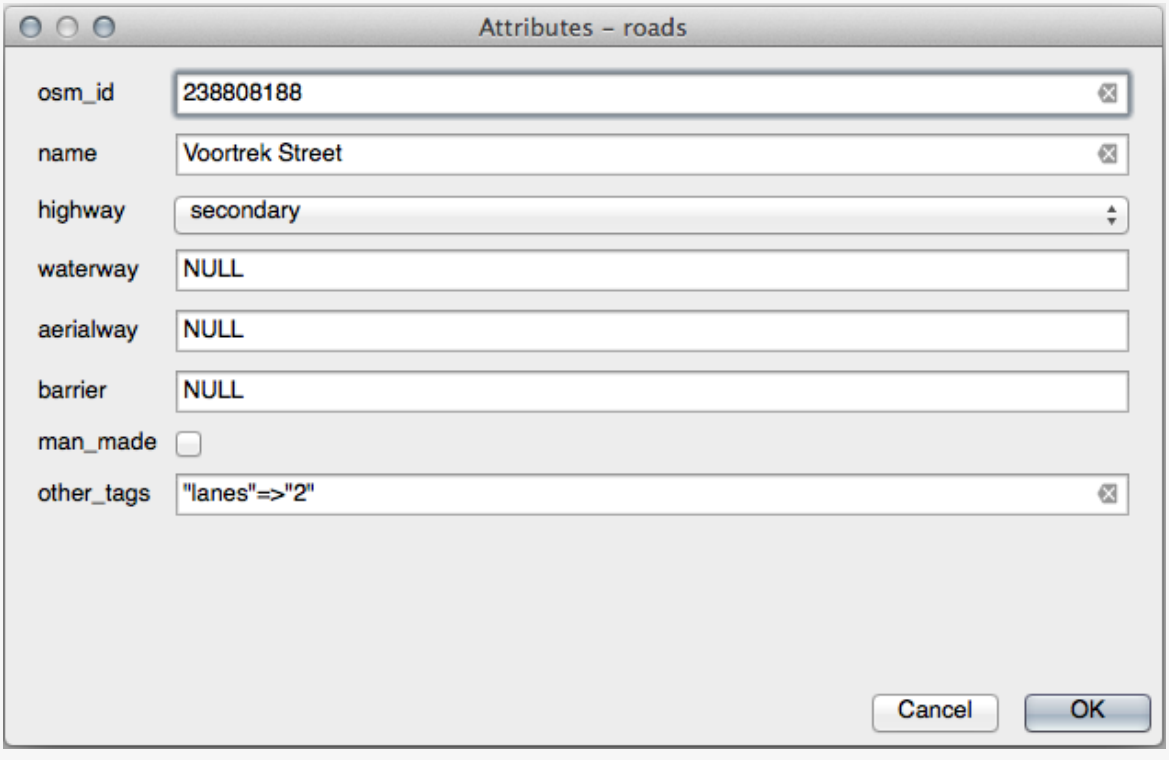
道路がとることのできるタイプは明らかに限られており、 *TYPE* について、このレイヤ属性テーブルを確認すると、それらはあらかじめ定義されていることが分かります。

1. ウィジェットを **バリューマップ** に設定し、 **レイヤから値を読み込む** をクリックします。
2. Select `roads` in the *Label* dropdown and `highway` for both the *Value* and *Description* options:



3. *OK* を三回クリックします。

4. If you use the  Identify Features tool on a street now while edit mode is active, the dialog you get should look like this:

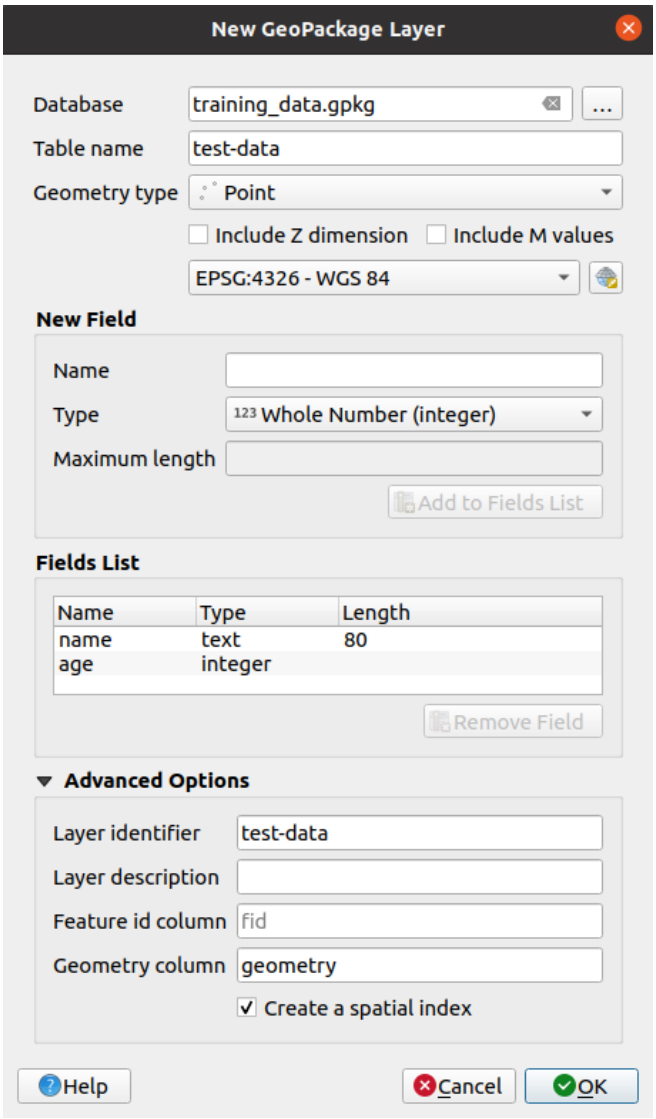


5.3.5 ★★★ （上級レベル） 自分でやってみよう: テストデータを作る

まったくのゼロから独自のカスタム フォームを設計することもできます。

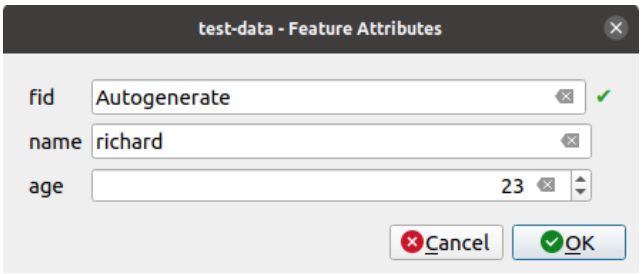
1. 次の 2 つの属性を持つ `test-data` という名前の簡単なポイントレイヤを作成します：

   - name (text)
   - age (integer)



2. デジタイズツールを使用して新しいレイヤ上にいくつかのポイントを追加してテスト用データを作成します。新しいポイントをキャプチャするたびに QGIS の既定の属性フォームが表示されます。

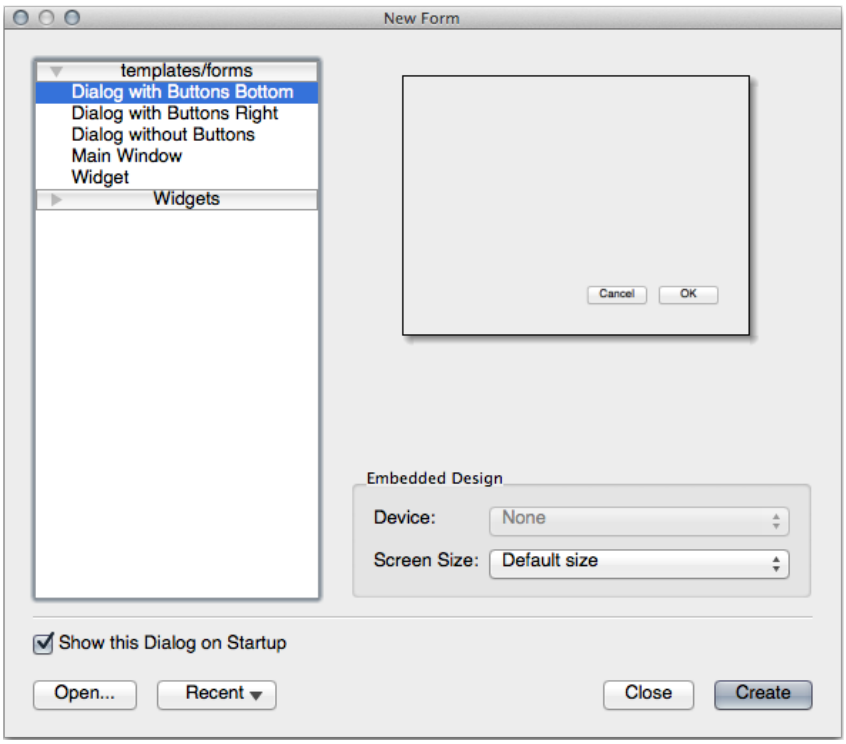
> **注釈**
>
> 以前の作業の時からスナップを有効にしたままの場合、スナップを無効にする必要があります。



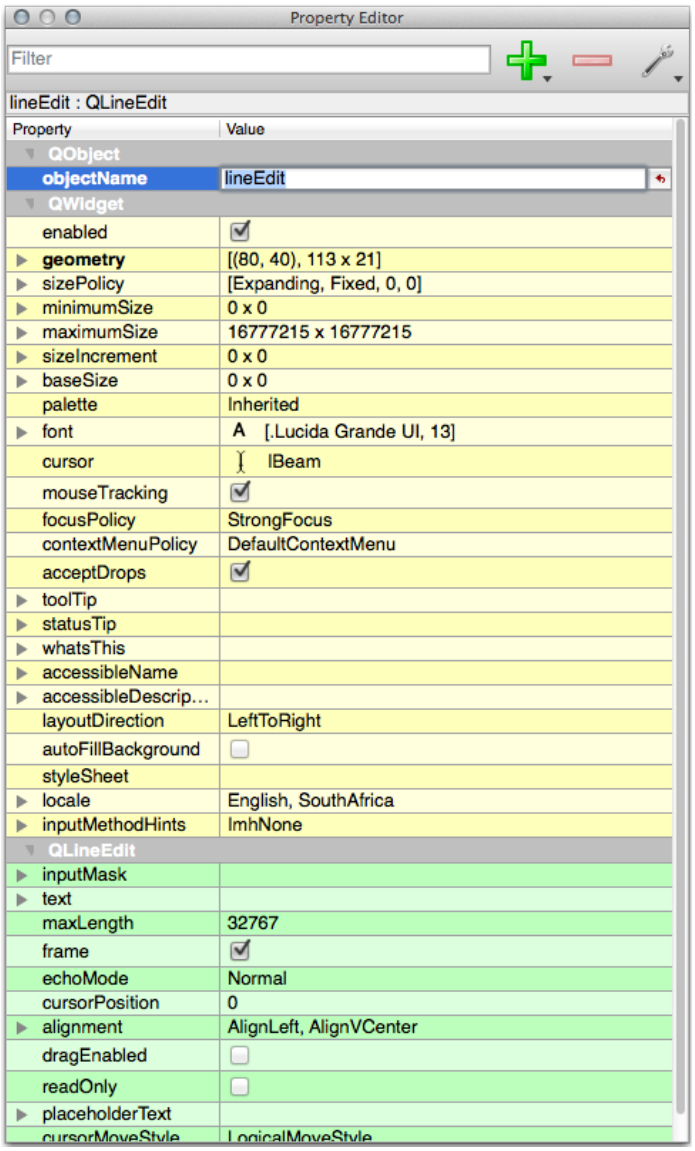
5.3.6 ★★★ （上級レベル） 理解しよう: 新しいフォームを作る

さて、属性データを取得するときに使う、独自のカスタムフォームを作成したいと思います。これを行うには、 *QT Designer* をインストールする必要があります（フォームを作成する人だけが必要です)。

1. *QT Designer* を起動します。
2. 表示されるダイアログで新しいダイアログボックスを作成します:



3. 画面の左側 (デフォルト) にある **ウィジェットボックス** で *Line Edit* アイテムを探します。
4. このアイテムをクリックしてフォームにドラッグします。フォーム上に新しい *Line Edit* が作成されます。
5. Line Edit 要素を選択すると、その **プロパティ** が画面の片側に沿って表示されます (デフォルトで右側):



6. その名前を `name` にします。

7. 同じ方法で、新しい *Spin Box* を作成し、その名前を `age` に設定します。

8. 新しい人を追加する というテキストを太いフォントで書いた **ラベル** を追加します（設定方法は、オブジェクトの **プロパティ** を見てください）。また、（ラベルを追加するのではなく）ダイアログ自体のタイトルを設定することもできます。

9. *Line Edit* と *Spin Box* に、 **ラベル** を追加します。

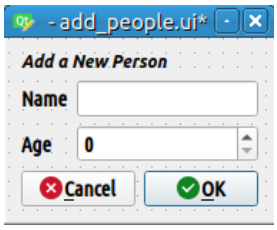
10. 自分の好きなように要素をアレンジしてください。

11. ダイアログの任意の場所をクリックします。

12. Find the *Lay Out in a Form Layout* button (in a toolbar along the top edge of the screen, by default). This lays out your dialog automatically.

13. ダイアログの最大サイズ (プロパティ) を `200` (幅) × `150` (高さ) に設定します。

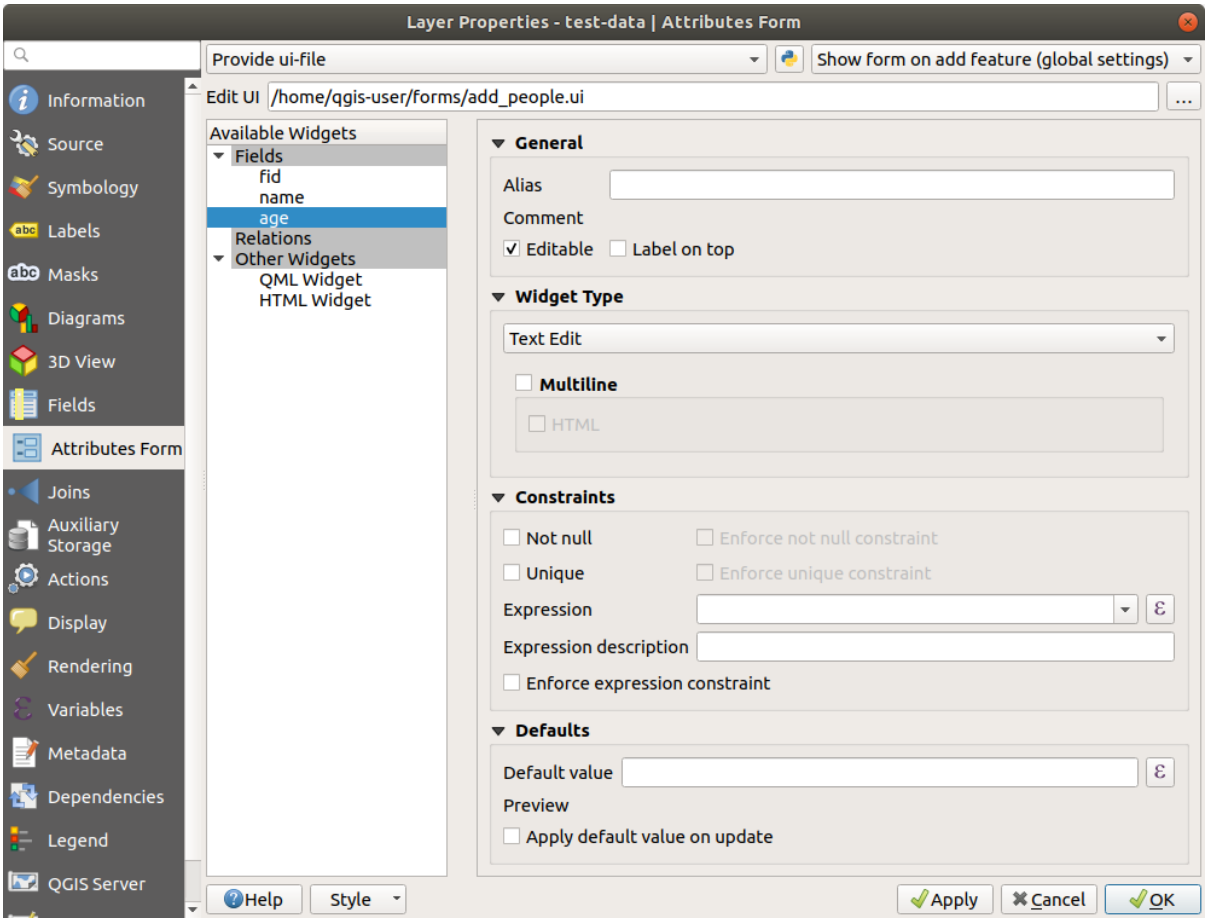
14. このようなフォームになるはずです:



15. 新しいフォームを `exercise_data/forms/add_people.ui` という名前で保存します
16. 保存が完了したら、 *Qt Designer* を終了します。

5.3.7 ★★★ （上級レベル） 理解しよう: レイヤをフォームに関連付ける

1. QGIS に戻ります
2. Double click the `test-data` layer in the legend to access its properties.
3. **レイヤプロパティ** ダイアログの **属性フォーム** タブをクリックします
4. **属性エディタレイアウト** ドロップダウンボックスで *ui*-**ファイルを提供する** を選択します。
5. 楕円形ボタンをクリックし、先ほど作成した `add_people.ui` ファイルを選択します:



6. **レイヤプロパティ** ダイアログで、 *OK* をクリックします

7. 編集モードに入り、新しいポイントを取り込みます
8. そうするとカスタムダイアログが表示されます (QGIS が通常作成するものの代わりに)。
9. 

ツールを使ってポイントの一つをクリックした場合、識別結果ウィンドウで右クリックして、コンテキストメニューから **地物フォームを表示** を選択すると、フォームを表示することができるようになりました。
10. このレイヤの編集モードになっている場合、コンテキストメニューには **地物フォームを編集** が表示され、最初のキャプチャ後でも新しいフォームで属性を調整することができるようになります。

5.3.8 結論

フォームを使用すればデータの編集や作成がもっと楽になります。ウィジェットの種類を編集するか全くのゼロから新しいフォームを作成することで、新しいデータをデジタイズをする人のエクスペリエンスをコントロールできます。それによって誤解や不必要なエラーを最小限に押さえることができます。

5.3.9 参考文献

もし上記の上級編を完了し、Python の知識があれば、データ検証、オートコンプリートなどを含む高度な機能を可能にする Python ロジックによるカスタム機能フォームの作成についての このブログエントリー をチェックするとよいでしょう。

5.3.10 次は?

地物フォームを開くことは QGIS ができる標準的な操作の 1 つです。一方で、自ら定義したカスタムアクションを実行させることもできます。これは次のレッスンのテーマです。

5.4 レッスン: アクション

前のレッスンで既定のアクションを見たので、今度は自分のアクションを定義してみましょう。

アクションとは、地物をクリックしたときに発生するものです。アクションは、例えば、オブジェクトに関する追加情報を取得するなど、地図に多くの機能を追加することができます。アクションを割り当てることで、地図にまったく新しい次元を追加することができます！

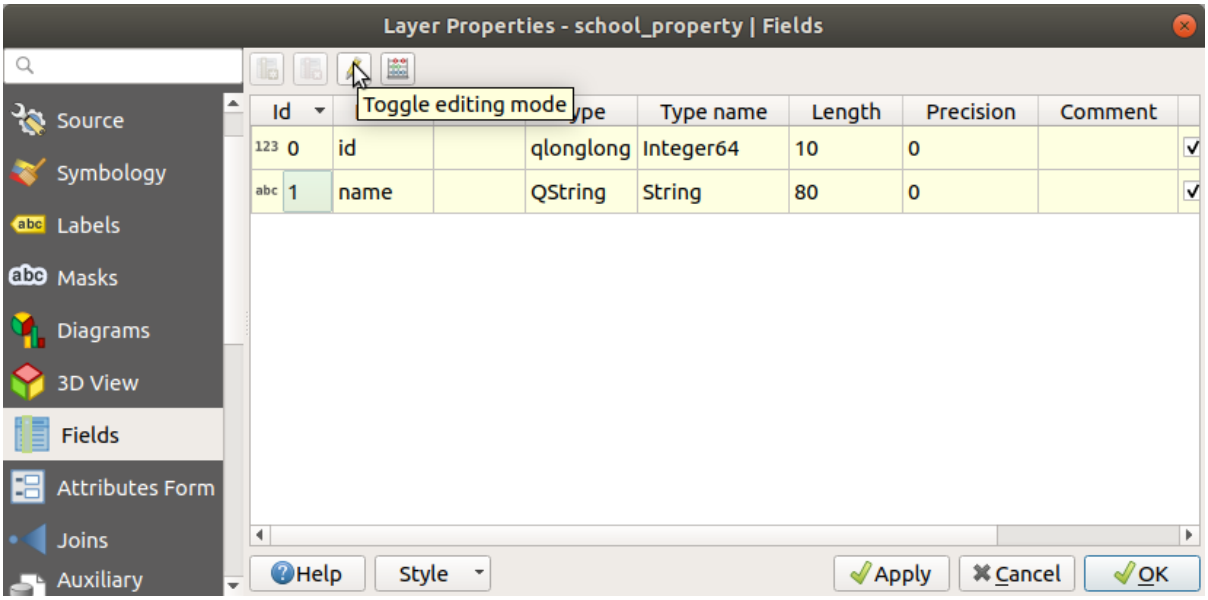
このレッスンの目標: カスタムアクションを追加する方法を学びます。

In this lesson you will use the `school_property` layer you created previously. The sample data include photos of each of the three properties you digitized. What we are going to do is to associate each property with its image. Then we will create an action that will open the image for a property when clicking on the property.

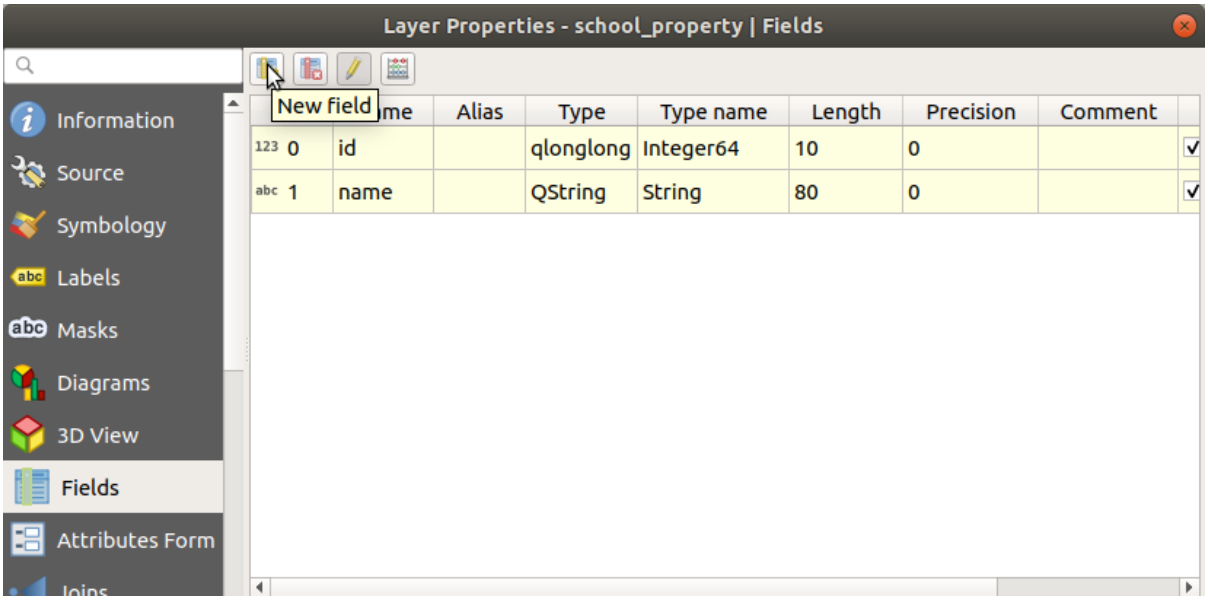
5.4.1 ★☆☆ （初級レベル） 理解しよう: 画像のためのフィールドを追加

school_property レイヤには、まだ画像と物件を関連付ける方法がありません。まず、この目的のためにフィールドを作成します。

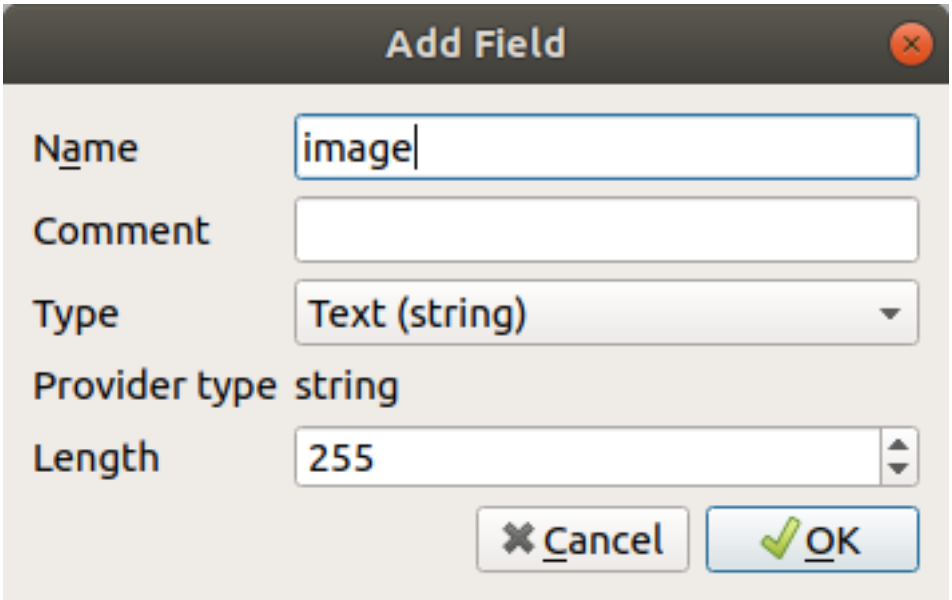
1. **レイヤプロパティ** ダイアログを開きます。
2. **属性** タブをクリックします。
3. 編集モードに切り替えます:



4. 新しい列を追加します:

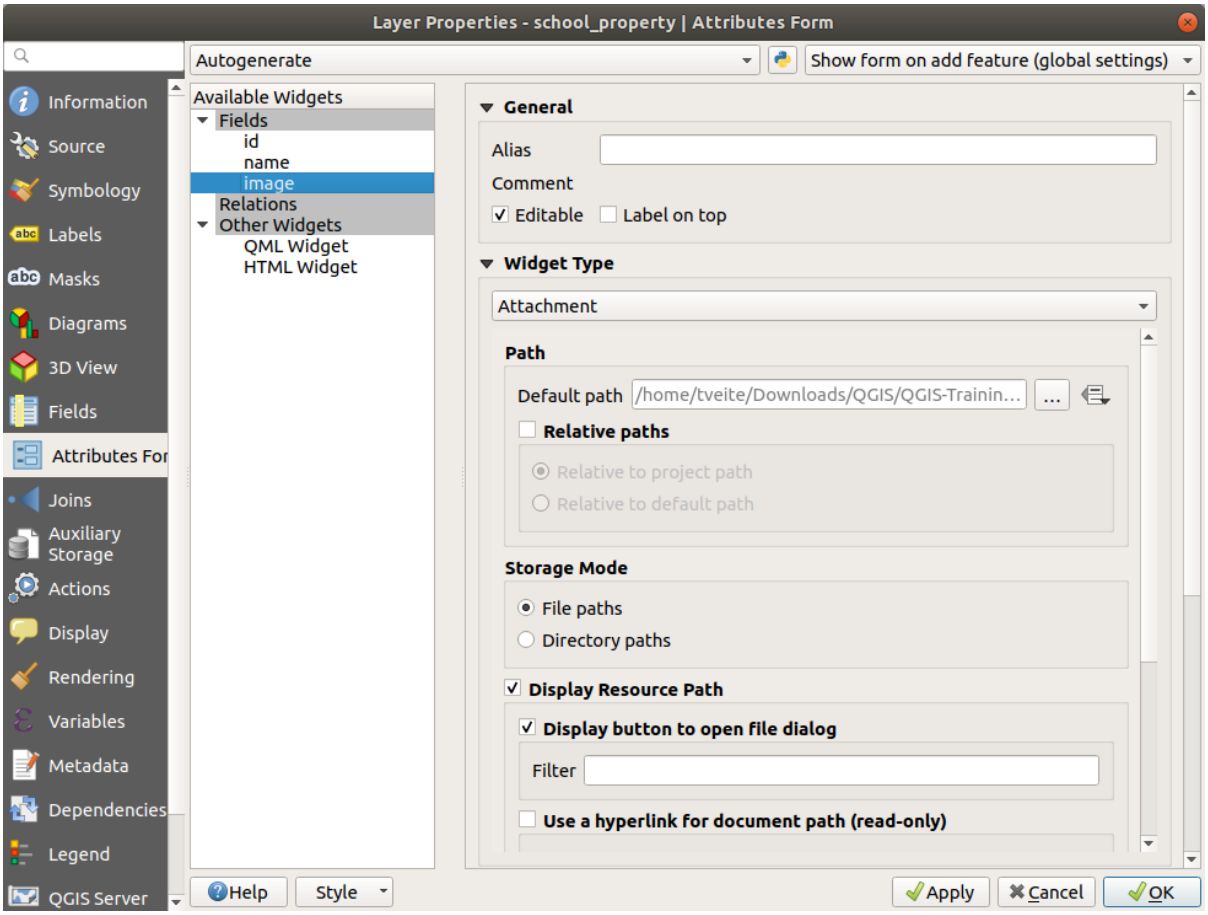


5. 下記の値を入力します:



6. フィールドが作成されたら、 **属性フォーム** タブに移動して、 `image` フィールドを選択します。

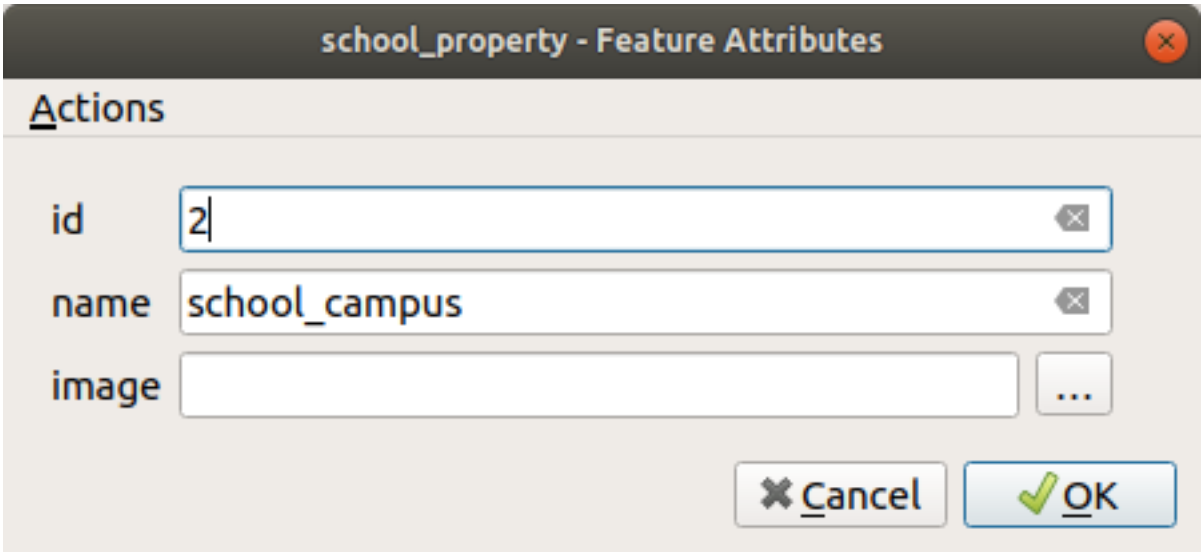
7. **ウィジェットタイプ** を **アタッチメント** に設定します：



8. **レイヤプロパティ** ダイアログで *OK* をクリックします。

9. Use the Identify Features tool to click on one of the three features in the `school_property` layer.

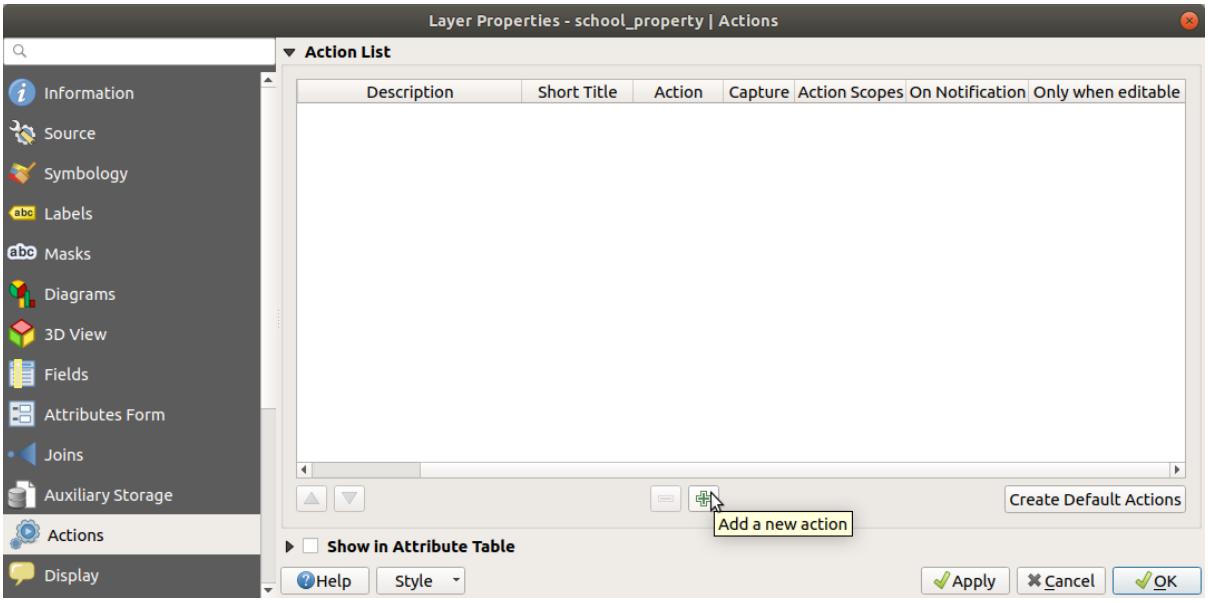
編集モードのままなので、ダイアログがアクティブになり、次のように表示されるはずです：



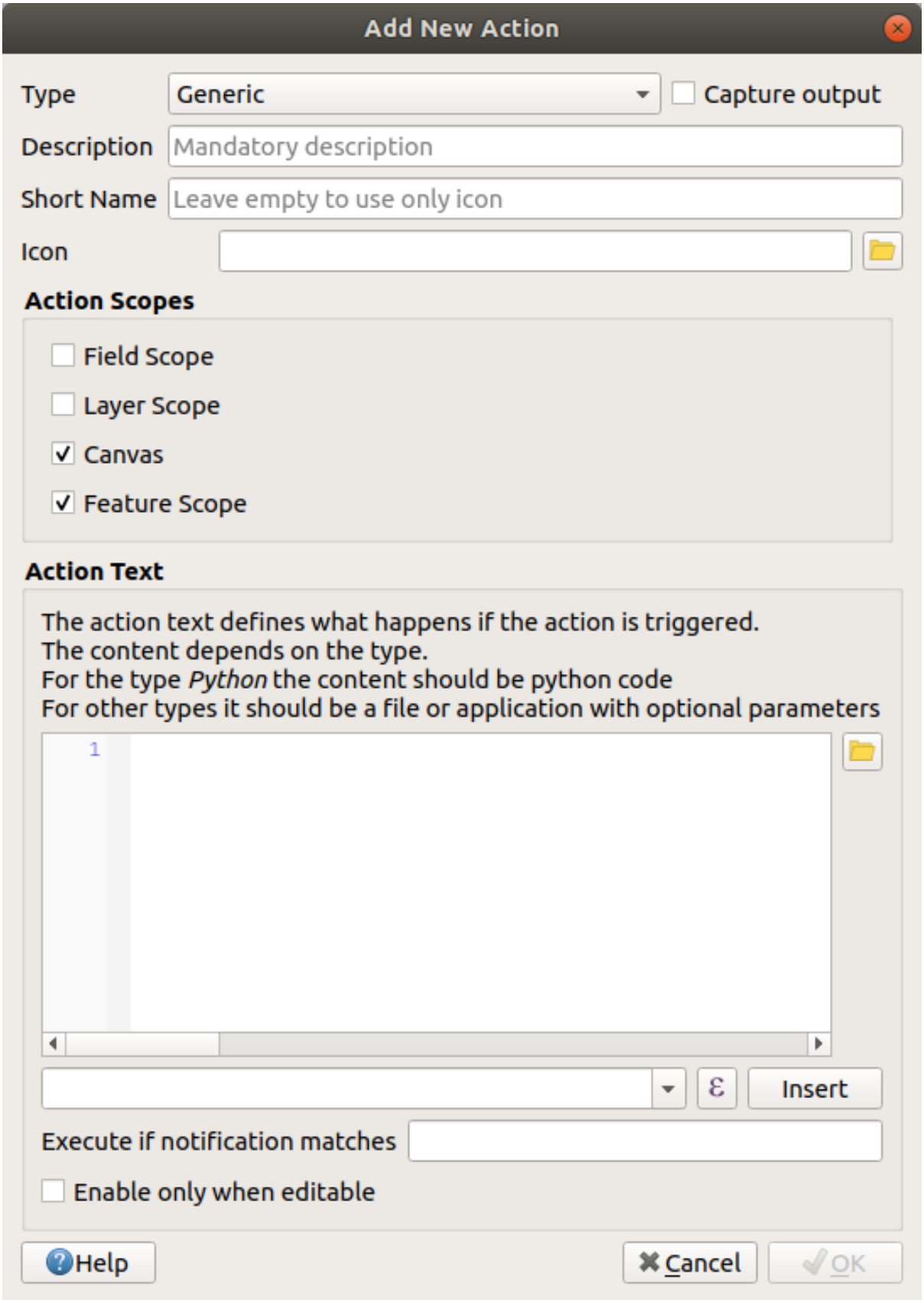
10. 参照ボタンをクリックします (*image* フィールドの横の ...)。
11. 画像のパスを選択します。画像は `exercise_data/school_property_photos/` にあり、関連付けるべき地物と同じ名前になっています。
12. *OK* をクリックします。
13. この方法ですべての画像と地物を正しく関連付けます。
14. 編集内容を保存し、編集モードを終了します。

5.4.2 ★☆☆ （初級レベル） 理解しよう: アクションを作る

1. Open the *Actions* tab for the `school_property` layer, and click on the  Add a new action button.



2. **新規アクションを追加** ダイアログで、 **説明** フィールドに `Show Image` という文字を入力します:



次に何をすべきかはオペレーティングシステムによって異なりますので、次で適切なコースを選択し

て下さい:

- Windows

  データ型 ドロップダウンリストをクリックし、 *URL* **を開く** を選択します。

- Ubuntu Linux

  アクション の下で、 *Gnome Image Viewer* 用に `eog` を記入するか、*ImageMagick* を使うために `display` を記入します。コマンドの後ろに空白をひとつ入れることを忘れないでください！

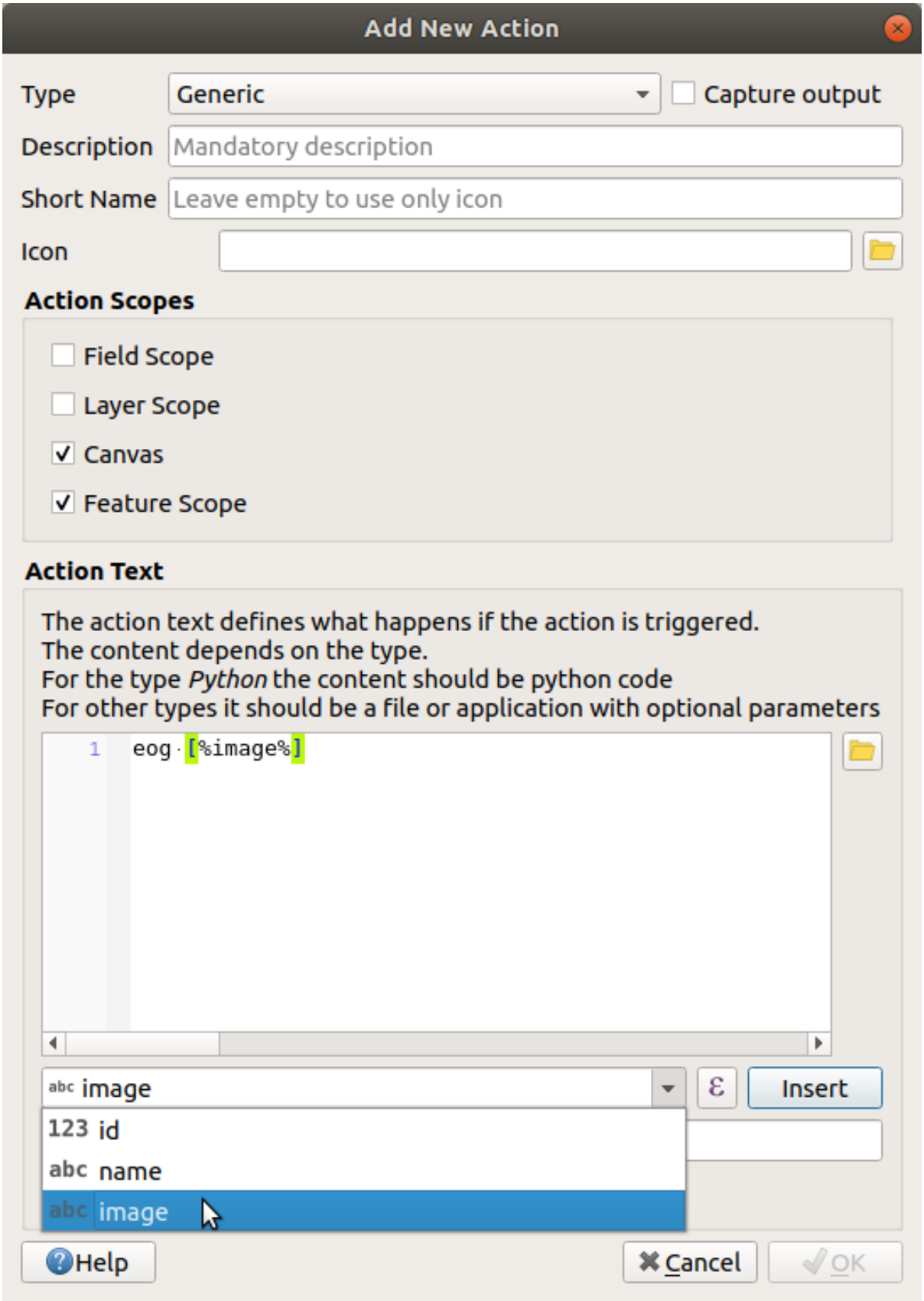
- macOS

  1. **データ型** ドロップダウンリストをクリックし、 *Mac* を選択します。
  2. **アクション** の下に `open` を記入します。コマンドの後ろに空白をひとつ入れることを忘れないでください！

これで、コマンドを書き続けることができます。

あなたは画像を開きたい。そして QGIS は画像の場所を知っています。あとは **アクション** に画像がどこにあるかを知らせるだけです。

3. リストから *image* を選択します:



4. **挿入** ボタンをクリックします。QGIS は **アクションテキスト** フィールドに `[% "image" %]` という語句を追加します。

5. *OK* ボタンをクリックして **新規アクションを追加** ダイアログを閉じます
6. *OK* をクリックして、 **レイヤプロパティ** ダイアログを閉じます

では新しいアクションを試してみましょう：

1. Click on the `school_property` layer in the *Layers* panel so that it is highlighted.
2. （**属性ツールバー** にある） 地物アクションの実行 ボタンを見つけます。
3. このボタンの右側にある下向き矢印をクリックします。このレイヤには、今のところ、作ったばかりのアクション 1 つだけが定義されています。



4. ボタン自体をクリックしてツールをアクティブにします。
5. このツールを使用して、3 つの地所のいずれかをクリックします。

   その物件の画像が開くはずです。

5.4.3 ★★☆ （中級レベル） 理解しよう: インターネットを検索する

地図を見ていて、ある農場のある地域についてもっと知りたいとします。その地域について何も知らないあなたが、その地域についての一般的な情報を見つけたいとします。今、パソコンを使っていることを考えると、まず最初にその地域の名前を Google で検索するのではないでしょうか。そこで、QGIS に自動で検索させることにしましょう！

1. Open the attribute table for the `landuse` layer.

   Google の検索には、土地利用分野ごとに `name` フィールドを使用する予定です。
2. 属性テーブルを閉じます。
3. **レイヤープロパティ** の **アクション** に戻ります。
4. **デフォルトアクションを作成** ボタンをクリックして、あらかじめ定義されたいくつかのアクションを追加します。
5. 下の 選択中のアクションを削除 ボタンを使って、短いタイトルが **検索ウェブ** の *URL* **を開く** アクションを除く、すべてのアクションを削除します。
6. 残っているアクションをダブルクリックして編集します
7. **説明** を `Google Search` に変更し、 **短い名前** フィールドの内容を削除してください。
8. **スコープ** で **キャンバス** がチェックされていることを確かめてください。

   次に何をすべきかはオペレーティングシステムによって異なりますので、次で適切なコースを選択して下さい:

- Windows

  タイプ で **開く** を選択します。これは Windows に Internet Explorer 等の既定のブラウザでインターネットアドレスを開かせます。

- Ubuntu Linux

  アクション の下に、`xdg-open` と記述します。これは、Ubuntu に、Chrome や Firefox などの既定のブラウザでインターネットアドレスを開くように指示します。

- macOS

  アクション の下に `open` と記述します。これは、macOS に、Safari などの既定のブラウザでインターネットアドレスを開くように指示しします。

これで、コマンドを書き続けることができます

上でどのコマンドを使った場合でも次に、開くべきインターネットアドレスを知らせなければいけません。Google を訪問させて語句を自動的に検索させます。

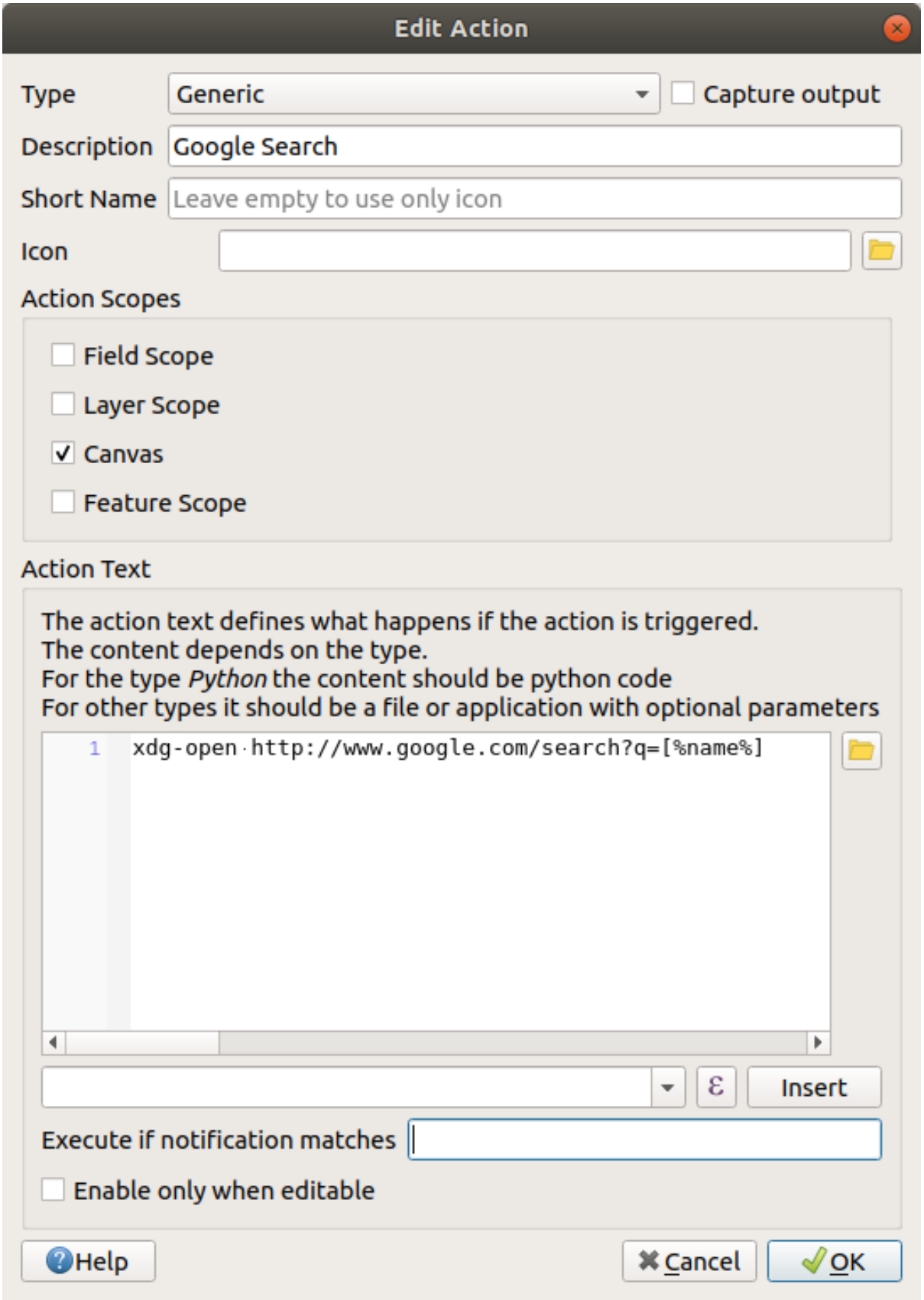
通常、Google を使うときは、Google の検索バーに検索フレーズを入力します。しかし今回の場合、コンピュータにこれをやってもらいたいのです。Google に検索を指示する方法（検索バーを直接使いたくない場合）は、インターネットブラウザに `https://www.google.com/search?q=SEARCH_PHRASE` というアドレスを与えます。 `SEARCH_PHRASE` は検索したいものです。ここでは、まだ検索するフレーズがわからないので、最初の部分（search フレーズなし）だけを入力します。

9. **アクション** フィールドに、`https://www.google.com/search?q=` と記述してください。これを書き込む前に、最初のコマンドの後にスペースを追加することを忘れないでください!

   次に QGIS がブラウザに、クリックした地物の `name` の値を検索するように Google に指示するようにします。

10. *name* フィールドを選択します。

11. *Insert* ボタンをクリックします:



これが意味するところは、QGIS がブラウザを開いて、アドレス `https://www.google.com/search?q=[% "name" %]` に送信するということです。`[% "name" %]` は、検索するフレーズとして `name`

フィールドの内容を使用するように QGIS に指示します。

例えば、クリックした土地利用区域の名前が `Marloth Nature Reserve` であった場合、QGIS は `https://www.google.com/search?q=Marloth%20Nature%20Reserve` をブラウザに送信し、ブラウザは Google にアクセスし、Google は "Marloth Nature Reserve" を検索することになります。

12. まだの方は、上記で説明したようにすべて設定してください。
13. *OK* ボタンをクリックして **新規アクションを追加** ダイアログを閉じます
14. *OK* をクリックして、 **レイヤプロパティ** ダイアログを閉じます

では新しいアクションをためします。

1. With the `landuse` layer active in the *Layers* panel, click on the down arrow to the right of the  Run feature action button, and select the only action (`Google Search`) defined for this layer.
2. 地図上に表示されている土地利用区域のどれかをクリックしてください。ブラウザが起動し、そのエリアの `name` 値として記録されている場所を Google で検索し始めます。

> **注釈**
>
> アクションがうまく動作しない場合は、すべてが正しく入力されたことをチェックしてください。タイプミスはこの種の作業でよくあることです!

5.4.4 ★★★ （上級レベル） 理解しよう: QGIS で直接 Web ページを開く

上記では、外部ブラウザでウェブページを開く方法について見てきました。この方法には、エンドユーザーが自分のシステムでそのアクションを実行するために必要なソフトウェアを持っているかどうかという、未知の依存関係を追加してしまうという欠点があります。エンドユーザーがどの OS を使っているかわからない場合、同じ種類のアクションのための同じ種類の基本コマンドを持っているとは限らないことは、ご覧のとおりです。OS のバージョンによっては、上記のブラウザを開くためのコマンドが全く動作しないこともあります。これは、どうしようもない問題かもしれません。

しかし、QGIS は、信じられないほど強力で多用途な Qt ライブラリの上に乗っています。また、QGIS のアクションは、任意のトークン化された（つまり、フィールド属性の内容に基づいた変数情報を使用した）Python コマンドを使用することができます！

Python アクションを使用して Web ページを表示する方法を説明します。これは外部ブラウザでサイトを開くのと同じ一般的なアイデアですが、Qt QWebView クラス（webkit ベースの html ウィジェットです）を使ってポップアップウィンドウにコンテンツを表示するので、ユーザーのシステム上にブラウザは必要ありません。

今回は Wikipedia を使ってみましょう。つまり、要求された URL は次のようになります:

```
https://wikipedia.org/wiki/SEARCH_PHRASE
```

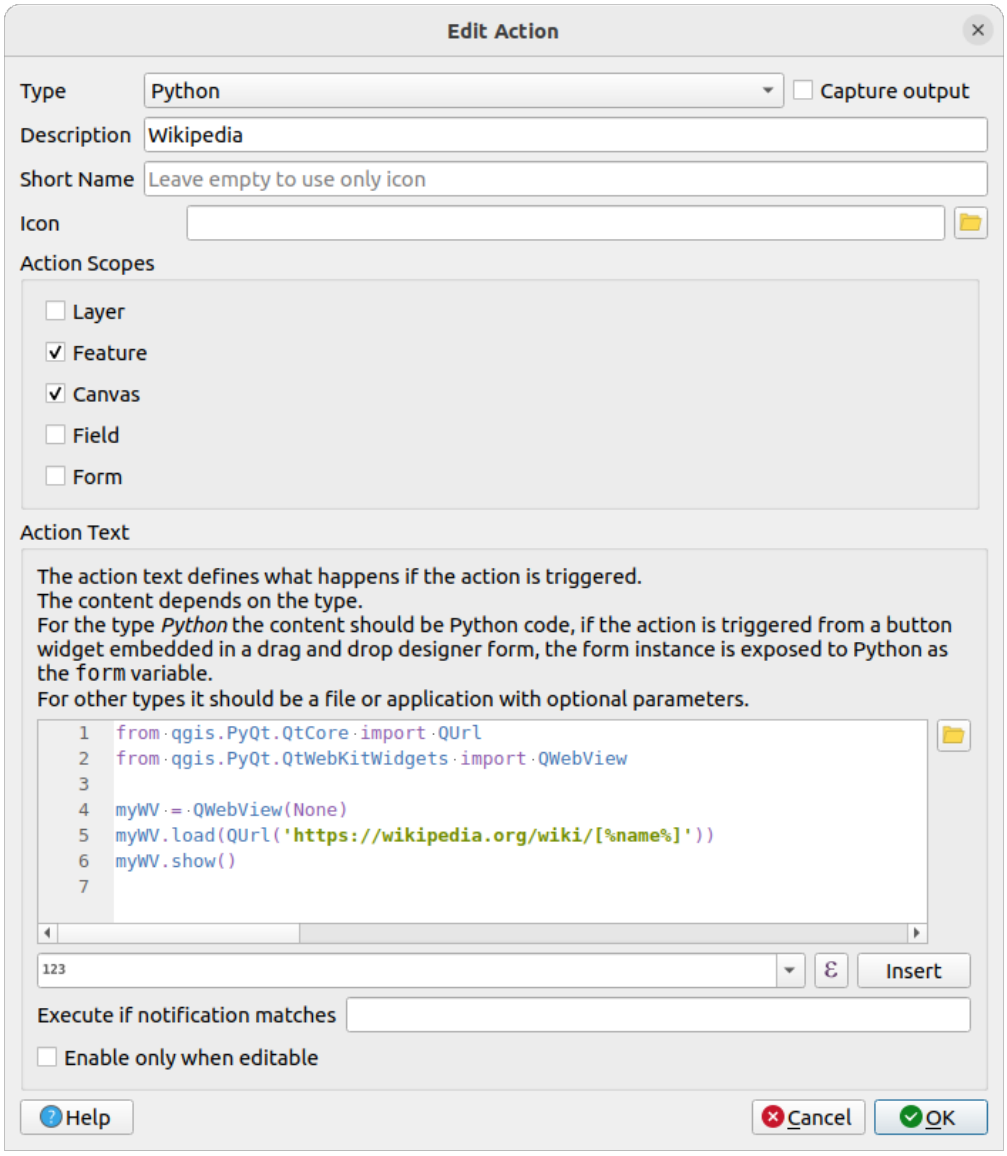
レイヤアクションを作成するには:

1. **レイヤプロパティ** ダイアログを開いて **アクション** タブに移動します。
2. 次のアクションのプロパティを使って新しいアクションを設定します。

- **型**: Python
- **説明**: Wikipedia
- **スコープ**: 地物, キャンバス
- **アクションテキスト**:

```
from qgis.PyQt.QtCore import QUrl
from qgis.PyQt.QtWebKitWidgets import QWebView

myWV = QWebView(None)
myWV.load(QUrl('https://wikipedia.org/wiki/[%name%]'))
myWV.show()
```



ここでは:

- アクションが呼び出されると、[%name%] は実際の属性値に置き換えられます (以前と同様です)。

- このコードでは、単に新しい `QWebView` インスタンスを作成してその URL を設定し、 `show()` を呼び出してユーザーのデスクトップにウィンドウとして表示させるだけです。

また、この方法を使えば、ユーザーが特定の画像ビューアーをシステム上に持っていることを要求せずに、画像を表示することができます。

3. 先程作成した Wikipedia アクションを使って、上記の方法で Wikipedia のページを読み込んでみてください。

5.4.5 結論

アクションを使うと、QGIS で同じマップを表示するエンドユーザーにとって便利な追加機能をマップに付与することができます。呼び出されるプロセスは、Python だけでなく、あらゆるオペレーティングシステムのシェルコマンドにすることができるため、組み込むことができる機能は無限大です！

5.4.6 次は?

さて、あらゆる種類のベクトルデータの作成を行ったので、問題を解決するためにデータを分析する方法を学びます。それが次のモジュールのテーマです。

第6章　モジュール: ベクタ解析

これまでにいくつかの地物を編集したので、次はそれらを使って他に何ができるかを知る必要があります。属性を持つ地物を持つことはいいですが、すべてが実行されたとき、通常の GIS でない地図ではできないことが本当にはわかりません。

GIS の主な利点は以下です：*GIS* **は質問に答えることができます** 。

次の３つのモジュールでは、GIS の機能を使って **研究課題** に答えるよう努めます。例えばあなたが不動産業者であり、 Swellendam において次の基準を持っているお客様のために住宅を探しています:

1. Swellendam にある必要がある。
2. 学校前の距離が、合理的にアクセスできる距離（例えば 1km）である必要がある。
3. サイズが 100m 四方以上である必要がある。
4. 主要道路から 50m より近い。
5. レストランから 500m 以内にある。

次のいくつかのモジュールの中では、この新しい住宅開発に適した農地の物件を見つけるために、GIS 解析ツールの力を利用します。

6.1 レッスン: データを再投影および変換する

ここでもう一度、座標参照系 (CRS) の話をしましょう。以前にも少し触れたことがありますが、実用上どのような意味があるのかについては触れていませんでした。

このレッスンの目標: ベクターデータセットの再投影および変換をします。

6.1.1 ★☆☆ （初級レベル） 理解しよう: 投影法

この時点で全てのデータと地図自体のある CRS は、 *WGS84* と呼ばれています。これは、データを表現するためのとても一般的な地理座標系 (GCS) です。しかし、これから見ていくように問題があります。

1. 現在の地図を保存してください
2. 次に、 `exercise_data/world/world.qgs` の下にある世界地図を開いてください
3. Zoom in to South Africa by using the  Zoom In tool

4. 画面下の **ステータスバー** にある **縮尺** フィールドにスケールを設定してみてください。南アフリカの上で、この値を `1:5 000 000`（1 対 500 万）に設定してください。

5. **縮尺** フィールドを見ながら、マップをパンします

スケールが変化していることに気づいたでしょうか？それは、画面の中心にある `1:5 000 000` でズームした 1 点から遠ざかっているからです。その点から離れたところでは、縮尺が異なります。

理由を理解するために、地球の球体を考えます。そこには北から南に複数の線が引かれています。これらの経線は、赤道では遠く離れていますが、極で出会っています。

GCS では、この球体で作業をしているのですが、画面は平らです。球体を平面で表現しようとすると、テニスボールを切り開いて平らにしようとするのと同じような歪みが発生します。これは地図上では、経線は極点（合流するはずの場所）でも同じように離れていることを意味します。つまり、地図上で赤道から遠ざかるにつれて、見えるものの縮尺はどんどん大きくなっていくのです。このことは、現実的には、地図上に一定の縮尺が存在しないことを意味します！

この問題を解決するため、かわりに投影座標系（PCS）を使用してみましょう。PCS では縮尺変更のための余裕を作り、それを修正する方法でデータを「投影」または変換します。そのため、一定の縮尺を維持するために、PCS を使用するために私たちのデータを投影変換する必要があります。

6.1.2 ★☆☆ （初級レベル） 理解しよう: 「オンザフライ」再投影

デフォルトでは、QGIS はデータを「オンザフライ」で再投影します。この意味は、データそのものが別の CRS にある場合でも、QGIS はそれを任意の CRS にあるかのように投影することができるということです。

QGIS の右下にある  現在の CRS ボタンをクリックすることで、プロジェクトの CRS を変更することができます。

1. 表示されたダイアログで、`global` という単語を **フィルタ** フィールドに入力します。その下の **あらかじめ定義された** *CRS* フィールドに、いくつかの CRS が表示されるはずです。

2. *WGS 84 / NSIDC EASE-Grid 2.0 Global | EPSG:6933* のエントリをクリックして選択し、*OK* をクリックします。

   南アフリカの形状が変化することに注意してください。すべて投影法の変更によって地球の見た目としての形状が変わります。

3. 前と同様に、`1:5 000 000` の縮尺に拡大します。

4. 地図をパンニングします。

   縮尺は同じままであることに注意します！

「オンザフライ」再投影は異なる CRS のデータセットを組み合わせて使う際にも用いられます。

1. 南アフリカのデータのみを含む別のベクタレイヤを地図に追加します。これは `exercise_data/world/RSA.shp` として見つかります。

2. それを読み込みます。その CRS を見る簡単な方法は、凡例のレイヤにマウスカーソルを置くことです。それは `EPSG:3410` です。

   何に気づきますか？

The layer is visible even if it has a different CRS from the `continents` one.

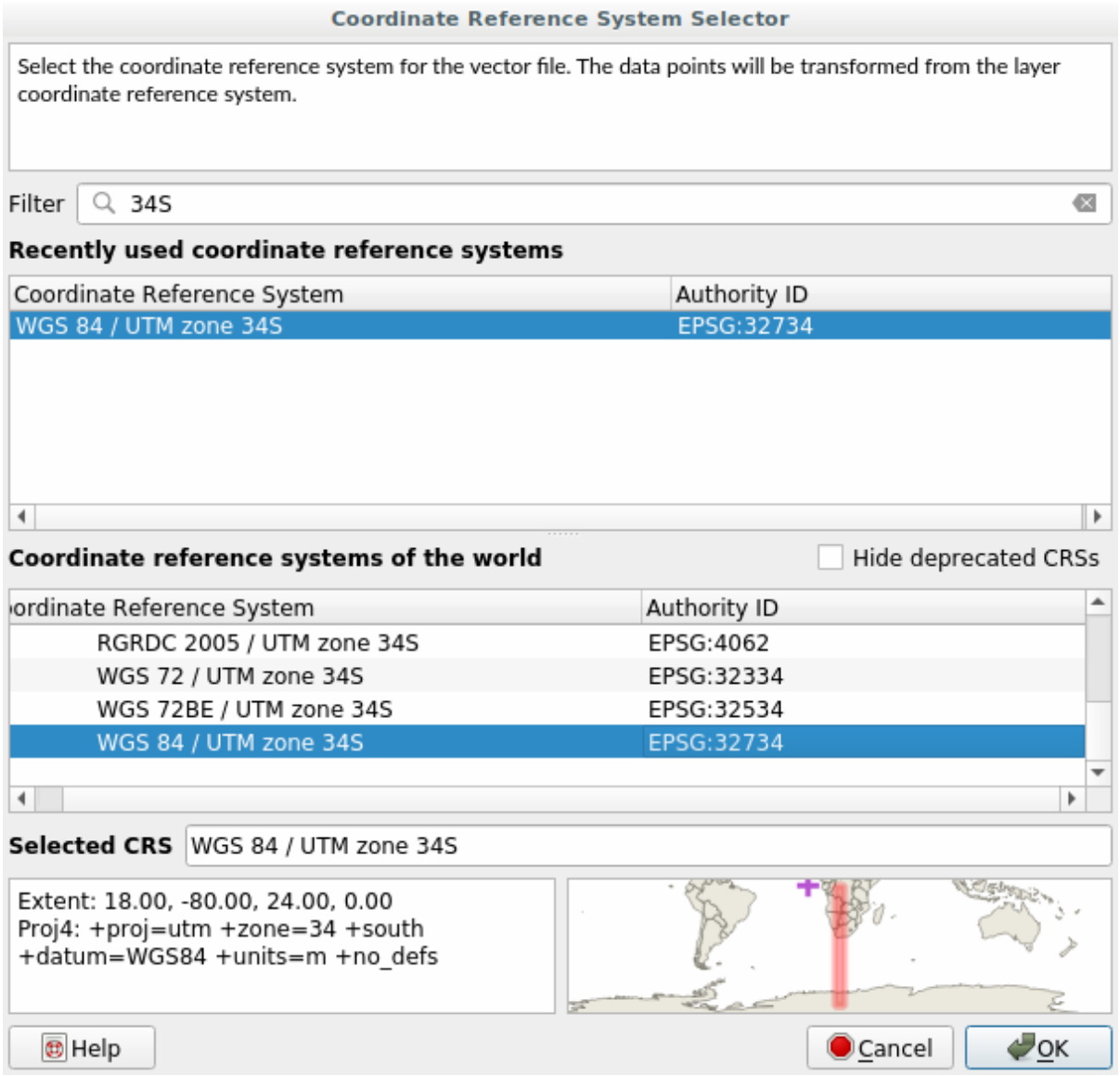
6.1.3 ★★☆ （中級レベル） 理解しよう: 他の CRS へデータセットを保存する

時には、既存のデータセットを別の CRS でエクスポートする必要があります。次のレッスンで説明するように、レイヤ上で距離計算をする必要がある場合、投影座標系でレイヤを持つことが常に良いことです。

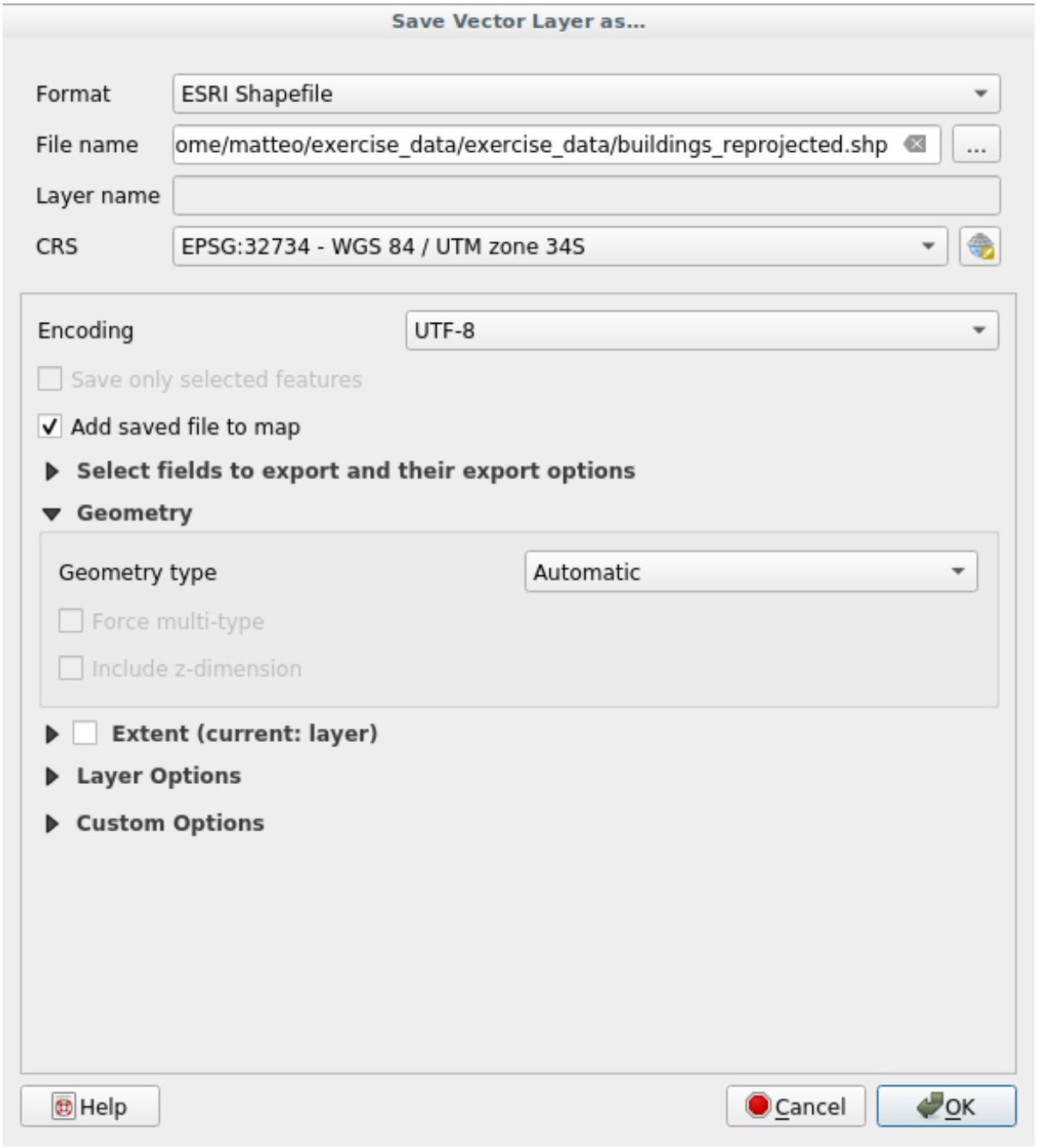
「オンザフライ」再投影は、**プロジェクト** に関連するものであり、単一のレイヤに関連するものではないことに注意し てください。つまり、レイヤを **正しい** 位置で見たとしても、プロジェクトとは異なる CRS を持つことがあります。

レイヤを別の CRS でエクスポートすることが簡単にできます。

1. `training_data.gpkg` から `buildings` データセットを追加します
2. **レイヤ** パネルで `buildings` レイヤを右クリックします
3. 表示されたメニューから **エクスポート** **新規ファイルに地物を保存**... を選択します。すると、 **名前をつけてベクタレイヤを保存**... ダイアログが表示されます。
4. **ファイル名** フィールドの隣にある **ブラウズ** ボタンをクリックします
5. `exercise_data/` に移動して、新しいレイヤの名前を `buildings_reprojected.shp` に指定します。
6. *CRS* の値を変更します。最近使用された CRS のみがドロップダウンメニューに表示されます。ドロップダウンメニューの横にある、 CRS の選択 ボタンをクリックします。
7. *CRS* **の選択** ダイアログが表示されます。その *Filter* **フィルタ** フィールドで `34S` を検索します。
8. リストから *WGS 84 / UTM zone 34S | EPSG:32734* を選択します。



9. 他のオプションは変えずにおきます。**名前を付けてベクタレイヤを保存**... ダイアログは次のようになります:



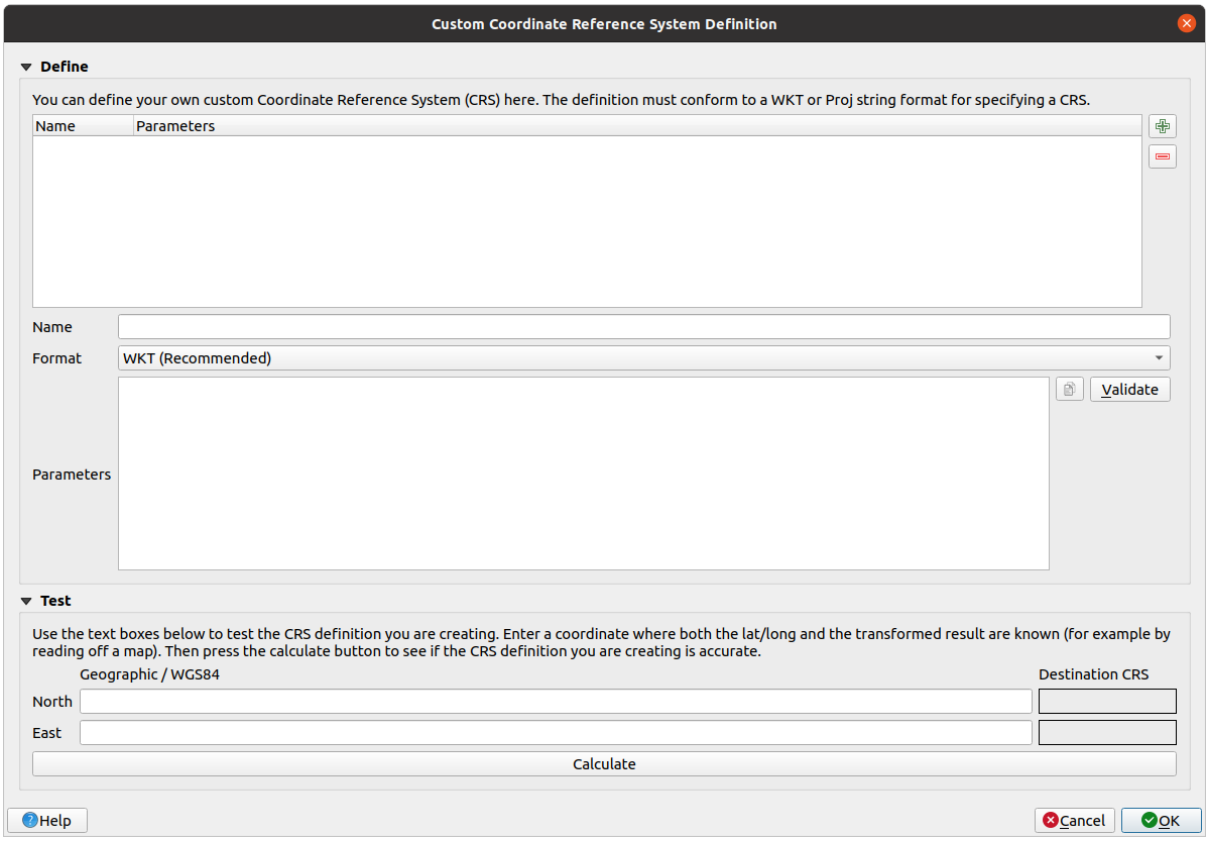
10. *OK* をクリックします

これでレイヤの新旧の投影法を比較すると、2 つの異なる CRS でありながら、重なっていることがわかります。

6.1.4 ★★★ （上級レベル） 理解しよう: 独自の投影法を作る

投影法は QGIS にデフォルトで含まれるものよりも多くあります。独自の投影法も作成できます。

1. 新しい地図を始めます
2. `world/oceans.shp` データセットを読み込みます
3. **設定** **カスタム投影法**... を実行すると、このようなダイアログが表示されます。



4. CRS を追加 ボタンをクリックし、新しい投影法を作成します
5. 面白い投影法として `Van der Grinten I` というのがあります。その名前を **名前** フィールドに入力してください。

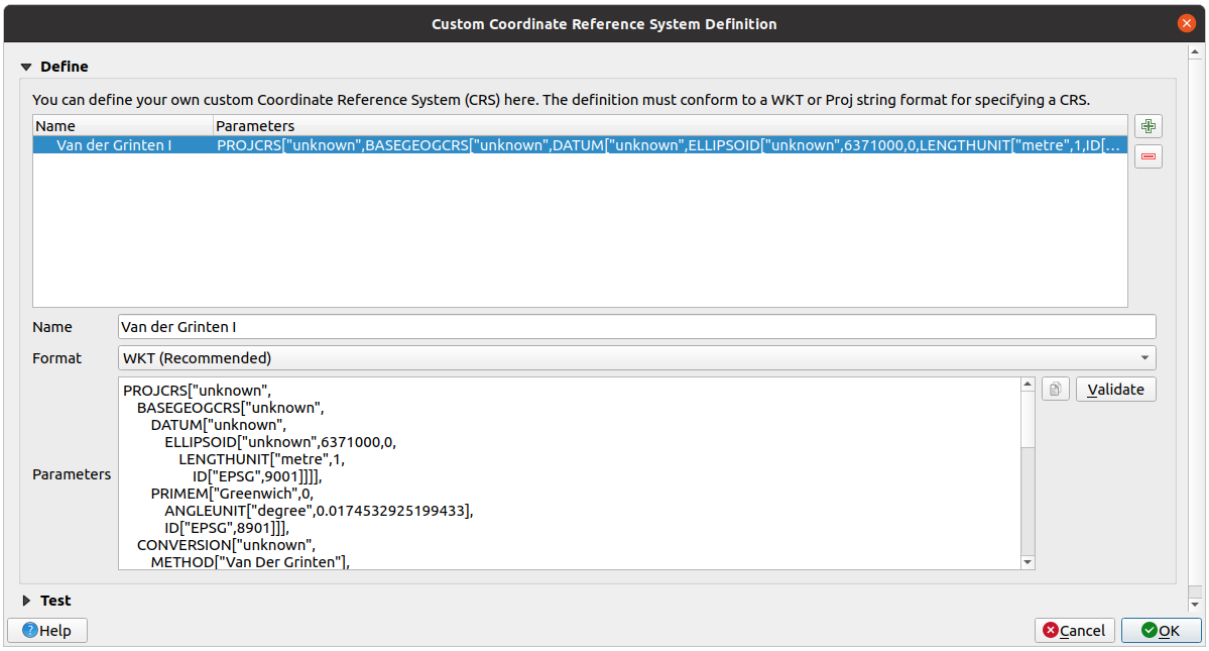
   他のほとんどの投影法がそうであるように、この投影法は、長方形のものの代わりに円形フィールドに地球を表します。
6. **形式** で、 *WKT*（**推奨**） を選択します
7. **パラメータ** フィールドに次の文字列を追加します：

```
PROJCRS["unknown",
    BASEGEOGCRS["unknown",
        DATUM["unknown",
            ELLIPSOID["unknown",6371000,0,
                LENGTHUNIT["metre",1,
```

(次のページに続く)

(前のページからの続き)

```
                    ID["EPSG",9001]]]],
        PRIMEM["Greenwich",0,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8901]]],
    CONVERSION["unknown",
        METHOD["Van Der Grinten"],
        PARAMETER["Longitude of natural origin",0,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8802]],
        PARAMETER["False easting",0,
            LENGTHUNIT["metre",1],
            ID["EPSG",8806]],
        PARAMETER["False northing",0,
            LENGTHUNIT["metre",1],
            ID["EPSG",8807]]],
    CS[Cartesian,2],
        AXIS["(E)",east,
            ORDER[1],
            LENGTHUNIT["metre",1,
                ID["EPSG",9001]]],
        AXIS["(N)",north,
            ORDER[2],
            LENGTHUNIT["metre",1,
                ID["EPSG",9001]]]]
```

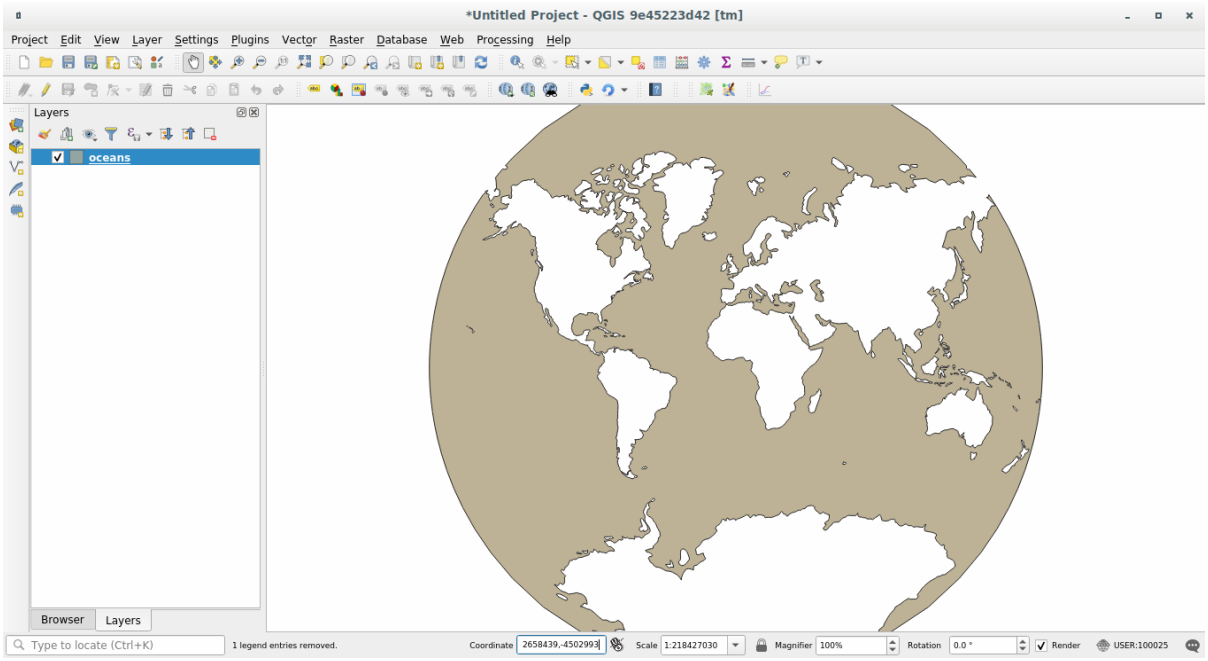


8. *OK* をクリックします

9. ステータスバーの右側にある 現在の CRS ボタンをクリックし、プロジェクトの CRS を変更します

10. 新しく定義した投影法を選択します（ **フィルタ** フィールドでその名前を検索します）

11. この投影法を適用するため地図は再投影され、こうなります：



6.1.5 結論

異なる投影法は、異なる目的のために有用です。正しい投影法を選択することにより、地図上の地物が正確に表現されていることを確認できます。

6.1.6 参考文献

このレッスンの *Advanced* セクションの教材は、 この記事 から引用したものです。

Coordinate Reference Systems に関する詳細な情報はこちらをご覧ください。

6.1.7 次は?

次のレッスンでは、QGIS の様々なベクトル解析ツールを使って、ベクトルデータを解析する方法を学びます。

6.2 レッスン: ベクタ解析

また、ベクタデータを解析することで、異なる地物が空間的にどのように相互作用しているかを明らかにすることができます。解析に関連する機能はたくさんあるので、ここではそのすべてを紹介することはしません。むしろ、問題を提起し、QGIS が提供するツールを使ってそれを解決してみることにします。

このレッスンの目標: 質問を尋ね、分析ツールを使ってそれを解決すること。

6.2.1 ★☆☆ （初級レベル） GIS プロセス

その前に、問題を解決するためのプロセスの概要を簡単に説明しておくと便利でしょう。それは次のものです:

1. 問題の状態
2. データの入手
3. 問題の分析
4. 結果のプレゼン

6.2.2 ★☆☆ （初級レベル） 問題

解決する問題を決定することから手順を開始しましょう。たとえば、不動産業者が以下の基準を持っている顧客のために Swellendam にある居住用の不動産を探しています：

1. それは Swellendam にあること
2. それは学校から車で行ける距離であること（たとえば 1km）
3. 100 平方 m 以上の大きさであること
4. 主要な道路から 50m 以内であること
5. レストランの 500m 以内であること

6.2.3 ★☆☆ （初級レベル） データ

これらの質問に答えるために、以下のデータが必要になります:

1. その地域の住宅用の物件（建物）
2. 街の中と周辺の道路
3. 学校とレストランの位置
4. 建物の大きさ

これらのデータは OSM を通じて入手可能であり、このマニュアルを通じて使用したデータセットもこのレッスンに使用できることが分かるはずです。

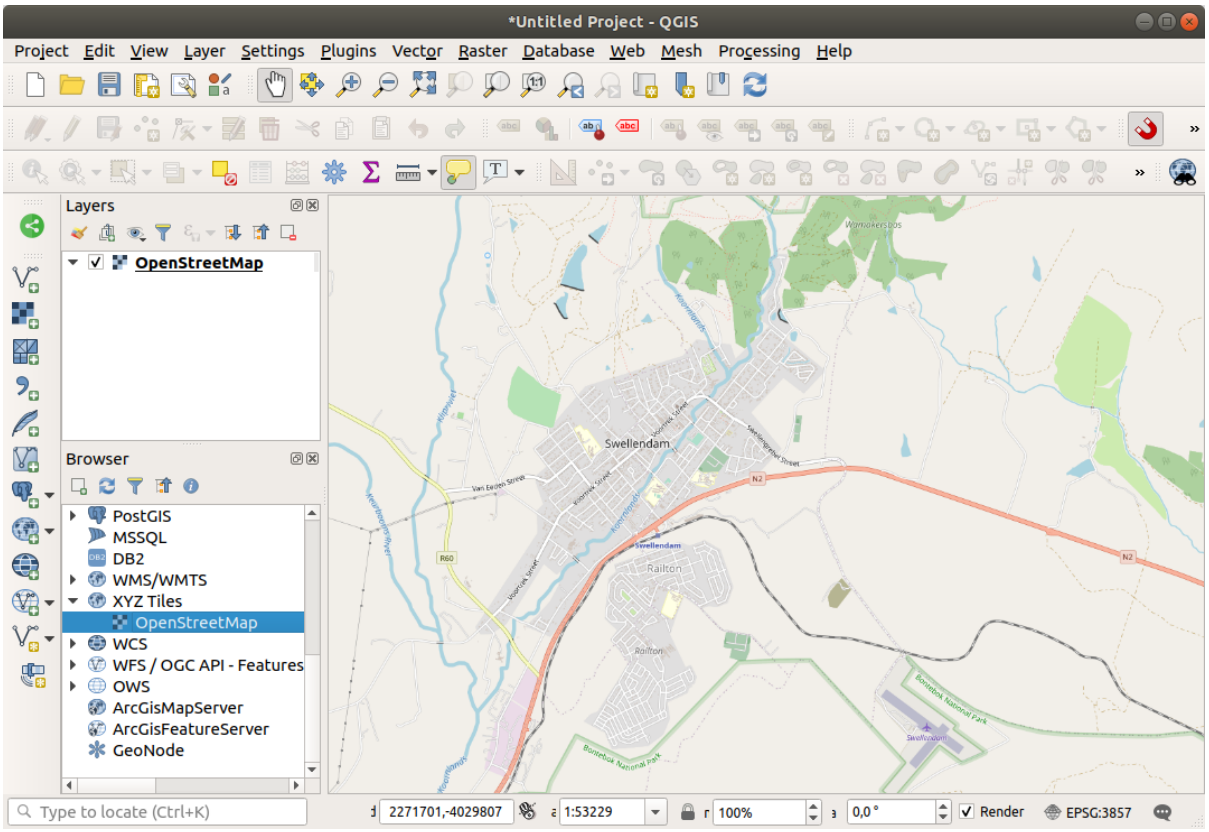
他の地域のデータをダウンロードしたい場合は、 *Introduction Chapter* にジャンプして、その方法を読んでください。

> **注釈**
>
> OSM のダウンロードは一貫したデータフィールドを持っていますが、範囲と詳細は多様になってしまいます。たとえば選択した領域にレストランについての情報が含まれていないとわかった場合は、別の地域を選択することが必要な場合があります。

6.2.4 ★☆☆ （初級レベル） 理解しよう: プロジェクトを開始してデータを取得する

まず、作業するデータを読み込むする必要があります。

1. 新しい QGIS プロジェクトを始めます
2. 必要であれば、背景地図を追加することができます。**ブラウザ** を開き、*XYZ Tiles* メニューから *OSM* 背景地図をロードします。



3. `training_data.gpkg` Geopackage データベースには、この章で使用するほとんどのデータセットがあります:
    1. `buildings`

2. roads
3. restaurants
4. schools

それらを読み込み、さらに landuse.sqlite も読み込みます。

4. レイヤの範囲を拡大すると、Swellendam, 南アフリカが表示されます。

   Before proceeding we will filter the roads layer, in order to have only some specific road types to work with.

   OSM データセットの中のいくつかの道路は unclassified, tracks, path, footway としてリストアップされています。これらの道路をデータセットから除外し、この演習に適した他の道路タイプに焦点を当てたいと思います。

   さらに、OSM のデータはどこでも更新されるとは限らないので、``NULL``の値も除外する。

5. ``roads``レイヤを右クリックし、 **フィルタ**... を選択します。

6. ポップアップするダイアログで、これらの地物を次の式でフィルタリングします:

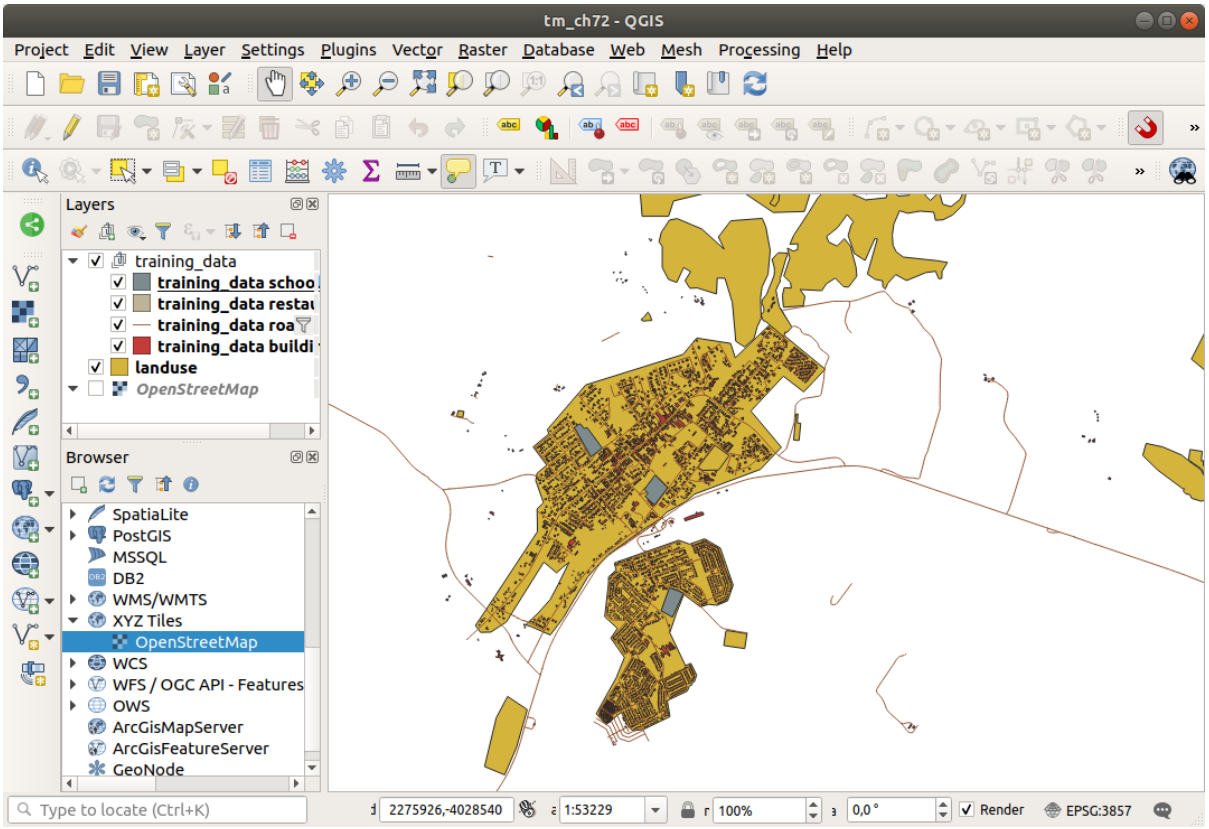
```
"highway" NOT IN ('footway', 'path', 'unclassified', 'track') AND "highway" IS␣
↪NOT NULL
```

   NOT と IN の 2 つの演算子を連結すると、 highway フィールドにこれらの属性値を持つすべての地物を除外することができます。

   IS NOT NULL と AND 演算子を組み合わせると、highway フィールドに値がない道路は除外されます。

   Note the  icon next to the roads layer. It helps you remember that this layer has a filter activated, so some features may not be available in the project.

すべてのデータが入った地図は、次のように表示されます:



6.2.5 ★☆☆ （初級レベル） 自分でやってみよう: レイヤ CRS の変換

レイヤ内の距離を測定するつもりなので、レイヤの CRS を変更する必要があります。これを行うには、各レイヤを順番に選択し、新しい投影法で新しいレイヤに保存し、その新しいレイヤをマップにインポートする必要があります。

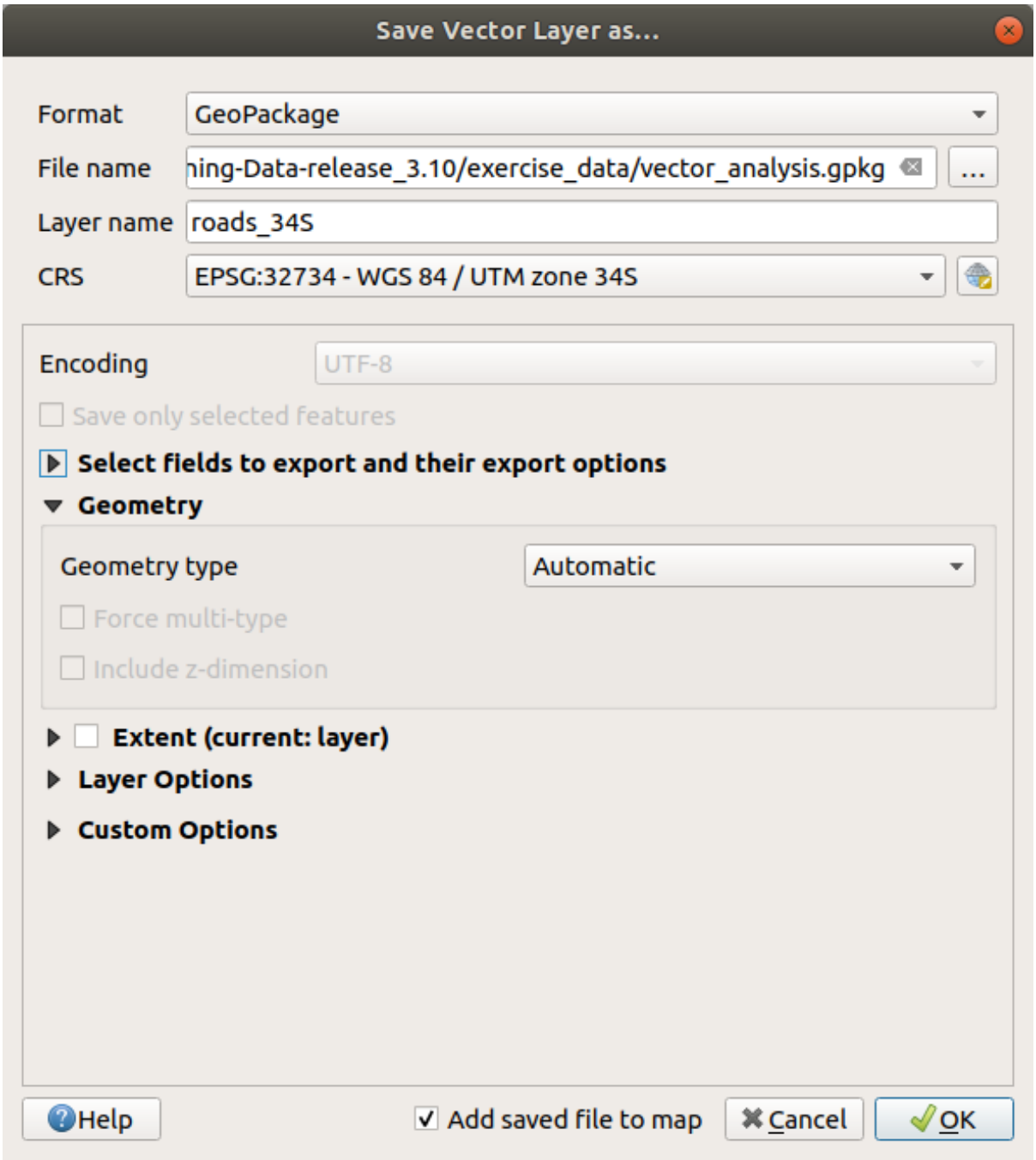
例えば、各レイヤを ESRI シェープファイル形式のデータセットとしてエクスポートしたり、既存の GeoPackage ファイルにレイヤを追加したり、別の GeoPackage ファイルを作成し、そこに新たに再投影したレイヤで埋め尽くしたりすることが可能です。ここでは、最後の方法を紹介しますので、 `training_data.gpkg` はきれいなままです。ご自分に合ったワークフローを自由に選択してください。

> **注釈**
>
> この例では、*WGS 84 / UTM zone 34S* CRS を使用していますが、あなたの地域に適した UTM CRS を使用する必要があります。

1. Right click the `roads` layer in the *Layers* panel
2. **エクスポート** --> **新規ファイルに地物を保存...** をクリックします
3. **名前をつけてベクタレイヤを保存** ダイアログで、 *GeoPackage* を **形式** として選択します
4. ... の **ファイル名** をクリックし、新しい GeoPackage の名前を `vector_analysis` にします
5. **レイヤ名** を `roads_34S` に変更します

6. *CRS* を *WGS 84 / UTM zone 34S* に変更します

7. *OK* をクリックします:



これにより、新しい GeoPackage データベースが作成され、`roads_34S` レイヤが追加されます。

8. この作業を各レイヤに対して繰り返してください。`vector_analysis.gpkg` GeoPackage ファイルに元の名前の後に `_34S` を付けた新しいレイヤを作成します。

   macOS では、ポップアップするダイアログで *Replace* ボタンを押して QGIS に既存の GeoPackage を上書きさせます。

> **注釈**
>
> 既存の GeoPackage にレイヤを保存する場合、同じ名前のレイヤが既に存在しなければ、QGIS はそのレイヤを GeoPackage 内の既存のレイヤの隣に **追加** します。

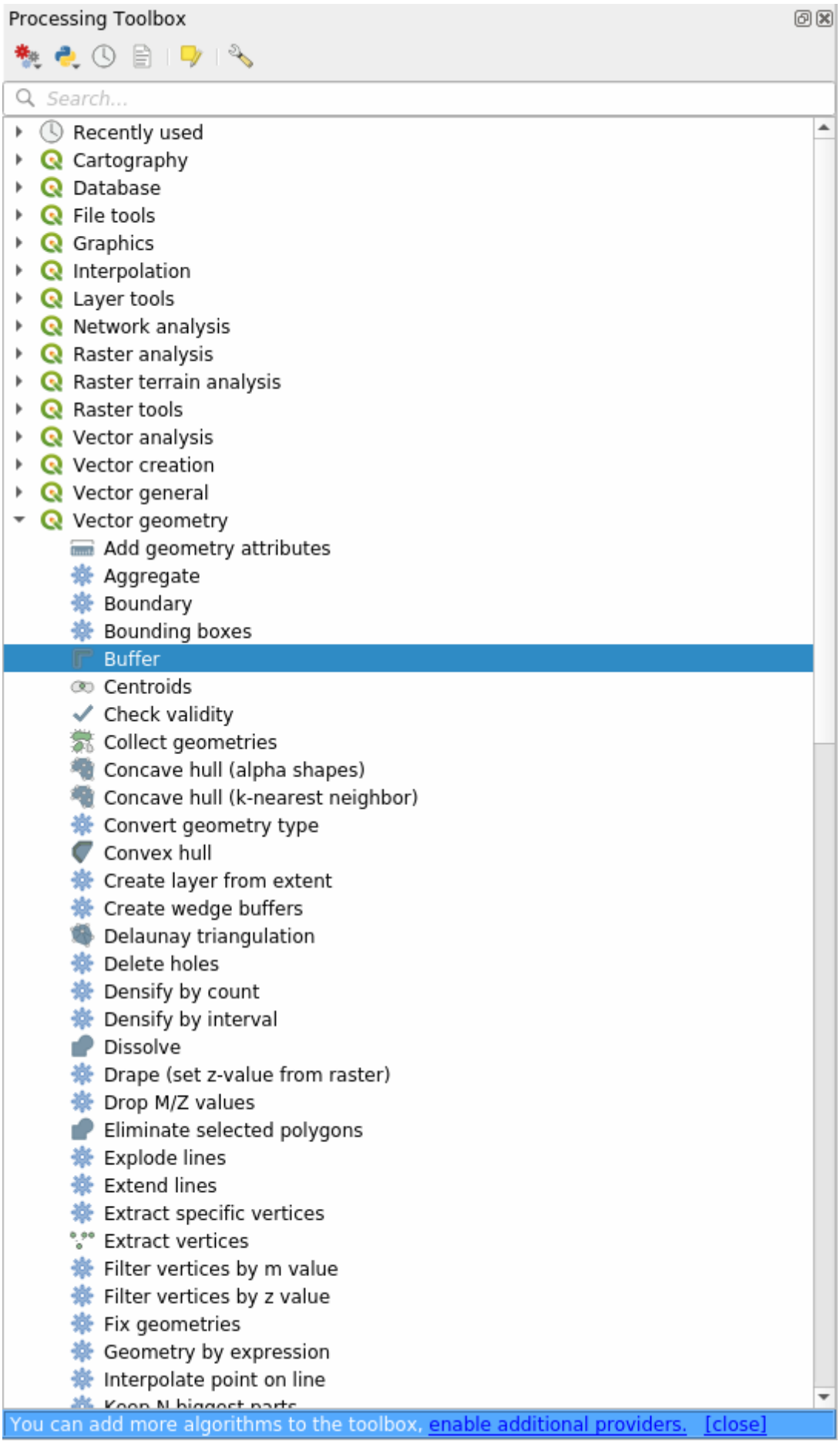
9. プロジェクトから古いレイヤを削除する
10. すべてのレイヤの処理が完了したら、任意のレイヤを右クリックして **レイヤの領域にズーム** をクリックすると、地図が関心域にフォーカスされます。

さて、OSM データを UTM 投影に変換したので、計算を始められます。

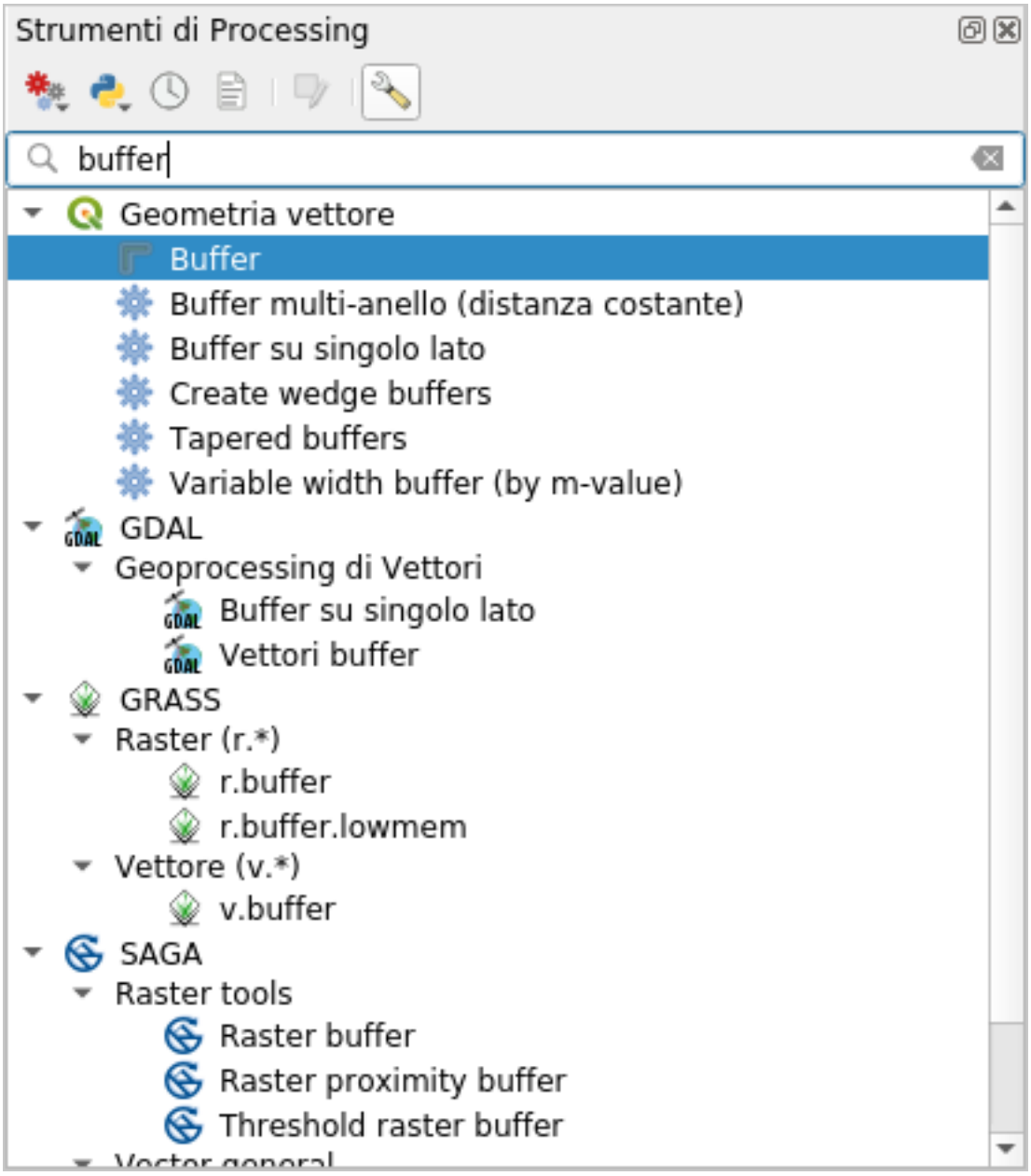
6.2.6 ★☆☆ （初級レベル） 理解しよう: 学校と道路からの距離

QGIS では、任意のベクタオブジェクト間の距離を計算することができます。

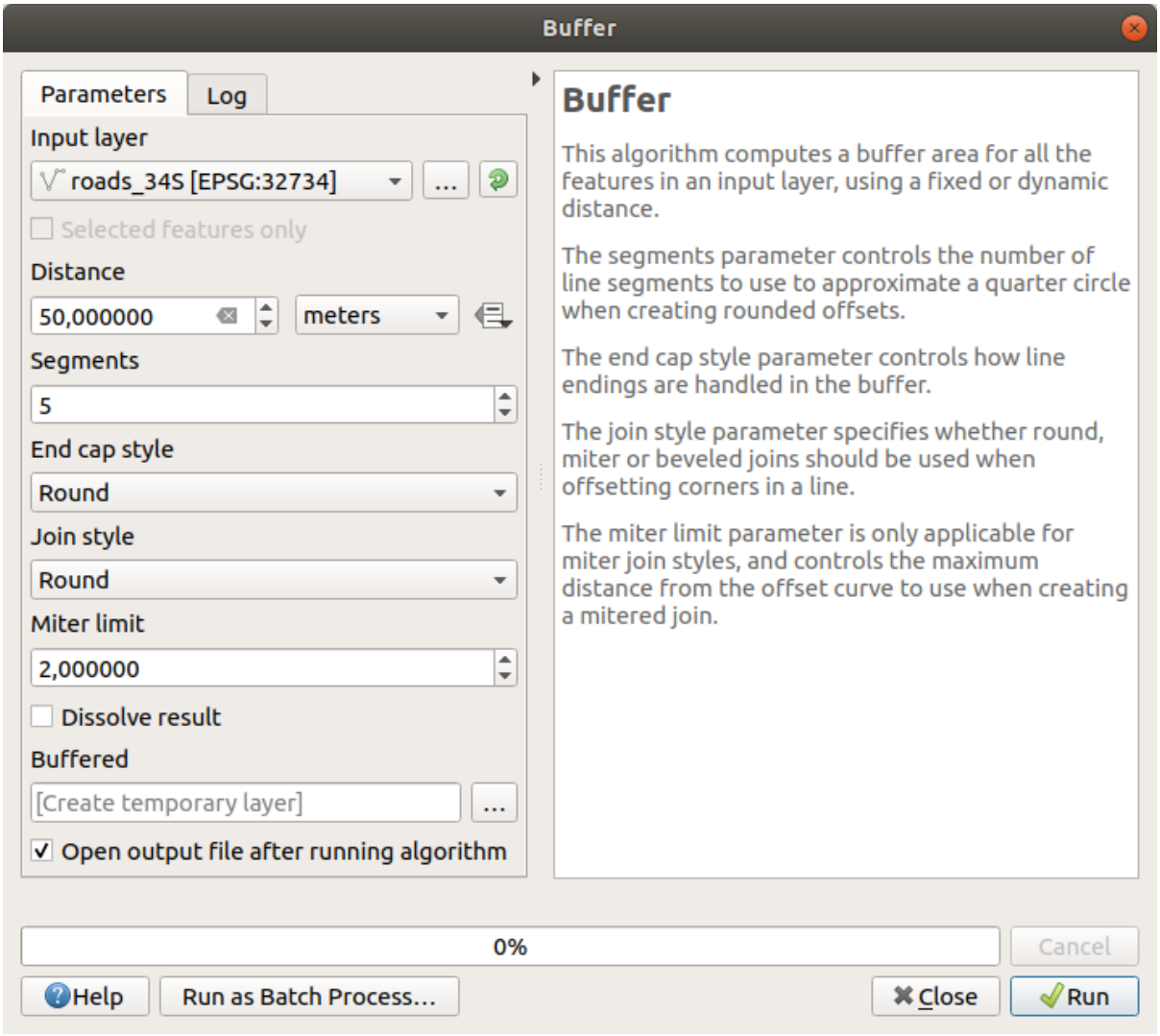
1. `roads_34S` と `buildings_34S` のレイヤのみが表示されていることを確認します（作業中の地図を扱い易くするため）
2. **プロセシング　ツールボックス** をクリックして、QGIS の解析の **コア** を開きます。基本的に、 **すべての** アルゴリズム（ベクタ **および** ラスタ分析）がこのツールボックスで利用可能です。
3. まずは *Buffer* アルゴリズムを使用して `roads_34S` の周囲の領域を計算します。このアルゴリズムは **ベクタジオメトリ** グループで見つけることができます。


Processing Toolbox
Search...
Recently used
Cartography
Database
File tools
Graphics
Interpolation
Layer tools
Network analysis
Raster analysis
Raster terrain analysis
Raster tools
Vector analysis
Vector creation
Vector general
Vector geometry
Add geometry attributes
Aggregate
Boundary
Bounding boxes
Buffer
Centroids
Check validity
Collect geometries
Concave hull (alpha shapes)
Concave hull (k-nearest neighbor)
Convert geometry type
Convex hull
Create layer from extent
Create wedge buffers
Delaunay triangulation
Delete holes
Densify by count
Densify by interval
Dissolve
Drape (set z-value from raster)
Drop M/Z values
Eliminate selected polygons
Explode lines
Extend lines
Extract specific vertices
Extract vertices
Filter vertices by m value
Filter vertices by z value
Fix geometries
Geometry by expression
Interpolate point on line
You can add more algorithms to the toolbox, enable additional providers. [close]

または、ツールボックスの上部にある検索メニューに バッファ と入力してください:



4. ダブルクリックしてアルゴリズムダイアログを開きます
5. 入力レイヤとして `roads_34S` を選択し、 **距離** を 50 に設定し、残りのパラメータは既定値を使用します。

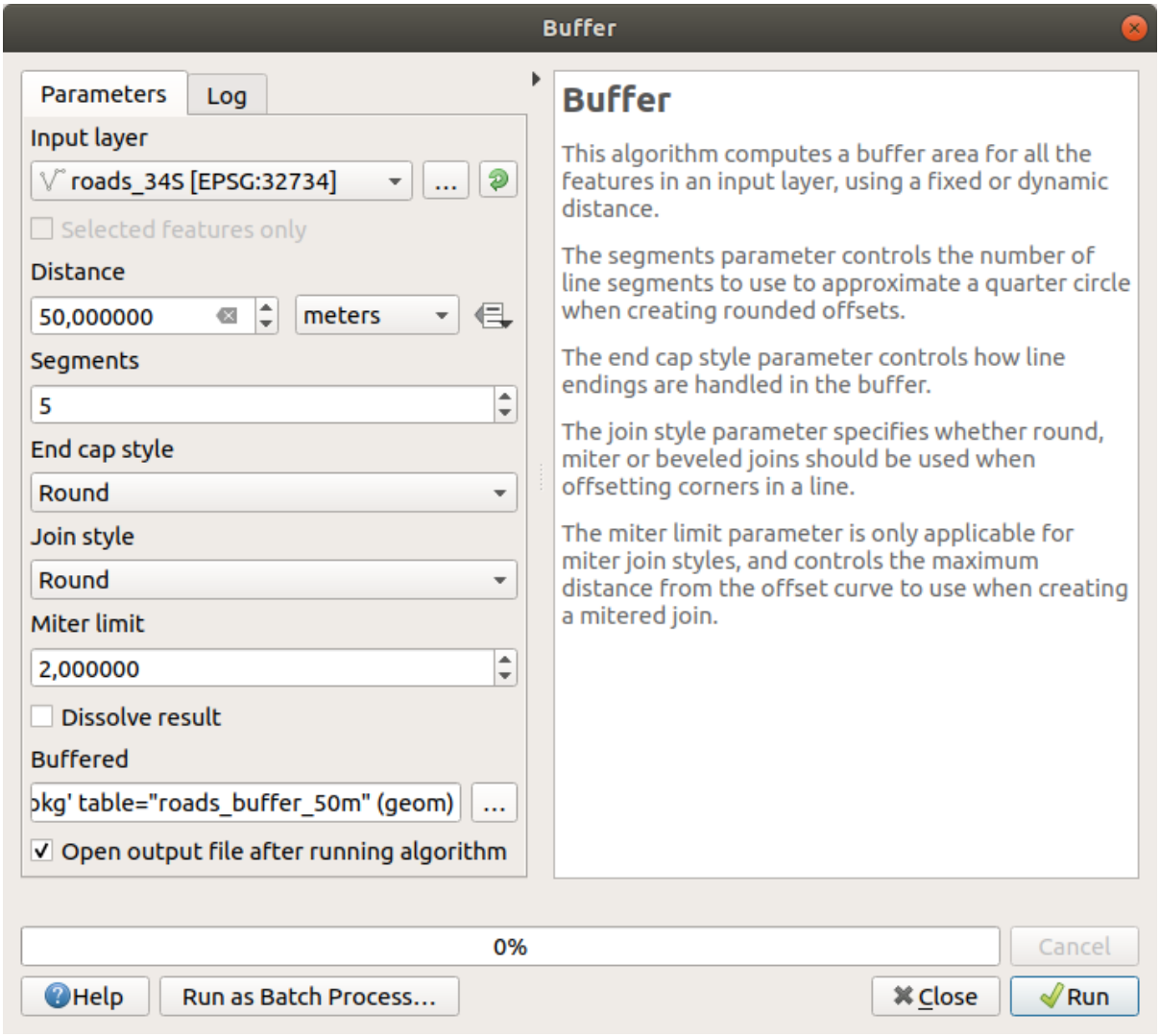


6. 既定の **距離** はメートル単位です。これは、入力データセットが、メートルを基本的な測定単位として使用する投影座標系なためです。コンボボックスを使って、キロメートルやヤードなど、他の投影単位を選択することができます。

> **注釈**
>
> 地理座標系を持つレイヤにバッファを作成しようとすると、プロセシングは警告を発し、メトリック座標系にレイヤーを再投影するよう提案します。

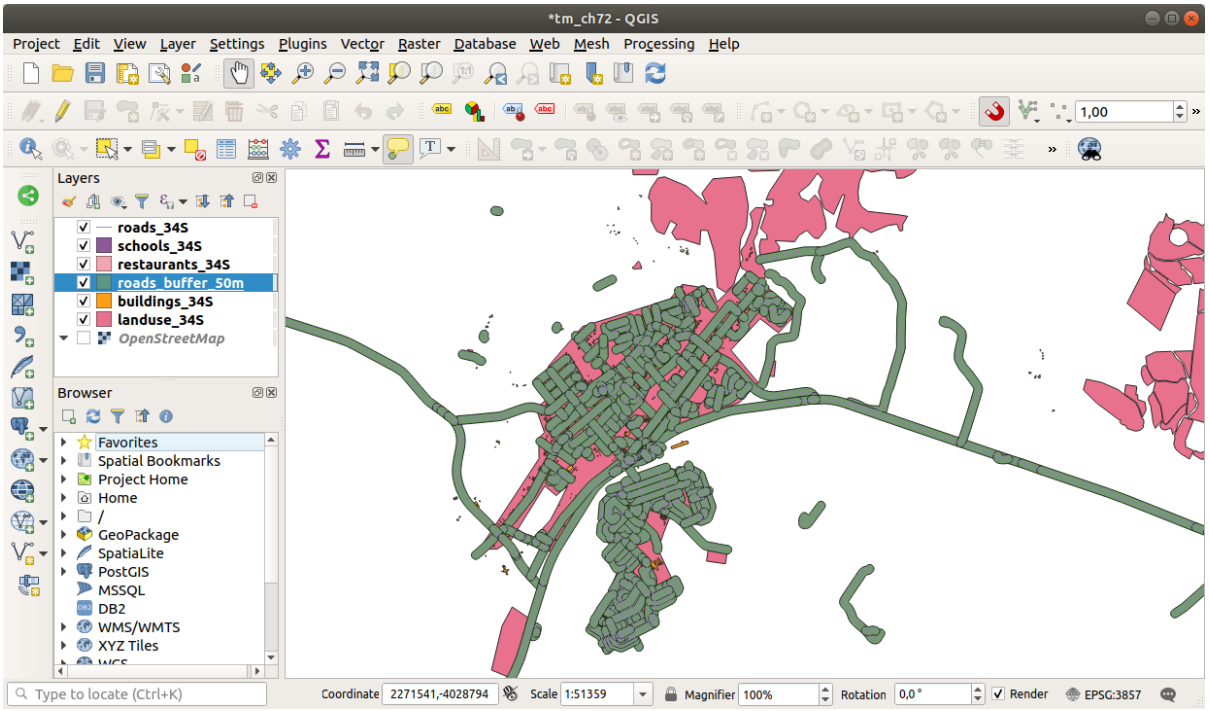
7. 既定で **プロセシング** は一時的なレイヤを作成し、それらを **レイヤ** パネルに追加します。また、その結果を GeoPackage データベースに追加することもできます:

   1. ... ボタンをクリックして、 *GeoPackage* **に保存**... を選択します
   2. 新しいレイヤの名前 `roads_buffer_50m` にします
   3. これを `vector_analysis.gpkg` ファイルに保存します



8. **実行** をクリックし、次に **バッファ** ダイアログを閉じます

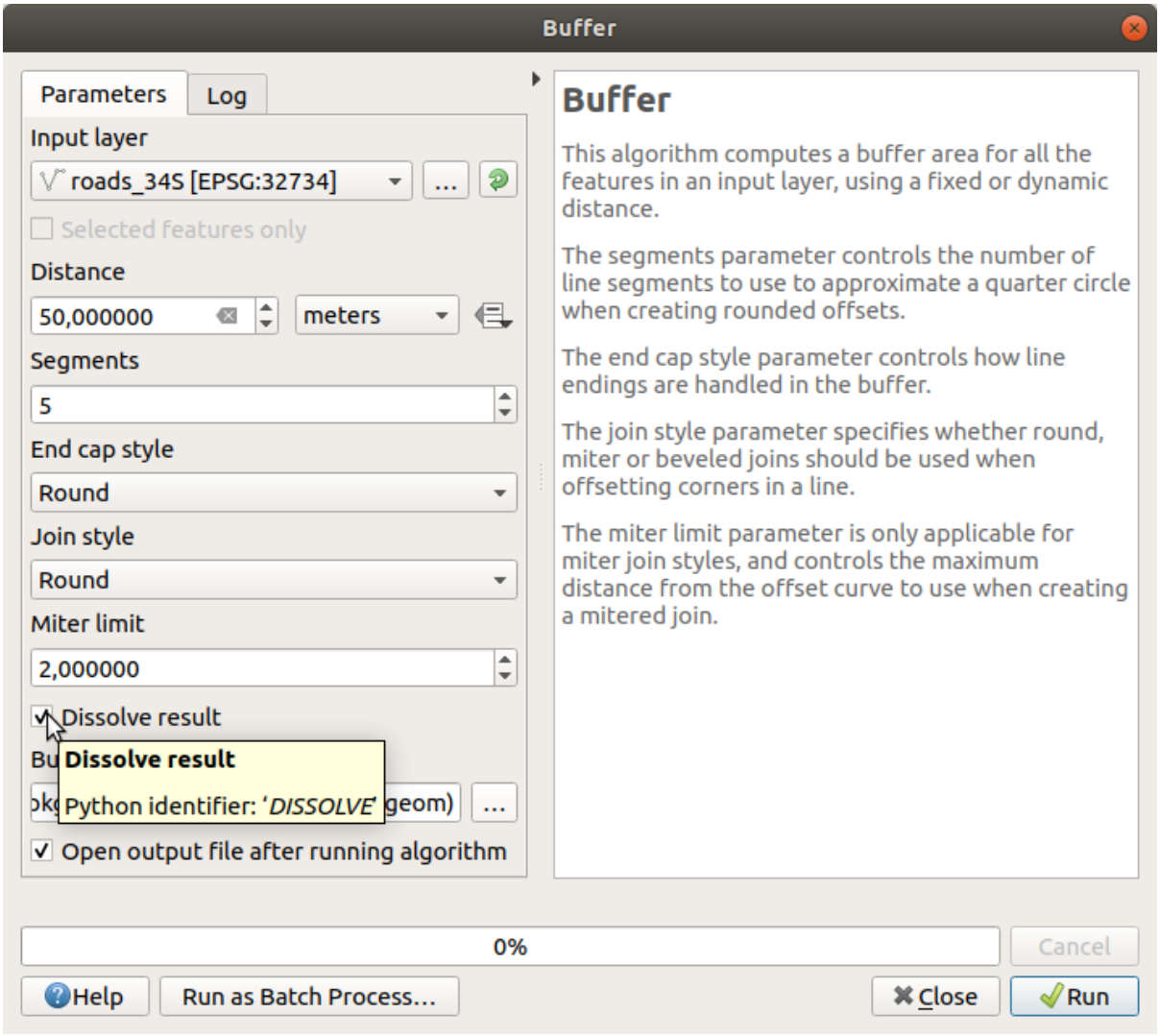
   地図は次のようになっているでしょう：



新しいレイヤが **レイヤ** リストの一番上にある場合、おそらく地図の大部分が見えなくなりますが、これで地域内の道路から 50m 以内のすべての地域が表示されます。

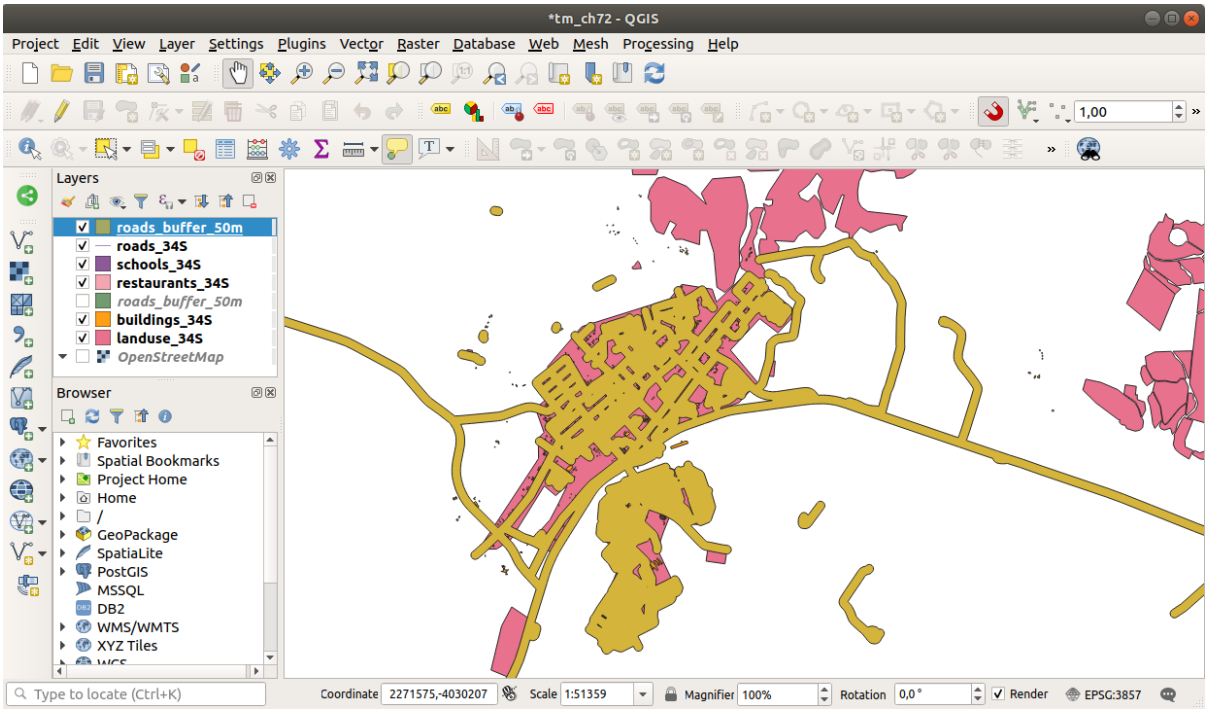
バッファの中に、それぞれの道路に対応する明確な地域があることに注意してください。この問題を解消するために;

1. Uncheck the `roads_buffer_50m` layer and re-create the buffer with *Dissolve results* enabled.



2. Save the output as `roads_buffer_50m_dissolved`
3. **実行** をクリックして、 **バッファ** ダイアログを閉じます

レイヤを **レイヤ** パネルに追加すると、次のような表示になります:



今不要な下位区分はありません。

注釈

ダイアログの右側にある *Short Help* は、このアルゴリズムがどのように動作するかを説明しています。より詳しい情報が必要な場合は、下部の **ヘルプ** ボタンをクリックすると、アルゴリズムのより詳細なガイドが表示されます。

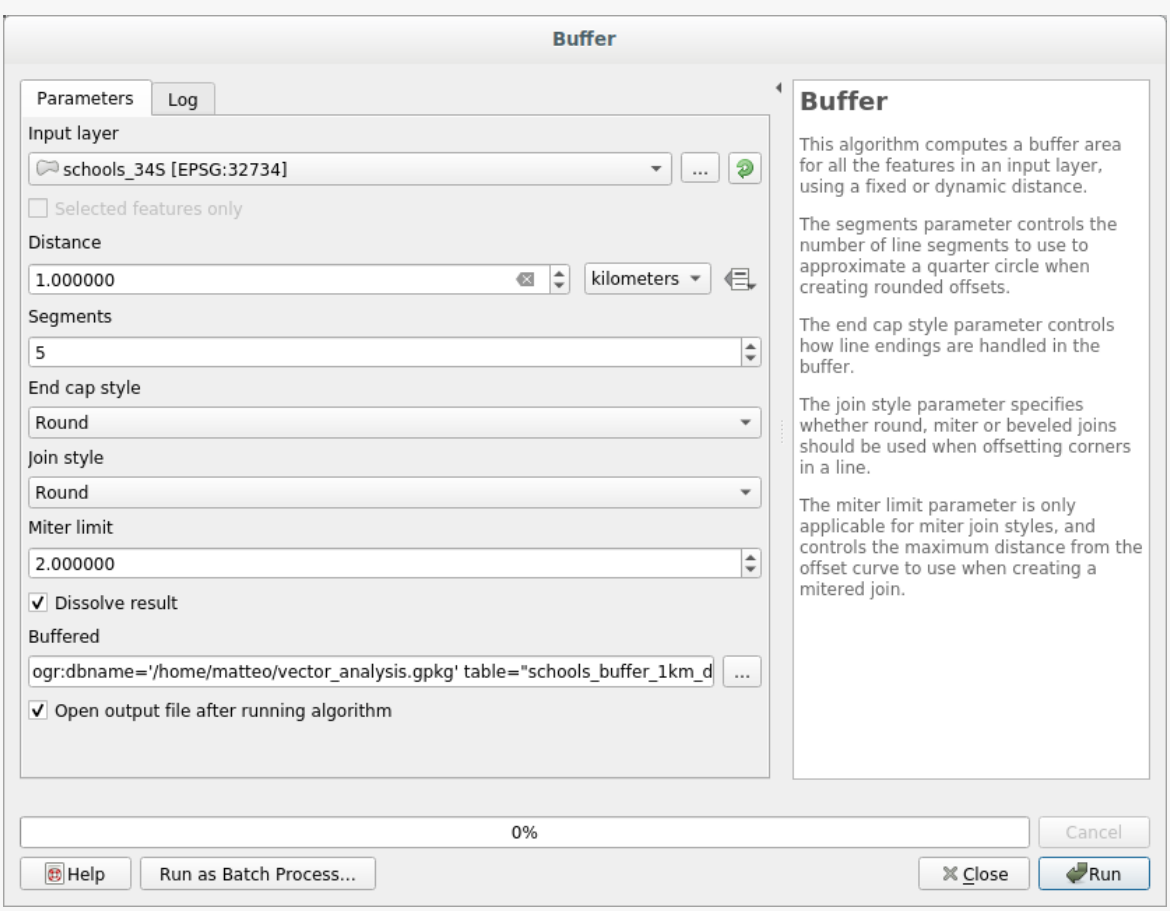
6.2.7 ★☆☆ （初級レベル） 自分でやってみよう: 学校からの距離

上記と同じアプローチを使用し、学校のためのバッファを作成します。

半径は 1 km とします。新しいレイヤーを vector_analysis.gpkg ファイルに schools_buffer_1km_dissolved という名前で保存します。

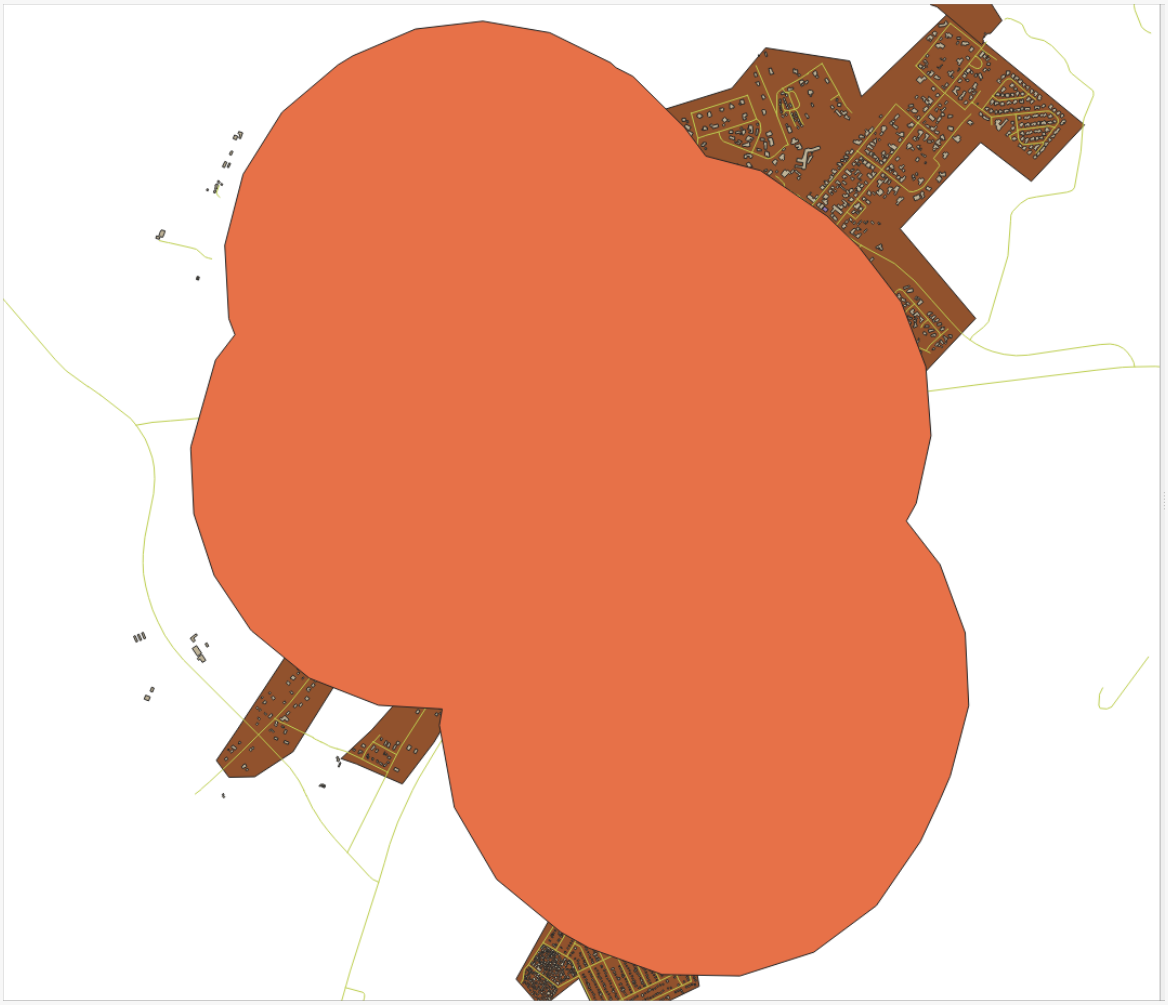
答え

- バッファダイアログはこのように表示されます:

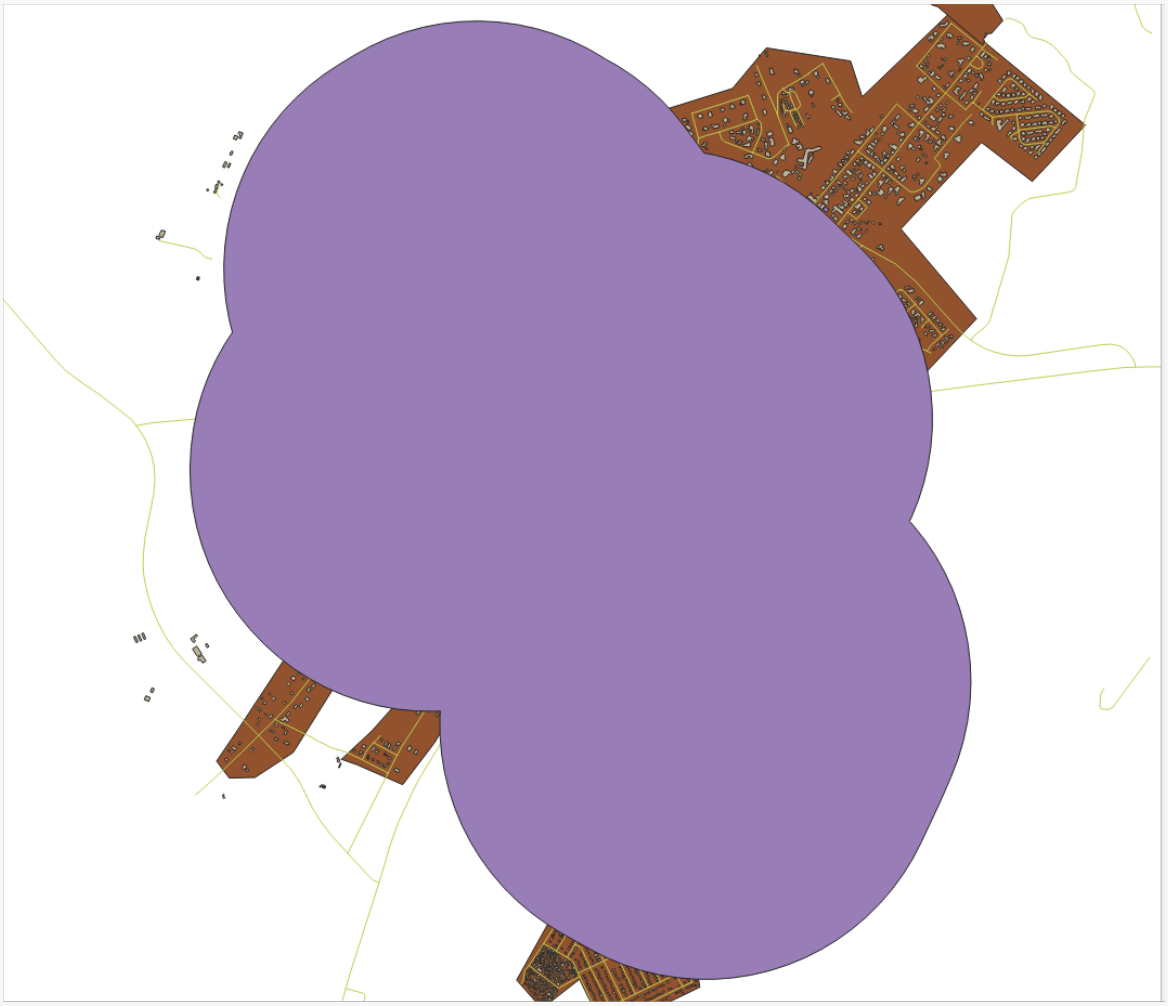


The *Buffer distance* is 1 kilometer.

- The *Segments to approximate* value is set to `20`. This is optional, but it's recommended, because it makes the output buffers look smoother. Compare this:



これと比較してみましょう:

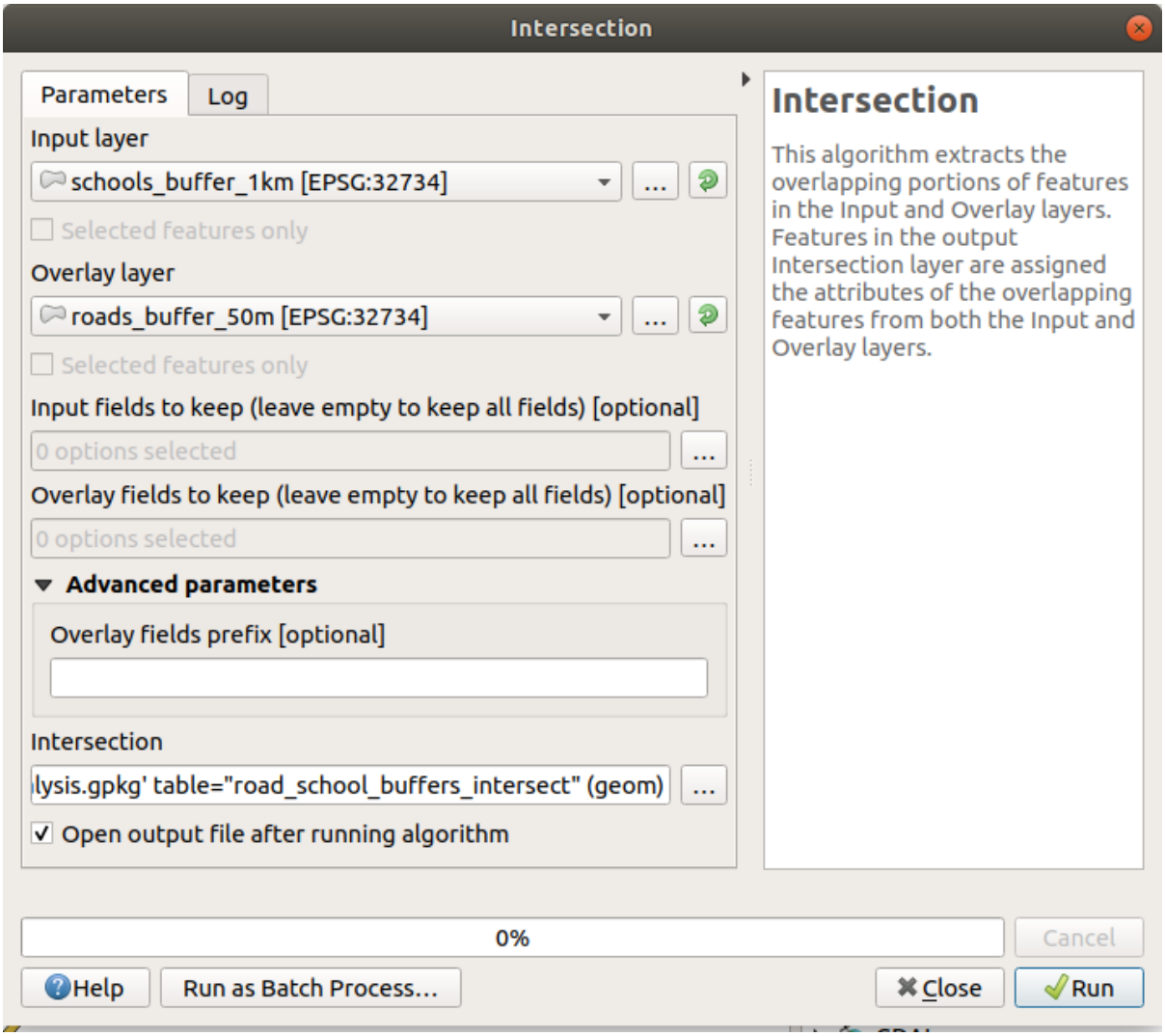


The first image shows the buffer with the *Segments to approximate* value set to `5` and the second shows the value set to `20`. In our example, the difference is subtle, but you can see that the buffer's edges are smoother with the higher value.

6.2.8 ★☆☆ （初級レベル） 理解しよう: 重複している地域

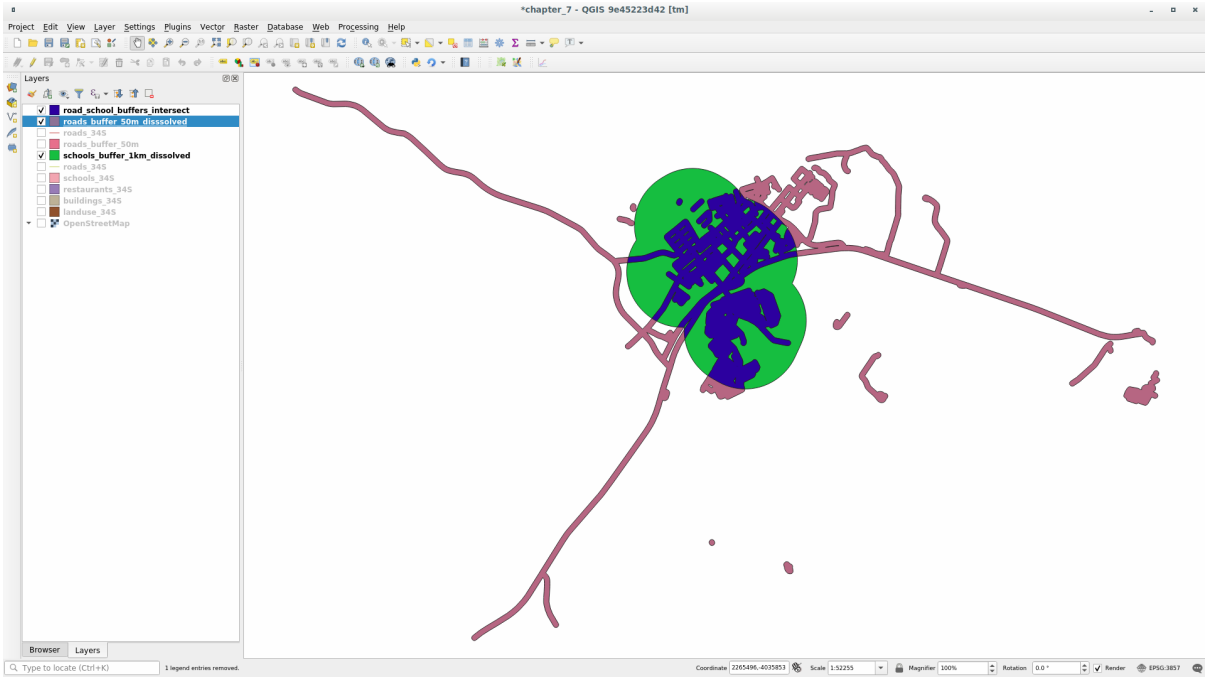
これで、道路から 50m 以内の地域と、1km 以内に学校がある地域（道路を通らない直行便）が特定できました。しかし、明らかに、この両方の条件を満たす地域だけが欲しいのです。そのためには、**交差** *(intersect)* ツールを使用する必要があります。このツールは **プロセシングツールボックス** の **ベクタオーバーレイ** グループで見つけることができます。

1. 2 つのバッファレイヤを **入力レイヤ** と **オーバーレイレイヤ** として使用し、`vector_analysis.gpkg` GeoPackage を **交差** *(intersection)* で、 **レイヤ名** `road_school_buffers_intersect` を選択します。残りは提案のままにしておきます（既定）。

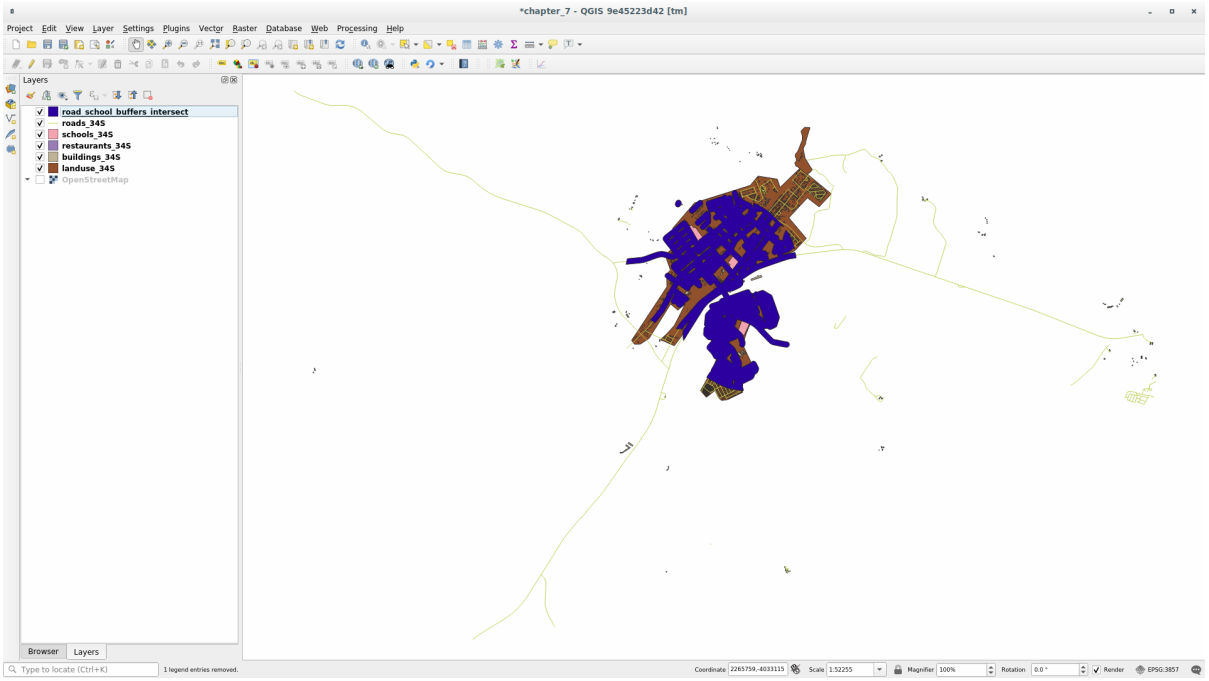


2. **実行** をクリックします。

   下の画像の青い領域が両方の距離基準を満たしたところです。



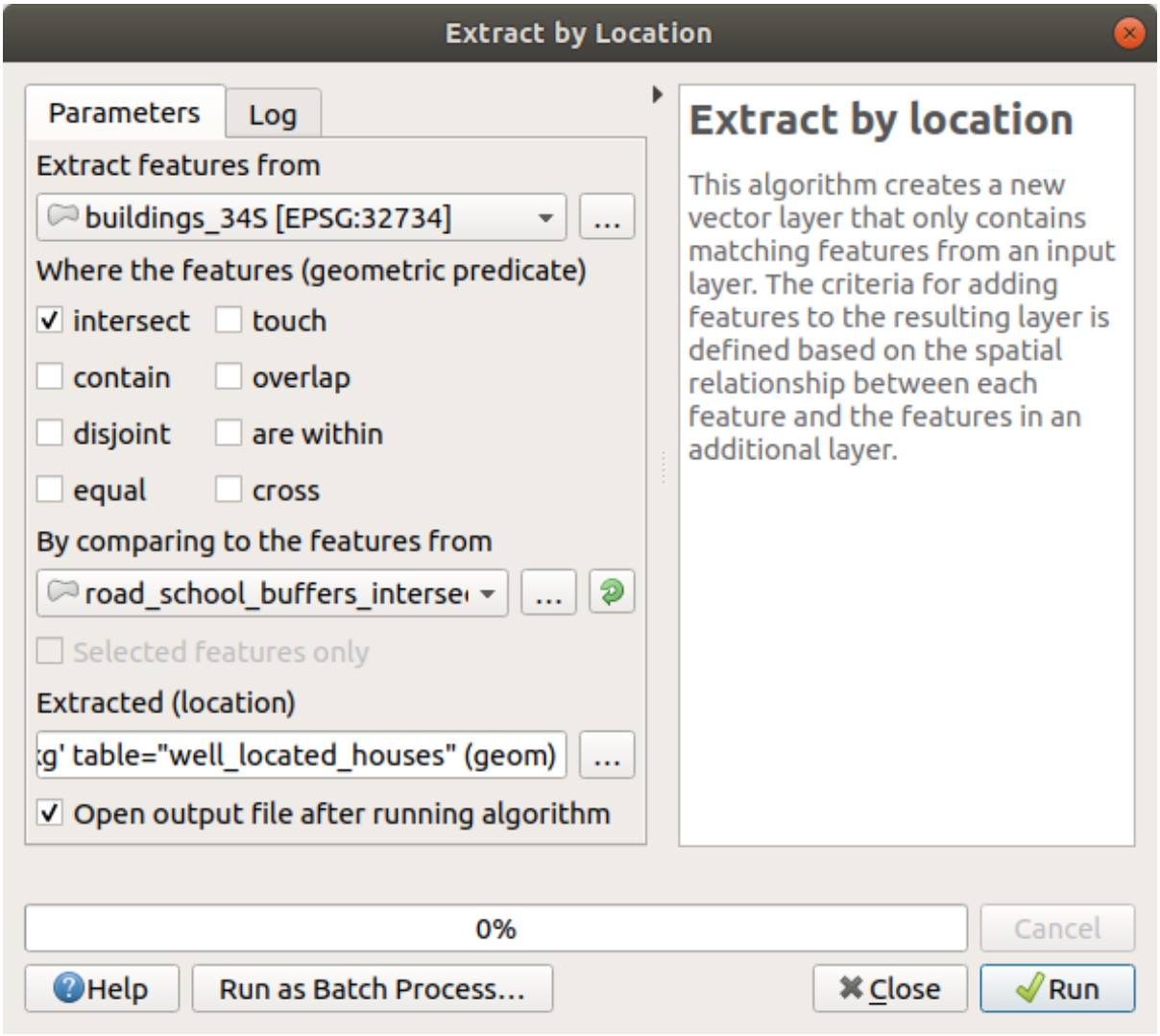
3. 2 つのバッファレイヤを除去しそれらが重なる場所を示すものだけ残しても良いです。それがそもそも本当に知りたかったものですので：



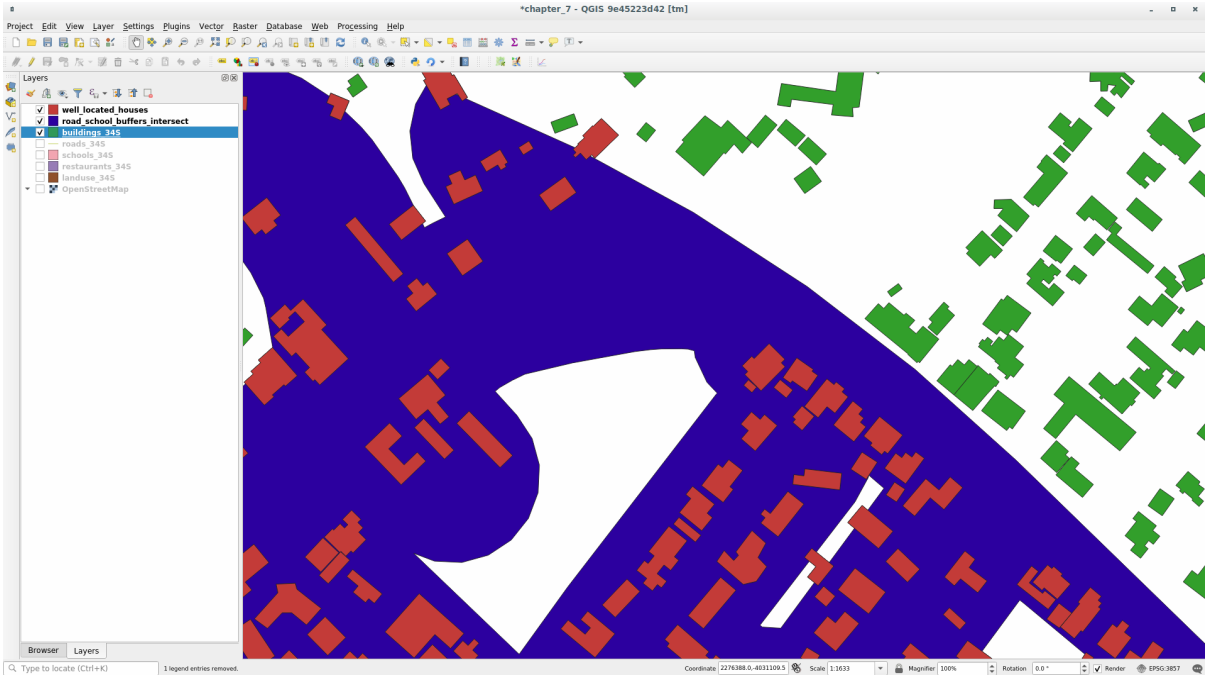
6.2.9 ★☆☆ （初級レベル） 理解しよう: 建物を抽出する

これで、ビルが重なるべき領域ができました。次に、その範囲にある建物を抽出したいですね。

1. **プロセシングツールボックス** から、メニュー項目 **ベクタ選択** **場所による抽出** を探します
2. **抽出する地物のあるレイヤ** で `buildings_34S` を選択します。**空間的関係** で **交差する** *(intersect)* をチェックし、**比較対象の地物のあるレイヤ** でバッファの交差レイヤを選択します。`vector_analysis.gpkg` に保存し、レイヤの名前を `well_located_houses` にします。



3. **実行** をクリックしてダイアログを閉じます
4. You will probably find that not much seems to have changed. If so, move the `well_located_houses` layer to the top of the layers list, then zoom in.



赤色の建物は私たちの基準に合致しているもの、緑色の建物は合致していないものです。

5. これで 2 つのレイヤーが分離され、レイヤリストから `buildings_34S` を削除することができます。

6.2.10 ★★☆ （中級レベル） 自分でやってみよう: さらに建物をフィルタ

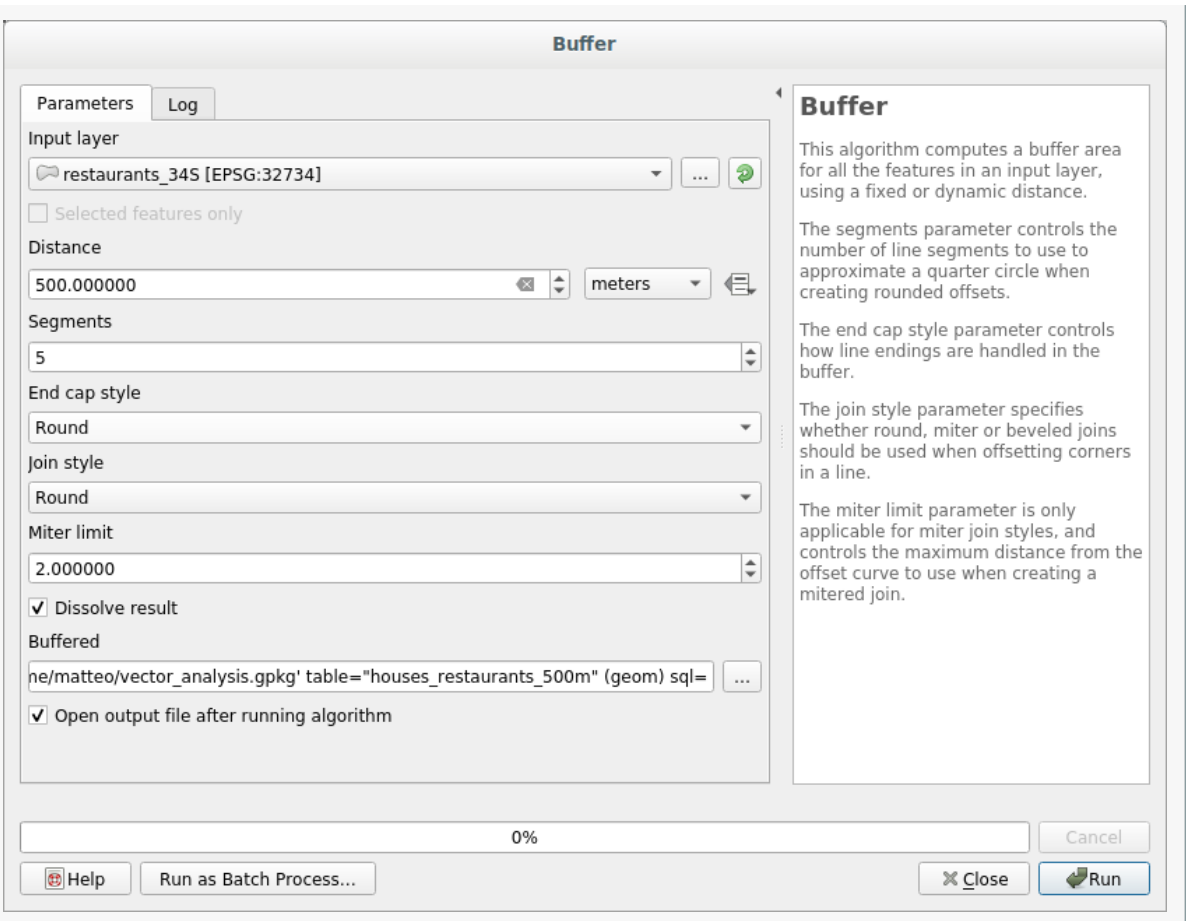
今、私たちの学校の 1 キロ内や道路の 50 メートル内のすべての建物を表示するレイヤーを持っています。今その選択を、レストランの 500 メートル内にある建物を表示するだけに削減する必要があります。

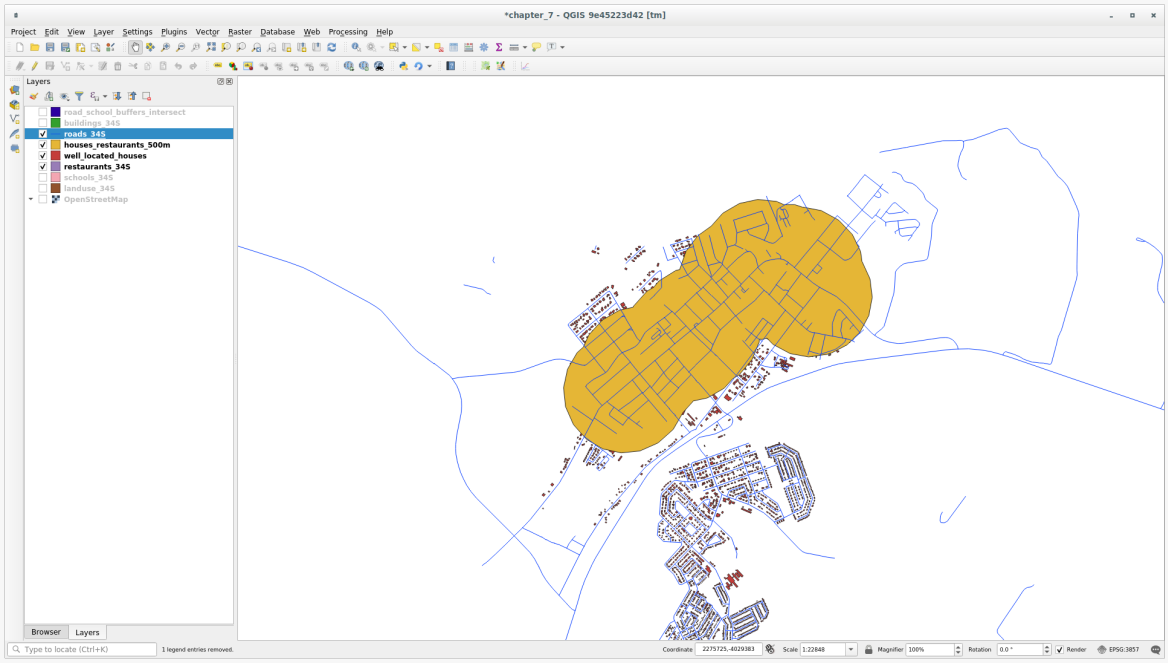
Using the processes described above, create a new layer called `houses_restaurants_500m` which further filters your `well_located_houses` layer to show only those which are within 500m of a restaurant.

答え

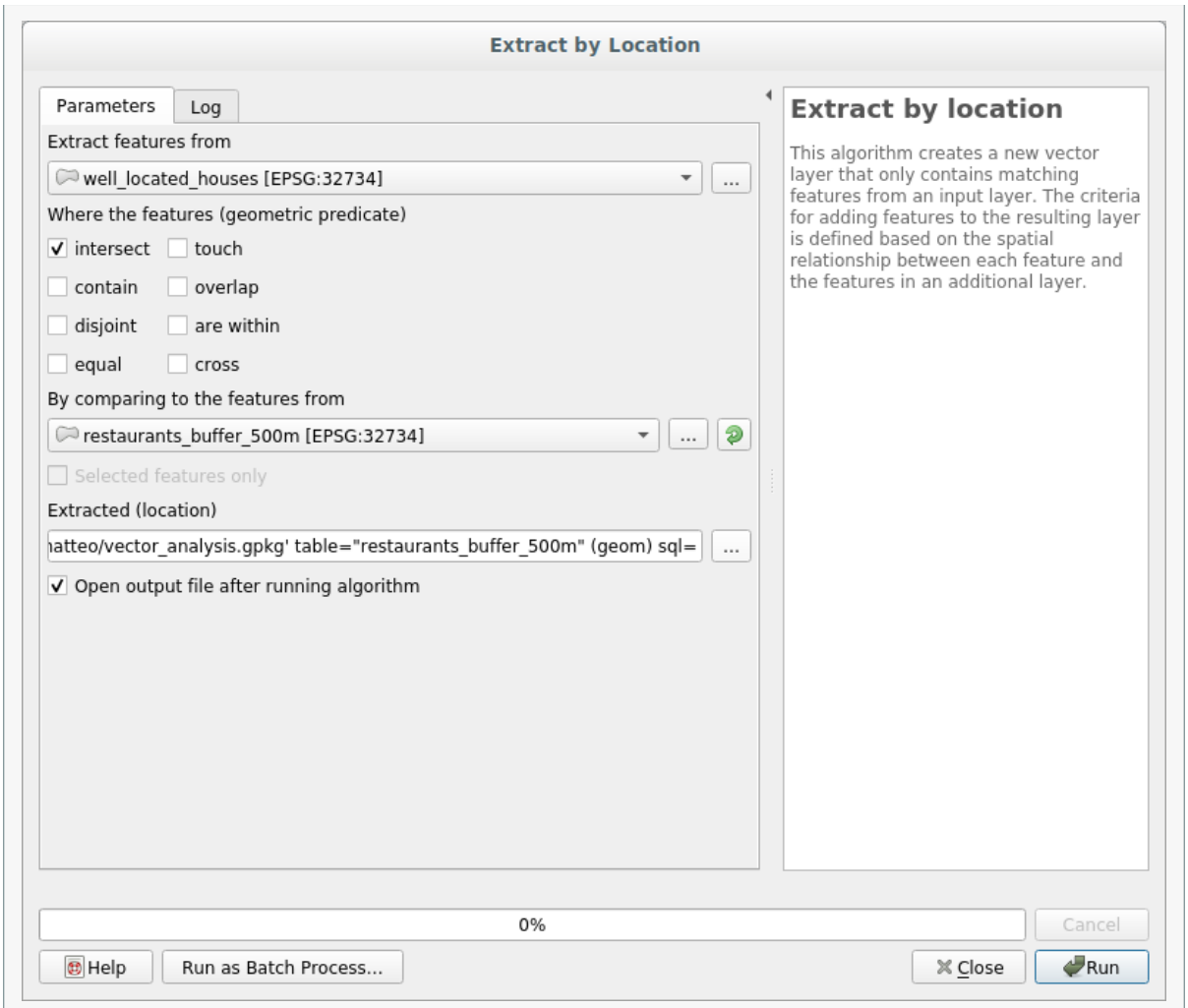
To create the new `houses_restaurants_500m` layer, we go through a two step process:

1. まず、レストランの周囲 500m にバッファを作り、地図にレイヤを追加します:

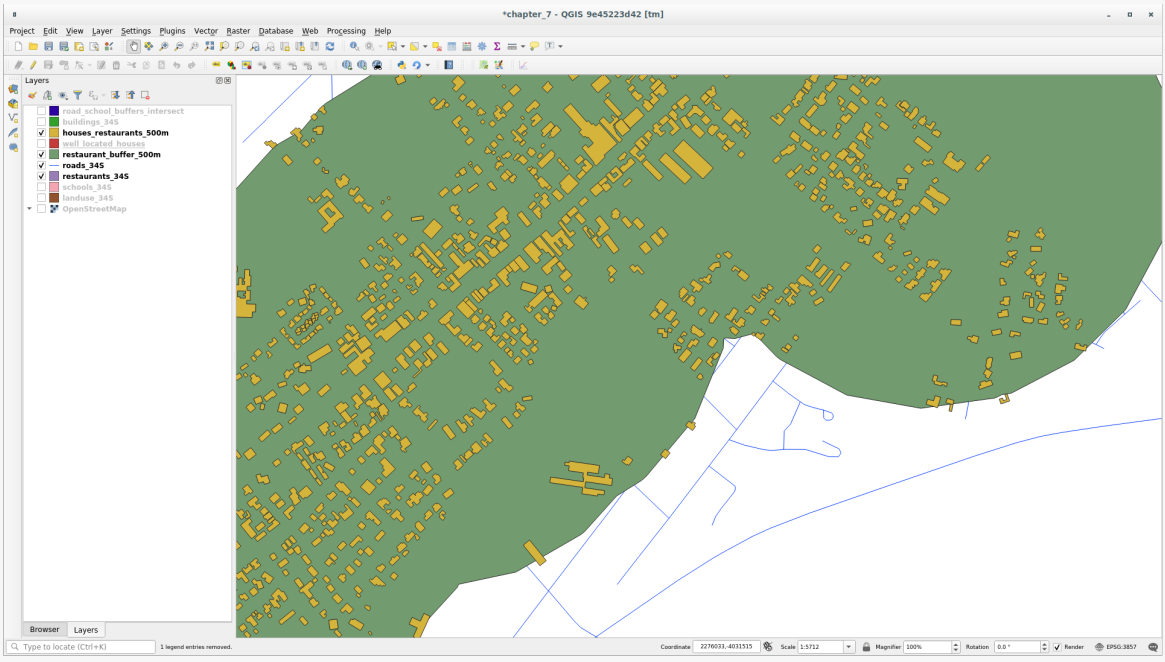




2. 次に、そのバッファ領域内の建物を抽出します;



道路から 50m、学校から 1km、レストランから 500m の範囲にある建物だけが地図に表示されるようになりました:

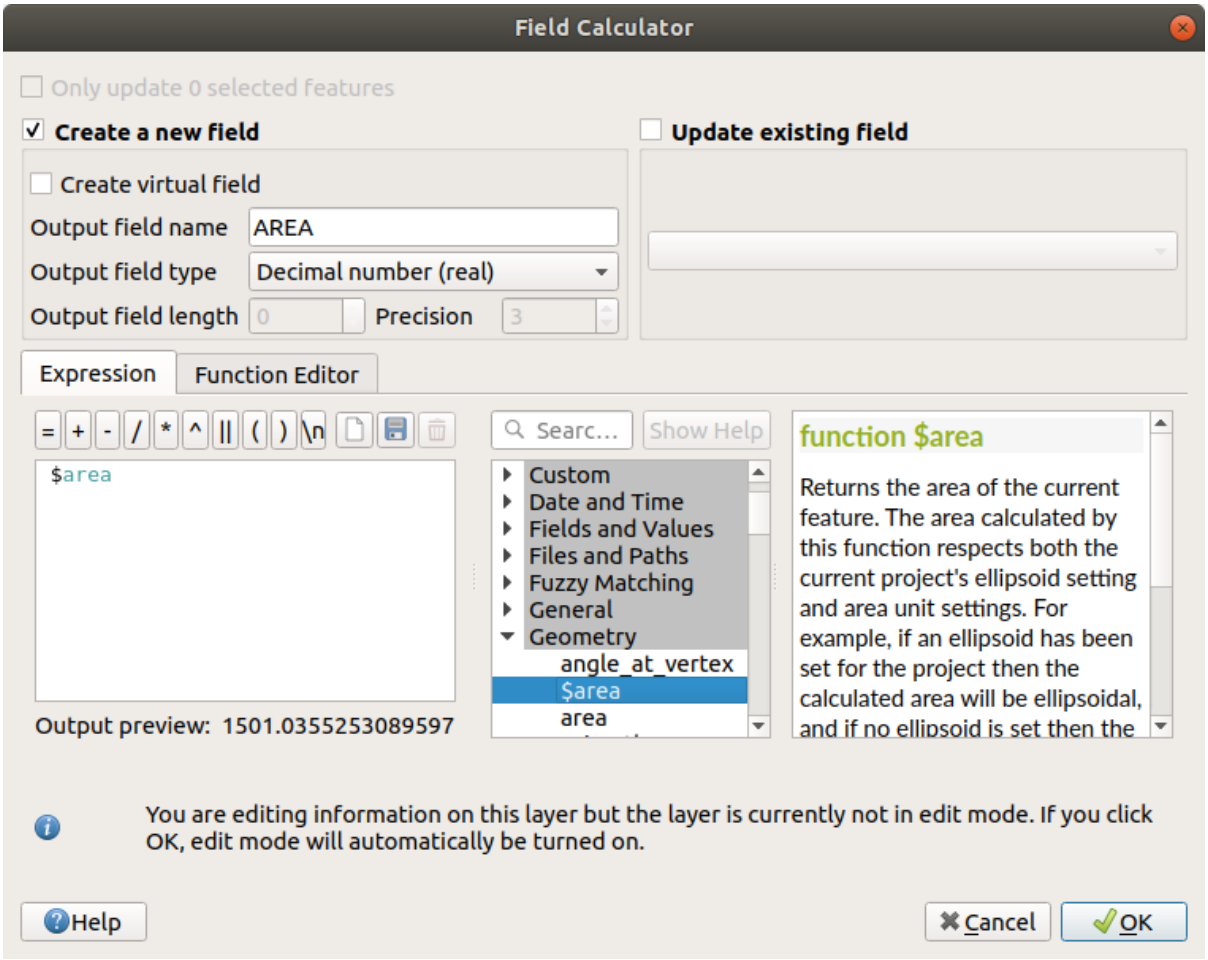


6.2.11 ★☆☆ （初級レベル） 理解しよう: 正しいサイズの建物の選択

どの建物が適正規模（100 平方メートル以上）かを確認するためには、その大きさを計算する必要があります。

1. Select the `houses_restaurants_500m` layer and open the *Field Calculator* by clicking on the  Open Field Calculator button in the main toolbar or in the attribute table window

2. **新規フィールドを作成** を選択し、 **出力する属性の名前** を `AREA` に、 **フィールド型** を **小数点付き数値** *(real)* に、 **地物** グループを選択して、 `$area` を選択します。

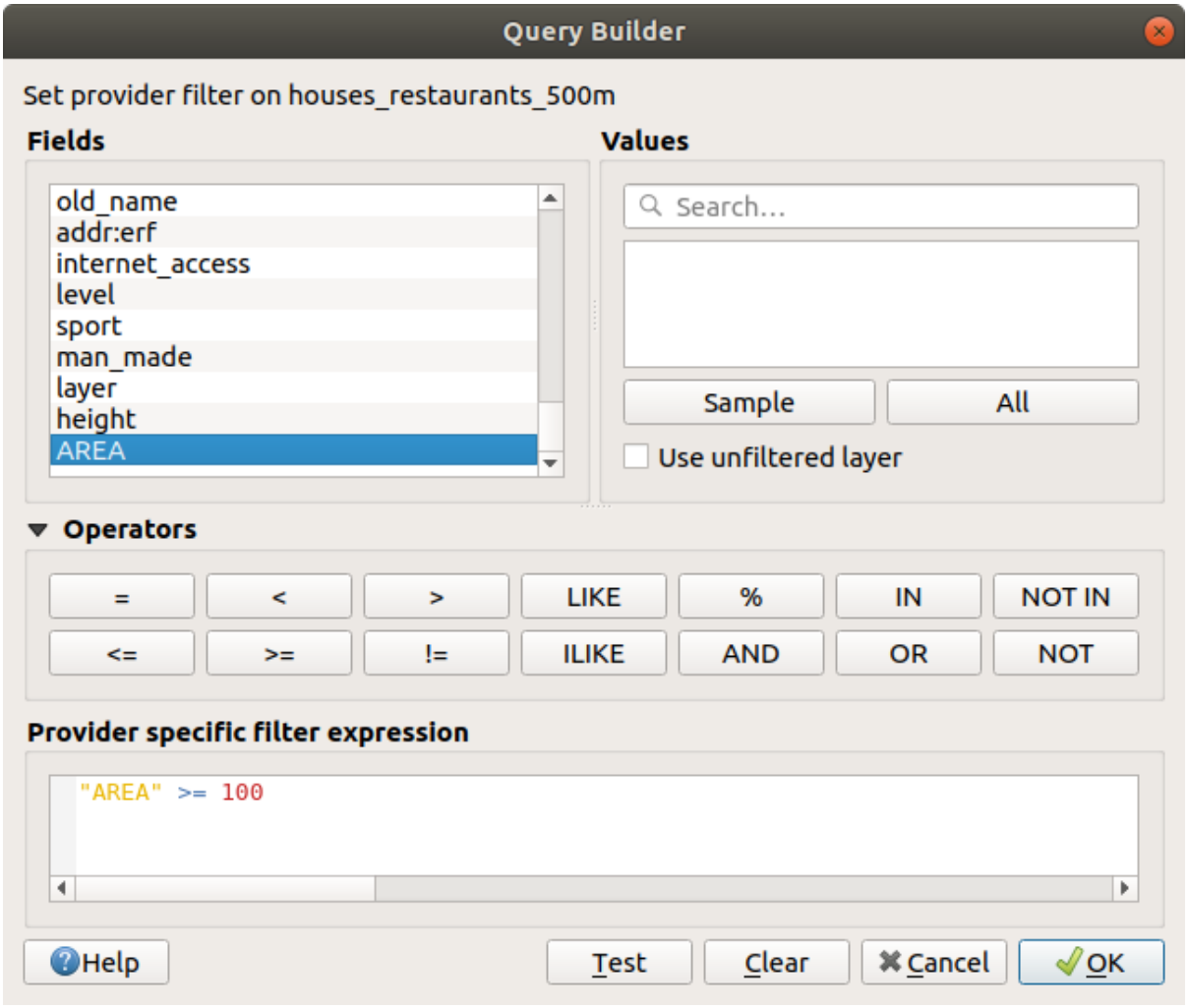


   新しいフィールド `AREA` には、各建物の面積 (平方メートル) が入ります。

3. *OK* **をクリックします。属性テーブルの末尾に** `*AREA*` フィールドが追加されました。

4. 編集を終了するには  編集モード切替 ボタンをクリックし、プロンプトが表示されたら編集を保存します。

5. レイヤプロパティの **ソース** タブで、 **プロバイダ地物フィルタ** を `"AREA >= 100` に設定します。



6. *OK* をクリックします。

これで地図には、開始条件に一致する 100 平方メートル以上の建物だけが表示されるようになりました。

6.2.12 ★☆☆ （初級レベル） 自分でやってみよう:

上記で学んだ方法で、解決策を新しいレイヤとして保存します。このファイルは同じ GeoPackage データベース内に `solution` という名前で保存されます。

6.2.13 結論

GIS の問題解決アプローチと QGIS のベクトル解析ツールを併用することで、複数の条件を持つ問題を迅速かつ容易に解決することができましたね。

6.2.14 次は?

次のレッスンでは、ある地点から別の地点までの道路に沿った最短距離の計算方法について見ていきます。

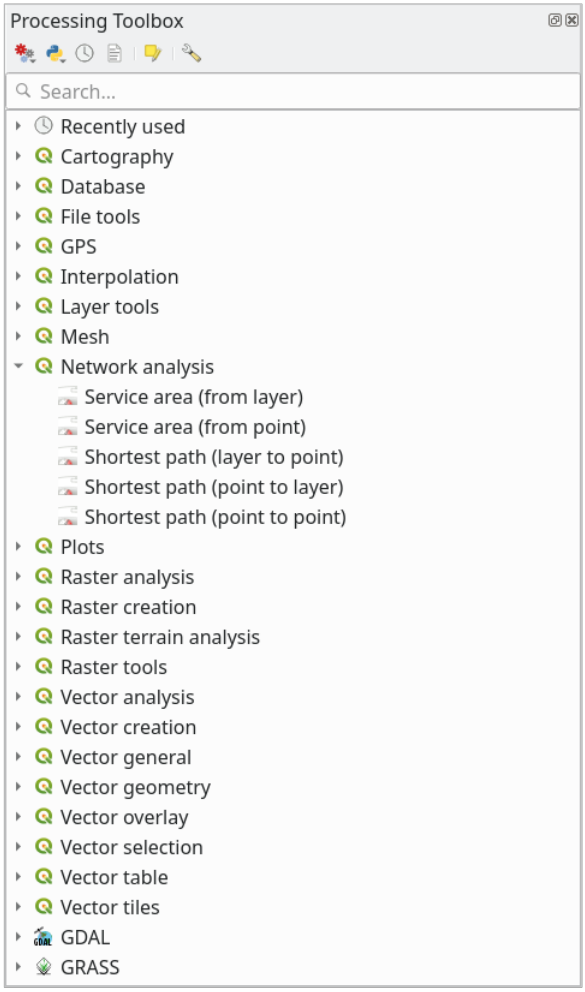
6.3 レッスン: ネットワーク分析

2 点間の最短距離を計算することは、一般的な GIS タスクです。このためのツールは **プロセシングツールボックス** で見つけることができます。

このレッスンの目的 ネットワーク分析 アルゴリズムを使うことを学びます。

6.3.1 ★☆☆ （初級レベル） 理解しよう: ツールとデータ

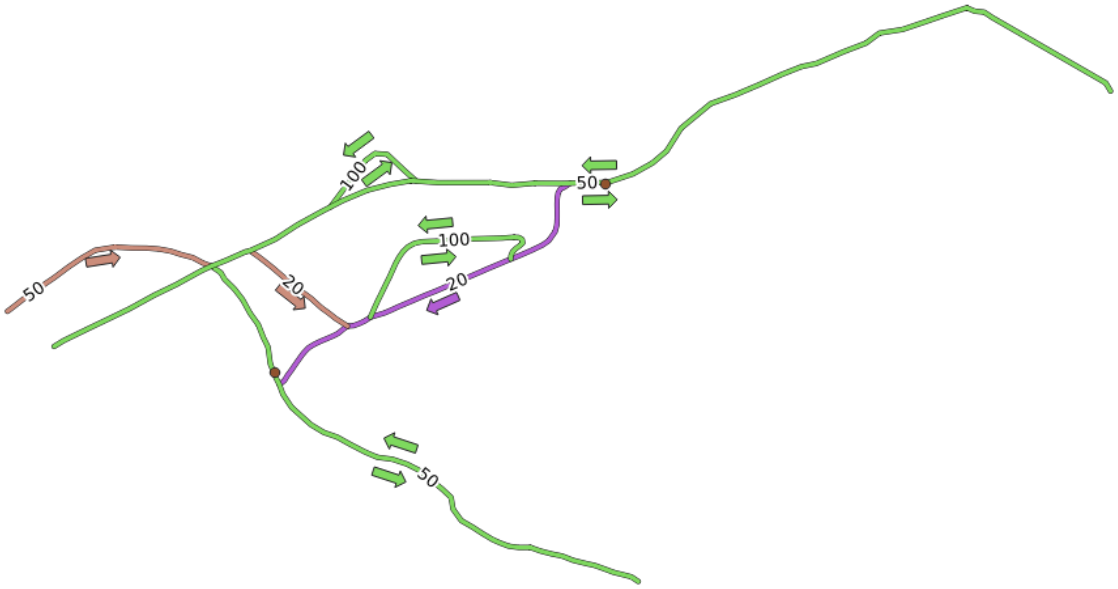
すべてのネットワーク解析アルゴリズムは **プロセシング** - > **ネットワーク分析** メニューにあります。利用可能なツールがたくさんあることがわかります：



プロジェクト `exercise_data/network_analysis/network.qgz` を開いてください。2 つのレイヤーが含まれています:

- `network_points`
- `network_lines`

The `network_lines` layer has already a style that helps to understand the road network.



最短経路ツールは、ネットワークの 2 点間の最短経路または最短経路を計算する方法を提供します:

- 地図上で選択された始点と終点
- 地図上で選択された始点と、ポイントレイヤから取得された終点
- ポイントレイヤから取得された始点と地図上で選択された終点

では始めましょう。

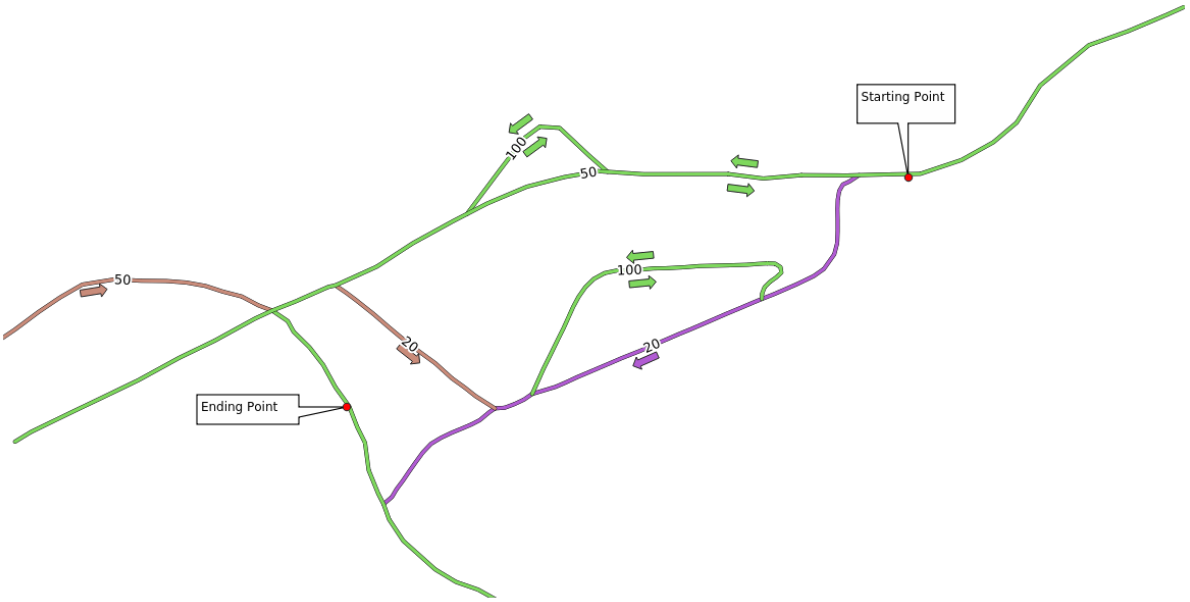
6.3.2 ★☆☆ （初級レベル） 最短経路を計算する（2 地点間）

ネットワーク解析　 最短経路（*2* 地点間） を使うと、地図上の二つの手動で選択した地点間の最短距離を計算できます。

この例では、2 地点間の **最短** （最速ではない）経路を計算します。

1. **最短経路（指定始点から指定終点）** アルゴリズムを開きます
2. Select `network_lines` for *Vector layer representing network*
3. **計算するパスの種類** に 最短 を使用します

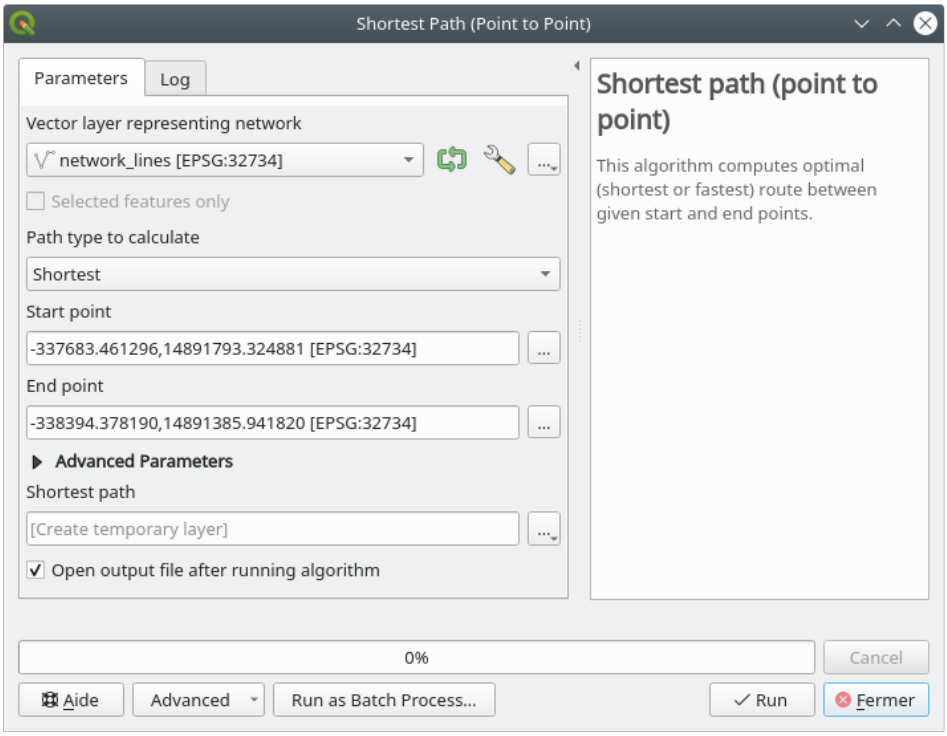
   この 2 点を解析の起点と終点とします:



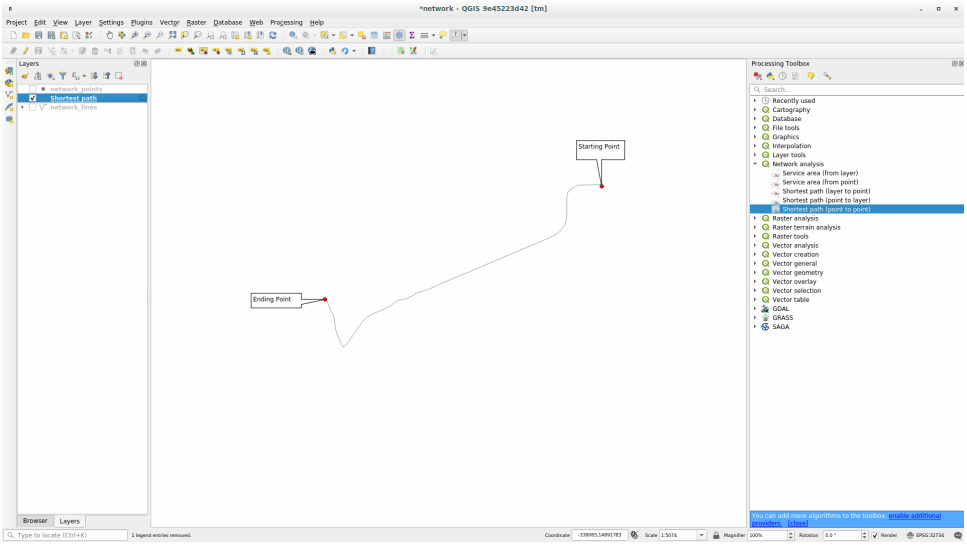
4. **始点** *(x, y)* `の隣にある` *:guilabel:...`* ボタンをクリックし、絵の中の *Starting Point`* とタグ付けされている場所を選択します。正確に選択するために、スナップオプションを有効にしてください。クリックした地点の座標が追加されます。

5. 同じことをしますが、 **終点** に `Ending point` というタグが付けられた場所を選びます。

6. **実行** ボタンをクリックします：



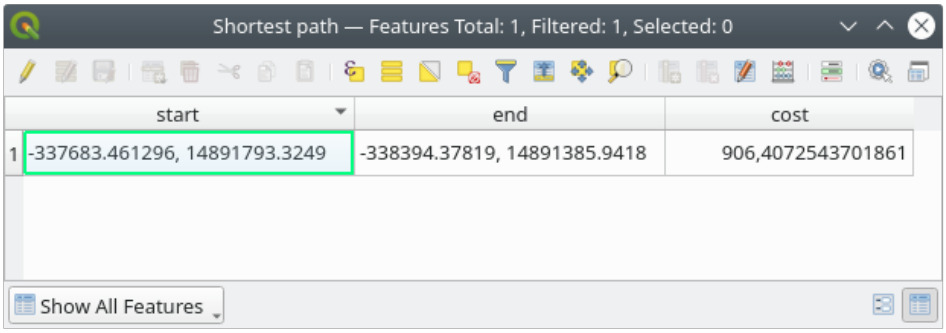
7. 選択したポイント間の最短パスを表す新しいラインレイヤが作成されます。 `network_lines` レイヤのチェックを外して、結果をよりよく確認します。



8. 出力レイヤの属性テーブルを開きます。始点と終点の座標と **コスト** を表す 3 つのフィールドが含まれています。

   最短 を **計算するパスの種類** として選択したので、 **コスト** は 2 つの位置間の **距離** をレイヤ単位で表します。

   この場合、選択された点の間の **最短** の距離は約 1000 メートルです。



ツールの使い方がわかったので、他の場所も自由に試してください。

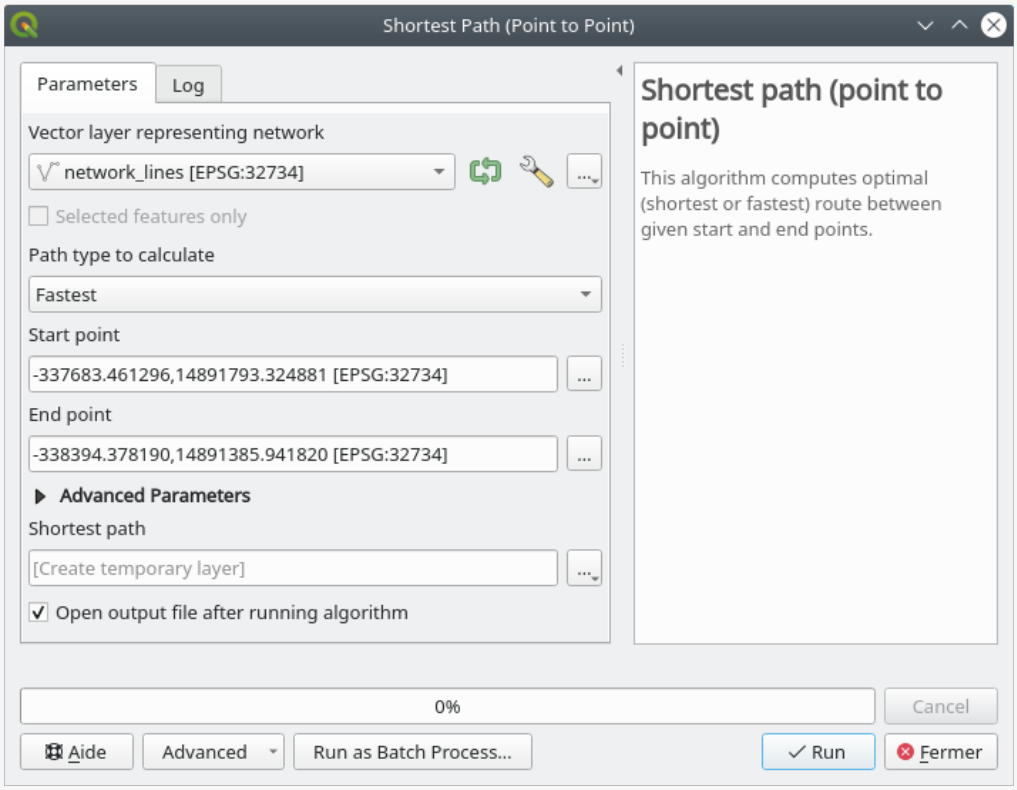
6.3.3 ★★☆ （中級レベル） 自分でやってみよう: 最速経路

前の演習と同じデータを使用して、2 点間の最速経路を計算してみます。

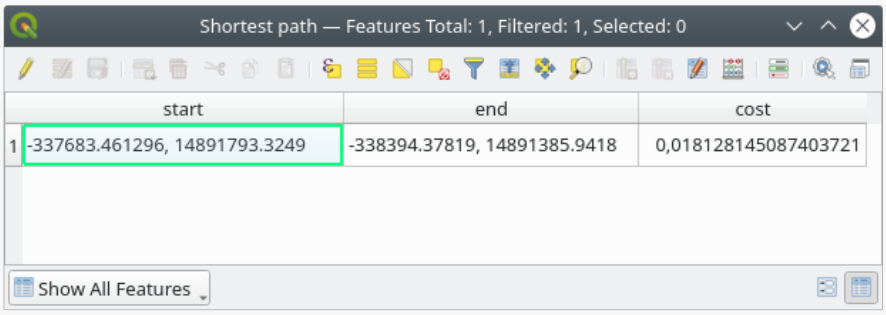
始点から終点までどのくらいの時間が必要ですか？

答え

1. **ネットワーク解析** **最短経路（指定始点から指定終点）** を開き、ダイアログを以下のように埋めます:



2. **計算するパスの種類** が 最速 であることを確認します。
3. **実行** をクリックしてダイアログを閉じます。
4. 出力レイヤの属性テーブルを開きましょう。 *cost* フィールドには、2 点間の移動時間が（時間単位で）表示されます。

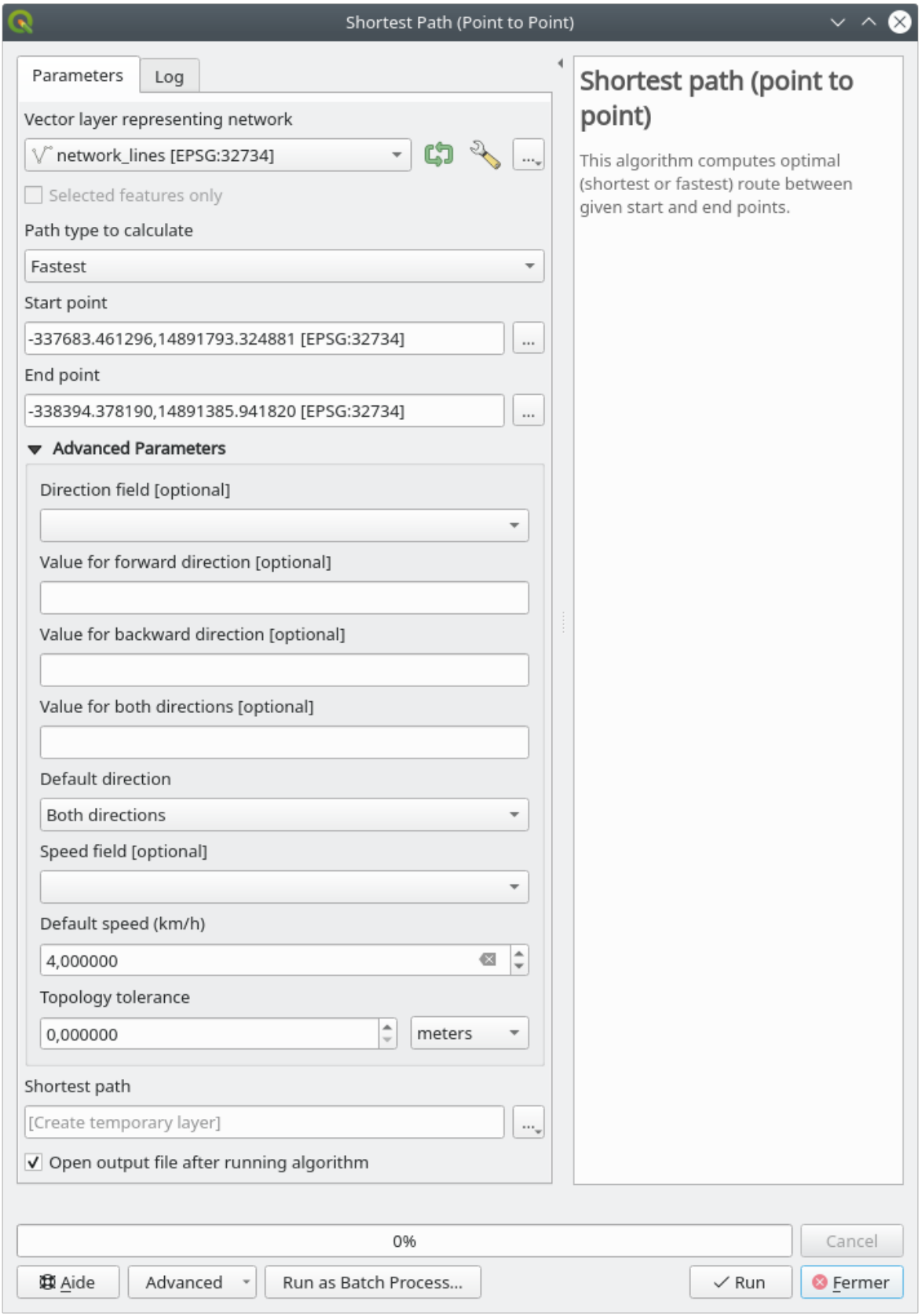


6.3.4 ★★☆ （中級レベル） 理解しよう: 高度なオプション

ネットワーク解析ツールのいくつかのオプションを調べてみましょう。 **前の演習** では、2 点間の **最速** のルートを計算しました。ご想像のとおり、時間は移動 **速度**　によって異なります。

これまでの演習と同じレイヤと開始点、終了点を使用します。

1. **最短経路（指定始点から指定終点）** アルゴリズムを開きます
2. **入力レイヤ**、 **開始点** *(x, y)* 、 **終了点** *(x, y)* を先程と同じように埋めます
3. **計算するパスの種類** `として ``最速 ` を選択します
4. **詳細パラメータ** メニューを開きます
5. **デフォルトの速度** *(km/h)* をデフォルトの `50` から `4` に変更します

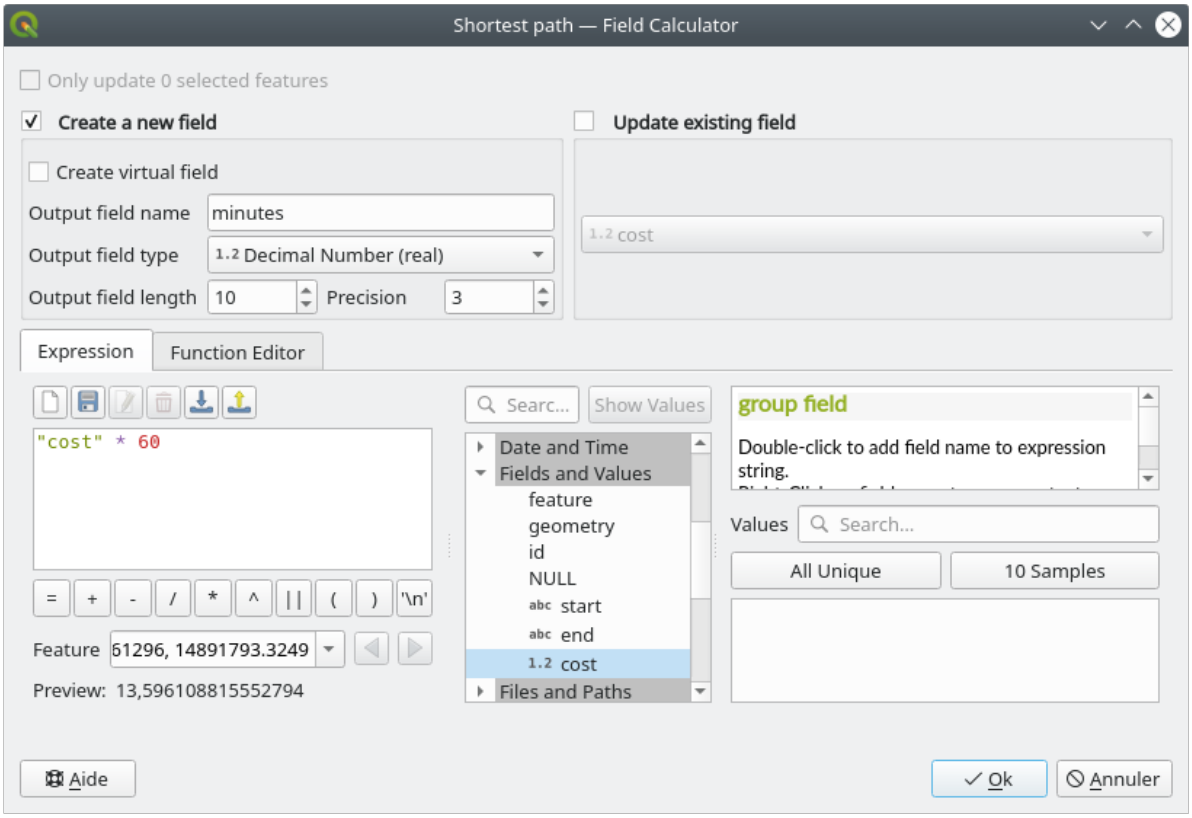


6. **実行** をクリックします

7. アルゴリズムが終了したら、ダイアログを閉じて、出力レイヤの属性テーブルを開きます。

cost フィールドには、選択した速度パラメータに応じた値が含まれます。 *cost* フィールドを、端数のある時間から、より読みやすい **分** の値に変換できます。

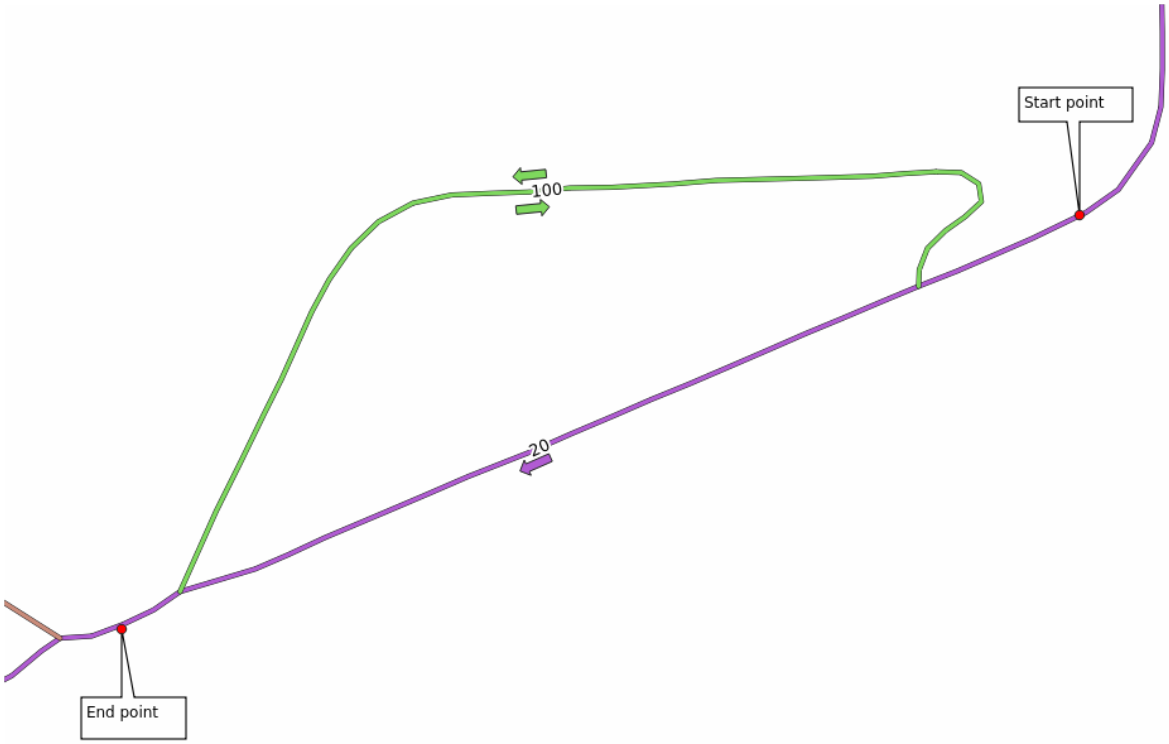
8.  アイコンをクリックしてフィールド計算機を開きます
9. 経路のコストを格納するために新しいフィールド *minutes* を追加します。



これだけです！これで、ネットワーク全体の速度が時速 4 キロの場合に、ある地点から別の地点まで何分かかるかがわかります。

6.3.5 ★★★ （上級レベル） 速度制限のある最短パス

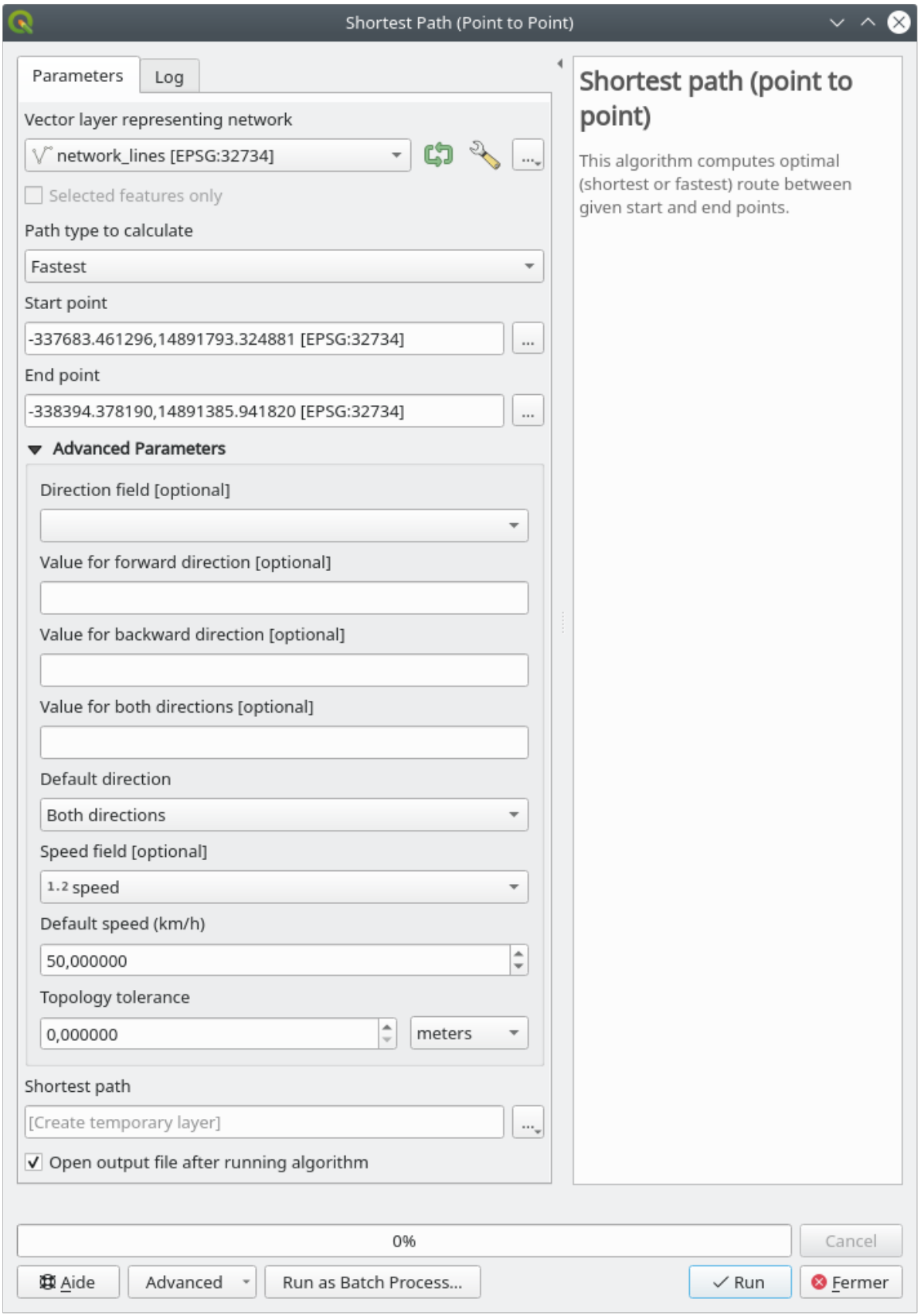
ネットワーク解析ツールボックスには他にも興味深いオプションがあります。次の地図を見てください：



各道路の制限速度を考慮した **最速** ルートを知りたい（ラベルは **制限速度** を km / h で表しています）。制限速度を考慮しない最短経路はもちろん紫色の経路になります。しかし、その道路では制限速度 20 km / h ですが、緑色の道路では 100 km / h 出せます！

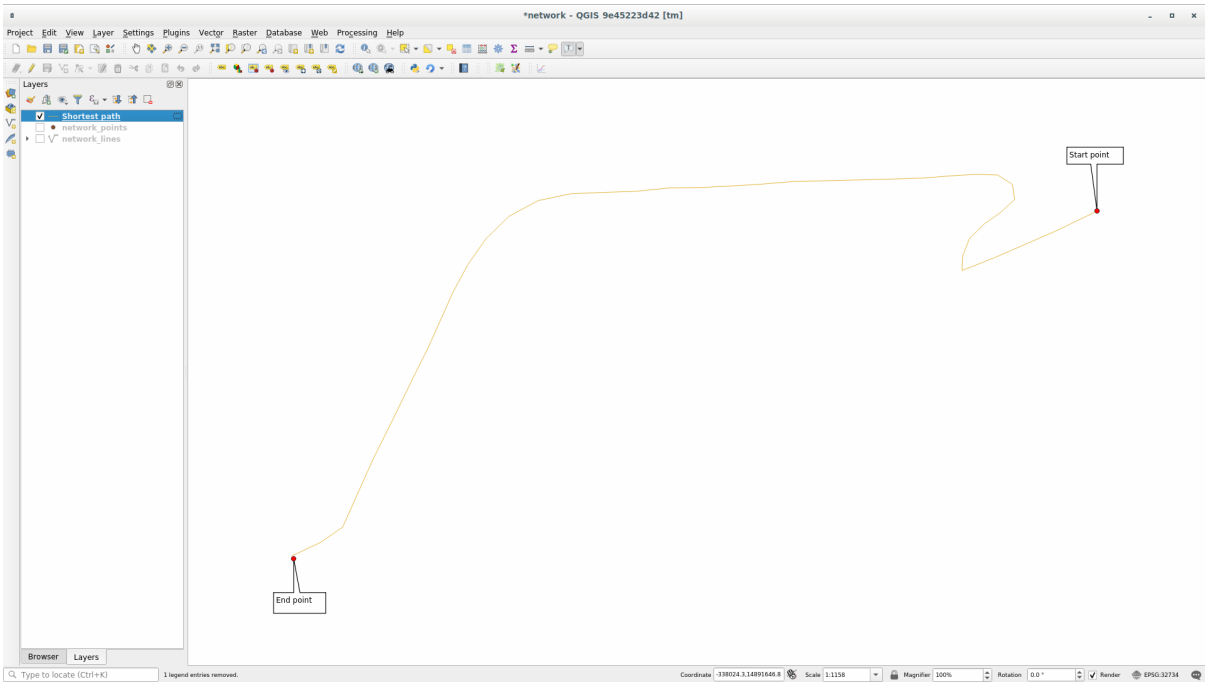
最初の練習でやったように、**ネットワーク解析** **最短経路（ポイント間）** を使用して、手動で始点と終点を選びます。

1. **ネットワーク解析** **最短経路（指定始点から指定終点）** アルゴリズムを開く
2. Select `network_lines` for the *Vector layer representing network* parameter
3. **計算するパスの種類** **として** **最速** を選択します
4. 先ほどと同じように、**開始点** *(x, y)* と **終了点** *(x, y)* を選択します
5. **詳細パラメータ** メニューを開きます
6. **速度を示す属性（フィールド）** パラメータとして *speed* フィールドを選択します。このオプションを使用すると、アルゴリズムは各道路の制限速度を考慮に入れます。



7. **実行** ボタンをクリックします

8. 結果を見やすくするために、 network_lines レイヤを非表示にします



ご覧のとおり、最速ルートは最短ルートに対応していません。

6.3.6 ★★☆ （中級レベル） サービスエリア（レイヤから）

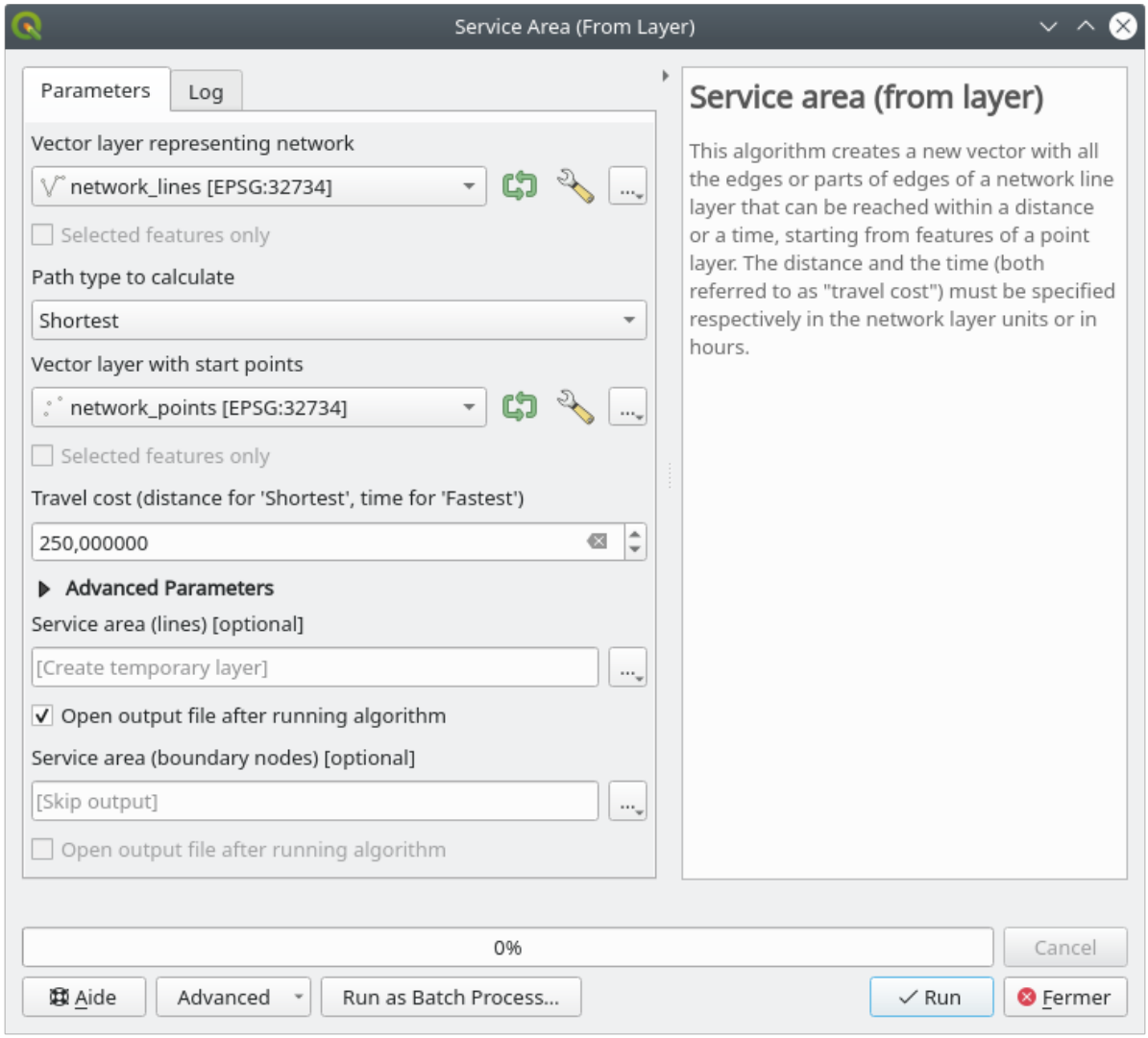
ネットワーク解析　 サービスエリア（始点レイヤ） アルゴリズムは次の質問に答えることができます：ポイントレイヤがあるとき、距離または時間の値が与えられると到達可能なすべてのエリアはどうなりますか？

> **注釈**
>
> **ネットワーク解析　 サービスエリア（始点レイヤ）** は同じアルゴリズムですが、地図上のポイントを手動で選択できます。

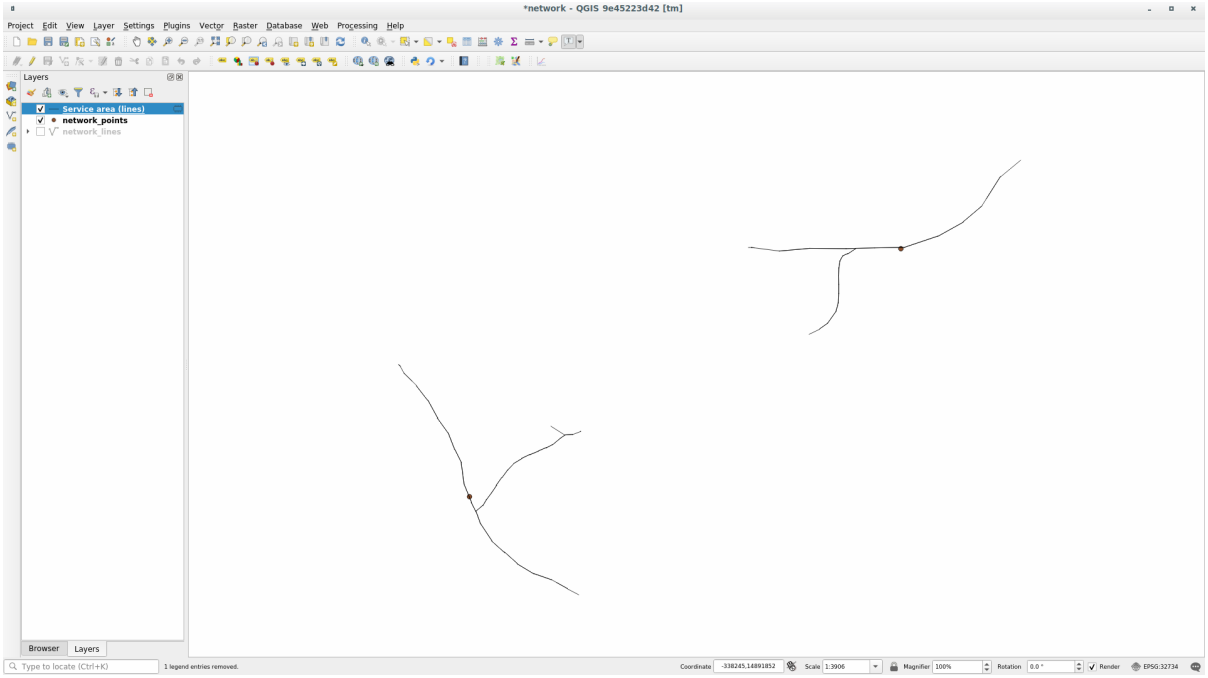
Given a distance of 250 meters we want to know how far we can go on the network from each point of the network_points layer.

1. network_points 以外のレイヤのチェックを外します
2. **ネットワーク解析　 サービスエリア（始点レイヤ）** アルゴリズムを開きます
3. **ネットワークを表すベクタレイヤ** に network_lines を選択します
4. **始点のあるベクタレイヤ** に network_points を選択します
5. **計算するパスの種類** に 最短 を選択します
6. **求めたい旅行コスト** パラメータに 250 を入力します

7. **実行** をクリックしてダイアログを閉じます



出力レイヤは、250 メートルの距離を与えられてポイント地物から到達できる最大径路を表します。



すごいですよね？

6.3.7 結論

これで、**ネットワーク解析** アルゴリズムを使用して、最短と最速経路の問題を解決する方法がわかりました。

これで、ベクタレイヤデータに対して空間統計を実行する準備が整いました。さあ行きましょう！

6.3.8 次は?

次は、ベクタデータセットに空間統計アルゴリズムを実行する方法について説明します。

6.4 レッスン: 空間統計

> **注釈**
>
> Linfiniti と S Motala（ケープ半島工科大学）が開発したレッスン

空間統計を使用すると、特定のベクトルデータセットで何が起こっているかを分析して理解できます。 QGIS には統計分析のための多くの便利なツールが含まれています。

このレッスンの目標： プロセシングツールボックス 内で QGIS の空間統計ツールを使用する方法を知ること。

6.4.1 ★☆☆ （初級レベル） 理解しよう: テストデータセットの作成

ランダムな点群を作成し、作業用のデータセットを取得します。

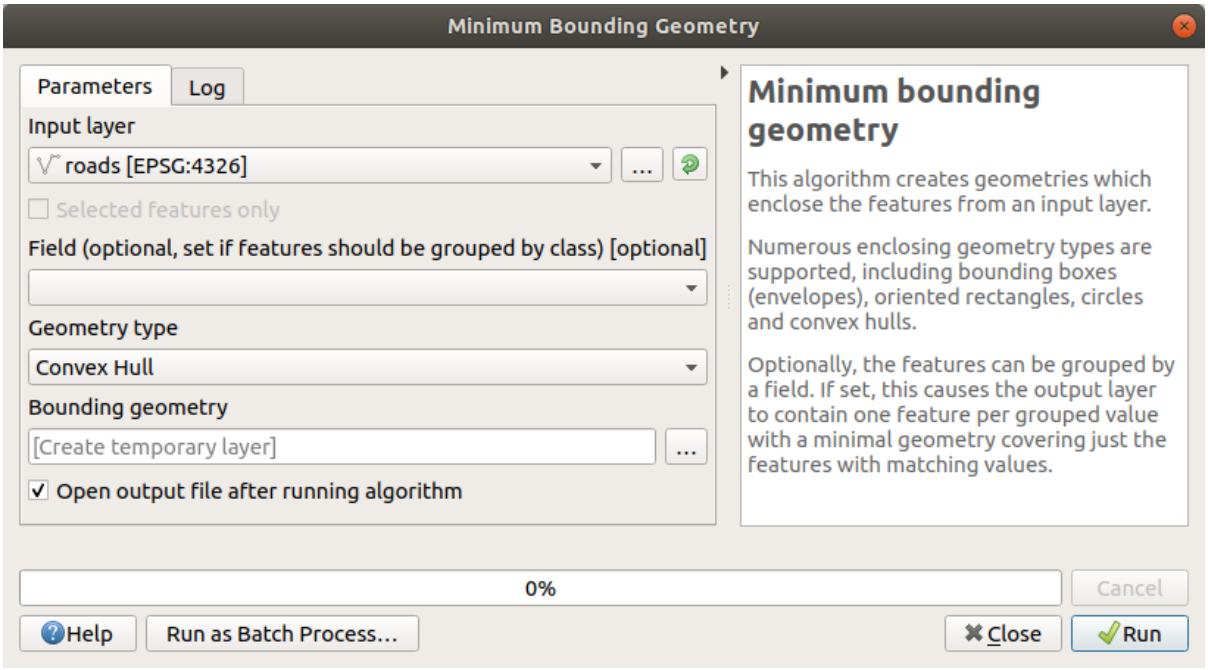
そのためには、ポイントを作成する領域を定義するポリゴンデータセットが必要になります。

街路でカバーされるエリアを使用します。

1. 新しいプロジェクトを開始します
2. `roads` データセットと、 `exercise_data/raster/SRTM/` にある `srtm_41_19`（標高データ） を追加します。

> **注釈**
>
> SRTM DEM レイヤの CRS は、道路レイヤの CRS とは異なる場合があります。 QGIS は両方のレイヤを単一の CRS で再投影しています。次の演習では、この違いは重要ではありませんが、自由に再投影してください（このモジュールの前半で示したように）。

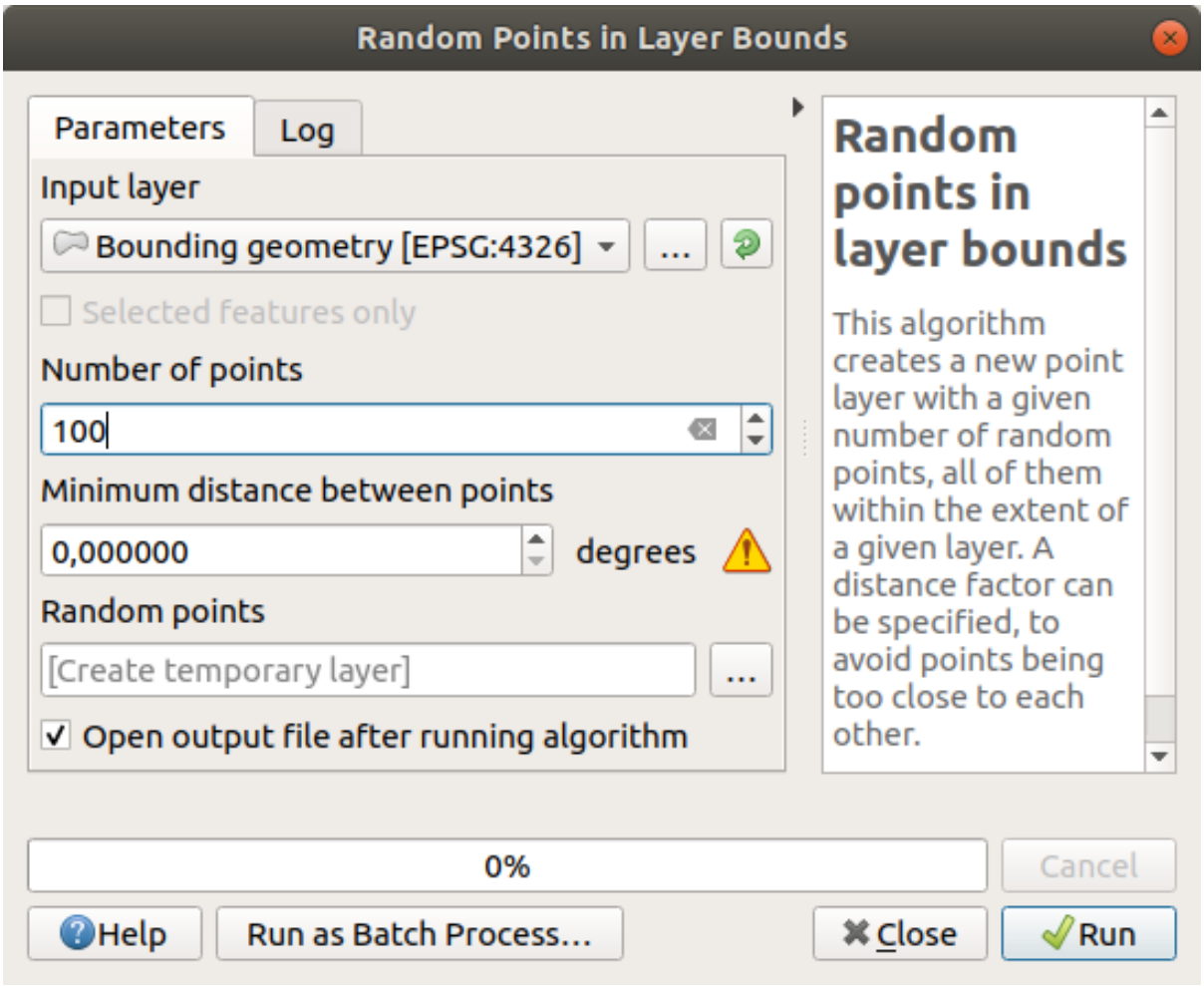
3. **プロセシング** ツールボックスを開きます
4. **ベクタジオメトリ　　最小境界ジオメトリ** ツールを使用して **ジオメトリタイプ** として 凸包 を選択し、すべての道路を囲むエリアを生成します:



ご存知のように、出力を指定しない場合、 **プロセシング** は一時的なレイヤを作成します。レイヤをすぐに、または後の段階で保存するのはあなた次第です。

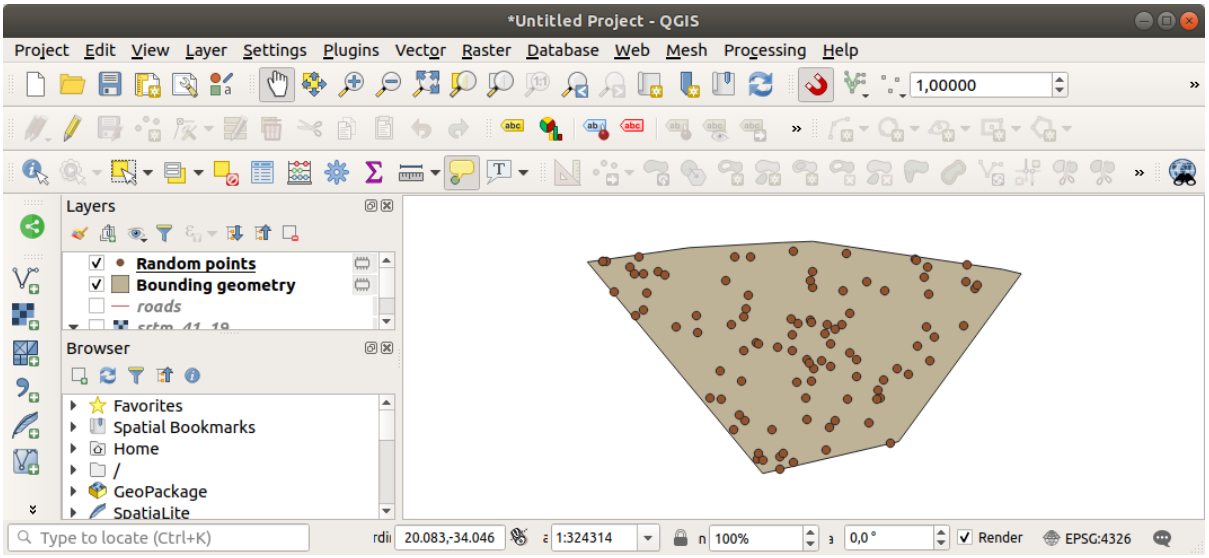
ランダム点群の作成

- **ベクタ作成　　レイヤ領域にランダム点群** にあるツールを使って、この領域に最小距離 `0.0` のランダムな点を 100 個作成します:



> **注釈**
>
> 黄色の警告サインは、そのパラメータが距離に関係していることを示しています。アルゴリズムは *Bounding geometry* レイヤが地理座標系にあることを思い出させているだけです。この例では、このパラメータを使用しないため、無視して構いません。

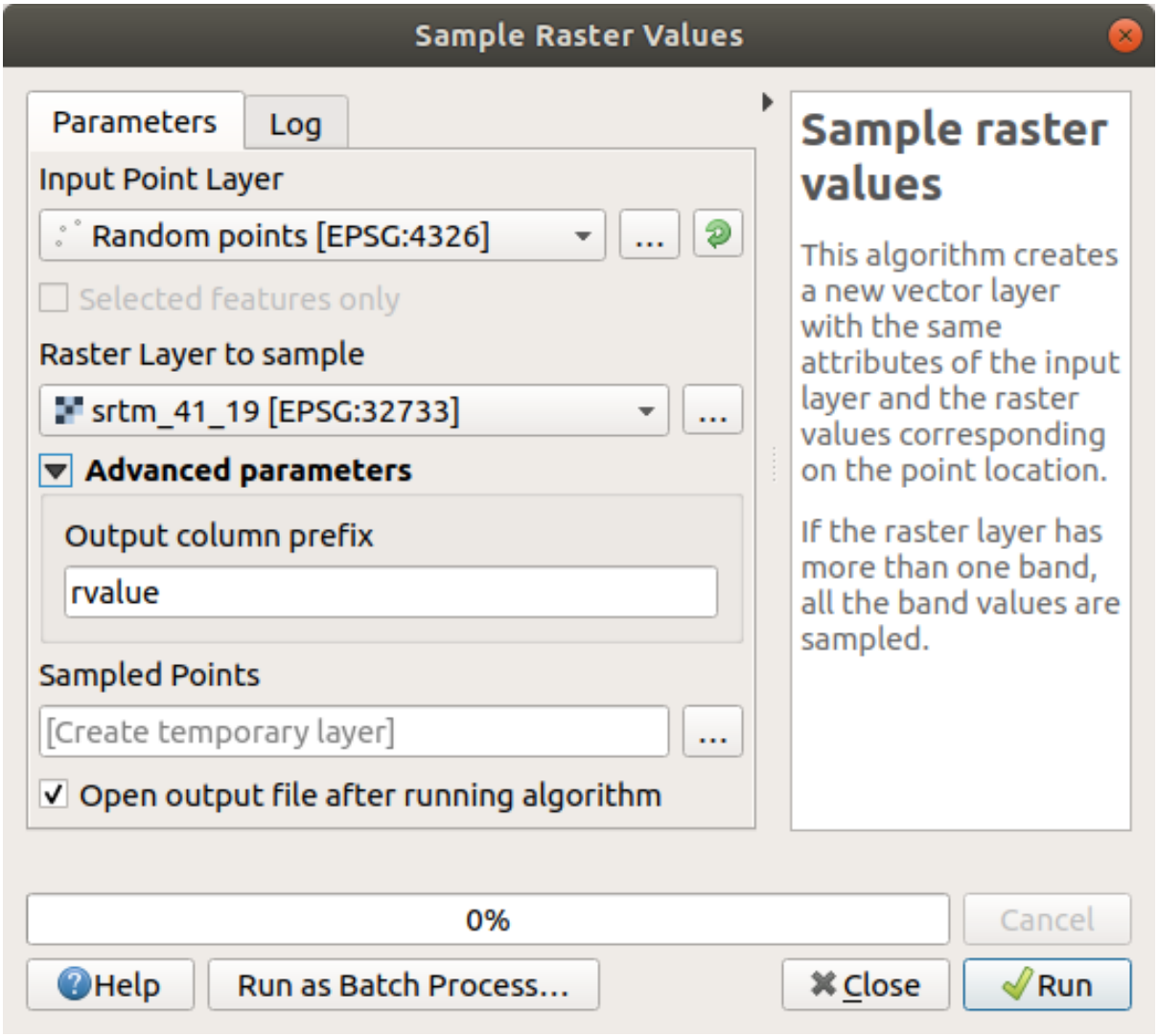
必要に応じて、生成されたランダム点を凡例の一番上に移動させると見やすくなります:



データのサンプリング

ラスタからサンプルデータセットを作成するには、 **ラスタ解析** **ベクタレイヤにラスタ値を付加** アルゴリズムを使用する必要があります。このツールは、ポイントの位置でラスタをサンプリングし、ラスタ内のバンドの数に応じて、新しいフィールドにラスタ値を追加します。

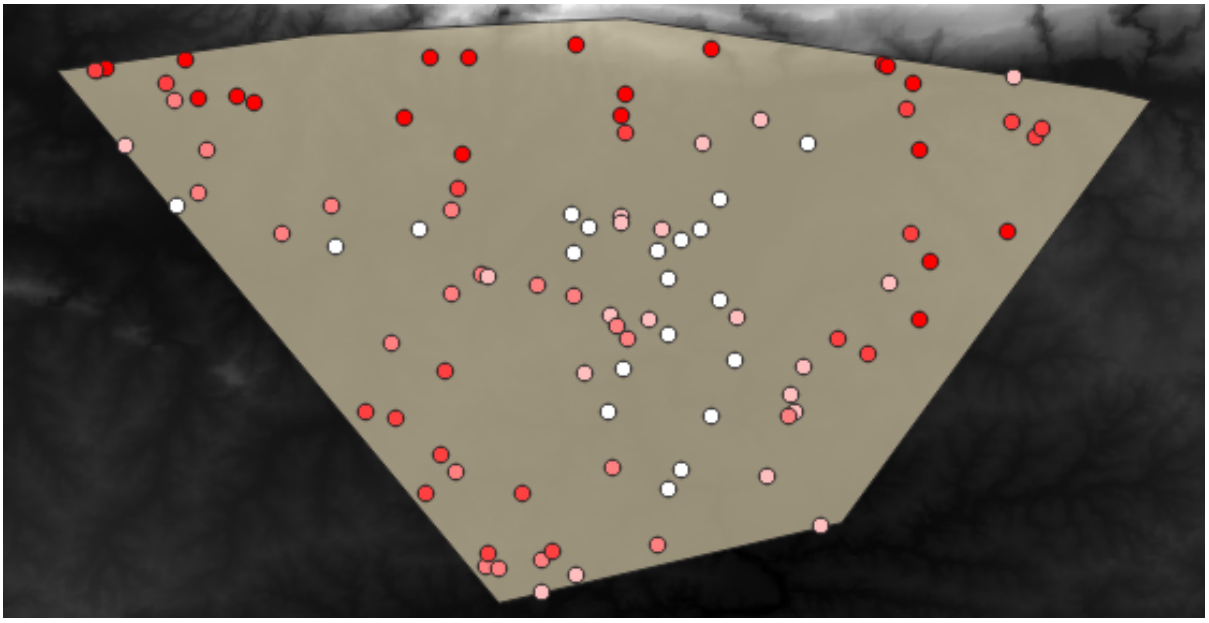
1. **ベクタレイヤにラスタ値を付加** アルゴリズムダイアログを開きます
2. サンプリングポイントを含むレイヤとして `Random_points` を選択し、値を取得するバンドとして SRTM ラスタを選択します。新しいフィールドのデフォルト名は `rvalue_N` です。ここで、 `N` はラスタバンドの番号です。必要に応じて、プレフィックスの名前を変更できます。



3. **実行** を押します

これで、ラスタファイルからサンプリングされたデータを サンプリングした出力 レイヤの属性テーブルで確認することができます。それらは、選択した名前の新しいフィールドに入ります。

サンプルレイヤはここに示すとおりです:



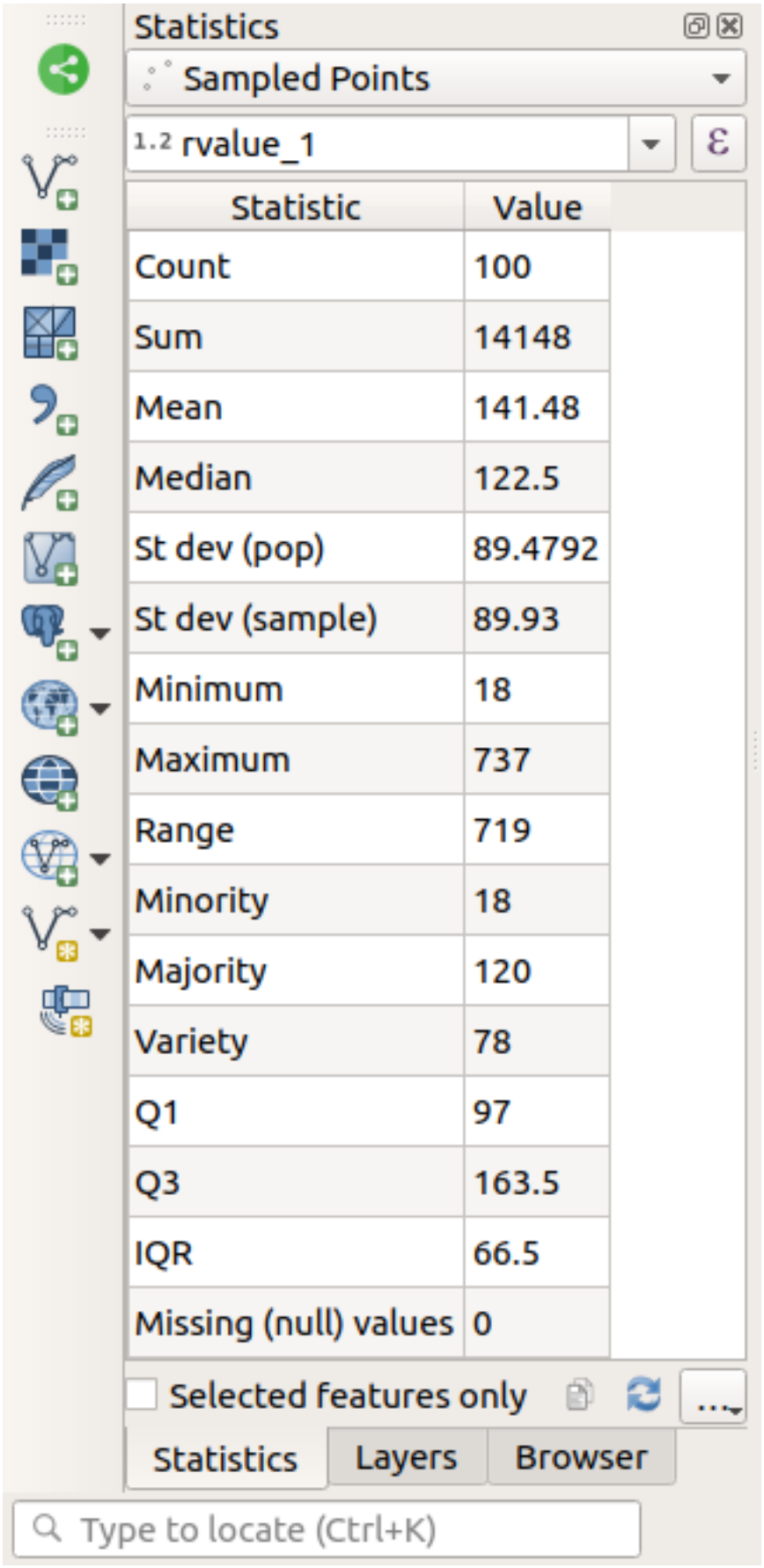
サンプルポイントは、赤いポイントがより高い高度になるように、「rvalue_1」フィールドを使用して分類されます。

このサンプルレイヤは、残りの統計演習で使用します。

6.4.2 ★☆☆ （初級レベル） 理解しよう: 基本統計

さて、このレイヤに対して基本統計を取得しましょう。

1. **属性ツールバー** にある  統計の要約を表示 アイコンをクリックします。新しいパネルがポップアップします。
2. 表示されたダイアログで、サンプリングした出力 レイヤをソースとして指定します。
3. フィールドコンボボックスで *rvalue_1* フィールドを選択します。このフィールドは、統計情報を計算するフィールドです。
4. 算出された統計情報で **統計量の出力** パネルが自動的に更新されます:


Statistics
Sampled Points
1.2 rvalue_1
Statistic
Value
Count
100
Sum
14148
Mean
141.48
Median
122.5
St dev (pop)
89.4792
St dev (sample)
89.93
Minimum
18
Maximum
737
Range
719
Minority
18
Majority
120
Variety
78
Q1
97
Q3
163.5
IQR
66.5
Missing (null) values
0
Selected features only
Statistics
Layers
Browser
Type to locate (Ctrl+K)

> **注釈**
>
> 統計をクリップボードにコピー ボタンをクリックして値をコピーし、結果をスプレッドシートに貼り付けることができます。

5. 終了したら **統計量の出力** パネルを閉じます

さまざまな統計が用意されています:

個数
サンプル/値の個数

Sum
値を足したもの。

平均値（Mean）
平均（アベレージ）値は、単純に値の合計を値の個数で割ったものです。

中間値
すべての値を最小から最大に並べると、中央値（または、N が偶数の場合は 2 つの中央値の平均）が値の中央値になります。

標準偏差（母集団）
標準偏差。値が平均値の周りのどの程度近くに密集しているかの指標を与えます。標準偏差が小さいほど、値が平均値により近づく傾向があります。

最小値
値の最小値です。

最大
値の最大値です。

範囲（Range）
最小および最大値間の差です。

第１四分位
データの第１四分位。

第３四分位
データの第３四分位

欠落 (NULL) 値
欠落値の個数。

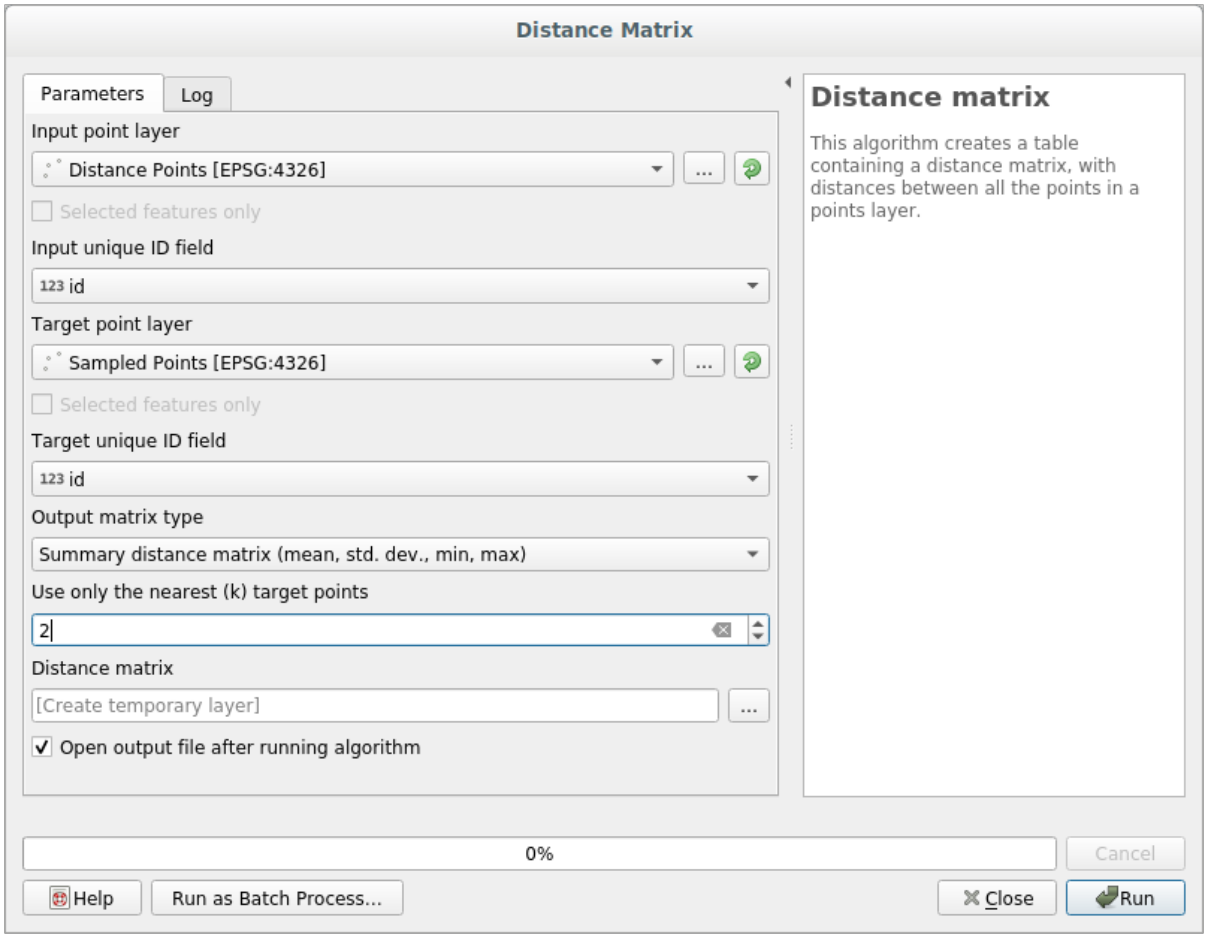
6.4.3 ★☆☆ （初級レベル） 理解しよう: 点間の距離についての統計を計算

1. 新しい一時的な点レイヤを作成します。
2. 編集モードに入り、他の点のどこかにある 3 点をデジタイズします。

   または、先ほどと同じランダム点群生成方法で、**3** 点だけ指定します。
3. Save your new layer as `distance_points` in the format you prefer.

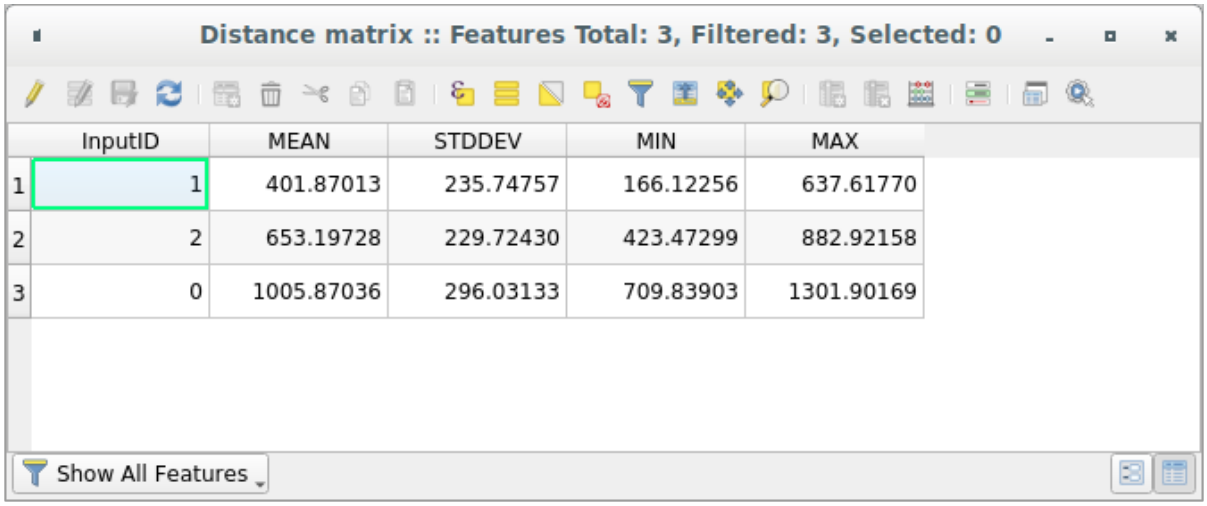
2 つのレイヤのポイント間の距離の統計を作成するには:

1. **ベクタ解析**　**距離行列** ツールを開きます。
2. 入力レイヤとして `distance_points` を、ターゲットレイヤとして `Sampled Points` を選択します。
3. `id` フィールドを一意なフィールド参照として設定します
4. **出力形式** オプションを **距離統計行列** に変更します。
5. **使用する近傍点の個数** の値を 2 に設定します。
6. 必要に応じて、出力レイヤをファイルとして保存するか、アルゴリズムを実行して後で一時出力レイヤを保存することができます。



7. **実行** をクリックすると、距離行列レイヤが生成されます。

8. Open the attribute table of the generated layer: values refer to the distances between the `distance_points` features and their two nearest points in the `Sampled Points` layer:



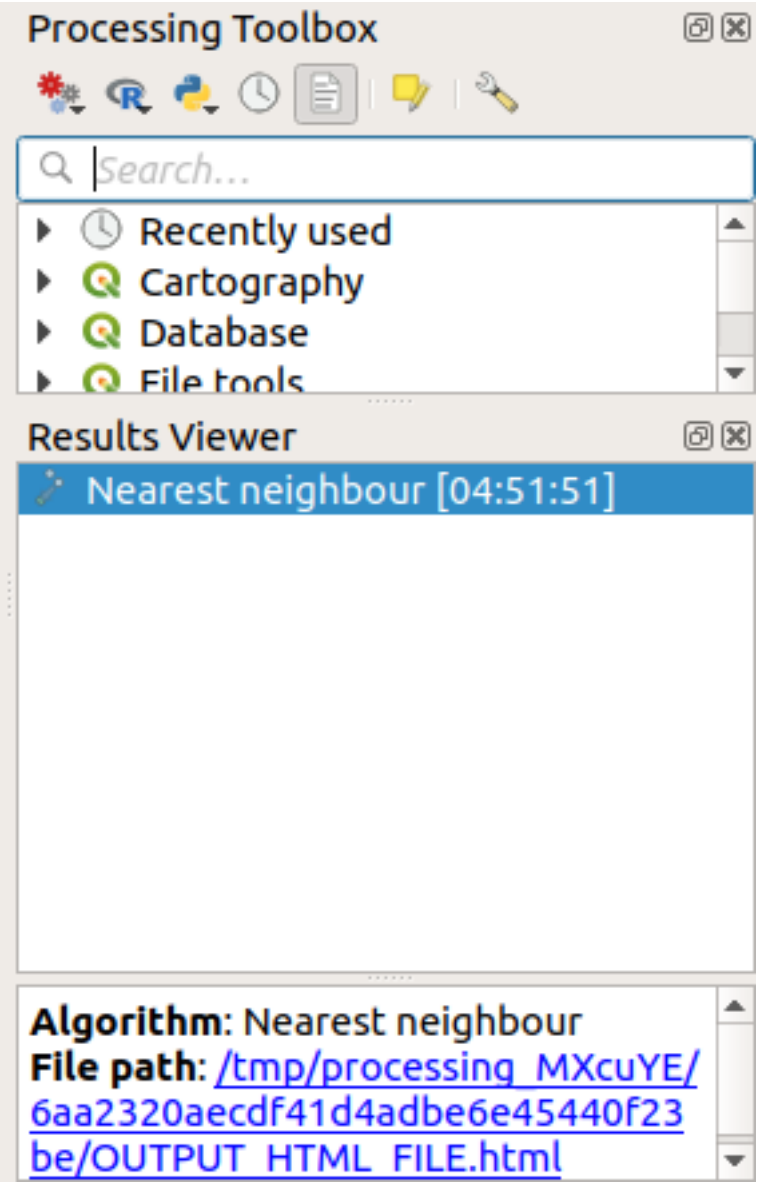
これらのパラメータにより、**距離行列** ツールは、入力レイヤの各ポイントについて、ターゲットレイヤの 2 つの最も近いポイントとの距離の統計を計算します。出力レイヤのフィールドには、計算された距離の平均、標準偏差、最小値、最大値が格納されます。

さらにテストを行う場合は、**出力形式** オプションまたはターゲット点の数を変更するとよいでしょう。

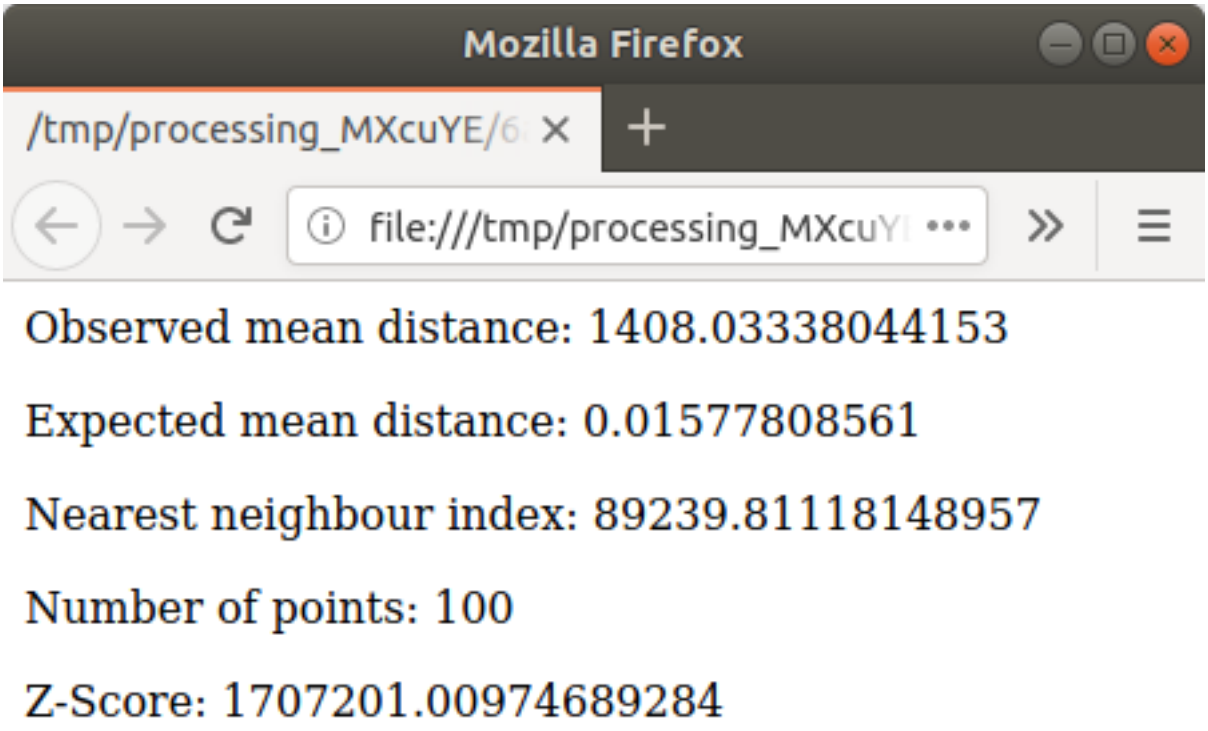
6.4.4 ★☆☆ （初級レベル） 理解しよう: 最近傍解析（レイヤ内）

ポイントレイヤの再近傍解析をするには:

1. **ベクタ解析** **最近傍解析** を選びます。
2. In the dialog that appears, select the `Random points` layer and click *Run*.
3. 結果は、プロセシング **結果ビューア** パネルに表示されます。



4. 青いリンクをクリックすると、結果の `html` ページが表示されます:



6.4.5 ★☆☆ （初級レベル） 理解しよう: 加重平均座標

データセットの平均座標を取得するために:

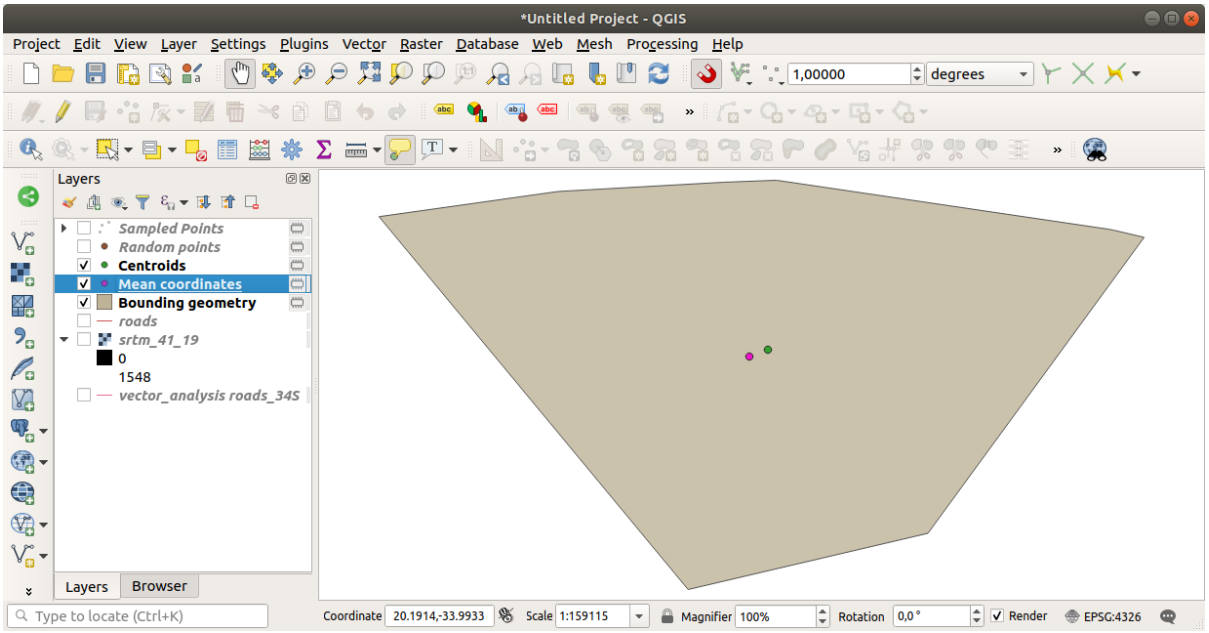
1. **ベクタ解析　 加重平均座標（重心の平均）**
2. In the dialog that appears, specify `Random points` as *Input layer*, and leave the optional choices unchanged.
3. **実行** をクリックします。

ランダムサンプルを作成する際に使用したポリゴンの中心座標と比較してみましょう。

1. **ベクタジオメトリ　 重心**
2. 表示されたダイアログで、入力レイヤとして `Bounding geometry` を選択します。

ご覧のように、平均座標（ピンクの点）と調査地域の中心（緑の点）は必ずしも一致していません。

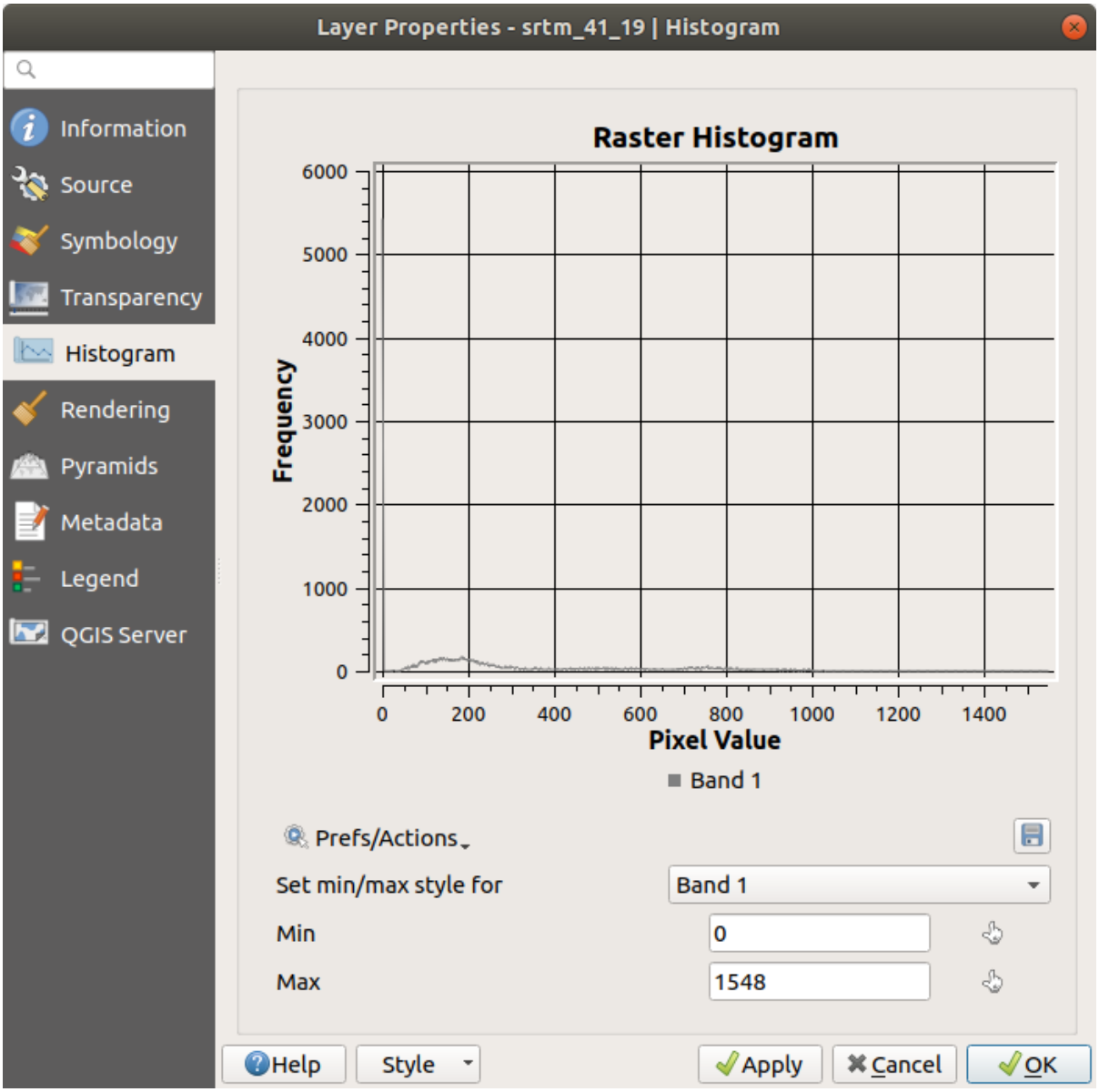
図心はレイヤの重心（正方形の重心は正方形の中心）であり、平均座標はすべてのノード座標の平均を表します。



6.4.6 ★☆☆ （初級レベル） 理解しよう: 画像ヒストグラム

データセットのヒストグラムは、その値の分布を示しています。 QGIS でこれを示す最も簡単な方法は、画像ヒストグラムを使用することです。これは、任意の画像レイヤ（ラスタデータセット）の **レイヤプロパティ** ダイアログで利用できます。

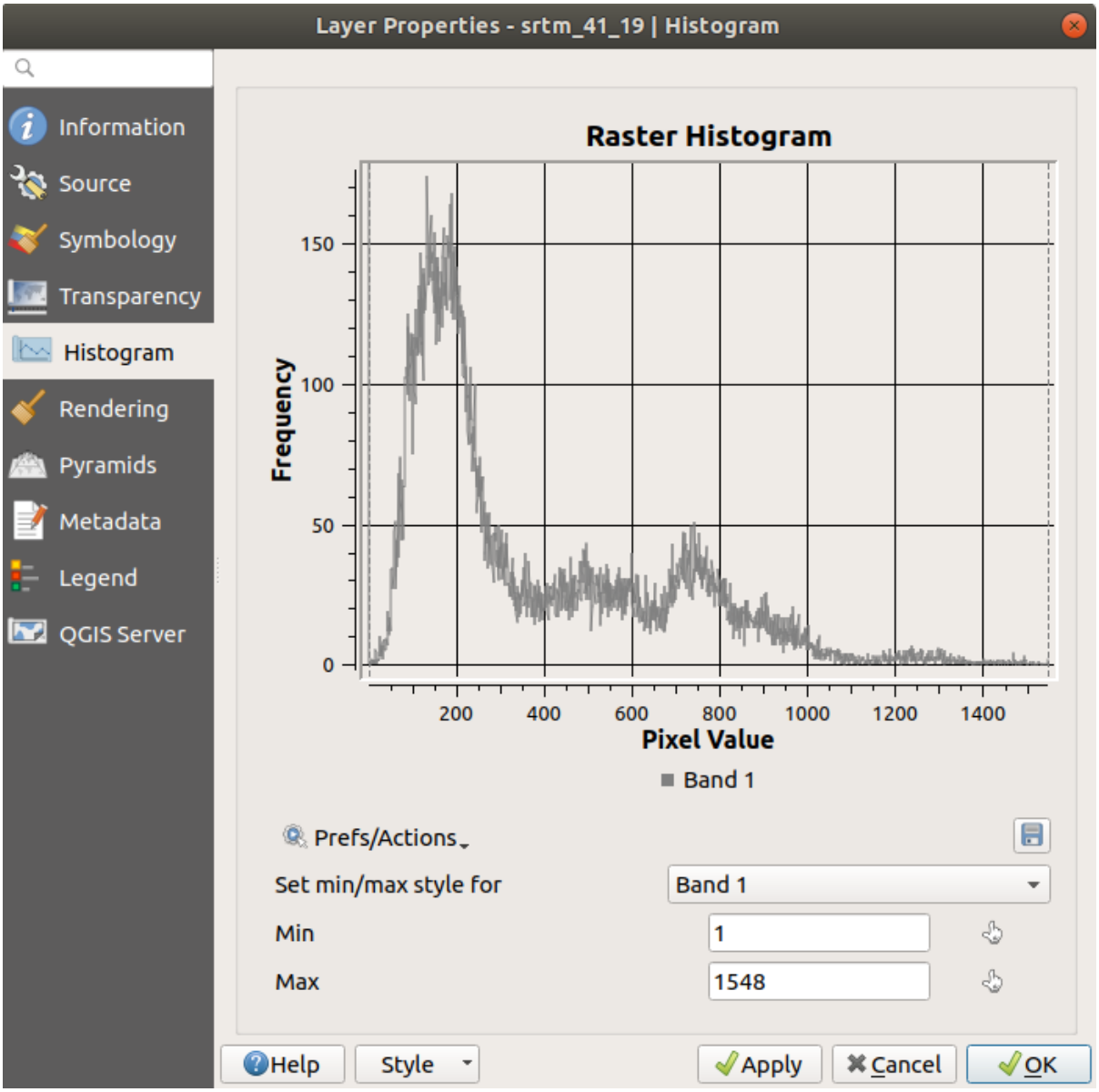
1. **レイヤ** パネルで `srtm_41_19` レイヤを右クリックします
2. **プロパティ** を選びます
3. **ヒストグラム** タブを選択します。グラフィックを生成するには、 **ヒストグラムの計算** ボタンをクリックする必要がある場合があります。ラスタ値の度数分布を示すグラフが表示されます。



4. グラフは  プロットの保存 ボタンで画像としてエクスポートすることができます

5. レイヤに関するより詳細な情報は、 **情報** タブで見ることができます（平均値と最大値は推定値であり、正確でない可能性があります）。

平均値は「332.8」（推定 324.3）、最大値は「1699」（推定 1548）！ヒストグラムを拡大できます。値が「0」のピクセルがたくさんあるため、ヒストグラムは垂直方向に圧縮されたように見えます。ズームインして「0」のピーク以外のすべてをカバーすると、詳細が表示されます：



注釈

平均値と最大値が上記と同じでない場合は、最小値/最大値の計算が原因である可能性があります。 **シンボロジ** タブを開き、 **最小 / 最大値設定** メニューを展開します。 **最小/最大** を選択し、 **適用** をクリックします。

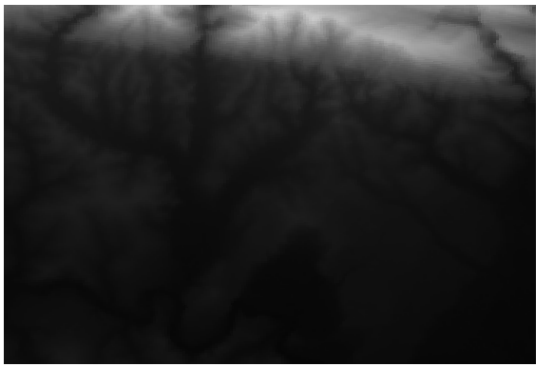
ヒストグラムは、値の分布を示すものであり、必ずしもすべての値がグラフ上に表示されるわけではないことに留意してください。

6.4.7 ★☆☆ （初級レベル） 理解しよう: 空間補間

Let's say you have a collection of sample points from which you would like to extrapolate data. For example, you might have access to the `Sampled points` dataset we created earlier, and would like to have some idea of what the terrain looks like.

1. まず、**プロセシングツールボックス** にある *GDAL* **ラスタ解析** **グリッド（最近傍探索 *IDW*）** ツールを起動します。
2. **入力レイヤ（点）** に `Sampled points` を選びます
3. **重み付けの累乗** を `5.0` に設定します
4. **詳細パラメータ** で、 **内挿する *Z* 値の属性（フィールド）** を `rvalue_1` に設定します
5. 最後に **実行** をクリックし、処理が終了するまで待ちます
6. ダイアログを閉じます

これは、元のデータセット（左）とサンプルポイントから構築されたデータセット（右）の比較です。サンプルポイントの位置がランダムであるため、見た目が異なる場合があります。





ご覧のとおり、地形の詳細な印象を得るには、100 個のサンプルポイントでは十分ではありません。これは非常に一般的な考え方ですが、誤解を招く可能性もあります。

6.4.8 ★★☆ （中級レベル） 自分でやってみよう: 補間方法の違い

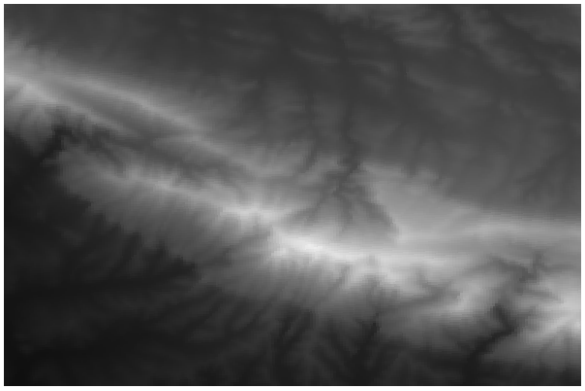
1. 上記の手順を使用して、10000 個のランダムな点のセットを作成します

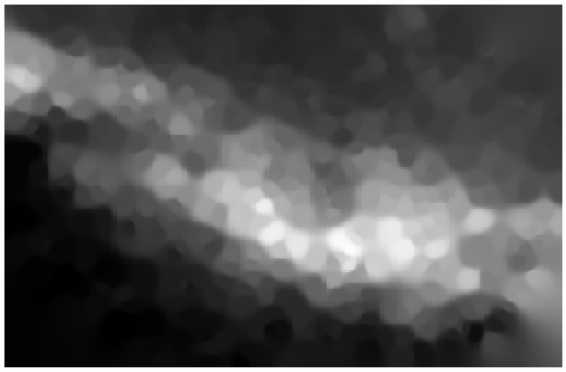
   注釈

   点の個数が多い場合、処理に時間がかかることがあります。

2. これらの点を利用して、オリジナルの DEM をサンプリングします
3. このデータセットには *Grid (IDW with nearest neighbor searching)* ツールを使用します。
4. **累乗** と **スムージング** をそれぞれ `5.0` と `2.0` に設定します。

結果（ランダムな点の位置に応じて）多かれ少なかれ、このようになります。





これは、サンプルポイントの密度が高いため、地形をより適切に表現できます。サンプルが大きいほど良い結果が得られることを忘れないでください。

6.4.9 結論

QGIS には、データセットの空間統計プロパティを分析するためのツールがいくつかあります。

6.4.10 次は?

ベクタ解析について説明しましたが、次はラスタで何ができるかを見てみましょう。これが次のモジュールで行うことです！

第7章　モジュール: ラスタ

以前デジタイズするためにラスタを使用しましたが、ラスタデータを直接使うこともできます。このモジュールでは、それを QGIS でどのように行うのかがわかります。

7.1 レッスン: ラスタデータの操作

ラスタデータは、ベクタデータとは全く異なります。ベクタデータは、線や領域で結ばれていることもある、頂点で構成されるジオメトリを伴う別々の地物を持ちます。しかし、ラスタデータは画像と同じです。実世界に存在するさまざまな物体の性質を表現していますが、それらは独立した物体として存在しているわけではありません。異なる値を持つピクセルで表現されます。

このモジュールでは、既存の GIS 解析を補完するためにラスタデータを使用することになります。

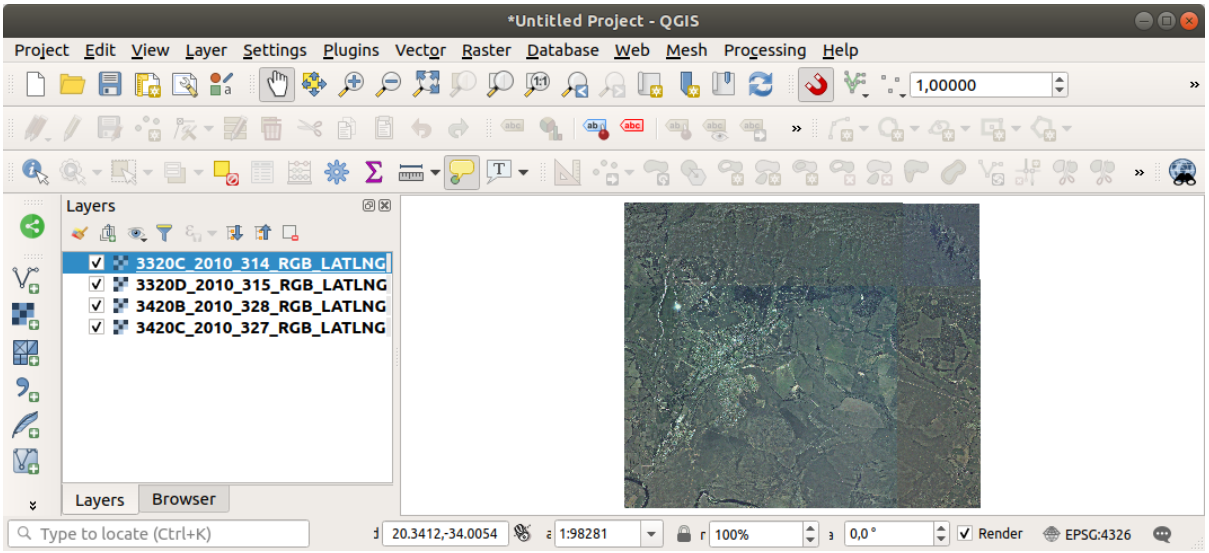
このレッスンの目標: QGIS でラスタデータを扱う方法を習得する。

7.1.1 ★☆☆ （初級レベル） 理解しよう: ラスタデータを読み込む

ラスタデータは、ベクタデータと同じ方法で読み込むことができますが、 **ブラウザ** パネルを使うことをお勧めします。

1. **ブラウザ** パネルを開いて、 `exercise_data/raster` フォルダを展開します。
2. このフォルダにあるすべてのデータを読み込みます:
    - `3320C_2010_314_RGB_LATLNG.tif`
    - `3320D_2010_315_RGB_LATLNG.tif`
    - `3420B_2010_328_RGB_LATLNG.tif`
    - `3420C_2010_327_RGB_LATLNG.tif`

以下のような地図が表示されるはずです:



以上、4 枚の航空写真で調査地域をカバーしました。

7.1.2 ★☆☆ （初級レベル） 理解しよう: 仮想ラスタの作成

このように、ソリューションレイヤは 4 つの画像すべてにまたがっています。これはつまり、常に 4 枚のラスタを扱う必要があることを意味します。これは理想的ではありません。1 つのファイルで作業できた方が良いでしょう。

幸いなことに、QGIS ではまさにこれを行うことができます。実際に新しいラスタファイルを作成することなく、 **仮想ラスタ** を作成することができます。これはしばしばその機能を説明している、 **カタログ** とも呼ばれます。これは実際の新しいラスタではありません。むしろ、既存の複数のラスタを簡単にアクセスできるようにするために 1 つのカタログ: 1 つのファイルに整理する方法です。

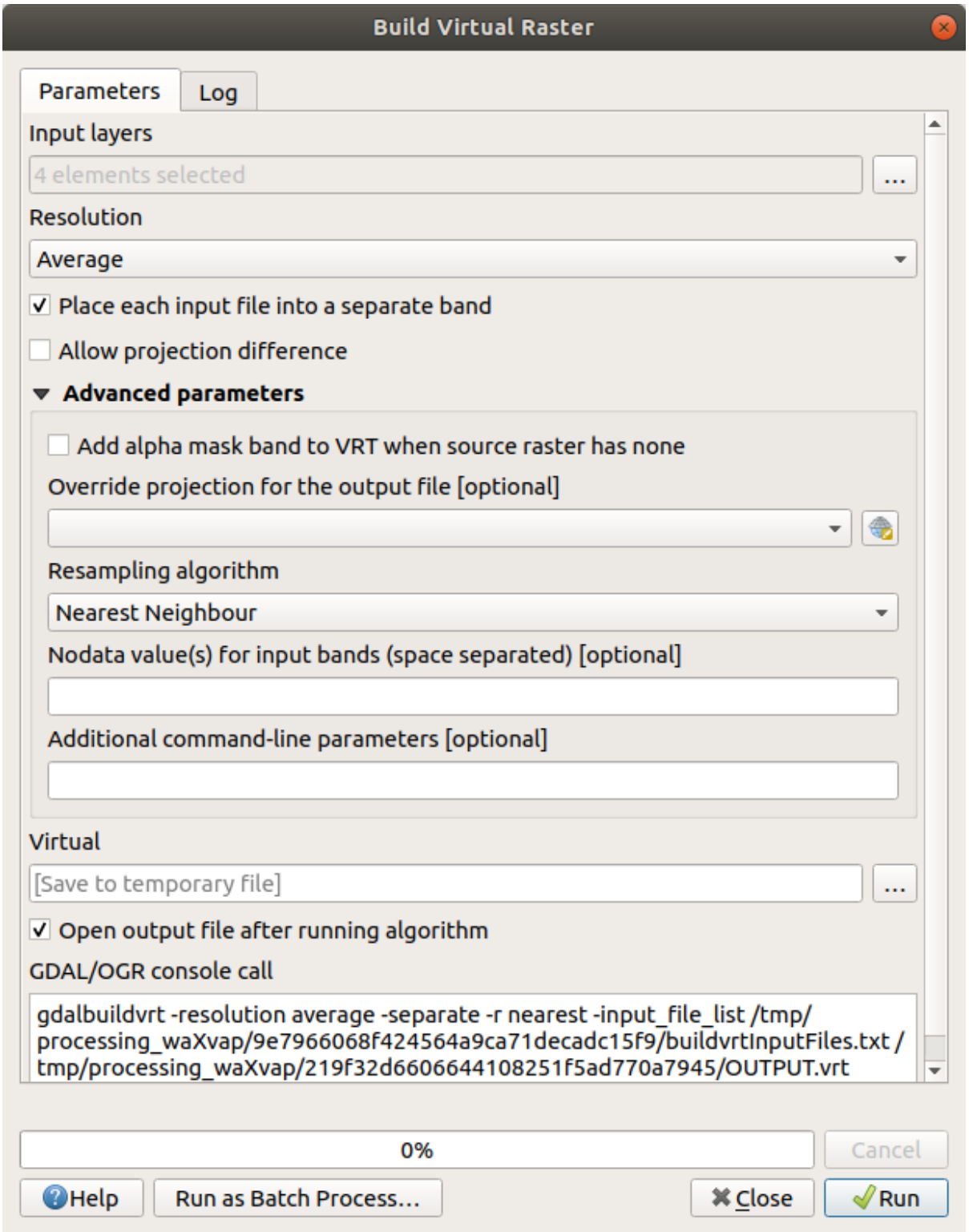
カタログを作るには、 **プロセシング** **ツールボックス** を使います。

1. *GDAL* **ラスタその他** から **仮想ラスタを構築** アルゴリズムを起動します;
2. 表示されたダイアログで、 *Input layers* パラメータの隣にある ... ボタンをクリックし、すべてのレイヤをチェックするか **すべて選択** ボタンを使用します;
3. *Place each input file into a separate band* のチェックを外します。ダイアログの下部にある *GDAL/OGR* **コンソールコール** に生成されるコードに注目してください。これは **実行** をクリックしたときに実行されるコマンドです。ダイアログのさまざまなオプションを変更すると、そのテキストが更新されます。

> **注釈**
>
> そのテキストをコピーして `OSGeo Shell` (Windows ユーザ）または `Terminal`（Linux および OSX ユーザ）に貼り付けて、コマンドを実行することができることを覚えておいてください。また、それぞれの GDAL コマンドに対してスクリプトを作成することができます。これは、手順に時間がかかっているときや、特定のタスクをスケジュールしたいときにとても便利です。コマンドのシンタックスに関するより詳しいヘルプを得るには、 **ヘルプ** ボタンを使用します。

4. 最期に **実行** をクリックします。

> **注釈**
>
> これまでのモジュールでご存知のように、**プロセシング** はデフォルトで一時レイヤを作成します。ファイルを保存するには **仮想ラスタ** の下にある ... ボタンをクリックします。



これで **レイヤ** パネルから元の 4 つのラスタを削除して、出力された仮想カタログラスタだけを残すことができます。

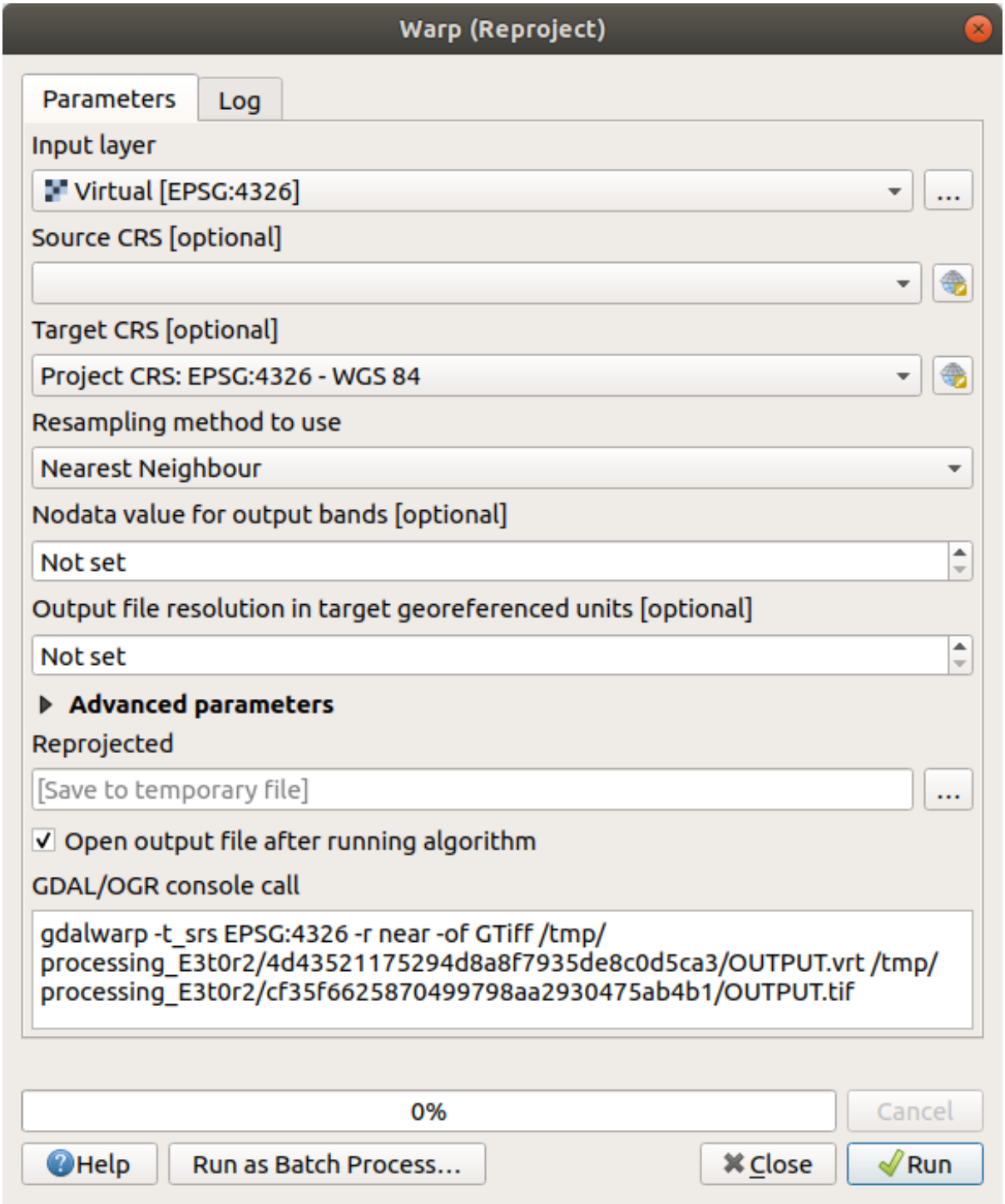
7.1.3 ★★★ （上級レベル） ラスタデータの変換

上記の方法では、カタログを使用してデータセットを仮想的にマージし、それらを「その場で」再投影できます。しかし、もし長期間使用することになるデータを用意しているならば、マージされ再投影された新しいラスタを作成する方が効率的かもしれません。はじめに用意をするのに少し時間がかかりますが、そうしておけば地図でラスタを使う際のパフォーマンスが向上します。

ラスタを再投影する

GDAL **ラスタ投影** から **再投影** *(warp)* を開いてください。

また、仮想ラスタ（カタログ）の再投影、マルチスレッド処理の有効化なども可能です。


Warp (Reproject)
Parameters
Log
Input layer
Virtual [EPSG:4326]
Source CRS [optional]
Target CRS [optional]
Project CRS: EPSG:4326 - WGS 84
Resampling method to use
Nearest Neighbour
Nodata value for output bands [optional]
Not set
Output file resolution in target georeferenced units [optional]
Not set
Advanced parameters
Reprojected
[Save to temporary file]
Open output file after running algorithm
GDAL/OGR console call
gdalwarp -t_srs EPSG:4326 -r near -of GTiff /tmp/
processing_E3t0r2/4d43521175294d8a8f7935de8c0d5ca3/OUTPUT.vrt /tmp/
processing_E3t0r2/cf35f6625870499798aa2930475ab4b1/OUTPUT.tif
0%
Cancel
Help
Run as Batch Process...
Close
Run

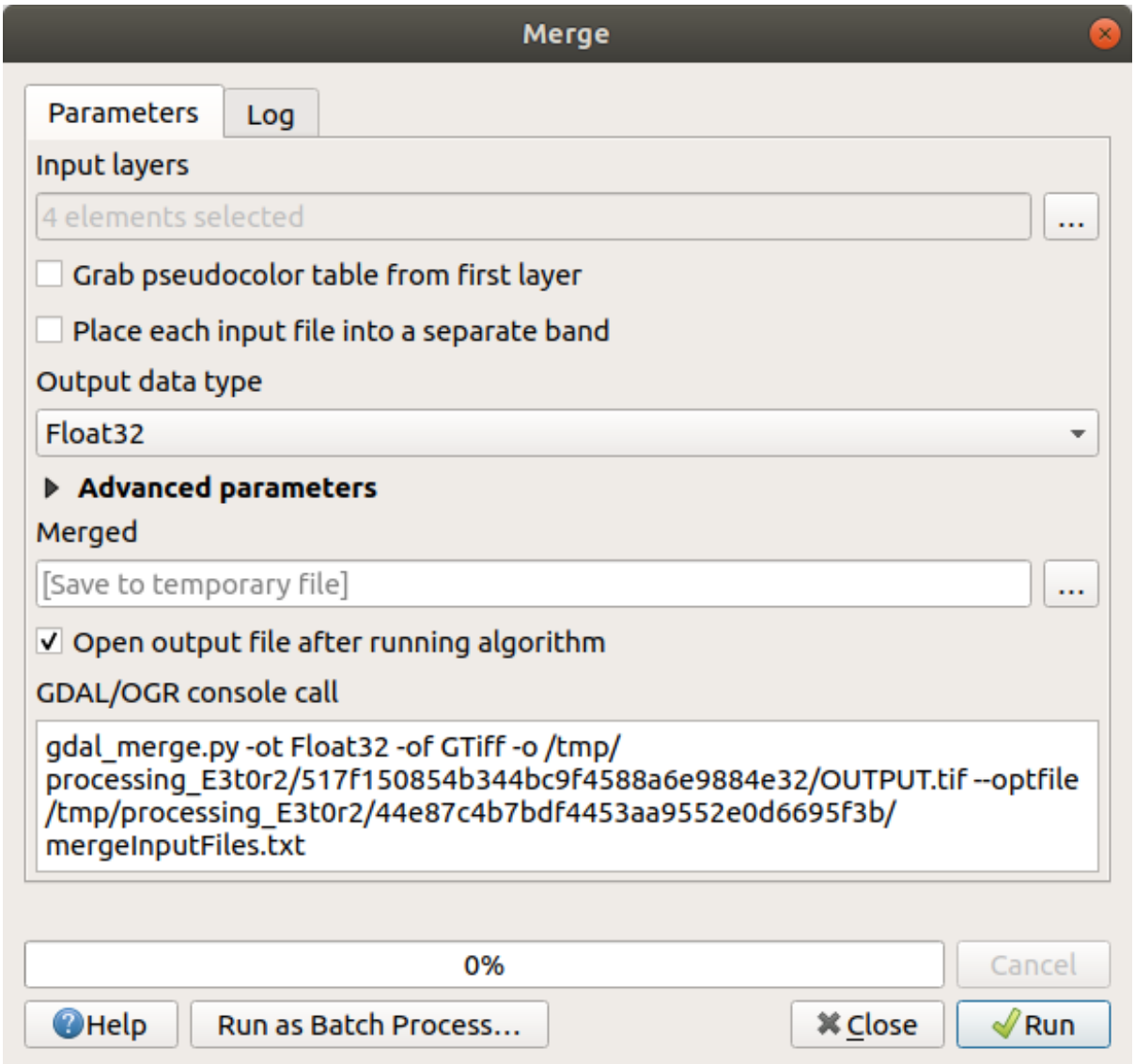
ラスタをマージする

新しいラスタレイヤを作成してディスクに保存する必要がある場合、マージアルゴリズムを使用することができます。

> **注釈**
>
> マージするラスタファイルの数や解像度によっては、新しく作成されるラスタファイルが非常に大きくなることがあります。代わりに **仮想ラスタの作成** セクションで説明したように、ラスタカタログを作成することを検討してください。

1. *GDAL* **ラスタその他** メニューから、 **結合** *(gdal_merge)* アルゴリズムをクリックします。
2. **仮想ラスタの作成** で行ったように、 ... ボタンを使って結合するレイヤを選択します。

   また、入力に仮想ラスタを指定すると、そのラスタを構成する全てのラスタが処理されます。
3. GDAL ライブラリを知っていれば、 **詳細パラメータ** メニューを開いて、独自のオプションを追加することも可能です。



7.1.4 結論

QGIS では既存のプロジェクトにラスターデータを入れることが簡単です。

7.1.5 次は?

次は航空画像ではないラスターデータを使用して、同様にラスターの場合にシンボル化はどのように有用であるかを見ていきます。

7.2 レッスン: ラスタのシンボロジを変更する

ラスタデータは航空写真ばかりではありません。ラスタデータには他にも様々なものがあり、それらの多くでは、適切に可視化し有用となるようにシンボル化することが不可欠です。

このレッスンの目標: ラスタレイヤのシンボロジを変更します。

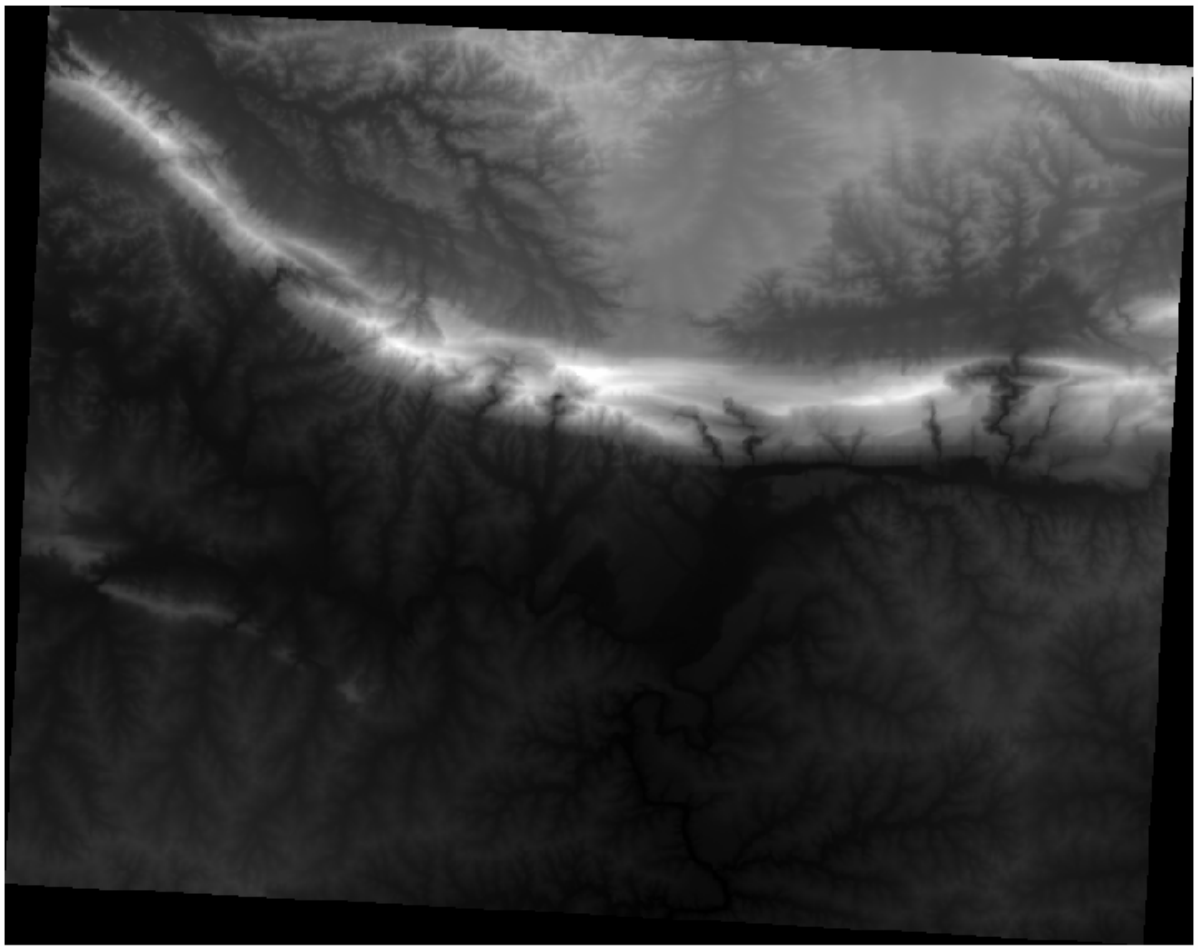
7.2.1 ★☆☆ （初級レベル） 自分でやってみよう:

1. **ブラウザ** パネルを使用して、 `exercise_data/raster/SRTM/` にある `srtm_41_19.tif` をロードします
2. **レイヤパネル** でこのレイヤを右クリックし、 **レイヤの領域にズーム** を選択し、このレイヤの範囲にズームします。

このデータセットは、**デジタル標高モデル** *(DEM)* です。地形の標高（高度）を表した地図で、山や谷がどこにあるのかなどを確認することができます。

前節のデータセットでは各ピクセルに色情報が含まれていましたが、*DEM* では各ピクセルに標高の値が含まれています。

DEM を読み込むと、グレースケールで表示されていることがわかります:



QGIS は可視化のために画像のピクセル値に自動的に引き伸ばしをかけていますが、この仕組みについては、この先で詳しく説明します。

7.2.2 ★☆☆ （初級レベル） 理解しよう: ラスタレイヤのシンボロジを変更する

ラスタシンボロジを変更するには、2 種類のオプションがあります:

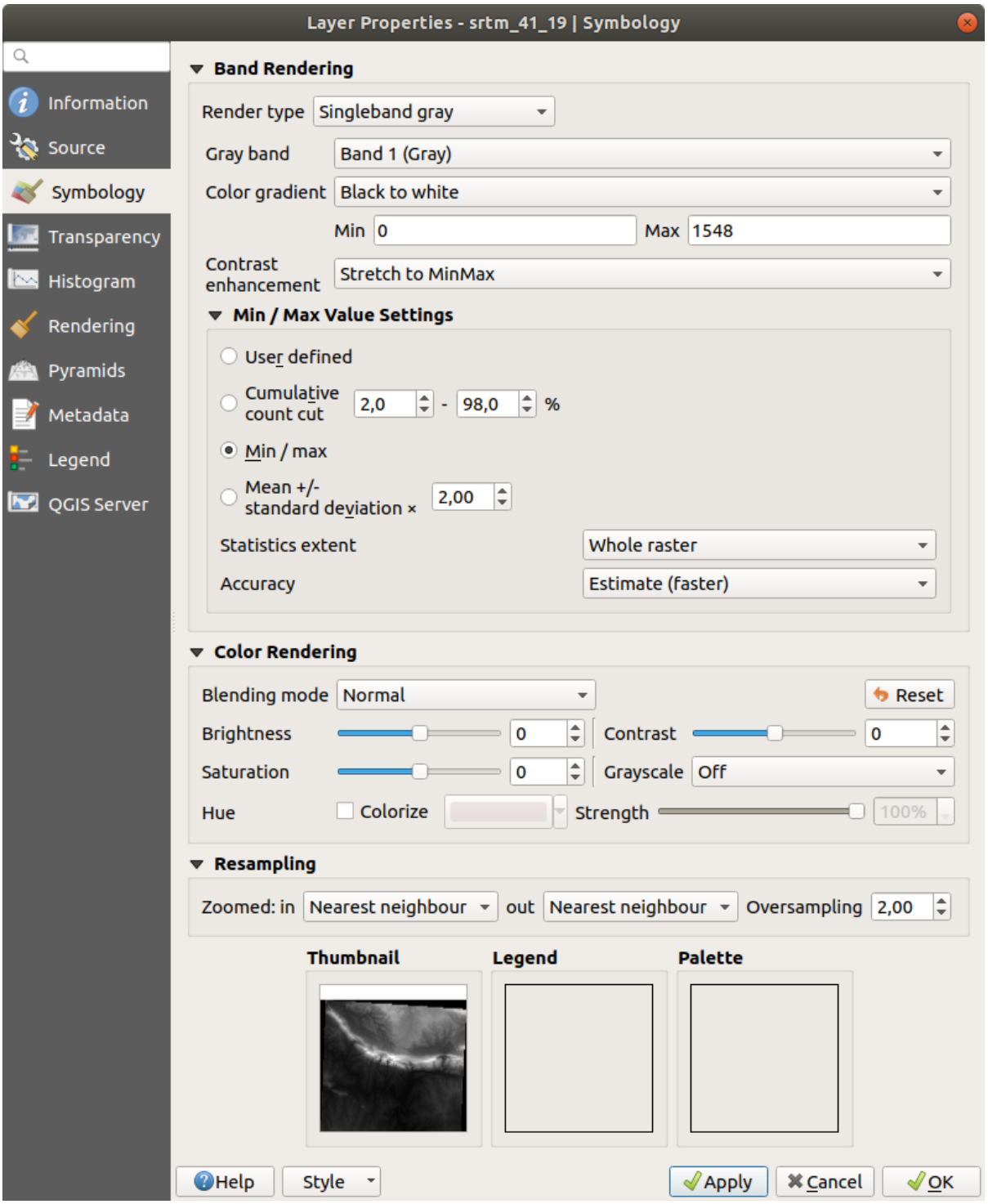
1. レイヤツリーでレイヤを右クリックし、 **プロパティ** オプションを選択して、 **レイヤプロパティ** ダイアログを開きます。次に、 **シンボロジ** タブに切り替えます
2. **レイヤ** パネルのすぐ上にある  レイヤのスタイルパネルを開く ボタンをクリックします（ショートカット F7)。これにより、 **レイヤスタイル** パネルが開き、そこで  シンボロジ タブに切り替えることができます。

好みの方法を選んでください。

7.2.3 ★☆☆ （初級レベル） 理解しよう: 単バンドグレー

ラスタファイルを読み込んだとき、それが前節のような写真画像でない場合、デフォルトのスタイルはグレースケールのグラデーションに設定されます。

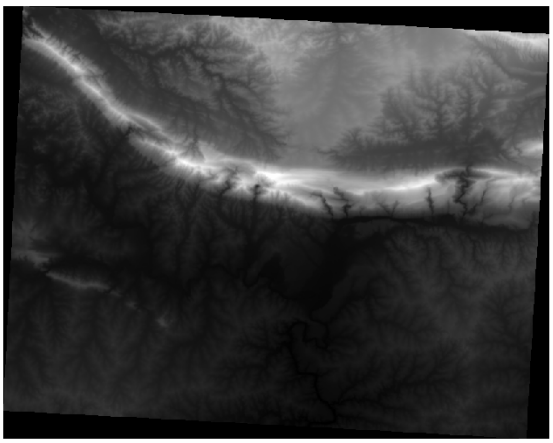
このレンダラーの特徴を探ってみましょう。



デフォルトの **グラデーション** は 黒から白 に設定されており、低いピクセル値は黒、高いピクセル値は白になるように設定されています。この設定を 白から黒 に反転させて、その結果を確認してみてください。

とても重要なのは **コントラスト** パラメータです。デフォルトでは 最小最大範囲に引き伸ばす に設定されており、ピクセル値が最小値と最大値に引き伸ばされることを意味します。

コントラストを強化した場合（左）としていない場合（右）の違いを見てください:





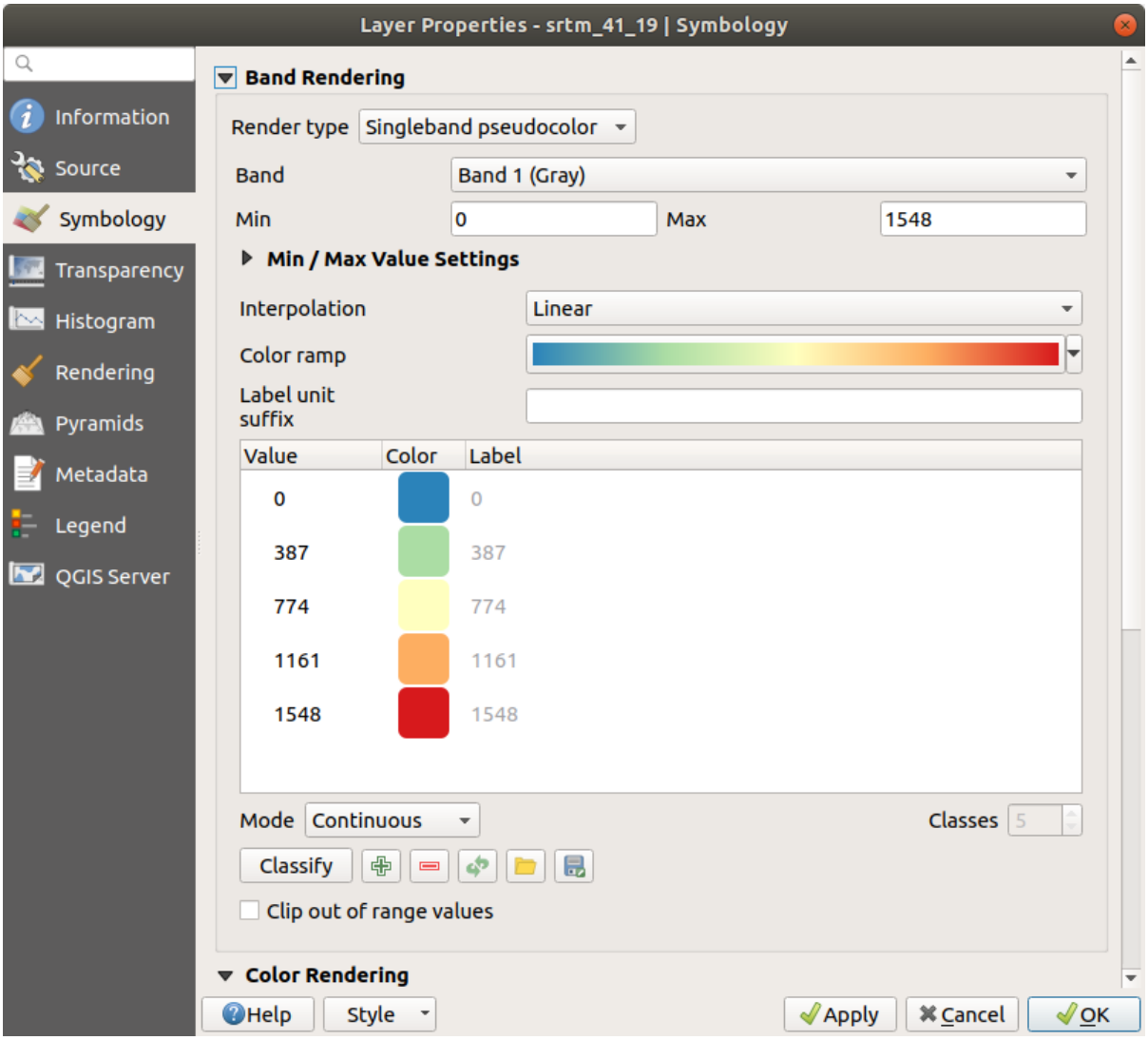
しかし、引き伸ばしに使用すべき最小値と最大値とは何でしょうか？現在 **最小 / 最大値設定** の下にあるものです。最小値と最大値を計算し、それを引き伸ばしに使用する方法は多くあります。

1. **ユーザー定義**: **最小値** *(Min)* と **最大値** *(Max)* を手入力します
2. **累積範囲**: これは、極端に低い値や高い値がある場合に便利です。これは値の 2% (または選択した値) を **カット** します
3. **最小/最大**: ラスタの最小値と最大値の **実測値** または **推定値**
4. **平均 +/- 標準偏差**: 値は平均値と標準偏差で計算されます。

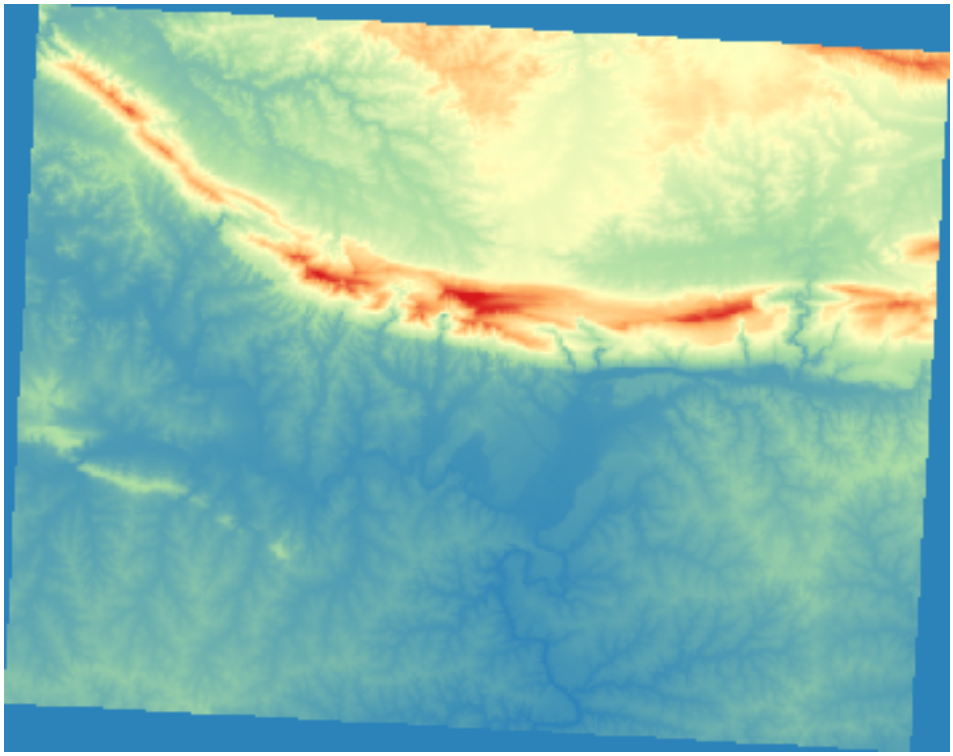
7.2.4 ★☆☆ （初級レベル） 理解しよう: 単バンド疑似カラー

グレースケールは、ラスタレイヤのスタイルとして必ずしも優れているとは言えません。DEM をよりカラフルにしてみましょう。

- **レンダリングタイプ** を **単バンド疑似カラー** に変更します。読み込まれたデフォルトの色が気に入らない場合は、別の **カラーランプ** を選択します
- 新しい色分類を生成するには **分類** ボタンをクリックします
- 自動的に生成されない場合は、*OK* ボタンをクリックして、この分類を DEM に適用します



このようにラスターが見えるでしょう。

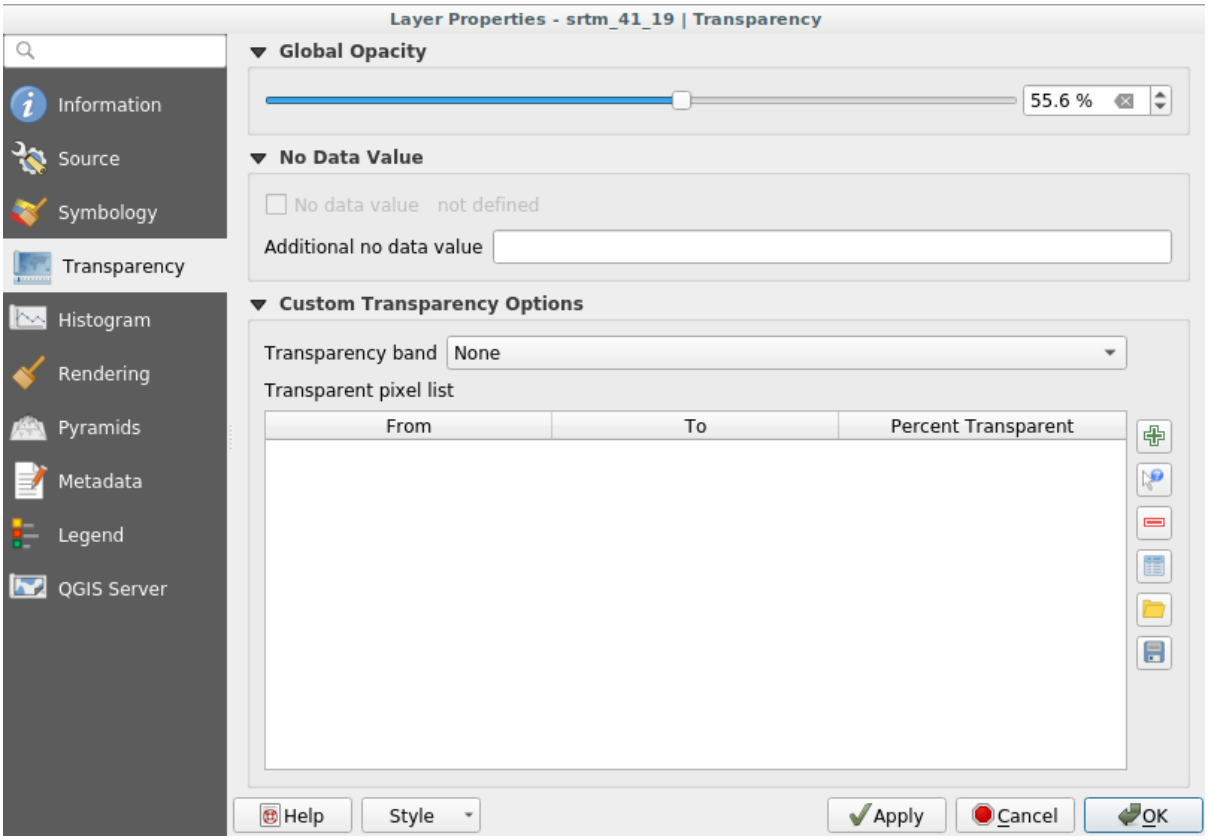


これは DEM を見る面白い方法です。これで、ラスタの値が再び適切に表示され、低い部分の青から高い部分の赤へと変化しているのがわかると思います。

7.2.5 理解しよう: 透明度を変える

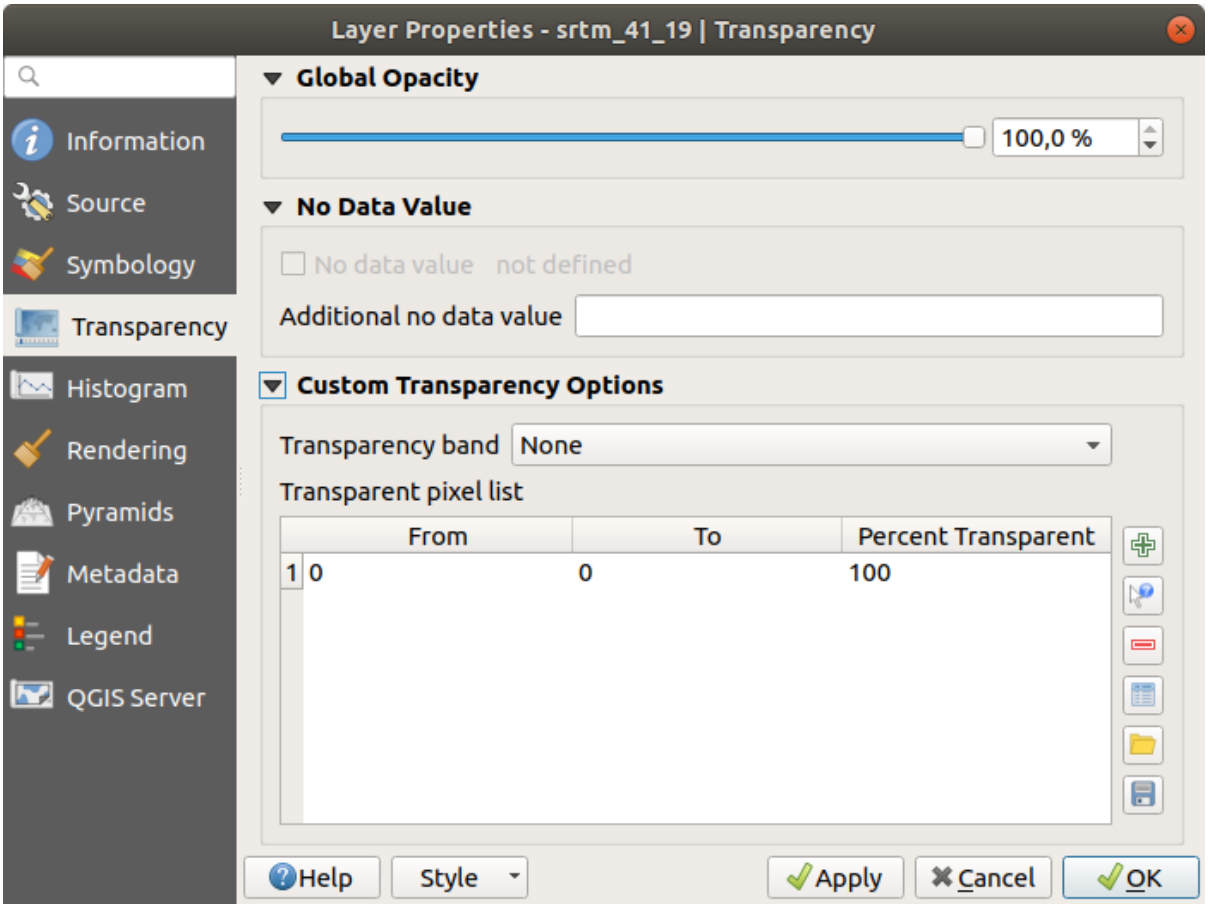
ラスタレイヤ全体の透明度を変えることで、ラスタ自体に覆われている他のレイヤを見ることができ、調査地域の把握がしやすくなることもあります。

ラスタ全体の透明度を変更するには、 **透明度** タブに切り替え、 **グローバルな不透明度** のスライダーを使用して不透明度を下げます:

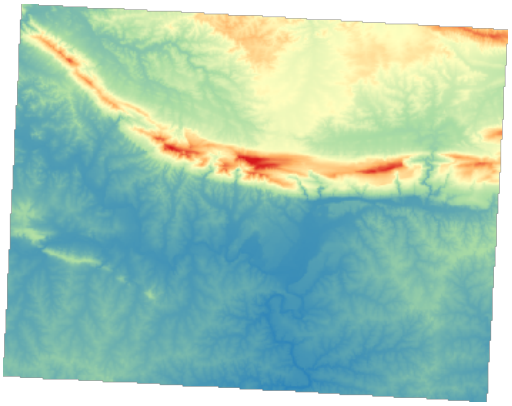


さらに興味深いのは、いくつかのピクセル値について透明度を変更することです。例えば、今回使用したラスタでは、角の部分に均質な色が見えます。これらのピクセルを透明に設定するには、 **透明度** タブにある **カスタム透過オプション** にアクセスします。

- 手動で値を追加 ボタンをクリックすると、値の範囲を追加して、その透過率を設定することができます
- 単独の値については、 画面から値を追加 ボタンがより便利です
- 画面から値を追加 ボタンをクリックします。ダイアログが消え、地図と対話できるようになります。
- DEM の隅にある均質な色をクリックします
- 透過率テーブルがクリックした値で埋められるのがわかると思います:



- *OK* をクリックしてダイアログを閉じ、変更内容を確認します。



わかりましたか？隅は 100%透過になりました。

7.2.6 結論

これらは、ラスタシンボロジを使い始めるための基本的な機能の一部です。このほかにも、パレット値やユニーク値を使ってレイヤを記号化したり、マルチスペクトル画像で異なるバンドを異なる色で表現したり、自動的な陰影起伏効果（DEM ラスタファイルでのみ有効）を作ったりと、QGIS では多くのオプションを利用することができます。

7.2.7 参照

SRTM データセットは CGIAR CSI ウェブサイトから入手しました。

7.2.8 次は?

私たちはデータを適切に表示することができるようになりました。それではさらに、それを解析する方法を調べてみましょう。

7.3 レッスン: 地形解析

ある種のラスタからはそれが表す地形の洞察をより多く得ることができます。数値標高モデル (DEM) がこの点では特に有用です。このレッスンでは先ほどからの住宅開発案の調査地域についてより詳しく調べるのに地形解析ツールを使用します。

このレッスンの目標: 地形に関する詳細な情報を取得するために地形解析ツールを使用します。

7.3.1 ★☆☆ （初級レベル） 理解しよう: 陰影図を計算する

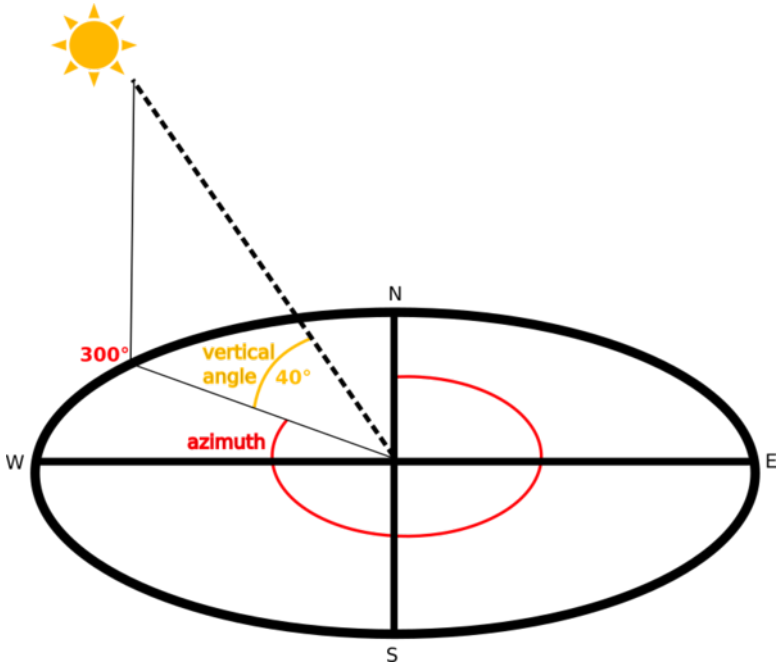
前のレッスンで使用したのと同じ DEM レイヤを使用することにします。この章をゼロから始める場合は、**ブラウザ** パネルを使用して、 `raster/SRTM/srtm_41_19.tif` を読み込んでください。

DEM レイヤは、地形の標高を表示しますが、少し抽象的に見えることがあります。DEM レイヤは、必要な地形の 3D 情報をすべて含んでいますが、3D オブジェクトのようには見えません。地形の印象を良くするために、**陰影図** を計算することができます。これは、光と影を使って地形をマッピングするラスタで、3D に見える画像を作成することができます。

ここでは、 **ラスタ　ラスタ地形解析** メニューにあるアルゴリズムを使用することにします。

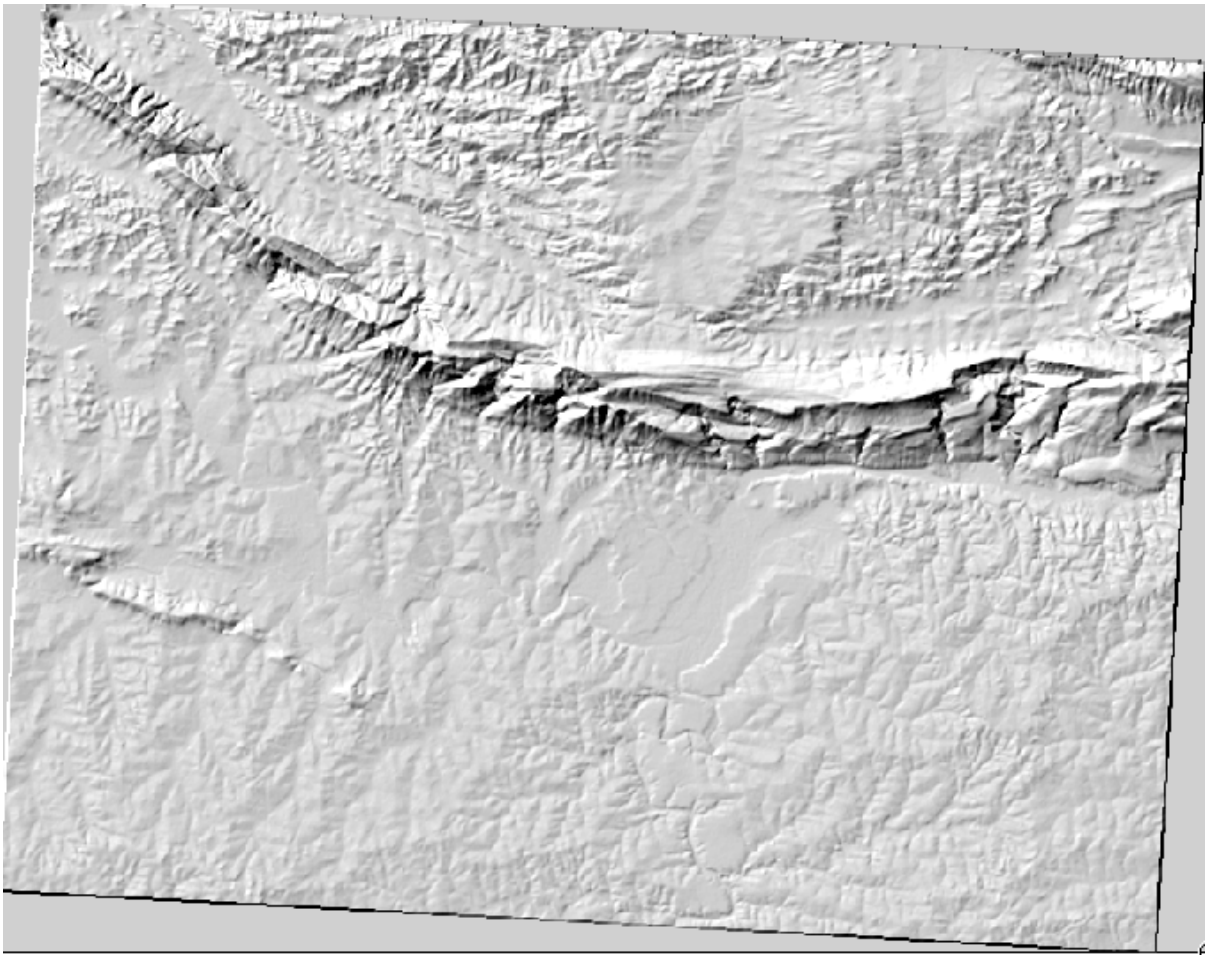
1. **陰影図** メニューをクリックします
2. このアルゴリズムでは、光源の位置を指定することができます: **方位角** *(azimuth)* **は** *0*（**北**）**から** *90*（**東**）、*180*（**南**）、*270*（**西**）**の値を持ち、** *:guilabel:* **太陽高** は光源の高さを 0 から 90 度までで指定します。
3. 以下の値を使用することにします:

- Z **係数** に 1.0
- **方位角（水平角）** に 300.0 °
- **太陽高** に 40.0 °



4. 新しいフォルダ exercise_data/raster_analysis/ に hillshade.tif という名前でファイルを保存します
5. 最後に **実行** をクリックします

You will now have a new layer called hillshade that looks like this:



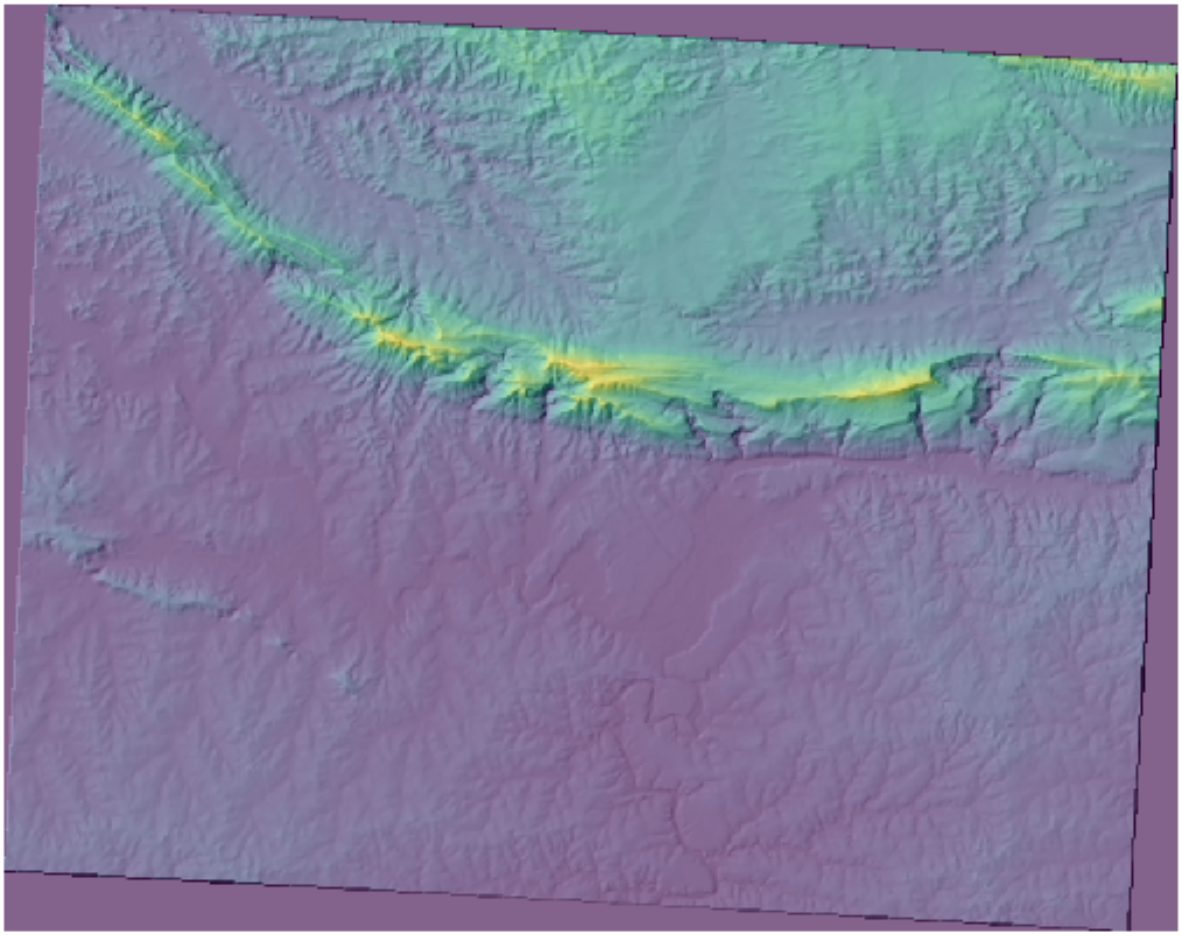
きれいで 3 次元的に見えますが、これは改善できるでしょうか? 陰影図はそれだけでは石膏模型のように見えます。どうにかしてそれを他のよりカラフルなラスタと一緒に使用できないでしょうか? もちろんできます。オーバーレイとして陰影図を使用します。

7.3.2 ★☆☆ （初級レベル） 理解しよう: 陰影図をオーバーレイとして使う

陰影図は一日のある時点の日光について非常に有用な情報を提供することができますが審美的な目的で使うこともできます。それを使えば地図をよりよく見せることができます。陰影図をほとんど透過させる設定がその鍵となります。

1. Change the symbology of the original `srtm_41_19` layer to use the *Pseudocolor* scheme as in the previous exercise
2. Hide all the layers except the `srtm_41_19` and `hillshade` layers
3. Click and drag the `srtm_41_19` to be beneath the `hillshade` layer in the *Layers* panel
4. Set the `hillshade` layer to be transparent by clicking on the *Transparency* tab in the layer properties
5. **全体の不透明度** を `50%` に設定します。

   このような結果が得られます:



6. Switch the `hillshade` layer off and back on in the *Layers* panel to see the difference it makes.

Using a hillshade in this way, it's possible to enhance the topography of the landscape. If the effect doesn't seem strong enough to you, you can change the transparency of the `hillshade` layer; but of course, the brighter the hillshade becomes, the dimmer the colors behind it will be. You will need to find a balance that works for you.

終了したら、プロジェクトを保存することを忘れないでください。

7.3.3 理解しよう: 最適な地域を見つける

前回、 **ベクタ解析** のレッスンで取り上げた不動産業者の問題を思い出してください。ここでは、買い手が建物を購入し、その敷地に小さなコテージを建てたいと考えているとします。南半球では、理想的な開発用地には以下のようなエリアが必要であることが分かっています:

- 北向きである
- 傾斜が 5 度以下
- 傾斜が 2 度以下であれば、傾斜方向は問題になりません。

最適なエリアを探しましょう。

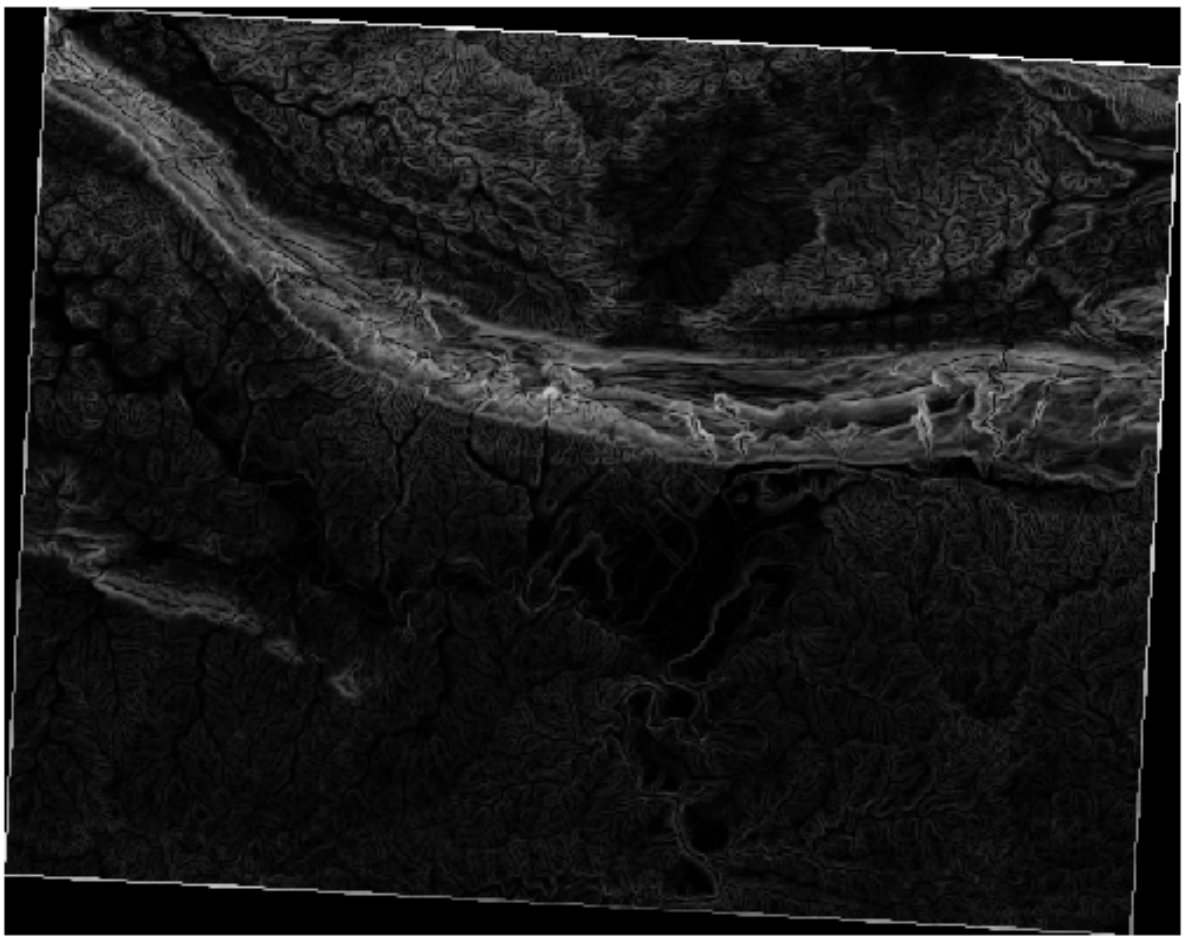
7.3.4 ★★☆ （中級レベル） 理解しよう: 傾斜を計算する

傾斜 は、地形がどの程度急であるかを知らせます。例えば、そこの土地に家を建てたいのであれば、比較的平坦な土地が必要です。

傾斜を計算するには、 **プロセシング** **ラスタ地形解析** の **傾斜** *(slope)* アルゴリズムを使用する必要があります。

1. アルゴリズムを開きます
2. Choose `srtm_41_19` as the *Elevation layer*
3. *Z* **係数** を `1.0` に維持します
4. 出力を `slope.tif` という名前のファイルとして、 `hillshade.tif` と同じフォルダーに保存します
5. **実行** をクリックします

各ピクセルが対応する傾斜の値を持つ、地形の傾斜が表示されます。黒いピクセルは平坦な地形、白いピクセルは急な地形を表しています:



7.3.5 ★★☆ （中級レベル） 自分でやってみよう: 傾斜方位を計算する

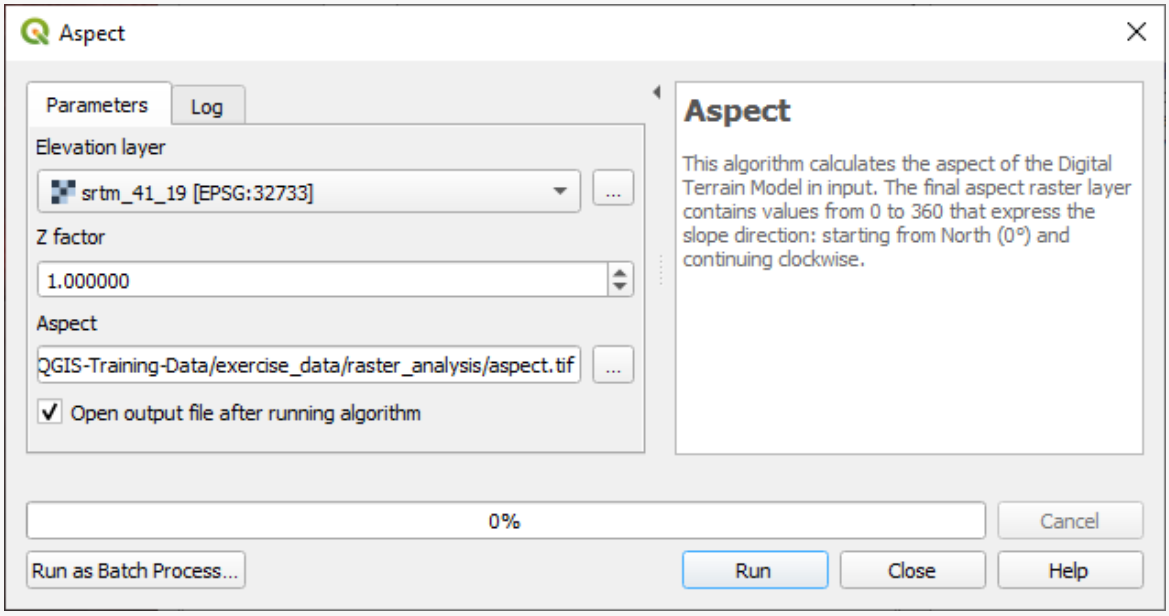
傾斜方位 は、地形の傾斜が向いている磁針方向です。傾斜方位が 0 だと北向き、90 だと東向き、180 だと南向き、270 だと西向きの斜面であることを意味します。

この調査は南半球で行われているため、不動産物件は太陽光が残るように北向きの斜面に建てるのが理想的です。

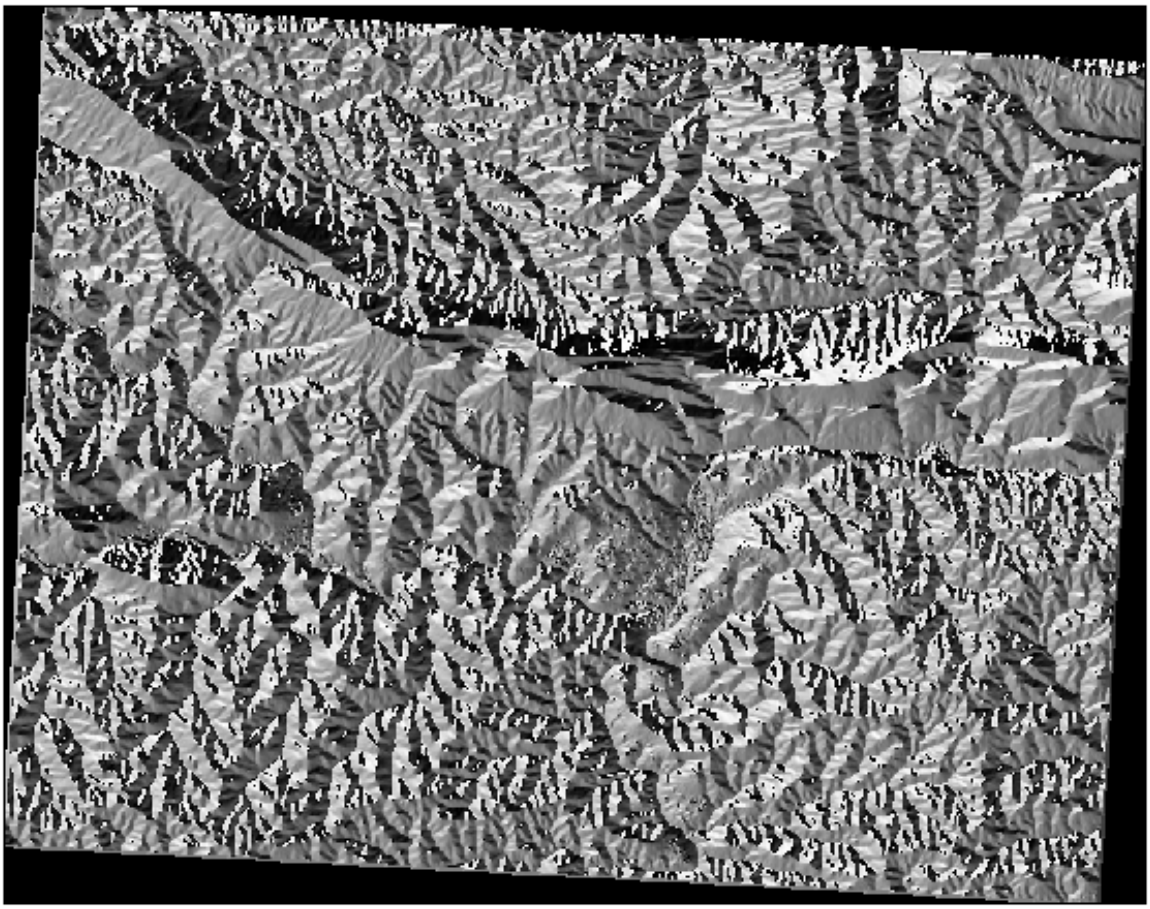
プロセシング　ラスタ地形解析 の **傾斜方位** *(aspect)* アルゴリズムを使用して、 `aspect.tif` レイヤを `slope.tif` と共に保存することができます。

答え

傾斜方位 *(aspect)* ダイアログをこのように設定します:



結果はこうなります:



7.3.6 ★★☆ （中級レベル） 理解しよう: 北向きの傾斜方位を見つける

今、ラスタは傾斜方位だけでなく傾斜も表示していますが、どこが理想的な条件を一度に満たしているのか知るすべはありません。どうやって解析すればいいのでしょうか？

その答えは **ラスター計算機** です。

QGIS では、さまざまなラスタ計算機が利用できます:

- **ラスタ　ラスタ計算機**
- プロセシングの:
  - **ラスタ解析**　*Raster calculator*
  - *GDAL*　**ラスタその他**　**ラスタ計算機**
  - *SAGA*　*Raster Calculus*　*Raster Calculator*

各ツールは同じ結果を導いていますが、構文が若干異なったり、演算子の有無が異なったりすることがあります。

ここでは、**プロセシングツールボックス** の **ラスタ解析** *Raster calculator* を使います

1. それをダブルクリックしてツールを開きます。

   - ダイアログの左上には、読み込んだすべてのラスタレイヤが `name@N` としてリストアップされます。`name` はレイヤの名前で `N` はバンドです。
   - 右上の部分には、さまざまな演算子が表示されています。ラスタが画像であると考えるのは少し待ってください。ラスタは数字で埋め尽くされた 2 次元の行列と考えるのが良いでしょう。

2. 北は 0（ゼロ）度なので、地形が北を向くためには、その傾斜方位が 270 度以上または 90 度以下である必要があります。したがって、式は次のようになります:

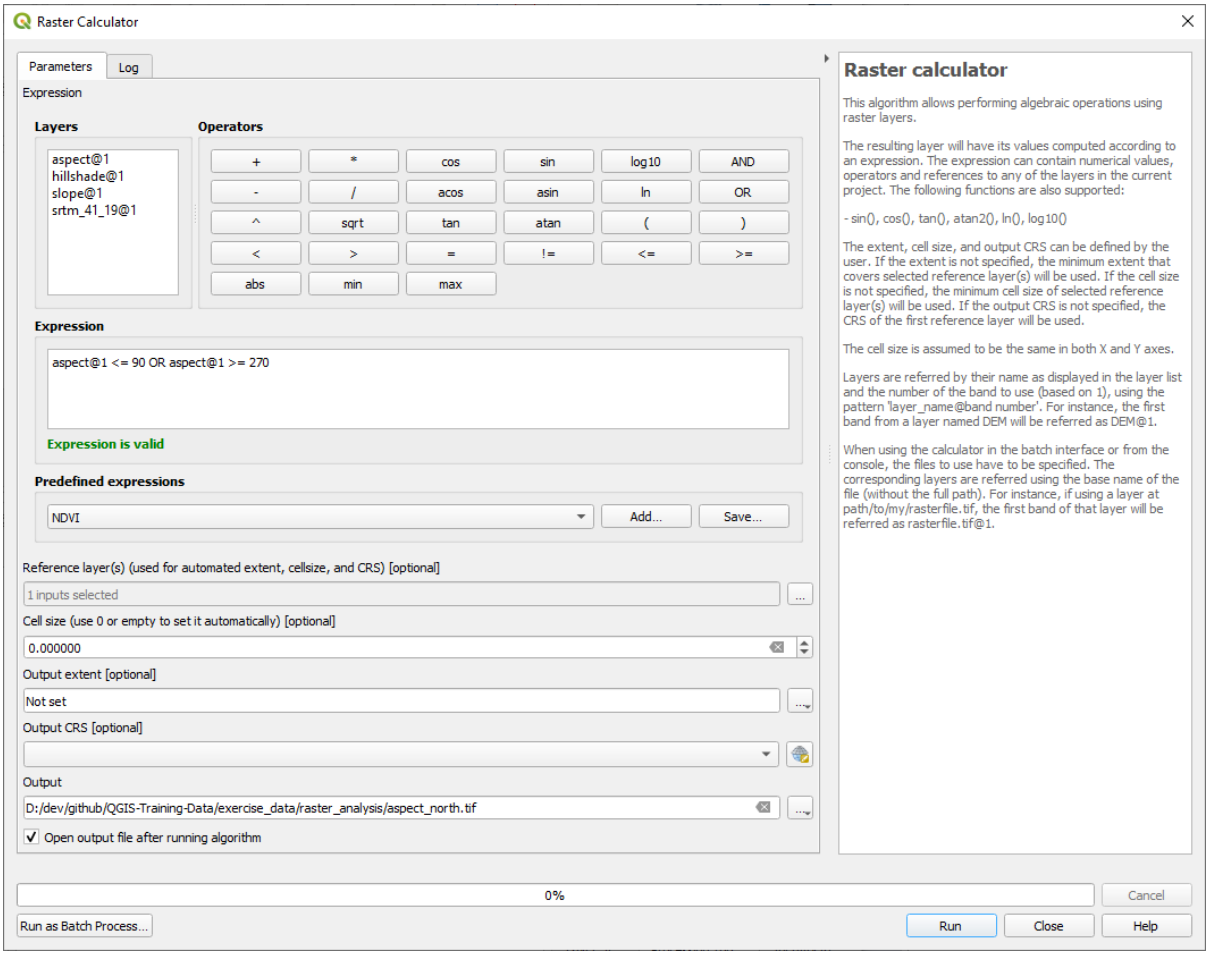
   ```
   aspect@1 <= 90 OR aspect@1 >= 270
   ```

3. 次に、セルサイズ、範囲、CRS などのラスタの詳細を設定する必要があります。これは手動で行うこともできますし、参照レイヤ を選択することで自動的に設定することもできます。この最後のオプションは、 *Reference layer(s)* パラメータの隣にある ... ボタンをクリックすることで選択できます。

4. In the dialog, choose the `aspect` layer, because we want to obtain a layer with the same resolution.

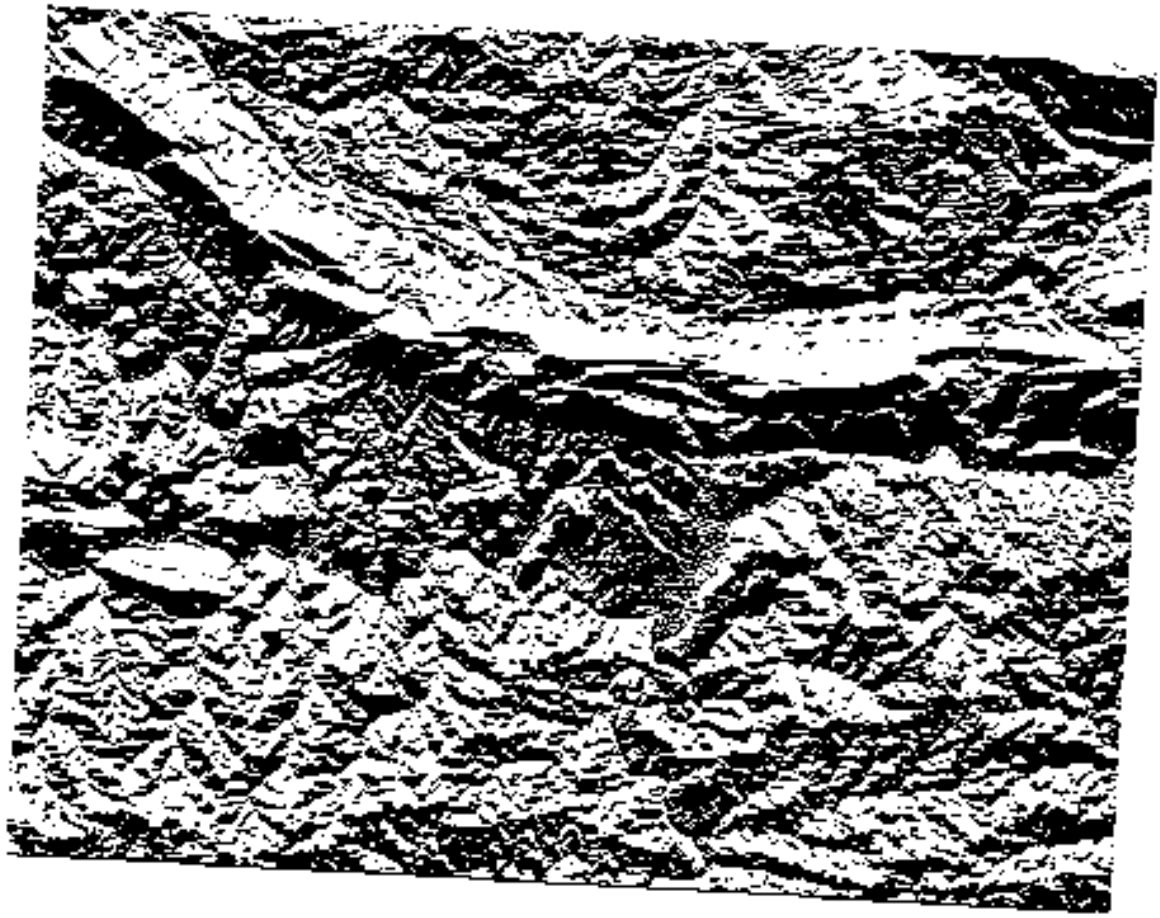
5. レイヤを `aspect_north.tif` として保存します。

   ダイアログはこのようになります:



6. 最期に **実行** をクリックします。

結果はこのようになります:



出力値は 0 または 1 です。どういうことでしょうか？ラスターの各ピクセルに対して、書いた式は条件にマッチするかどうかを返します。したがって、最終的な結果は **偽** (0) と **真** (1) になります。

7.3.7 ★★☆ （中級レベル） 自分でやってみよう: その他の条件

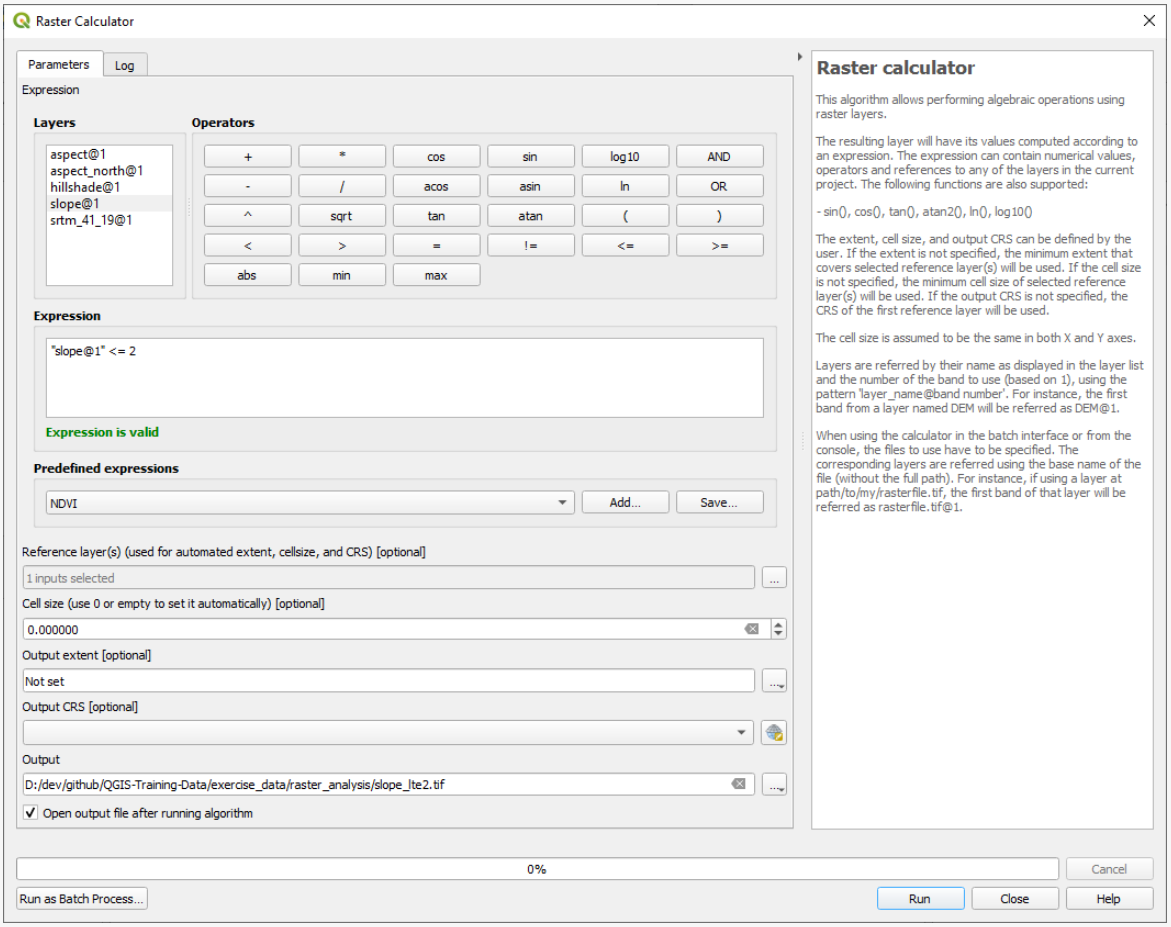
傾斜方位を行ったので、DEM から新規レイヤを 2 枚作成します。

- 最初のものは傾斜が 2 度以下の地域を特定します
- 2 つ目も同様ですが、傾斜は 5 度以下であることが必要です。
- それらを `exercise_data/raster_analysis` の下に `slope_lte2.tif` と `slope_lte5.tif` という名前で保存してください。

答え

- *Raster calculator* のダイアログを設定します:
- 次の式: `slope@1 <= 2`

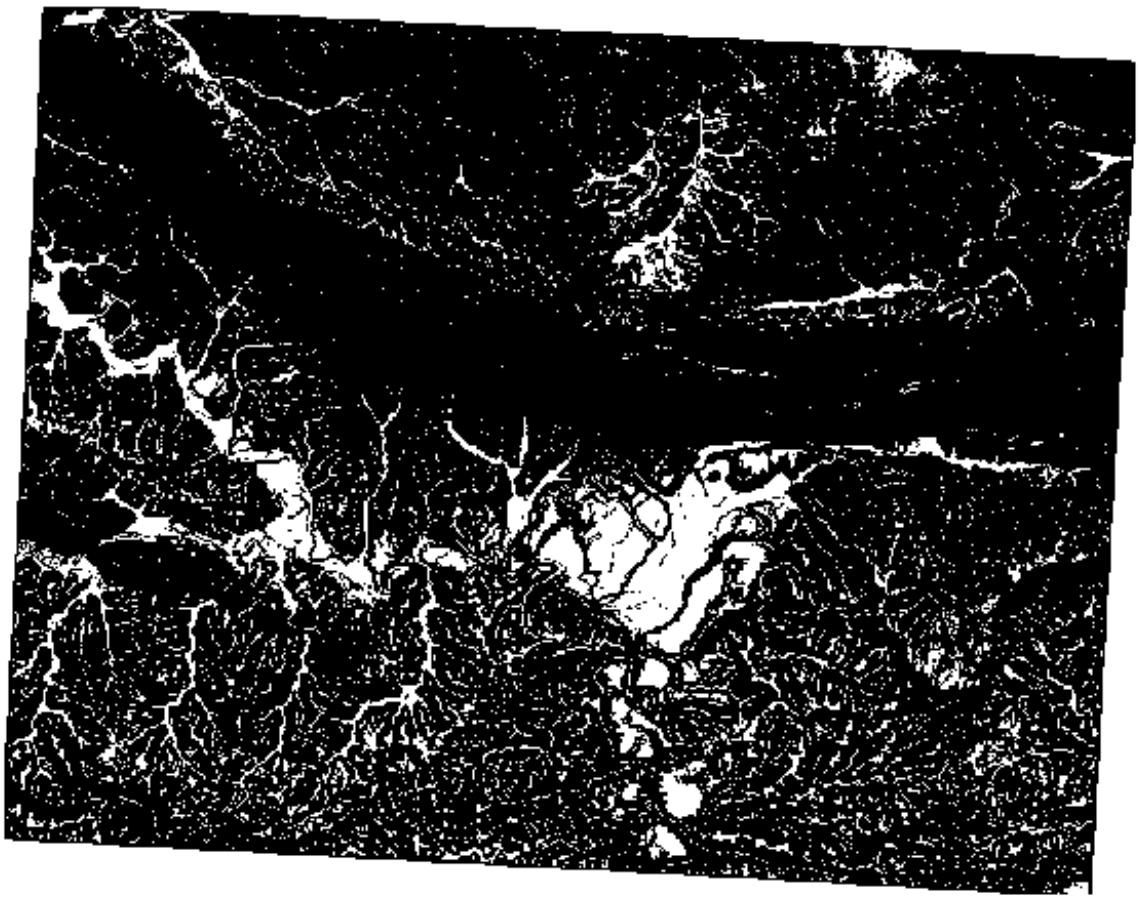
- `slope` レイヤを *Reference layer(s)* とします



- 5 度のバージョンは、式とファイル名の 2 を 5 に置き換えてください。

結果はこうなります:

- 2 度:



- 5 度:



7.3.8 ★★☆ （中級レベル） 理解しよう: ラスタ解析結果を組み合わせる

これで、DEM から 3 つのラスタレイヤを生成できました:

- `aspect_north`: terrain facing north
- `slope_lte2`: slope equal to or below 2 degrees
- `slope_lte5`: slope equal to or below 5 degrees

この条件を満たす場合、画素値は 1 となります。それ以外の場所では `0` となります。したがって、これらのラスタを掛け合わせると、すべてのピクセルの値が 1 となります（残りは `0` となります）。

条件を満たすものは:

- 傾斜が 5 度以下で、北向きの地形
- 傾斜が 2 度以下で、地形の向きは考慮しない。

ですから、傾斜が 5 度以下、かつ北向きの地形、または 傾斜が 2 度以下の地形を探す必要があります。そのような地形は開発に適しているでしょう。

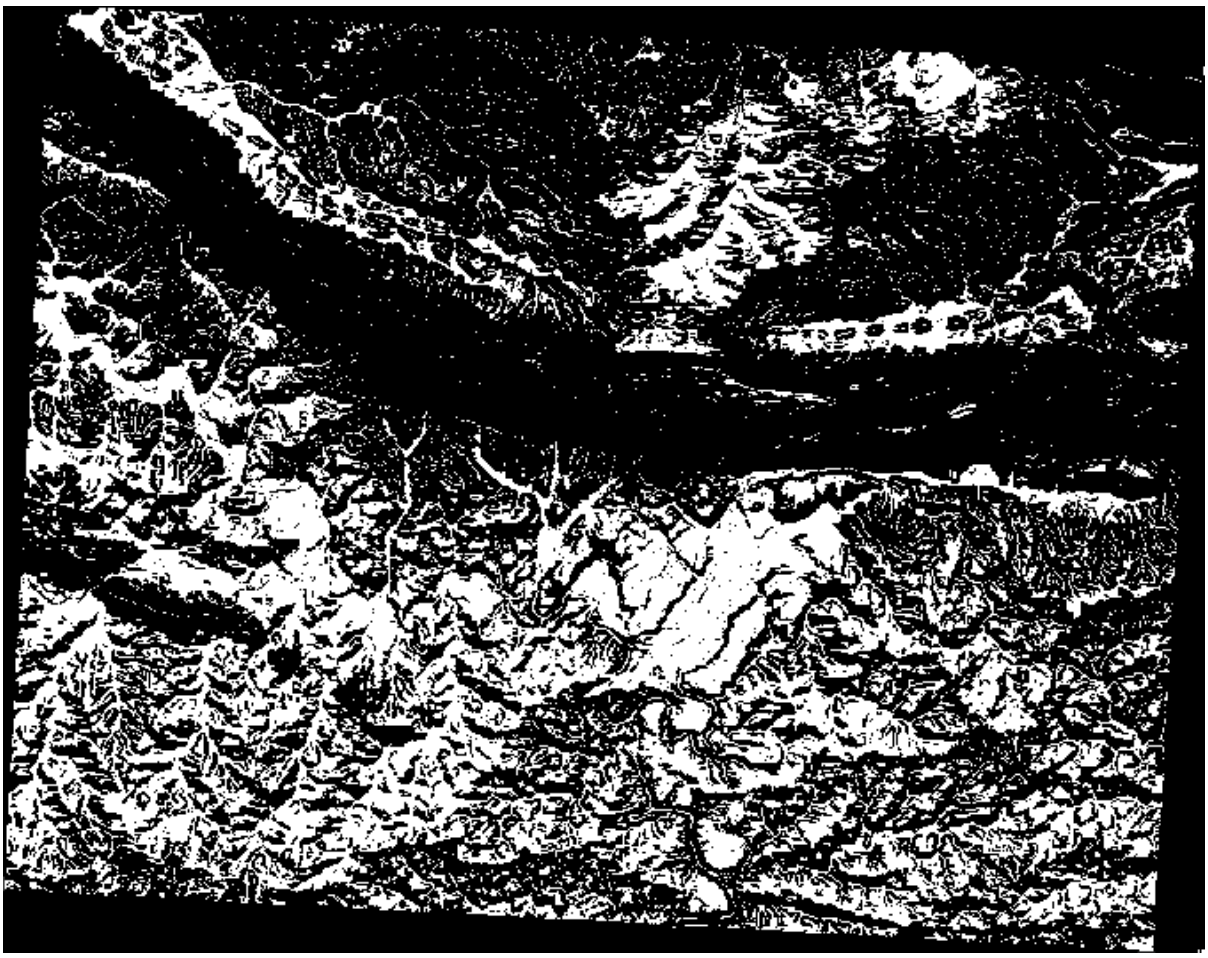
これらの抽出条件を満たすエリアを計算します:

1. 再び *Raster calculator* を開きます
2. この式を **式** で使用します:

```
( aspect_north@1 = 1 AND slope_lte5@1 = 1 ) OR slope_lte2@1 = 1
```

3. *Reference layer(s)* パラメータに `aspect_north` を設定します（他のレイヤを選択しても問題ありません - これらはすべて `srtm_41_19` から計算されています）
4. 出力結果を `exercise_data/raster_analysis/` の下に `all_conditions.tif` という名前で保存します
5. **実行** をクリックします

結果：



ヒント

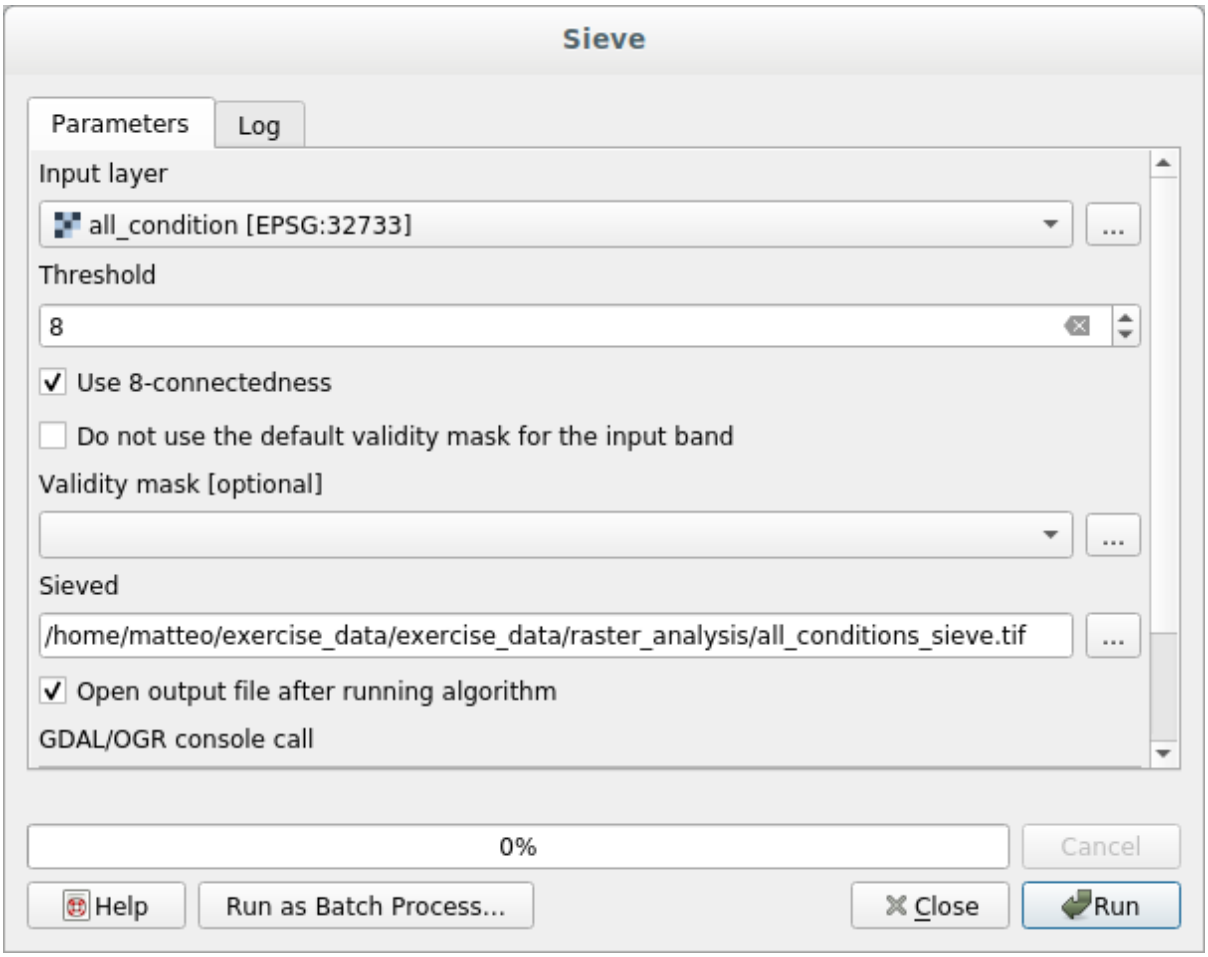
これまでの手順は、次のコマンドを使用することで簡略化することができます:

```
((aspect@1 <= 90 OR  aspect@1 >= 270) AND slope@1 <= 5) OR slope@1 <= 2
```

7.3.9 ★★☆ （中級レベル） 理解しよう: ラスタを簡素化する

上の画像からわかるように、複合的な解析によって、条件を満たす非常に小さな領域がたくさんできます（白い部分）。しかし、これらは小さすぎて何も構築できないため、解析にはあまり役に立ちません。このような小さくて使えない領域をすべて取り除いてしまいましょう。

1. **ふるい** *(sieve)* ツールを開きます（*GDAL* **ラスタ解析** in **プロセシングツール** ）
2. **入力レイヤ** に `all_conditions` を、**出力ファイル** に `all_conditions_sieve.tif`（`exercise_data/raster_analysis/` 以下）をセットします。
3. **閾値** を 8（最小 8 連続ピクセル）に設定し、 **８方向の連結関係をチェックする** をチェックします。

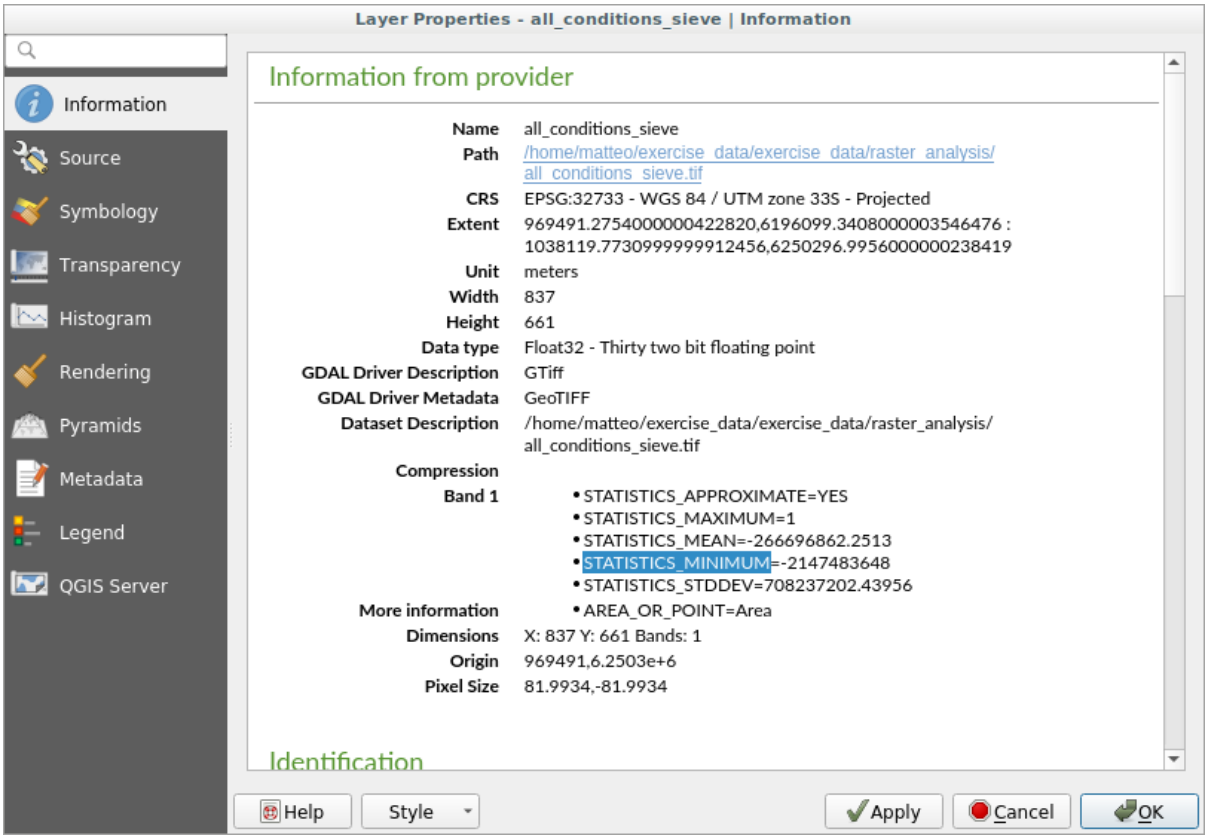


処理が完了すると、新しいレイヤーが読み込まれます。



どうなっているのでしょうか？その答えは、新しいラスターファイルのメタデータにあります。

4. **レイヤプロパティ** ダイアログの **情報** タブでメタデータを表示します。STATISTICS_MINIMUM の値を見てください:



このラスタは、元となったラスタと同様に、値「1」と「0」のみを特徴とするはずですが、非常に大きな負の数も持っています。データを調査すると、この数値はヌル値として機能することがわかります。我々はフィルタリングされなかった部分のみを対象にしているので、このヌル値をゼロに設定しましょう。

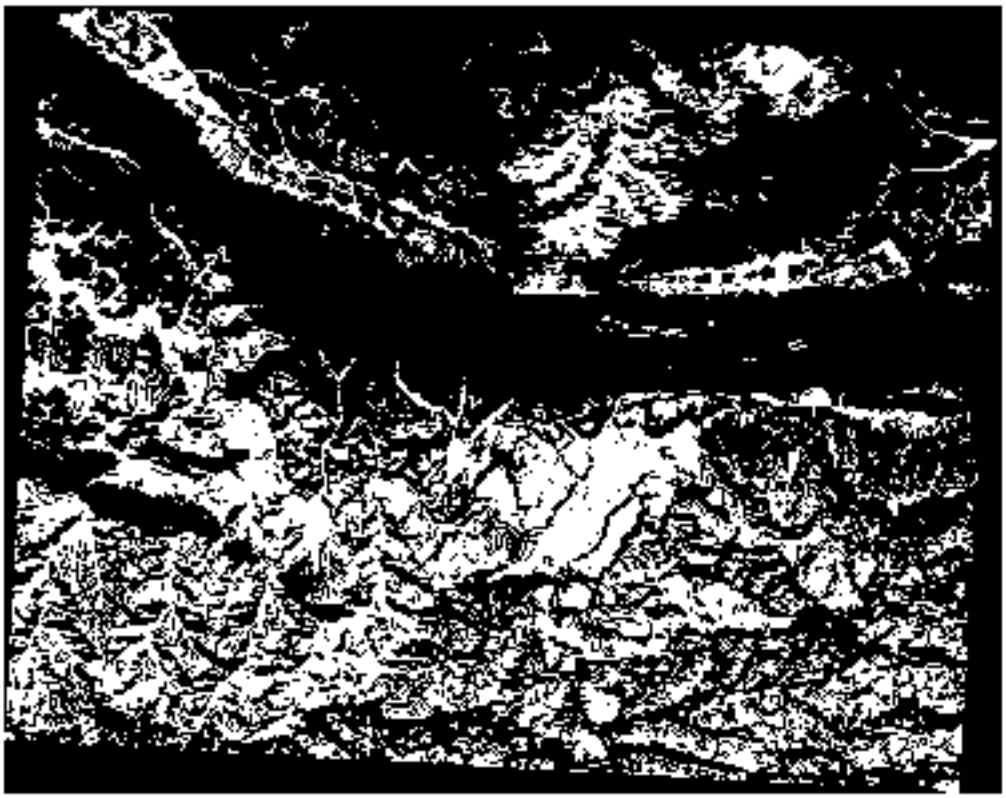
5. **ラスタ計算機** を開き、次の式を組み立てます:

```
(all_conditions_sieve@1 <= 0) = 0
```

これは、すべての非負の値を維持し、負の数をゼロに設定し、値 1 を持つすべての領域をそのまま残します。

6. 出力結果を `exercise_data/raster_analysis/` の下に `all_conditions_simple.tif` という名前で保存してください。

出力はこのようになります:



これは期待されたもので、以前の結果を簡素化したものです。あなたが得た結果が期待したものでない場合は、メタデータ（および該当する場合はベクタの属性）を見ると問題を解決するための要点がわかることを覚えておいて下さい。

7.3.10 ★★☆ （中級レベル） 理解しよう: ラスタを再分類する

*ラスタ計算機*を使用して、ラスタレイヤの計算を行いました。もう一つ、既存のレイヤから情報を抽出するための強力なツールがあります。

話を aspect レイヤに戻します。このレイヤは 0 から 360 の範囲の数値を持っていることが分かっています。このレイヤを傾斜方位に応じて他の不連続な値（1〜4）に*再分類*したいのです:

- 1 = 北（0 から 45 及び 315 から 360）;
- 2 = 東（45 から 135）
- 3 = 南（135 から 225）
- 4 = 西（225 から 315）

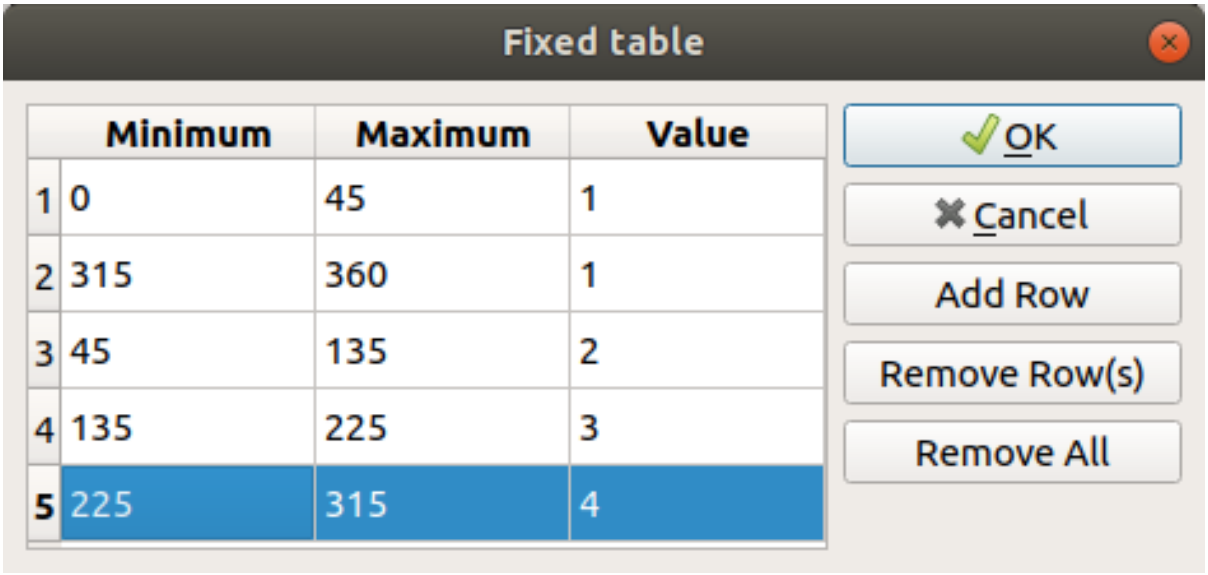
この操作はラスタ計算機でも可能ですが、計算式が非常に大きくなってしまいます。

代替ツールとしては、 **プロセシングツールボックス** の **ラスタ解析** にある **区分表（テーブル）で再分類** ツールを使用します。

1. ツールを開きます
2. Choose aspect as the *Raster layer*

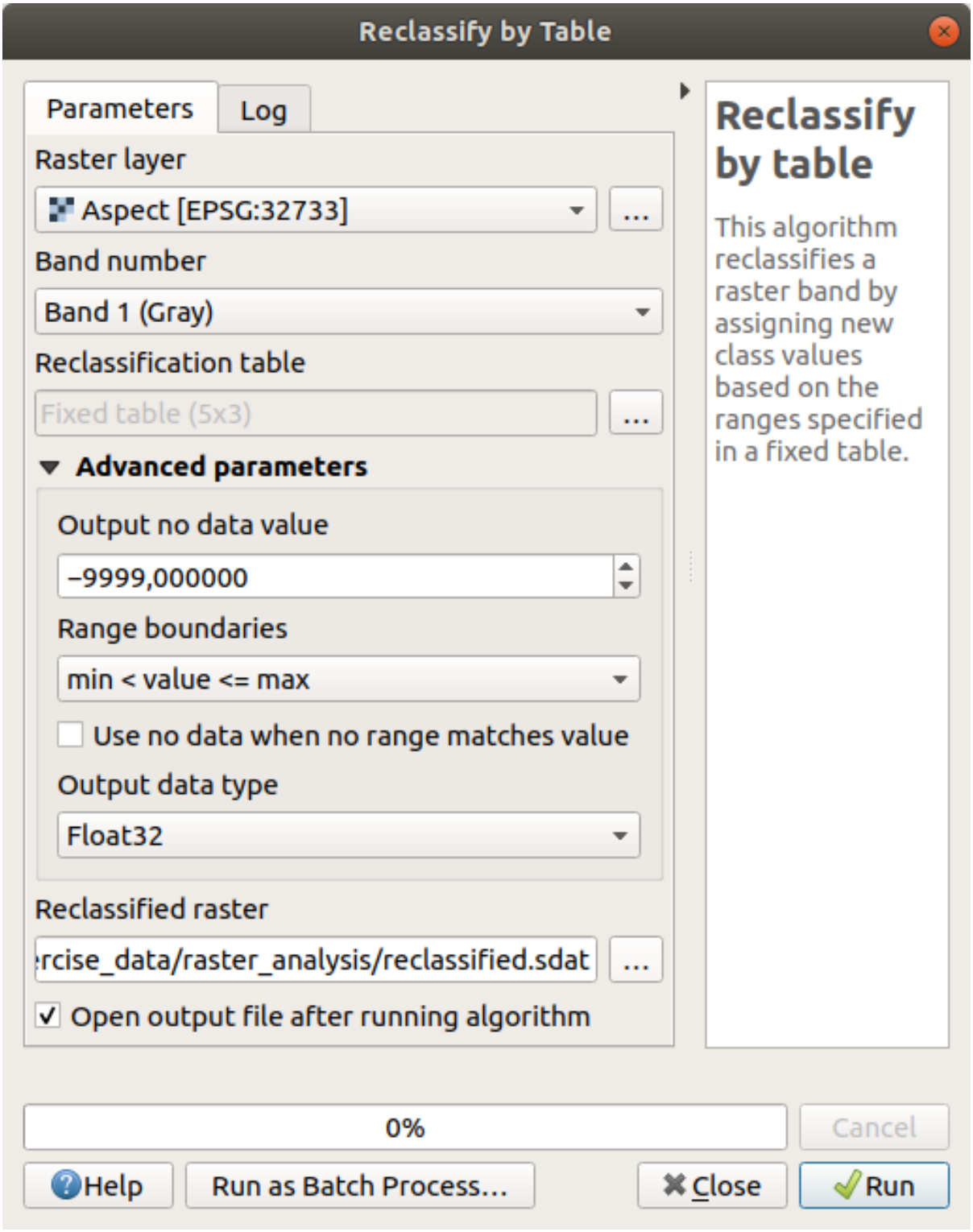
3. **再分類の区分表（テーブル）** の ... をクリックします。表のようなダイアログがポップアップ表示され、各区分の最小値、最大値、新しい値を選択することができます。

4. **行を追加** ボタンをクリックし、行を 5 つ追加します。各行を下図のように記入し、 *OK* をクリックします:



   各クラスの閾値を扱うためにアルゴリズムが用いる方法は、 **分類区分の境界上の扱い** によって定義されます。

5. レイヤを `reclassified.tif` という名前で `exercise_data/raster_analysis/` フォルダに保存します



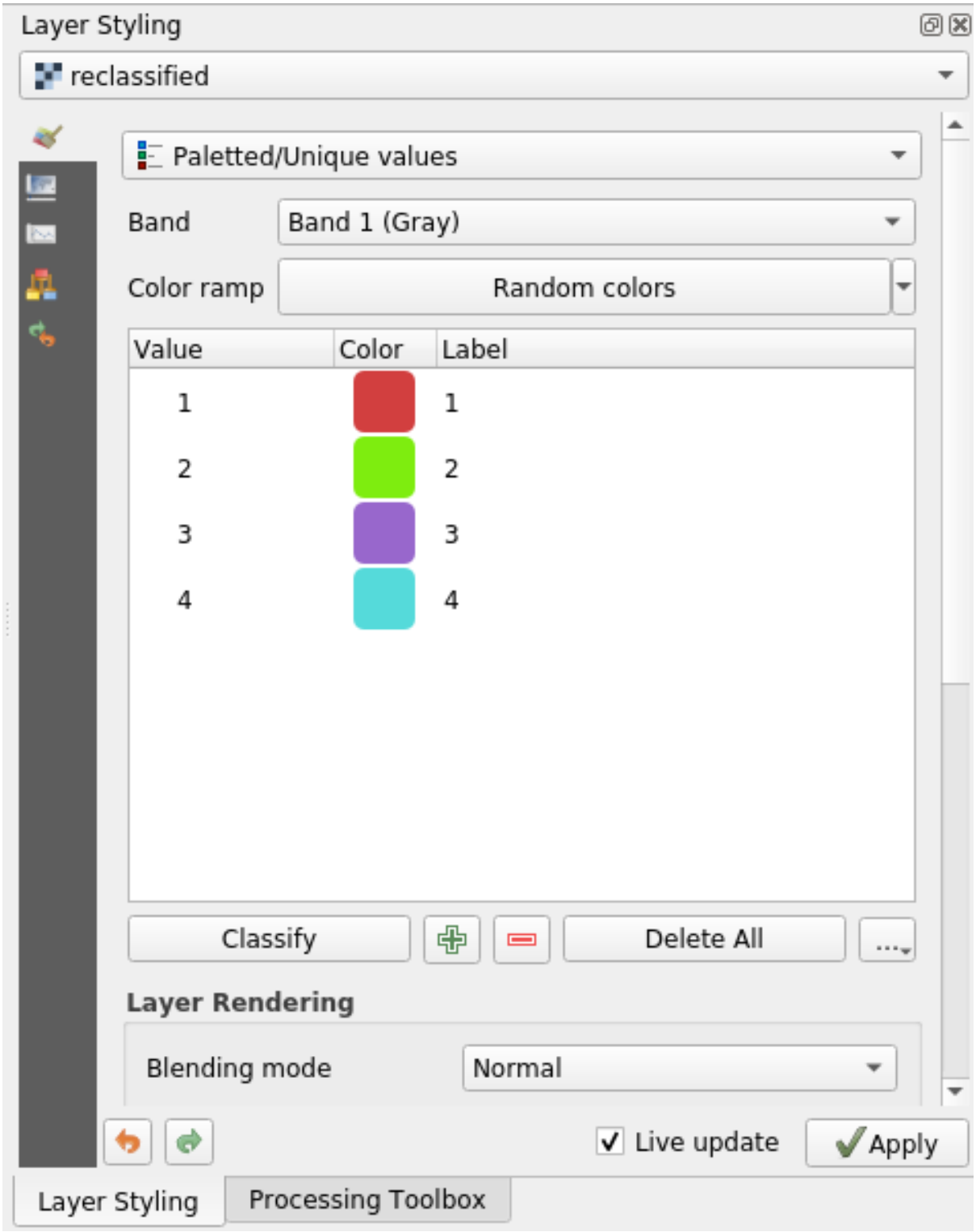
6. **実行** をクリックします

If you compare the native `aspect` layer with the `reclassified` one, there are not big differences. But by looking at the legend, you can see that the values go from 1 to 4.

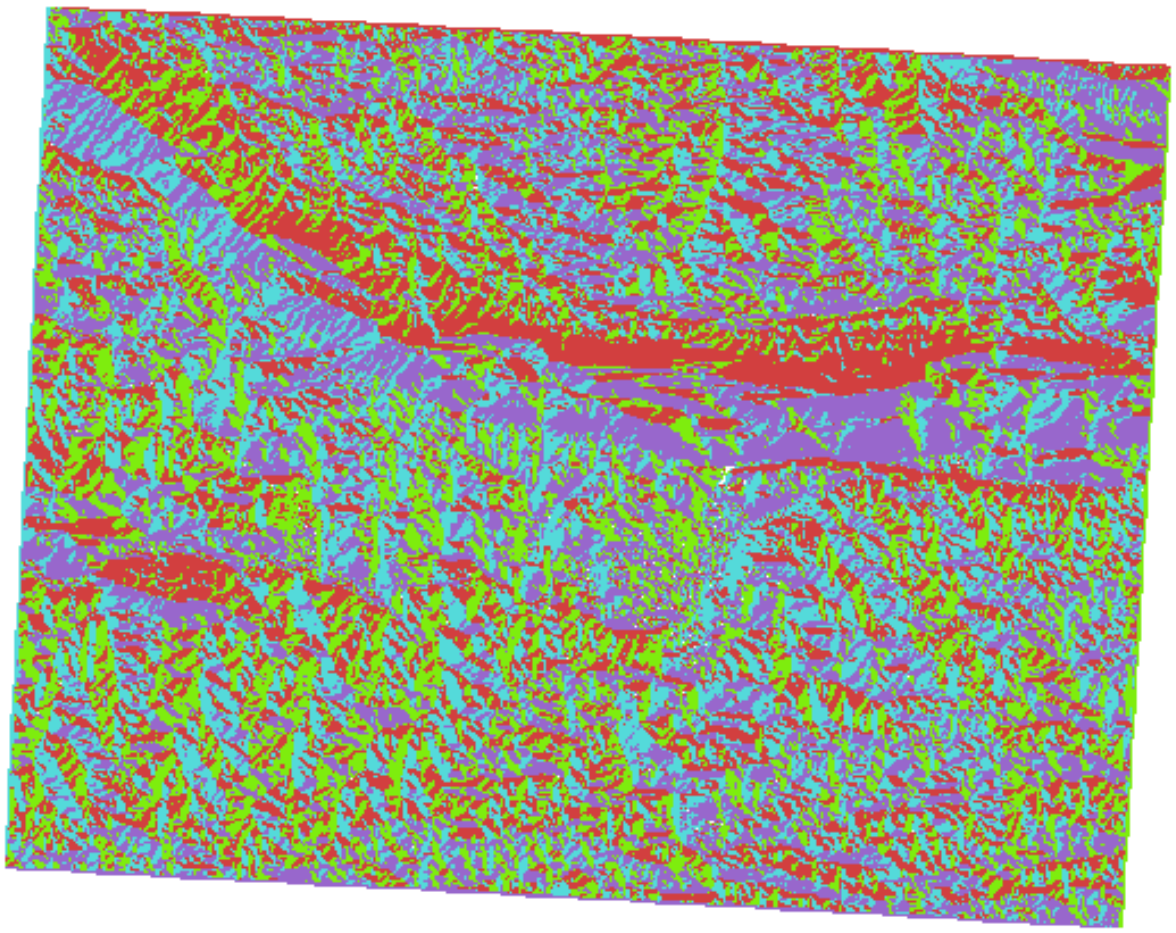
このレイヤをより良いスタイルに仕上げていきましょう。

1. **レイヤスタイル** パネルを開きます

2. **単バンドグレー** の代わりに、 **カテゴリ値パレット** を選びます
3. **分類** ボタンをクリックして、自動的に値を取得し、ランダムな色を割り当てます:



出力は以下のようになります（ランダムに生成されたため、異なる色を使用することも可能です）:



この再分類とレイヤに適用されたパレットスタイルにより、傾斜方位領域を即座に区別することができます。

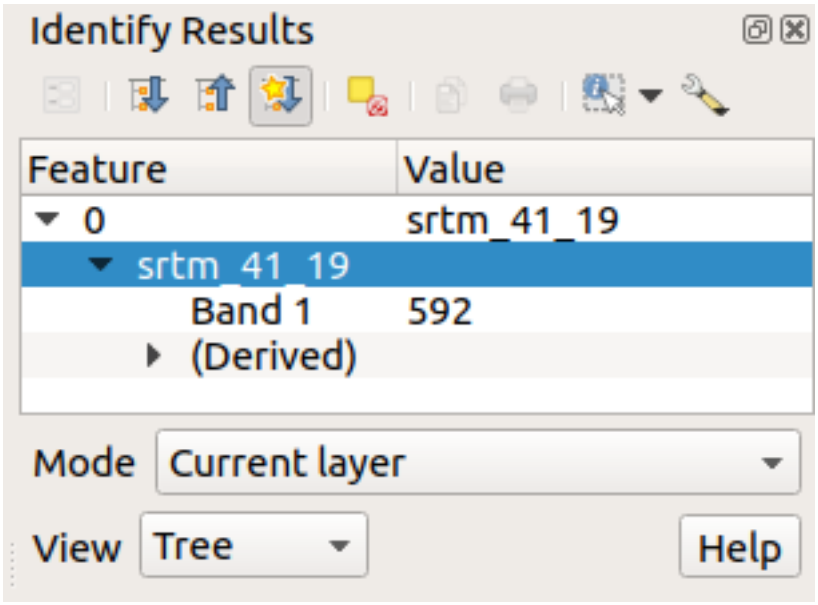
7.3.11 ★☆☆ （初級レベル） 理解しよう: ラスタを問い合わせる

ベクタレイヤとは異なり、ラスタレイヤは属性テーブルを持ちません。各ピクセルには 1 つまたは複数の数値が含まれます（シングルバンドまたはマルチバンドのラスタ）。

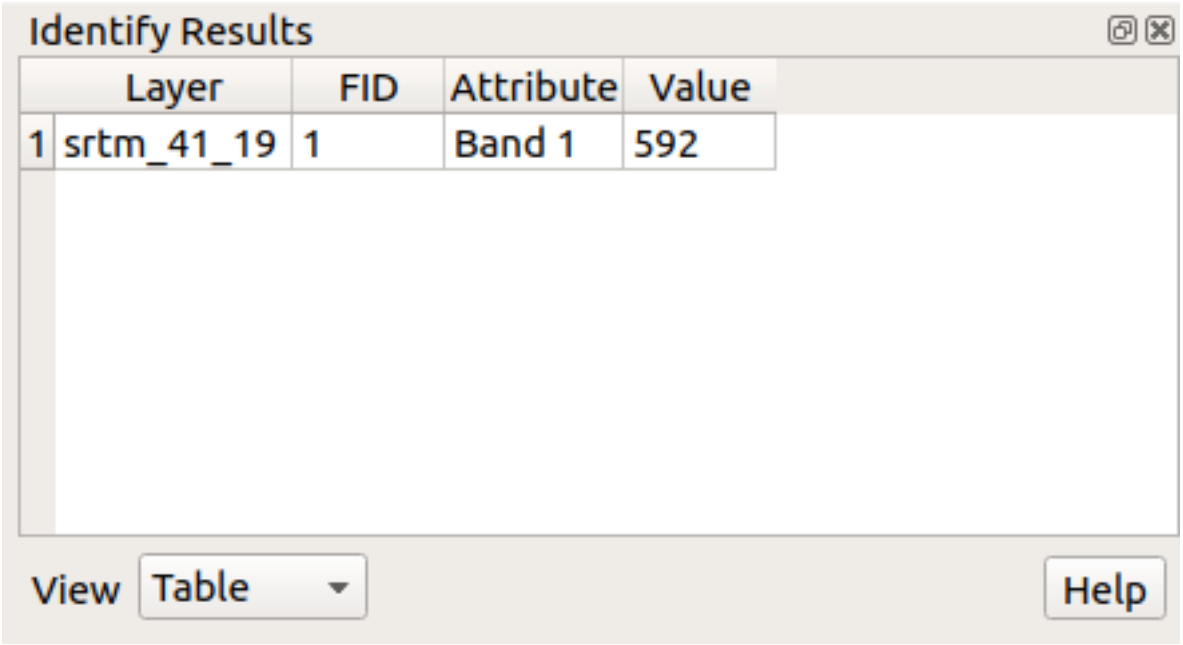
この演習で使用したすべてのラスタレイヤは、1 つのバンドだけで構成されています。レイヤによって、ピクセルの値は標高、傾斜方位、傾斜の値を表すことがあります。

ラスタレイヤに問い合わせてピクセルの値を取得するにはどうすればよいのでしょうか？  ボタンを使えばいいのです!

1. 属性ツールバーからツールを選択します。
2. Click on a random location of the `srtm_41_19` layer. *Identify Results* will appear with the value of the band at the clicked location:

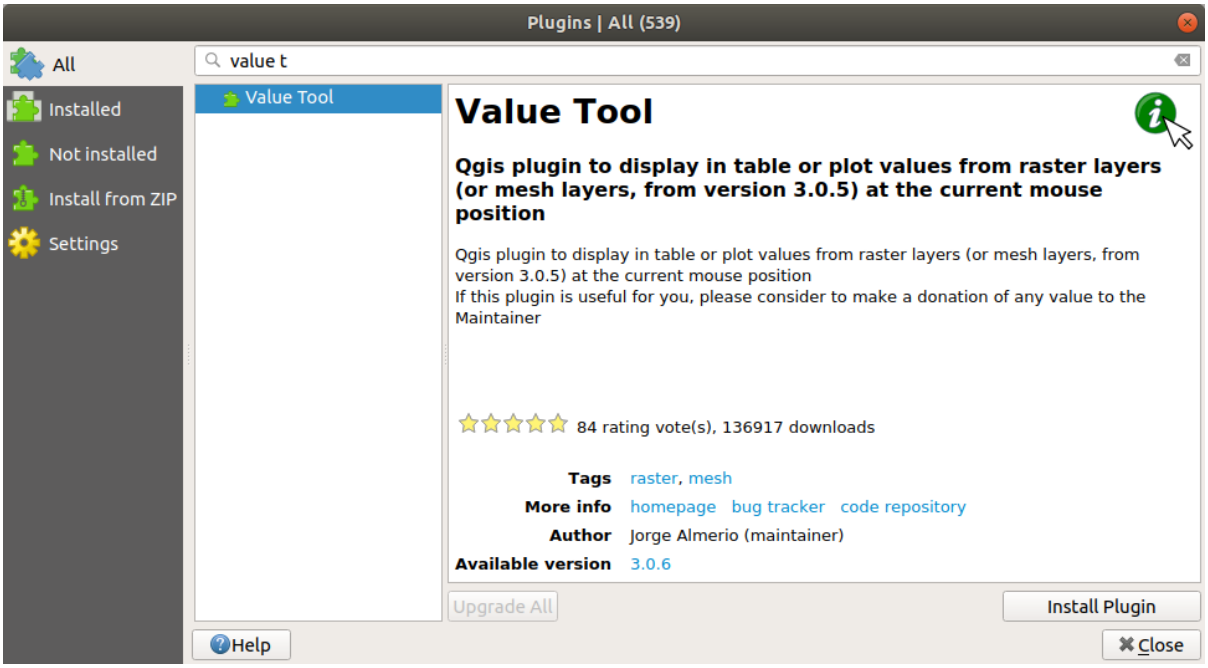


3. パネル下部にある **ビュー** メニューから **テーブル** を選択すると、 **地物情報** パネルの出力を、現在のツリー モードから テーブル に変更することができます:



ラスター値を取得するために各ピクセルをクリックすることは、しばらくすると煩わしくなるかもしれません。この問題を解決するために、*Value Tool* プラグインを使うことができます。

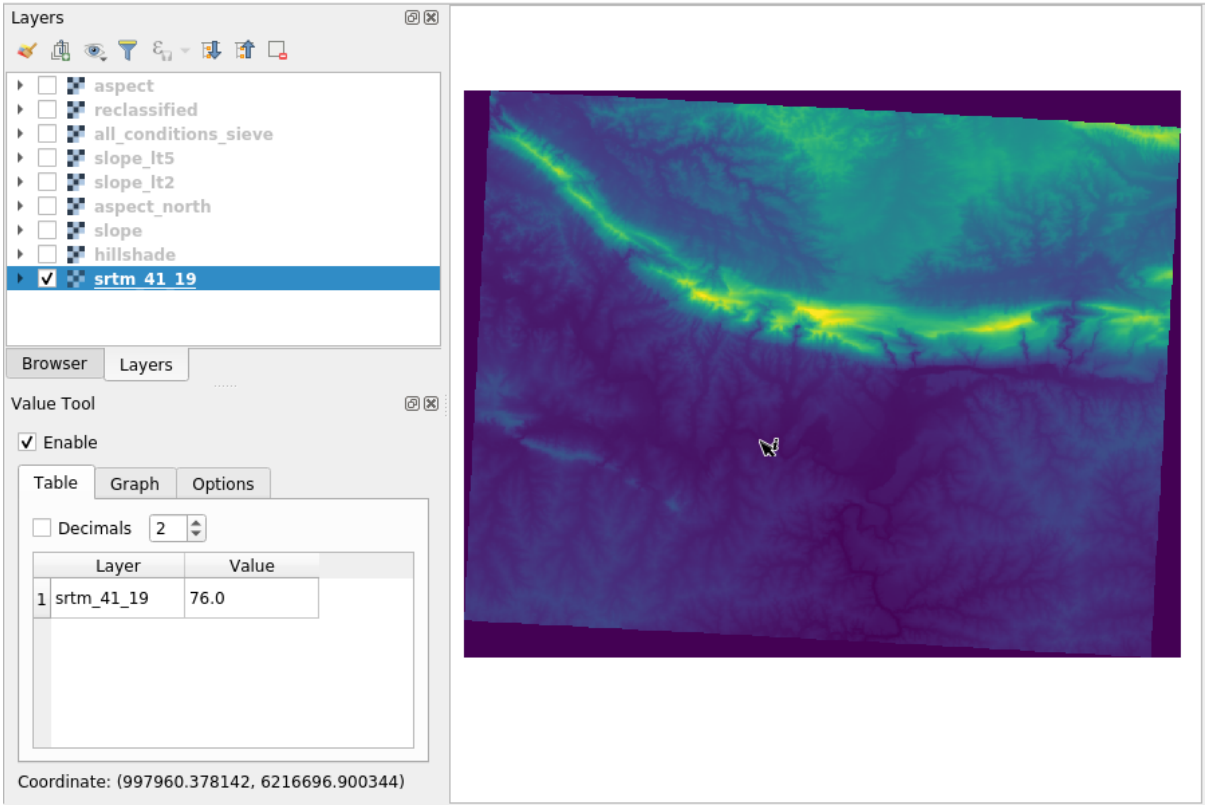
1. **プラグイン**　**プラグインの管理とインストール**... を選びます
2. **すべて** タブで、検索ボックスに `value t` と入力します
3. *Value Tool* プラグインを選択し、 **インストール** を押して、ダイアログを **閉じる** します。



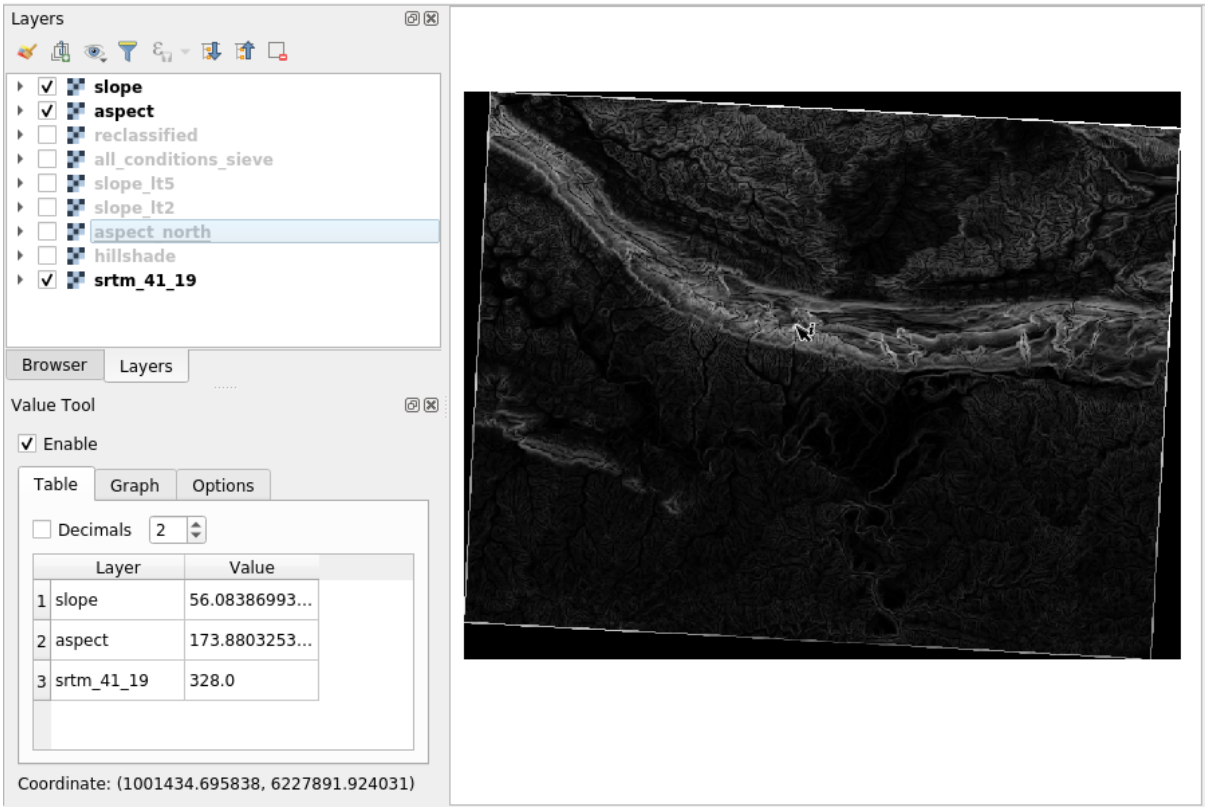
新しい *Value Tool* パネルが表示されます。

> **Tip**
>
> パネルを閉じた場合は、**ビュー　パネル**　*Value Tool* で有効にするか、ツールバーのアイコンをクリックすることで再び開くことができます。

4. プラグインを使用するには、*Enable* チェックボックスをチェックし、**レイヤ** パネルで `srtm_41_19` レイヤがアクティブ（チェック済み）であることを確認します。
5. カーソルを地図上に移動させると、ピクセルの値が表示されます。



6. But there is more. The Value Tool plugin allows you to query **all** the active raster layers in the *Layers* panel. Set the `aspect` and `slope` layers active again and hover the mouse on the map:



7.3.12 結論

DEM から様々な種類の分析結果を取り出す方法を見てきました。陰影起伏や傾斜、傾斜方位の計算をしました。またこれらの結果をさらに解析し結合するためにラスタ計算機の使用方法を見てきました。最後に、レイヤを再分類する方法と結果をクエリする方法を学びました。

7.3.13 次は?

２つの分析結果が得られました：潜在的に適した小地所を示すベクター分析の結果と潜在的に適した地形を示すラスター分析の結果です。この問題の最終的な結果に到達するためにどのようにこれらを組み合わせるか？それが次のレッスンのトピックです。次のモジュールで始まります。

第8章　モジュール: 分析の完了

分析は 2 部あります：ベクターの部とラスターの部。このモジュールでは、それらを組み合わせる方法を説明します。分析を完了して最終結果を提示します。

8.1 レッスン: ラスタからベクタへの変換

ラスタ形式とベクタ形式の間で変換できると、GIS の問題を解決するときに、またこれら二つの地理データの形式に特有の様々な解析方法を使用するときに、ラスタとベクタデータの両方を利用できます。GIS の問題を解決するためのデータソースと処理方法を検討する際に、より柔軟性が上がります。

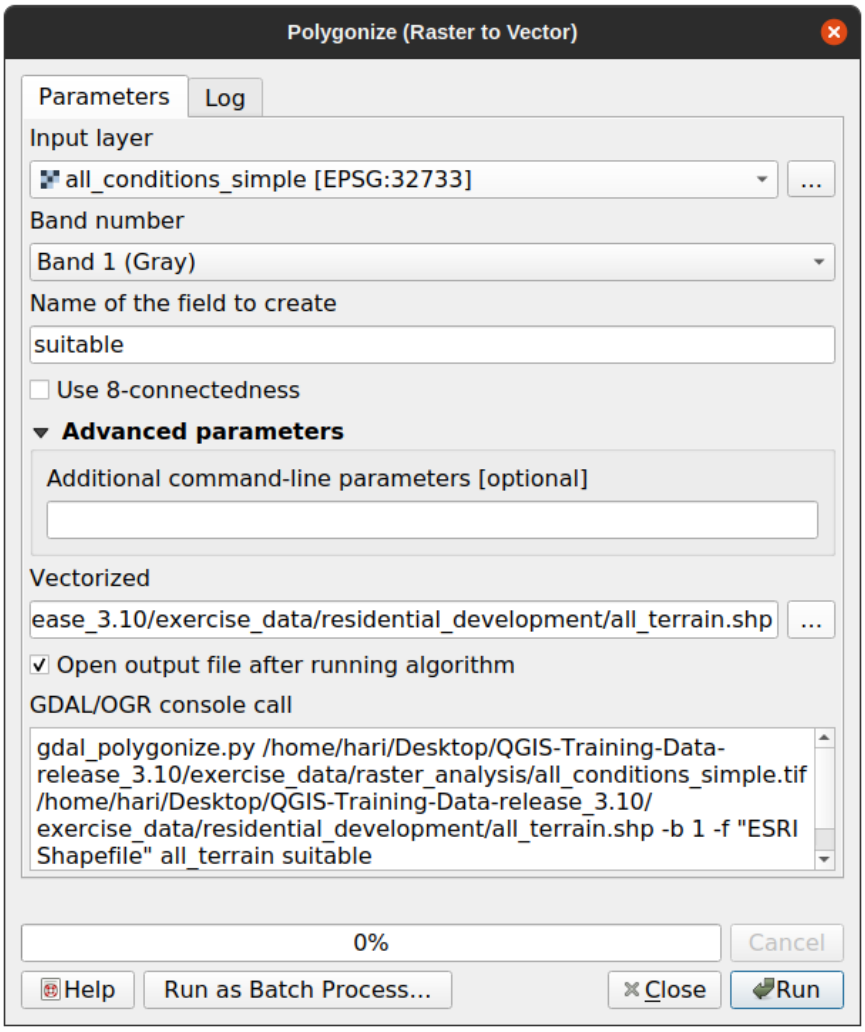
ラスタとベクタの解析を組み合わせるには、データのタイプを一方からもう一方に変換する必要があります。前のレッスンのラスタ結果をベクタに変換してみましょう。

このレッスンの目標： 解析を完了するためにラスタ結果をベクタにすること。

8.1.1 ★★☆ （中級レベル） 理解しよう: ラスタのベクタ化 ツール

Start with the map from the last module, `raster_analysis.qgs`. There you should have the `all_conditions_simple.tif` calculated during the previous exercises.

- **ラスタ　変換　ラスタのベクタ化** *(polygonize)* をクリックします。ツールのダイアログが表示されます。
- このように設定します:



- Change the field name (describing the values of the raster) to `suitable`.
- レイヤを `exercise_data/residential_development` の下に `all_terrain.shp` として保存します。

これで、ラスタのすべての値を含むベクタファイルができましたが、興味があるのは適切な領域だけです。つまり、 *suitable* の値が 1 であるポリゴンです。このレイヤーをより明確に可視化したい場合は、スタイルを変更することができます。

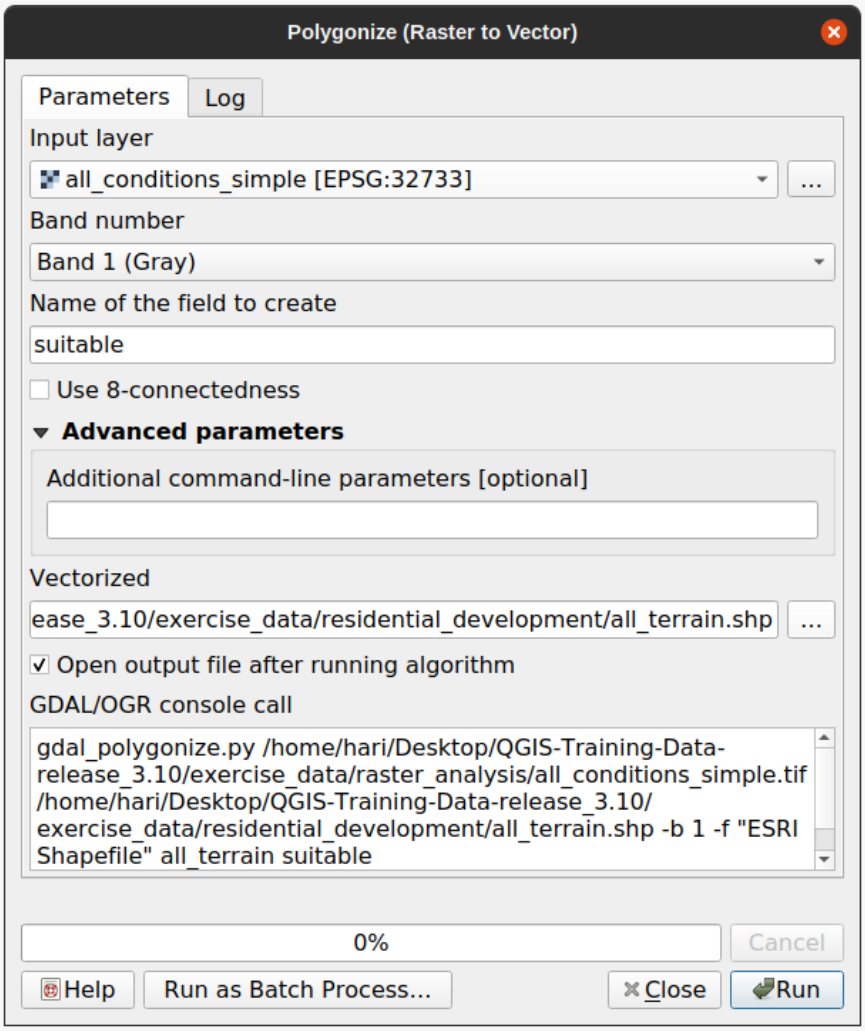
8.1.2 ★★☆ （中級レベル） 自分でやってみよう:

ベクタ分析のモジュールに戻って参照ください。

- *suitable* の値が 1 であるポリゴンのみを含むベクタファイルを新規に作成します。
- 新しいファイルを `exercise_data/residential_development/` の下に `suitable_terrain.shp` という名前で保存します。

答え

1. Open the *Query Builder* by right-clicking on the `all_terrain` layer in the *Layers* panel, and selecting the *Properties* *Source* tab.
2. `"suitable" = 1` というクエリを作ります。
3. *OK* をクリックすると、この条件を満たさないポリゴンがすべてフィルタリングされます。元のラスターの上に表示すると、領域が完全に重なっているはずです:

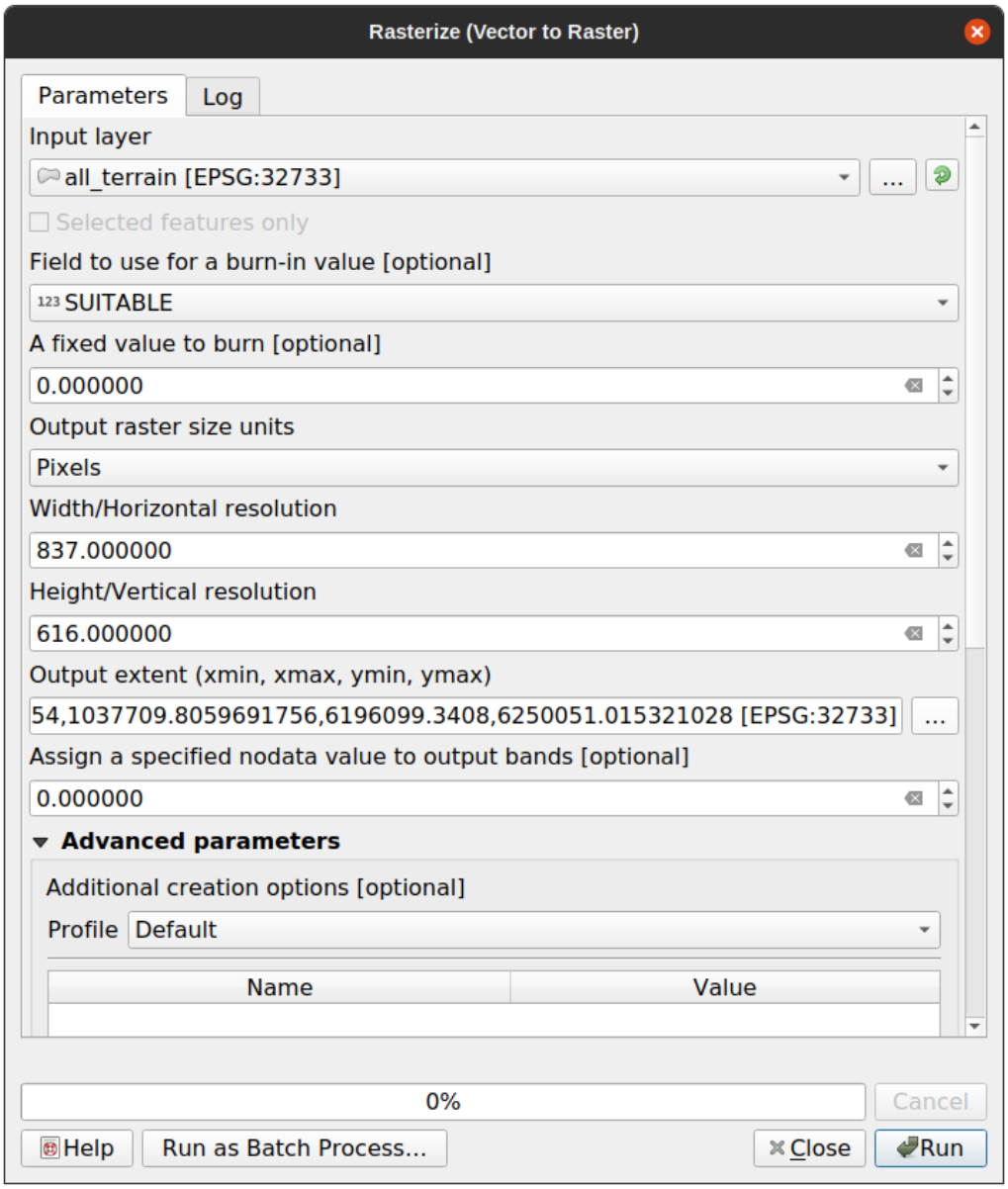


4. You can save this layer by right-clicking on the `all_terrain` layer in the *Layers* panel and choosing *Save As...*, then continue as per the instructions.

8.1.3 ★★☆ （中級レベル） 理解しよう: ベクタのラスタ化 ツール

現在の問題では不要ですが、上記で実行した変換とは逆の変換について知っておくと便利です。前の手順で作成した `suitable_terrain.shp` ベクタファイルをラスタに変換します。

- **ラスタ　変換　ベクタのラスタ化** *(rasterize)* をクリックしてツールを起動し、その後、下のスクリーンショットのようにそれを設定します:



- *Input layer* is `all_terrain`.
- フィールド名は *suitable* です。
- **出力ラスタサイズ単位** は **ピクセル** 。
- **幅** と **高さ** は、それぞれ `837` と `661`。
- Get the *Output extent* from the `all_terrain` layer.

- 出力ファイル *Rasterized* を `exercise_data/residential_development/raster_conversion.tif` に設定します。

> **注釈**
>
> 出力画像のサイズは、ここではベクタ化された元のラスタと同じになるように指定されています。画像の大きさを表示するには、そのメタデータを開きます（ **レイヤのプロパティ** 中の **メタデータ** タブ）。

- *OK* をクリックして変換処理を開始します。
- 完了したら、新たなラスタを元のものと比較することによって、その成功を測ります。両者は正確にピクセル単位で一致する必要があります。

8.1.4 結論

ラスタとベクタ形式の間で変換すると、データの適用可能性を広げることができ、データの劣化につながる必要はありません。

8.1.5 次は?

今はベクタ形式で利用可能な地形解析の結果がありますので、どの建物を住宅開発のために検討すべきかの問題を解決するためにそれらを使用できます。

8.2 レッスン: 分析を組み合わせる

ベクタ化されたラスタ解析の結果を使うと、適当な地形の上の建物のみを選択することができます。

このレッスンの目標: 適当な小地所を選び出すためにベクタ化された地形の結果を使う。

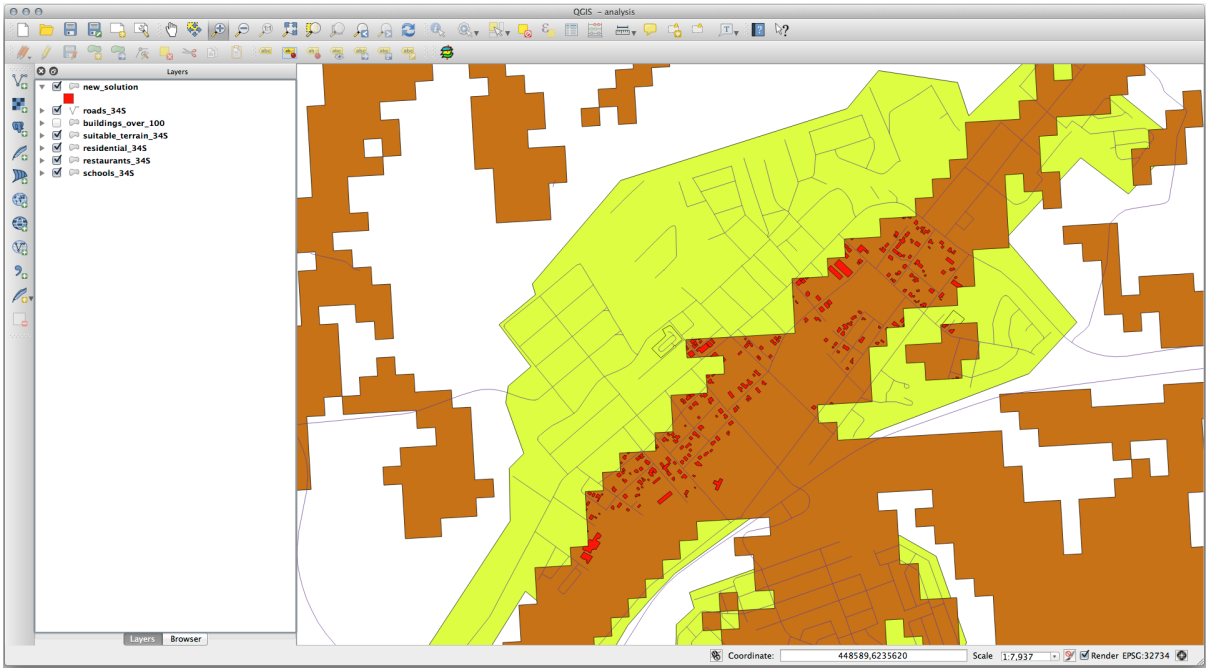
8.2.1 ★★☆ （中級レベル） 自分でやってみよう:

1. 現在の地図 (`raster_analysis.qgs`) を保存します。
2. 以前にベクタ解析中に作成した地図を開きます（ファイルを `analysis.qgs` として保存したはずです）。
3. **レイヤ** パネルで次のレイヤを有効にします:
   - `hillshade`,
   - `solution` (or `buildings_over_100`)
4. これらのレイヤに加え、以前作業したときに既に地図に読み込まれているはずの `suitable_terrain.shp` データセットも追加します。

5. もし、レイヤが足りない場合は、 `exercise_data/residential_development/` にあるはずです

6. Use the *Intersection* tool (*Vector* *Geoprocessing Tools*) to create a new vector layer called `new_solution.shp` which contains only those buildings which intersect the `suitable_terrain` layer.

あなたは解として特定の建物を示すレイヤを持っているはずです。例えば:



8.2.2 ★★☆ （中級レベル） やってみよう: 結果を検査する

Look at each of the buildings in your `new_solution` layer. Compare them with the `suitable_terrain` layer by changing the symbology for the `new_solution` layer so that it has outlines only. What do you notice about some of the buildings? Are they all suitable just because they intersect with the `suitable_terrain` layer? Why or why not? Which ones would you deem to be unsuitable?

答え

You may notice that some of the buildings in your `new_solution` layer have been "sliced" by the *Intersection* tool. This shows that only part of the building - and therefore only part of the property - lies on suitable terrain. We can therefore sensibly eliminate those buildings from our dataset.

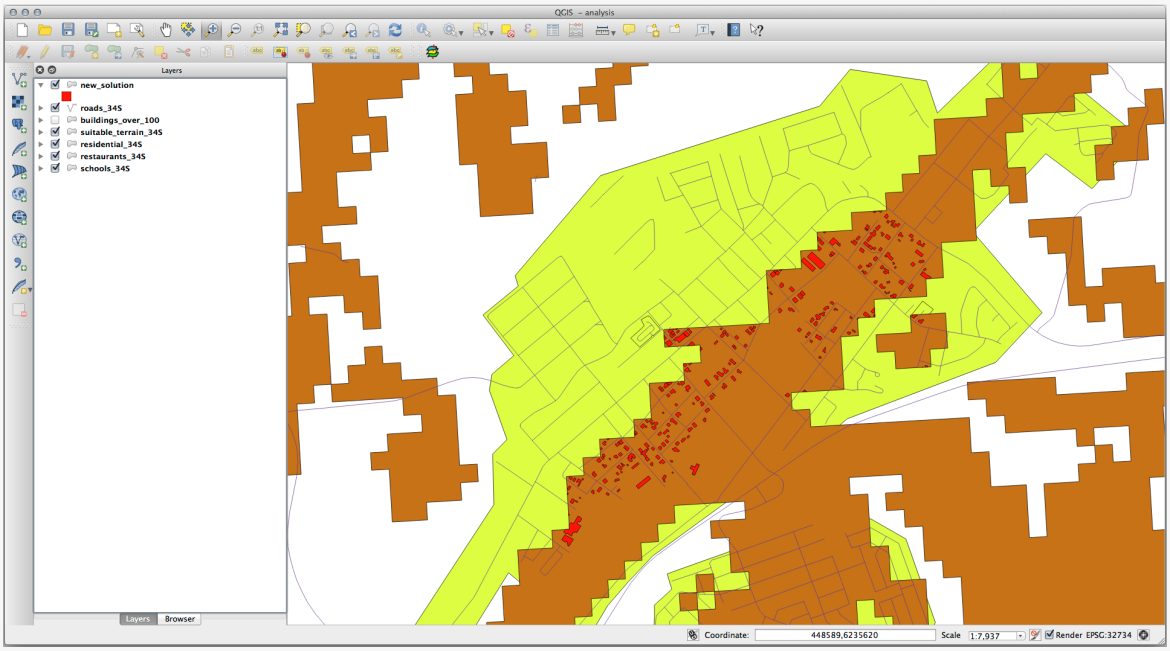
8.2.3 ★★☆ （中級レベル） やってみよう: 解析の改良

結果に含まれていた建物の中には本当には適していないものがありましたので解析を改良しましょう。

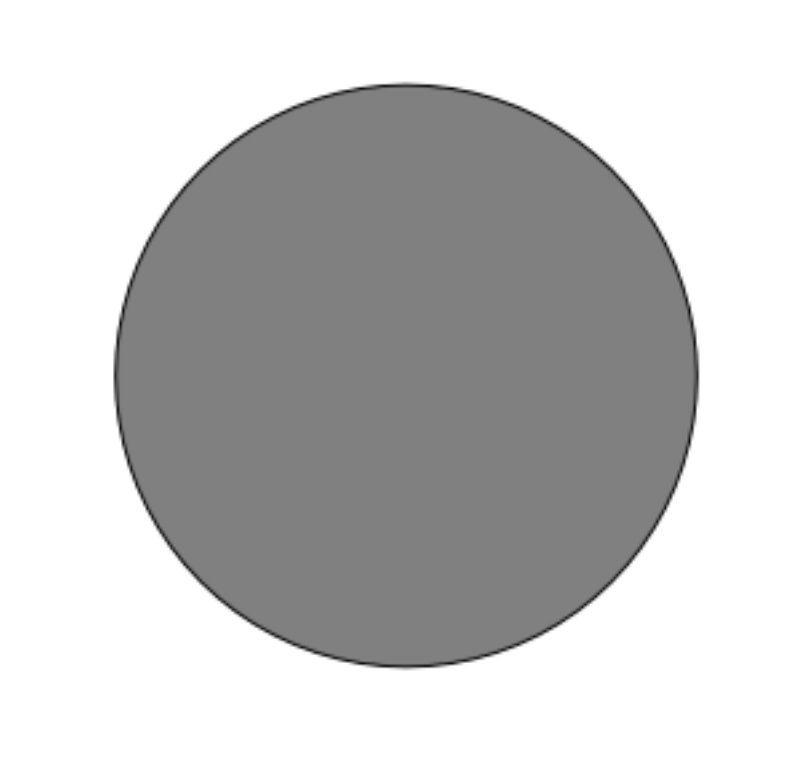
We want to ensure that our analysis returns only those buildings which fall entirely within the `suitable_terrain` layer. How would you achieve this? Use one or more Vector Analysis tools and remember that our buildings are all over 100m squared in size.

答え

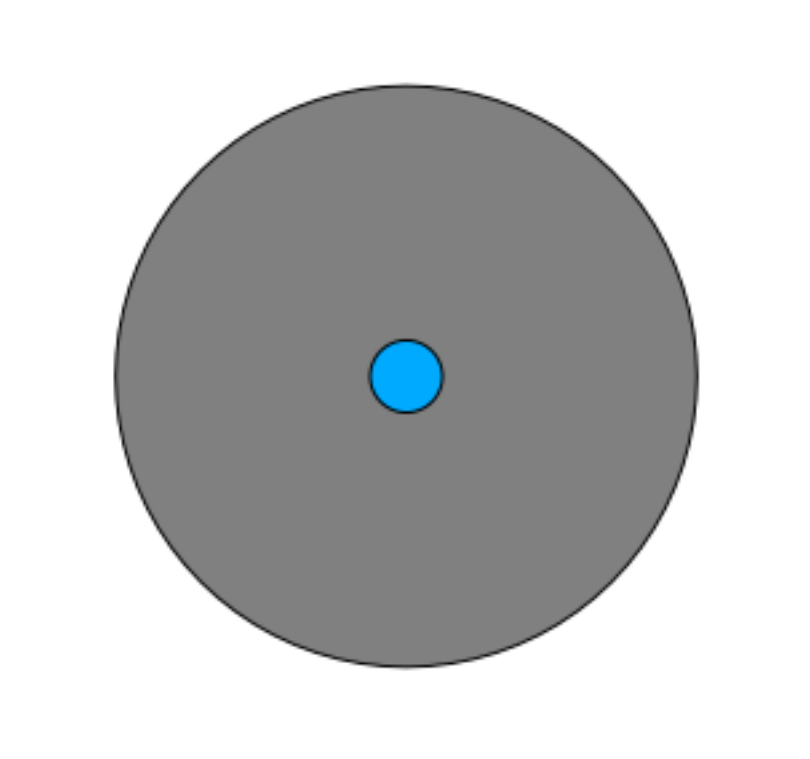
現時点では、このような解析ができるはずです:



四方 100m に渡って連続する円形の区域を考えてみましょう。



半径 100m 以上の場合、その大きさから（全方向から）100m を引くと、真ん中に一部が残ることになります。

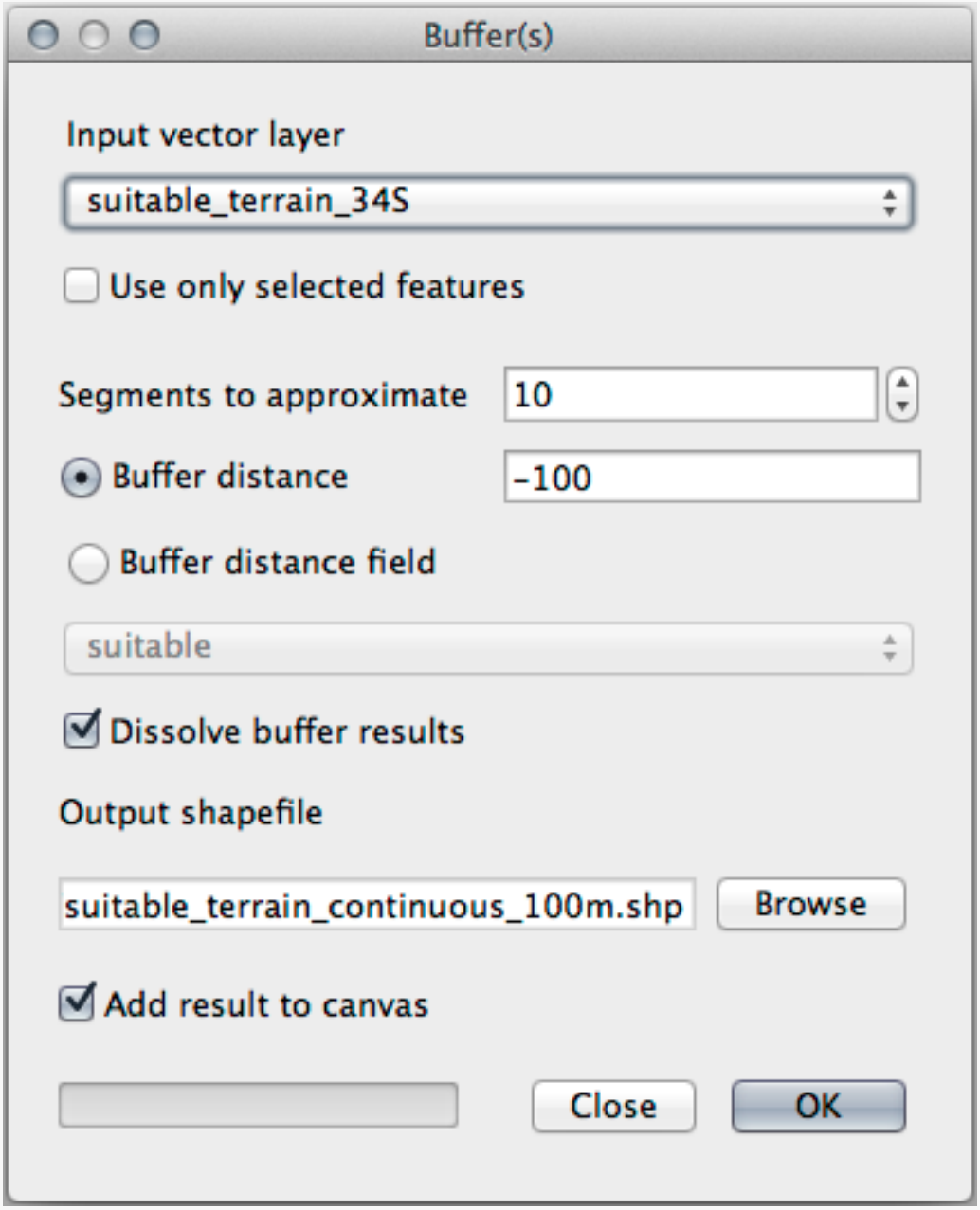


Therefore, you can run an *interior buffer* of 100 meters on your existing `suitable_terrain` vector layer. In the output of the buffer function, whatever remains of the original layer will represent areas where there is

suitable terrain for 100 meters beyond.

実際に試してみましょう:

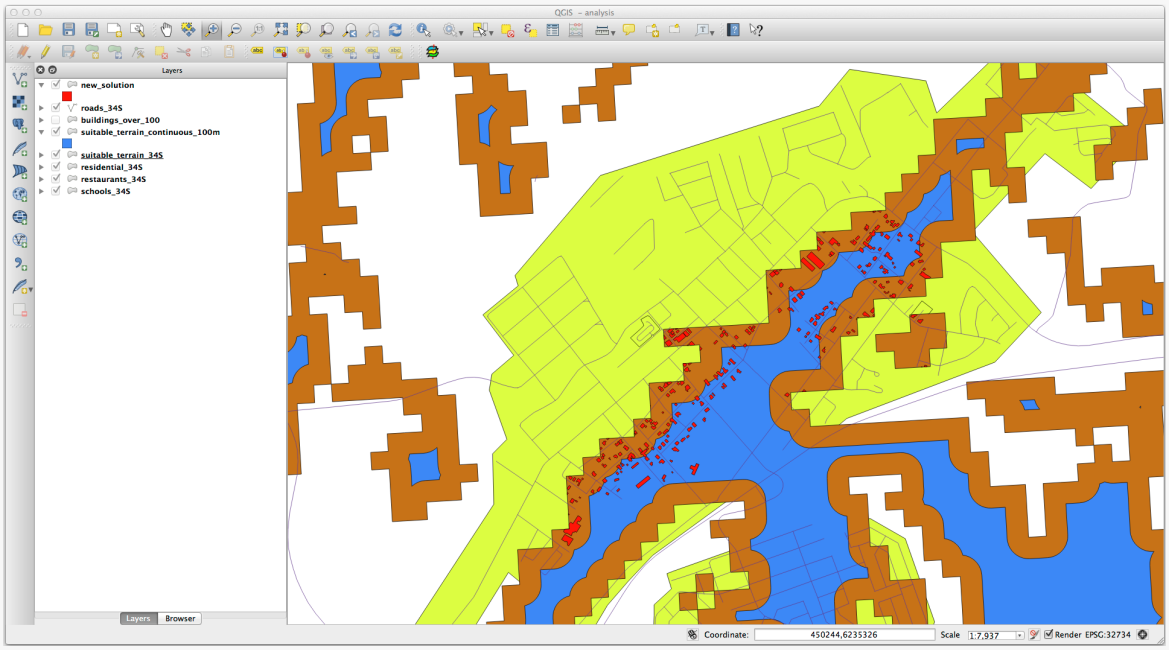
1. :menuselection:`ベクタ --> 空間演算ツール --> バッファ (buffer)`でバッファダイアログを表示します。
2. このように設定します:



3. Use the `suitable_terrain` layer with `10` segments and a buffer distance of `-100`. (The distance is automatically in meters because your map is using a projected CRS.)
4. 出力結果を `exercise_data/residential_development/` に `suitable_terrain_continuous100m.shp` として保存してください。

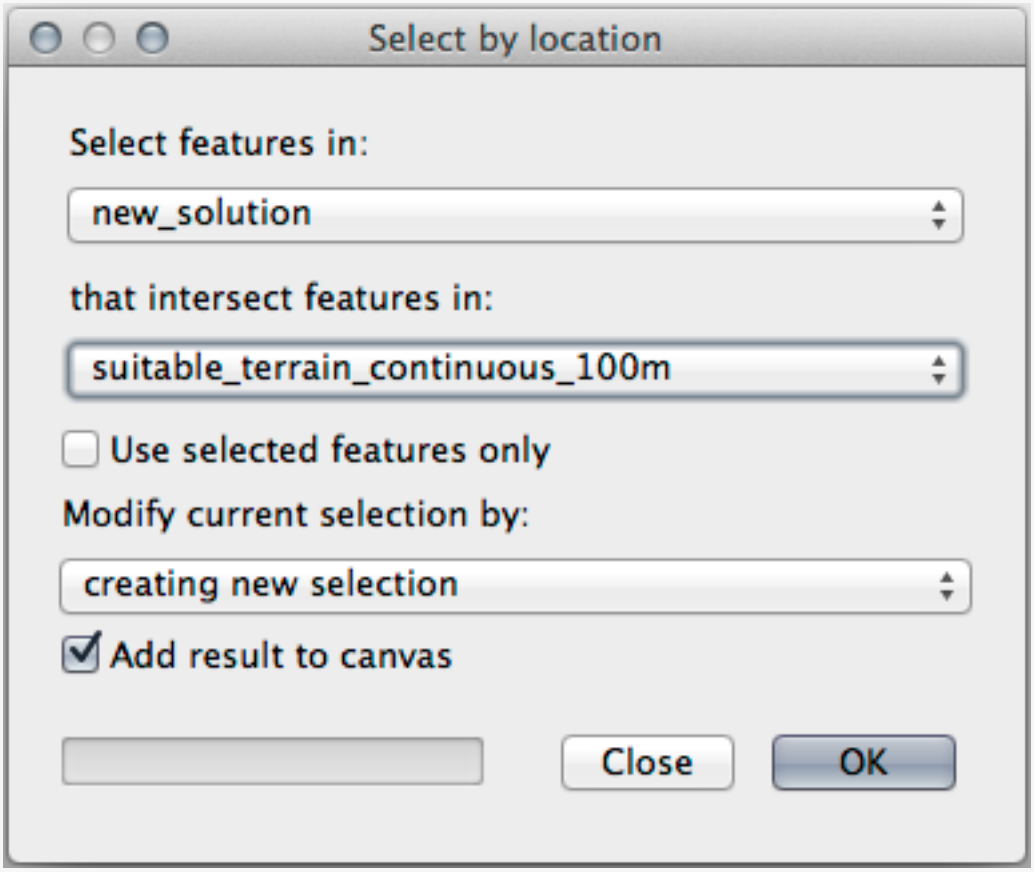
5. If necessary, move the new layer above your original `suitable_terrain` layer.

   結果はこのようになります:



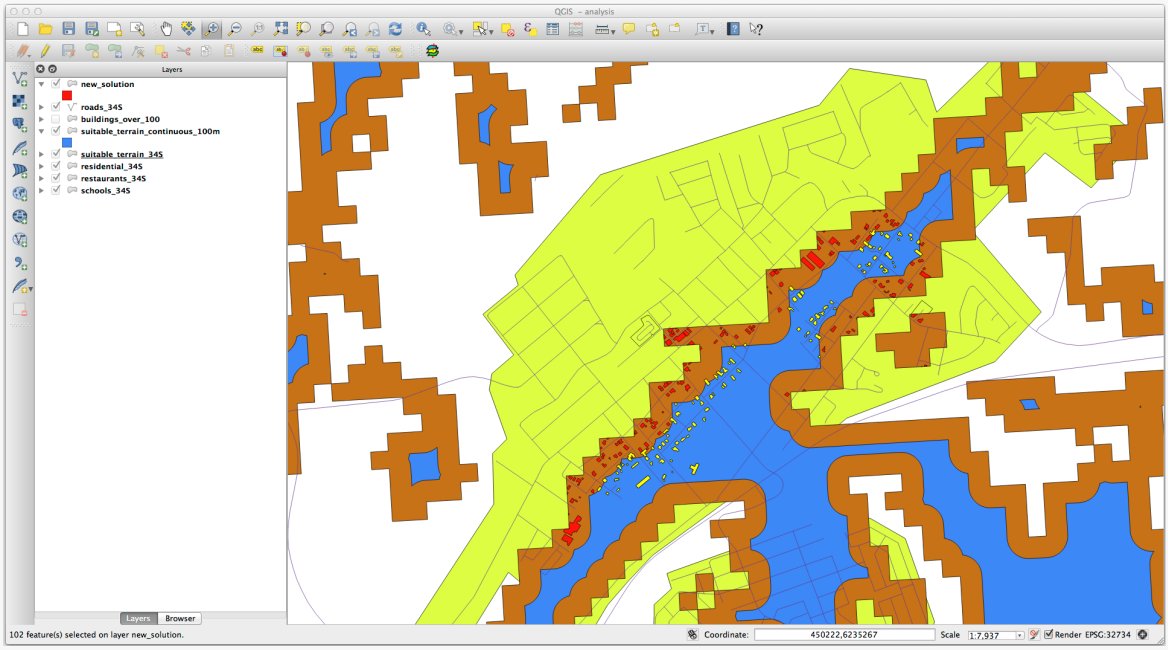
6. ここで、**場所による選択 ツール**（*:menuselection:* `**ベクタ --> 調査ツール --> 場所による選択**）を使います。

7. 次のように設定します:



8. Select features in `new_solution` that intersect features in `suitable_terrain_continuous100m.shp`.

   これがその結果です:



The yellow buildings are selected. Although some of the buildings fall partly outside the new

suitable_terrain_continuous100m layer, they lie well within the original suitable_terrain layer and therefore meet all of our requirements.

9. 選択部分を exercise_data/residential_development/ の下に final_answer.shp という名前で保存してください。

8.2.4 結論

これで当初の研究課題に答え、どの地所を開発するべきかに関する推薦の意見を（理由をもって、分析に支えられて）提示できます。

8.2.5 次は?

次は 2 番目の研究課題の一部としてこれらの結果を提示します。

8.3 課題

印刷レイアウトを使用して、解析結果を表す新しい地図を作成します。次のレイヤを含めてください:

- places (with labels),
- hillshade,
- solution (or new_solution),
- roads and
- either aerial_photos or DEM.

それに付随する短い説明文を書いて下さい。適当な建物への推薦を説明するだけではなく家の購入とその後の開発を考えるのに使用された基準を文章に含めなさい。

8.4 レッスン: 補足実習

このレッスンでは、QGIS を使った完全な GIS 解析を指導します。

注釈

このレッスンは、Linfiniti Consulting（南アフリカ）と Siddique Motala（ケープペニンシュラ工科大学）によって開発されました

8.4.1 問題文

あなたは、ケープ半島とその周辺で、珍しいフィンボス植物種に適した生息地を見つけるという任務を負っています。調査範囲は、北のメルクボスストラントと南のストランドの間のケープタウンとケープ半島をカバーしています。植物学者は、問題の種が好む次の条件を提供しました：

- 東向きの斜面で育ちます
- 15%から 60%の勾配の斜面で育ちます
- 年間降水量 > 1000 mm の地域で育ちます
- 人里から 250m 以上離れた場所でのみ発見されます
- 発生する植生の区域は面積で 6000 m²以上

大学の学生として、あなたは土地の 4 つの異なる適地でその植物を探すことに同意しました。あなたは、住んでいるケープタウン大学に最も近い区域の中からこの 4 つの適地を選びたいと思います。 GIS スキルを使って、どこを見に行くべきかを判断してください。

8.4.2 解決策の概要

この演習のデータは、 `exercise_data/more_analysis` フォルダにあります。

ケープタウン大学に最も近い 4 つの適地を見つけます。

解決策には次のようなものがあります：

1. DEM ラスタレイヤを解析して、東向き斜面や正しい勾配を持つ斜面を探し出す
2. 降雨量ラスタレイヤを解析して、正しい降雨量の地域を見つける
3. ゾーニングベクタレイヤを解析して、人里から離れた、適正な大きさの区域を探し出す

8.4.3 理解しよう: 地図を設定する

1. 画面の右下隅にある  現在の CRS ボタンをクリックします。表示されるダイアログの *CRS* タブで、[フィルター] ツールを使用して「33S」を検索します。エントリ *WGS 84 / UTM* **ゾーン** *33S* （EPSG コード 32733 ）を選択します。
2. *OK* をクリックします

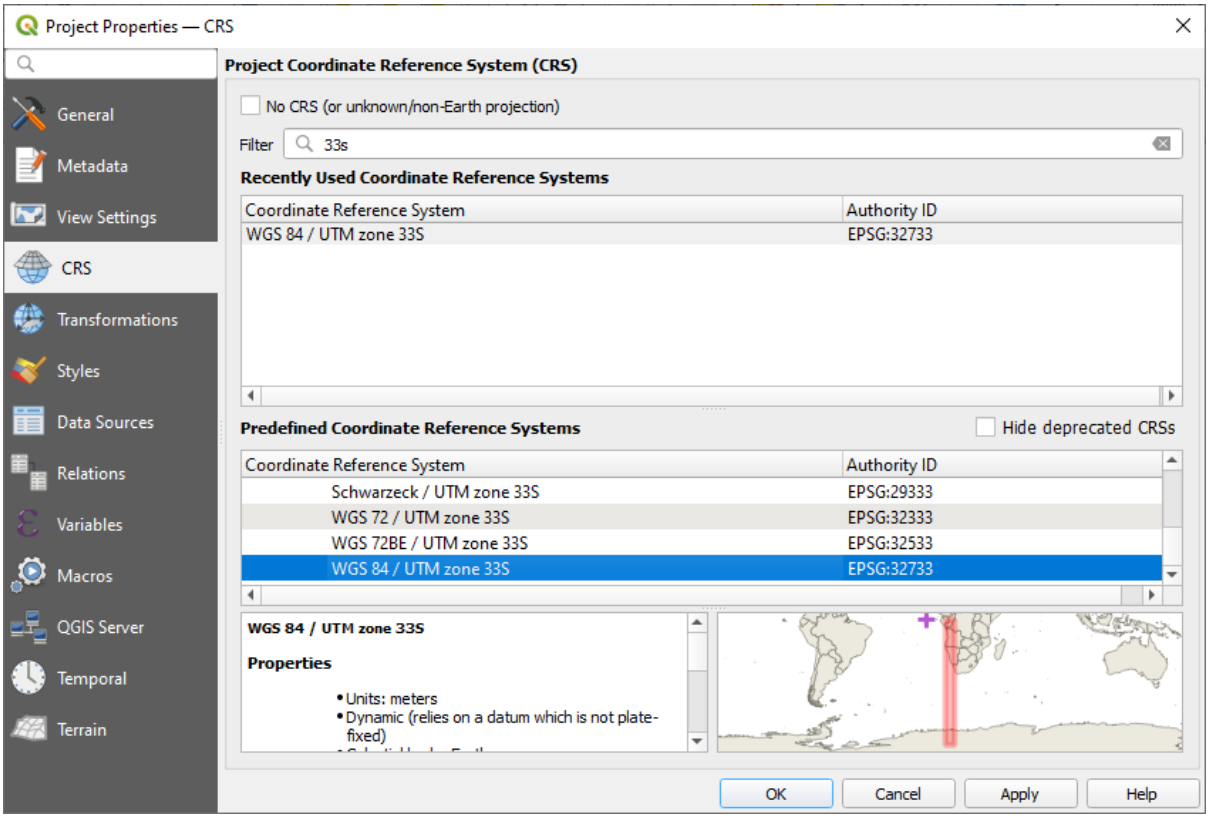


図 8.1: CRS を設定する

3. プロジェクトファイルを保存するには、ツールバーの プロジェクトを保存 ボタンをクリックするか、**プロジェクト　名前を付けて保存**... メニュー項目を使用します。

   これを `Rasterprac` という新しいディレクトリに保存します。これは、コンピュータのどこかに作る必要があります。作成したレイヤもこのディレクトリに保存します。プロジェクトを `your_name_fynbos.qgs` として保存します。

8.4.4 地図へデータを読み込む

データを処理するためには、必要なレイヤ（街路名、ゾーン、雨量、DEM、地区）をマップキャンバスに読み込む必要があります。

ベクタについては...

1. **データソースマネージャツールバー** の データソースマネージャーを開く ボタンをクリックし、表示されるダイアログの **ベクタ** タブを選ぶか、または **レイヤ　レイヤを追加　ベクタレイヤを追加**... メニュー項目を使用します
2. **ファイル** が選択されていることを確認します
3. ... ボタンをクリックして、ベクタデータセットをブラウズします

4. 表示されたダイアログで、exercise_data/more_analysis/Streets ディレクトリを開いてください
5. ファイル Street_Names_UTM33S.shp を選択します
6. **開く** をクリックします。

   ダイアログが閉じ、元のダイアログが表示されます。その際、 **ベクタデータセット** の隣のテキストフィールドにファイルパスが指定されています。これにより、正しいファイルが選択されていることを確認することができます。また、このフィールドにファイルパスを手動で入力することもできます。
7. **追加** をクリックします。ベクタレイヤがマップに読み込まれます。色は自動的に割り当てられます。色は後で変更します。
8. レイヤの名前を Streets に変更します
   1. **レイヤ** パネル（デフォルトでは、画面の左側に沿ったペイン）で、そのアイコンを右クリックします
   2. 表示されたダイアログで **レイヤの名前を変更** をクリックして名前を変更し、完了したら Enter キーを押してください
9. ベクタの追加を繰り返しますが、今回は Zoning ディレクトリにある Generalised_Zoning_Dissolve_UTM33S.shp ファイルを選びます。
10. その名前を Zoning に変更します。
11. ベクタレイヤ admin_boundaries/Western_Cape_UTM33S.shp も地図に読み込みます。
12. その名前を Districts に変更します。

ラスタについては...

1. 

ボタンをクリックし、表示されたダイアログで  **ラスタ** タブを有効にするか、 **レイヤ** **レイヤを追加**  **ラスタレイヤを追加**... メニューアイテムを使います
2.  **ファイル** が選択されていることを確認します
3. 該当するファイルに移動して選択し、 **開く** をクリックします
4. 次の 2 つのラスタファイル、 DEM/SRTM.tif と rainfall/reprojected/rainfall.tif のそれぞれについて、この操作を行います
5. SRTM ラスタを DEM に、降雨量ラスタを Rainfall に名前を変更します（頭文字は大文字）

8.4.5 レイヤ順序を変更する

レイヤ パネルでレイヤをクリック、上下にドラッグして、地図上に表示される順序を変え、できるだけ多くのレイヤが見えるようにします。

これですべてのデータが読み込まれ、正しく表示されるようになったので、解析を開始することができます。最初にクリッピング操作を行うとよいでしょう。これは、どうせ使わない部分の値を計算するために処理能力を浪費しないためです。

8.4.6 正しい地区の検索

前述の調査区域の関係で、地区を以下のものに限定する必要があります:

- `Bellville`
- `Cape`
- `Goodwood`
- `Kuils River`
- `Mitchells Plain`
- `Simon Town`
- `Wynberg`

1. **レイヤ** パネルで `Districts` レイヤを右クリックします。
2. 表示されたメニューから、 **フィルタ**... メニュー項目を選択します。**クエリビルダ** ダイアログが表示されます。
3. ここで、候補となる地区のみを選択するクエリを作成します:
    1. **属性** リストで `NAME_2` フィールドをダブルクリックして、下の **プロバイダ特有のフィルタ式** テキストフィールドに表示させます
    2. *IN* ボタンをクリックすると、SQL クエリに追加されます
    3. 始め丸括弧を入力します
    4. （現在は空の） **値** リストの下にある **すべて** ボタンをクリックします。

       少しすると、選択されたフィールド (`NAME_2`) の値で **値** リストが構成されます。
    5. **値** リストの中の値 `Bellville` をダブルクリックして、SQL クエリに追加します。
    6. コンマを追加し、`Cape` 地区を追加するためにダブルクリックします
    7. 残りの地区についても、前のステップを繰り返してください
    8. 括弧を閉じます

図 8.2: クエリビルダ

最終的なクエリは次のようになります（括弧内の地区の順序は重要ではありません）：

```
"NAME_2" in ('Bellville', 'Cape', 'Goodwood', 'Kuils River',
             'Mitchells Plain', 'Simon Town', 'Wynberg')
```

注釈

また、OR 演算子も使うことができます。クエリは次のようになります:

```
"NAME_2" = 'Bellville' OR "NAME_2" = 'Cape' OR
"NAME_2" = 'Goodwood' OR "NAME_2" = 'Kuils River' OR
"NAME_2" = 'Mitchells Plain' OR "NAME_2" = 'Simon Town' OR
"NAME_2" = 'Wynberg'
```

1. *OK* を二回クリックします。

   地図に表示される地区は、上記のリストに記載されているものに限定されるようになりました。

8.4.7 ラスタのクリップ

関心領域を手に入れたので、この区域へラスタを切り抜くことができます。

1. メニュー **ラスタ** **抽出** **マスクレイヤで切り抜く**... を選択して、クリッピングダイアログを表示させます
2. **入力レイヤ** ドロップダウンリストで、DEM レイヤを選びます
3. **マスクレイヤ** ドロップダウンリストから、Districts レイヤを選びます
4. スクロールダウンして、 **出力ファイル** テキストフィールドに出力先を指定します。 ... ボタンをクリックして **ファイルに保存**... を選びます
   1. Rasterprac ディレクトリに移動します
   2. ファイル名に DEM_clipped.tif を入力します
   3. 保存します
5. **アルゴリズムの終了後に出力ファイルを開く** がチェックされていることを確認します
6. **実行** をクリックします

   クリッピング操作が完了したら、クリッピング領域を再利用するために **マスクレイヤで切り抜く** ダイアログを開いたままにします
7. **入力レイヤ** ドロップダウンリストから Rainfall ラスタレイヤを選択し、 Rainfall_clipped.tif という名前で出力を保存します
8. 他のオプションは変更しないでください。すべてそのままにして、**実行** をクリックします。
9. 2 回目のクリッピング操作が完了したら、 **マスクレイヤで切り抜く** ダイアログを閉じます

10. 地図を保存します

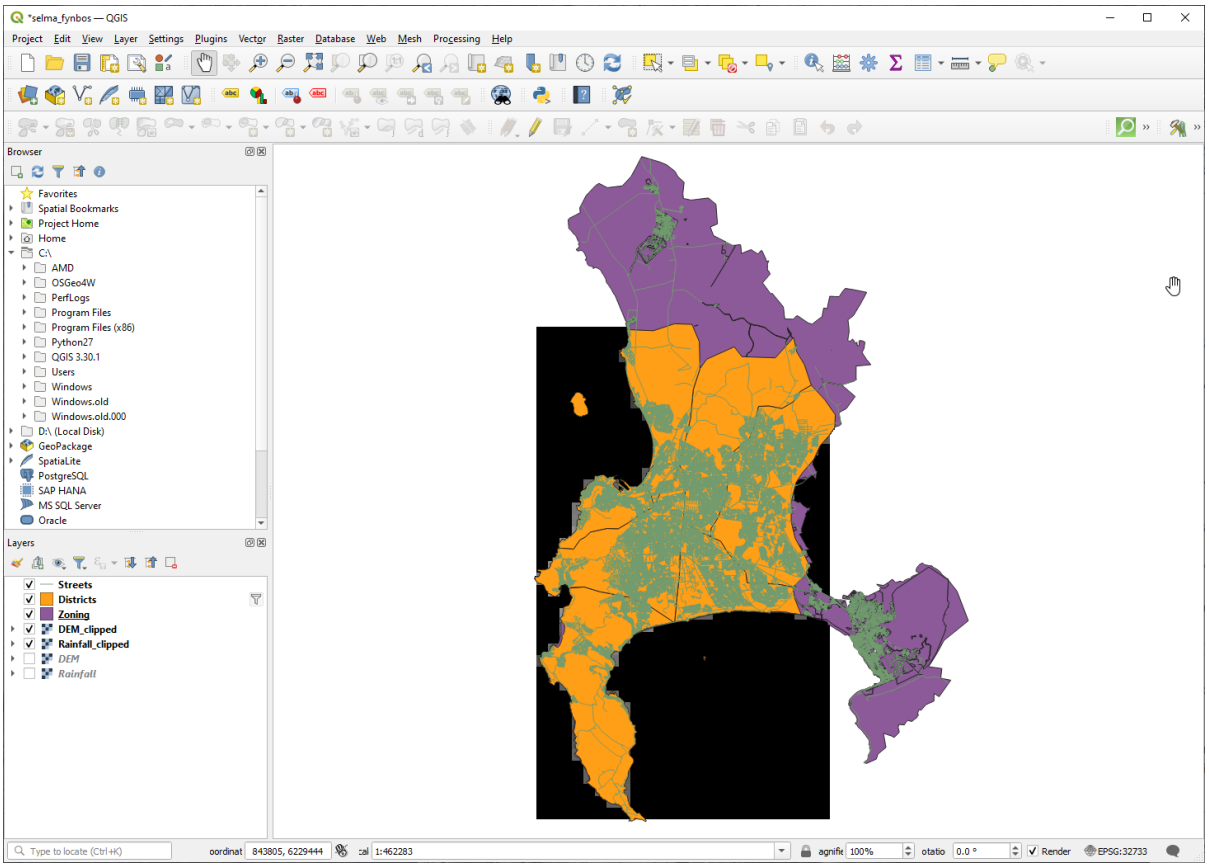


図 8.3: フィルタされたベクタ、切り抜かれたラスタ、順序変更されたレイヤを持ったマップビュー

ラスタを揃える

解析のためには、ラスタが同じ CRS を持ち、位置が揃っていることが必要です。

まず、雨量データの解像度を 30m（ピクセルサイズ）に変更します:

1. **レイヤ** パネルで、`Rainfall_clipped` がアクティブレイヤであることを確認します（つまり、クリックされてハイライトされています）
2. **ラスタ　投影法　再投影** *(warp)...* メニューをクリックし、 **再投影** *(warp)* ダイアログを開きます
3. **リサンプリング法** の下にあるドロップダウンメニューから *Bilinear (2x2 kernel)* を選択します
4. **変換先** *CSR* **の単位での解像度** を `30` に設定します
5. **再投影したラスタファイル** までスクロールし、出力を `rainfall/reprojected` ディレクトリに `Rainfall30.tif` という名前で保存します。
6. **アルゴリズムの終了後に出力ファイルを開く** がチェックされていることを確認します

図 8.4: 再投影 (Warp) Rainfall_clipped

それでは DEM を位置合わせします:

1. **レイヤ** パネルで、`DEM_clipped` をアクティブレイヤにします（つまり、クリックしてハイライトさせます）
2. **ラスタ** **投影法** **再投影** *(warp)...* メニューをクリックし、 **再投影** *(warp)* ダイアログを開きます
3. **ラスタの** *CRS* の下にあるドロップダウンメニューから **プロジェクト** *CRS: EPSG:32733 - WGS 84 / UTM zone 33S* を選択します
4. **リサンプリング法** の下にあるドロップダウンメニューから *Bilinear (2x2 kernel)* を選択します
5. **変換先** *CSR* **の単位での解像度** を `30` に設定します
6. **出力ファイルの矩形範囲** までスクロールダウンします。テキストボックスの右側のボタンを使って、**レイヤから計算** *Rainfall30* を選択します。
7. *Reprojected* までスクロールして、出力を `DEM/reprojected` ディレクトリに `DEM30.tif` という名前で保存します。
8. **アルゴリズムの終了後に出力ファイルを開く** がチェックされていることを確認します

何が起こっているのかを適切に確認するために、レイヤーのためのシンボルを変更する必要があります。

8.4.8 ベクタレイヤのシンボロジを変更する

1. In the *Layers* panel, right-click on the `Streets` layer
2. 表示されるメニューから **プロパティ** を選びます
3. 表示されたダイアログで **シンボロジ** タブに切り替えます
4. トップウィジェットにある **ライン** エントリをクリックします
5. 下のリストから記号を選択するか、新しい記号を設定します（色、透明度、...）
6. Click *OK* to close the *Layer Properties* dialog. This will change the rendering of the `Streets` layer.
7. Follow a similar process for the `Zoning` layer and choose an appropriate color for it

8.4.9 ラスタレイヤのシンボロジを変更する

ラスタレイヤのシンボロジはいくらか異なります。

1. Open the *Properties* dialog for the `Rainfall30` raster layer
2. **シンボロジ** タブに切り替えます。このダイアログは、ベクタレイヤに使用されるバージョンと大きく異なっていることにお気づきでしょう。
3. **最小／最大値設定** を展開します
4. ボタン **平均 +/- 標準偏差** が選択されていることを確認します
5. 関連するボックスの値が `2.00` になっていることを確認します

6. **コントラスト** に **最小最大値に引き伸ばす** と表示されていることを確認します
7. **グラデーション** は、**白から黒** に変更します
8. *OK* をクリックします



図 8.5: ラスタのシンボロジ

`Rainfall30` ラスタが表示されていれば、色が変わり、各ピクセルの異なる輝度値を見ることができるはずです

9. この作業を `DEM30` レイヤーに対して繰り返します。ただし、引き伸ばしに使用する標準偏差は `4.00` に設定します

8.4.10 地図をクリーンアップします

1. オリジナルの `Rainfall` と `DEM` レイヤ、および `Rainfall_clipped` と `DEM_clipped` を **レイヤ** パネルから削除します:
   - これらのレイヤ上で右クリックし、 **削除** を選択します。

   > **注釈**
   >
   > これは、記憶装置からデータを削除しません。地図から外すだけです。

2. 地図を保存します
3. **レイヤ** パネル中のベクタレイヤを、横にあるボックスをオフにして非表示にできます 。これによって地図のレンダリングが速くなり、時間の節約になります。

8.4.11 陰影図の作成

陰影図を作成するには、この目的のために書かれたアルゴリズムを使う必要があります。

1. **レイヤ** パネルで、`DEM30` がアクティブレイヤであることを確認します（つまり、クリックしてハイライトにします）
2. メニューから **ラスタ** **解析** **陰影図** *(hillshade)...* を選択すると、 **陰影図** *(hillshade)* ダイアログが表示されます
3. **陰影図** *(hillshade)* までスクロールし、 `Rasterprac` ディレクトリに `hillshade.tif` として出力を保存します
4. **アルゴリズムの終了後に出力ファイルを開く** がチェックされていることを確認します
5. **実行** をクリックします

6. 処理が完了するのを待ちます。

図 8.6: ラスタ解析 陰影図

新しい `hillshade` レイヤが **レイヤ** パネルに表示されました。

1. **レイヤ** パネルで `hillshade` レイヤーを右クリックして、 **プロパティ** ダイアログを表示します
2. **透過性** タブをクリックし、 **グローバルな不透明度** スライダを `20%` に設定します
3. *OK* をクリックします
4. 透明な陰影図が切り取られた DEM の上に重なっているときの効果に注意してください。効果を確認するため、レイヤの順番を変えるか、`Rainfall30` のレイヤをクリックしてオフにする必要があるかもしれません。

8.4.12 傾斜

1. **ラスタ** **解析** **傾斜** *(slope)...* メニュー項目をクリックし、 **傾斜** *(slope)* アルゴリズムのダイアログを開きます
2. `DEM30` を **入力レイヤ** に選択します
3. ☑ **傾斜の単位はパーセント（デフォルトは度）** をチェックします。傾斜は異なる単位（パーセントまたは度）で表現することができます。我々の基準では、対象の植物は 15%から 60%の勾配の斜面に生育するとされています。そこで、傾斜データがパーセントで表現されていることを確認する必要があります。
4. 出力のための適切なファイル名と場所を指定します。
5. ☑ **アルゴリズムの終了後に出力ファイルを開く** がチェックされていることを確認します
6. **実行** をクリックします



図 8.7: ラスタ解析 傾斜

傾斜画像が計算され、地図に追加されました。いつも通りグレースケールでレンダリングされています。もっとカラフルなシンボロジに変更します:

1. レイヤの **プロパティ** ダイアログを開きます（通常通り、レイヤの右クリックメニューから行います）
2. **シンボロジ** タブをクリックします
3. **単バンドグレー**（ **レンダリングタイプ** ドロップダウンメニュー）と書かれている箇所を **単バンド疑似カラー** に変更します
4. **最小/最大値設定** の **平均** +/- **標準偏差** *x* に `2.0` を選びます
5. 適当な **カラーランプ** を選びます

6. **実行** をクリックします

8.4.13 やってみよう: 傾斜方位

傾斜を計算するのと同じ方法で、 **ラスタ** **解析** メニューから **傾斜方位**... を選択します。

定期的にプロジェクトを保存することを忘れないでください。

8.4.14 ラスターを再分類する

1. **ラスタ** **ラスタ計算機**... を選択します
2. **出力レイヤ** の場所として Rasterprac ディレクトリを指定し（ ... ボタンをクリックします)、slope15_60.tif というファイル名で保存します
3. **結果をプロジェクトに追加する** のボックスが選択されていることを確認します。

   In the *Raster bands* list on the left, you will see all the raster layers in your *Layers* panel. If your Slope layer is called slope, it will be listed as slope@1. Indicating band 1 of the slope raster.
4. 勾配は 15 度から 60 度の間である必要があります。

   インターフェイスのリスト項目とボタンを使って、次の式を組み立ててください:

   ```
   (slope@1 > 15) AND (slope@1 < 60)
   ```
5. 適切な場所とファイル名で **出力レイヤー** フィールドを設定してください。
6. **実行** をクリックします。



図 8.8: ラスタ計算機 傾斜

次に、同じ方法で正しい傾斜方位（東向き：45 から 135 度の間）を求めます。

1. 次の式を組み立てます:

   ```
   (aspect@1 > 45) AND (aspect@1 < 135)
   ```

出来上がったラスタで、東向きの斜面がすべて白くなっていれば、うまくいったことがわかります（まるで朝日に照らされているかのようです)。

同じ方法で正しい雨量（1000 mm 以上）を求めてください。次の式を使います:

```
Rainfall30@1 > 1000
```

3 つの条件がそれぞれ別々のラスタになったので、それらを組み合わせて、どの区域がすべての条件を満たしているかを確認する必要があります。そのために、ラスタは互いに掛け合わされます。このとき、値が 1 で重複しているピクセルはすべて 1 の値を保持します（つまり、その場所は条件を満たしています）。しかし、3 つのラスタのうち、いずれかのピクセルが 0 の値を持つ場合（つまり、その場所は条件を満たしていない）、それは結果としては 0 となります。このようにして、結果には適切な基準をすべて満たす重複領域のみが含まれることになります。

8.4.15 ラスタを組み合わせる

1. **ラスタ計算機** を開きます（**ラスタ　ラスタ計算機...**）
2. 次の式を組み立てます（レイヤの適切な名前を付けてください）：

```
[aspect45_135] * [slope15_60] * [rainfall_1000]
```

3. 出力先を Rasterprac ディレクトリに設定します
4. 出力するラスタに aspect_slope_rainfall.tif という名前を付けます
5. **結果をプロジェクトに追加する** がチェックされていることを確認してください
6. **実行** をクリックします

新しいラスタでは、3 つの条件を全て満たした領域が適切に表示されるようになりました。

プロジェクトを保存してください。



図 8.9: 3 つの条件全てを満たしたマップビュー

次に満たすべき基準は、その地域が都市部から 250 m 離れていることです。この条件を満たすには、計算する地域が農村部の中にあり、その地域の端から 250 m 以上離れていることを確認する必要があります。したがって、まずすべての農村地域を見つける必要があります。

8.4.16 農村地域を検索する

1. **レイヤ** パネル内のすべてのレイヤを非表示にします
2. Zoning ベクタレイヤの非表示を解除します
3. その上で右クリックして、 **属帝テーブル** ダイアログを表示します。ここでは、土地がさまざまな方法でゾーニングされていることに注意してください。私たちは、農村部を分離したいのです。属性テーブルを閉じます。
4. Zoning レイヤを右クリックして **フィルタ...** を選択し、 **クエリビルダ** ダイアログを表示させます
5. 次のクエリを組み立てます:

```
"Gen_Zoning" = 'Rural'
```

行き詰まったときは、先の説明を参照してください。

6. *OK* をクリックして、 **クエリビルダ** ダイアログを閉じます。クエリは 1 つの地物を返すはずです。

クエリビルダ Zoning

Zoning レイヤにある農村部のポリゴンが表示されるはずです。これらを保存する必要があります。

1. Zoning の右クリックメニューで、 **エクスポート** **新規ファイルに地物を保存**... を選びます。
2. レイヤを Rasterprac ディレクトリの下に保存します
3. 出力ファイルを rural.shp と名付けます
4. *OK* をクリックします
5. プロジェクトを保存してください

ここで農村地域の端から 250m 内にある領域を除外する必要があります。これは以下に説明するように、負のバッファを作成することによって行います。

8.4.17 負のバッファを作成する

1. メニュー項目 **ベクタ** **空間演算ツール** **バッファ**... をクリックします
2. 表示されたダイアログで、入力ベクタレイヤとして rural を選択します（**選択した地物のみ** ` にはチェックを入れません）
3. **距離** に -250 を設定します。負の値は、内部バッファになることを意味します。ドロップダウンメニューで、単位がメートルであることを確認します。
4. **結果を融合する** をチェックします
5. **出力レイヤ** では、出力ファイルを Rasterprac ディレクトリに置き、その名前を rural_buffer.shp とします
6. **保存** をクリックします
7. **実行** をクリックし、処理が完了するのを待ちます
8. **バッファ** ダイアログを閉じます。

   rural_buffer レイヤと rural レイヤがどのように違うかを見て、バッファが正しく機能したことを確認してください。違いを観察するために、描画順序を変更する必要があるかもしれません。
9. rural レイヤを削除します
10. プロジェクトを保存してください

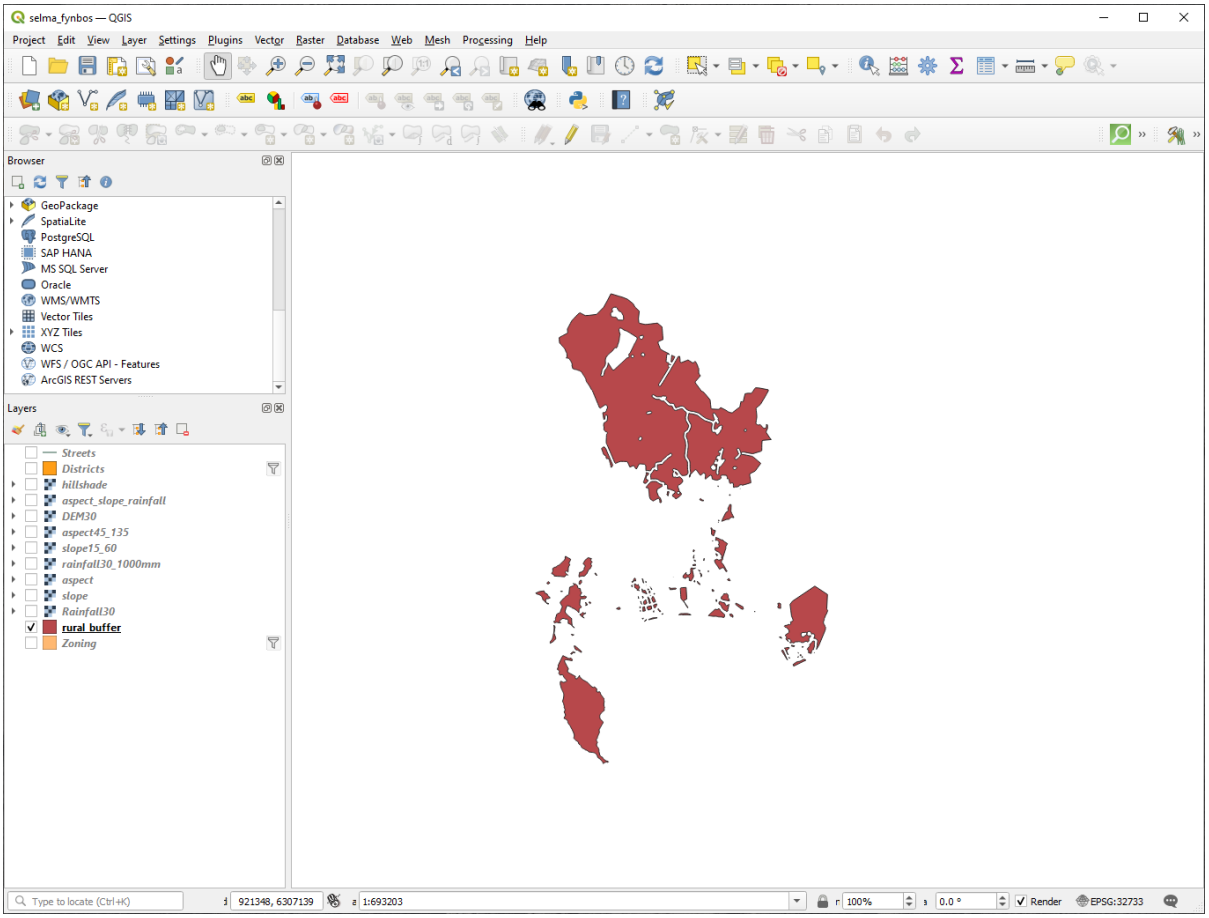


図 8.10: 農村バッファを持ったマップビュー

ここで、rural_buffer のベクタレイヤと aspect_slope_rainfall のラスタを結合する必要があります。これらを結合するためには、どちらかのレイヤのデータ形式を変更する必要があります。今回の場合、面積を計算するにはベクタレイヤの方が便利なので、ラスタをベクトル化することにします。

8.4.18 ラスタをベクタ化する

1. メニュー項目 **ラスタ　変換　ラスタのベクタ化** *(polygonize)...* をクリックします
2. aspect_slope_rainfall ラスタを **入力レイヤ** として選びます
3. **作成するフィールドの名前** に suitable を設定します（デフォルトのフィールド名は DN - デジタル番号データです）
4. 出力を保存します。**ベクタ化** の下にある **ファイルに保存** を選択します。保存場所を Rasterprac に設定し、ファイル名を aspect_slope_rainfall_all.shp にします。
5. **結果をプロジェクトに追加する** がチェックされていることを確認してください
6. **実行** をクリックします
7. 処理が完了したらダイアログを閉じます

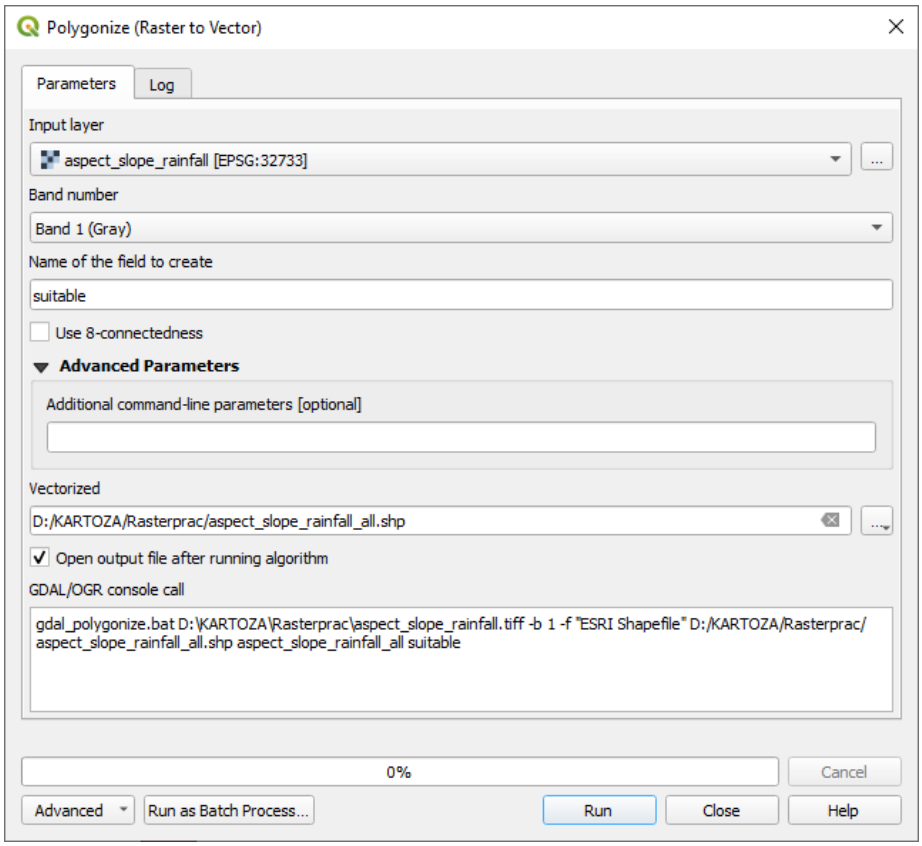


図 8.11: ラスタからベクタ

ラスタのすべての領域がベクタ化されているので、`suitable` フィールドの値が 1 である領域のみを選択する必要があります。(デジタルナンバー

1. 新しいベクタレイヤ用に **クエリビルダ** ダイアログを開きます（右クリック - **フィルタ**... ）
2. 次のクエリを組み立てます:

```
"suitable" = 1
```

3. *OK* をクリックします
4. クエリの完了（3 つの条件をすべて満たす、つまり値が 1 の領域だけが表示される）を確認したら、レイヤの右クリックメニューの **エクスポート　新規ファイルに地物を保存**... を使い、結果から新しいベクタファイルを作成します
5. ファイルを `Rasterprac` ディレクトリに保存します
6. ファイルは `aspect_slope_rainfall_1.shp` と名付けます
7. 地図から `aspect_slope_rainfall_all` レイヤを削除します
8. プロジェクトを保存します

When we use an algorithm to vectorize a raster, sometimes the algorithm yields what is called "Invalid geometries", i.e. there are empty polygons, or polygons with mistakes in them, that will be difficult to analyze in the future. So, we need to use the *Fix Geometry* tool.

8.4.19 ジオメトリを修復する

1. In the *Processing Toolbox*, search for `Fix geometries`, and *Execute...* it
2. **入力レイヤ** には `aspect_slope_rainfall_1` を選択します
3. **出力レイヤ** で **ファイルに保存...** を選択し、出力を `Rasterprac` に、ファイル名を `fixed_aspect_slope_rainfall.shp` にして保存します。
4. **結果をプロジェクトに追加する** がチェックされていることを確認してください
5. **実行** をクリックします
6. 処理が完了したらダイアログを閉じます

ラスタをベクトル化し、ジオメトリを修復したので、 `fixed_aspect_slope_rainfall` レイヤと `rural_buffer` レイヤとの交点を見つけることによって、傾斜方位、勾配、雨量基準と人里からの距離基準を組み合わせることができるようになりました。

8.4.20 ベクタの交点を求める

1. メニューの **ベクタ** **空間演算ツール** **交差** *(intersect)...* をクリックします
2. 表示されたダイアログで、`rural_buffer` レイヤを **入力レイヤ** `として選択します
3. **オーバーレイレイヤ** で、`fixed_aspect_slope_rainfall` レイヤを選択します
4. **交差** では、出力ファイルを `Rasterprac` ディレクトリに配置します
5. 出力ファイル名は `rural_aspect_slope_rainfall.shp` とします
6. **保存** をクリックします
7. **実行** をクリックし、処理が完了するのを待ちます
8. **交差** ダイアログを閉じます

   重なっている部分だけが残っていることを確認し、交差が正しく行われたことを確認します
9. プロジェクトを保存してください

リストにある次の条件は、面積が `6000` ㎡以上であることです。このプロジェクトに適した大きさのエリアを特定するために、これからポリゴン面積を計算することになります。

8.4.21 各ポリゴンの面積を計算する

1. 新しいベクタレイヤの右クリックメニューを開く
2. **属性テーブルを開く** を選択します
3. テーブルの左上にある  編集モード切替 ボタンをクリックするか、 `Ctrl+e` を押します
4. テーブルの上部にあるツールバーの  フィールド計算機を開く ボタンをクリックするか、 `Ctrl+i` を押してください

5. 表示されたダイアログで、 **新規フィールドを作成** がチェックされていることを確認し、 **出力する属性（フィールド）の名前** を area に設定します。フィールド型は小数点付き数値 (real) である必要があります。**精度** を 1 （小数点以下 1 桁）に設定します。

6. **式** エリアに次を入力します:

```
$area
```

これはフィールド計算機がベクタレイヤの各ポリゴンの面積を計算し、新しい整数列（ area という）に計算された値を入力することを意味します。

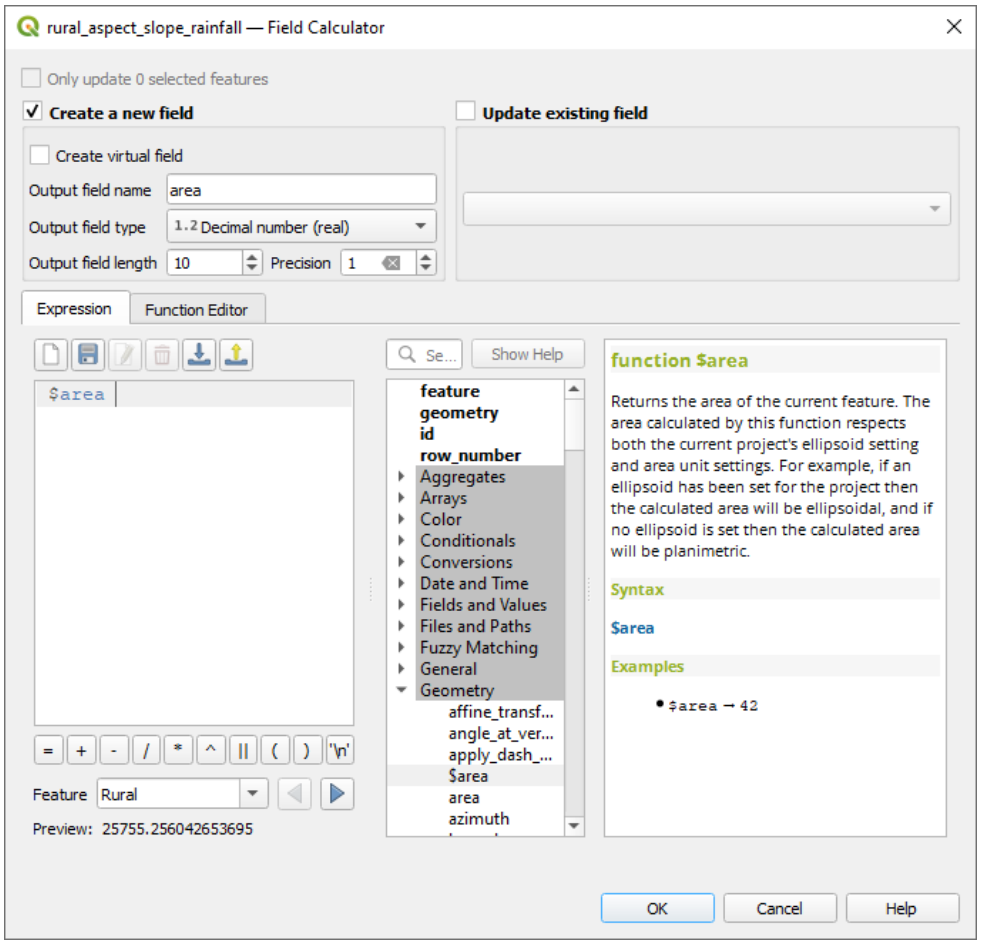


図 8.12: フィールド計算機

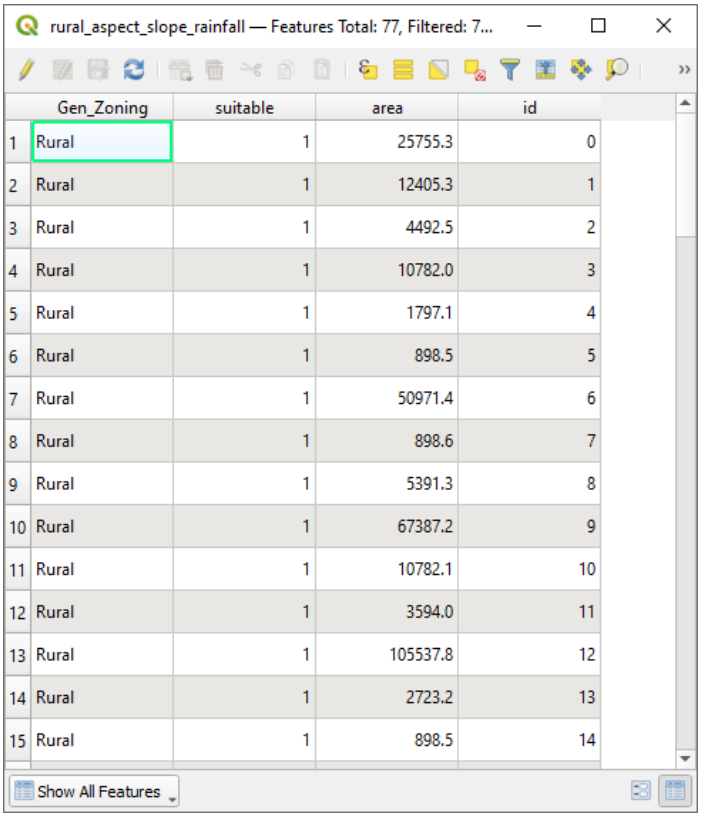
7. *OK* をクリックします

8. 同じことを id という別の新しいフィールドに対しても行います。**フィールド計算機式** に次を入力します:

```
$id
```

これは、各ポリゴンは識別目的のためのユニークな ID を持っていることを保証します。

9. 編集モード切替 をもう一度クリックし、編集を保存するように指示があれば保存します



	Gen_Zoning	suitable	area	id
1	Rural	1	25755.3	0
2	Rural	1	12405.3	1
3	Rural	1	4492.5	2
4	Rural	1	10782.0	3
5	Rural	1	1797.1	4
6	Rural	1	898.5	5
7	Rural	1	50971.4	6
8	Rural	1	898.6	7
9	Rural	1	5391.3	8
10	Rural	1	67387.2	9
11	Rural	1	10782.1	10
12	Rural	1	3594.0	11
13	Rural	1	105537.8	12
14	Rural	1	2723.2	13
15	Rural	1	898.5	14

図 8.13: area と id カラムのある属性テーブル

8.4.22 与えられたサイズの面積を選択する

今この面積が知られています：

1. 6000 m²以上のポリゴンのみを選択するクエリを（いつも通りに）作成します。クエリは次のようになります:

```
"area" > 6000
```

2. 選択範囲を Rasterprac ディレクトリに suitable_areas.shp という名前の新しいベクタレイヤとして保存します。

これで、希少なフィンボス植物の生息条件をすべて満たす適地が揃いました。この中から、ケープタウン大学に最も近い 4 つの地域を選びます。

8.4.23 ケープタウン大学のデジタイズ

1. 以前と同様に Rasterprac ディレクトリに新しいベクタレイヤを作成します。ただし今回は **ジオメトリタイプ** として **点** *(Point)* を使用し、 university.shp という名前を付けます
2. 正しい CRS（プロジェクト CRS: EPSG:32733 - WGS 84 / UTM zone 33S）であることを確認してください
3. 新しいレイヤの作成を完了します（*OK* をクリックします）
4. 新しい university レイヤと Streets レイヤ以外のすべてのレイヤを非表示にします。
5. 背景地図 (OSM) を追加します
   1. **ブラウザ** パネルから *XYZ Tiles* *OpenStreetMap* に移動します
   2. OpenStreetMap のエントリを **レイヤ** パネルの一番下にドラッグ＆ドロップします

   インターネットブラウザで、ケープタウン大学の所在地を調べてみてください。ケープタウンのユニークな地形を考えると、大学は非常にわかりやすい場所にあります。QGIS に戻る前に、大学の場所とその近くにあるものをメモしておきましょう。
6. **レイヤ** パネルで Streets レイヤが有効になっていること、そして university レイヤがハイライトされていることを確認してください
7. メニューから **ビュー** **ツールバー** を選択し、 **デジタイジングツールバー** が選択されていることを確認します。すると、鉛筆が描かれたツールバーアイコン（ 編集モード切替）が表示されているはずです。これは **編集モード切替** ボタンです。
8. **編集モード切替** ボタンをクリックすると、**編集モード** になります。これにより、ベクタレイヤを編集することができます
9. 点地物を追加する ボタンをクリックします。これは 編集モード切替 ボタンの近くにあるはずです
10. **地物を追加** ツールをアクティブにし、ケープタウン大学の推定位置を左クリックします
11. id を要求されたら、任意の整数を指定します
12. *OK* をクリックします
13. レイヤ編集内容を保存 ボタンをクリックします
14. 編集を中止するには **編集モード切替** ボタンをクリックします
15. プロジェクトを保存してください

8.4.24 ケープタウン大学から最寄りの場所を検索します

1. **プロセシングツールボックス** から、**属性の最近接結合** アルゴリズム（**ベクター一般　属性の最近接結合**）を選択し、実行します。
2. **入力レイヤ** は `university` 、 **第 *2* の入力レイヤ** は `suitable_areas` とします
3. Set an appropriate output location and name (`Joined layer`)
4. **近接地物の個数** を 4 に設定します
5. **結果をプロジェクトに追加する** がチェックされていることを確認してください
6. 残りのパラメータはデフォルト値のままにします
7. **実行** をクリックします

出来上がったポイントレイヤには 4 つの地物が含まれ、それらはすべて大学の位置とその属性、さらに近くの適切な領域の属性（``id``を含む）とその位置までの距離を持ちます。

1. 結合結果の属性テーブルを開きます
2. 最も近い 4 つの適切な領域の `id` を記録し、属性テーブルを閉じます
3. `suitable_areas` レイヤの属性テーブルを開きます
4. 大学に最も近い 4 つの地域を選択するクエリを作成します（`id` フィールドを使って選択します）

これは、研究の質問への最終的な答えです。

提出物は、あなたの選んだ魅力的なラスタ（例えば、DEM や傾斜ラスタなど）の上に半透明の陰影図レイヤを含む、完全にラベル付けされたレイアウトを作成してください。また、大学名と `suitable_areas` レイヤを含み、大学に最も近い 4 つの適切なエリアをハイライトしてください。地図製作のベストプラクティスにしたがって、出力地図を作成してください。

第9章　モジュール: プラグイン

プラグインを使用すると、QGIS の機能的な提供を拡張することができます。このモジュールでは、プラグインを有効にして使用する方法をお見せします。

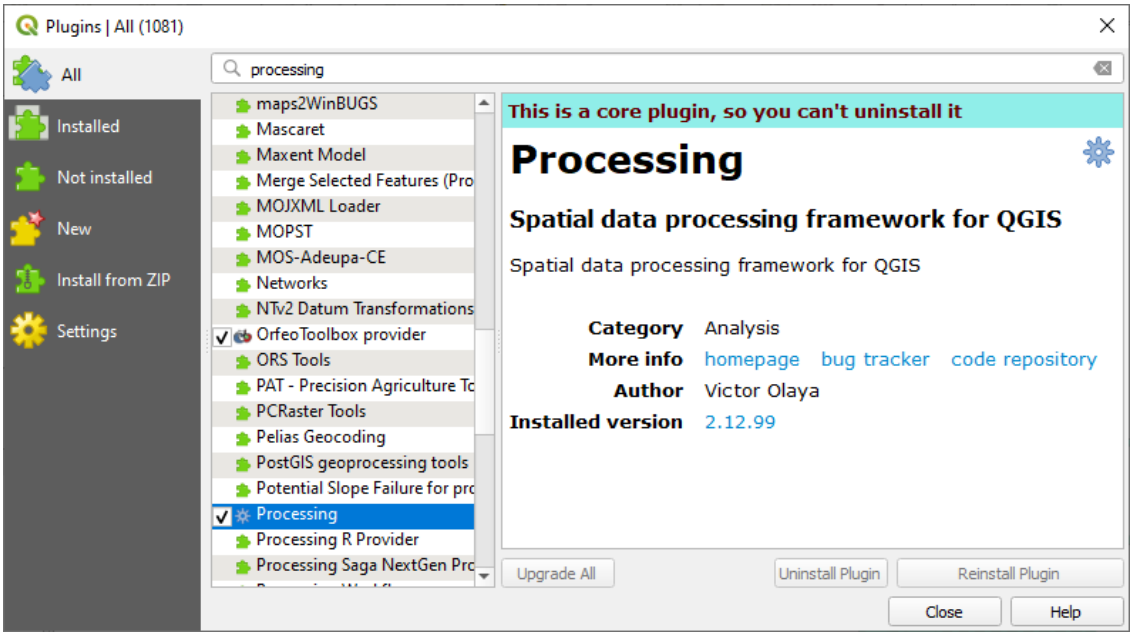
9.1 レッスン: プラグインのインストールと管理

プラグインを使い始めるにはそれらをどのようにダウンロードし、インストールし、有効化するのか知る必要があります。では、 **プラグインインストーラ** と **プラグインマネージャ** の使い方を学びましょう。

このレッスンの目標: QGIS のプラグインシステムを理解して使います.

9.1.1 ★☆☆ （初級レベル） 理解しよう: プラグインを管理する

1. **プラグインマネージャ** を開くには、メニュー **プラグイン　プラグインの管理とインストール** をクリックします。
2. 開かれたダイアログで **プロセッシング** プラグインを探しましょう。



3. このプラグインの横のボックスをクリックしてチェックを外し、それを無効にします。

4. **閉じる** をクリックします。

5. メニューを見てみると、 **プロセシング** メニューがなくなっていることに気がつくと思います。これは、今まで使っていた多くの処理関数が消えてしまったことを意味します! 例えば、 **ベクタ** と **ラスタ** メニューを見てください。これは、それらが **プロセシング** プラグインの一部であり、使用するにはプラグインを有効にする必要があるからです。

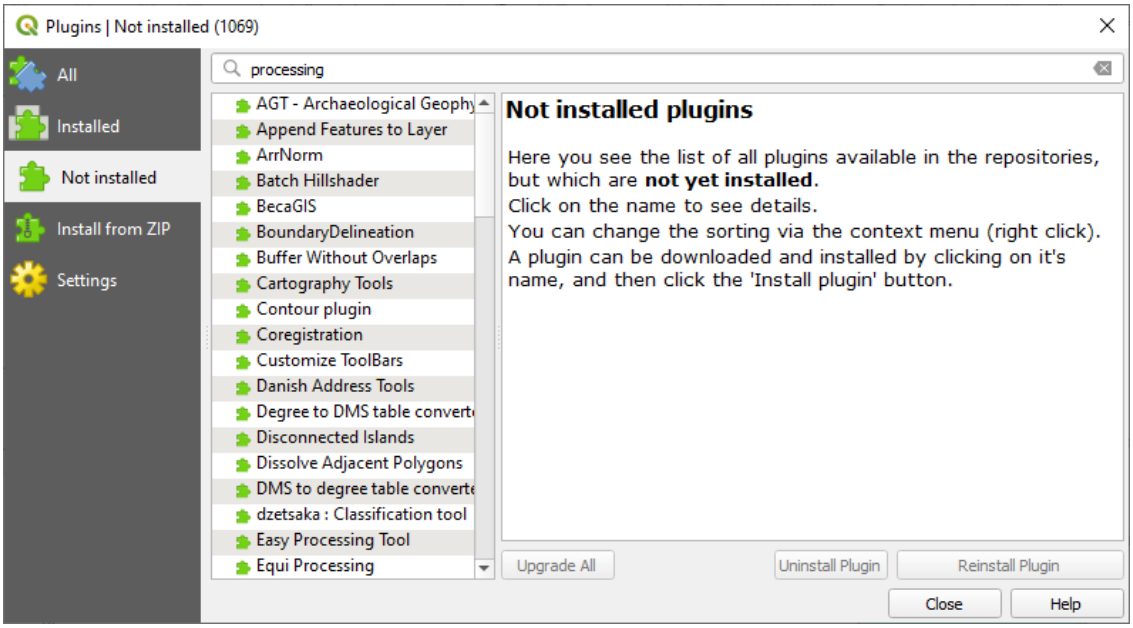
6. 再び **プラグインマネージャ** を開き、 *Processing* プラグインの横にあるチェックボックスをクリックして、再度有効化します。

7. ダイアログを **閉じる** します。*Processing* メニューと関数が再び利用可能になるはずです。

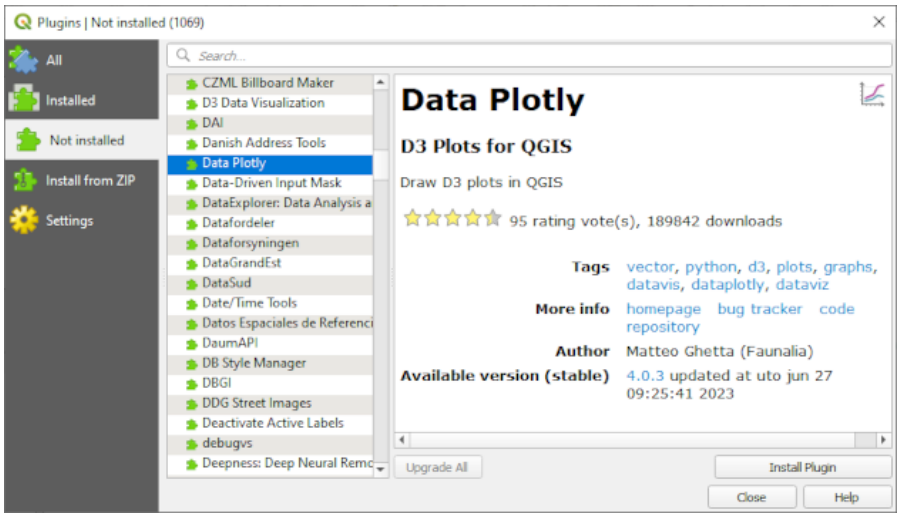
9.1.2 ★☆☆ （初級レベル） 理解しよう: 新しいプラグインをインストールする

有効化・無効化できるプラグインのリストは、現在インストールされているプラグインから引用されています。新しいプラグインをインストールするには:

1. **プラグインマネージャ** ダイアログで **未インストール** オプションを選択します。インストール可能なプラグインがここにリストアップされます。このリストは、あなたの既存のシステム設定によって異なります。



2. プラグインを選択すると、そのプラグインに関する情報が表示されます



3. プラグイン情報パネルの下にある **インストール** ボタンをクリックして、興味のあるプラグインをインストールしてください。

> **注釈**
>
> プラグインに何らかのエラーがある場合、 **無効** タブにリストアップされます。この場合、プラグインの所有者に連絡して、問題を解決することができます。

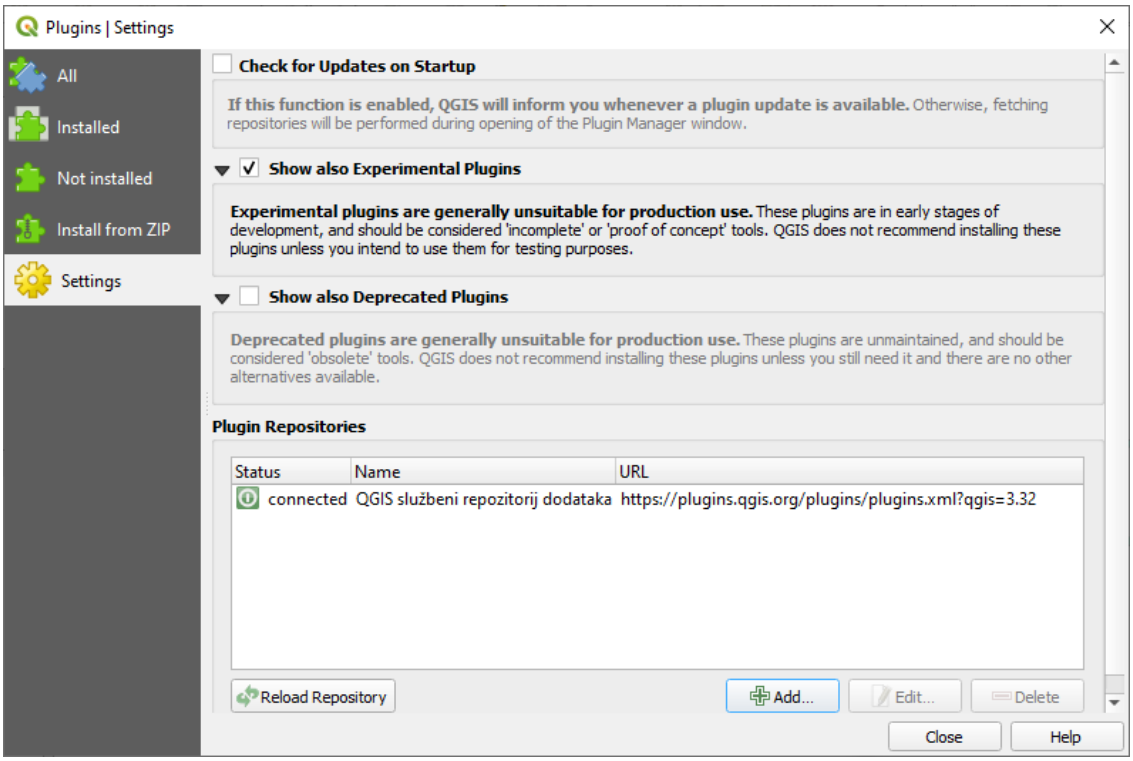
9.1.3 ★☆☆ （初級レベル） 理解しよう: 追加のプラグインリポジトリを設定する

利用可能なプラグインはどのプラグイン **リポジトリ** が設定されているかによって変わります。

QGIS のプラグインは、オンライン上のリポジトリに保存されています。デフォルトでは、公式リポジトリのみが有効です。つまり、そこで公開されているプラグインにのみアクセスすることができます。利用可能なツールの多様性を考えると、このリポジトリはあなたのニーズのほとんどを満たしてくれるはずです。

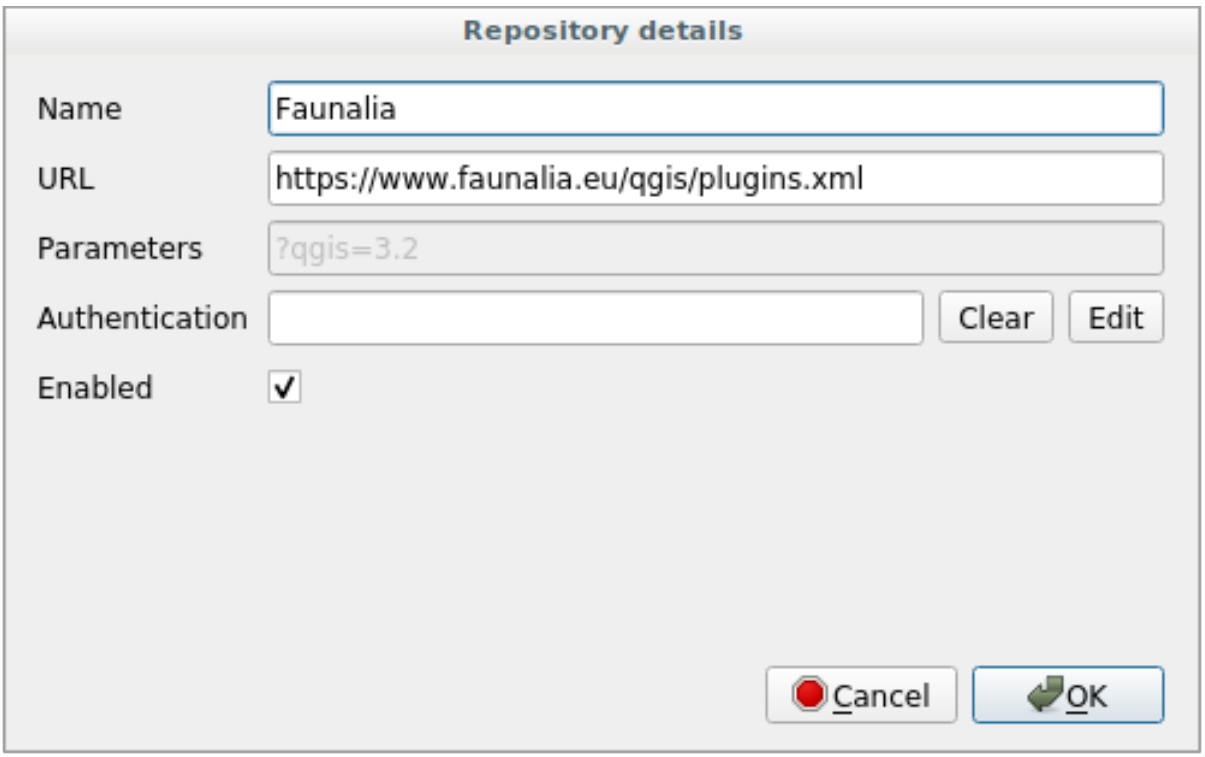
しかしデフォルトのものより多くのプラグインを試すこともできます。まず追加リポジトリを設定します。それには:

1. **プラグインマネージャ** ダイアログの **設定** タブを開いてください

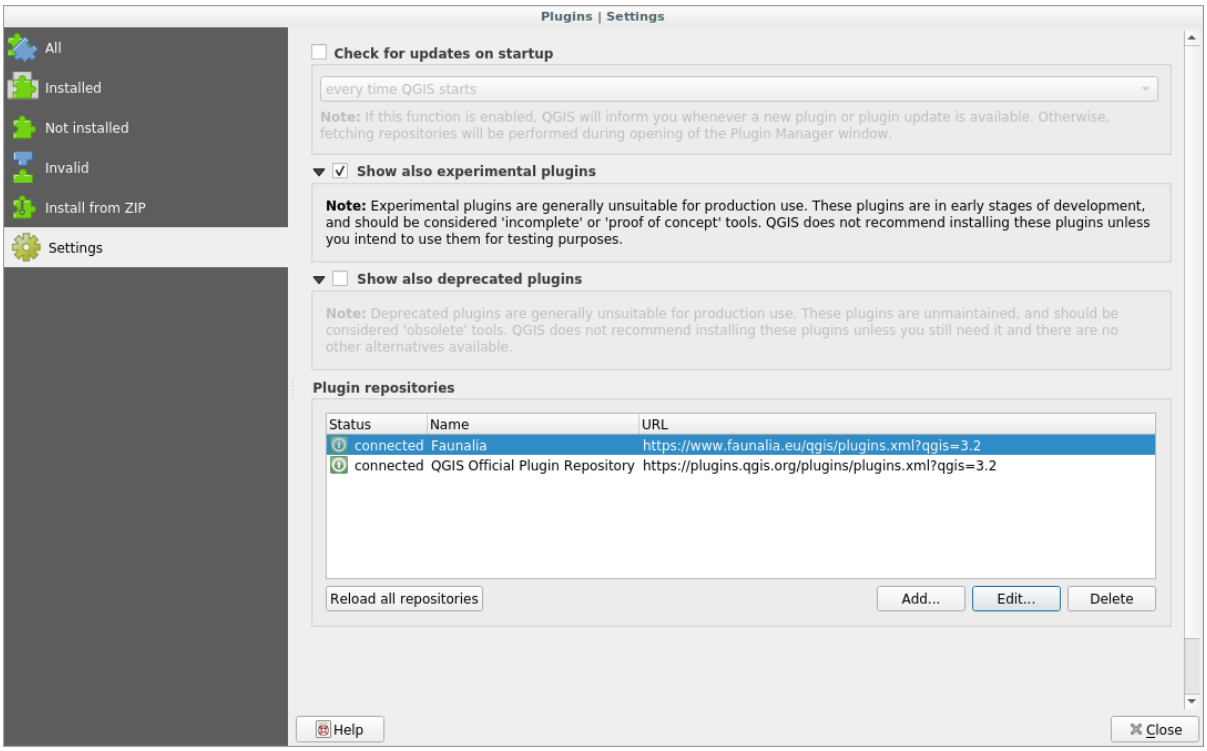


2. 新しいリポジトリを追加するには **追加** をクリックします。

3. 設定したい新しいリポジトリの名前と URL を入力します。 **有効化** チェックボックスがチェックされていることを確認します。



4. 新しいプラグインリポジトリがプラグインリポジトリの一覧の中に表示されます



5. 開発の初期段階にあるプラグインも表示するには **実験的プラグインを表示** チェックボックスをチェックします。
6. ここで **未インストール** タブに戻ると、追加プラグインがインストールできることがわかります。
7. プラグインをインストールするには、リストでプラグインをクリックし、 **インストール** ボタンをクリックします。

9.1.4 結論

QGIS にプラグインをインストールするのは、簡単で効果的でしょう！

9.1.5 次は?

次に、例としていくつかの便利なプラグインを紹介します。

9.2 レッスン: 便利な QGIS プラグイン

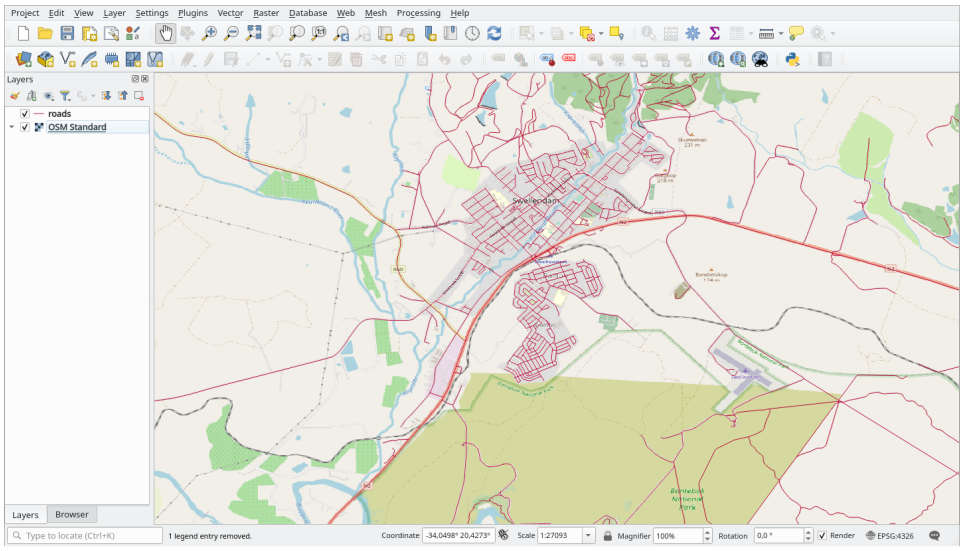
プラグインをインストールして有効化・無効化することができるようになりました。それではいくつかの便利なプラグインの例を見て、これが実際にどのように役に立つか見ていきましょう。

このレッスンの目標: プラグインインターフェイスに慣れ、いくつかの便利なプラグインを使ってみる。

9.2.1 ★☆☆ （初級レベル） 理解しよう: QuickMapServices プラグイン

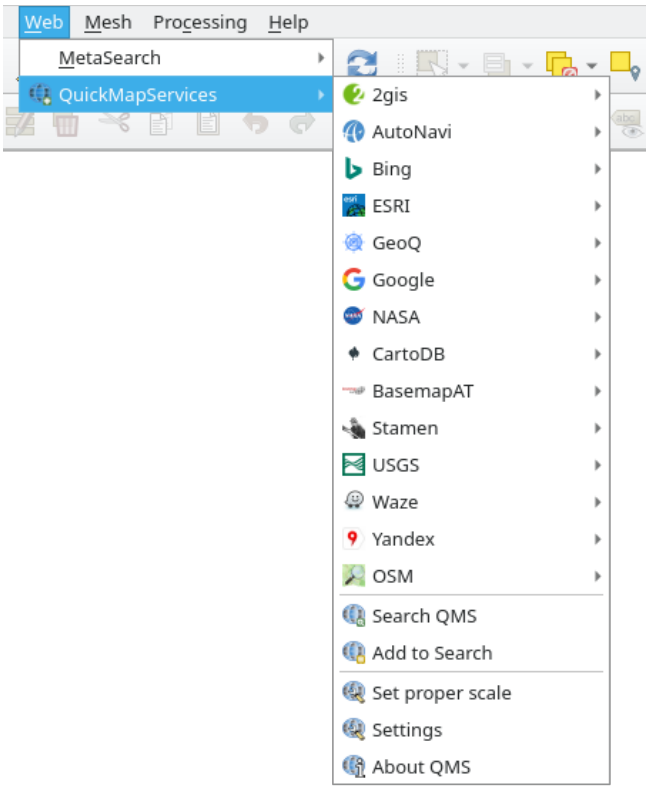
QuickMapServices プラグインは、QGIS プロジェクトにベースマップを追加するシンプルで使い勝手の良いプラグインです。様々なオプションや設定があるので、その機能の一部をご紹介します。

1. Start a new map and add the `roads` layer from the `training_data` Geopackage.
2. **QuickMapServices** プラグインをインストールしてください。
3. *Web* *QuickMapServices* をクリックします。最初のメニューは、様々なマッププロバイダ（`OSM`、`NASA`）と利用可能なマップの一覧です。
4. エントリをクリックすると、ベースマップがプロジェクトに読み込まれます。



いいですね！しかし、QMS の主な強みのひとつは、多くのデータプロバイダにアクセスできることです。それらを追加してみましょう。

1. *Web* *QuickMapServices* *Settings* をクリックします
2. *More services* タブに移動します。
3. このタブのメッセージをよく読んで、同意したら *Get Contributed pack* ボタンをクリックします。
4. **保存** をクリックします。
5. *Web* *QuickMapServices* メニューを再度開くと、利用可能なプロバイダーが増えていることがわかります。



6. あなたのニーズに最も適したものを選択し、プロジェクトにデータを読み込みます！

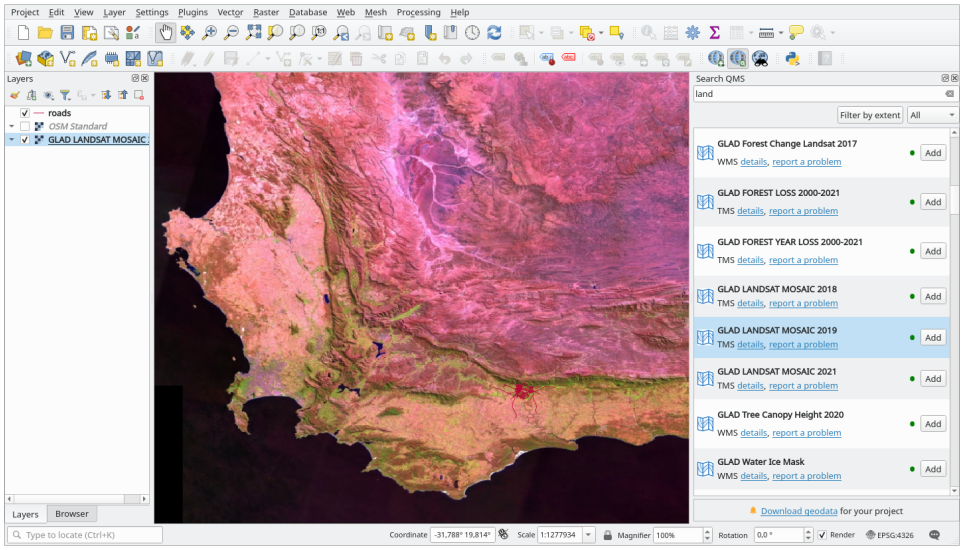
また、現在利用可能なデータ・プロバイダーを検索することもできます。

1. *Web* *QuickMapServices* *Search QMS* をクリックして、プラグインの検索タブを開きます。プラグインのこのオプションは、利用可能なベースマップをマップキャンバスの現在の範囲や検索語によってフィルタリングすることができます。

2. *Filter by extent* をクリックすると、利用可能なサービスが 1 つ表示されるはずです。サービスが見つからない場合は、ズームアウトして世界（または現在地）をパンするか、キーワードで検索してください。

3. 返されたデータセットの横にある *Add* ボタンをクリックすると、それが読み込まれます。

4. ベースマップが読み込まれ、マップの背景ができます。



9.2.2 ★☆☆ （初級レベル） 理解しよう: QuickOSM プラグイン

QuickOSM プラグインは、驚くほどシンプルなインターフェースで、OpenStreetMap のデータをダウンロードすることができます。

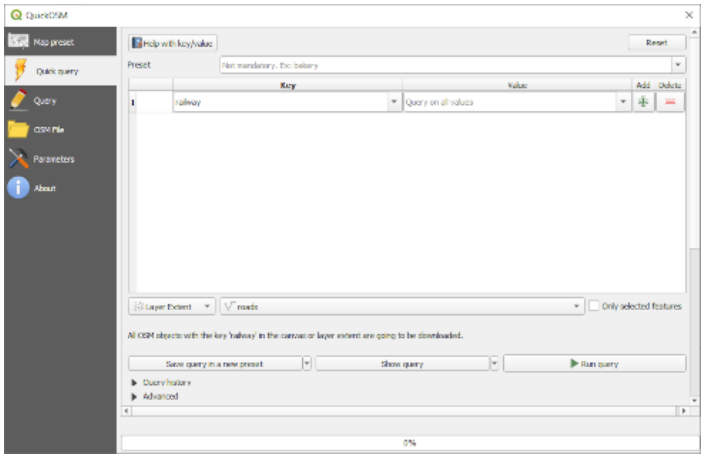
★☆☆ （初級レベル） 理解しよう: Quick Query を使う

1. Start a new empty project and add the `roads` layer from the `training_data` GeoPackage.
2. **QuickOSM** プラグインをインストールしてください。このプラグインは QGIS ツールバーに 2 つの新しいボタンを追加し、またそれには **ベクタ**　*QuickOSM* メニューからアクセスすることができます。
3. QuickOSM ダイアログを開きます。このプラグインには様々なタブがありますが、ここでは *Quick Query* のタブを使います。
4. 一般的な *Key* を選択して特定の機能をダウンロードすることもできますし、より具体的に *Key* と *Value* のペアを選択することも可能です。

> **Tip**
>
> もしあなたが *Key* と *Value* のシステムについてよく知らないのであれば、 *Help with key/value* ボタンをクリックしてみてください。OpenStreetMap のこのコンセプトの完全な解説のウェブページが開きます。

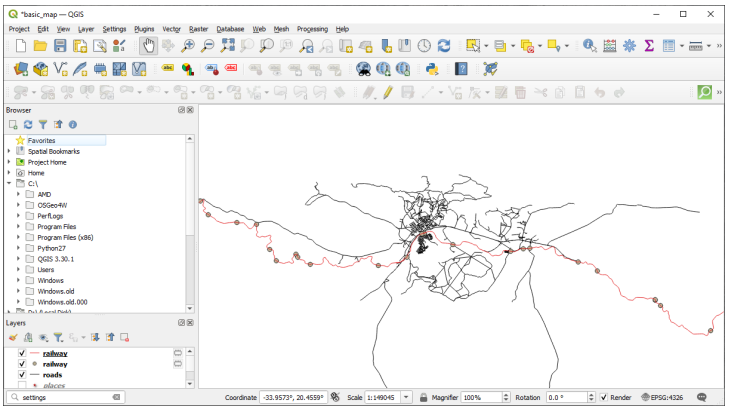
5. Look for `railway` in the *Key* menu and let the *Value* be empty: so we are downloading all the `railway` features without specifying any values.
6. Select *Layer Extent* in the next drop-down menu and choose `roads`.

7. *Run query* ボタンをクリックします。



数秒後、プラグインは OpenStreetMap で `railway` としてタグ付けされたすべての地物をダウンロードし、直接マップにロードします。

これだけです! すべてのレイヤは凡例に読み込まれ、マップキャンバスに表示されます。



警告

QuickOSM はデータをダウンロードする際に一時的なレイヤを作成します。恒久的に保存したい場合は、レイヤの横にある アイコンをクリックし、好みのオプションを選択します。または、QuickOSM の *Advanced* メニューを開き、 *Directory* メニューでデータの保存先を選択することもできます。

★★★ （上級レベル） 理解しよう: QuickOSM クエリエンジン

QuickOSM プラグインからデータをダウンロードする一番早い方法は、 *Quick query* タブを使い、いくつかの小さなパラメータを設定することです。しかし、もっと具体的なデータが必要な場合は？

あなたが OpenStreetMap のクエリマスターであれば、QuickOSM プラグインを個人のクエリにも使用できます。

QuickOSM は、Overpass の素晴らしいクエリエンジンとともに、驚くべきデータパーサを備えており、あなたのニーズに合わせたデータをダウンロードすることができます。

例えば、Dolomites という特定の山岳地域に属する山頂をダウンロードしたいとします。

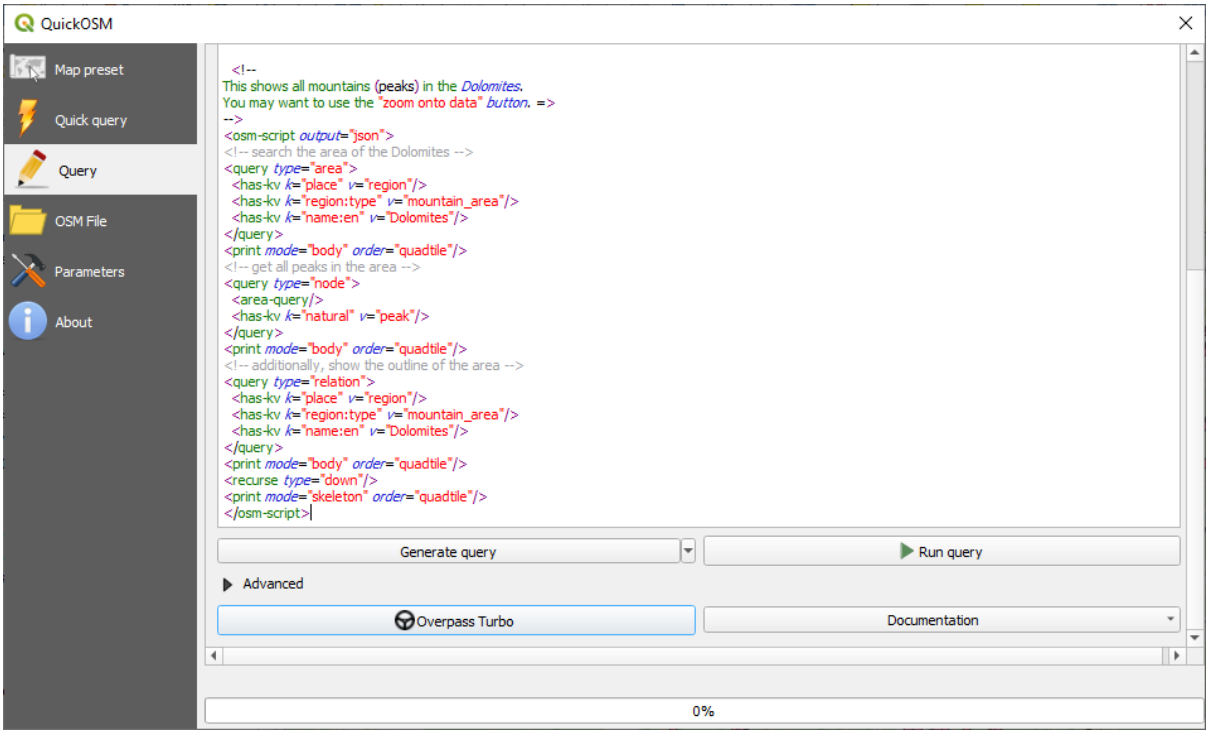
このタスクは *Quick query* タブでは達成できません。もっと具体的に、自分でクエリを書く必要があります。これをやってみましょう。

1. 新しいプロジェクトを始めます。
2. QuickOSM プラグインを開き、*Query* タブをクリックします。
3. 次のコードをコピーして、クエリキャンバスに貼り付けます:

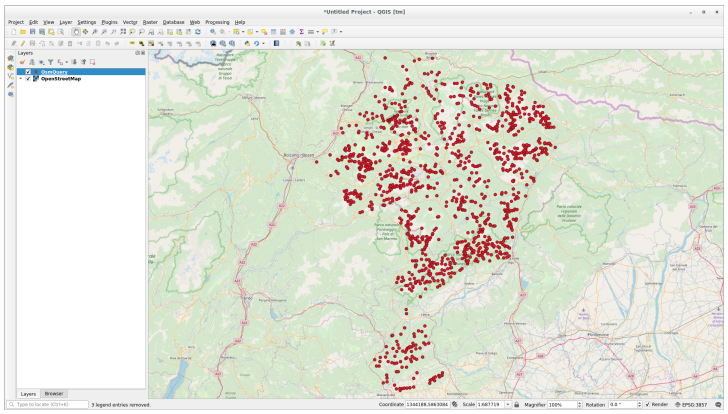
```
  <!--
This shows all mountains (peaks) in the Dolomites.
You may want to use the "zoom onto data" button. =>
-->
<osm-script output="json">
<!-- search the area of the Dolomites -->
<query type="area">
  <has-kv k="place" v="region"/>
  <has-kv k="region:type" v="mountain_area"/>
  <has-kv k="name:en" v="Dolomites"/>
</query>
<print mode="body" order="quadtile"/>
<!-- get all peaks in the area -->
<query type="node">
  <area-query/>
  <has-kv k="natural" v="peak"/>
</query>
<print mode="body" order="quadtile"/>
<!-- additionally, show the outline of the area -->
<query type="relation">
  <has-kv k="place" v="region"/>
  <has-kv k="region:type" v="mountain_area"/>
  <has-kv k="name:en" v="Dolomites"/>
</query>
<print mode="body" order="quadtile"/>
<recurse type="down"/>
<print mode="skeleton" order="quadtile"/>
</osm-script>
```

> **注釈**
>
> このクエリは `xml` に似た言語で書かれています。もし、 `Overpass QL` に慣れているのであれば、この言語でクエリを記述することができます。

4. *Run Query* をクリックします:



山頂レイヤがダウンロードされ、QGIS に表示されます:



Overpass Query language を使うと、複雑なクエリを書くことができます。いくつかの例を見て、クエリ言語を探求してみてください。

9.2.3 ★☆☆ （初級レベル） 理解しよう: DataPlotly プラグイン

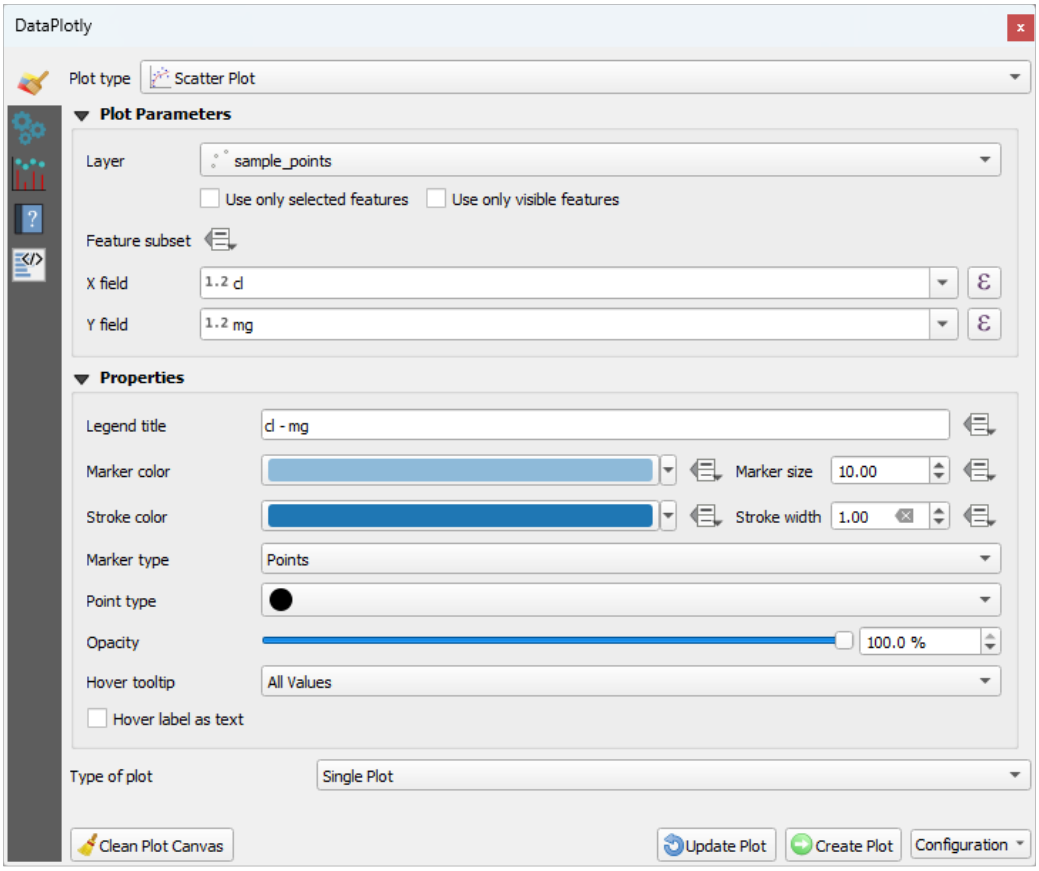
DataPlotly プラグインは、 plotly ライブラリにより、ベクタ属性データの D3 プロットを作成することができます。

1. 新しいプロジェクトを開始します
2. Load the `sample_points` layer from the `exercise_data/plugins` folder
3. プラグインをインストールするには、 **★☆☆ （初級レベル） 理解しよう: 新しいプラグインをインストールする** で説明されているガイドラインに従って、 *Data Plotly* を検索してください

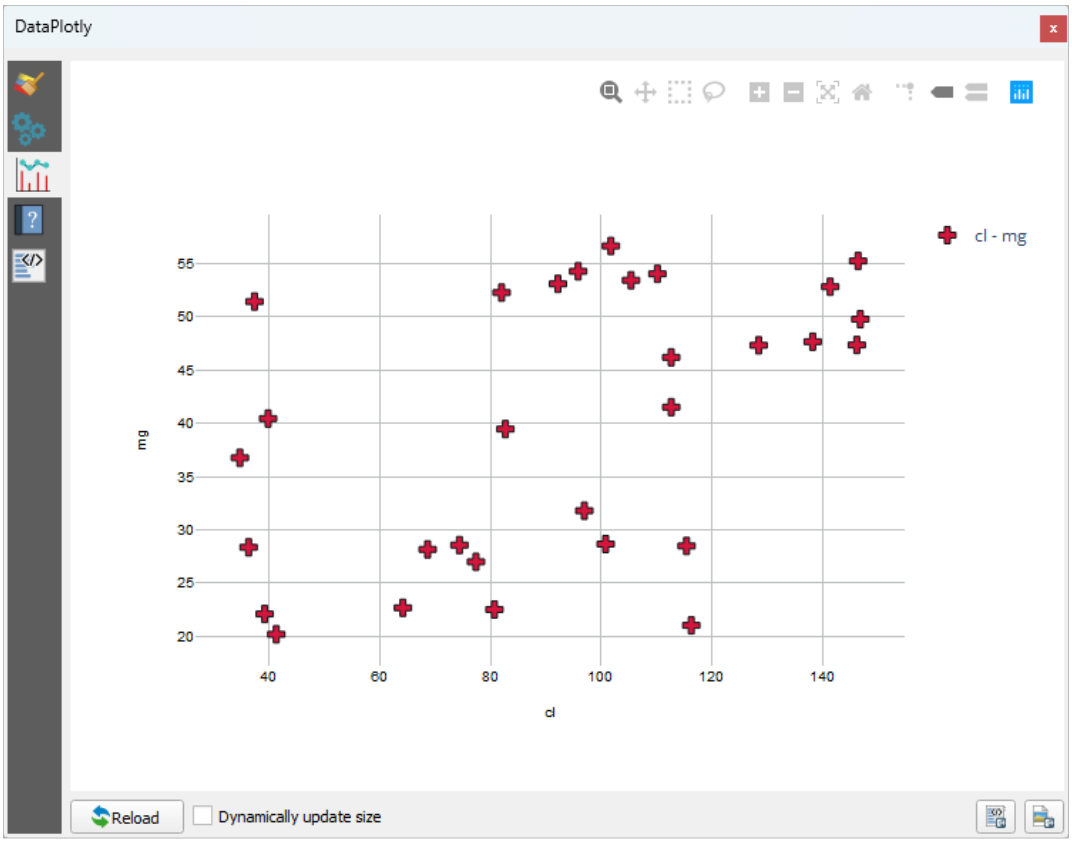
4. ツールバーの新しいアイコンをクリックするか、*Plugins* *Data Plotly* メニューから、プラグインを開きます

In the following example we are creating a simple `Scatter Plot` of two fields of the `sample_points` layer. In the DataPlotly Panel:

1. Choose `sample_points` in the Layer filter, *cl* for the *X Field* and *mg* for the *Y Field*:



2. 色やマーカーの種類、透明度など、さまざまな設定を変更することができます。いくつかのパラメータを変更して、下のようなプロットを作成してみてください。



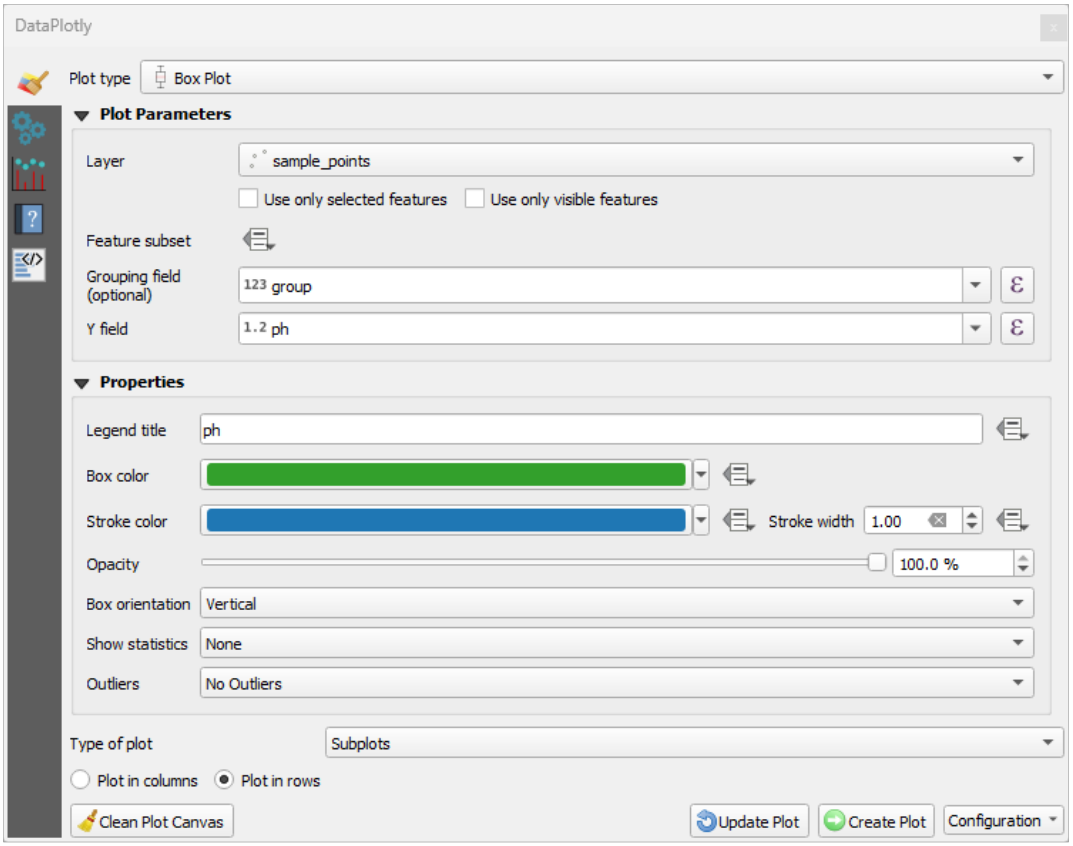
3. すべてのパラメータを設定したら、プロットを作成するために *Create Plot* ボタンをクリックします。

プロットは対話的です: つまり、プロットキャンバスのサイズ変更、移動、拡大・縮小のために上部のすべてのボタンを使うことができます。さらに、プロットの各要素は対話的です: プロット上で 1 つまたは複数の点をクリックまたは選択すると、対応する点がプロットキャンバスで選択されます。

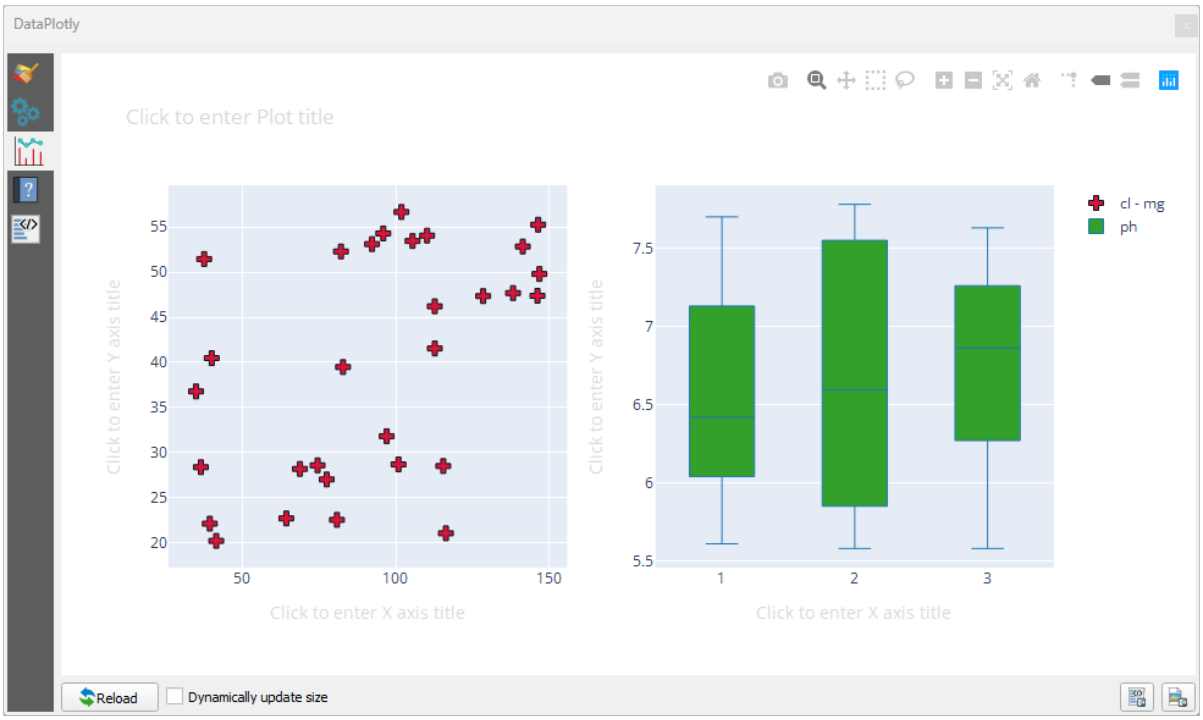
プロットの右下にある  または  ボタンをクリックすると、プロットを `png` 静止画像または `html` ファイルとして保存することができます。

もっとあります。同じページに異なる変数で異なるプロットタイプを表示する 2 つ（またはそれ以上）のプロットがあると便利なことがあります。これをやってみましょう!

1. プラグインパネルの左上隅にある  ボタンをクリックして、メインのプロット設定タブに戻ります
2. *Plot Type* を *Box Plot* に変更します
3. *group* を *Grouping Field* に、 *ph* を *Y Field* に選択します
4. パネルの下部で、 *Type of Plot* を *SinglePlot* から *SubPlots* に変更し、デフォルトオプション *Plot in Rows* を選択したままにします。



5. プロットを作成するには、 *Create Plot* ボタンをクリックします



これで散布図と箱ひげ図の両方が同じプロットページに表示されます。この状態でも、それぞれのプロット項目をクリックして、マップキャンバスの対応する地物を選択することができます。

Tip

それぞれのプロットは  タブの中で利用できる独自のマニュアルページを持っています。すべてのプロットタイプを探って、利用可能な他のすべての設定を見てみてください。

9.2.4 結論

QGIS ではたくさんの便利なプラグインを利用することができます。ビルトインツールを用いてこれらのプラグインを管理すれば、新しいプラグインを見つけ、それらを最適に利用することができます。

9.2.5 次は?

次はリモートサーバーにホストされているレイヤーをリアルタイムで使う方法を見ていきます。

第10章　モジュール: オンラインリソース

地図のデータソースを検討する際に、あなたが使っているコンピュータ上に保存したデータに限定される必要はありません。インターネットに接続されている限り、ロードできるオンラインのデータソースがあります。

このモジュールでは、２種類の Web ベースの GIS サービスについて学習します: Web Mapping Services (WMS) と Web Feature Services (WFS)。

10.1 レッスン: Web Mapping Service

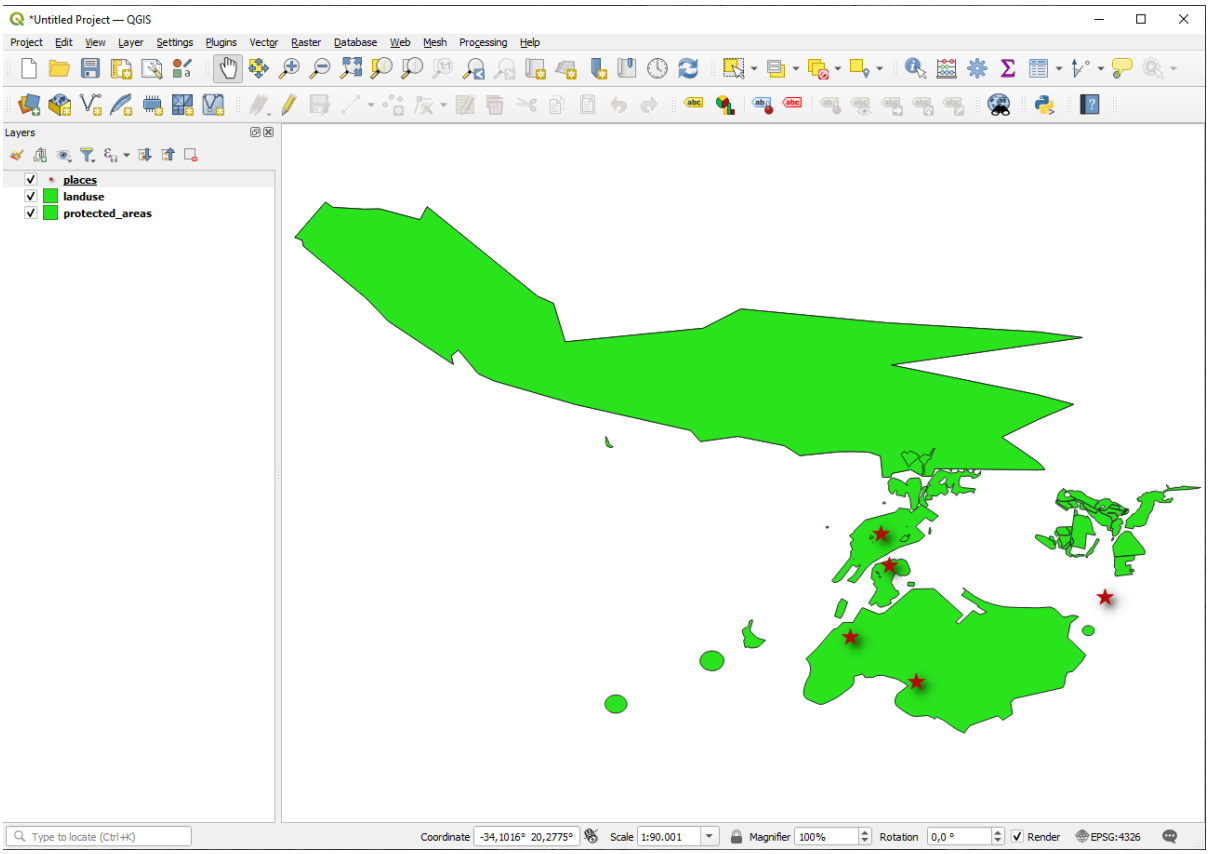
Web Mapping Service (WMS) は、リモートサーバ上でホストされたサービスです。ウェブサイトと同じ様に、サーバーへ接続することでアクセスができます。QGIS を使用すると、既存の地図に直接 WMS をロードできます。

プラグインのレッスンから、例えば Google から新しいラスタ画像をロードできることを記憶しているでしょう。しかし、これは一度きりの処理です：画像はダウンロードしたら変更されません。WMS は、地図上でパンや拡大すると自動的に表示が更新されるライブサービスであるところが異なります。

このレッスンの目標: WMS を使用して制限を知ること。

10.1.1 ★☆☆ （初級レベル） 理解しよう: WMS レイヤを読み込む

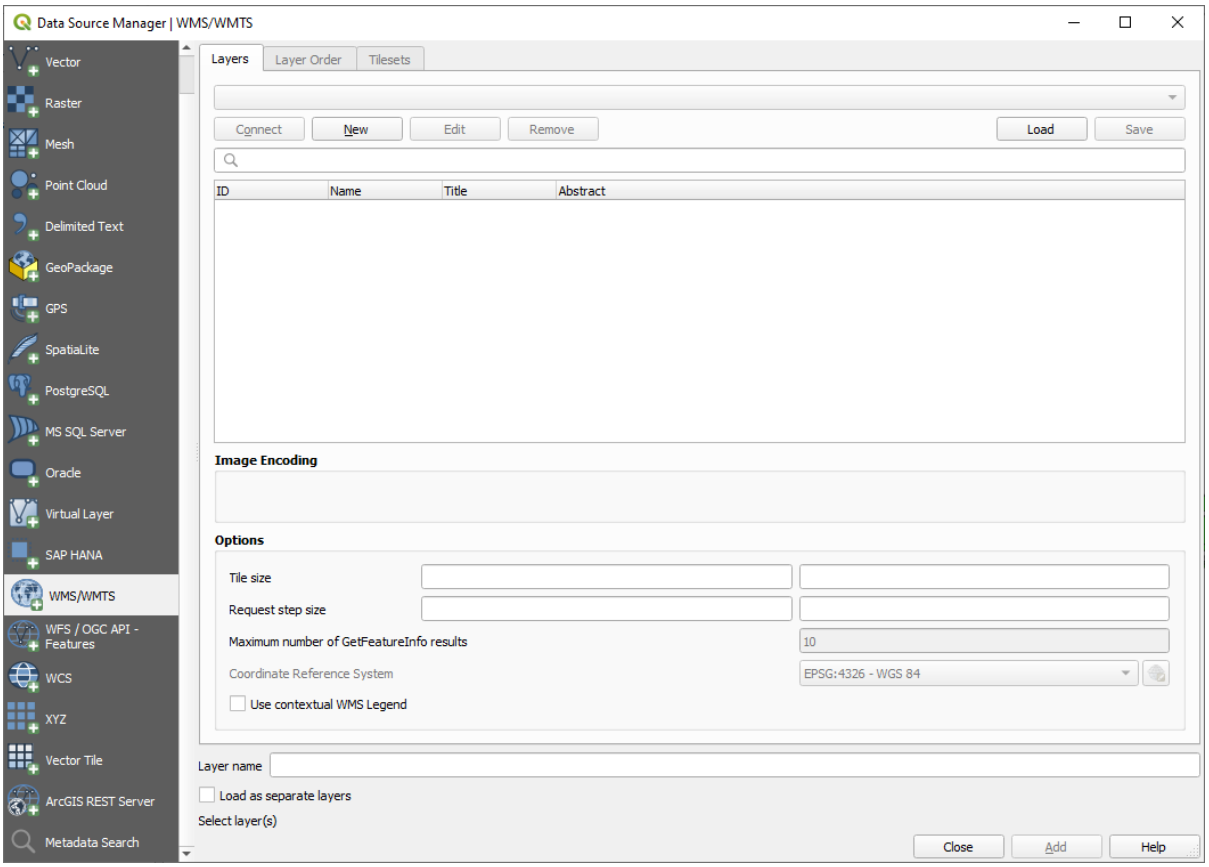
For this exercise, you can either use the basic map you made at the start of the course, or just start a new map and load some existing layers into it. For this example, we used a new map and loaded the original `places`, `landuse` and `protected_areas` layers and adjusted the symbology:



1. 新しい地図にこれらのレイヤを読み込みます。またはこれらのレイヤを表示のみ行いオリジナルの地図を使用します。

2. WMS レイヤの追加を開始する前に、"オンザフライ" 投影を無効にします (**プロジェクト　プロパティ...　座標参照系** *(CRS)* タブで、 *CRS* **なし（または未知/非地球）** をチェックします。これにより、レイヤが正しく重ならなくなることがありますが、後で修正しますので安心してください。

3. WMS レイヤを追加するには、 ボタンをクリックして **データソースマネージャ** ダイアログを開き、 *WMS/WMTS* タブを有効にしてください。

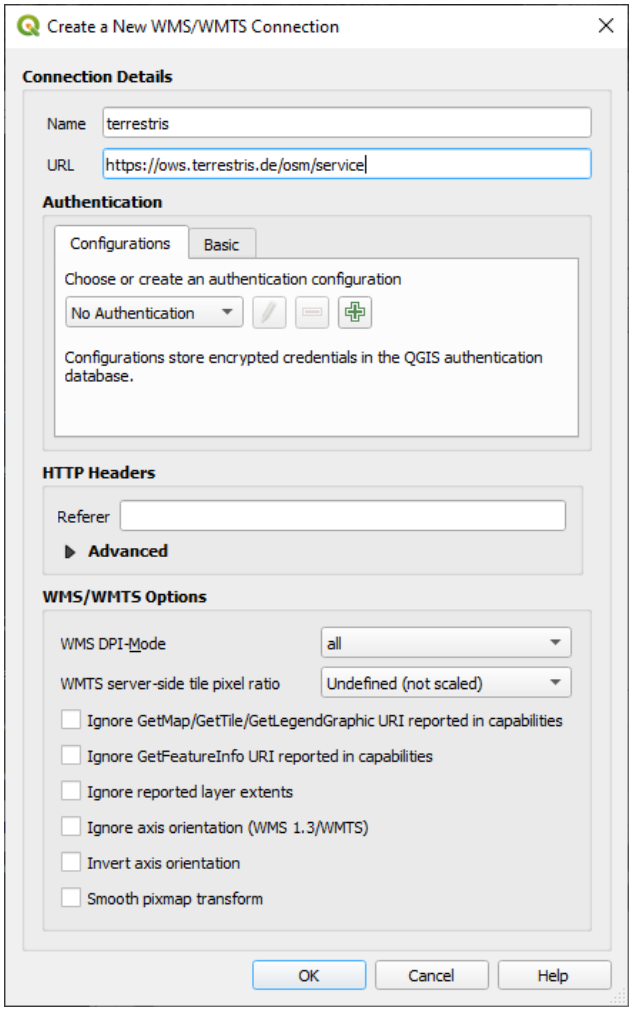


Remember how you connected to a SpatiaLite or GeoPackage database at the beginning of the course. The `landuse`, `buildings`, and `roads` layers are stored in a database. To use those layers, you first needed to connect to the database. Using a WMS is similar, with the exception that the layers are on a remote server.

4. WMS への新しい接続を作成するには、 **新規** ボタンをクリックしてください。

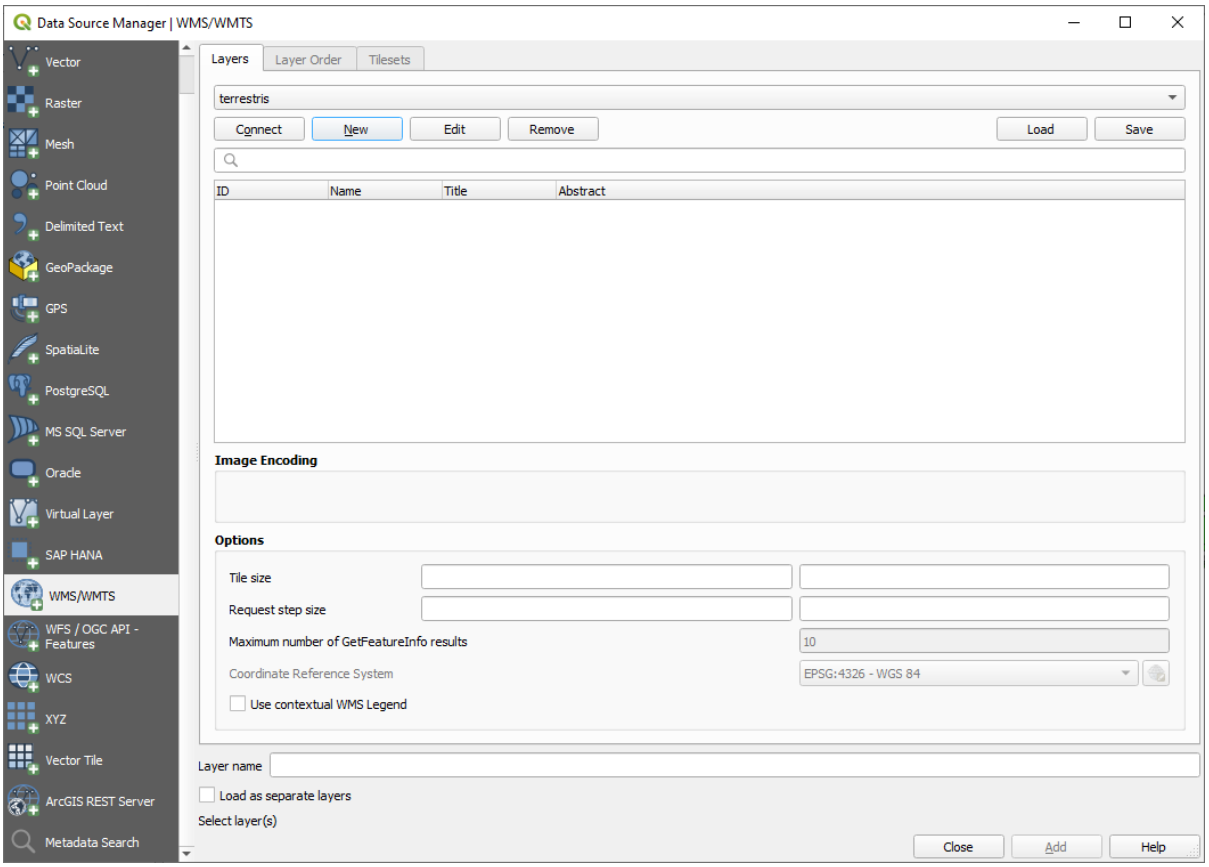
   続けるには WMS アドレスが必要です。インターネットには利用できる無料の WMS サーバがいくつかあります。そのうちのひとつは terrestris 、で OpenStreetMap データセットを使用しています。

5. この WMS を利用するには、このように、今表示しているダイアログで設定します。

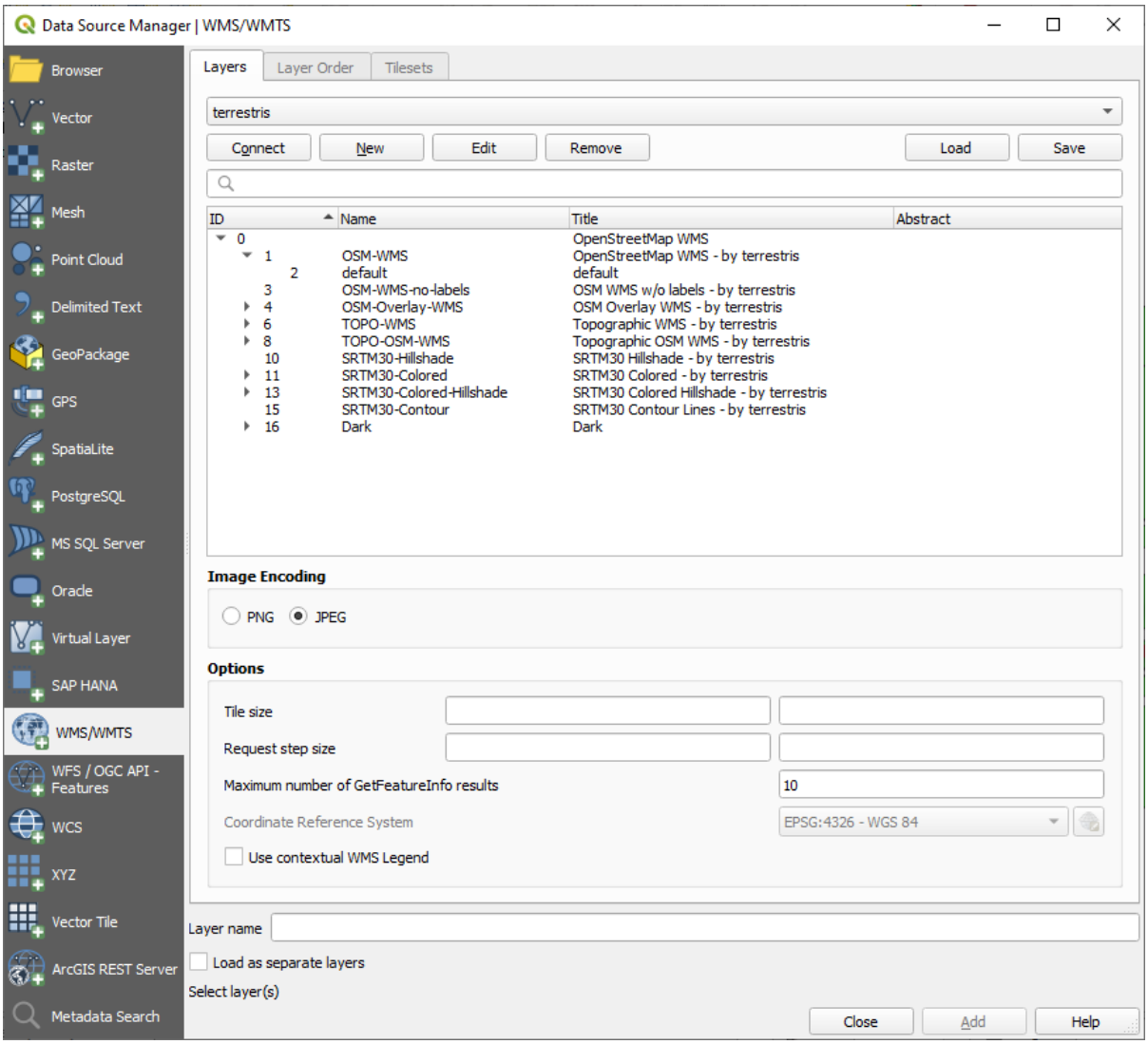


- *Name* フィールドの値は `terrestris` にします。
- *URL* フィールドの値は `https://ows.terrestris.de/osm/service` にします。

6. *OK* をクリックします。リストされている新しい WMS サーバが表示されます。

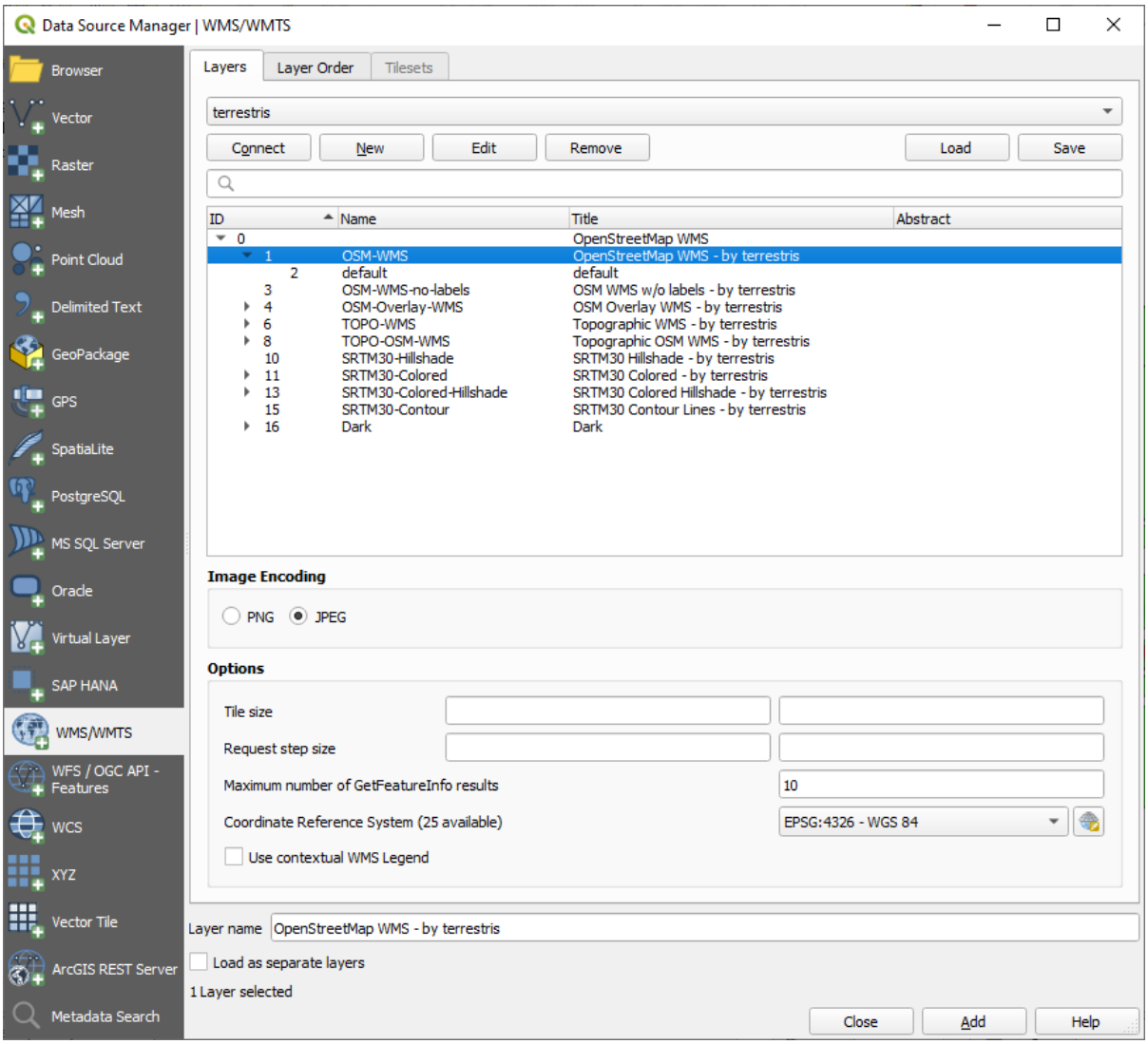


7. **接続** をクリックしてください。すると下のリストに、これらの新しいエントリがロードされるはずです:



これらはこの WMS サーバーにホストされているすべてのレイヤです。

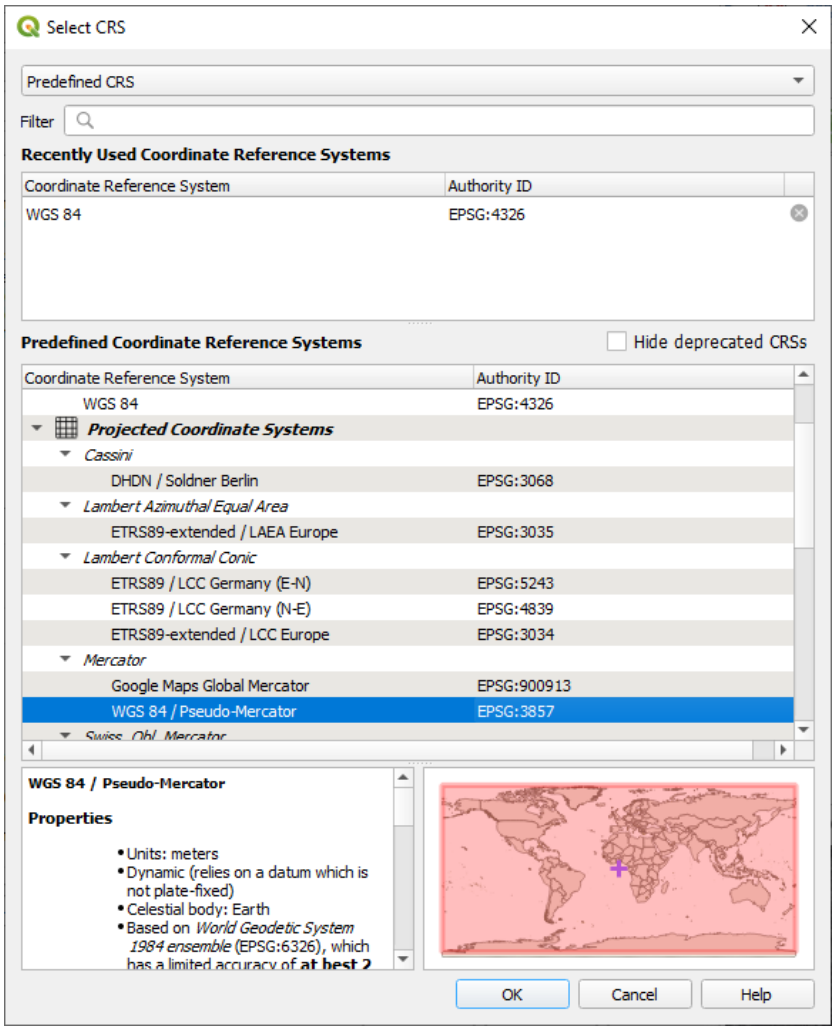
8. Click once on the `OSM-WMS` layer. This will display the default *Coordinate Reference System* in use and the number of CRSes supported by the dataset:



私たちの地図には ``EPSG:4326 - WGS 84`` は使わないので、ニーズに合うものを探しましょう。

1. CRS を選択 ボタンをクリックします。標準の **座標参照系選択** ダイアログが表示され、データが公開されているすべての CRS が表示されます。

2. 私たちには **投影された** CRS が必要なので、EPSG:3857 の *WGS 84 / Pseudo-Mercator* を選択してみましょう。一番上の **フィルタ** ウィジェットを使うことができます。



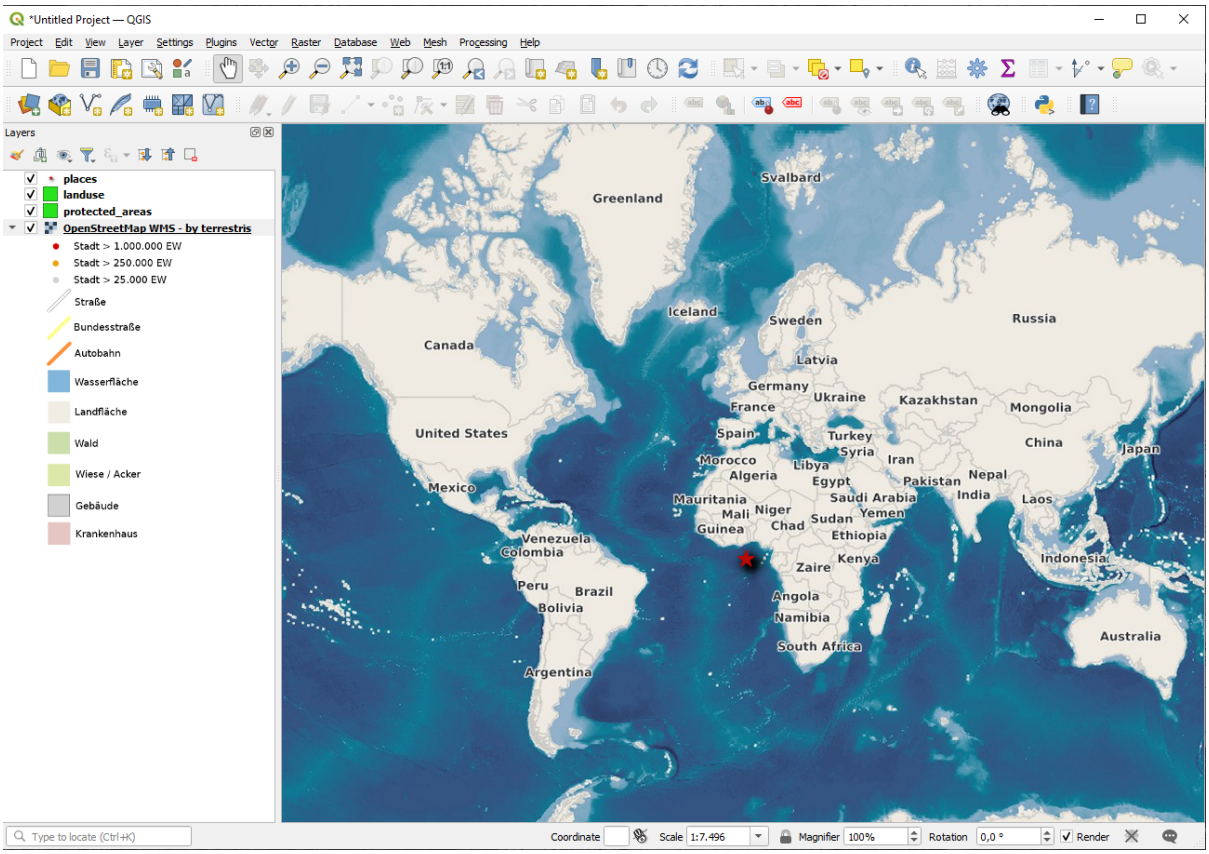
1. *OK* をクリックします。入力したものに関係した座標参照系が変わりました。

9. Click *Add* to load the layer in your project using *Layer name* (default is `OpenStreetMap WMS - by terrestris`).

10. 自動的に閉じないときは **データソースマネージャ** ダイアログを閉じてください

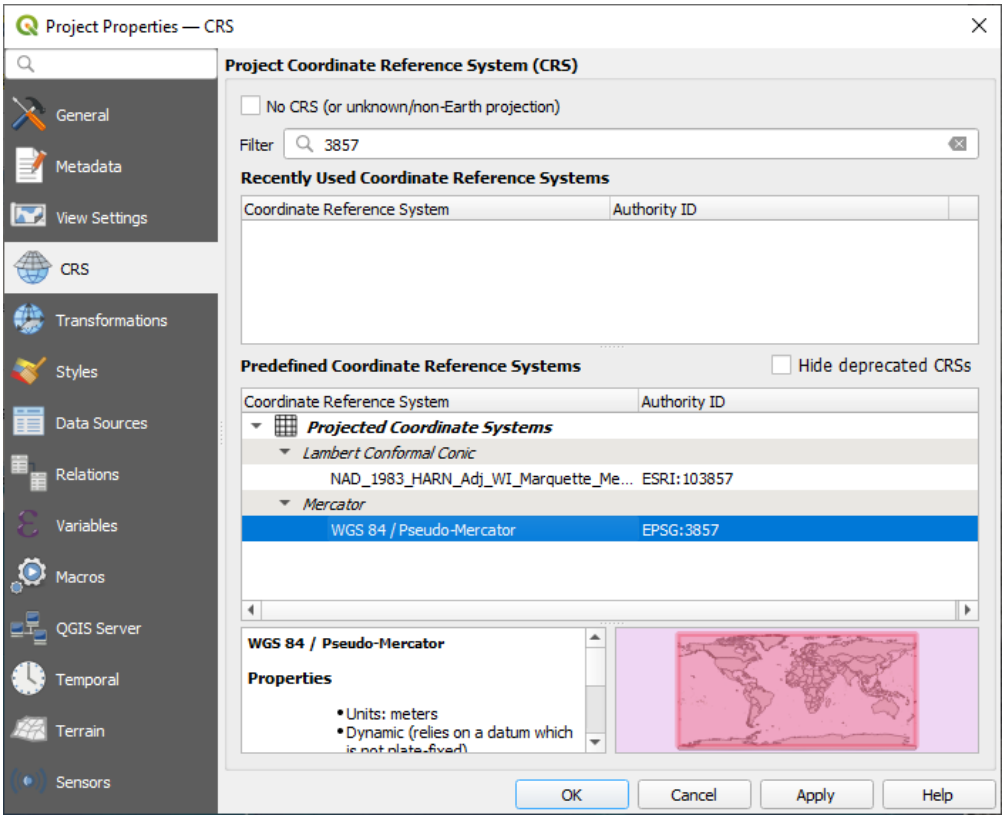
11. **レイヤ** パネルで、それをリストの最下部にクリック＆ドラッグします。

12. レイヤを全体的に表示するには、ズームアウトします。レイヤが正しく配置されていないことがわかります（アフリカの西に近い）。これは、「オンザフライ」投影が無効になっているためです。



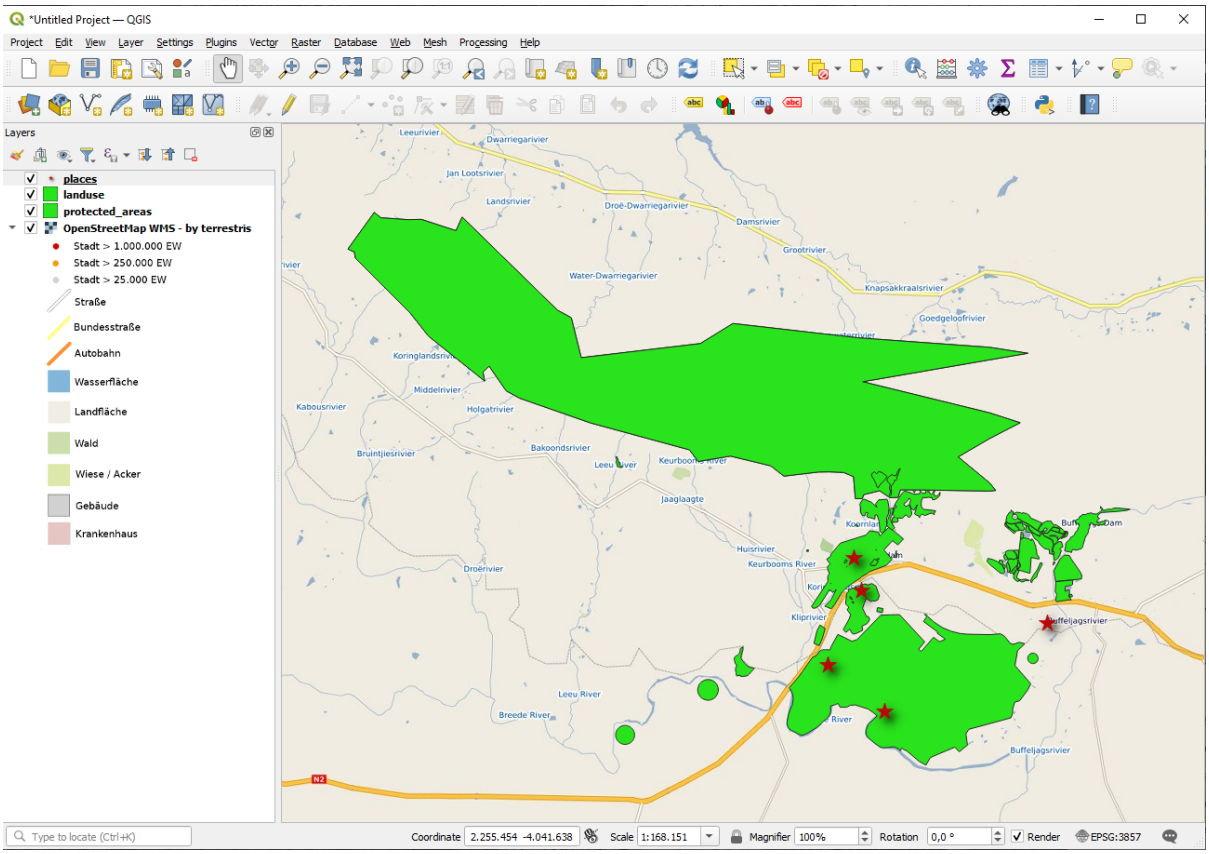
13. Let's enable the reprojection again, but using the same projection as the `OpenStreetMap WMS` layer, which is *WGS 84 / Pseudo Mercator*.

    1. **プロジェクト　プロパティ...**　*CRS* タブを開きます
    2. Uncheck *CRS* **なし（または未知/非地球）** のチェックを外します
    3. リストから *WGS 84 /* **疑似メルカトル** を選択します。



4. *OK* をクリックします。

14. 次に、 **レイヤ** パネルで自分のレイヤの一つを右クリックして、 **レイヤの領域にズーム** をクリックします。すると、Swellendam 領域が表示されるはずです:



WMS レイヤの街路と私たちの街路がどれほど重なっているかに注目してください。それは良い兆候です！

WMS の性質と限界

今までに、この WMS レイヤはその中に実際に多くの地物を持っていることに気づいたかもしれません。それは道路、河川、自然保護区、などなどがあります。しかも、それはベクターで構成されているように見えるにもかかわらず、ラスターのようですが、そのシンボルを変更できません。何故でしょうか？

これが WMS が動作する方法です：この地図は、紙面の通常の地図のように、画像として受け取っています。普段使うのは、QGIS が地図としてレンダリングする、ベクタレイヤです。しかし WMS を使うと、それらのベクタレイヤは WMS サーバー上にあり、地図としてレンダリングされ、画像としてその地図が送信されます。QGIS ではこの画像を表示できますが、すべてのことはサーバー上で処理されるため、そのシンボルは変更できません。

これはいくつかの利点を有しています。なぜならシンボルを心配する必要はありません。すでにできあがっていますし、的確に設計された WMS 上で見栄え良くなるはずです。

他方、気に入らなかったとしてもシンボルは変更できません。そして何かが WMS サーバー上で変更されると、それらは地図上でも同様に変更されます。ときどき代わりに Web 地物サービス（WFS）を使いたくなるのはこのためです。それは WMS スタイルの地図の一部としてではなく、別のベクタレイヤを与えます。

ただし、これは次のレッスンでカバーされます。まず、別の WMS レイヤーを追加しましょう。

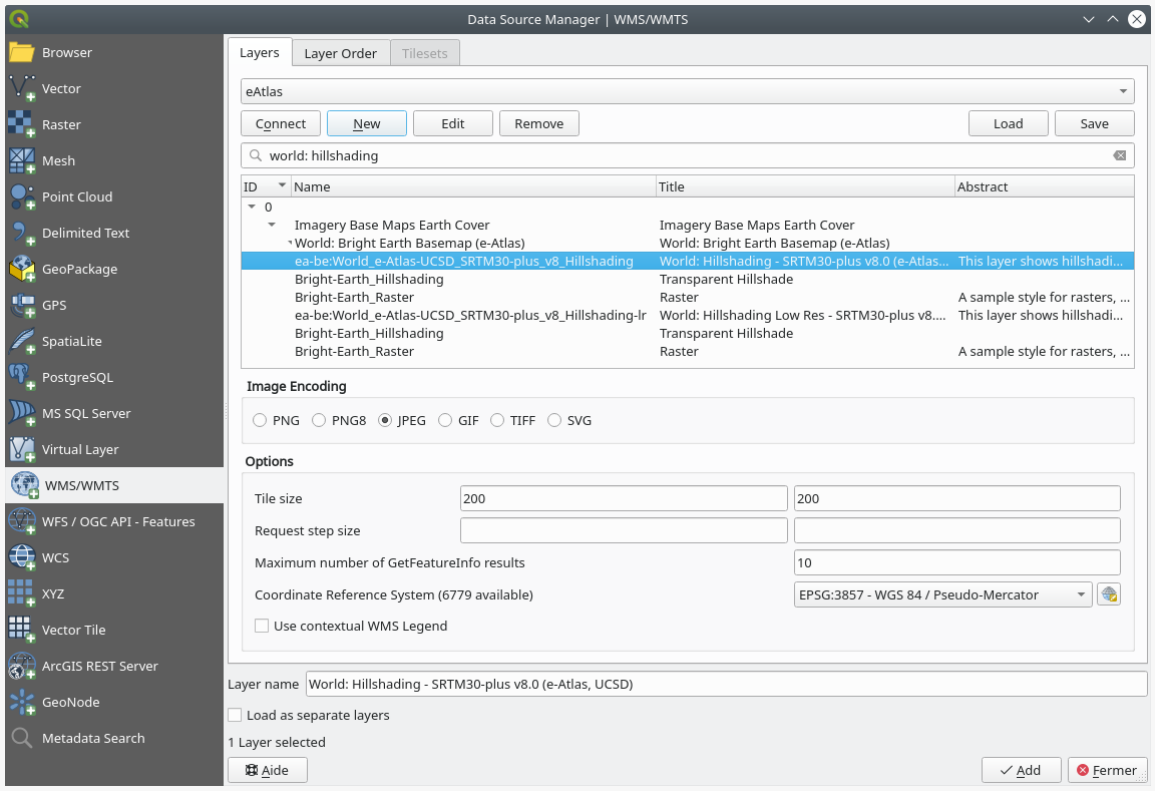
10.1.2 ★☆☆ （初級レベル） 自分でやってみよう:

1. 次の URL にある、eAtlas WMS サーバーを追加します：`https://maps.eatlas.org.au/maps/wms`
2. Load a `World: Hillshading` layer into the map.
3. その **エンコーディング** を *JPEG* に、その **タイルサイズ** オプションを `200` × `200` に設定して、読み込みを速くしたくなるかもしれません。
4. マップはこのようになるはずです（レイヤーの順序を変えたり、透過率を適用する必要があるかもしれません）：



答え

1. データソースマネージャの WMS/WMTS タブに移動し、新しい接続エントリーを作成します
2. テキストボックスを使って、レイヤのリストをフィルタし、対応するレイヤを選択します
3. その CRS を他のマップと同じように *EPSG:3857 - WGS 84 / Pseudo Mercator* にチェック/変更することを忘れないでください。



4. レイヤが読み込まれた後、その *Opacity* 値を変更することができます（**透過率** プロパティタブの下）。

10.1.3 ★★★ （上級レベル） 自分でやってみよう:

WMS を使用する難しさの一部は、良い（無料）サーバーを見つけることです。

- 新しい WMS を directory.spatineo.com （またはオンラインの他の場所）で見つけてください。それは、関連する料金や制限がなく、Swellendam の研究領域をカバーしている必要があります。

  WMS を使用するために必要なことだけでその URL（と説明の好ましいいくつかの並べ替え）であることを忘れないでください。

答え

Spatineo は、OGC データを検索できる多くの場所の一つです。テキストエリアに名前、キーワード、関心のある場所を入力し、検索結果が表示されるか確認してください。このレッスンでは、WMS のみを含むように結果をフィルタしたいと思われるかもしれません。

多くの WMS サーバーが常に利用可能ではないことにお気づきかもしれません。これは一時的な場合もあれば、永続的な場合もあります。執筆時点で動作していた WMS サーバーの例は、https://demo.mapserver.org/cgi-bin/wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities の *MapServer Demonstration Server* WMS です。これは料金を必要とせず、アクセス制限もなく、グローバルです。した

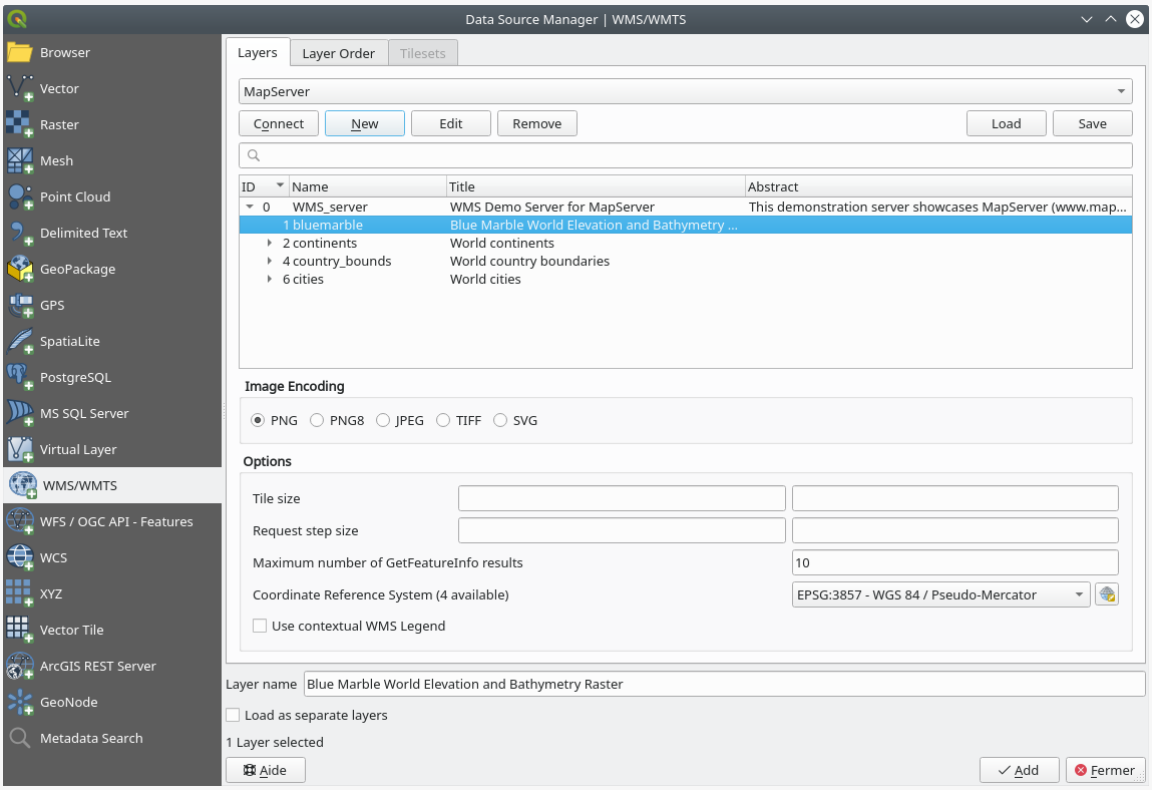
がって、要件を満たしています。しかし、これは単なる一例であることに留意してください。他にもたくさんの WMS サーバーがあります。

10.1.4 ★★★ （上級レベル） 自分でやってみよう:

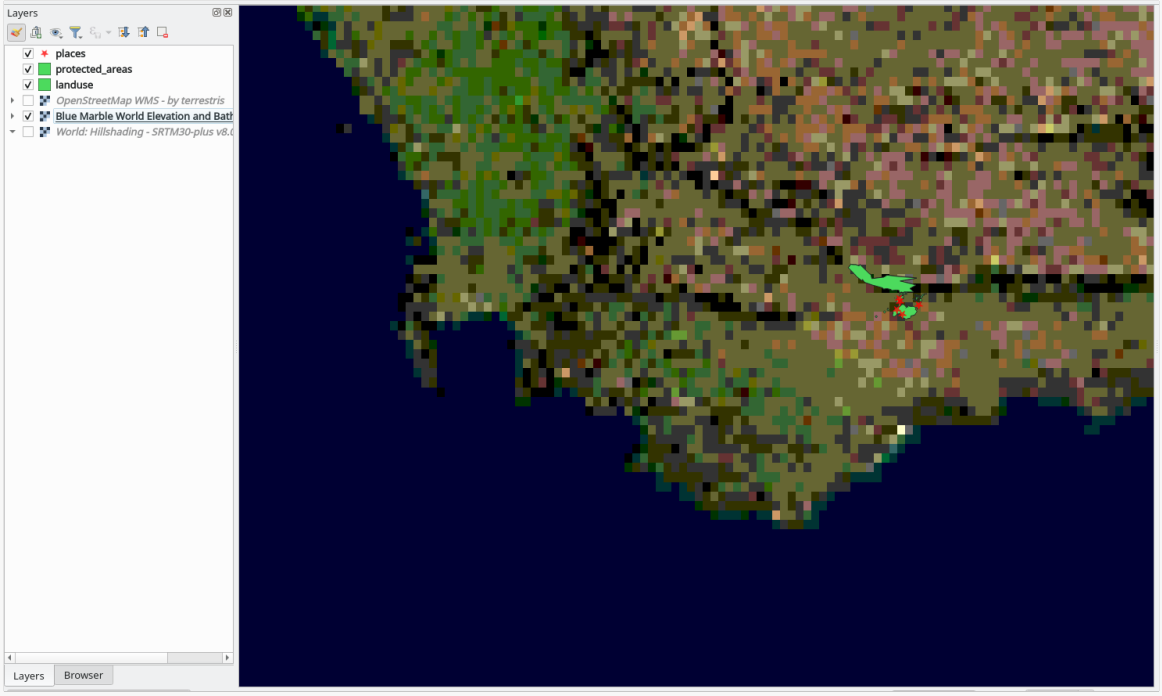
- Add the `bluemarble` layer from the *MapServer Demonstration Server*. Is that a suitable dataset for our study area?

答え

1. 他の WMS レイヤをすべて非表示にして、バックグラウンドで不必要にレンダリングされないようにします。
2. 前と同じ方法で、新しいサーバーとそのサーバー上でホストされている適切なレイヤを追加します:



3. Swellendam エリアを拡大すると、このデータセットの解像度が低いことに気づくでしょう:



そのため、今回の地図にはこのデータを使わない方がよいでしょう。Blue Marble のデータは、地球規模や国土規模での使用に適しています

10.1.5 結論

WMS を使用して、既存の地図データのための背景として、非アクティブ地図を追加できます。

10.1.6 参考文献

- Spatineo Directory
- OpenStreetMap.org list of WMS servers

10.1.7 次は?

背景として動かない地図を追加しましたが、地物（例えば以前に追加した他のベクタレイヤなど）を追加することも可能だということがわかるとうれしいでしょう。リモートサーバから地物を追加することは、ウェブ地物サービス（WFS）を使用することにより可能です。それが次のレッスンのトピックです。

10.2 レッスン: Web Feature Service

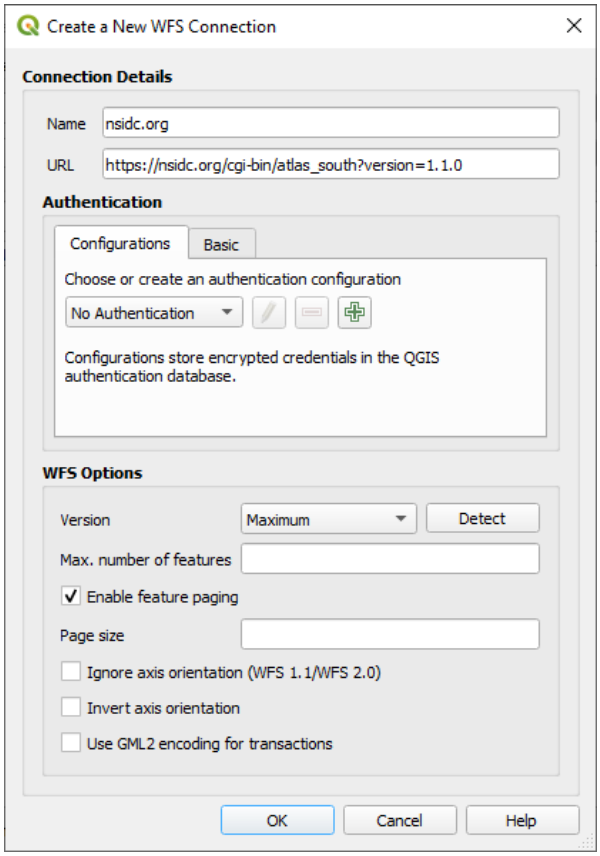
Web Feature Service (WFS) は QGIS で直接読み込める形式の GIS データをユーザーに提供します。編集できない地図のみを提供する WMS とは異なり、WFS では地物それ自体へアクセスできます。

このレッスンの目標: WFS を使用して WMS との違いを理解します。

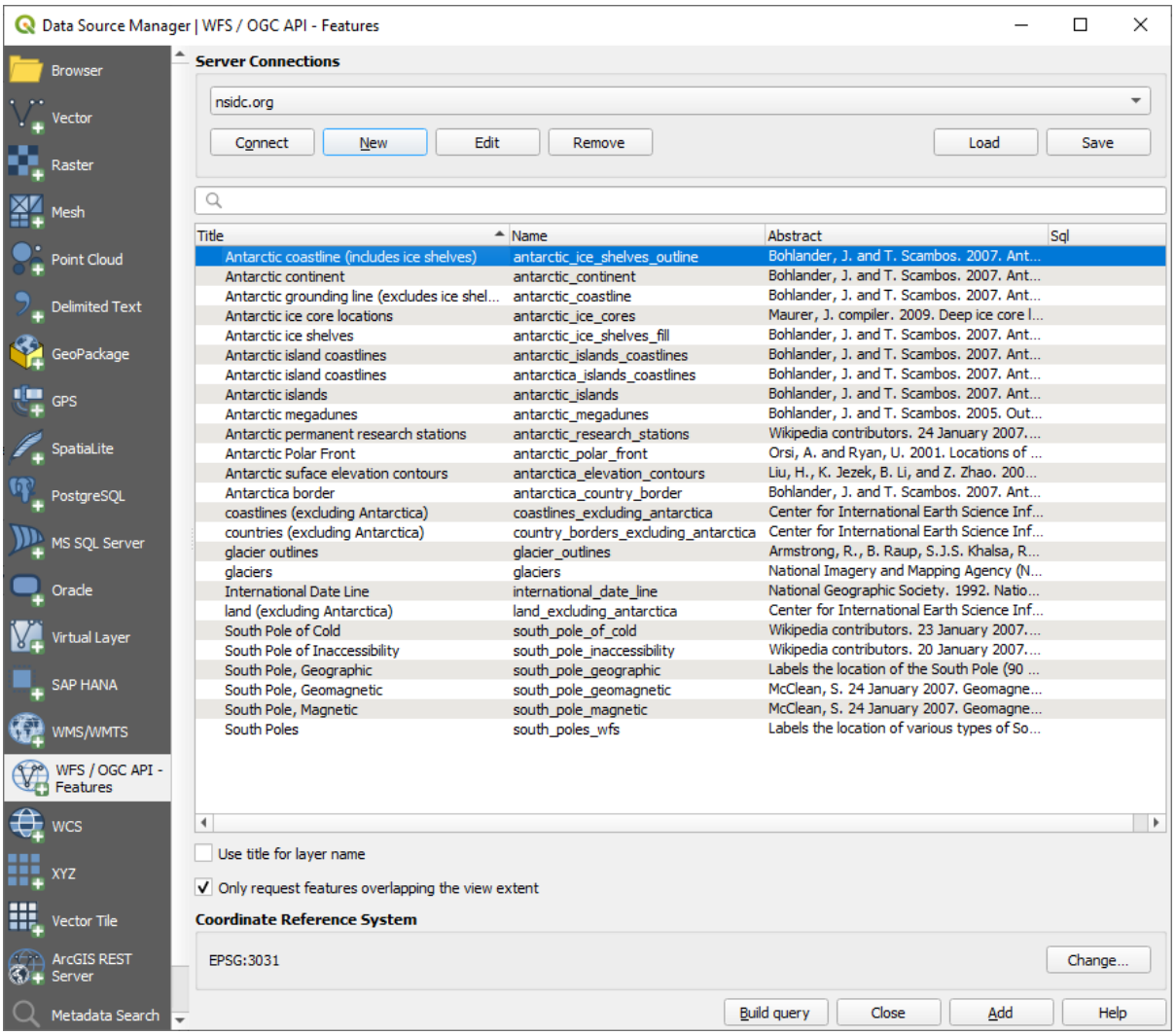
10.2.1 ★☆☆ （初級レベル） 理解しよう: WFS レイヤを読み込む

1. 新しい地図を開始します。これはデモを目的としており保存されません。
2. 

ボタンをクリックします。
3.  *WFS / OGC API* - **地物** タブを有効にします。
4. **新規** ボタンをクリックします。
5. 表示されたダイアログで、 **名前** に `nsidc.org` を、 *URL* に `https://nsidc.org/cgi-bin/atlas_south?version=1.1.0` を入力します。



6. *OK* をクリックすると新しい接続が **サーバーコネクション** に表示されます。
7. **接続** をクリックします。利用可能なレイヤのリストが表示されます:

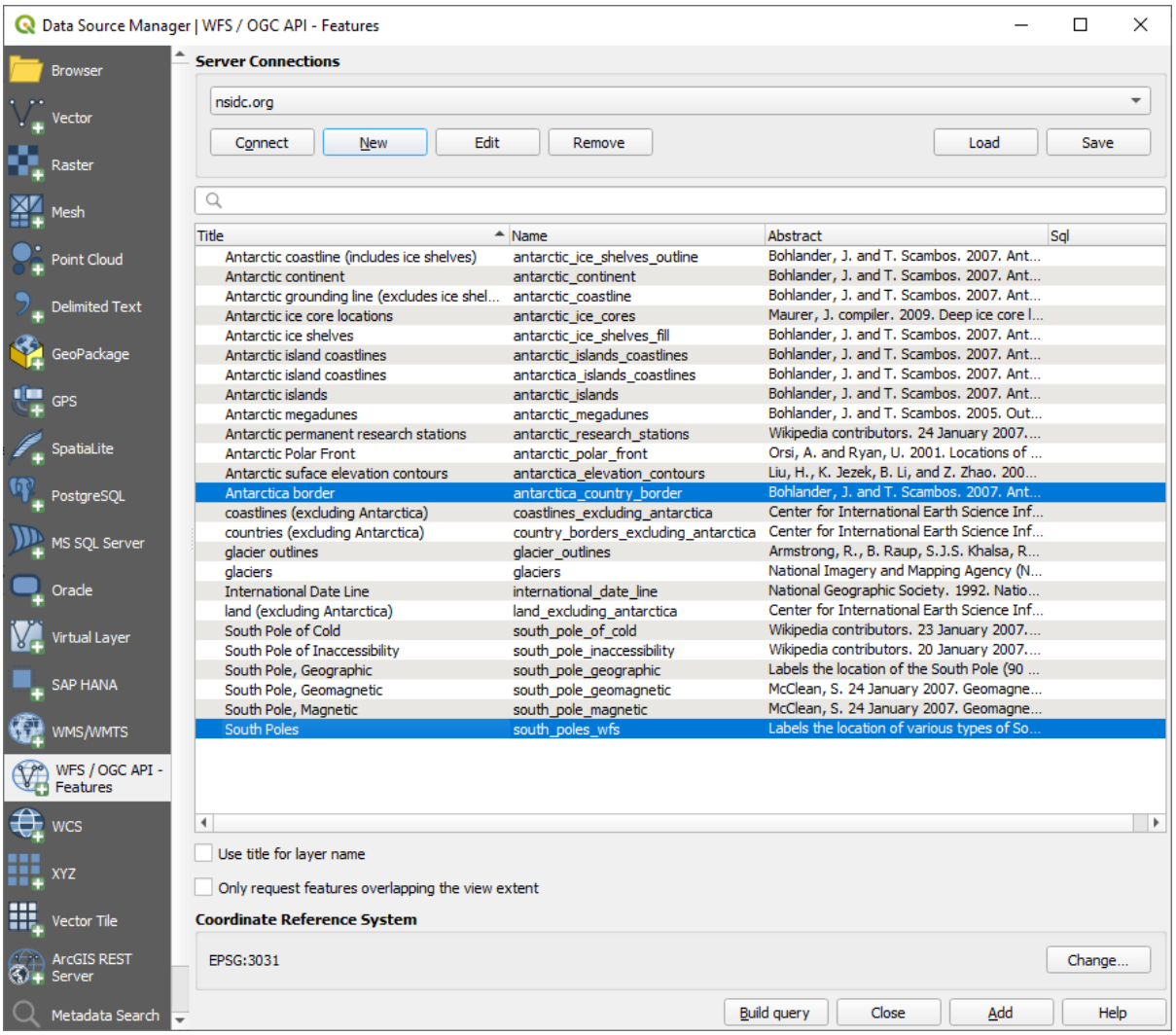


8. レイヤリストの下にある **ビュー領域に重なる地物のみをリクエストする** オプションのチェックを外します。現在のマップキャンバスが、私たちの関心領域: 南極大陸をカバーしていない可能性があるからです。

9. Find the layer `antarctica_country_border`. You can use the *Filter* box at the top.

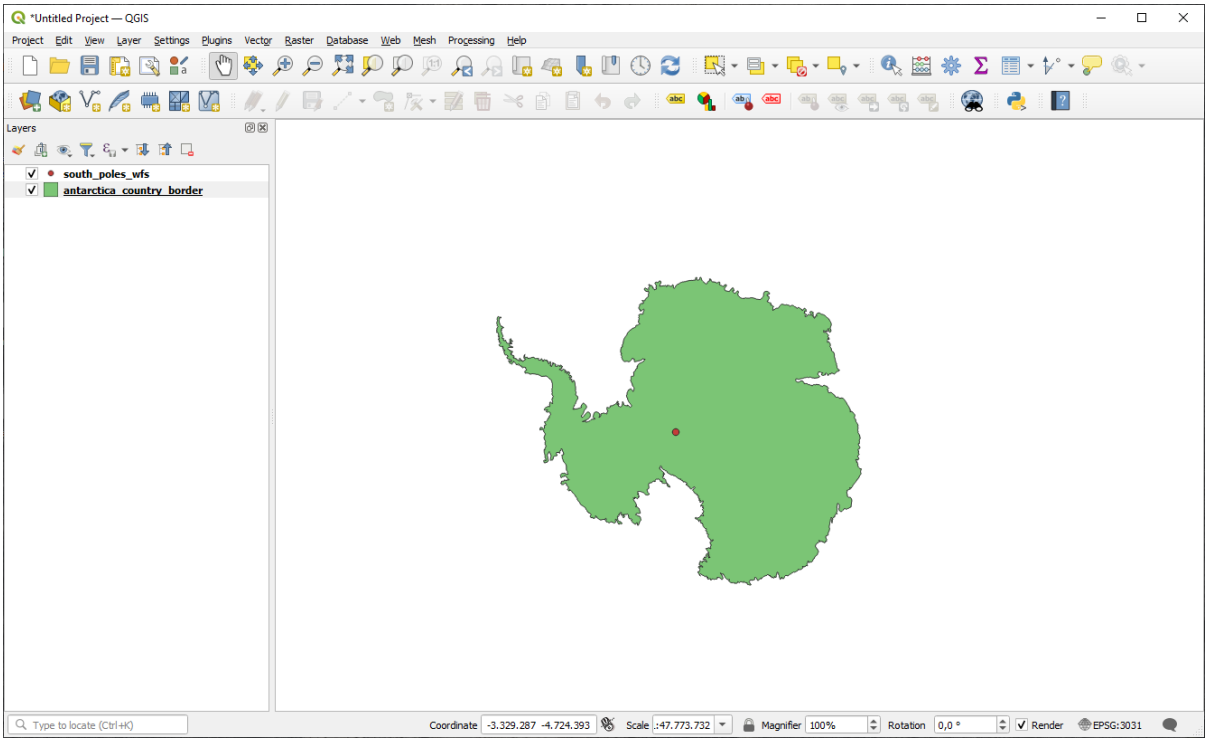
10. レイヤをクリックして選択します:

11. Find and select also the layer `south_poles_wfs`. You might need to hold `Ctrl`.



12. **追加** をクリックします。

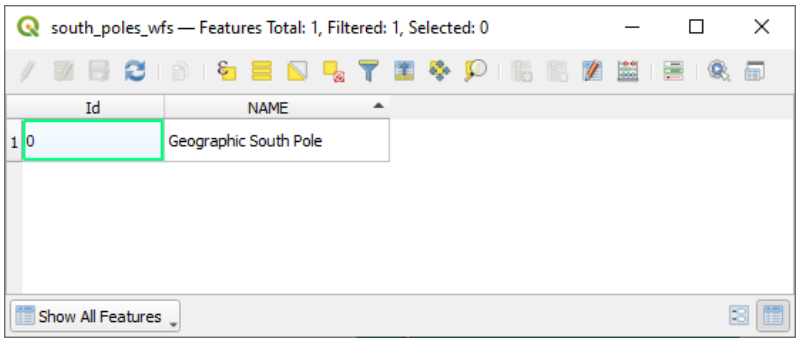
    レイヤを読み込むのにしばらく時間がかかるかもしれません。読み込みが完了すると、地図上に南極大陸の輪郭とその上にいくつかの地点が表示されます。



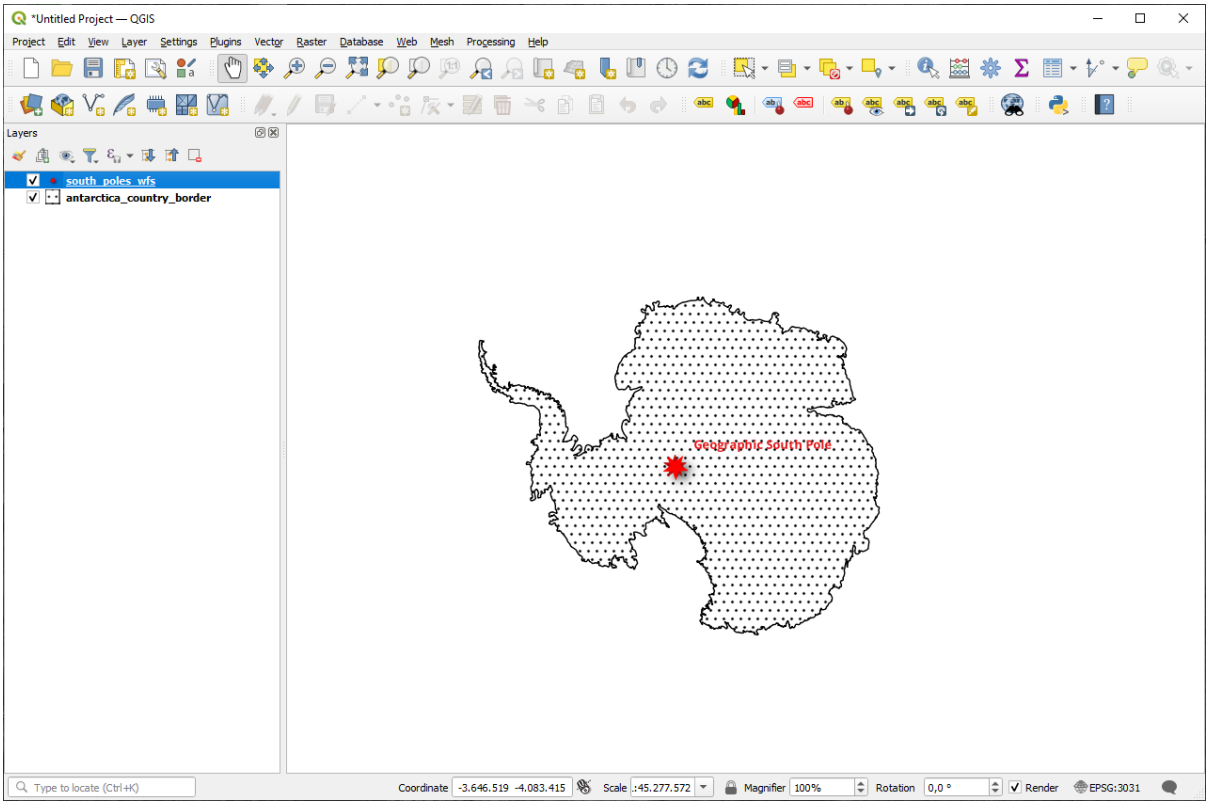
WMS のレイヤーとどう違うのでしょうか？

13. いずれかのレイヤを選択すると、地物選択ツールと属性テーブルツールが有効になることが確認できます。これらはベクタレイヤです。

14. Select the `south_poles_wfs` layer and open its attribute table. You should see this:



ポイントは属性を持つのでシンボル体系を変更し、ラベルを付けることができます。例を示します:



WMS レイヤとの違い

Web Feature Service はレンダリングされた地図ではなくレイヤ自体を返します。データへの直接アクセスができるので、シンボロジの変更やデータを分析することができます。しかし、その代償として、送信されるデータ量は格段に多くなります。レイヤが複雑な形を持つ場合や沢山の属性、多くの地物を持つ場合に特に顕著になります。多くのレイヤをロードしている場合でも同様です。このため、WFS レイヤは一般的には読み込みに非常に長い時間がかかります。

10.2.2 ★★☆ （中級レベル） 理解しよう: WFS レイヤをクエリする

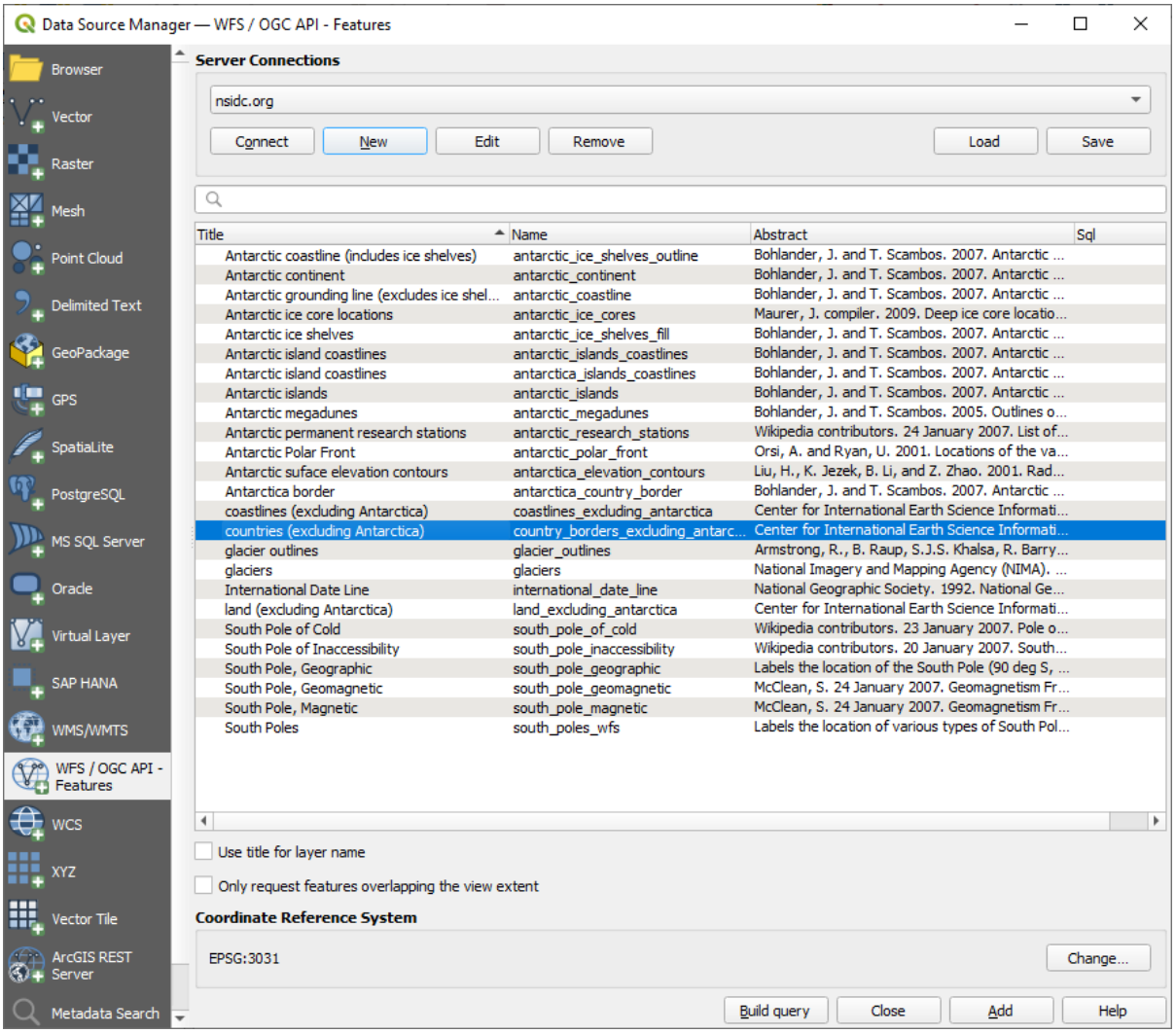
WFS レイヤをロードした後にクエリすることは可能ですが、ロードする前にクエリする方が多くの場合に効率的です。そのようにして必要な地物だけを要求すればはるかに少ない帯域幅の使用で済むことになります。

For example, on the WFS server we're currently using, there is a layer called `countries (excluding Antarctica)`. Let's say that we want to know where South Africa is relative to the `south_poles_wfs` layer (and perhaps also the `antarctica_country_border` layer) that's already been loaded.

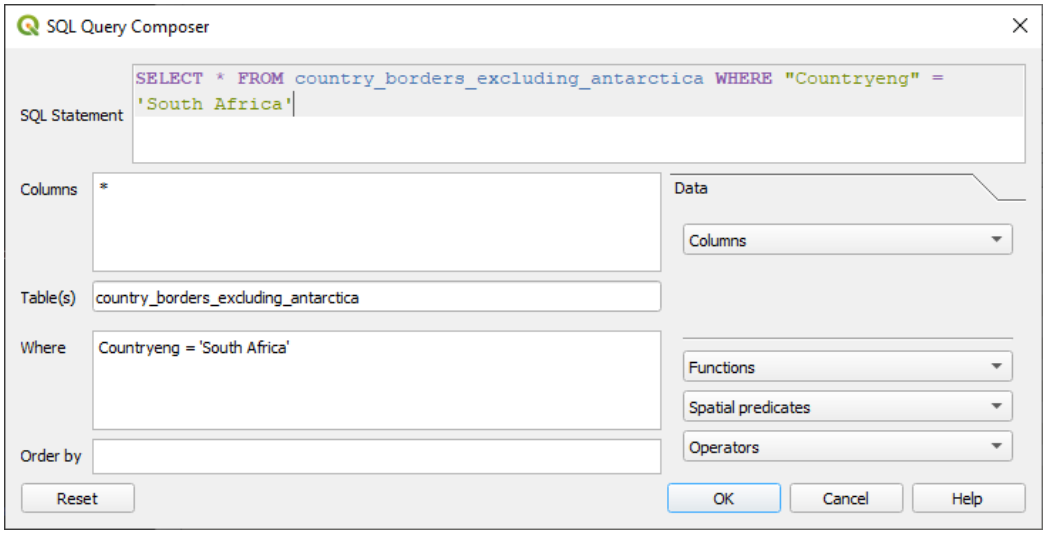
There are two ways to do this. You can load the whole `countries ...` layer, and then build a query as usual once it's loaded. However, transmitting the data for all the countries in the world and then only using the data for South Africa seems a bit wasteful of bandwidth. Depending on your connection, this dataset can take several minutes to load.

別の方法は、サーバーからレイヤを読み込む前にフィルタとしてクエリを作成することです。

1. **データソースマネージャ** ダイアログで *WFS / OGC API* - **地物** タブを有効にします
2. 先ほど使用したサーバーに接続すると、利用可能なレイヤのリストが表示されるはずです。
3. Find and double-click the `countries (excluding Antarctica)` layer. The layer name is `country_borders_excluding_antarctica`. You can also select the layer and press *Build query* button at the bottom of the dialog:

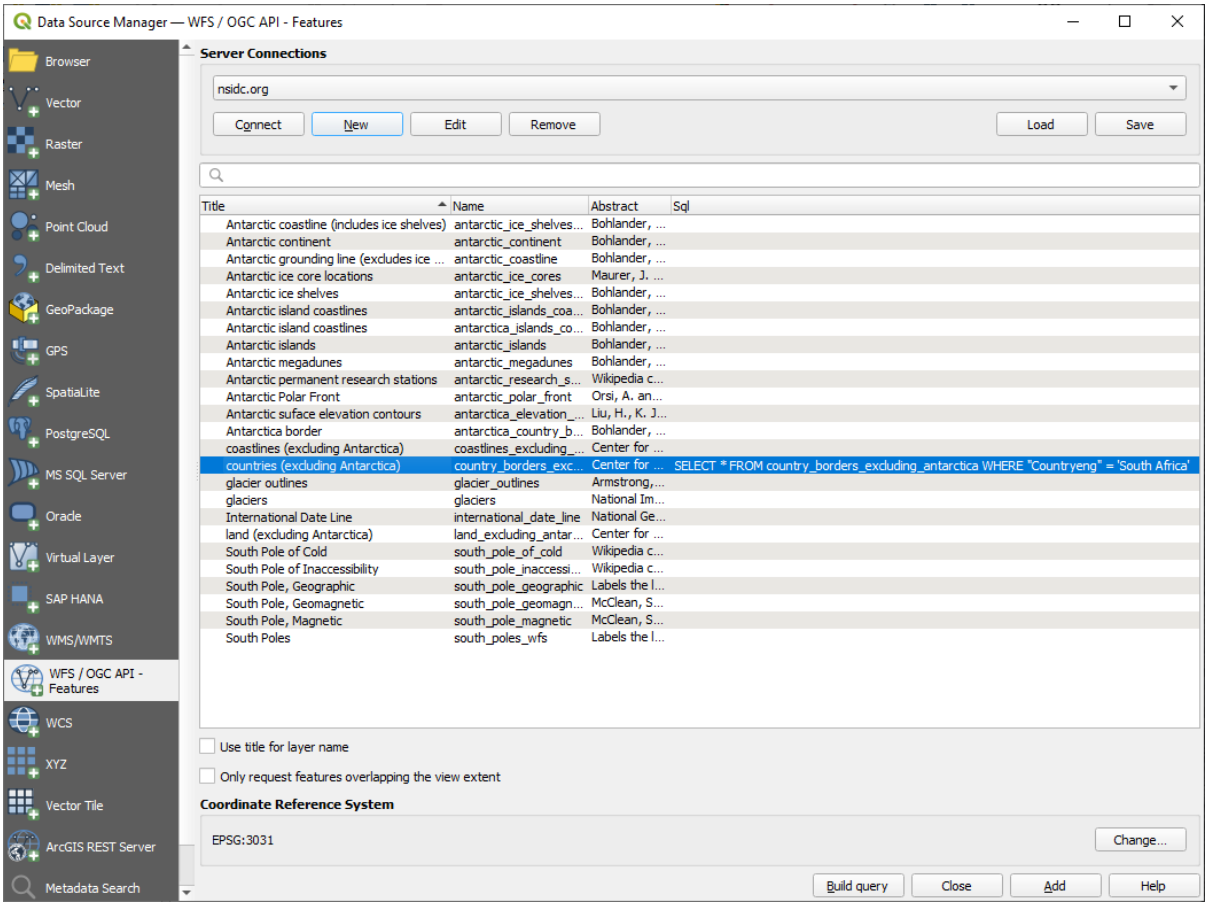


4. 表示されたダイアログで、次のクエリ `SELECT * FROM country_borders_excluding_antarctica WHERE "Countryeng" = 'South Africa'` を *SQL* **文** ボックスに入力します。

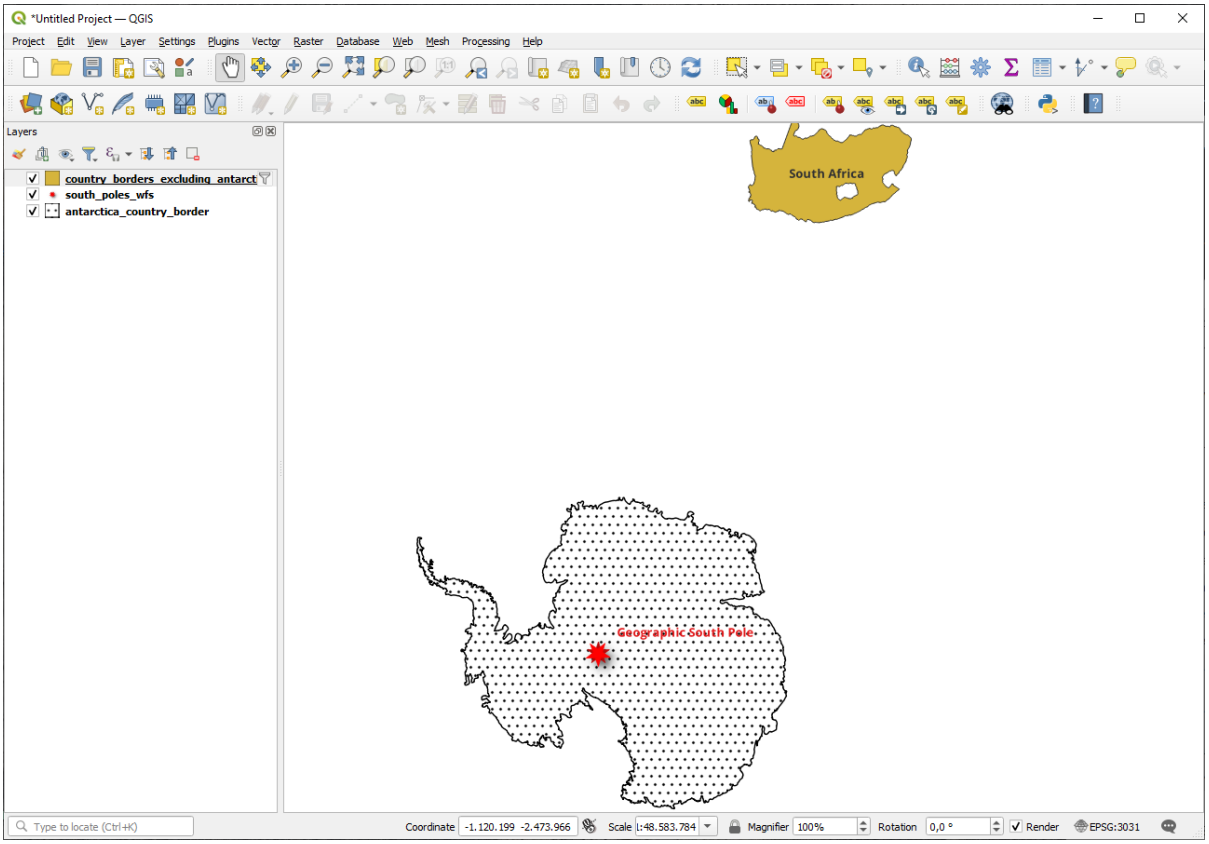


5. *OK* を押します。

6. 使用された式はターゲットレイヤの *SQL* の値として表示されます:



7. 上記のようにレイヤを選択した状態で、 **追加** をクリックします。そのレイヤから、Countryeng の値が South Africa である国だけが読み込まれます:



country_borders_excluding_antarctica レイヤの横にある  アイコンに気づきましたか？これは、読み込まれたレイヤはフィルタされており、プロジェクトにすべての地物が表示されていないことを示しています。

8. 実際にやる必要はありませんが、もし両方の方法を試してみたらフィルタする前にすべての国をロードする方法に比べてはるかに早いことがわかります!

WFS の有用性に関するノート

ニーズが非常に具体的である場合、必要とする地物を持っている WFS サーバーが見つかることは稀です。Web Feature Services が比較的まれである理由は、地物全体を表現するには大量のデータを送信する必要があるからです。それゆえに画像だけを送信する WMS ではなく WFS をホストすることは費用対効果があまりよくありません。

したがって、あなたが出合うであろう WFS の最も一般的なタイプはおそらくインターネット上ではなく、ローカルネットワーク上か自分のコンピュータ上にあるでしょう。

10.2.3 結論

レイヤの属性とジオメトリに直接アクセスする必要がある場合には WFS レイヤは WMS レイヤより好ましいですが、ダウンロードされるデータの量 (速度の問題そして容易に利用可能な公開 WFS サーバーの不足へとつながります) を考慮すると、必ずしも WMS の代わりに WFS が使用できるとは限りません。

10.2.4 次は?

次は、QGIS サーバーを使用して OGC サービスを提供する方法を説明します。

第11章　モジュール: QGIS Server

Tudor Bărăscu によって投稿されたモジュール。

このモジュールでは、QGIS サーバーをインストールして使用する方法をカバーします。

QGIS サーバーについて更に学ぶには、 QGIS-Server-manual を読んでください。

11.1 レッスン: QGIS Server をインストールする

このレッスンの目標: Debian Stretch に **QGIS Server** をインストールする方法を学ぶこと。ごくわずかな変更で、Ubuntu やその派生版のような Debian ベースのディストリビューションでもこのレッスンに従うことができます。

> **注釈**
>
> Ubuntu では、管理者権限を必要とするコマンドの前に sudo を付けることで、そのコマンドを一般ユーザーとして使用することができます。Debian では、sudo を使用せず、管理者 (root) として作業することができます。

11.1.1 ★★☆ （中級レベル） 理解しよう: パッケージからインストールする

このレッスンでは、ここ にあるパッケージからのインストールのみを行います。

QGIS Server を以下でインストールします:

```
apt install qgis-server --no-install-recommends --no-install-suggests

# if you want to install server plugins, also:
apt install python3-qgis
```

QGIS Server は、QGIS デスクトップ（付属の X サーバーと共に）が同じマシンにインストールされていない状態で、使用する必要があります。

11.1.2 ★★☆ （中級レベル） 理解しよう: QGIS Server 実行可能ファイル

QGIS Server 実行可能ファイルは、 `qgis_mapserv.fcgi` です。 `/usr/lib/cgi-bin/qgis_mapserv.fcgi` のような何かを出力する `find / -name 'qgis_mapserv.fcgi'` を実行して、どこにインストールされたかを確認できます。

オプションとして、この時点でコマンドラインテストを行いたい場合は、`/usr/lib/cgi-bin/qgis_mapserv.fcgi --version` コマンドを実行すると、次のような出力が得られるはずです：

```
QGIS 3.44.4 'Solothurn' (108203c541e)
QGIS code revision 108203c541e
Qt version 5.15.3
Python version 3.11.10
GDAL/OGR version 3.11.0
PROJ version 9.6.2
EPSG Registry database version v12.013 (2025-05-26)
GEOS version 3.13.0-CAPI-1.19.0
SQLite version 3.49.2
OS Ubuntu 24.04
```

WMS のリクエストの方法は後述します。

11.1.3 ★★★ （上級レベル） HTTP Server の構築

インストールした QGIS Server にインターネットブラウザからアクセスするためには、HTTP サーバーを使用する必要があります。Apache HTTP サーバーのインストール方法は httpserver セクションで詳しく説明しています。

注釈

（Linux デスクトップに含まれる）X Server を起動せずに QGIS Server をインストールした場合で、`GetPrint` コマンドを使いたいときは、偽の X サーバをインストールして、QGIS Server にそれを使うように指示する必要があります。これは Xvfb installation process に従って行うことができます。

11.1.4 ★★☆ （中級レベル） 理解しよう: 別の仮想ホストを作成

QGIS Server を指す別の Apache の仮想ホストを作成してみましょう。好きな名前を選択できます（`coco.bango` 、 `super.duper.training` 、 `example.com`、など）が、単純化のために `myhost` を使用するつもりです。

- `/etc/hosts` に `127.0.0.1 x` を追加して `myhost` という名前がローカルホストの IP を指すように次のコマンドで設定するか： `sh -c "echo '127.0.0.1 myhost' >> /etc/hosts"` または `gedit /etc/hosts` でファイルを手動で編集してください。
- ターミナルで `ping myhost` コマンドを実行することによって、 `myhost` が localhost を指していることを確認できます。

```
qgis@qgis:~$ ping myhost
PING myhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.024 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.029 ms
```

- curl http://myhost/cgi-bin/qgis_mapserv.fcgi したり、Debian のボックスブラウザから URL にアクセスすることによって、 myhost サイトから QGIS Server にアクセスできるかどうか試してみましょう。おそらく得られるのは：

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /cgi-bin/qgis_mapserv.fcgi was not found on this server.</p>
<hr>
<address>Apache/2.4.25 (Debian) Server at myhost Port 80</address>
</body></html>
```

- Apache は myhost という名前のサーバーを指すリクエストに答えることになっていることを知りません。バーチャルホストを設定するのに一番簡単な方法は、 /etc/apache2/sites-available ディレクトリに myhost.conf ファイルを作ることです。このファイルの内容は qgis.demo.conf と同じですが、 ServerName 行は ServerName myhost にする必要があります。またこれは任意ですが、ログを保存する場所を変更することもできます。そうしないときは、2 つのバーチャルホストのログが共有されます。
- 次に、a2ensite myhost.conf でバーチャルホストを有効にし、service apache2 reload で Apache サービスを再ロードしてみましょう。
- 再度 http://myhost/cgi-bin/qgis_mapserv.fcgi URL にアクセスするために試してみると、すべてが今稼働していることに気づくでしょう！

11.1.5 結論

Debian ベースの Linux ディストリビューション上で、QGIS Server で Apache を設定する方法を、パッケージから異なる QGIS Server のバージョンをインストールする方法を学びました。

11.1.6 次は?

QGIS Server をインストールし、HTTP プロトコルでアクセスできるようになったので、提供できるサービスのいくつかにアクセスする方法を学ぶ必要があります。次のレッスンのテーマは、QGIS Server の WMS サービスにアクセスする方法を学ぶことです。

11.2 レッスン: WMS サーバーを運用する

この演習に使用されるデータは、ダウンロードした *training data* の `qgis-server-tutorial-data` サブディレクトリにあります。 便宜上、またパーミッションの問題を回避するために、これらのファイルは `/home/qgis/projects` ディレクトリに保存されていると想定します。したがって、次の手順をあなたのパスに適合させてください。

デモデータには、QGIS サーバーで提供されるように既に準備されている `world.qgs` という名前の QGIS プロジェクトが含まれています。独自のプロジェクトを使用したい場合や、プロジェクトの準備方法を知りたい場合は、 Creatingwmsfromproject セクションを参照してください。

注釈

このモジュールは、オーディエンスがパラメーターとパラメーターの値を簡単に区別できるように URL を提示します。通常のフォーマットは次のとおりです。

```
...&field1=value1&field2=value2&field3=value3
```

このチュートリアルで使用するのは：

```
&field1=value1
&field2=value2
&field3=value3
```

それらを Mozilla Firefox に貼り付けると適切に機能しますが、Chrome のような他の Web ブラウザでは `field:parameter` のペアの間に不要なスペースを追加する可能性があります。したがって、この問題が発生した場合は、Firefox を使用するか、URL を変更して 1 行の形式にすることができます。

Web ブラウザまたは curl で WMS GetCapabilities リクエストを作成しましょう：

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetCapabilities
&map=/home/qgis/projects/world.qgs
```

前のレッスンの Apache 設定では、 `QGIS_PROJECT_FILE` 変数はデフォルトプロジェクトを `/home/qgis/projects/world.qgs` に設定します。しかし、上記のリクエストでは、 **map** パラメーターを明示的に使用し、どのプロジェクトでも使用できることを示しました。上記の要求から **map** パラメーターを削除すると、QGIS サーバーは同じ応答を出力します。

WMS クライアントを `GetCapabilities` の URL に指すことで、クライアントは Web マップサーバーの情報のメタデータを持つ XML 文書を返します。例えば、レイヤーはどのレイヤーに対応していますか、地理的なカバレッジ、形式、WMS のバージョン等

QGIS は ogc-wms でもあるので、上記の GetCapabilities の URL を利用して新しい WMS サーバー接続を作成できます。 **レッスン**: *Web Mapping Service* または ogc-wms-servers セクションを参照してください。

QGIS プロジェクトに `countries` WMS レイヤーを追加すると、以下のような画像が得られます：

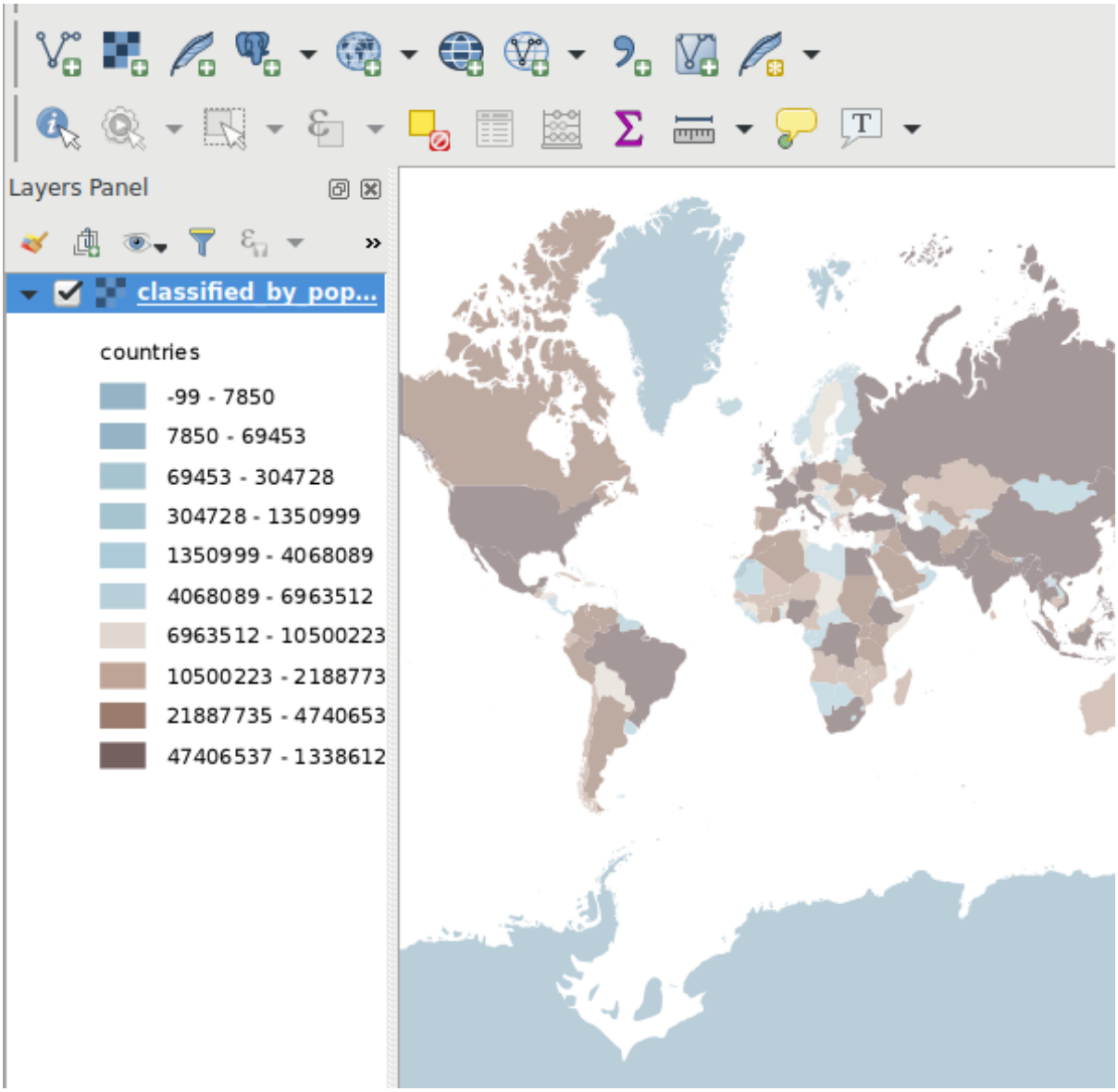


図 11.1: QGIS サーバーの国レイヤー WMS サービスを消費する QGIS デスクトップ

注釈

QGIS サーバーは `world.qgs` プロジェクトで定義されたレイヤーを提供します。QGIS でプロジェクトを開くと、各国のレイヤーに複数のスタイルがあることがわかります。QGIS サーバーはこれも認識しており、要求に応じてスタイルを選択できます。上記の画像では `classified_by_population` スタイルが選択されています。

11.2.1 ログ出力

サーバーを設定するときは、ログは常に何が起こっているかを示す重要なものです。 *.conf ファイルに以下のログを設定します：

- /logs/qgisserver.log にある QGIS サーバーログ
- qgisplatform.demo.access.log にある qgisplatform.demo Apache アクセスログ
- qgisplatform.demo.error.log にある qgisplatform.demo Apache エラーログ

ログファイルはテキストファイルなので、テキストエディタを使用してチェックできます。 sudo tail -f/logs/qgisserver.log という端末で tail コマンドを使うこともできます。

これは、そのログファイルに書き込まれたものを端末に出力し続けます。次のように、ログファイルごとに 3 つの端末を開くこともできます。

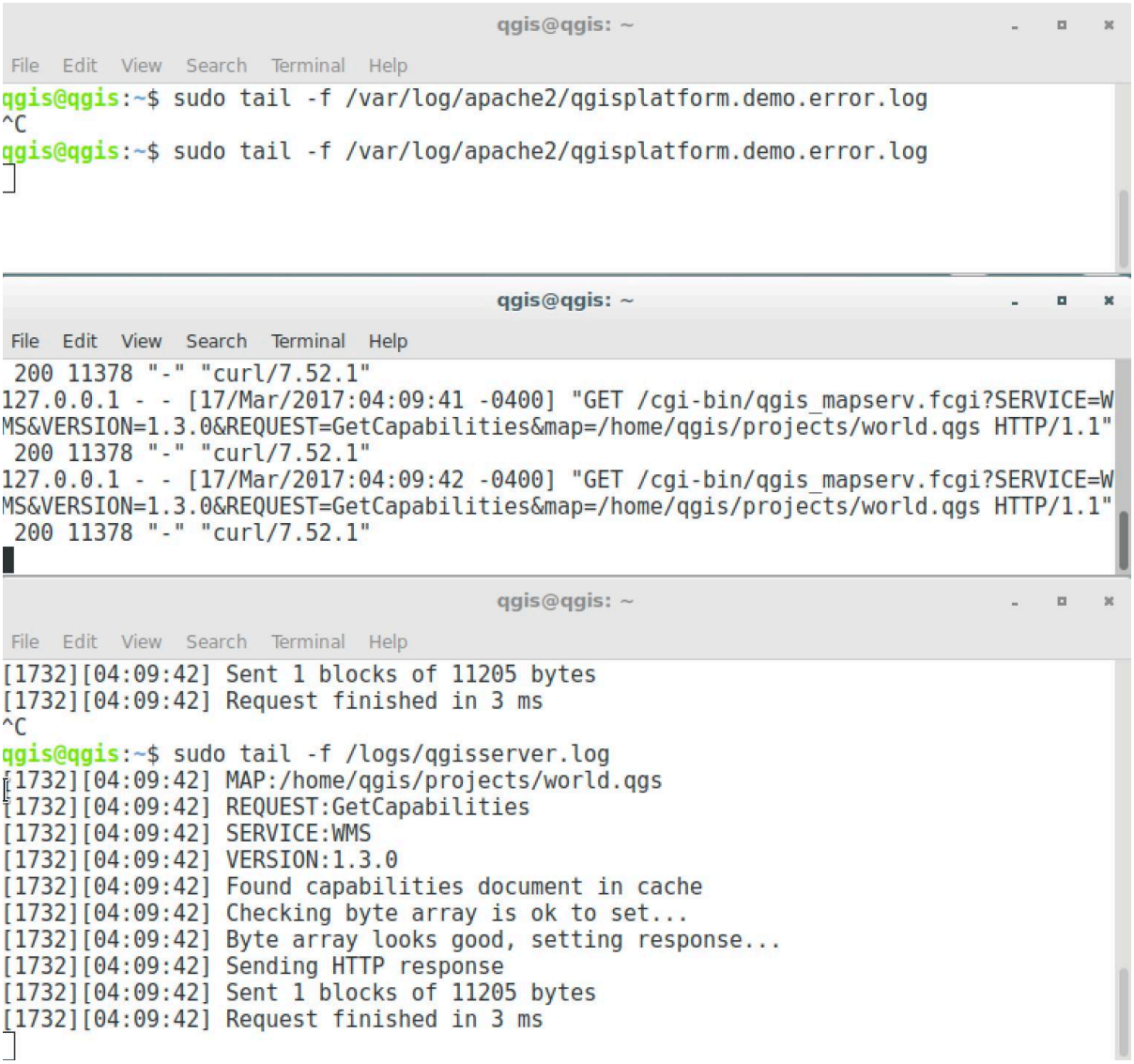


図 11.2: tail コマンドを使って QGIS サーバーのログ出力を視覚化する

QGIS デスクトップを使用して QGIS サーバーの WMS サービスを使用すると、QGIS がアクセスログ内の

サーバーに送信するすべての要求、QGIS サーバーログの QGIS サーバーのエラーなどが表示されます。

注釈

- 次のセクションのログを見ると、何が起きているのかをよりよく理解することができます。
- QGIS サーバーのログを見ながら Apache を再起動することで、動作の仕方についてのいくつかの追加情報を見つけることができます。

11.2.2 GetMap リクエスト

`countries` レイヤーを表示するために、他の WMS クライアントと同様、QGIS デスクトップでも `GetMap` リクエストを使用しています。

簡単なリクエストは以下のようになります：

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?MAP=/home/qgis/projects/world.qgs
&SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&BBOX=-432786,4372992,3358959,7513746
&SRS=EPSG:3857
&WIDTH=665
&HEIGHT=551
&LAYERS=countries
&FORMAT=image/jpeg
```

上記の要求は次の画像を出力するはずです：

図：QGIS サーバーへの簡単な GetMap リクエスト

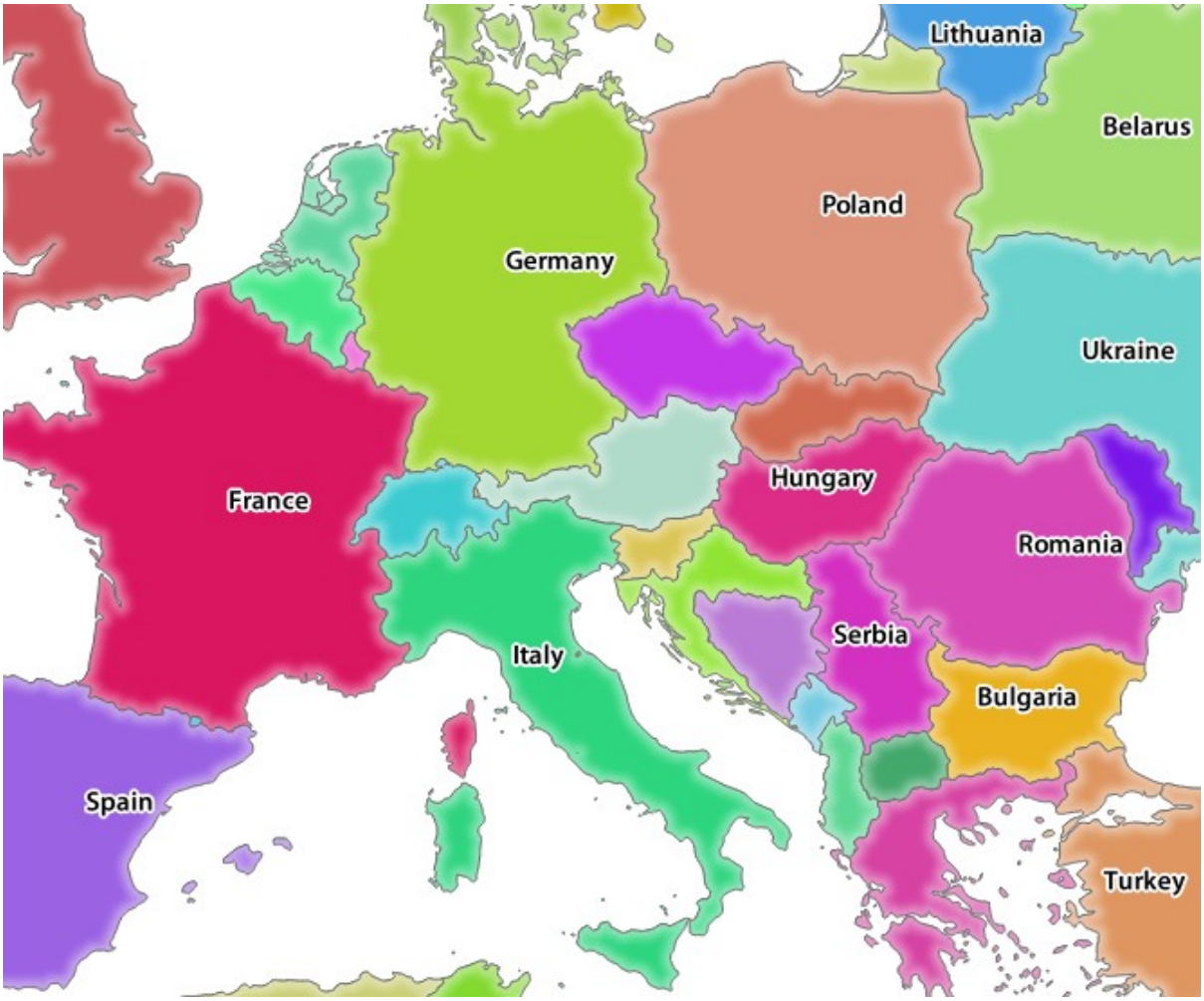


図 11.3: 単純な GetMap 要求の後の Qgis サーバーの応答

11.2.3 ★★☆ （中級レベル） 自分でやってみよう: 画像とレイヤのパラメータを変更する

上記の要求に基づいて、 `countries` レイヤを別のものに置き換えましょう。

他にどのレイヤーがあるのか　　を知るためには、QGIS の `world.qgs` プロジェクトを開き、その内容を見てください。ただし、WMS クライアントは QGIS プロジェクトへのアクセス権を持たず、機能文書の内容を参照するだけです。

また、QGIS プロジェクトに存在するいくつかのレイヤーが WMS サービスを提供する際に QGIS によって無視されるように構成オプションがあります。

したがって、QGIS デスクトップを `GetCapabilities` の URL に向けるとレイヤーリストを見ることができ、 `GetCapabilities` XML レスポンスで他のレイヤー名を見つけることができます。

あなたが見つけて働くことができるレイヤー名の 1 つは `countries_shapeburst` です。他を見つけるかもしれませんが、そのような小縮尺では見えないかもしれないので、空白の画像をレスポンスとして得ることがあることをご承知おきください。

返された画像タイプを image/png に変更するなど、上からの他のパラメーターを使って遊ぶこともできます。

11.2.4 ★★☆ （中級レベル） 理解しよう: フィルタ、不透明度、スタイルのパラメータを使う

別のレイヤー、いくつかの 基本パラメーター, **FILTER** and **OPACITIES** を追加するが、標準の STYLES パラメーターも使用する別のリクエストを実行してみましょう。

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?MAP=/home/qgis/projects/world.qgs
&SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&BBOX=-432786,4372992,3358959,7513746
&SRS=EPSG:3857
&WIDTH=665
&HEIGHT=551
&FORMAT=image/jpeg
&LAYERS=countries,countries_shapeburst
&STYLES=classified_by_name,blue
&OPACITIES=255,30
&FILTER=countries:"name" IN ( 'Germany' , 'Italy' )
```

上記の要求は次の画像を出力するはずです：

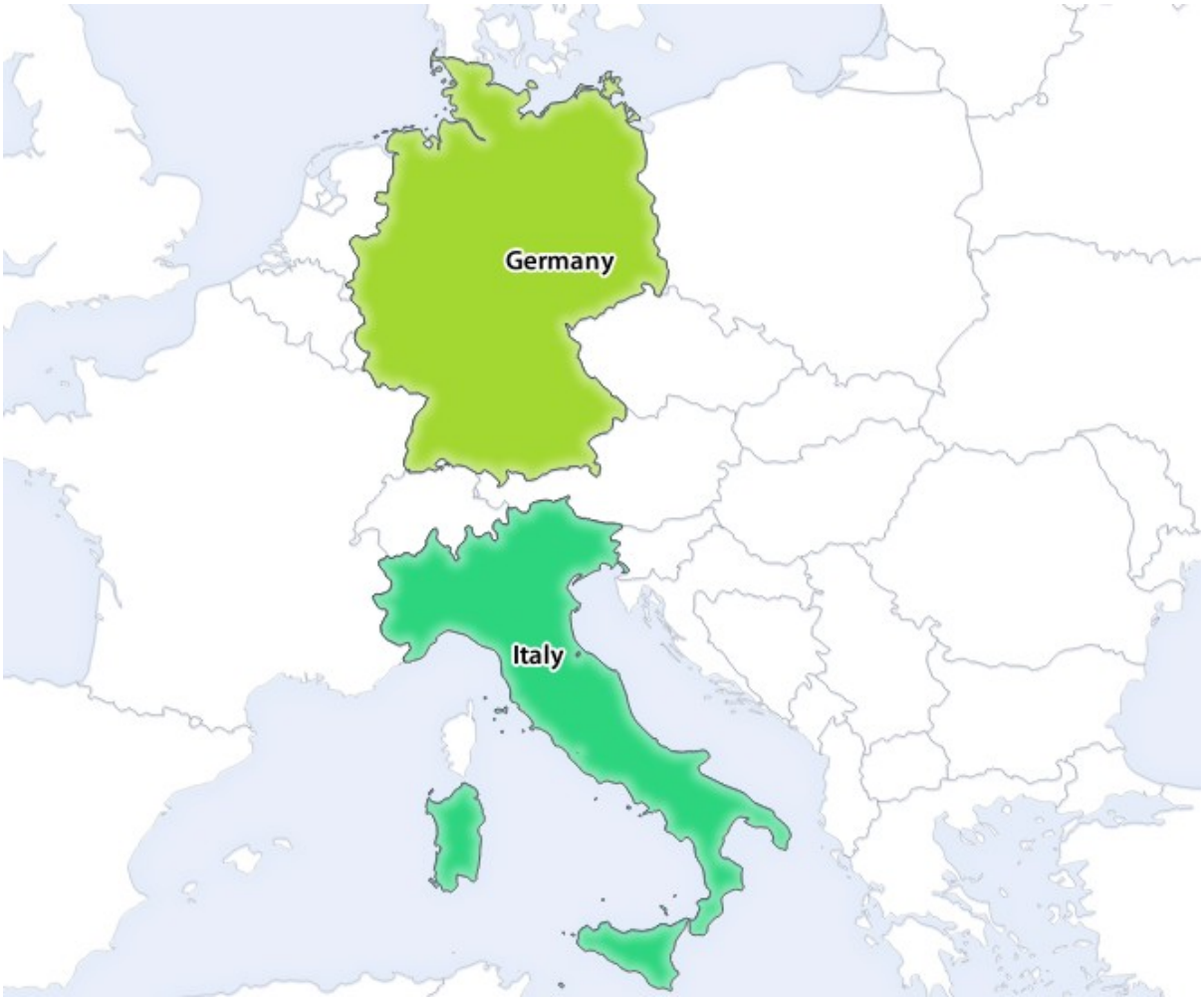


図 11.4: FILTER パラメーターと OPACITIES パラメーターを使用した GetMap 要求への応答

上の画像からわかるように、QGIS サーバーには、 **Germany** と **Italy** のみを国レイヤーからレンダリングするように指示しました。

11.2.5 ★★☆ （中級レベル） 理解しよう: レッドラインを使う

redlining 機能と Basics セクションで詳しく説明されている **SELECTION** パラメータを利用した別の GetMap リクエストを実行してみましょう：

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?MAP=/home/qgis/projects/world.qgs
&SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&BBOX=-432786,4372992,3358959,7513746
&SRS=EPSG:3857
&WIDTH=665
&HEIGHT=551
```

(次のページに続く)

(前のページからの続き)

```
&LAYERS=countries,countries_shapeburst
&FORMAT=image/jpeg
&HIGHLIGHT_GEOM=POLYGON((590000 6900000, 590000 7363000, 2500000 7363000, 2500000␣
↪6900000, 590000 6900000))
&HIGHLIGHT_SYMBOL=<StyledLayerDescriptor><UserStyle><Name>Highlight</Name>
↪<FeatureTypeStyle><Rule><Name>Symbol</Name><LineSymbolizer><Stroke><SvgParameter␣
↪name="stroke">%233a093a</SvgParameter><SvgParameter name="stroke-opacity">1</
↪SvgParameter><SvgParameter name="stroke-width">1.6</SvgParameter></Stroke></
↪LineSymbolizer></Rule></FeatureTypeStyle></UserStyle></StyledLayerDescriptor>
&HIGHLIGHT_LABELSTRING=QGIS Tutorial
&HIGHLIGHT_LABELSIZE=30
&HIGHLIGHT_LABELCOLOR=%23000000
&HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFF
&HIGHLIGHT_LABELBUFFERSIZE=3
&SELECTION=countries:171,65
```

上記のリクエストを Web ブラウザに貼り付けると、次の画像が出力されます。



図 11.5: REDLINING 機能と SELECTION パラメーターによるリクエストへの応答

上記の画像から、171 ID と 65 ID を持つ国が **SELECTION** パラメーターを使用して黄色（ルーマニアとフランス）に強調表示され、 **REDLINING** 機能を使用して矩形を **QGIS チュートリアル** ラベル。

11.2.6 GetPrint リクエスト

QGIS サーバーの非常に優れた機能の 1 つは、QGIS デスクトップの印刷レイアウトを利用することです。wms_getprint セクションでそれについて学ぶことができます。

QGIS デスクトップで `world.qgs` プロジェクトを開くと、「人口分布」という名前の印刷レイアウトが見つかります。この驚くべき機能を例示する単純化された `GetPrint` リクエストは次のとおりです。

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi
?map=/home/qgis/projects/world.qgs
&SERVICE=WMS
&VERSION=1.3.0&
REQUEST=GetPrint
&FORMAT=pdf
&TRANSPARENT=true
&SRS=EPSG:3857
&DPI=300
&TEMPLATE=Population distribution
&map0:extent=-432786,4372992,3358959,7513746
&LAYERS=countries
```

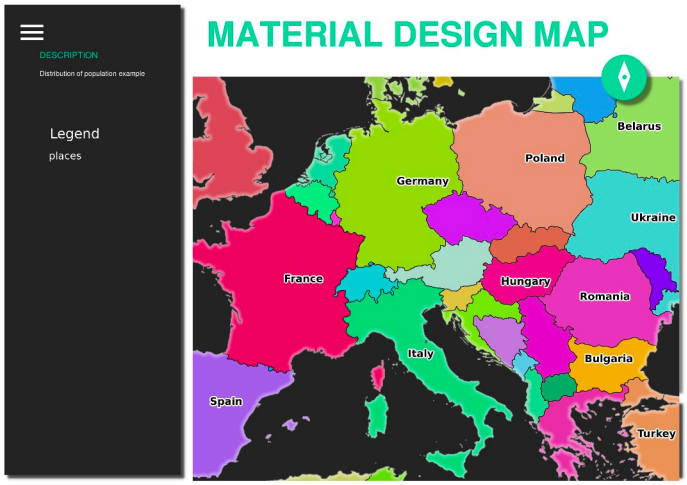


図 11.6: 上記の GetPrint リクエストに起因する pdf を表示します。

当然、あなたの `GetMap` 、 `GetPrint` などのリクエストを書くのは難しいです。

QGIS Web クライアント または QWC は、Web クライアントプロジェクトであり、QGIS サーバーと連携してプロジェクトを Web 上に公開したり、可能性についてより良く理解するために、QGIS サーバーリクエストの作成を手助けできます。

このようにインストールできます：

- ユーザ qgis が cd/home/qgis でホームディレクトリに行きます。
- ここ から QWC プロジェクトをダウンロードし、解凍します。
- /var/www/html ディレクトリへのシンボリックリンクは、バーチャルホストの設定で設定した DocumentRoot のようにしてください。アーカイブを以下のファイルで解凍した場合 /home/qgis/Downloads/QGIS-Web-Client-master を実行すると、sudo ln -s /home/qgis/Downloads/QGIS-Web-Client-master/var/www/html/ となります。
- ウェブブラウザから http://qgisplatform.demo/QGIS-Web-Client-master/site/qgiswebclient.html?map=/home/qgis/projects/world.qgs にアクセスしてください。

これで、次の図のように地図を表示できるはずです：



図 11.7: world.qgs プロジェクトを使用する QGIS Web クライアント

QWC の [印刷] ボタンをクリックすると、対話的に GetPrint 要求を作成できます。また QWC の ？ アイコンをクリックすると、利用可能なヘルプにアクセスして、QWC の可能性をよりよく知ることができます。

11.2.7 結論

QGIS サーバーを使って WMS サービスを提供する方法を学びました。

11.2.8 次は?

次に、有名な GRASS GIS のフロントエンドとして QGIS を使用する方法を見ていきます。

第12章　モジュール: GRASS

GRASS (Geographic Resources Analysis Support System 地理資源分析支援システム) は、幅広く便利な GIS 機能を持つオープンソース GIS として知られています。1984 に初めてリリースされ、それ以来、多くの改善や追加機能が見られました。QGIS では、パワフルな GIS ツールとして GRASS を直接利用できます。

12.1 レッスン: GRASS のセットアップ

QGIS で GRASS を使用するにはインターフェイスを少し異なる方法で考える必要があります。QGIS で直接作業しているのではなく QGIS **を通じて** GRASS で作業していることを覚えておいて下さい。したがって、Grass をサポートする QGIS Desktop がインストールされていることを確認してください。

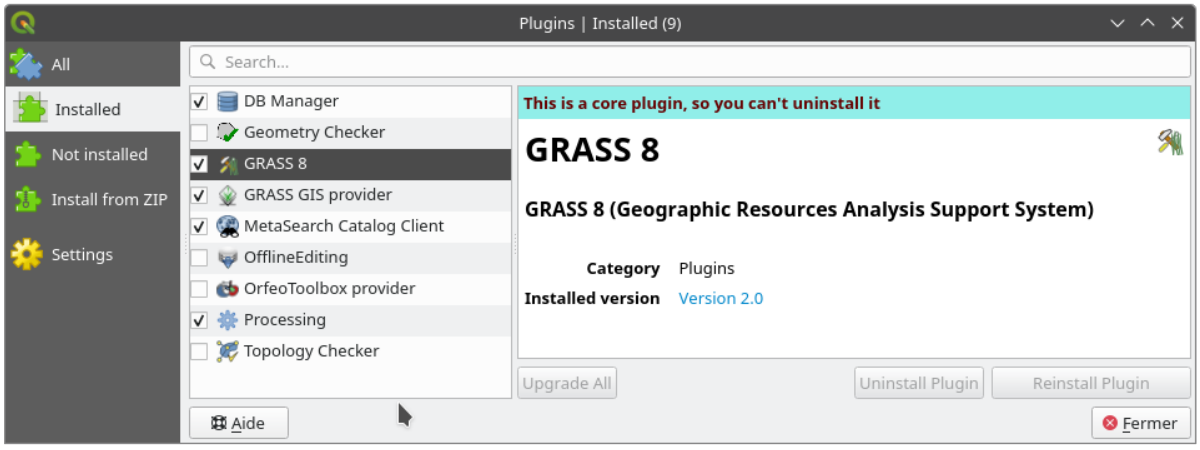
Windows で利用可能な GRASS で QGIS セッションを開くには、「QGIS デスクトップと GRASS」アイコンをクリックする必要があります。

このレッスンの目標: QGIS で GRASS のプロジェクトを始めます。

12.1.1 ★☆☆ （初級レベル） 理解しよう: 新しい GRASS セッションを始める

QGIS から GRASS を起動するには、他のプラグインと同様に有効化する必要があります:

1. まず新しい QGIS プロジェクトを開いてください。
2. **プラグインマネージャ** でリストにある *GRASS 8* を有効にします:



GRASS ツールバーと GRASS パネルが現れます:

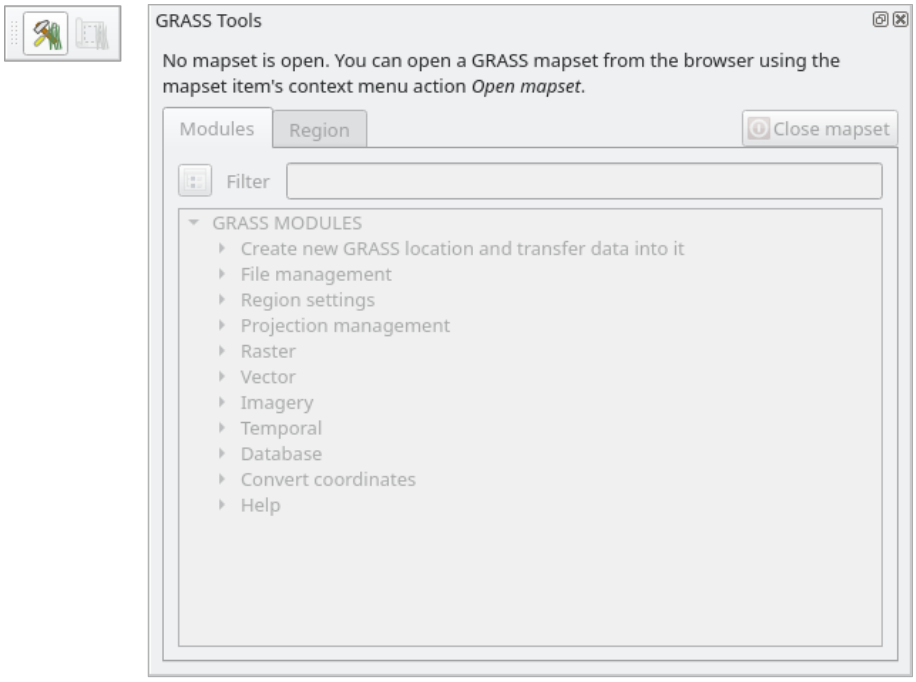


図 12.1: GRASS ツールバーとパネル

GRASS パネルはアクティブではありませんが、それは GRASS を使う前に マップセット を作成する必要があるためです。GRASS は常にデータベース環境で動作するので、使うすべてのデータは GRASS データベースにインポートする必要があります。

GRASS データベースは一見とても複雑に見えますが、構造は単純です。知っておくべき最も重要なことは、データベースの上位レベルは `Location` だということです。各 `Location` には異なる マップセット が含まれます：**すべての** マップセット 中には、GRASS がデフォルトで作成する、`PERMANENT` マップセットがあります。各 マップセット には特定の構造でデータ（ラスタ、ベクタなど）が含まれますが、心配する必要はありません。GRASS がこれを処理してくれます。

これだけ覚えてください： `Location` はデータを含んでいる `Mapset` を含みます。詳細な情報は GRASS website をご覧ください。

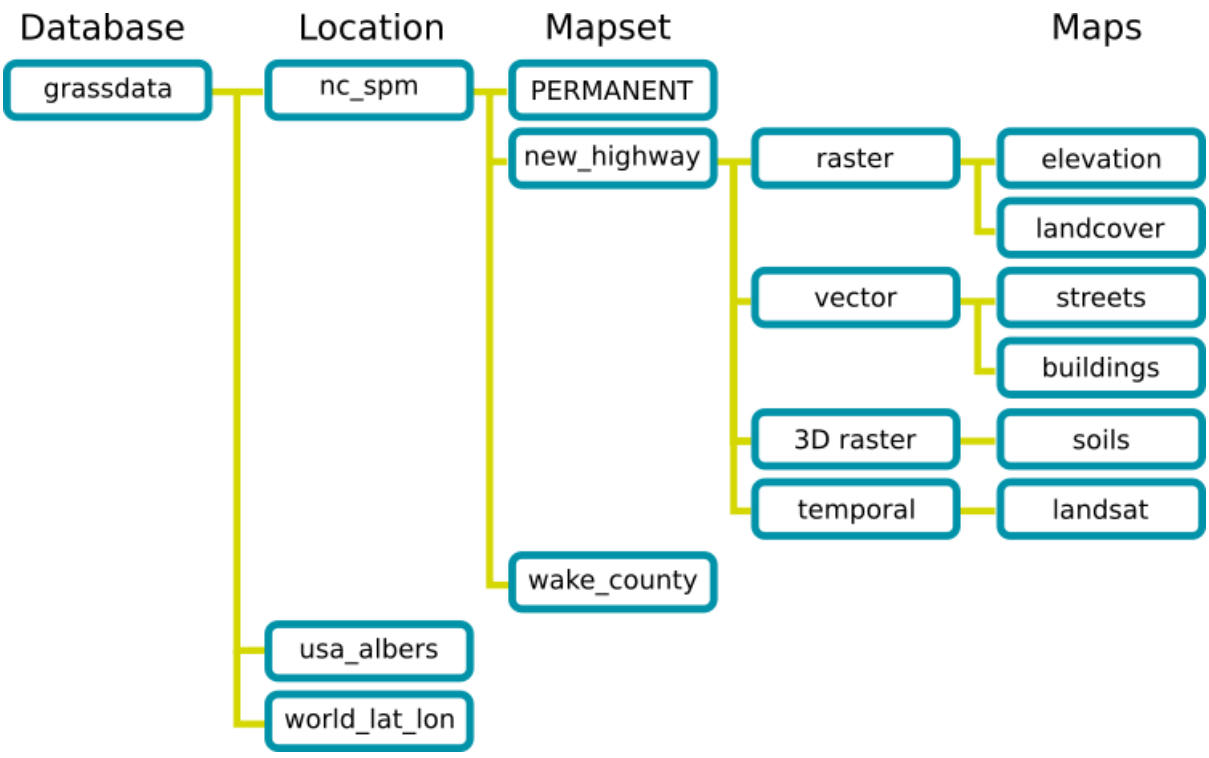
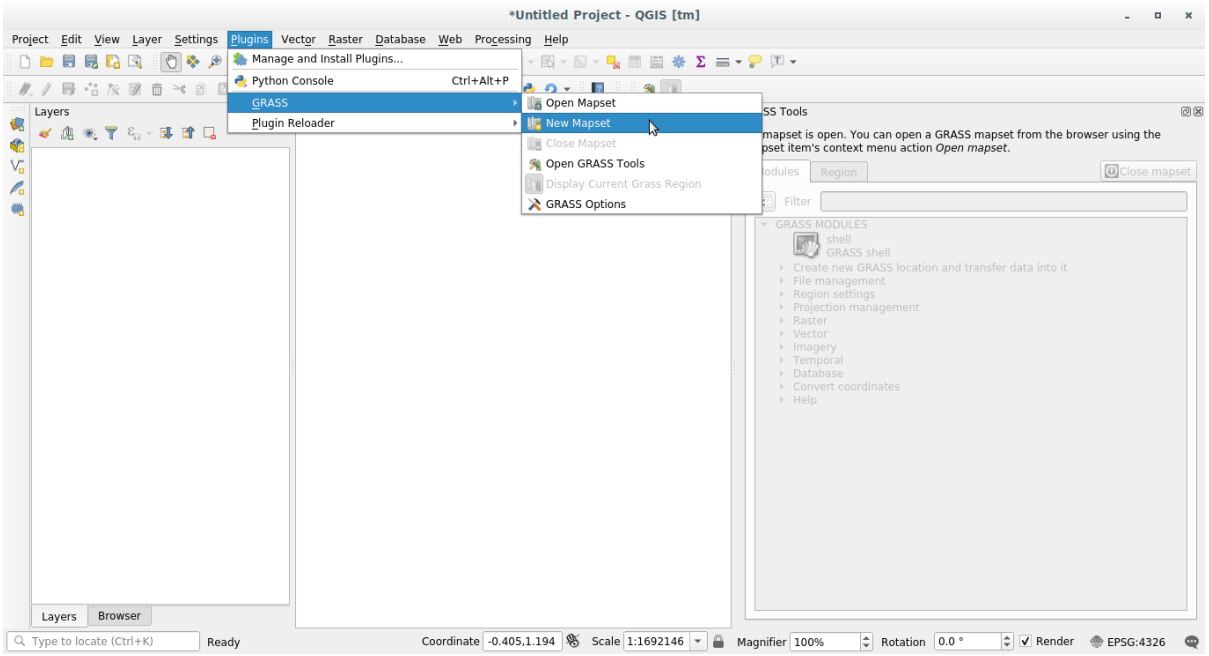


図 12.2: GRASS データベース構造（GRASS ドキュメントより）

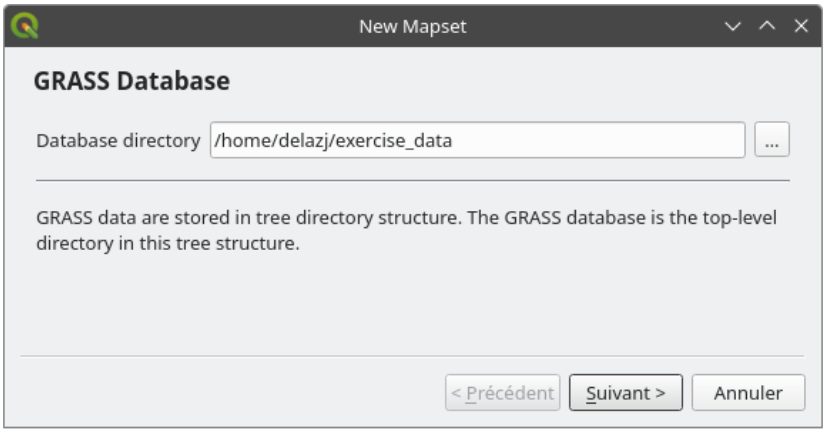
12.1.2 ★☆☆ （初級レベル） 理解しよう: 新しい GRASS プロジェクトを始める

1. **プラグイン** *GRASS* **新規** *Mapset* メニューをクリックします:



GRASS データベースの場所を選択するよう指示されます。

2. データベースを構成するためにそれを GRASS が使用するディレクトリとして設定します:



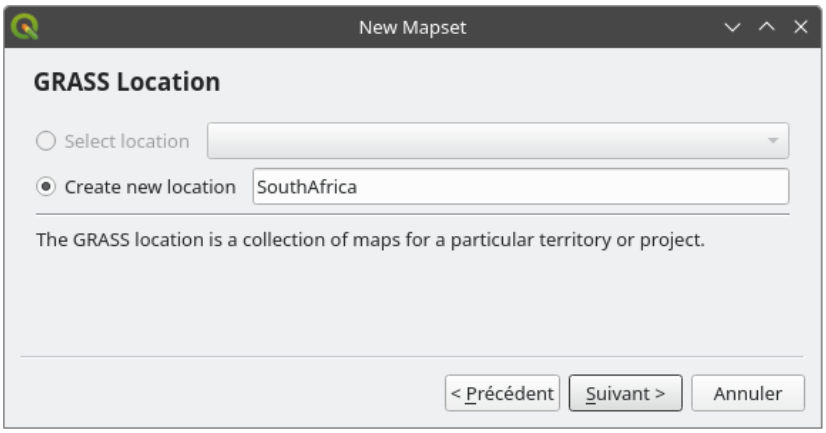
3. **次へ** をクリックします。

GRASS では ロケーション を作成する必要があります。 ロケーション は作業しようとしている地理的領域（Grass `Region` ともいわれる）の最大範囲を定義します。

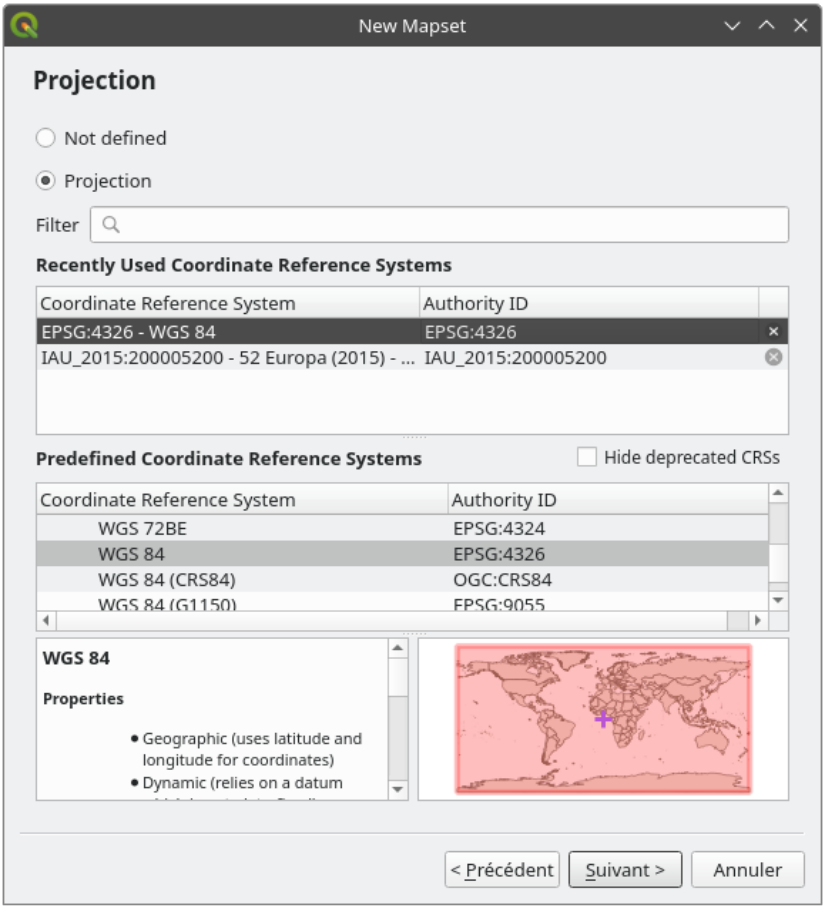
注釈

Region は、GRASS ですべてのレイヤが考慮されるエリアを記述するため、GRASS では非常に重要です。外にあるものはすべて考慮されません。でもご心配なく、ロケーションが作成された後、いつでも GRASS Region の範囲を変更できます

1. 新しいロケーションを `SouthAfrica` と呼びます:

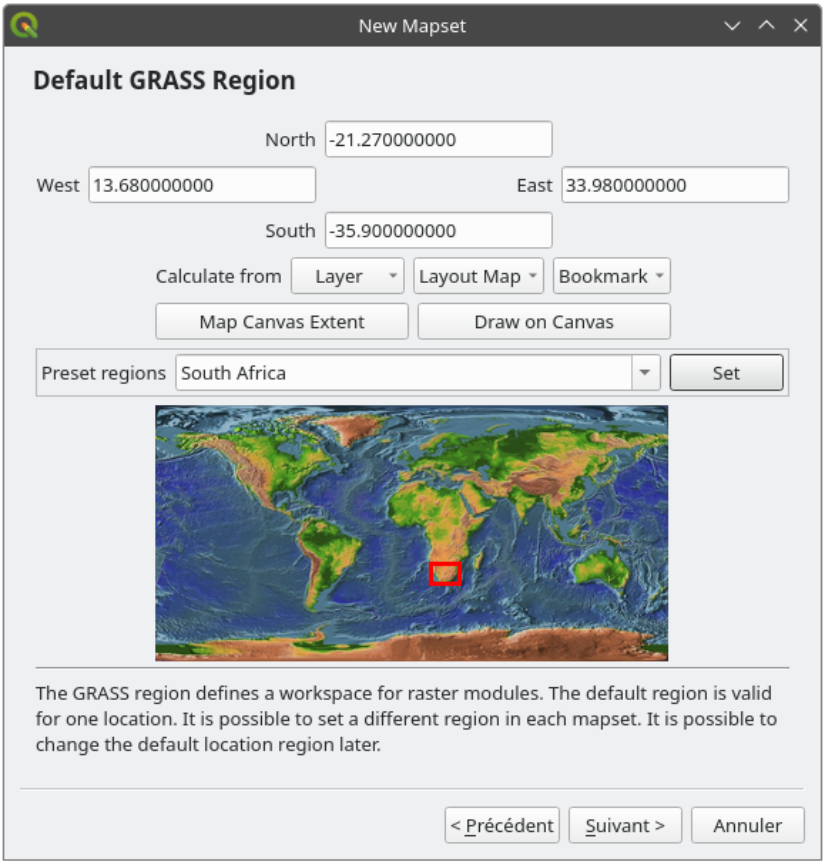


2. **次へ** をクリックします。
3. ここでは `WGS 84` を使うので、その CRS を検索して選択してください:



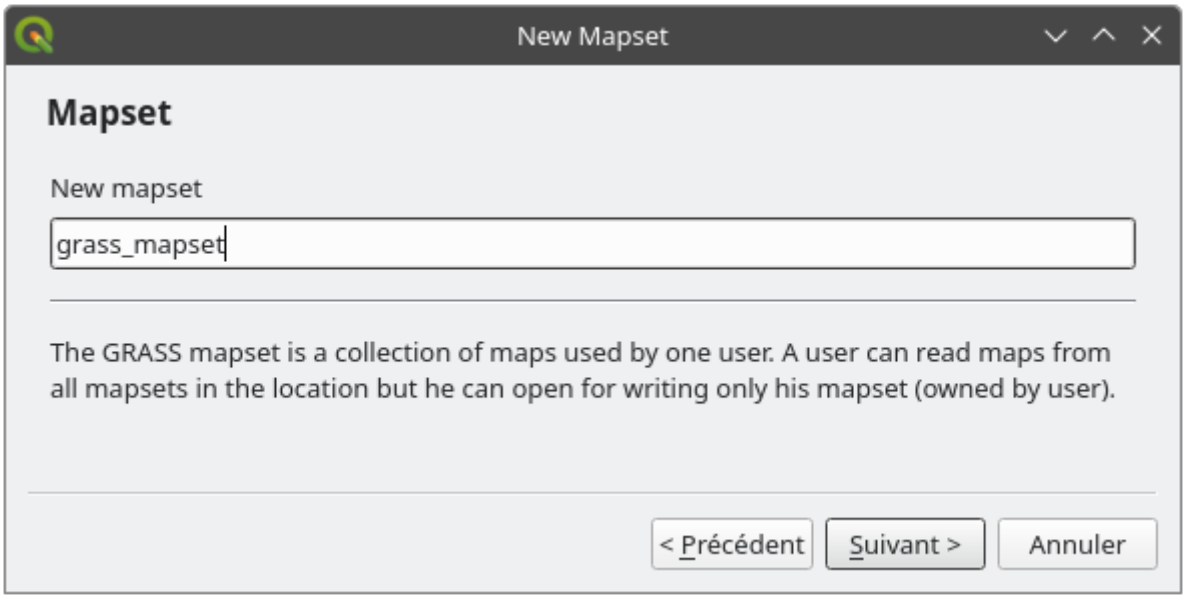
4. **次へ** をクリックします。

5. Now select the region `South Africa` from the dropdown and click *Set*:

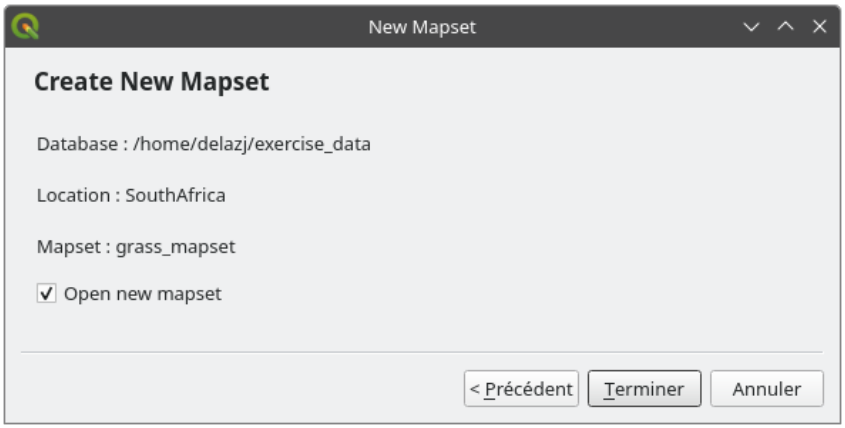


6. **次へ** をクリックします。

7. マップセットを作成します。それがあなたがこれから作業するマップファイルです。



入力が完了すると、入力した情報の要約を尋ねるダイアログが表示されます。



8. **完了** をクリックします。
9. 成功ダイアログで *OK* をクリックします。

GRASS パネルがアクティブになり、すべての GRASS ツールを使用できるようになるのがわかります。

12.1.3 ★☆☆ （初級レベル） 理解しよう: GRASS にベクタデータを読み込む

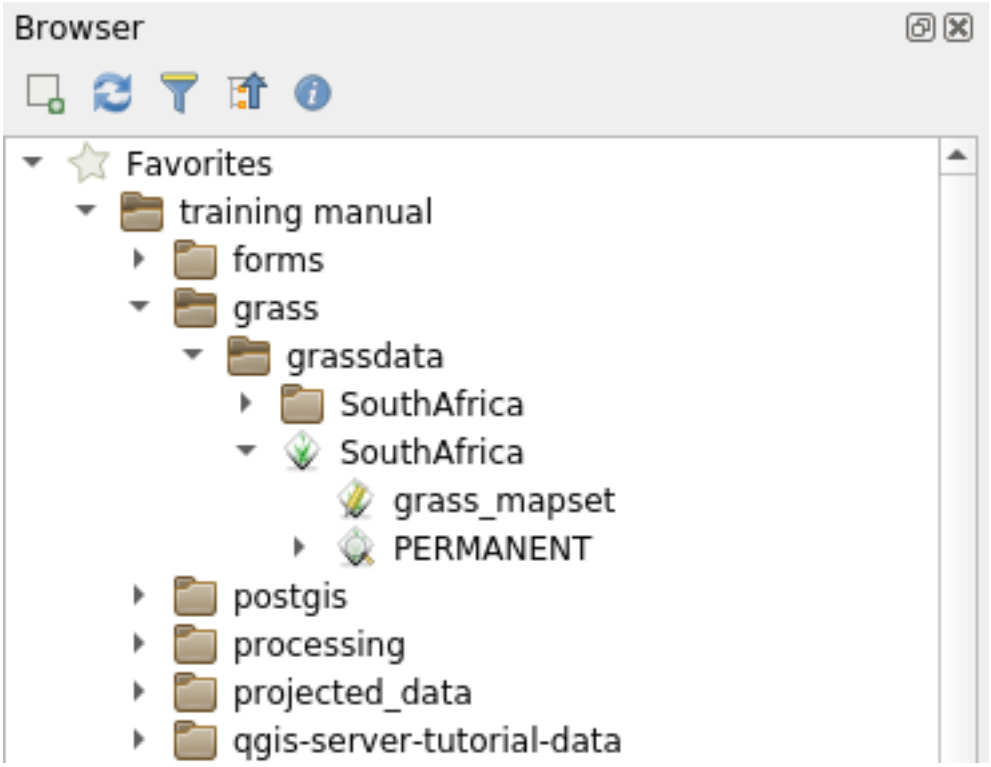
今はマップは空白であり、すべての GRASS ツールの使用を開始するには、データを GRASS データベース、特に マップセット に読み込む必要があります。 GRASS マップセット に読み込まれていないレイヤでは GRASS ツールを使用できません。

GRASS のデータベースにデータを読み込むには、様々な方法があります。まずは最初の 1 つから始めましょう。

★☆☆ （初級レベル） 理解しよう: QGIS ブラウザを使ってデータを読み込む

セクション ★☆☆ （初級レベル） ブラウザパネル では、QGIS にデータを読み込む最も簡単で早い方法はブラウザパネルであることがわかりました。

QGIS ブラウザは GRASS データを、**実際の** GRASS データとして認識します。それは、GRASS マップセットの横に GRASS アイコンが表示されることで確認できます。また、開いたマップセットの横には  のアイコンが表示されます。



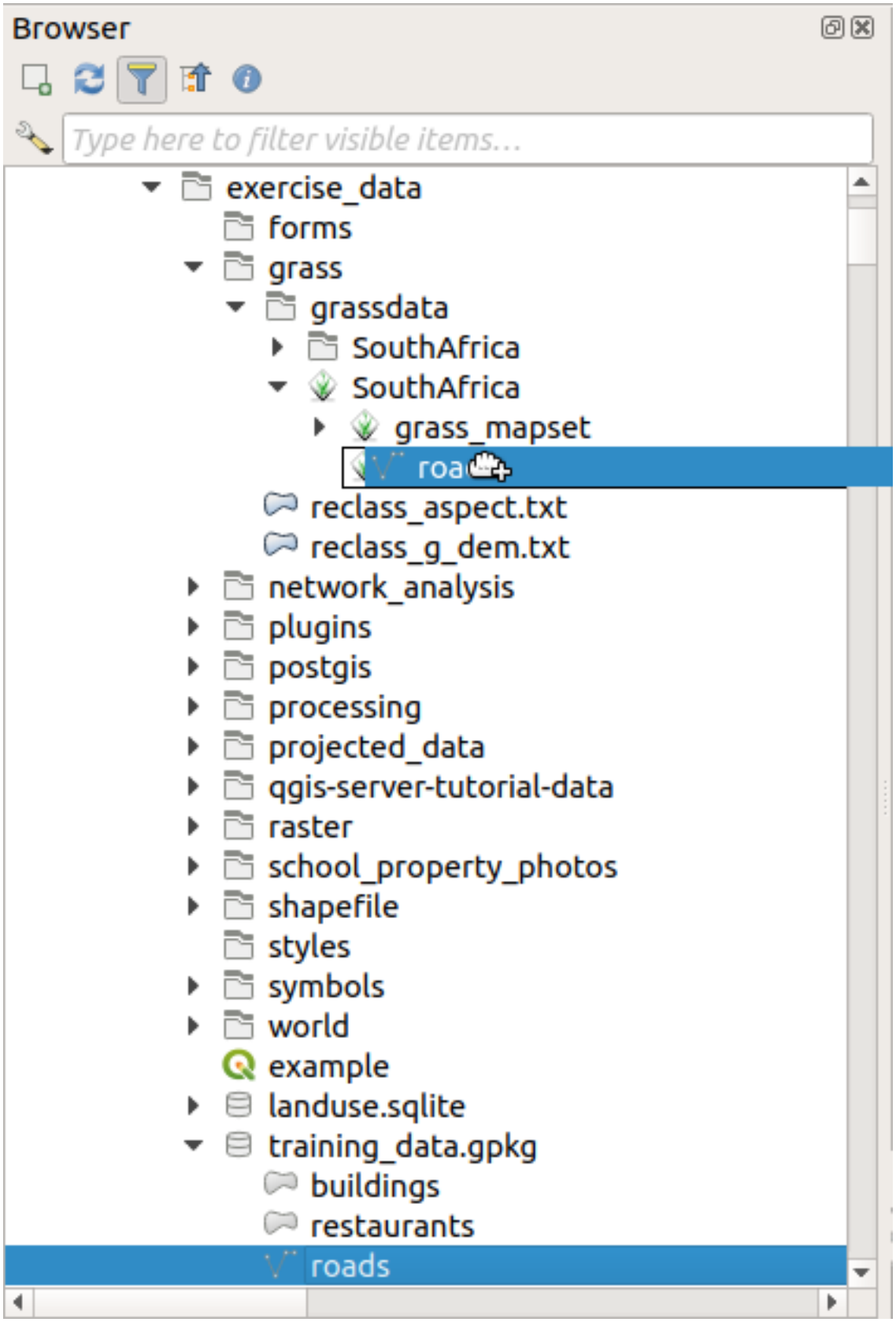
注釈

GRASS ロケーションの複製が通常のフォルダとして表示されます: GRASS マップセット データは、 フォルダ内にあるものです。

フォルダ内のレイヤは **ドラッグ＆ドロップ** で簡単に GRASS マップセットに取り込むことができます。

それでは、 `roads` レイヤを `SouthAfrica` ロケーションの `grass_mapset` マップセットにインポートしてみましょう。

ブラウザを開き、 `training_data.gpkg` GeoPackage ファイルから `roads` レイヤを `grass_mapset` Mapset に単純にドラッグしてください。



それでおしまい！マップセットを展開すると、インポートされた roads レイヤが表示されます。これで、インポートされたレイヤを他のすべてのレイヤと同様に QGIS に読み込むことができます。

Tip

レイヤ凡例パネルからブラウザパネルのマップセットにレイヤを読み込むこともできます。これにより、ワークフローが驚くほど高速になります！

★☆☆ （初級レベル） 理解しよう: GRASS パネルを使ってデータを読み込む

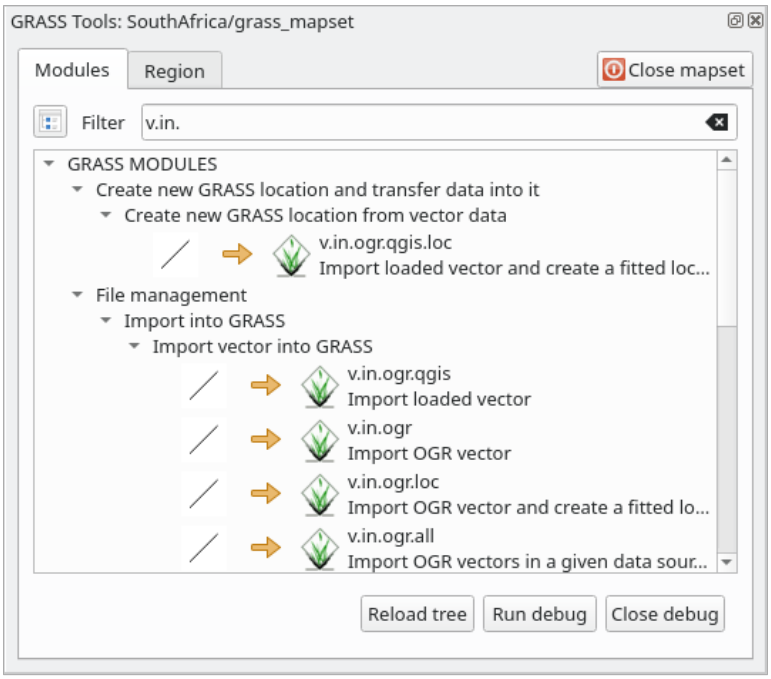
ここでは、 *long* メソッドを使用して、 `rivers.shp` レイヤーを同じマップセットにロードします。

1. 通常どおりデータを QGIS に読み込みます。 `rivers.shp` データセット（`exercise_data/shapefile/` フォルダにあります）を使用します

2. 読み込まれたらすぐに、`GRASS Panel` の **フィルタ** ボックスをクリックし、`v.in.ogr.qgis` という語を入力してベクタインポートツールを見つけます（**ファイル管理** *GRASS* **へのインポート** *GRASS* **へのベクタのインポート** で利用可能）。

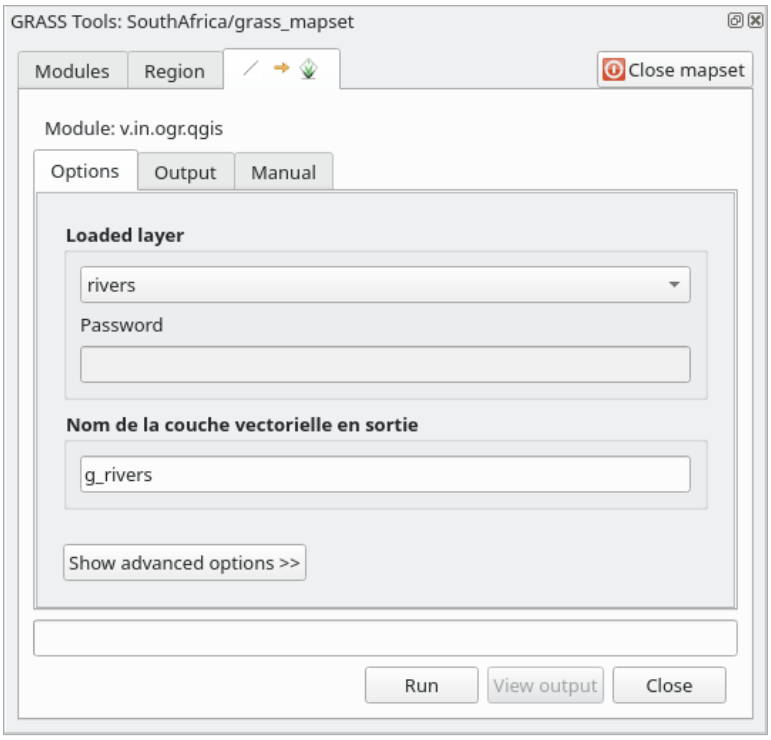
   警告

   似たようなツールが 2 つあります： `v.in.ogr.qgis` と `v.in.ogr.qgis.loc` 。探しているのは **１つめ** です。



   v は **ベクタ** 、 in は GRASS データベースにデータをインポートする関数を意味します。 ogr はベクタデータの読み込みに使用されるソフトウェア・ライブラリです。そして qgis は、そのツールが QGIS に既に読み込まれたベクタの中からベクタを探すことを意味します。

3. Once you've found this tool, click on it to bring up the tool itself. Choose the `rivers` layer in the *Loaded Layer* box and type and name it `g_rivers` to prevent confusion:



注釈

★★★ （上級レベル） **アドバンスドオプション** には別のインポートオプションが提供されています。これにはデータのインポートに使用する SQL クエリの WHERE 句を追加する機能が含まれます。

4. **実行** をクリックしてインポートを開始します。
5. それが終わったら **出力を見る** をクリックして新しくインポートされた GRASS レイヤを地図に表示します。
6. まずインポートツールを閉じて (**出力を見る** のすぐ右にある **閉じる** ボタンをクリックします)、そして *GRASS* **ツール** ウィンドウを閉じます。
7. Remove the original `rivers` layer.

今、QGIS の地図に表示されているのはインポートされた GRASS レイヤだけです。

12.1.4 ★☆☆ （初級レベル） 理解しよう: GRASS にラスタデータを読み込む

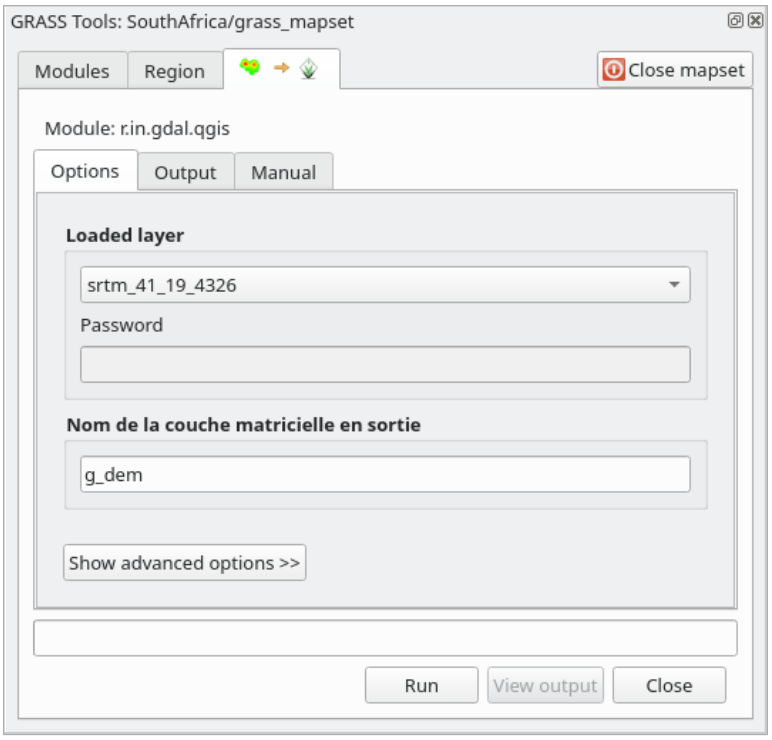
ベクタレイヤをインポートしたのと同じ方法でラスタレイヤをインポートできます。

GRASS マップセットにレイヤ `srtm_41_19_4326.tif` をインポートします。

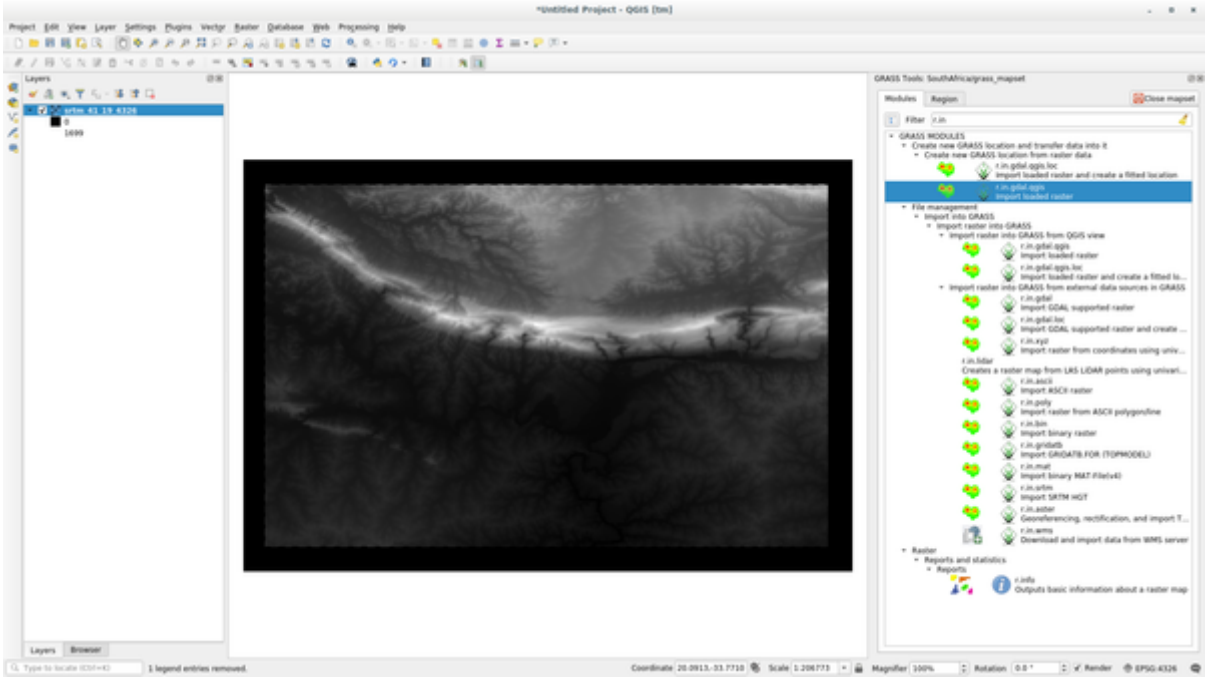
注釈

ラスタレイヤは既に正しい CRS、「WGS 84」にあります。レイヤが異なる CRS にある場合は、GRASS マップセットの同じ CRS に再投影する必要があります

1. QGIS で `srtm_41_19_4326.tif` レイヤを読み込みます
2. *GRASS* **ツール** ダイアログを再度開きます。
3. **モジュール** タブをクリックします。
4. `r.in.gdal.qgis` を検索し（**ファイル管理** *GRASS* **へのインポート** *GRASS* **へのラスターインポート** *QGIS* **ビューからの** *GRASS* **へのラスターインポート** から利用可能）、ツールをダブルクリックしてツールのダイアログを開きます。
5. 入力レイヤが `srtm_41_19_4326.tif` 、出力が `g_dem` となるように設定します。



6. **実行** をクリックします。
7. 処理が完了したら **出力を見る** をクリックします。
8. 現在のタブを閉じ、そしてダイアログボックスを閉じます。



9. これで元の srtm_41_19_4326.tif レイヤは削除しても問題ありません。

12.1.5 ★☆☆ （初級レベル） 自分でやってみよう: マップセットにレイヤを追加する

GRASS マップセットに、exercise_data/shapefile/ フォルダからベクタレイヤ water.shp と places.shp をインポートしてみてください。 rivers の場合と同様に、混乱を避けるために、インポートしたレイヤの名前を g_water および g_places に変更します。

答え

レイヤ（ベクタとラスタの両方）を GRASS マップセットに追加するには、ブラウザにドラッグアンドドロップするか（**★☆☆ （初級レベル） 理解しよう**: *QGIS* **ブラウザを使ってデータを読み込む** 参照）、ベクタに v.in.gdal.qgis をラスタレイヤに r.in.gdal.qgis を使用します。

12.1.6 ★☆☆ （初級レベル） 既存の GRASS マップセットを開く

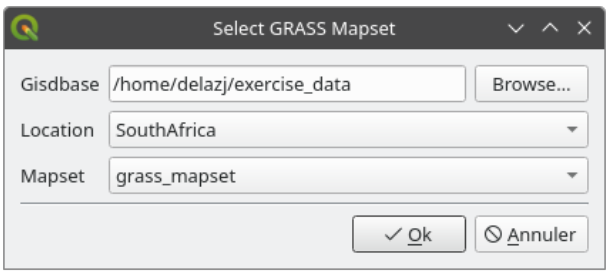
既に GRASS マップセットがある場合、別の QGIS セッションでそれを簡単に開くことができます。

GRASS マップセットを開く方法には複数の方法があります。そのいくつかを試してみましょう。

GRASS **ツール** ウィンドウの *Mapset* **を閉じる** ボタンをクリックして、マップセットを閉じましょう。

★☆☆ （初級レベル） 理解しよう: GRASS プラグインを使う

1. **プラグイン** ▸ *GRASS* ▸ *Mapset* **を開く** メニューをクリックします。
2. GRASS データベースフォルダを参照します。注意してください！ GRASS マップセットのフォルダではなく、親フォルダを選択する必要があります。実際、GRASS はデータベースのすべての「ロケーション」と各「ロケーション」のすべての「マップセット」を読み取ります：



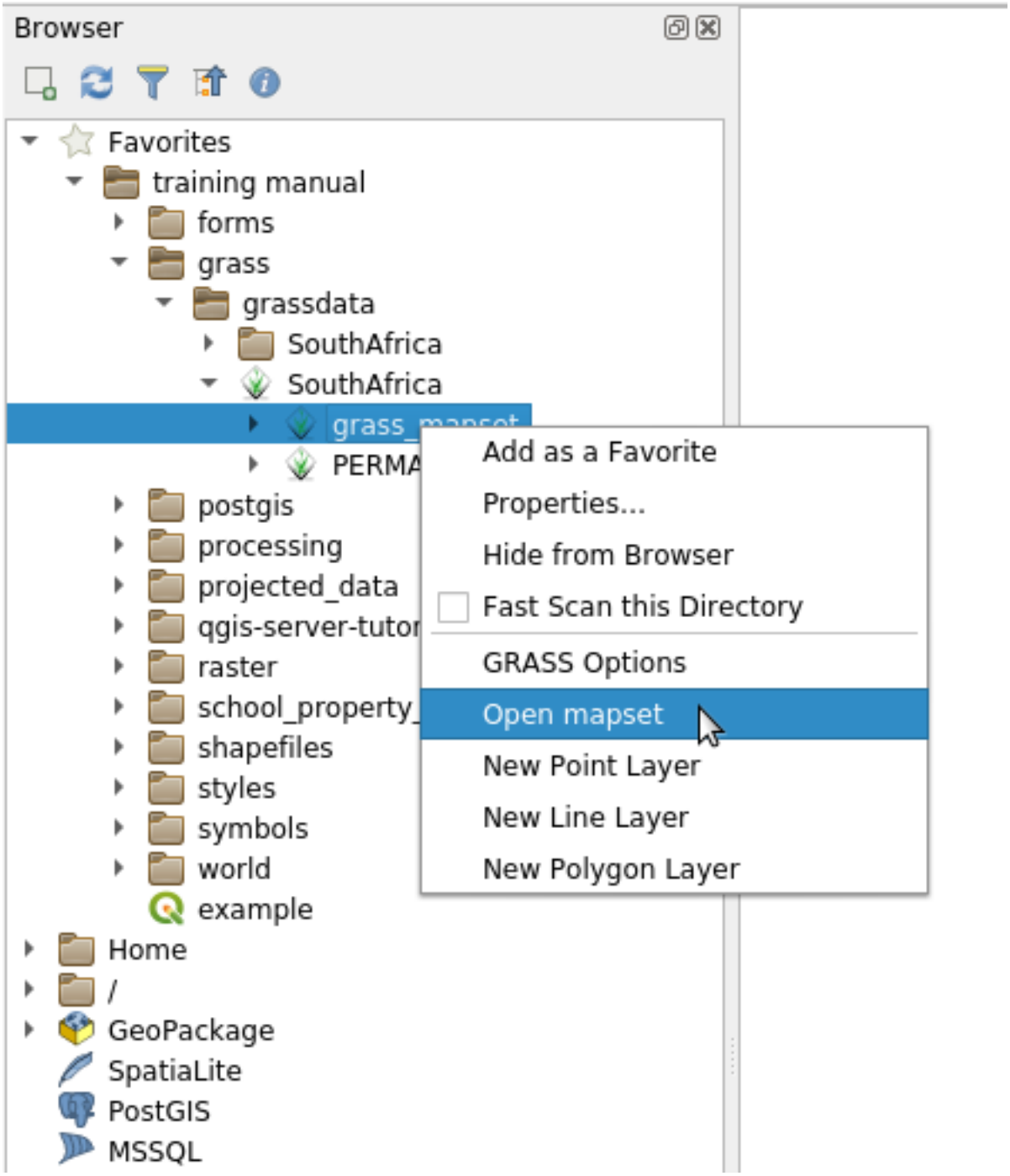
3. Choose the `Location` `SouthAfrica` and the `Mapset` `grass_mapset` that we have created before.

これで完了です。GRASS パネルがアクティブになり、マップセットが正しく開かれたことを意味します。

★☆☆ （初級レベル） 理解しよう: QGIS ブラウザを使う

QGIS ブラウザを使って マップセット を開くと、さらに速く、簡単になります:

1. *GRASS Tools* ウィンドウの *Mapset* **を閉じる** ボタンをクリックして、マップセットを閉じます（開いている場合）。
2. QGIS ブラウザで GRASS データベースのフォルダをブラウズします。
3. マップセットを右クリックします（マップセットは  GRASS アイコンが隣にあることを思い出してください）。いくつかのオプションがあるのがわかります。
4. *Mapset* **を開く** をクリックします:



これでマップセットが開き、使う準備ができました!

Tip

GRASS マップセット上で右クリックすると、様々な設定ができます。いろいろな設定を試して、便利なオプションを見てみましょう。

12.1.7 結論

GRASS はデータを空間データベース構造に読み込むため、GRASS のデータ収集ワークフローは QGIS の方法とは多少異なります。しかし、フロントエンドとして QGIS を使用することによって GRASS のデータソースとして QGIS 内の既存レイヤを使用でき、GRASS マップセットのセットアップを簡単にすることができます。

12.1.8 次は?

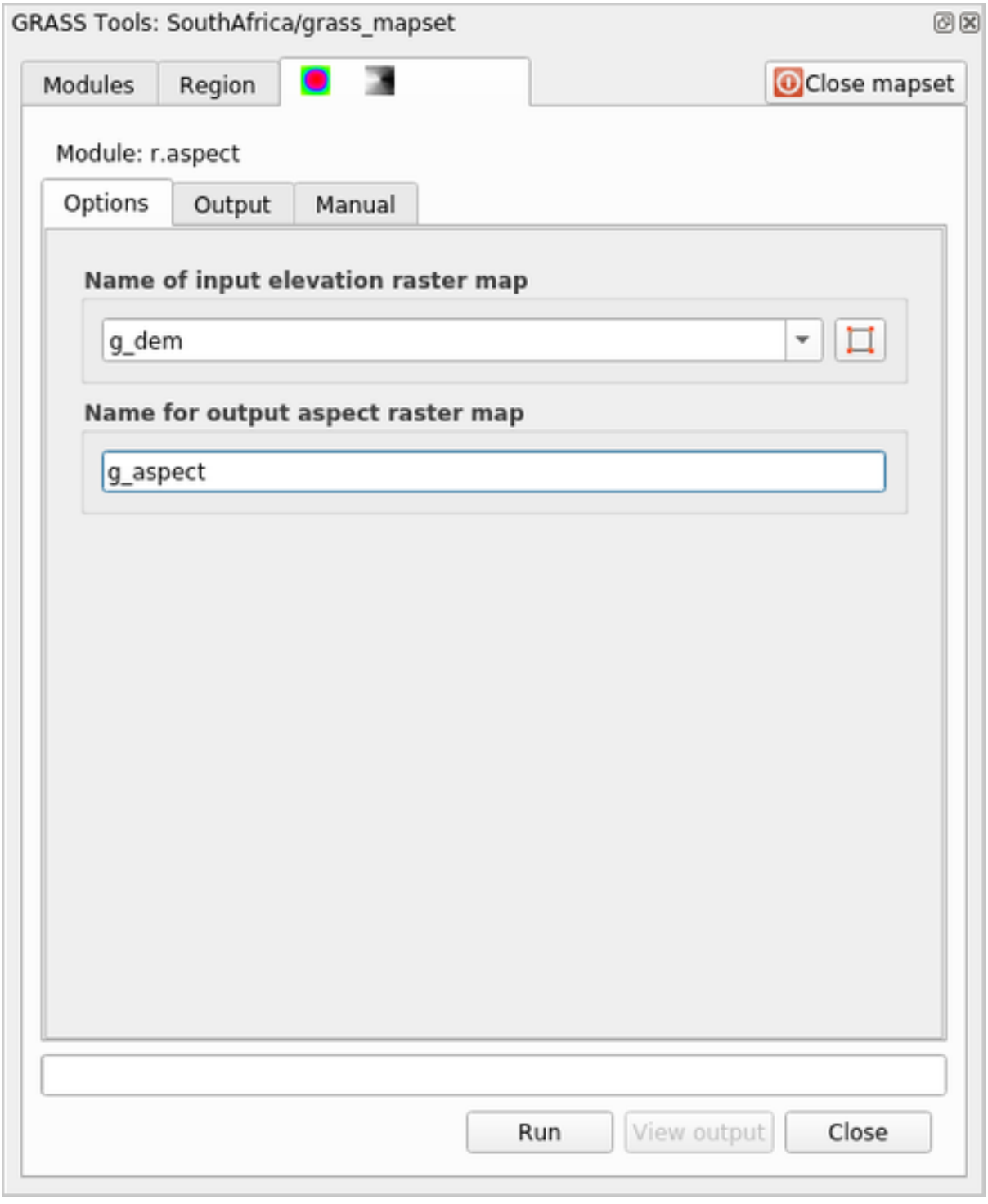
データは GRASS にインポートされました。GRASS の高度な分析操作を見ることができます。

12.2 レッスン: GRASS ツール

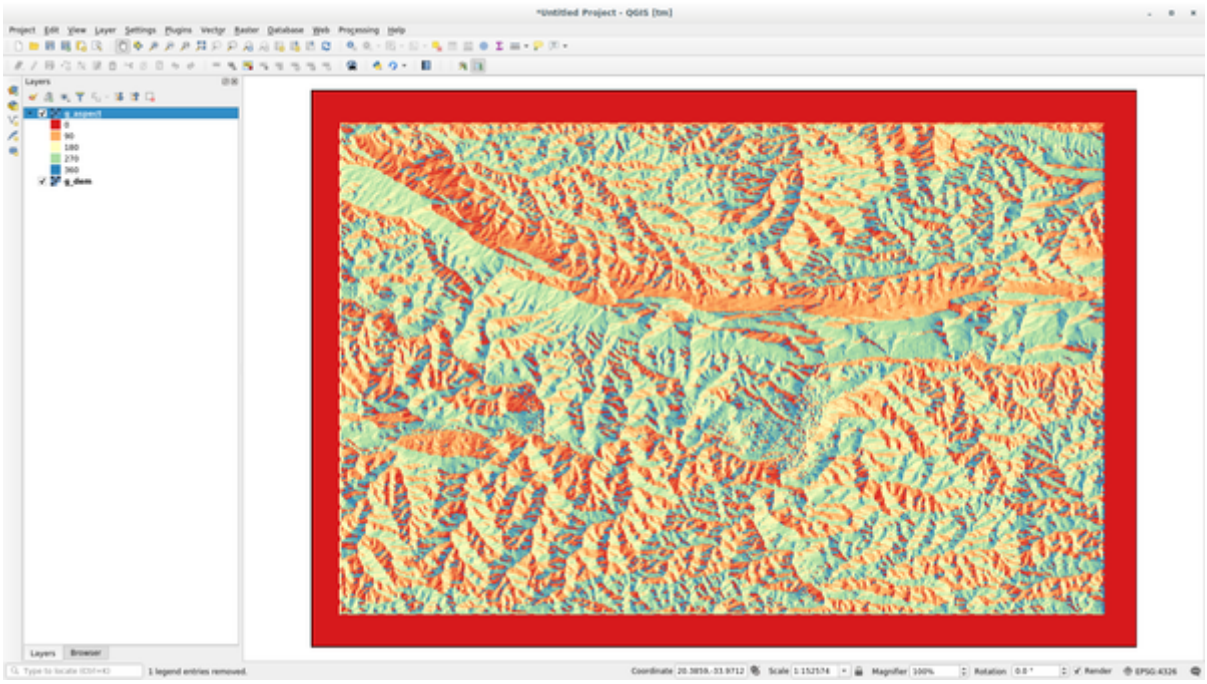
このレッスンでは、あなたに GRASS の機能についてのアイデアを与えるために選り抜きのツールを紹介します。

12.2.1 ★☆☆ （初級レベル） 理解しよう: 傾斜方位地図を作る

1. *GRASS* **ツール** タブを開きます
2. Load the `g_dem` raster layer from the *grass_mapset* Mapset
3. *r.aspect* モジュールを **モジュールリスト** タブの **フィルタ** フィールドを使って探します
4. そのツールを開いて次のように設定し、 **実行** ボタンをクリックします:



5. 処理が終了したら **出力を見る** をクリックし、結果のレイヤをキャンバスに読み込みます:

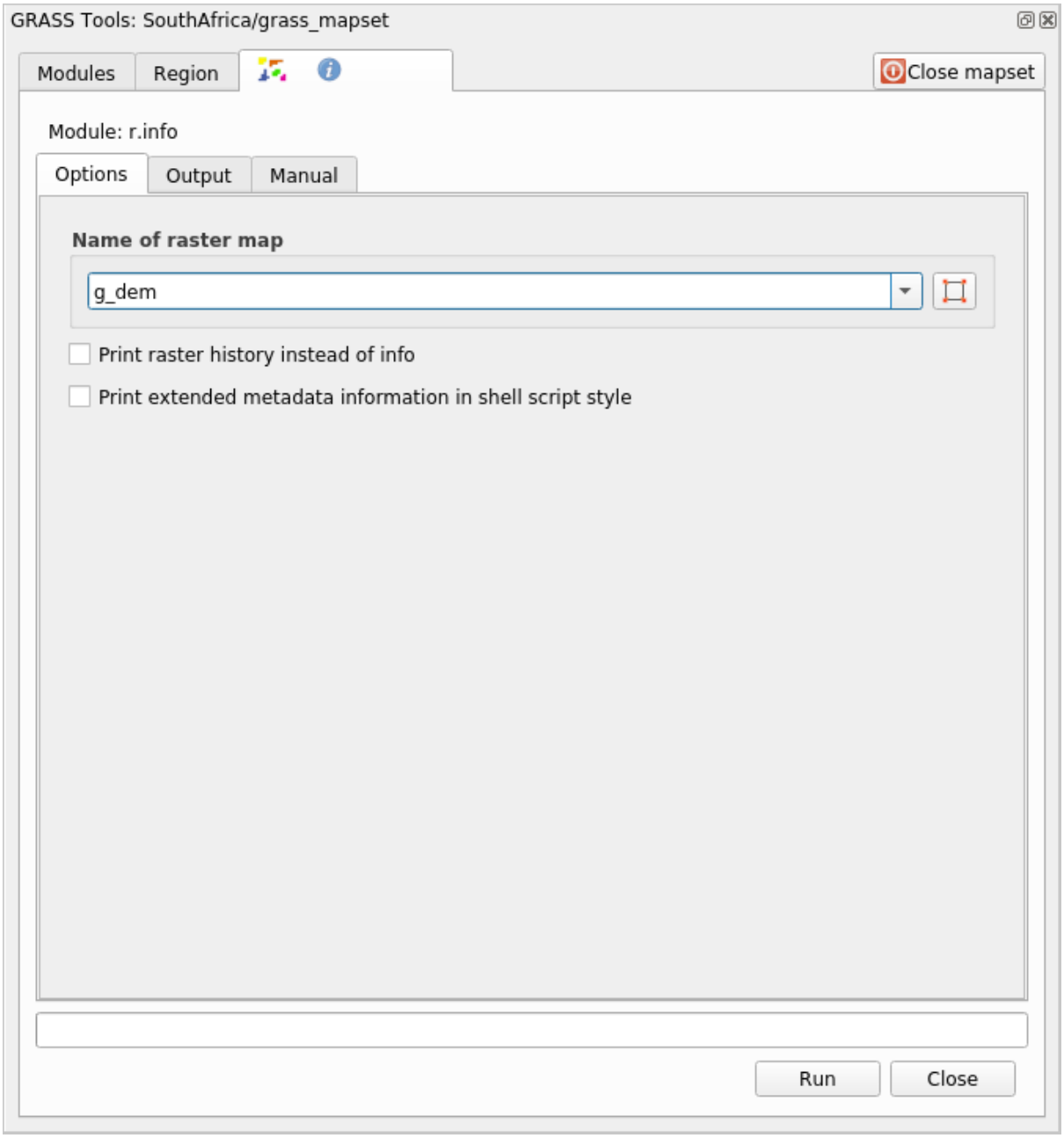


g_aspect レイヤは *grass_mapset* マップセット内に格納されているので、キャンバスからレイヤを削除してもいつでも再読み込みすることができます。

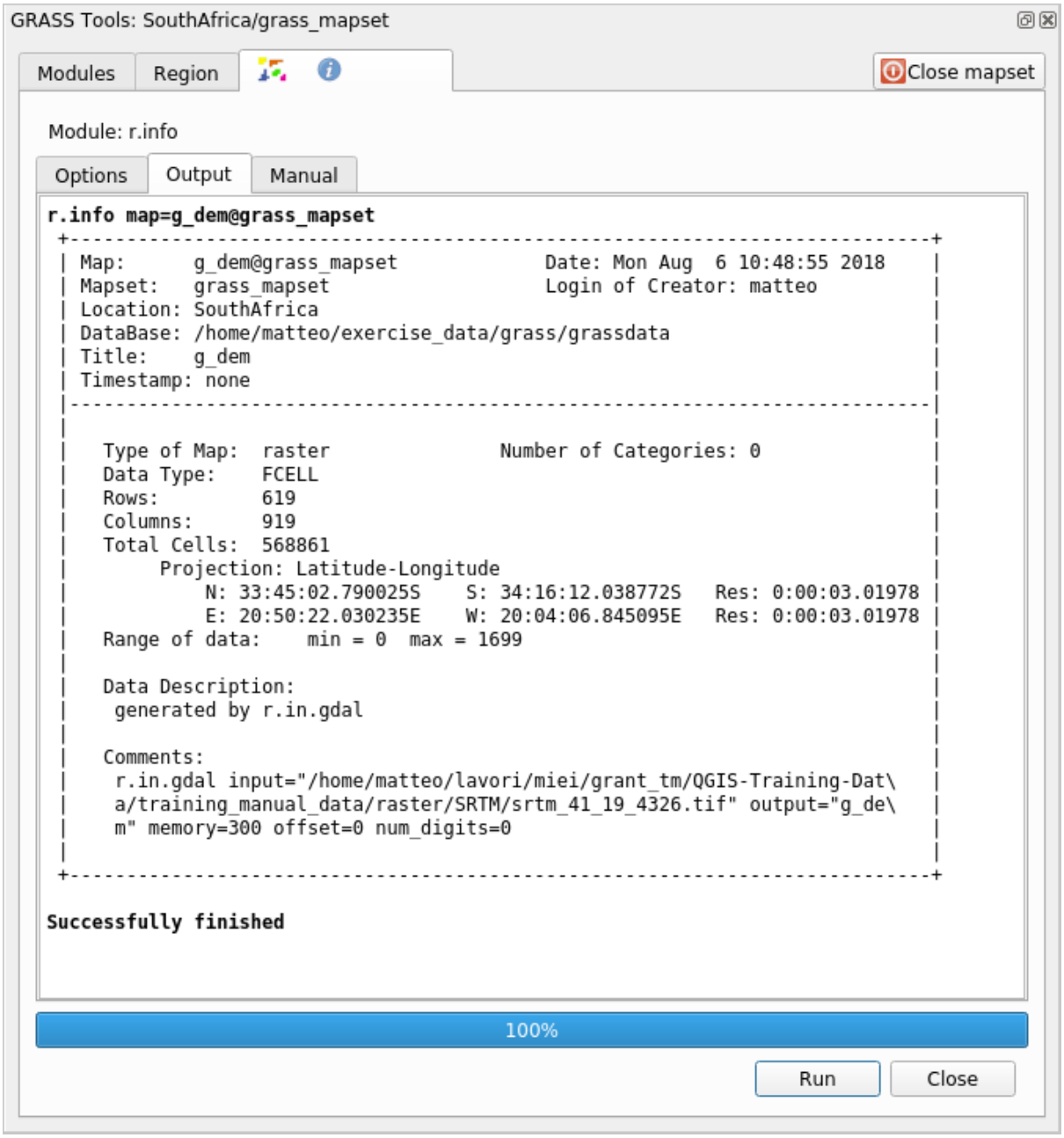
12.2.2 ★☆☆ （初級レベル） 理解しよう: ラスタレイヤの基本的な統計情報を取得する

We want to know some basic statistics of the g_dem raster layer.

1. *GRASS* **ツール** タブを開きます
2. Load the g_dem raster layer from the *grass_mapset* Mapset
3. **モジュールリスト** タブの **フィルタ** フィールドで *r.info* モジュールを検索してください
4. 次のようにツールを設定し、**実行** をクリックします:



5. 出力タブ内には、ファイルのパス、行数、列数など有用なラスタ情報が出力されているのがわかります。



12.2.3 ★★☆ （中級レベル） 理解しよう: Reclass ツール

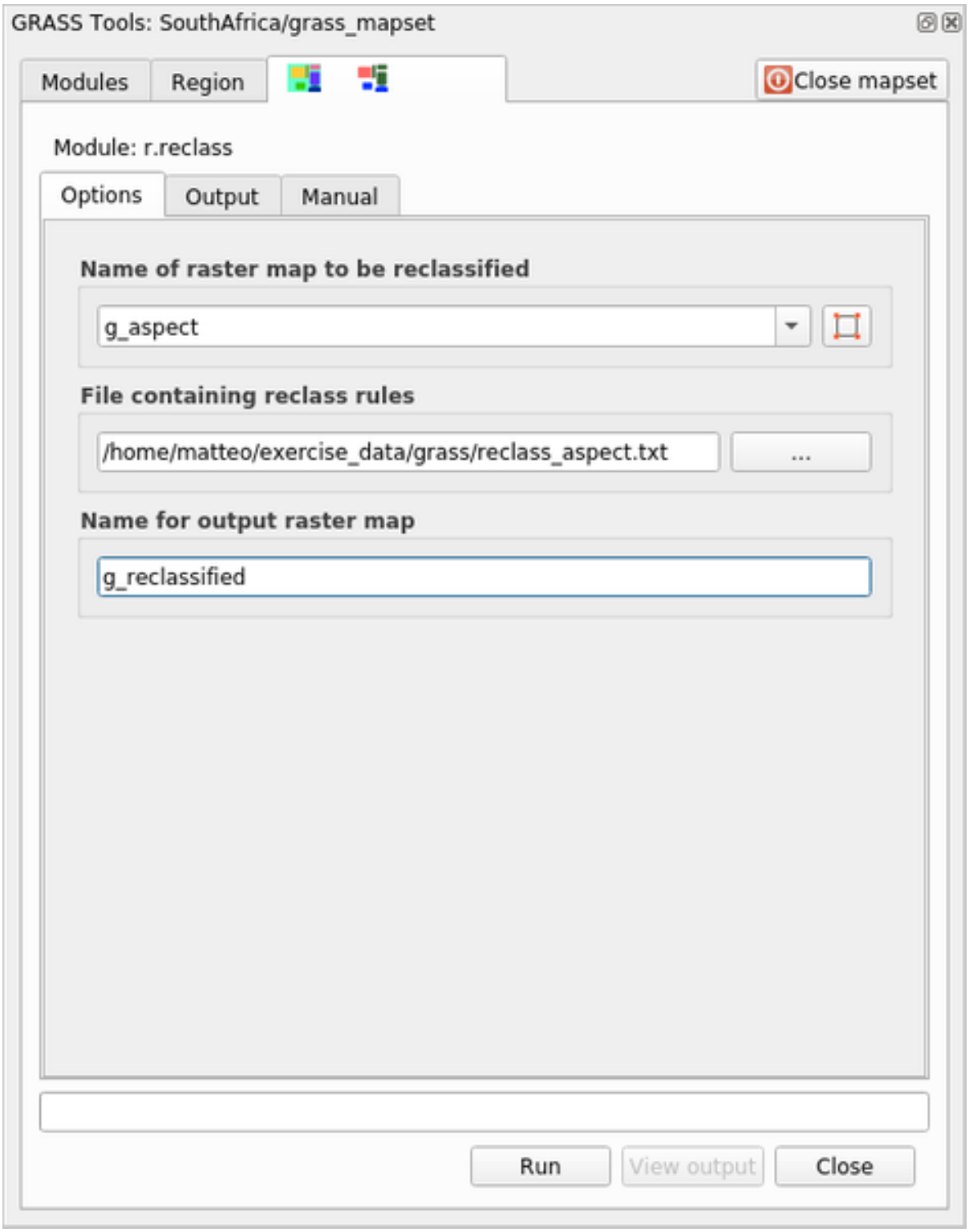
ラスタレイヤの再分類は非常に価値ある作業です。私たちは `g_dem` レイヤから `g_aspect` レイヤを作成したところです。その値域は 0（北）から 90（東）、180（南）、270（西）を経て、最後に 360（再び北）までとなっています。 `g_aspect` レイヤを再分類して、具体的な **規則*（北 = *1*, 東 = *2*, 南 = *3*, 西 = *4*）に従って、*4* つの **カテゴリー** だけを持つようにすることができます。

Grass 再分類ツールは、定義された規則を含む `txt` ファイルを受け取ります。規則の書き方は非常に簡単で、GRASS マニュアルに非常に詳しい説明があります。

Tip

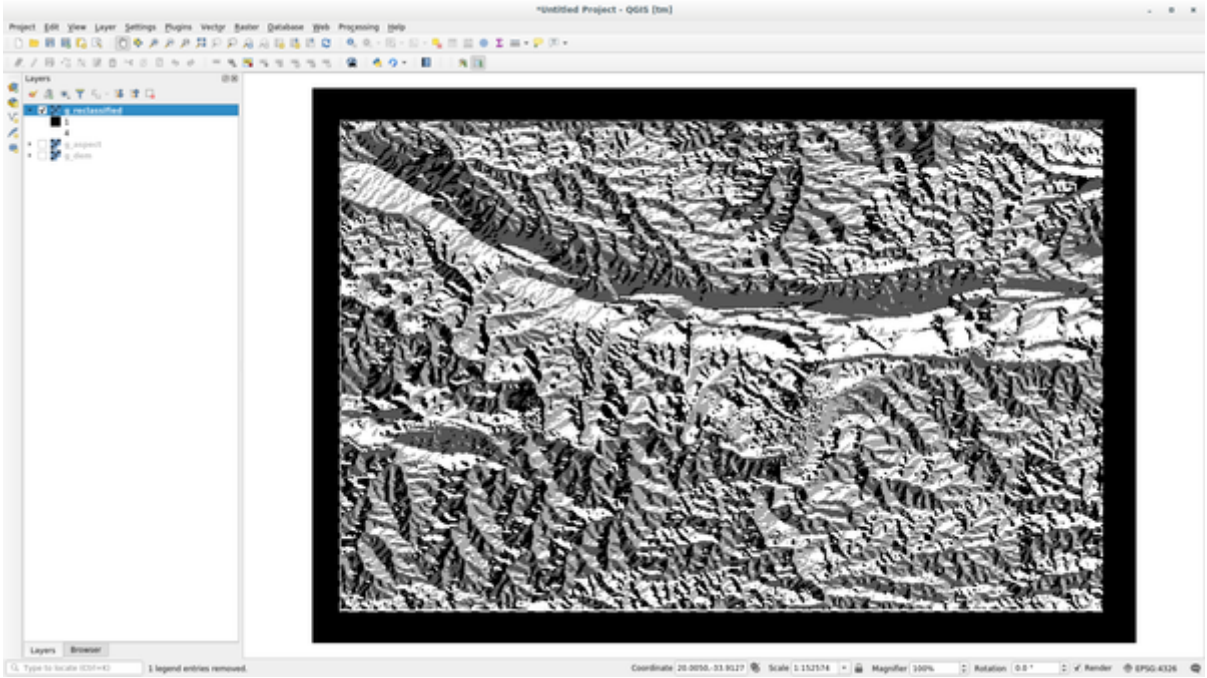
GRASS の各ツールには、それぞれ「マニュアル」タブがあります。使用するツールの説明をよく読んで、便利なパラメータを見逃さないようにしましょう。

1. `g_aspect` レイヤを読み込むか、作成していない場合は、**★☆☆（初級レベル）理解しよう: 傾斜方位地図を作る** セクションに戻りましょう。
2. *r.reclass* モジュールを **モジュール** タブの **フィルタ** フィールドで検索して探します
3. ツールを開き、次の図のように設定します。規則を含んでいるファイルは `exercise_data/grass/` フォルダに `reclass_aspect.txt` という名前であります。
4. **実行** をクリックし、処理が終了するまで待ちます。



5. **出力を見る** をクリックすると、再分類されたラスタがキャンバスに読み込まれます

   新しいレイヤは、4 つの値（1、2、3、4）だけで構成され、管理も加工もしやすくなっています。



Tip

テキストエディタで `reclass_aspect.txt` を開き、規則を確認し、慣れることから始めてください。さらに、GRASS のマニュアルをよく読んでみてください：多くの異なる例が示されています。

12.2.4 ★★☆ （中級レベル） 自分でやってみよう: 自分の規則で再分類する

Try to reclassify the `g_dem` layer into 3 new categories:

- 0 から 1000、新しい値 = 1
- 1000 から 1400、新しい値 = 2
- 1400 から最大のラスタ値、新しい値 = 3

答え

ラスタの最大値を調べため `r.info` ツールを実行します：そのコンソールで最大値が 1699 であることが分かります。これで規則を書く準備ができました。

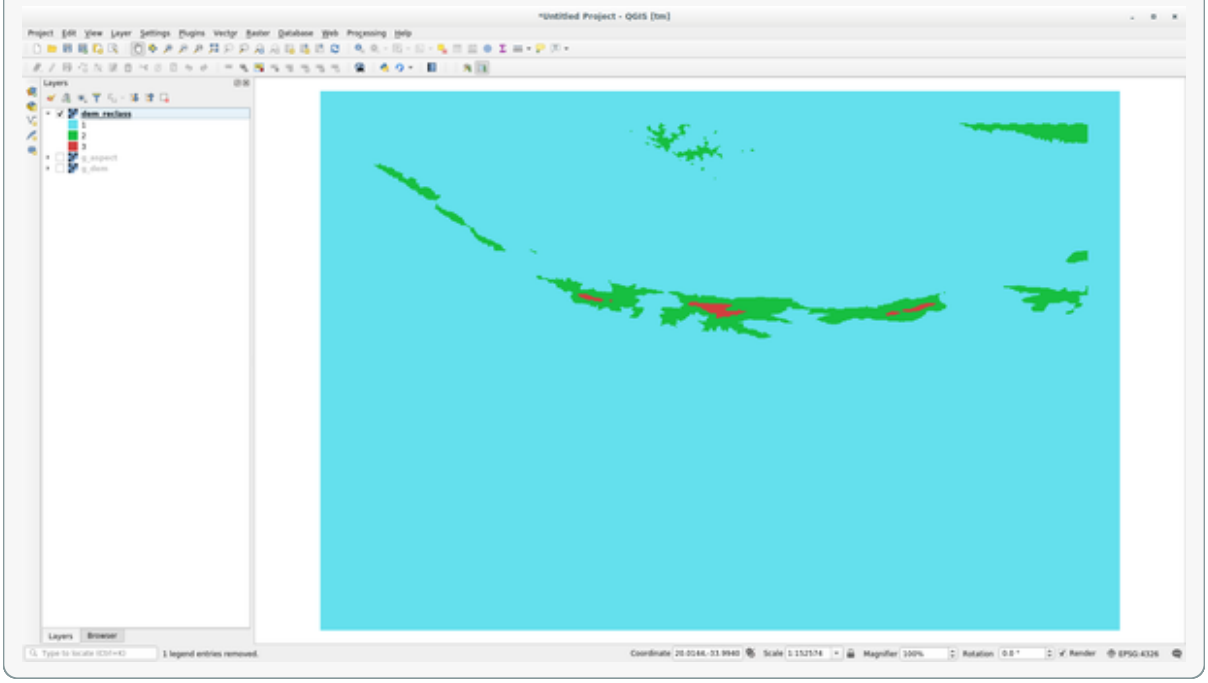
1. テキストエディタを開き、次の規則を加えます:

```
0 thru 1000 = 1
1000 thru 1400 = 2
1400 thru 1699 = 3
```

2. そのファイルを `my_rules.txt` ファイルとして保存し、テキストエディタを閉じます。

3. `r.reclass` ツールを実行します: `g_dem` レイヤを選び、先に保存した規則が入ったファイルを読み込みます。

4. **実行**、次に **出力を見る** をクリックします。色を変えると最終的な結果は次図のようになります:



12.2.5 ★★☆ （中級レベル） 理解しよう: Mapcalc ツール

Mapcalc ツールは QGIS の ラスタ計算機に似ています。1 つまたは複数のラスタレイヤに対して数学的操作を実行することができ、最終結果は計算された値を持つ新しいレイヤとなります。

The aim of the next exercise is to extract the values greater than 1000 from the `g_dem` raster layer.

1. **モジュール** タブの **フィルタ** フィールドで *r.mapcalc* モジュールを探してください。

2. ツールを起動します。

   Mapcalc ダイアログでは、ラスタまたはラスタの集合に対して実行する一連の解析を構築することができます。次のためにこれらのツールを使用します：

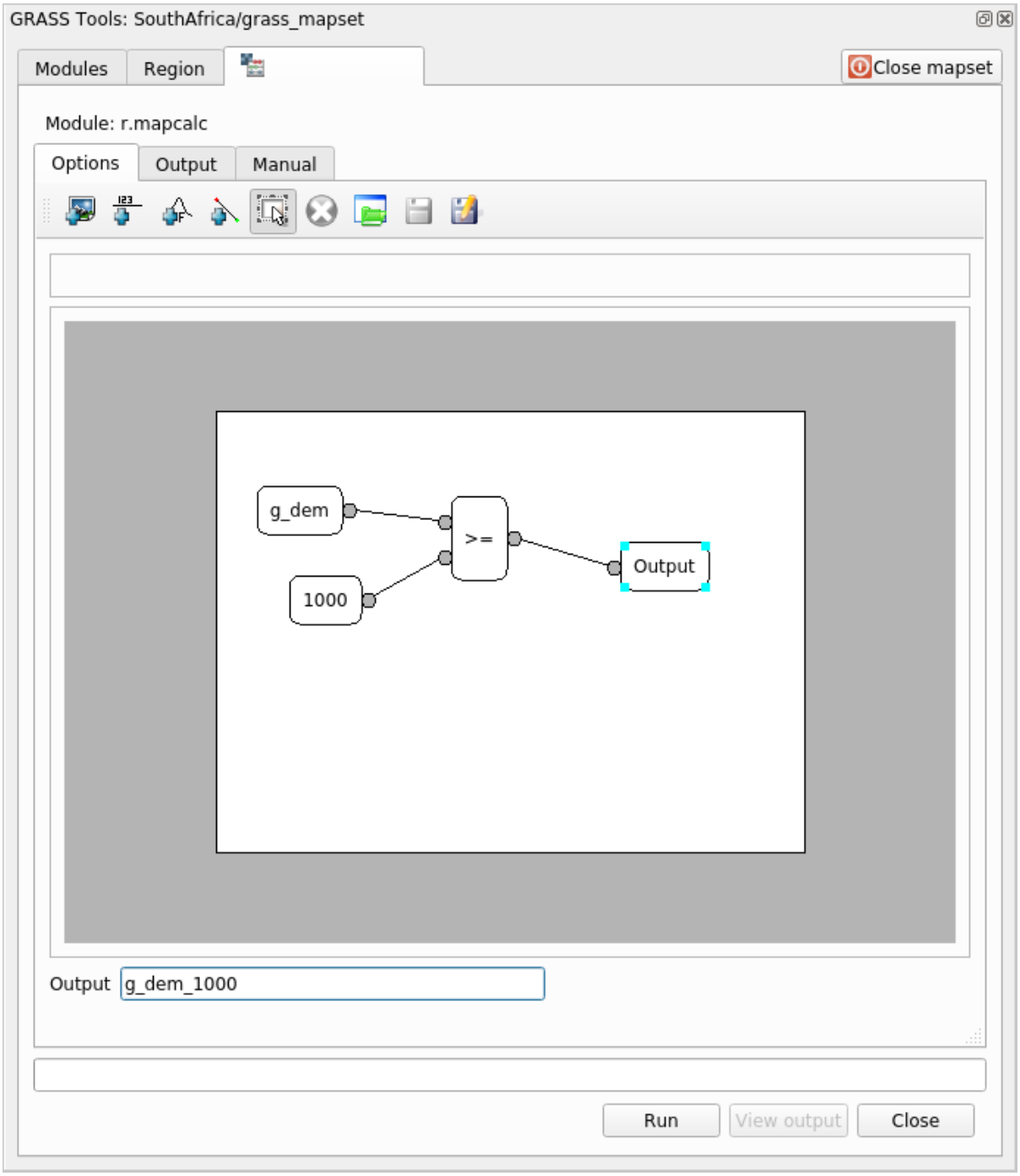


   順番に:

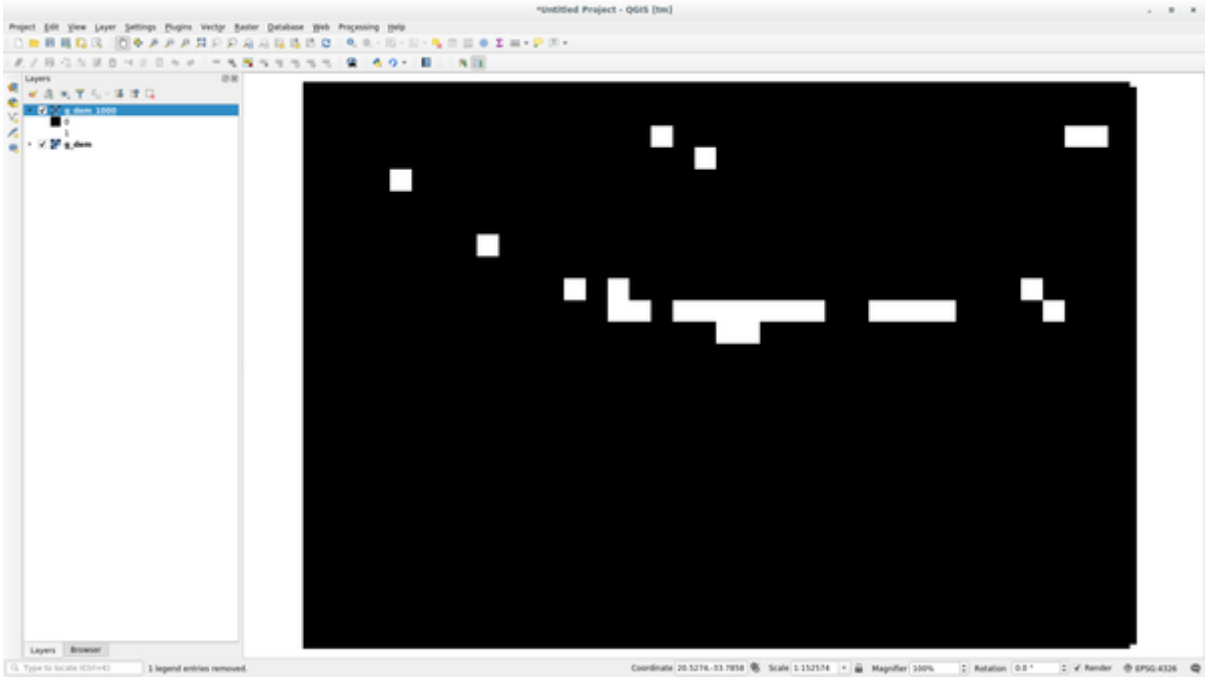
   - **地図を追加**：現在の GRASS マップセットからラスタファイルを追加する。
   - **定数値を加算**：関数で使う定数値、この例では 1000、を追加する
   - **演算子または関数を追加**：入力と出力に繋がる演算子または関数、この例では演算子 `greater equals than`、を追加する

- **接続を追加**： 要素を接続する。このツールを使って、あるアイテムの赤い点から別のアイテムの赤い点までクリック＆ドラッグします。接続線に正しく接続されているドットは灰色に変わります。線や点が赤い場合は、正しく接続されていません！
- **アイテムを選択**： アイテムを選択し、選択したアイテムを移動します。
- **選択したアイテムを削除**：選択されたアイテムを現在の mapcalc シートから削除しますが、マップセットからは削除しません（既存のラスタの場合）
- **開く**：決められた操作で既存のファイルを開きます
- **保存**：すべての操作をファイルに保存します
- **名前をつけて保存**： すべての操作を新しいファイルとしてディスクに保存します。

3. これらのツールを使って、次のアルゴリズムを組み立てます：



4. **実行** 次に **出力を見る** をクリックすると、出力がマップに表示されます：



これは地形が 1000 メートルより高い区域をすべて示しています。

Tip

GRASS Mapcalc ツールバーの最後のボタンをクリックすると、作成した数式を保存し、別の QGIS プロジェクトで読み込むこともできます。

12.2.6 結論

このレッスンでは、GRASS が提供する数多くのツールのほんの一部を紹介しました。自分で GRASS の機能を調べるには *GRASS* **ツール** ダイアログを開き、 **モジュールリスト** をスクロールしてください。あるいは、より構造的なアプローチとして、ツールの種類ごとに整理されている **モジュールツリー** タブの下を見てください。

第13章　モジュール: 学習評価

このセクション用にはご自身のデータをお使いください。必要なのは：

- POI（ポイント名と複数のカテゴリ）のポイントベクタデータセット
- 道路のラインベクタデータセット
- 土地利用（土地の境界を使用）のポリゴンベクタデータセット
- （航空写真のような）視覚的なスペクトル画像
- DEM（お持ちでない場合は CGIAR-CSI からダウンロードできます）

13.1 基図を作る

データ解析を行う前に、解析結果に文脈を与える基図が必要です。

13.1.1 ポイントレイヤを追加する

- ポイントレイヤを追加します。行っている課程のレベルに基づき、以下の該当するセクションに記載されているものだけ行います：

★☆☆ （初級レベル）

- 点に対して、場所名などのユニークな属性に応じてラベルを付けます。ラベルには小さいフォントを使用して目立たないようにします。情報は利用可能である必要がありますが、地図の主な地物であってはなりません。
- ポイント自体はカテゴリに基づいて異なる色に分類します。例えば、カテゴリには「観光地」、「警察署」、「街中心」などがありうるでしょう。

★★☆ （中級レベル）

- ★☆☆ （初級レベル） セクションと同じことをしてください。
- 重要性によって、ポイントサイズを分類します：より重要な地物はより大きなポイントで。しかしながら、サイズは 2.00 ポイントを超えないようにしてください。
- ポイント１つだけに位置づけられない地物については（例えば、地域/地方の名称、または大縮尺での町名）、ポイントは何も割り当てないでください。

★★★ （上級レベル）

- レイヤをシンボル化するためにポイントシンボルは使用しないでください。代わりに、ポイントの中央にラベルを使用してください。ポイントシンボル自体にはサイズがありません。
- **データで定義される設定** を使用してラベルを意味のあるカテゴリにスタイル付けします。
- 必要に応じて、属性データに適切な列を追加します。その際は架空のデータを作成しないこと - むしろ、 **フィールド計算機** を使用し、データセット内の適切な既存の値に基づいて新しい列を投入します。

13.1.2 ラインレイヤを追加する

- 道路レイヤを追加して、そのシンボルを変更します。道路にラベルを付けないでください。

★☆☆ （初級レベル）

- 幅広い線の明るい色に道路シンボルを変更します。また、やや透明にします。

★★☆ （中級レベル）

- 複数のシンボルレイヤを持つシンボルを作成します。結果のシンボルは実際の道路のように見えるはずです。これには単純な記号を使用できます。例えば、薄い白い実線が中央を走っている黒い線。より精巧なものにすることもできますが、結果として得られる地図が煩雑に見えるべきではありません。
- 地図を表示したい縮尺でデータセット中の道路の密度が高い場合は、道路のレイヤは 2 つ持っている必要があります：精巧な道路のような記号、およびより小さな縮尺での単純な記号。（適切な縮尺で切り替えるためには縮尺ベースの可視性を使用します。）。
- すべてのシンボルが複数のシンボルレイヤを持っている必要があります。それらを正しく表示するために記号を使用します。

★★★ （上級レベル）

- 上の ★★☆ （中級レベル） セクションと同じことをしてください。
- すべてのシンボルが複数のシンボルレイヤを持っている必要があります。それらを正しく表示するために記号を使用します。

13.1.3 ポリゴンレイヤを追加する

- 土地利用レイヤを追加し、そのシンボルを変更します。

★☆☆ （初級レベル）

- 土地利用に応じてレイヤを分類します。ソリッドカラーを使用してください。

★★☆ （中級レベル）

- 土地利用に応じてレイヤを分類します。適切な場合には、シンボルレイヤ、異なるシンボルタイプ等を組み込みます。しかしながら、結果が落ち着いた均一に見えるよう維持してください。これは背景の一部になることに留意してください！

★★★ （上級レベル）

- 「都市」、「農村」、「自然保護区」、などの一般的なカテゴリに土地利用を分類するために、規則に基づく分類を使用します

13.1.4 ラスタ背景を作成する

- DEM から陰影起伏を作成し、それを DEM 自体の分類されたバージョンのオーバーレイとして使用します。*Relief* プラグインも（プラグインのレッスン中で示すように）使用できるでしょう。

13.1.5 基図を完成させる

- 上記のリソースを使用して、レイヤのいくつかまたはすべてを使用して、基図を作成します。この地図は、すべての基本的なユーザーが方角を定めるために必要な情報を含むだけでなく、視覚的に統一されている/「単純」である必要があります。

13.2 データを分析する

- あなたは一定の基準を満たす土地を探しています。
- 自身の基準で決めることができますが、基準は文書化しておかなければなりません。
- これらの基準のためのいくつかのガイドラインがあります。
  - 対象となる土地は、土地利用の特定のタイプのものでなければなりません
  - それは道路から一定の距離内にあるか、あるいは道路が交差していなければなりません
  - それは、いくつかのポイントのカテゴリ、例えば病院などのように、から一定の距離内になければなりません

13.2.1 ★★☆ （中級レベル） / ★★★ （上級レベル）

- 検索結果のラスタ　　分析を含めます。ラスタの少なくとも一つの派生プロパティ、その傾斜方向や傾きなど、を考えます。

13.3 最終的な地図

- **印刷レイアウト** を使用して、分析結果を組み込んだ最終的な地図を作成します。
- 文書化基準に沿って文書にこの地図を含めます。地図が追加したレイヤーによってあまりにも視覚的に煩雑になりすぎたら、必要以上と感じるレイヤーの選択を解除します。
- 地図には、タイトルと凡例を含める必要があります。

第14章　モジュール: 林業への応用

モジュール 1 から 13 では、QGIS について、および QGIS でどのように作業するかをすでにたくさん学びました。基本的な GIS の林業への応用について学ぶことに関心がある方は、このモジュールを習うと、これまで学んできたことを応用する能力が身につきますし、また役立つ新しいツールをお見せします。



このモジュールの開発は EU にスポンサリングされました。

14.1 レッスン: 林業モジュールの紹介

林業への応用についてのこのモジュールを理解するには、このトレーニングマニュアルのモジュール 1 から 11 を通じて学んだ知識が必要です。この後のレッスンにある演習では、読者はすでに QGIS での基本的な操作の多くはできると仮定し、前に使用されていないツールだけが詳細に紹介されます。

にもかかわらず、QGIS での以前の経験を持っている方であればおそらく問題なく指示に従うことができるよう、モジュールはレッスンを通して基本的なレベルに従います。

このモジュールの追加のデータパッケージをダウンロードする必要があることに注意してください。

14.1.1 林業のサンプルデータ

> **注釈**
>
> このモジュールで使われているサンプルデータは、トレーニングマニュアルデータセットの一部で、`exercise_data\forestry\` フォルダにあります。

林業関連のサンプルデータ（林業地図、森林データ）は、EVO-HAMK 林業学校 から提供されたものです。データセットは、レッスンのニーズに合わせて修正されています。

一般的なサンプルデータ（航空写真、LiDAR データ、基本地図）は、フィンランド国土調査所のオープンデータサービスから入手し、演習の目的に適合させたものです。そのオープンデータファイルダウンロードサービスは、こちら から英語でアクセスできます。

> **⚠ 警告**
>
> トレーニングマニュアルの残りの部分については、このモジュールは、GIS データセットの追加、削除および変更の指示が含まれています。私たちは、その目的のためにトレーニングデータセットを提供しています。ここに記載されている技法をあなたのデータに使用する前に、必ず適切にバックアップをとっていることを確認してください！

14.2 レッスン: 地図をジオリファレンスする

林業においてよくある作業は、森林地域についての情報の更新でしょう。ある地域についての前回の情報が何年も前のもので、アナログで（つまり紙で）集められていたり、デジタイズされたが残っているのはその目録データの紙版しかなかったり、といったことがあり得ます。

その情報を、例えば後の目録と比較するために、GIS で使用したいと思うことはよくあります。これは、GIS ソフトウェアを使って手元の情報をデジタイズする必要があることを意味します。しかしデジタイズを始める前に、紙の地図をスキャンしジオリファレンスするという重要な最初のステップがあります。

このレッスンの目標： QGIS でジオリファレンスサツールを使用する方法を学ぶ。

14.2.1 ★☆☆ （初級レベル） 地図をスキャンする

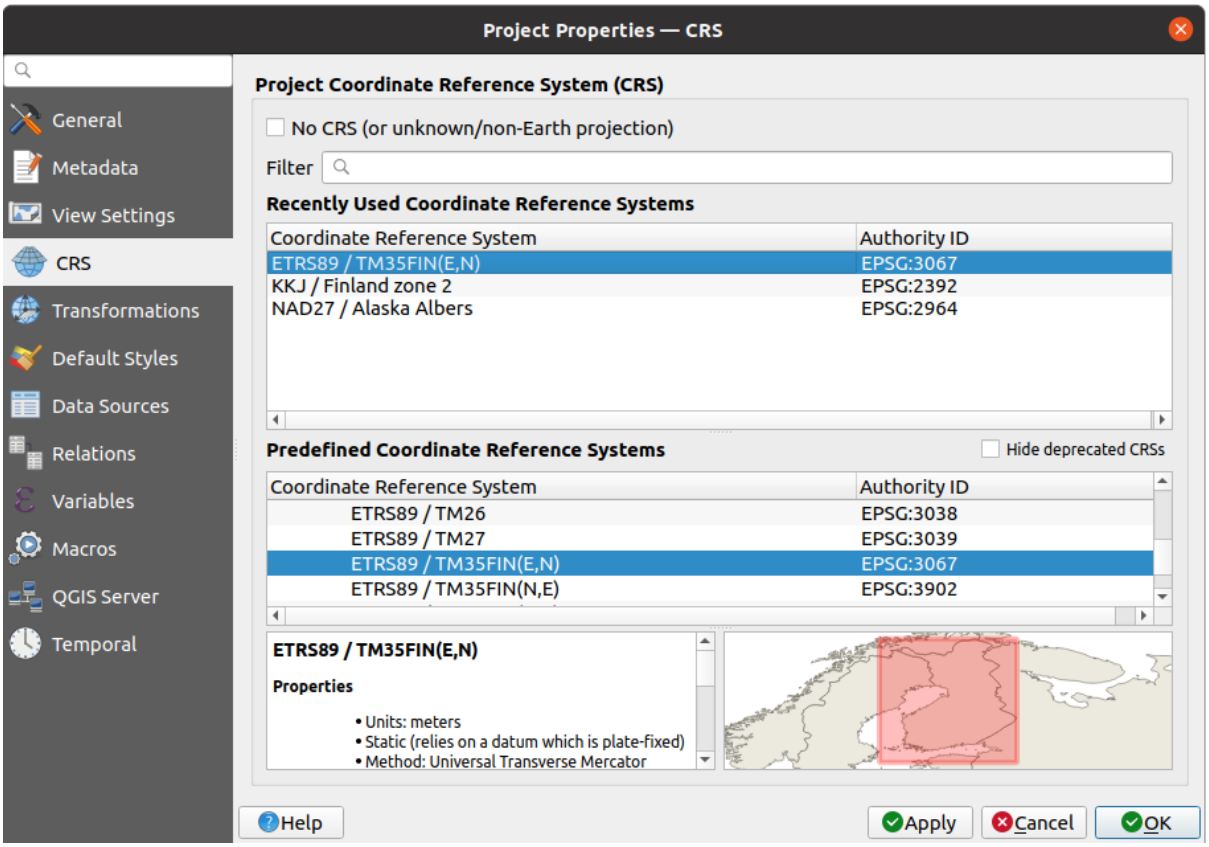
最初にしなければならない仕事は、地図をスキャンすることです。地図が大きすぎる場合は、分割してスキャンできますが、各部分について前処理とジオリファレンスの作業を繰り返す必要があることに留意してください。ですから、可能な限り少ない分割で地図をスキャンします。

このマニュアルで提供されるものと異なる地図を使用する場合は、ご自分のスキャナを使い、300 DPI の解像度で地図を画像ファイルとしてスキャンしてください。地図が色付きの場合は、カラーで画像をスキャンし、後でそれらの色を使用して地図からの情報を別々のレイヤ（例えば、林分、等高線、道路...）に分離できるようにしておきます。

この演習では､以前にスキャンした地図を使用します。この地図はデータフォルダ `exercise_data/forestry` に `rautjarvi_map.tif` として格納されています

14.2.2 ★☆☆ （初級レベル） 理解しよう: スキャンした地図をジオリファレンスする

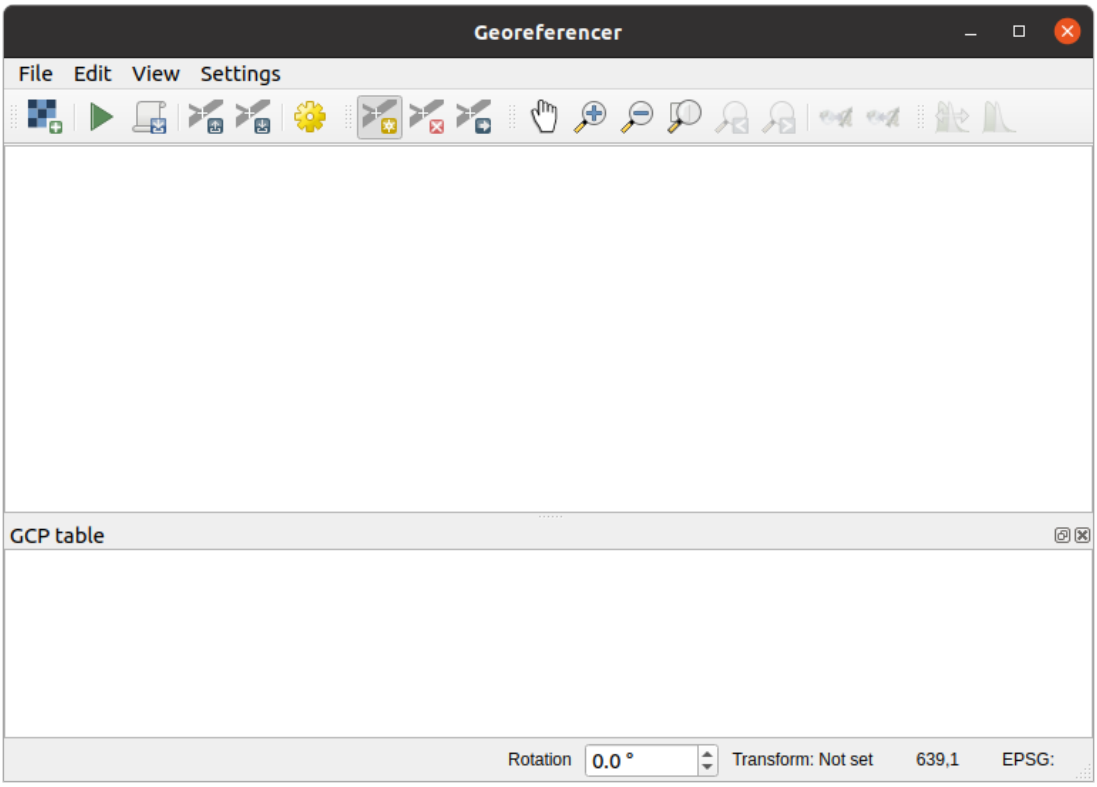
1. QGIS を開き、**プロジェクト** **プロパティ** *CRS* で、プロジェクトの CRS を `EPSG:3067 - ETRS89 / TM35FIN(E,N)` に設定します。これは現在フィンランドで使用されている CRS です。



2. QGIS プロジェクトを `map_digitizing.qgs` として保存します。

QGIS のジオリファレンスツールである **ジオリファレンサ** を使用します。地図をジオリファレンスするには:

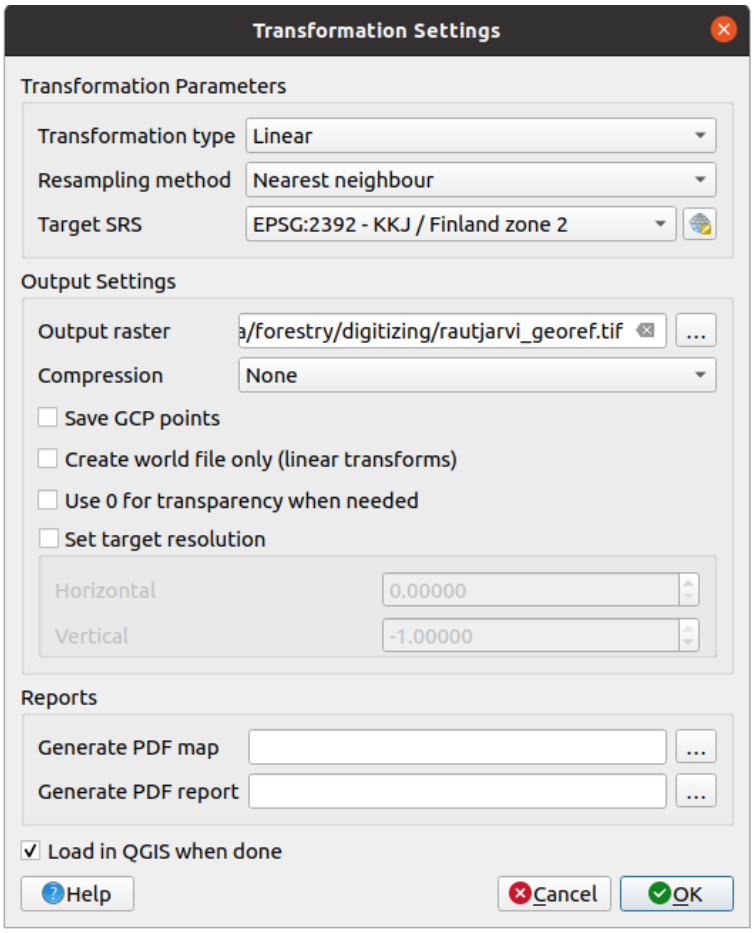
1. **レイヤ** **ジオリファレンサ**... でジオリファレンスツールを開きます。



2. 地図画像ファイル、 `rautjarvi_map.tif` をジオリファレンスする画像として追加します: **ファイル** ▸ **ラスタを開く** 。
3. *OK* をクリックします。

次に地図をジオリファレンスするための変換設定を定義する必要があります:

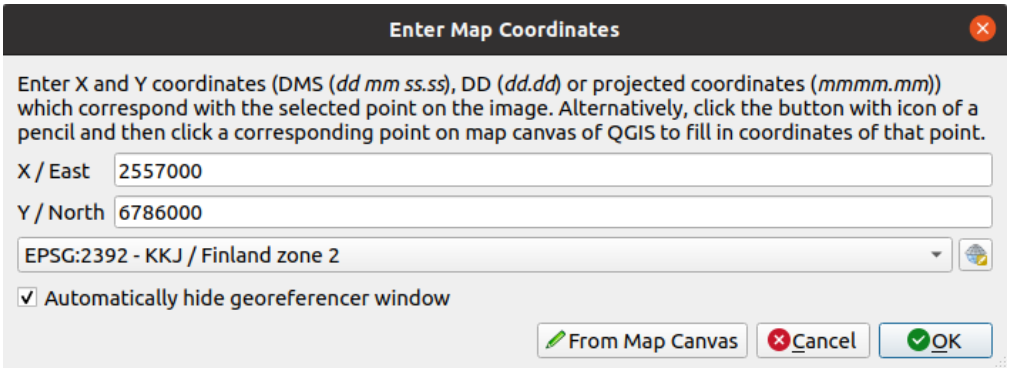
1. **設定** ▸ **変換を設定** を開きます。
2. **変換型** を 線形 に、 **リサンプリング方法** を 最近傍 に設定します。
3. **変換先** *CRS* オプションの隣にある CRS を選択 ボタンを押し、`EPSG:2392 - KKJ / Finland zone 2` CRS を選択します。これはこの地図が作られた 1994 年当時のフィンランドで使われていた CRS です。
4. **出力ファイル** ボックスの横にあるアイコンをクリックし、フォルダに移動して `exercise_data\forestry\digitizing` フォルダを作成し、ファイル名を `rautjarvi_georef.tif` にします。
5. **完了後にプロジェクトに読み込む** をチェックします
6. 残りのパラメータはデフォルトのままにします。



7. *OK* をクリックします。

この地図にいくつかある、座標を示す十字線を使って画像をジオリファレンスします。QGIS で通常行うズームやパンのツールを使って、ジオリファレンサ・ウインドウに表示された画像を確認することができます。

1. 地図の左下隅に拡大して、前述したように `KKJ / Finland zone 2` CRS による X と Y の座標ペアを持つ十字線があることに注意してください。この点は地図のジオリファレンスに使用する最初の地上基準点（グランドコントロールポイント）として使用します。
2. **点を追加** ツールを選択し、（必要に応じてパンとズームして）十字線の交点をクリックします。
3. **地図座標の入力** ダイアログに、地図上に表示される座標（X: 2557000, Y: 6786000）と、その CRS（`EPSG:2392 - KKJ / Finland zone 2`）を記入します



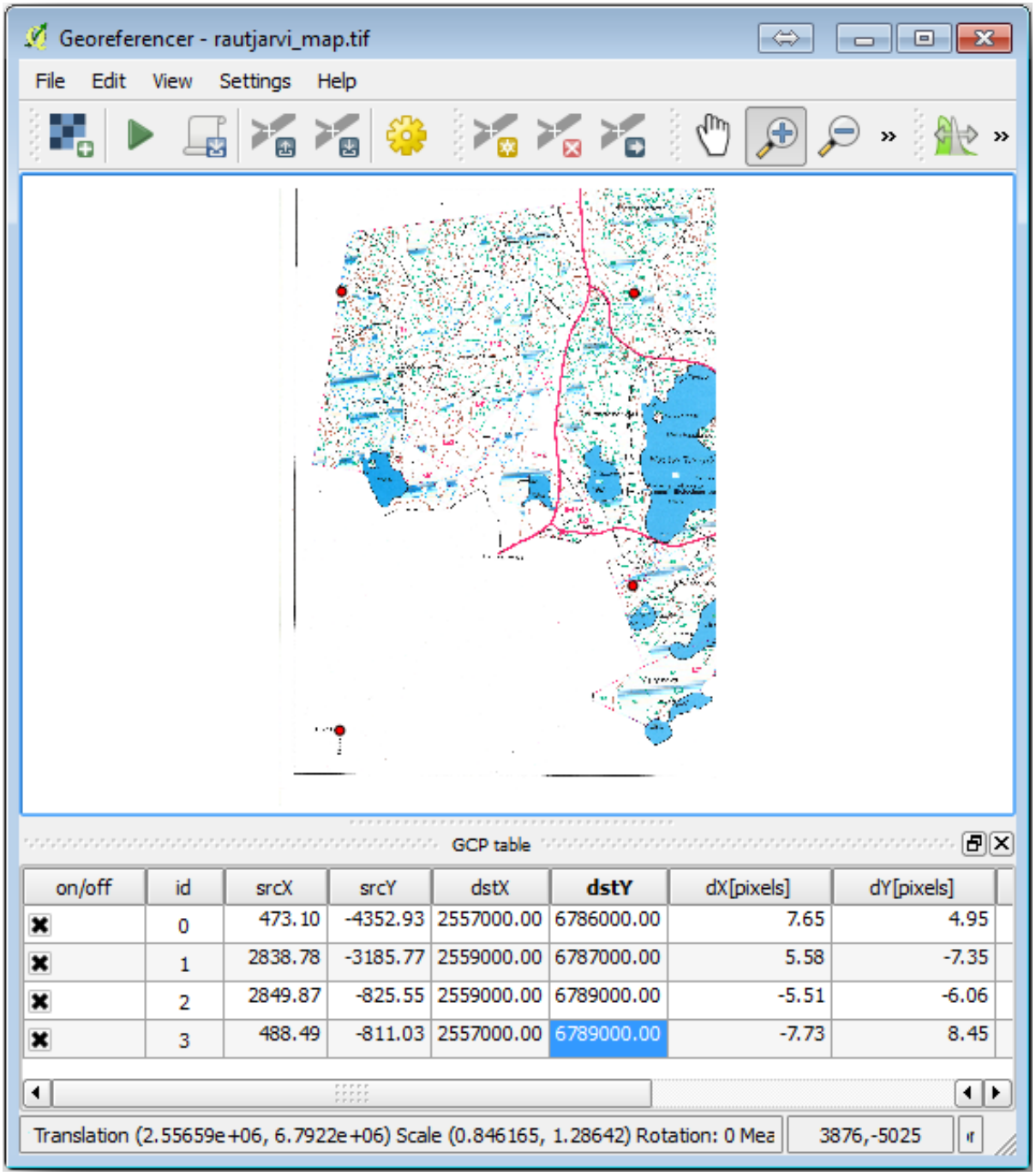
4. *OK* をクリックします。

   ジオリファレンスのための最初の座標の準備ができました。

5. 画像内の他の黒い十字線を探します。それらは北と東の方向にお互いに 1000 メートル離れています。これらの点の座標は、最初の点との関係で計算できるでしょう。

6. 画像を縮小して、他の十字線を見つけるまで右か上に移動し、何キロメートル移動したか推定してください。地上基準点はお互いにできるだけ離すようにしてください。

7. 少なくともあと 3 つの地上基準点を、最初のものと同じ方法でデジタイズします。次のようなものが出来上がるはずです:



すでに 3 つのデジタイズした地上基準点では、ジオリファレンスエラーを点から出る赤い線として見ることができます。ピクセル単位の誤差は、 *GCP* **テーブル** の *dX (***ピクセル***)* と *dY (***ピクセル***)* 列にも見ることができます。ピクセル単位の誤差が 10 ピクセルより大きくなってはいけません。もしそうであれば、デジタイズした点と入力した座標を見直して、何が問題なのかを探す必要があります。上の画像を参考にするとよいでしょう。

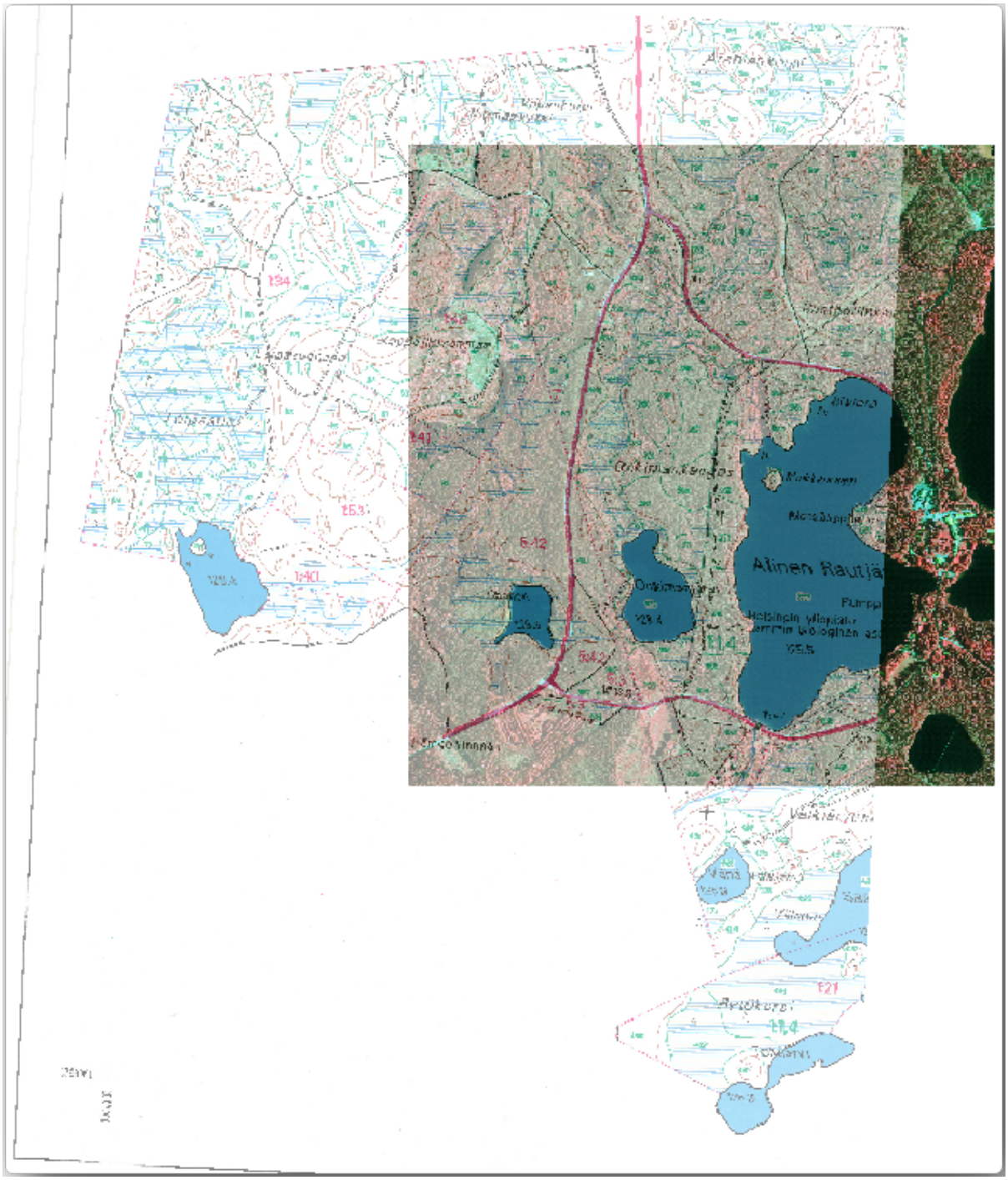
基準点が決まったら、後で使用するために保存しておきます:

1. **ファイル　名前をつけて保存**... に移動します。
2. フォルダ `exercise_data\forestry\digitizing` で、ファイルの名前を `rautjarvi_map.tif.points` にします。

最期に地図をジオリファレンスします:

1. **ファイル　ジオリファレンスを開始** に進みます。
2. ジオリファレンスの設定を編集したときに、ファイル名を `rautjarvi_georef.tif` にしたことに注意してください。

   これで QGIS プロジェクトにジオリファレンスされたラスタとして地図が表示されました。ラスタが少し回転しているように見えますが、これは単にデータが `KKJ / Finland zone 2` で、プロジェクトが `ETRS89 / TM35FIN(E,N)` であるためです。
3. データが正しくジオリファレンスされているかどうかを確認するには、以下の方法があります
   1. `exercise_data\forestry` フォルダにある、`rautjarvi_aerial.tif` という名前の航空写真を開きます。
   2. あなたの地図とこの画像は良く合うはずです: 地図の透明度を 50%に設定し、航空写真と比較してみてください。



4. QGIS プロジェクトへの変更を保存してください。次のレッスンはこの時点から続きます。

14.2.3 結論

これで紙地図をジオリファレンスし、QGIS で地図レイヤとして利用できるようになりました。

14.2.4 次は?

次のレッスンでは、地図上の林分をポリゴンとしてデジタイズし、目録データを追加します。

14.3 レッスン: 林分をデジタイズする

ジオリファレンスされた地図を単純な背景画像として使うのでなければ、次の自然なステップは、そこから要素をデジタイズすることです。それは **レッスン: 新しいベクタデータセットを作る** で学校の運動場をデジタイズしてベクタデータを作成する演習ですでにやっています。このレッスンでは、航空写真の代わりにジオリファレンスされた地図を使い、地図に緑色の線で表示されている林分境界線をデジタイズします。

このレッスンの目標： デジタイズ作業の助けになる技法を学び、林分をデジタイズし、最後にそれらに目録データを追加します。

14.3.1 ★☆☆ （初級レベル） 理解しよう: 林分境界を抽出する

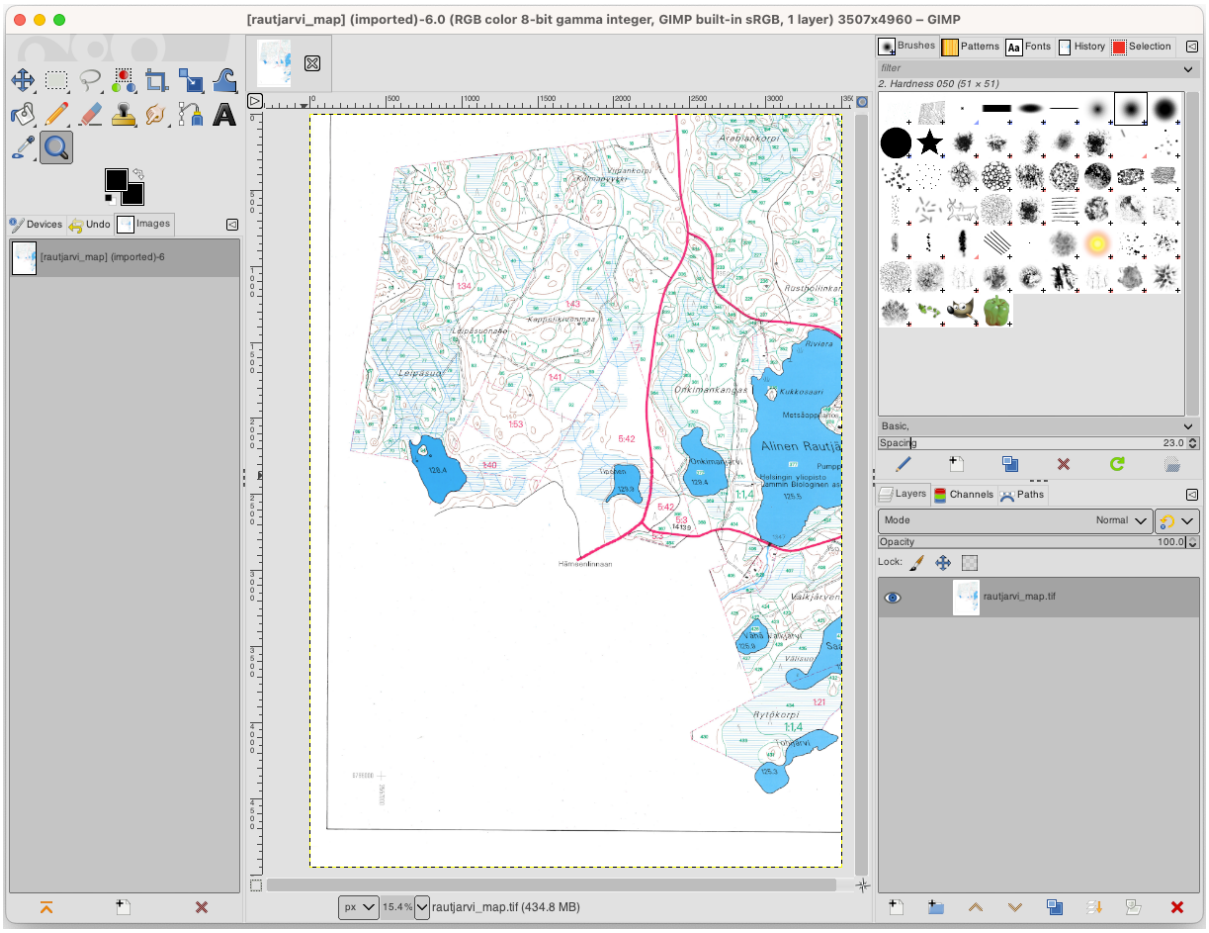
前のレッスンで保存した、`map_digitizing.qgs` プロジェクトを QGIS で開いてください。

地図をスキャンし、ジオリファレンスすれば、その画像をガイドとして見ながら直接デジタイズすることができます。デジタイズに使おうとしている画像が、例えば航空写真の場合には、それが最も適した方法でしょう。

ここでの場合のように、デジタイズに使用しているものが良い地図であれば、情報は要素の種類ごとに異なる色の線として明確に表示されているでしょう。これらの色は、 GIMP のような画像処理ソフトウェアを使用すれば、個別の画像として比較的簡単に抽出できます。このような個別の画像はデジタイズの支援に使用できます。それはこの後見ていきます。

最初のステップは、GIMP を使って林分だけ、つまりオリジナルのスキャン地図に見ることができる緑がかったすべての線、を含んだ画像を得ることです:

1. GIMP を開きます（まだインストールされていない場合は、インターネットからダウンロードするか、先生に頼んでください）。
2. 元の地図画像、`exercise_data/forestry` フォルダーにある `rautjarvi_map.tif` を **ファイル　開く** で開きます。林分が緑の線で表現されていることに注意してください（各ポリゴン内には林分の番号も緑で表示されています）。



3. これで林分の境界（緑がかったピクセル）を構成している画像内のピクセルを選択できます：

    1. ツール **選択**　**色域を選択** を開きます。

    2. このツールをアクティブにした状態で、画像を拡大し（`Ctrl` + マウスホイール）、林分の線が、ラインを形成するピクセルを区別できるほど大きくなるようにします。下の左の画像を見てください。

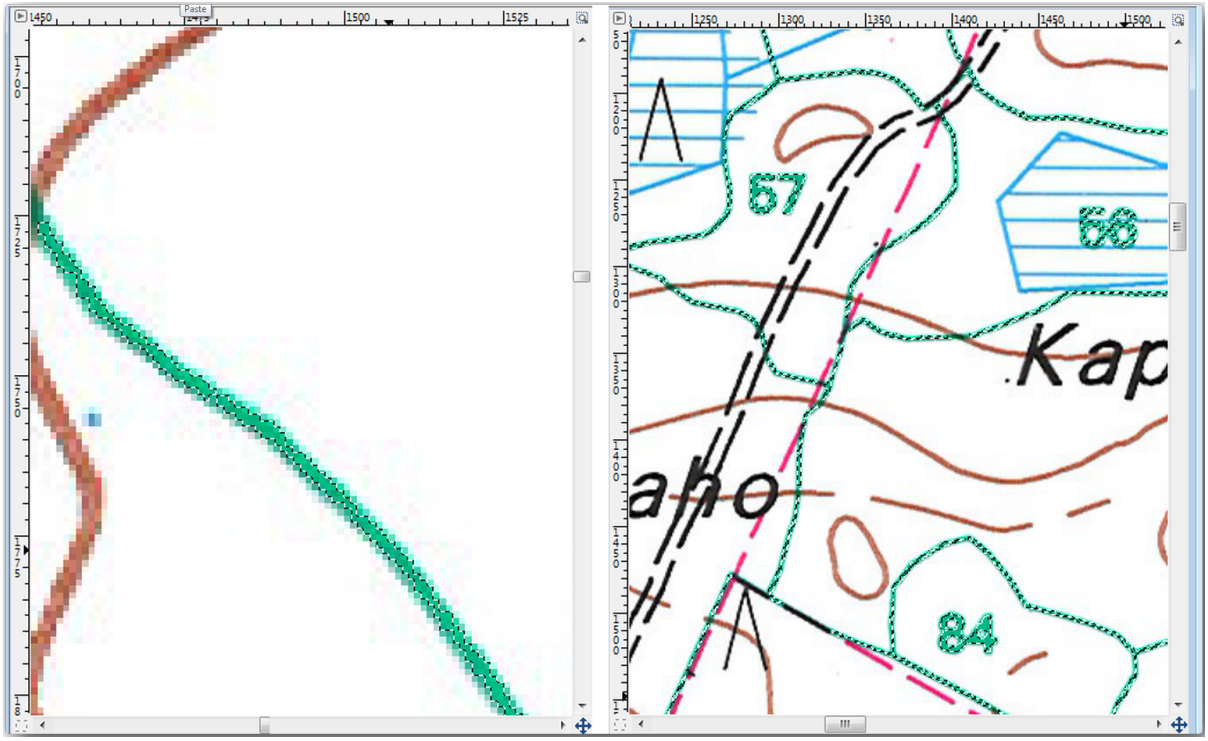
    3. 線の中央でマウスカーソルをクリックアンドドラッグし、ツールが、いくつかのピクセルの色の値を収集するようにします。

    4. マウスクリックを解除し、数秒待って。ツールによって収集された色と一致する画素が画像全体を通して選択されます。

    5. 画像全体で緑がかったピクセルが選択されていることを確認するために縮小します。

    6. 結果に満足できない場合は、クリックアンドドラッグ操作を繰り返します。

    7. ピクセルの選択は、下の右の画像のようになるはずです。

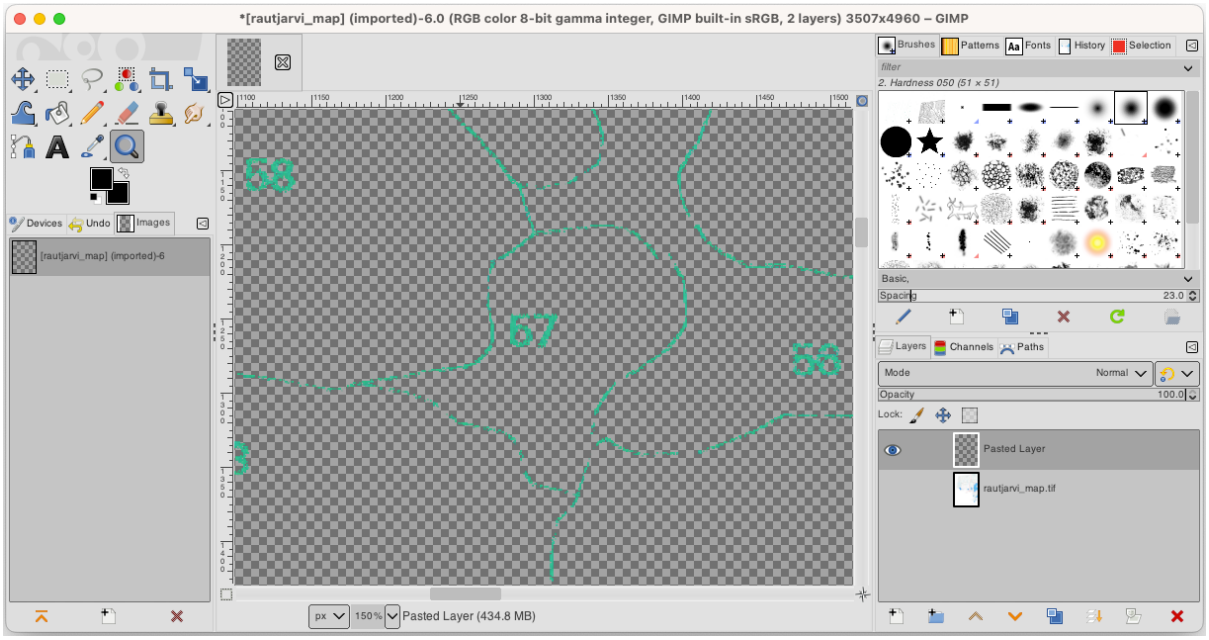


4. 選択を完了したら、新しいレイヤとしてこの選択をコピーして別の画像ファイルとして保存する必要があります。

    1. 選択したピクセルをコピーします（`Ctrl+C` または **編集**　**コピー**）

    2. そのピクセルを直接新しいレイヤに貼り付けます（**編集**　**貼り付け**　**新規レイヤ**）。GIMP は貼り付けられたピクセルを **レイヤ** パネルに新規レイヤ（貼り付けられたレイヤー）として表示します。

    3. Click the "eye" icon next to the original image layer to switch it off, so that only the `Pasted Layer` is visible:



5. 最後に **ファイル　エクスポート**... を選択し、 **ファイル形式の選択（拡張子で判別）** を *TIFF* **画像** とし、 `digitizing` フォルダーを選択して `rautjarvi_map_green.tif` という名前を付けます。聞かれたら、圧縮なしを選択します。

画像内の他の要素で同様の処理を行うことができます。例えば、道路を表す黒い線や地形等高線を表す茶色の線を抽出できるでしょう。しかし私たちにとっては林分で十分です。

14.3.2 ★☆☆（初級レベル）自分でやってみよう: 緑色画素の画像をジオリファレンス

前のレッスンで行ったように、この新しい画像は、データの残りの部分とともに使用できるようにするためにジオリファレンスする必要があります。

この画像は、ジオリファレンスツールに関する限り、基本的に元の地図画像と同じ画像であるため、地上基準点をデジタイズする必要はもうないことに注意してください。ここで、覚えておいていただきたいことがあります：

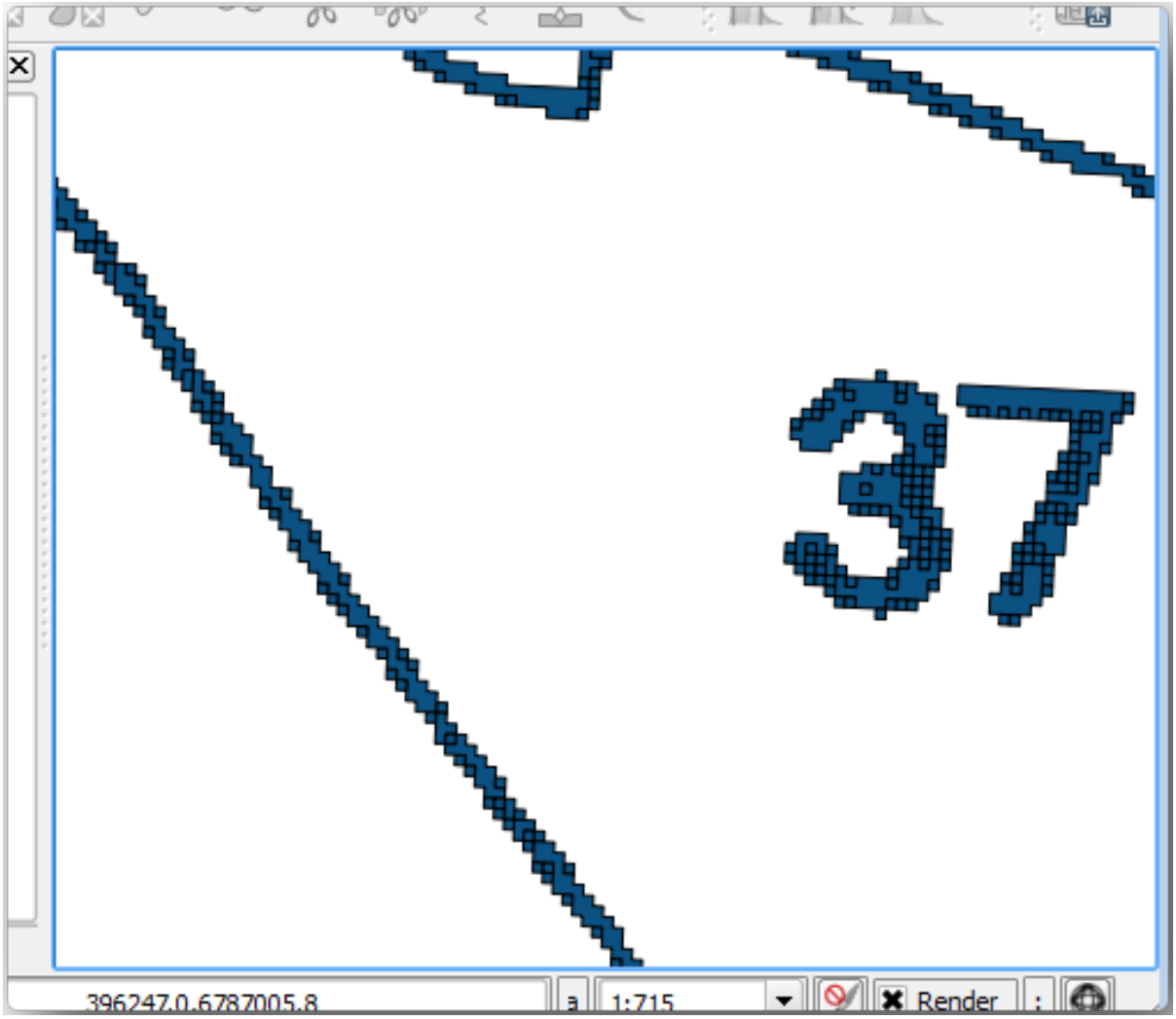
- このイメージも勿論 `KKJ / Finland zone2` CRS です。
- 保存した地上基準点を使用する必要があります、 **ファイル　*GCP* を読み込み** 。
- **変換設定** を確認することを忘れないでください。
- 出力ラスタの名前は `digitizing` フォルダーの `rautjavi_green_georef.tif` にします。

新しいラスタが元の地図ときちんと合っていることを確認してください。

14.3.3 ★☆☆ （初級レベル） 理解しよう: デジタイズを助けるポイントを作る

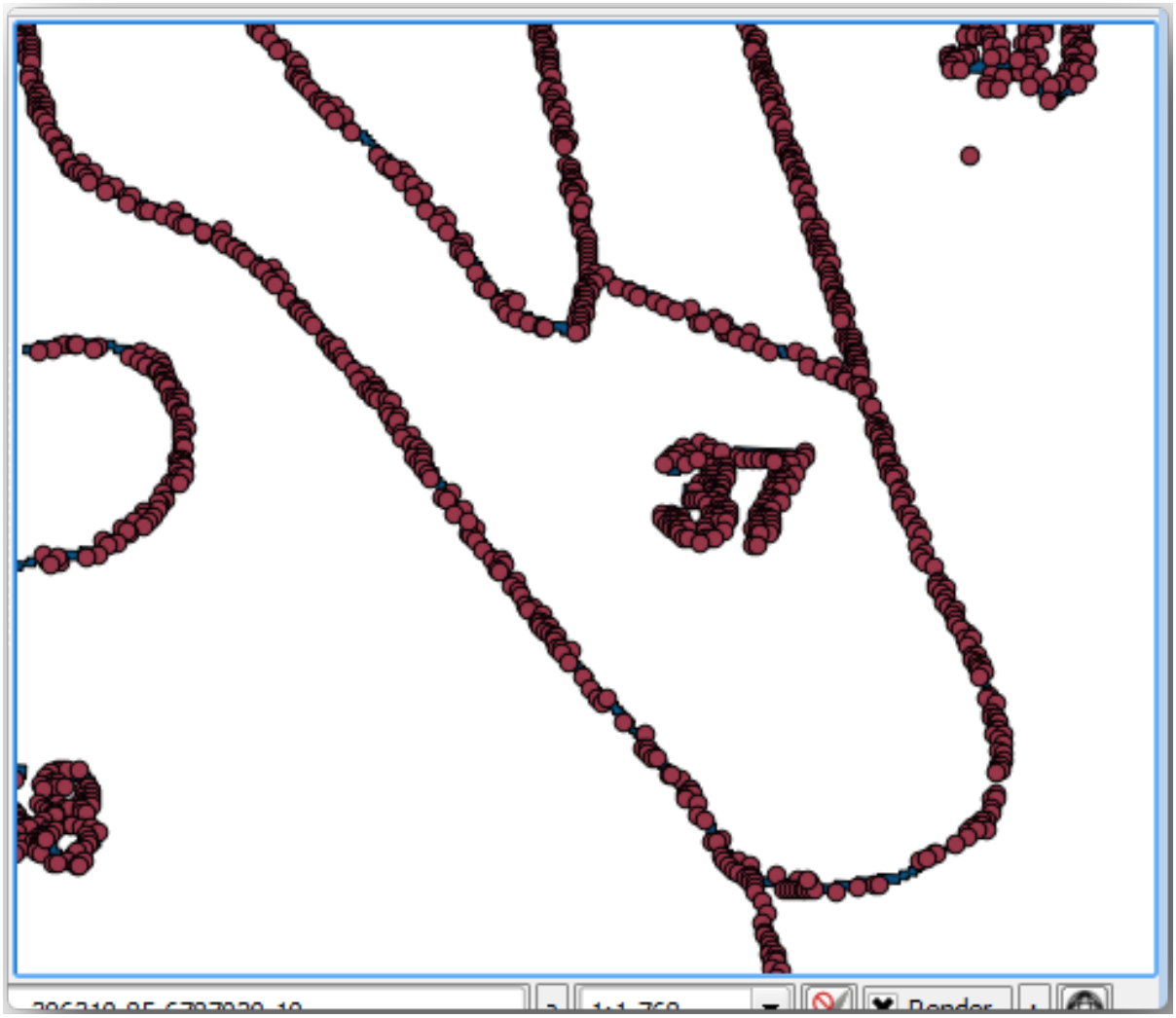
QGIS でのデジタイズツールが念頭にあると、デジタイズ中にこれらの緑のピクセルにスナップできたら便利だろうとすでにお考えかもしれません。それはまさに次にしようとしていること、つまり、QGIS で利用可能なスナップツールを使ってこれらのピクセルからポイントを作成し、後で林分の境界線に沿ってデジタイズするために使います。

1. **ラスタ　変換　ポリゴン化 (ラスタからベクタ)** ツールを使用して、緑の線をポリゴンに ベクタ化してください。やり方を覚えていない場合は、 **レッスン: ラスタからベクタへの変換** で確認できます。
2. `digitizing` フォルダーに `rautjarvi_green_polygon.shp` として保存します。
3. 拡大してポリゴンがどのように見えるかを見てください。このような物が見えるでしょう：



4. ポリゴンからポイントを得るための次の選択肢は、その重心を得ることです:
    1. **ベクタ　ジオメトリツール　重心...** を開きます。

2. Set *Input Layer* to rautjarvi_green_polygon (the polygon layer you have just created)
3. フォルダー digitizing の green_centroids.shp ファイルに **重心** の出力を設定します
4. **アルゴリズムの終了後に出力ファイルを開く** をチェックします
5. **実行** を押します。これでポリゴンの重心が新しいレイヤとして計算され、プロジェクトに追加されます。

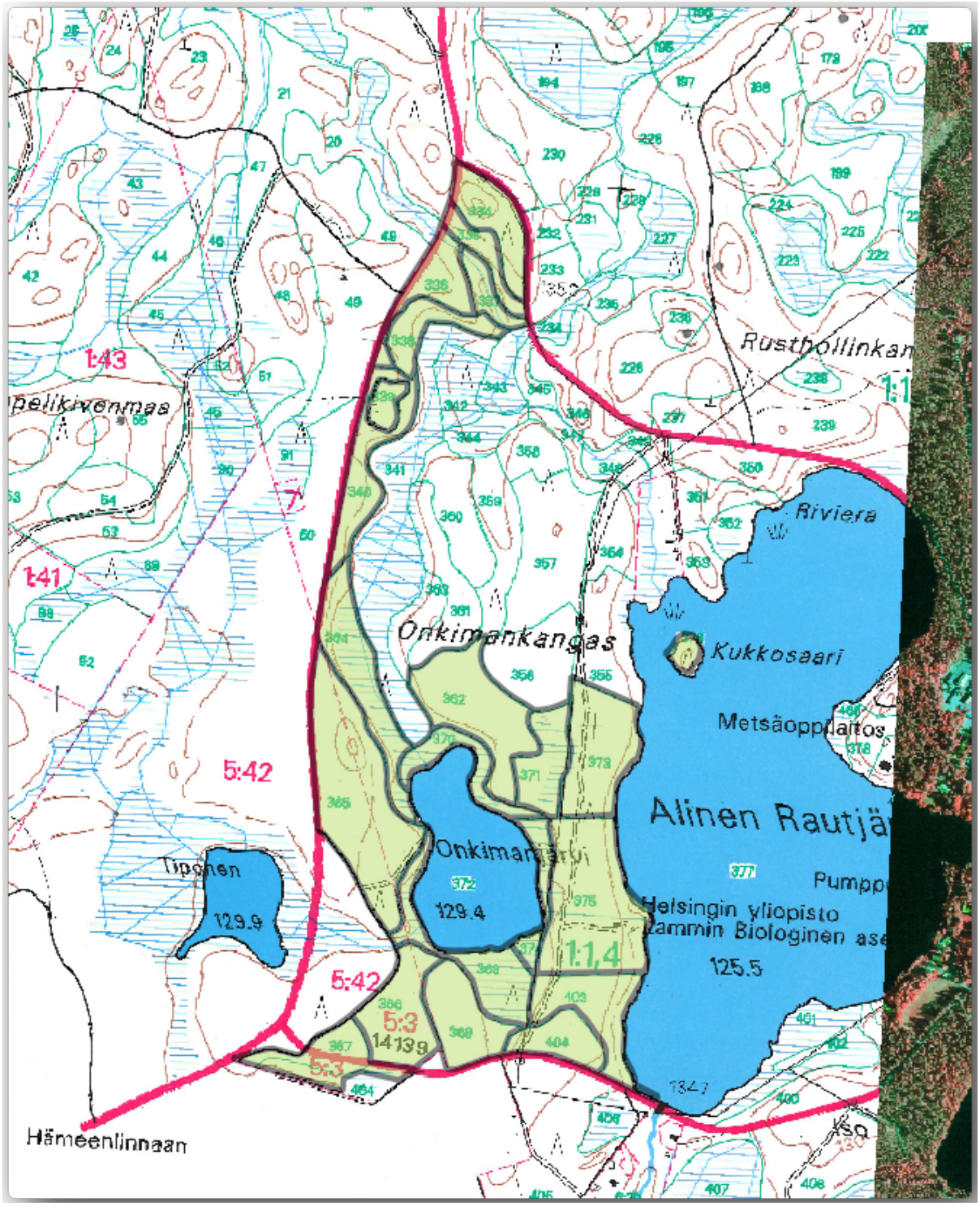


5. これで、rautjarvi_green_polygon レイヤを TOC から削除することができます。
6. 重心レイヤのシンボロジを次のように変えます:
   1. green_centroids の **レイヤプロパティ** を開きます。
   2. **シンボロジ** タブに移動します。
   3. **大きさ** を 1.00 にし、**地図単位** を選びます

ポイント１つ１つを区別する必要はありません。それらはスナップツールが使用するためそこにある必要があるだけです。これらのポイントを使用することで、それらがない場合よりもずっと簡単に元の線をたどれるようになりました。

14.3.4 ★☆☆ （初級レベル） 理解しよう: 林分をデジタイズする

これで実際のデジタイズ作業を開始する準備が整いました。 **ポリゴンタイプ** のベクタファイルの作成から始めますが、この演習では、関心域の一部をデジタイズしたシェープファイルがあります。主要道路（幅の広いピンクの線）と湖の間に残っている林分の半分のデジタイズを完了するだけです：

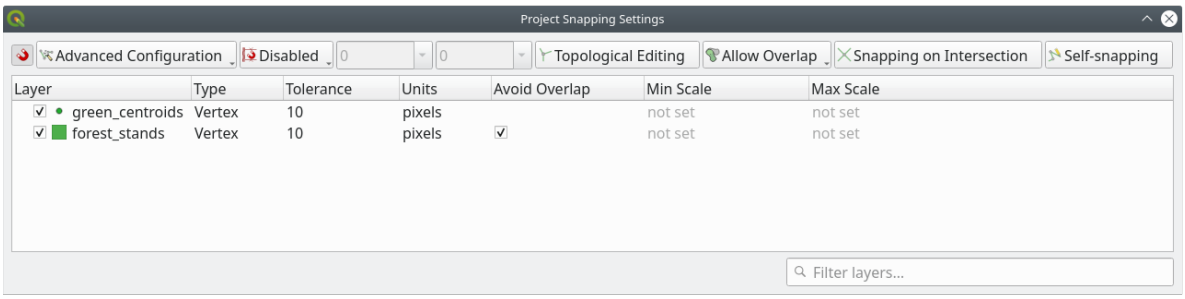


1. ファイルマネージャのブラウザで `digitizing` フォルダに移動します。
2. `forest_stands.shp` ベクタファイルを地図へドラッグ&ドロップします。
3. 新しいレイヤのシンボロジを変えて、デジタイズしたポリゴンがよく見えるようにします。

1. **塗りつぶし色** を緑にし、**不透明度** を 50% に変えます。
2. **シンプル塗りつぶし** を選んで **ストローク幅** を 1.00 mm にします。

さて、過去のモジュールを思い出すと、スナップオプションの設定と有効化が必要です:

1. **プロジェクト**　**スナップオプション**... に移動します
2.  スナップを有効にする を押し、**詳細設定** を選びます
3. Check the `green_centroids` and `forest_stands` layers
4. 各レイヤの **型** を :guilabel:`頂点` にします
5. 各レイヤの **許容範囲** を 10 にします
6. 各レイヤの **単位** を **ピクセル** にします
7. Check  *Avoid Overlap* for the `forest_stands` layer
8.  **トポロジ編集** を押します
9.  **詳細設定に従う** を選びます
10. ポップアップを閉じます



このスナップ設定により、デジタイズ中に重心レイヤのポイントやデジタイズしたポリゴンの頂点に近づくと、そのポイントにピンクの四角が表示されスナップされます。

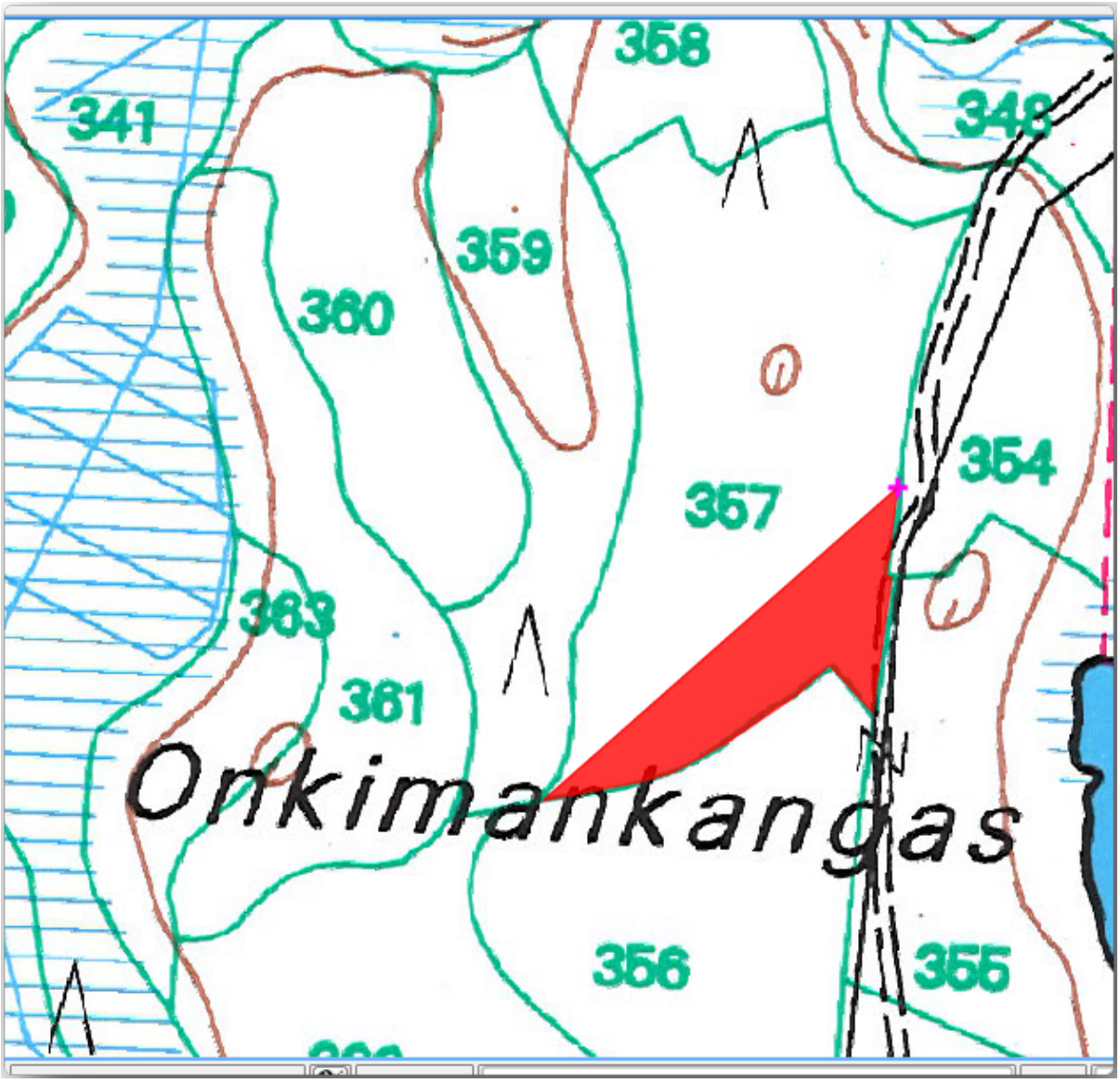
11. Finally, turn off the visibility of all the layers except `forest_stands` and `rautjarvi_georef`. Make sure that the map image has not transparency any more.

デジタイズを始める前に知っておく重要なポイント：

- 境界のデジタイズではあまりに正確にしようとしないでください。
- 境界が直線である場合は、2 つのノードだけでデジタイズしてください。一般的に、できるだけ少ないノードを使用してデジタイズしてください。
- 正確であることが必要と感じた場合のみ拡大して範囲を閉じてください。例えばいくつかの角や、ポリゴンを特定のノードで他のポリゴンと接続したいときです。
- デジタイズしながら、マウスの中ボタンを使用して拡大/縮小および地図を移動してください。
- 一度に 1 つのポリゴンをデジタイズしてください。
- 1 つのポリゴンをデジタイズした後、地図から読める林分 ID を書き込みます。

これでデジタイズを開始できます：

1. 林分番号 357 を地図ウィンドウに表示させます。
2. forest_stands レイヤを選択します。
3.  編集モード切り替え をクリックして編集を有効にします
4.  ポリゴン地物を追加 ツールを選びます。
5. 林分 357 のデジタイジングを始め、ドットのいくつかを繋ぎます。スナップを示すピンクの十字に注目してください。

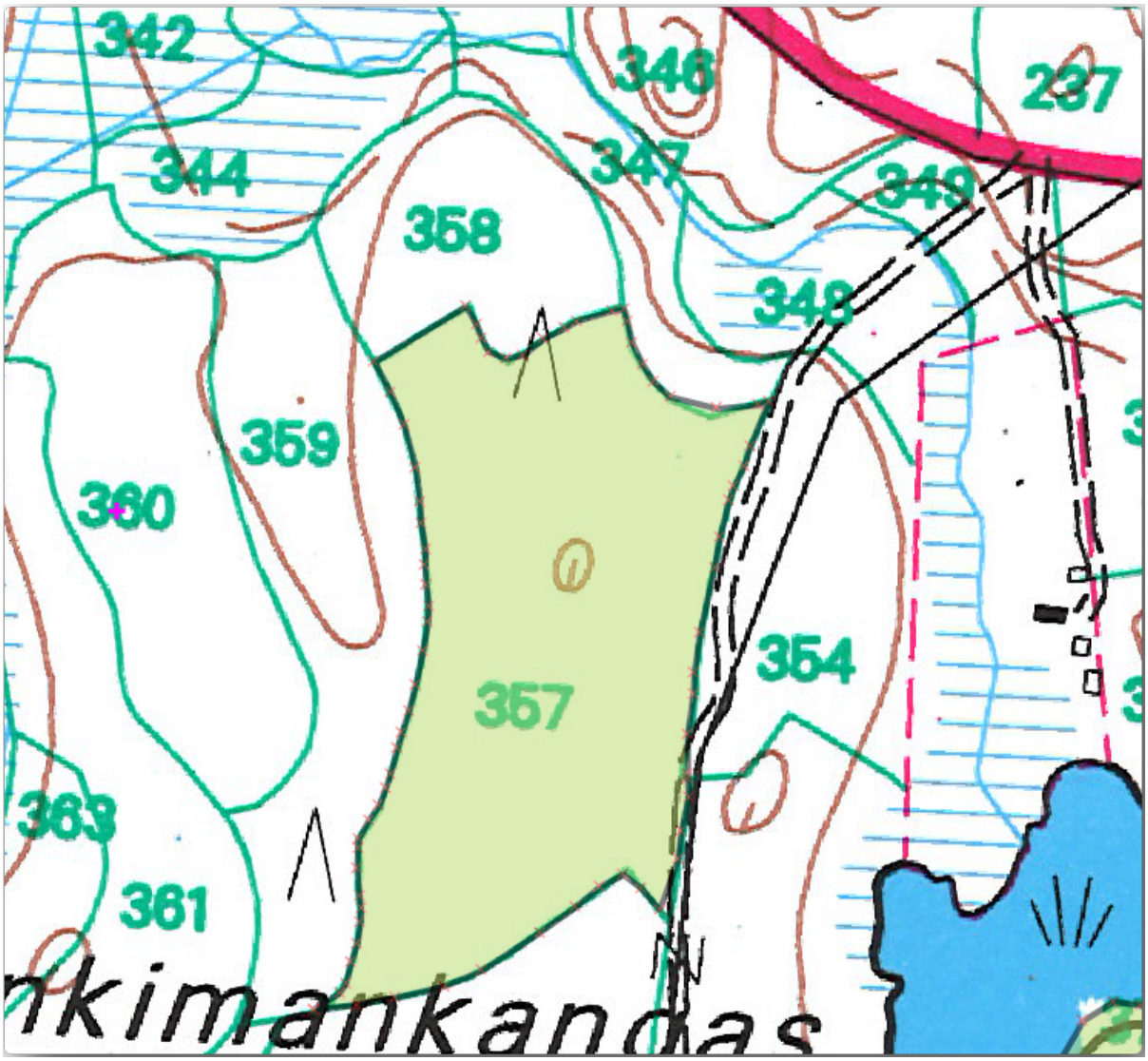


完了すると:

1. ポリゴンのデジタイジングを終わるには右クリックします。
2. フォームに林分 *ID* を入力します（この場合は 357）。
3. *OK* をクリックします。

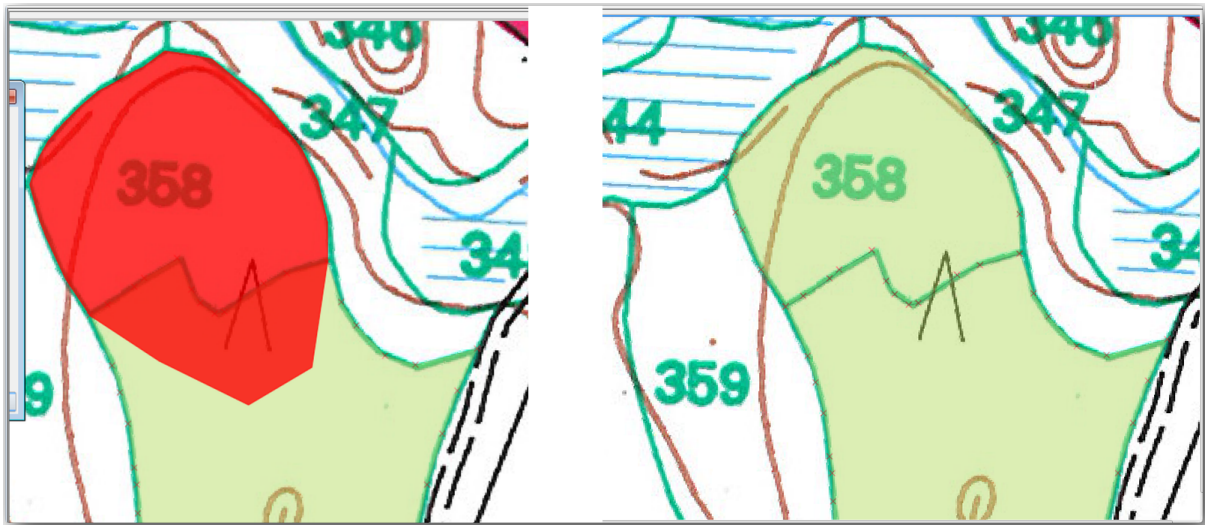
ポリゴンのデジタイジングを終了してもフォームが表示されない場合は、**設定　オプション　デジタイズ** で、**地物作成後に属性フォームをポップアップさせない** がチェックされていないことを確認してください。

デジタイズされたポリゴンは次のようになります:



Now for the second polygon, pick up the stand number 358. Make sure that *Avoid Overlap* is checked for the `forest_stands` layer (as shown above). This option ensures polygons do not overlap. So, if you digitize over an existing polygon, the new polygon will be trimmed to meet the border of the existing polygons. You can use this option to automatically obtain a common border.

1. 林分 357 と共通する角の一つで林分 358 のデジタイズを始めてください。
2. 両林分で共通なもう一つの角まで、普通に進めます。
3. 最後に、共通の境界が交差されていないことを確認しながらポリゴン 357 内のいくつかのポイントをデジタイズします。下の左の画像を参照してください。
4. 右クリックして林分 358 の編集を終了します。
5. *ID* を 358 と入力してください。
6. *OK* をクリックします。新しいポリゴンは、下の画像にあるように、林分 357 と共通の境界を持っている筈です。



既存のポリゴンと重なっていた部分は自動的に切り落とされ、意図した通りの共通の境界が残されています。

14.3.5 ★☆☆ （初級レベル） 自分でやってみよう: 林分のデジタイズを完了する

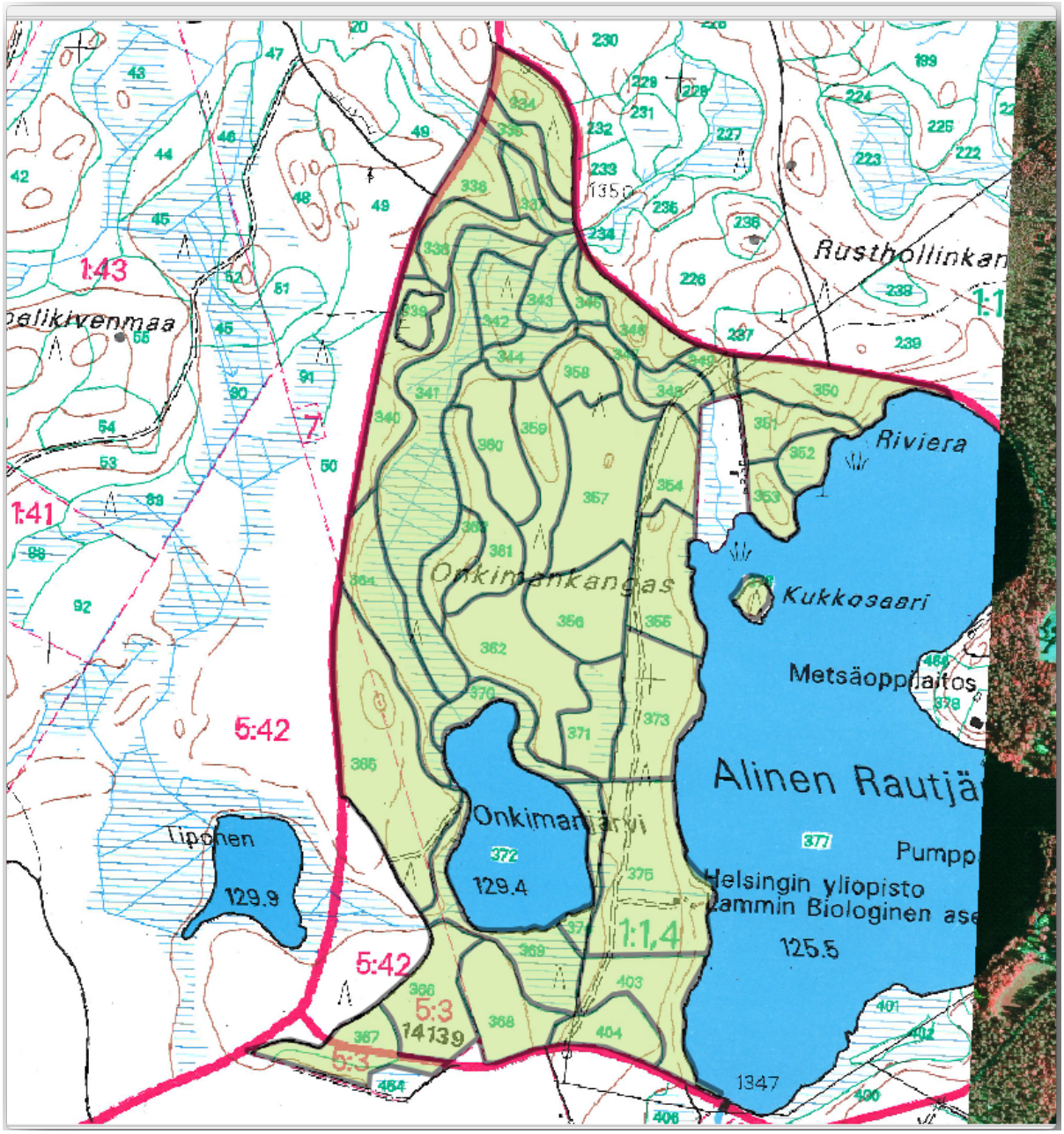
これで、2 つの林分ができました。そして、続行する方法について良いアイデア。主要道路と湖によって制限されているすべての林分がデジタイズされるまで、自分でデジタイズを続けてください。

それは大変な作業のように見えるかもしれませんが、すぐに林分をデジタイズするのに慣れるでしょう。それは約 15 分かかります。

デジタイズ中に、ポリゴンを編集したり削除したノードを、分割またはマージする必要がある場合があります。 **レッスン: 地物のトポロジ** で必要なツールについて学びましたが、今がそれらを読み直す良い機会でしょう。

トポロジの編集を有効 がアクティブな場合、2 つのポリゴンに共通のノードを移動すると、その共通の境界は、両方のポリゴンが同時に編集されるようになることを思い出してください。

結果は次のようになります:



14.3.6 ★☆☆ （初級レベル） 理解しよう: 林分データを結合する

地図のために持っている森林目録データは、紙に書かれている可能性があります。その場合は、最初にテキストファイルやスプレッドシートにそのデータを記述する必要があります。この演習では、1994 年目録（地図と同じ目録）からの情報は、カンマ区切りテキスト（CSV）ファイルとして準備ができています。

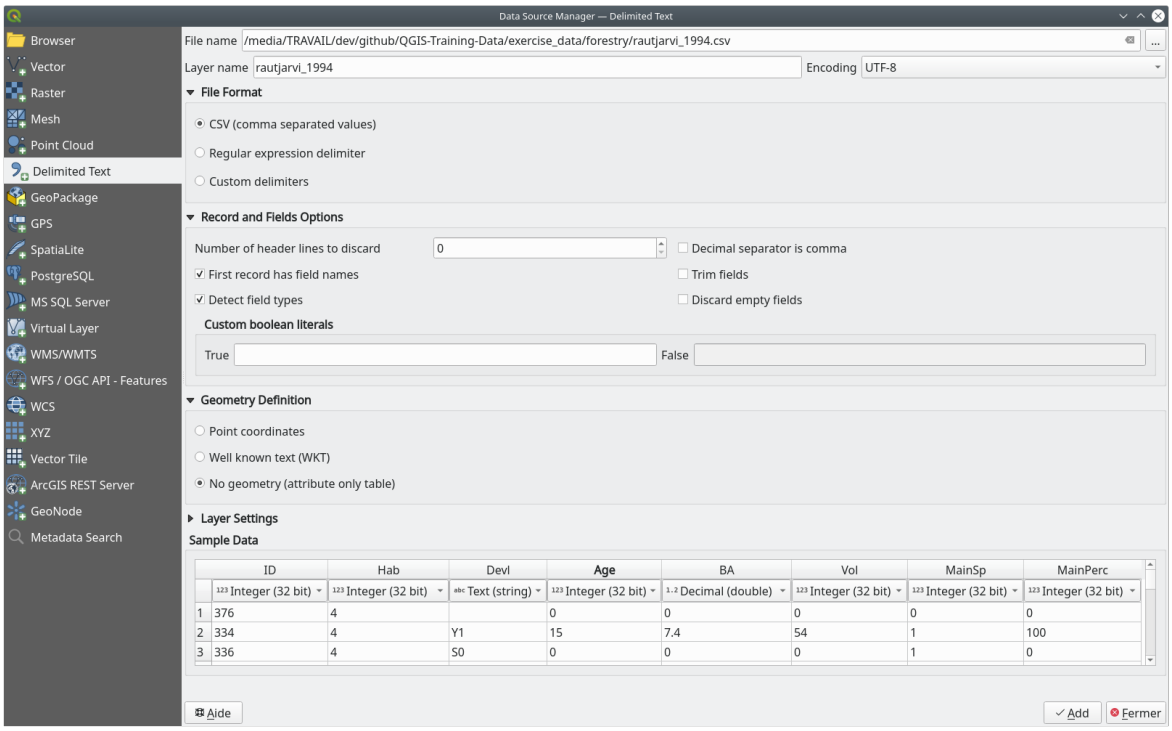
1. テキストエディタで `exercise_data\forestry` ディレクトリの `rautjarvi_1994.csv` ファイルを開き、インベントリデータファイルには *ID* という属性があり、林分の番号が書かれていることに注目します。これらの番号は、ポリゴンに入力した林分の ID と同じで、テキストファイルのデータをベ

クタファイルにリンクするために使うことができます。このインベントリデータのメタデータは、同じフォルダの `rautjarvi_1994_legend.txt` というファイルで見ることができます。

2. 次にこのファイルをプロジェクトに追加します:

   1. CSV テキストレイヤを追加 ツールを使用します。これは **レイヤ** **レイヤを追加** *CSV* **テキストレイヤを追加**... からアクセスできます。

   2. ダイアログで以下のように詳細を設定します:



   3. **追加** を押して、フォーマットされた *:file:`csv` ファイル*をプロジェクトに読み込みます。

3. `.csv` ファイルのデータとデジタル化されたポリゴンをリンクさせるために、2 つのレイヤーの間に結合を作成します:

   1. `forest_stands` レイヤのレイヤプロパティを開きます。

   2. **テーブル結合** タブに移動します。

   3. ダイアログボックスの下部にある 新規結合 (join) を追加 をクリックします。

   4. Select `rautjarvi_1994.csv` as the *Join layer*

   5. **結合基準の属性** に *ID* を設定します

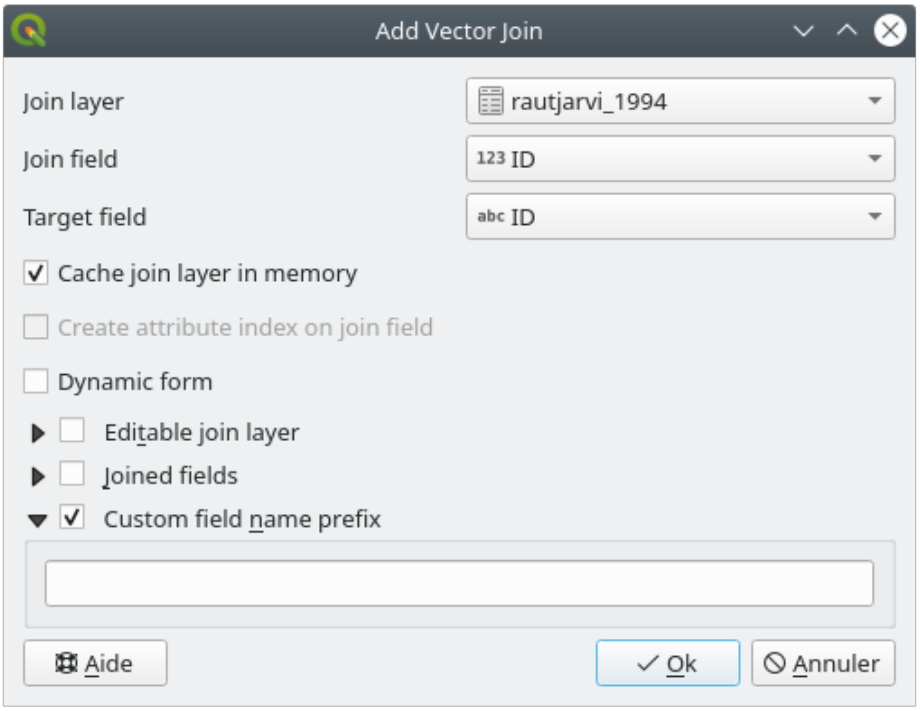
   6. **ターゲット属性** に *ID* を設定します

   7. *OK* を 2 回クリックします。

テキストファイルのデータがベクタファイルとリンクしているはずです。何が起こったかを見るには、`forest_stands` レイヤを選択し、 属性テーブルを開く を使ってください。目録データファイルのすべての属性が、デジタイズしたベクタレイヤにリンクされていることが確認できます。

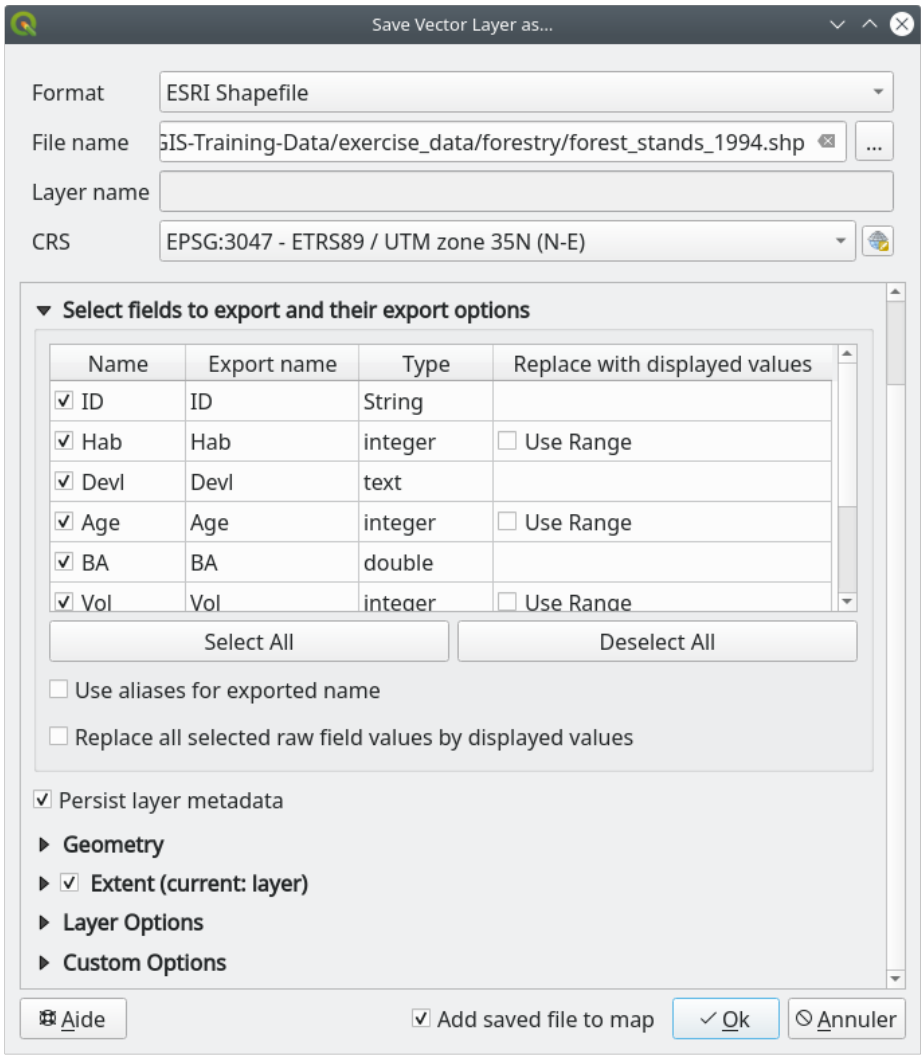
フィールド名の先頭に rautjarvi_1994_ が付いていることがわかります。これを変更するには:

1. forest_stands レイヤのレイヤプロパティを開きます。
2. **テーブル結合** タブに移動します。
3. Select rautjarvi_1994 as the *Join Layer*
4. 選択した結合を編集 ボタンをクリックして編集可能にします
5. **属性名の接頭辞** の下の接頭辞を取り除きます



.csv ファイルのデータは、ベクタファイルにリンクされているだけです。このリンクを恒久化し、実際にベクタファイルにデータを記録するには、forest_stands レイヤを新しいベクタファイルとして保存する必要があります。これを行うには:

1. forest_stands レイヤを右クリックします
2. **エクスポート　新規ファイルに地物を保存...** を選びます
3. **形式** に *ESRI Shapefile* を設定します
4. ファイル名を forestry フォルダ下の forest_stands_1994.shp に設定します
5. 新しいファイルをプロジェクトのレイヤとして含めるには、**保存されたファイルを地図に追加する** をチェックします



14.3.7 ★☆☆ （初級レベル） 自分でやってみよう: 面積と周囲の長さを追加する

これらの林分に関連する情報を収集するために、林分の面積と周囲の長さを計算することができます。あなたは **レッスン: 補足実習** で多角形の面積を計算しました。必要であれば、そのレッスンに戻り、林分の面積を計算してください。新しい属性に `Area` という名前を付け、計算された値がヘクタールであることを確認します。また、周囲の長さについても同じことができます。

これで `forest_stands_1994` レイヤの準備が整い、利用可能なすべての情報を詰め込むことができました。

プロジェクトを保存し、後で戻ってくる必要がある場合に備えて、現在の地図レイヤを保持します。

14.3.8 結論

これにはマウスを数回クリックする必要がありますが、これで古い登録データを QGIS で使用できるデジタル形式で手にいれました。

14.3.9 次は?

まっさらな新しいデータセットで異なる分析することで開始できますが、より最新のデータセットの分析を行うことにより興味があるのではないでしょうか。次のレッスンのトピックは、現在の航空写真やデータセットにいくつかの関連情報の追加を使用した林分の作成になります。

14.4 レッスン: 林分を更新する

古い目録の地図から情報をデジタイズし、林分に対応する情報を追加したので、次のステップは、森林の現在の状態の目録を作成することでしょう。

航空写真に従って、新しい林分をデジタイズします。前回のレッスンと同様に航空赤外線写真 (CIR) を使います。この種類の画像は、青色光の代わりに赤外光を記録したもので、植生域の調査に広く利用されています。

林分をデジタイズした後、このような保全条例により与えられた新たな制約などの情報を追加します。

このレッスンの目標： 新しい林分の集合を CIR 航空写真からデジタイズし、他のデータセットから情報を追加します。

14.4.1 ★☆☆ （初級レベル） 古い林分を現在の航空写真と比較する

フィンランド国土地理院はオープンデータ政策をとっており、航空写真、従来の地形図、DEM、LiDAR データなど、様々な地理データをダウンロードすることができます。このサービスは、ここ から英語でアクセス可能です。この演習で使用した航空画像は、このサービスからダウンロードした 2 枚のオルソ CIR 画像 (M4134F_21062012 と M4143E_21062012) から作成したものです。

1. QGIS を開き、プロジェクトの CRS を **プロジェクト　プロパティ...**　*CRS* で、 *ETRS89 / ETRS-TM35FIN* に設定します
2. CIR 画像 `rautjarvi_aerial.tir` をプロジェクトに追加する:
    1. ファイルマネージャのブラウザで `exercise_data\forestry\` フォルダに移動します
    2. `rautjarvi_aerial.tif` をプロジェクトにドラッグ＆ドロップします
3. QGIS プロジェクトを `digitizing_2012.qgs` として保存します

CIR の画像は 2012 年のものです。1994 年に造られた林分をほぼ 20 年後の状況と比較することができます。

1. 前のレッスンで作成した `forest_stands_1994.shp` レイヤを追加する:

    1. ファイルマネージャのブラウザで exercise_data\forestry\ フォルダに移動します
    2. プロジェクトにファイル forest_stands_1994.shp をドラッグ＆ドロップします
2. ポリゴンを透過して見えるようにレイヤのシンボロジを設定する:
    1. Right click forest_stands_1994
    2. **プロパティ** を選ぶ
    3. **シンボロジ** タブに移動する
    4. **塗りつぶし色** を透明な塗りつぶしに設定する
    5. **ストローク色** を紫に設定する
    6. **ストローク幅** を 0.50 mm に設定する
3. 均質な森として解釈する可能性があるものに、古い林分がどのように従っているか（あるいは従っていないか）を確認してください。
4. ズームとパンしてください。古い林分でなお静止画に対応するものとしないものがあることに気づくででしょう。

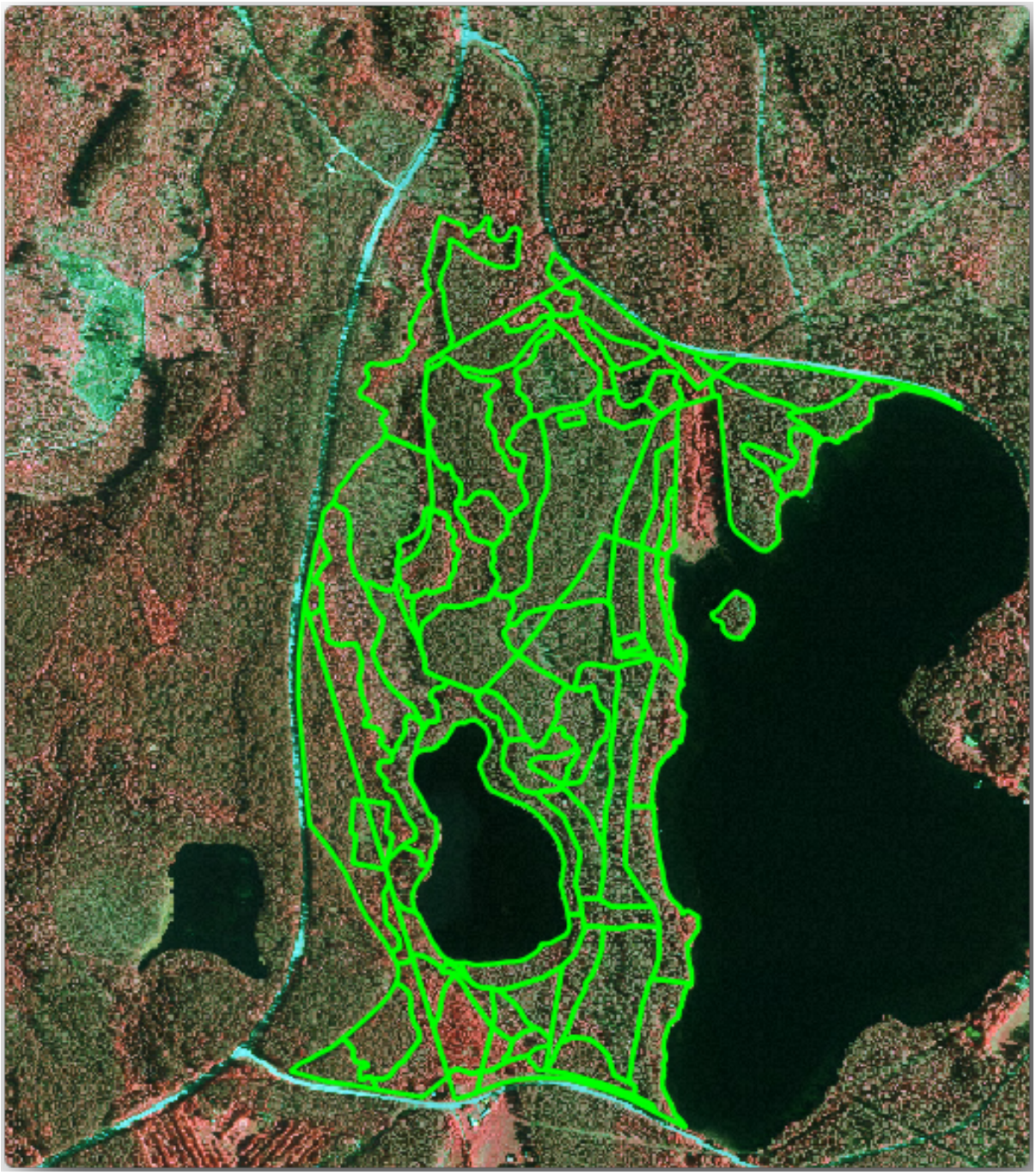
    20 年ほどが経過しており、また、様々な森林施業（伐採、間伐...）が行われたため、これは通常の状況です。また、1992 年当時にデジタイズしたときには林分が均質に見えていたが、時間が経過するにつれ、いくつかの森は異なる形で成長した可能性もあります。あるいは当時の森林目録の優先順位が現在のそれとは異なっていた可能性もあります。

次は、この画像に対する新しい林分を古い林分を使用せずに作成します。その後、違いを見るためにそれらを比較できます。

14.4.2 ★☆☆ （初級レベル） CIR 画像を解釈する

道路や湖によって制限される、古い調査でカバーされていた同じ地域を、デジタイズしてみましょう。前の演習のようにすでに林分のほとんどが含まれているベクタファイルを使って始められるので、地域全体をデジタイズする必要はありません。

1. Remove the layer forest_stands_1994
2. プロジェクトにファイル exercise_data\forestry\forest_stands_2012.shp を追加する
3. このレイヤのスタイルを、ポリゴンには塗りつぶしなしで、境界が表示されるように設定します
    1. forest_stands_2012 レイヤの **プロパティ** ダイアログを開く
    2. **シンボロジ** タブに移動する
    3. **塗りつぶし色** を透明な塗りつぶしに設定する
    4. **ストローク色** を緑に設定する
    5. **ストローク幅** を 0.50 mm に設定する

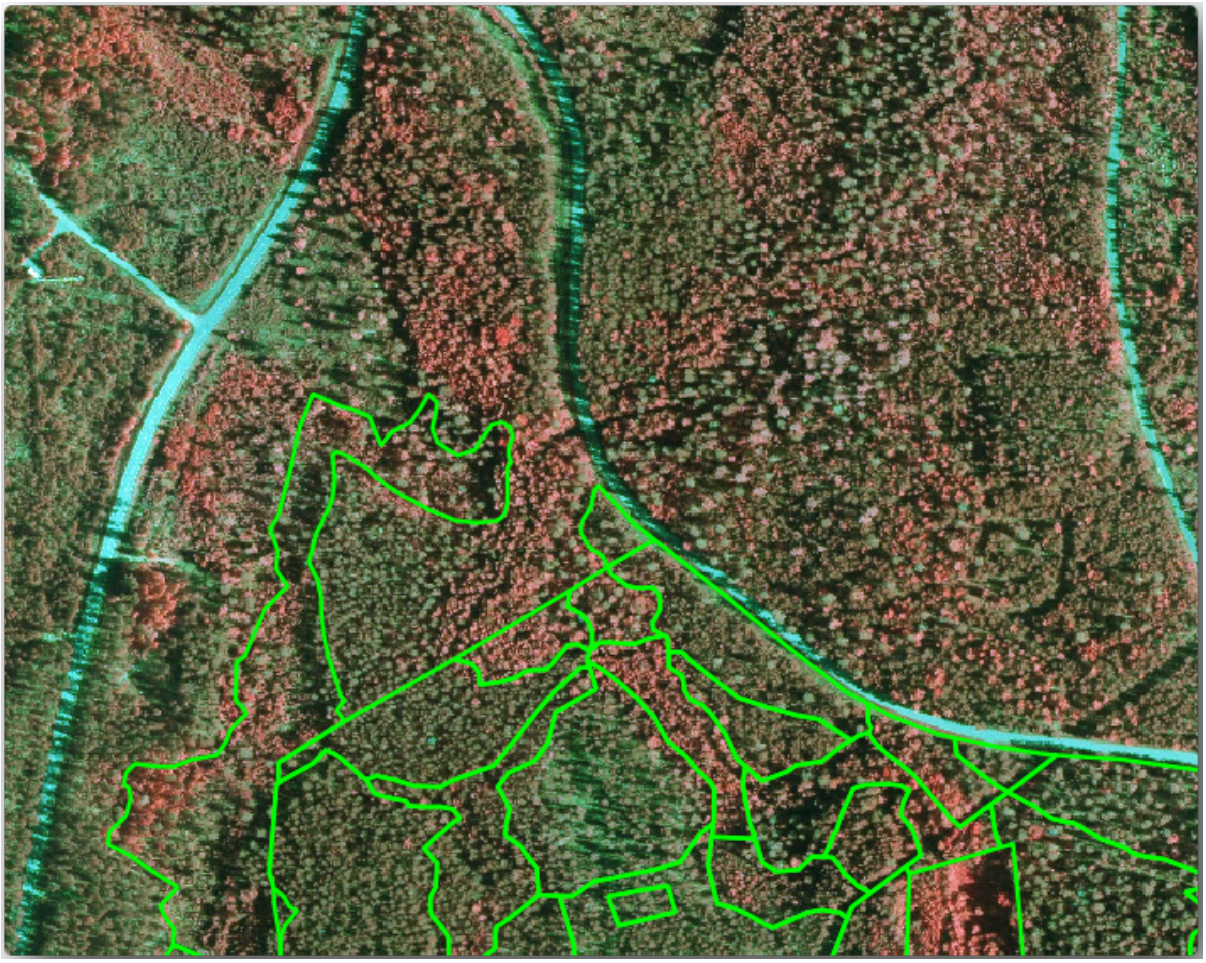


目録エリアの北側がまだ欠けていることがわかります。あなたの仕事は、この欠けている林分をデジタイズすることです。

始める前に、すでにデジタイズした林分および画像内の対応する森林を見直すことにいくらか時間を使ってください。林分境界がどのように決定されるかについて理解してみてください。林業の知識をいくらか持っている場合それが役立ちます。

考慮すべきいくつかの点:

- どの林に落葉樹があり（フィンランドではほとんどがシラカバ林）、どの森に針葉樹があるのか（この地域ではマツやトウヒ）。CIR 画像では、通常、落葉樹は明るい赤色で、針葉樹は濃い緑色で表示されます。
- 森林の樹齢は？樹冠の大きさは画像で確認できます。
- 異なる林分の密度はどれくらいでしょう？最近間伐が行われた林分は、樹冠と樹冠の間に隙間があり、周囲の他の林分と容易に区別できるはずです。
- 青みを帯びた地域は不毛の地形、道路や市街地、成長を開始していない作物などを示しています
- 林分を特定しようとする場合、画像を拡大し過ぎないでください。この画像は、1:3 000 から 1:5 000 の縮尺で十分です。下の画像（縮尺 1:4000）を見てください:



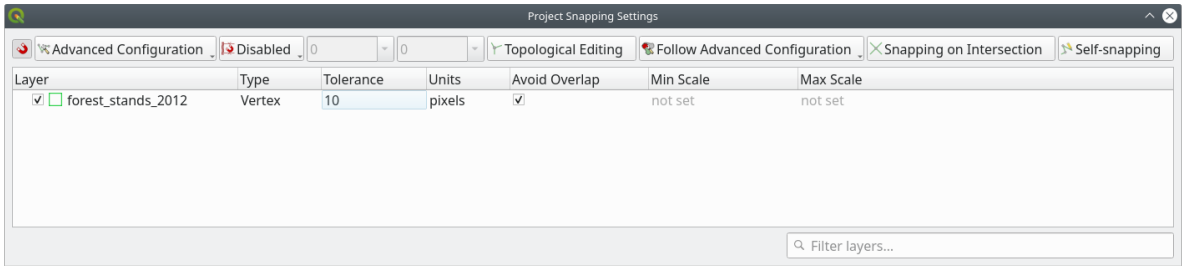
14.4.3 ★☆☆ （初級レベル） 自分でやってみよう: CIR 画像から林分をデジタイズする

林分をデジタイズするときは、樹種、林齢、林分密度... の観点から、可能な限り均一で取得しようとすべきです。しかし、詳細すぎてはいけません。さもないと、小さなたくさんの林分を作ることになり、それはまったく役に立たないでしょう。小さ過ぎず（少なくとも 0.5 ヘクタール）また大き過ぎない（せいぜい 3 ヘクタール）、林業の文脈において意味ある林分を取得しようとする必要があります。

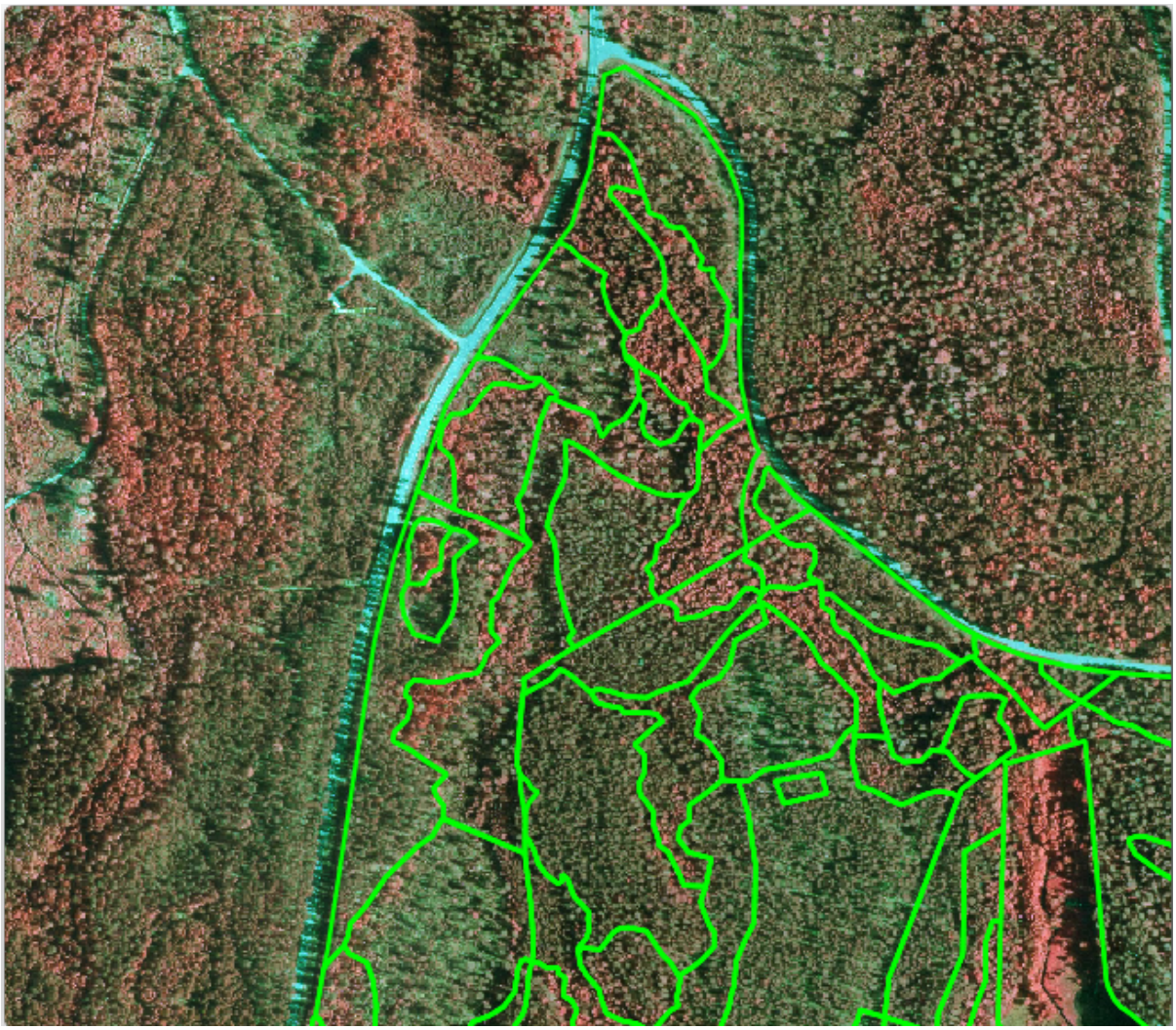
これらの点を念頭に、不足している林分をデジタイズします。

1. スナップとトポロジオプションを設定する:

1. **プロジェクト**　**スナップオプション**... に移動します
2.  スナップを有効にする を押し、**詳細設定** を選びます
3. Check the  `forest_stands_2012` layer
    1. **型** を :guilabel:`頂点 `に設定する
    2. **許容範囲** を `10` に設定する
    3. **単位** を **ピクセル** に設定する
    4. **重なりを避ける** の下のボックスをチェックする
    5.  **トポロジ編集** を押します
    6.  **詳細設定に従う** を選びます
    7. ポップアップを閉じます



2. Select the `forest_stands_2012` layer on the Layers list
3.  編集モード切り替え をクリックして編集を有効にします
4. 前のレッスンと同じ技法を使ってデジタイズを始めます。唯一の違いは、スナップするポイントレイヤがないことです。このエリアでは約 14 の新しい林分が得られるはずです。デジタイズしている間、`StandID` フィールドを `901` から始まる数字で埋めます。
5. 作業が完了したら、レイヤは次のようになります。



これで、CIR 画像から解釈された、2012 年のさまざまな森林を示すポリゴンの新しい集合ができあがりました。しかし、森林目録のデータがありません。そのため、森林を訪問し、各林分の森林属性を推定するために使用するサンプルデータを入手する必要があります。その方法は次のレッスンで説明します。

この区域で考慮する必要のある自然保護規制に関する追加情報を加えることもできます。

14.4.4 ★☆☆ （初級レベル） 理解しよう: 保全情報で林分を更新する

作業している区域には、森林計画を立てる際に考慮しなければならない保護規制がいくつかあります:

- エゾモモンガ（Pteromys の volans）の保護種の二箇所が同定されています。規定によれば、スポットの約 15 メートルの区域は手つかずのまま残さなければなりません。
- この区域の小川沿いに生育している特別に注目すべき河畔林を保護しなければなりません。現地を視察したところ、小川の両側 20 メートルを保護しなければならないことが判明しました。

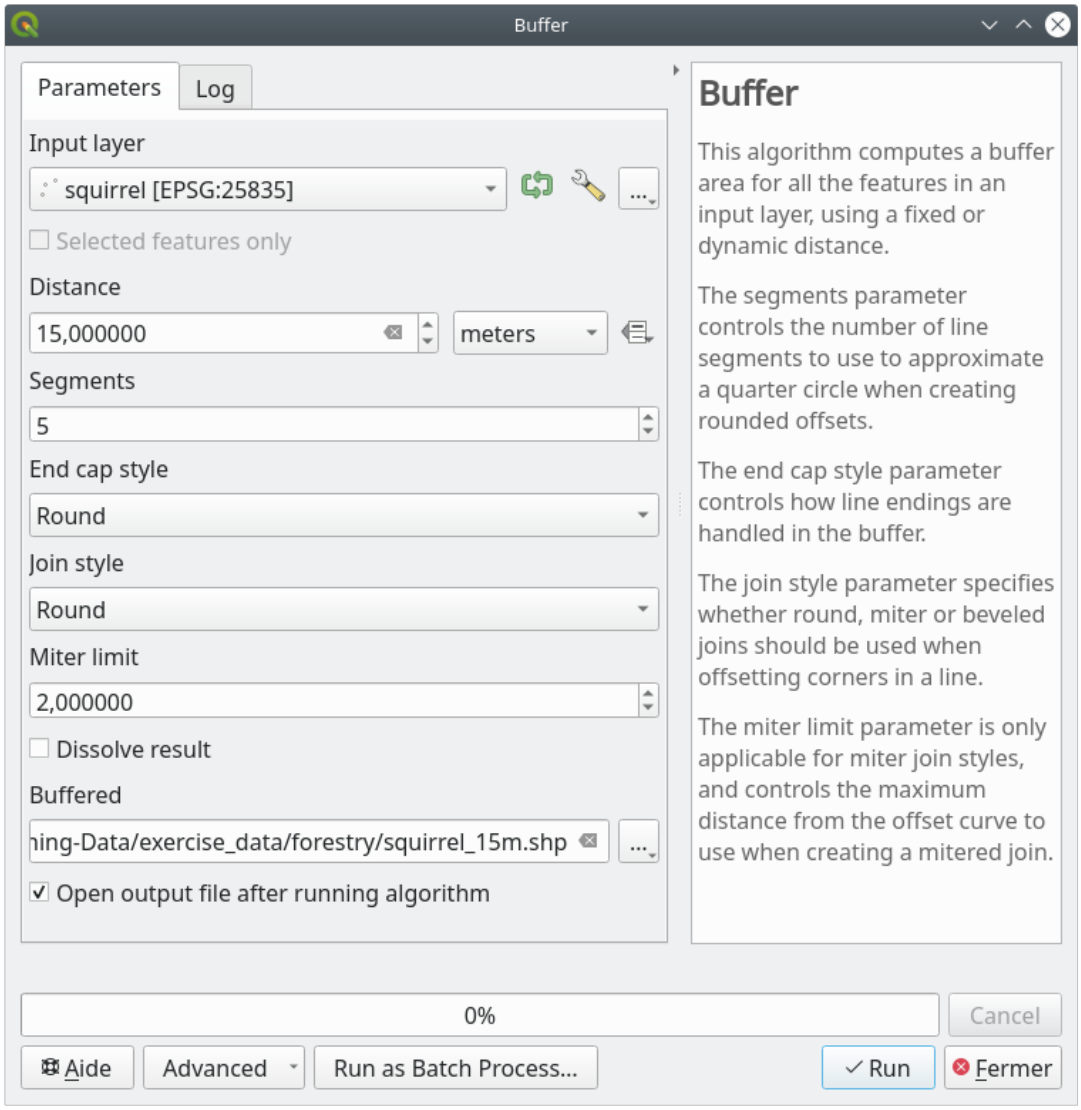
リスの指定区域に関する情報を含むベクタファイルと、北区域から湖に向かって流れるデジタイズした小川を含むベクタファイルがあります。

1. exercise_data\forestry\ フォルダから、 squirrel.shp と stream.shp ファイルをプロジェクトに追加する。
2.  属性テーブルを開く ツールを使って squirrel レイヤを表示する

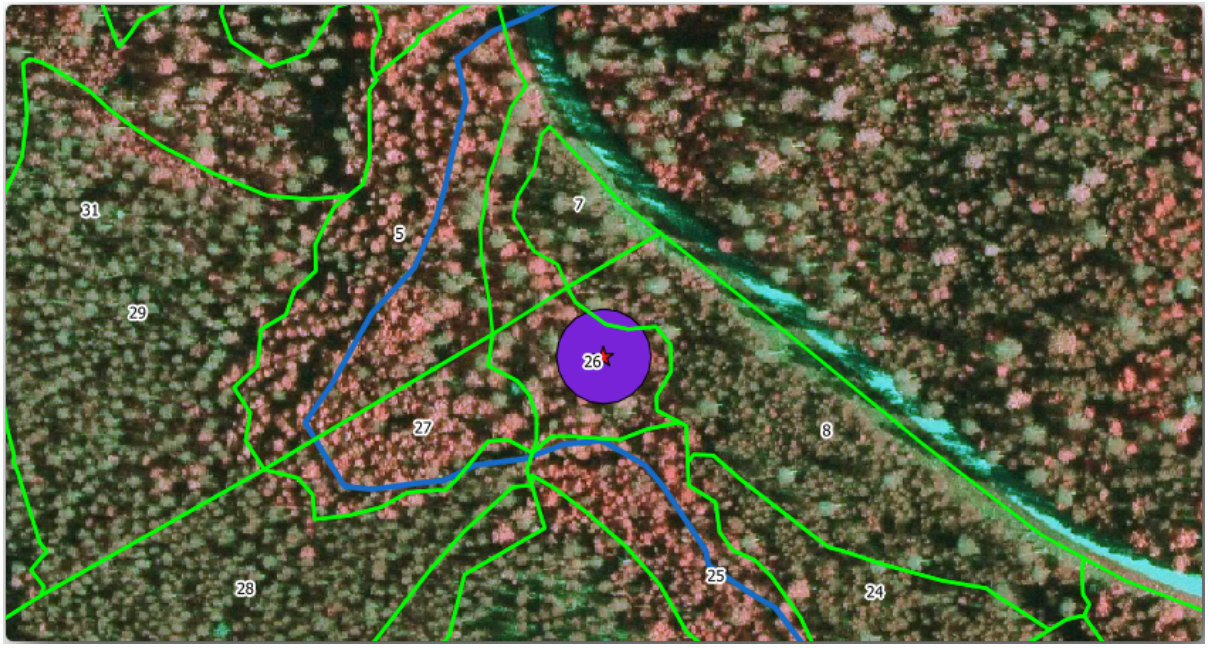
   エゾモモンガと定義されている 2 つの指定区域があること、及び保護すべき区域がその指定区域から 15 メートルの距離で示されていることがわかります。

保護する区域をより正確に区切りましょう。保護距離を使って、ポイント位置の周囲にバッファを作ります。

1. **ベクタ　空間演算ツール　バッファ** を開く。
2. Set *Input layer* to squirrel
3. **距離** に 15 メートル を設定する
4. **出力レイヤ** に exercise_data\forestry\squirrel_15m.shp を設定する
5. **アルゴリズムの終了後に出力ファイルを開く** をチェックする
6. **実行** をクリックします
7. プロセスが完了したら **閉じる** をクリックする



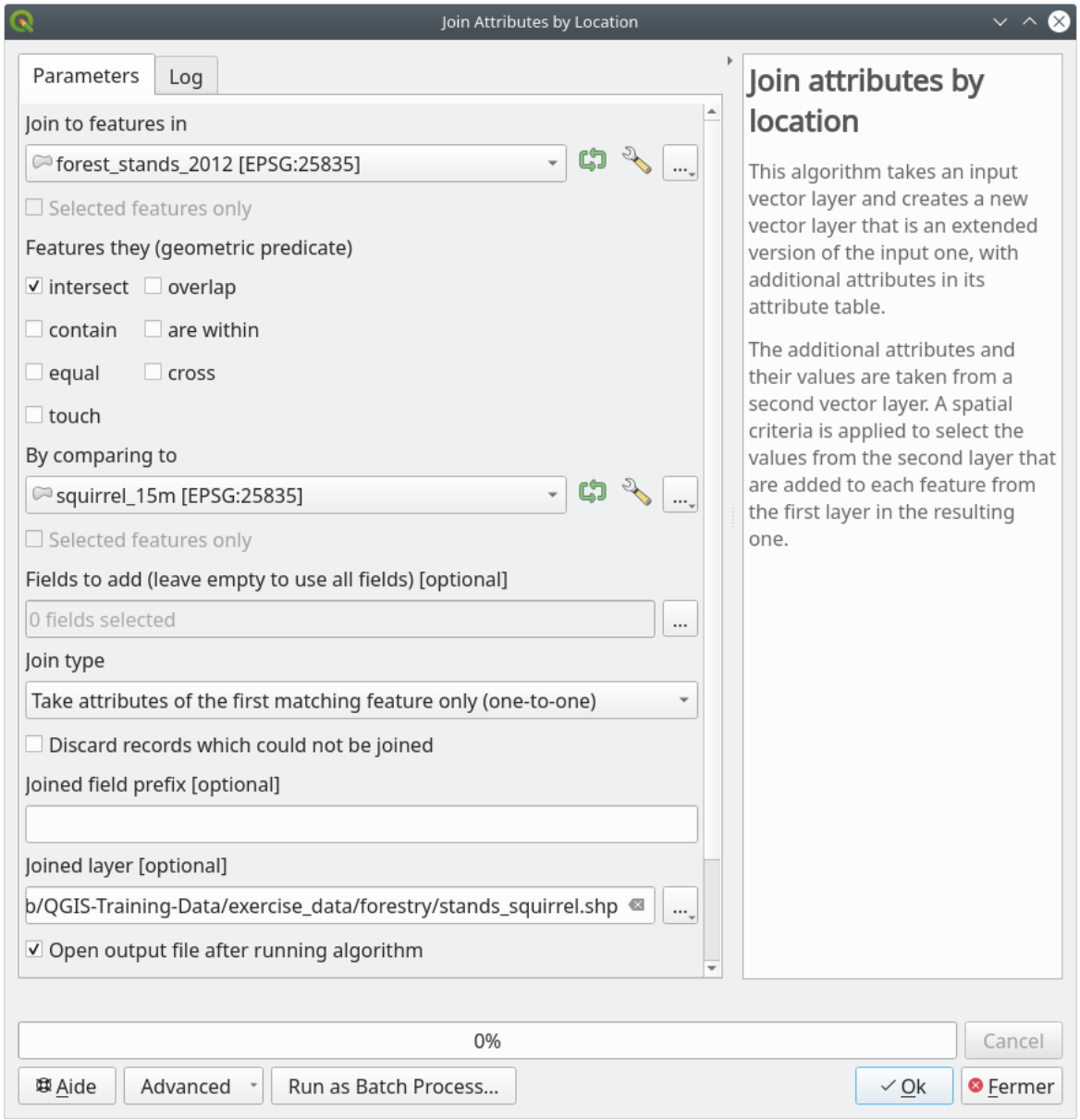
区域北部の指定区域を拡大すると、バッファ区域が隣接する 2 つの林分にまたがっていることがわかります。つまり、その林分で森林作業が行われるときは常に、保護指定区域も考慮に入れなければなりません。



リスの指定区域を保護するために、新しい林分に新しい属性（列）を追加し、保護されなければならない指定区域についての情報を格納します。この情報は森林作業が計画された時にいつでも利用でき、現場チームは作業開始前に手をつけてはいけない場所に印をつけることができます。

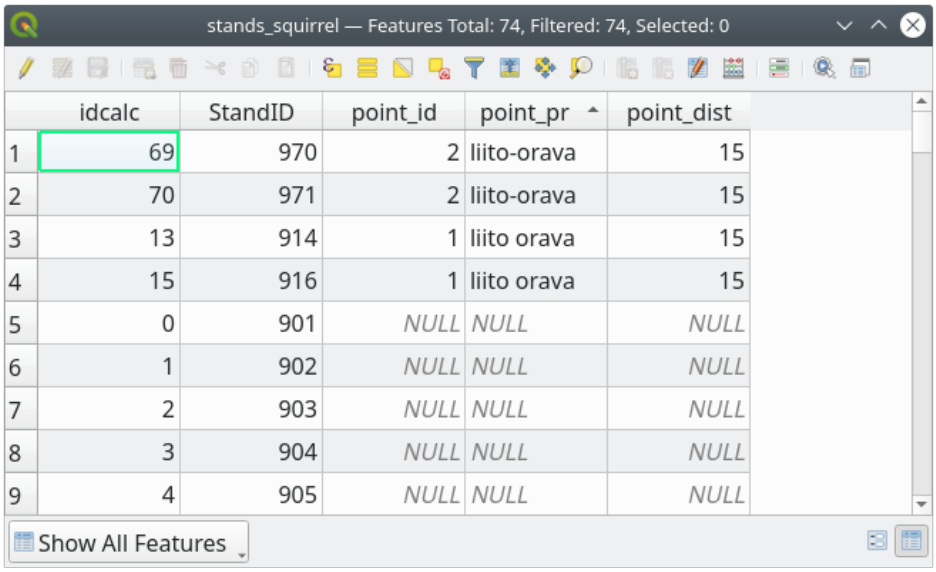
リスに関する情報を林分に結合するには、**属性の空間結合** アルゴリズムを使うことができます:

1. **ベクタ　データ管理ツール　属性の空間結合** を開く。
2. Set *Join to features in* to `forest_stands_2012`
3. **空間的関係** は **交差する** をチェックする
4. Set *By comparing to* to `squirrel_15m`
5. **結合型** に **最初に合致した地物の属性のみを取得（１対１結合）** を設定する
6. **結合対象がなかった地物を破棄** はチェックしないでおく
7. **出力レイヤ** に `exercise_data\forestry\stands_squirrel.shp` を設定する
8. **アルゴリズムの終了後に出力ファイルを開く** をチェックする
9. **実行** をクリックします
10. プロセスが完了したら、ダイアログを **閉じる** ことができます。



これでエゾモモンガの保護情報を表示する新しい林分レイヤ `stands_squirrel.shp` ができました。

1. `stands_squirrel` レイヤの属性テーブルを開く
2. テーブルヘッダの *point_pr* フィールドをクリックして、テーブルを並べ替える。



	idcalc	StandID	point_id	point_pr	point_dist
1	69	970	2	liito-orava	15
2	70	971	2	liito-orava	15
3	13	914	1	liito orava	15
4	15	916	1	liito orava	15
5	0	901	NULL	NULL	NULL
6	1	902	NULL	NULL	NULL
7	2	903	NULL	NULL	NULL
8	3	904	NULL	NULL	NULL
9	4	905	NULL	NULL	NULL

You can see that there are some forest stands that have the information about the protection locations. The information in the forest stands data will indicate to the forest manager that there are protection considerations to be taken into account. Then he or she can get the location from the `squirrel` dataset, and visit the area to mark the corresponding buffer around the location so that the operators in the field can avoid disturbing the squirrels environment.

14.4.5 ★☆☆ （初級レベル） 自分でやってみよう: 流域への距離で林分を更新する

保護されたリスの指定区域と同じアプローチで、林分に小川に関連する保護情報を更新することができます。いくつかのポイントがあります：

- バッファが小川の周囲 `20` メートルであることを覚えておく
- すべての保護情報を同じベクタファイルにしたいので、 `stands_squirrel.shp` をベースレイヤとして使用する。
- 出力の名前を `forest_stands_2012_protect.shp` とする

プロセスが完了したら、出力レイヤの属性テーブルを開き、小川に関連する河畔林の保護情報がすべて揃っていることを確認する。

結果に満足したら、QGIS プロジェクトを保存する。

14.4.6 結論

林分をデジタイズする CIR 画像をどのように解釈するかを見てきました。もちろんより正確な林分を作るためにはいくらか練習がいるでしょうし、通常は土壌地図のような他の情報を使用すればより良い結果が得られるでしょうが、これでこの種の業務のための基礎はわかりました。そして、他のデータセットからの情報を追加することで、非常に簡単な作業となりました。

14.4.7 次は?

デジタイズした林分は将来的に森林施業を計画するために使用されるでしょうが、まだ森林に関する詳細な情報を取得する必要があります。次のレッスンでは、調査にだけのデジタイズされた森林面積をサンプリングプロットのセットを計画する方法を見て、森林のパラメーターの全体的な見積もりを取得します。

14.5 レッスン: 体系的なサンプリングの設計

林分を表すポリゴンの集合は既にデジタイズしましたが、まだ森林についての情報を持っていません。その目的のために、この森林区域全体の目録作成の調査を設計し、そのパラメーターを推定できます。このレッスンでは、サンプリングプロットの体系的なセットを作成します。

森林目録を計画し始めるときは、目的、使用されるサンプルプロットの種類、目的を達成するために収集されるデータの種類を明確に定義することが重要です。それらは、個々の場合について森林管理目的のタイプに依存するでしょうし、誰か林業の知識を持つ人によって慎重に計画されなければなりません。このレッスンでは、体系的なサンプリングプロットの設計に基づいた理論的な目録を実装します。

このレッスンの目標： 森林区域を調査するために体系的なサンプリングプロットの設計を作成します。

14.5.1 森林の目録を作成する

森林の目録を作成するには、それぞれ異なる目的や条件に合わせて、いくつかの方法があります。たとえば、森林の目録を作成する 1 つの非常に正確な方法は（樹種だけを考慮する場合）、森林を訪問し、すべての木とその特徴のリストを作成することです。ご想像できるように、これはいくつかの小さな領域または一部の特殊な状況を除けば、一般的には適用できません。

森林について知る最も一般的な方法は、森林をサンプリングすることです。つまり、森林のさまざまな場所で計測を行い、その情報を森林全体に一般化することです。これらの計測は、多くの場合、簡単に計測できる小さな森林区域である **サンプルプロット** で行われます。サンプルプロットは、どんなサイズ（たとえば、50 m2、0.5 ha）や形式（たとえば、円形、長方形、可変サイズ）にもでき、方法（たとえば、ランダム、体系的、線に沿って）もさまざまです。サンプルプロットのサイズ、形式、および場所は、通常、統計的、経済的、および実用的な考慮事項に従って決定されます。林業の知識がない方は このウィキペディアの記事 を読むことをお勧めします。

14.5.2 ★☆☆（初級レベル）理解しよう: 体系的サンプリングプロット設計を実装する

作業している森について管理者は、この森のためには体系的なサンプリング設計が最も適切であると判断し、またサンプルプロットとサンプリングラインとの間に 80 メートルの一定の距離が信頼性の高い結果が得られると判断しました（この場合では 68 %の確率で平均誤差+-5 %）。可変サイズのプロットは、成長および成熟林分のため、この目録のための最も効果的な方法であると判断されているが、苗林分については 4 メートル固定の半径のプロットが使用されるでしょう。

実際には、後で野外チームによって使用されるポイントとしてサンプルプロットを表現する必要があるだけです：

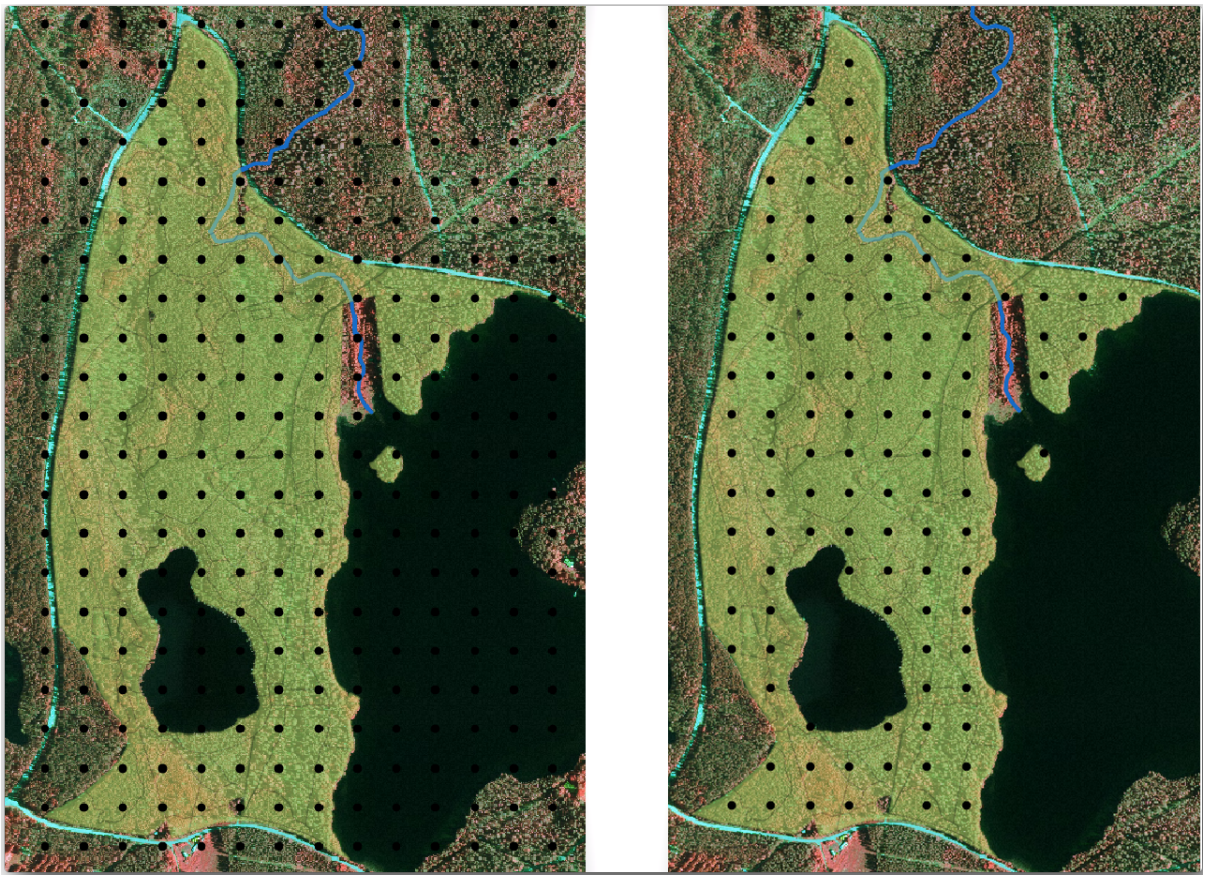
1. 前のレッスンからの `digitizing_2012.qgs` プロジェクトを QGIS で開きます。
2. Remove all the layers except `forest_stands_2012`.
3. ここでプロジェクトを `forest_inventory.qgs` として保存してください

今、お互いに 80 メートル離れたポイントの長方形のグリッドを作成する必要があります:

1. **ベクタ　　調査ツール**  **規則的な点群** を開きます。
2. Press the drop-down button next to the *Input extent* field and from the *Calculate from Layer* menu, select `forest_stands_2012`.
3. **点の間隔/数** の設定に、`80` メートル を入力します。
4. **点の間隔を使う** ボックスをチェックすると、この値がポイント間の距離を表していることを示します。
5. **規則的点群** で、出力を `forestry\sampling\' フォルダの :file:`systematic_plots.shp` として保存します。
6. **アルゴリズムの終了後に出力ファイルを開く** をチェックします。
7. **実行** を押します。

> **注釈**
>
> 提案された **規則的な点** は、選択されたポリゴン レイヤの範囲の左上隅から始まる体系的なポイントを作成します。この規則的な点にランダム性を追加したい場合は、0 から 80 (80 は点間の距離) の間でランダムに計算された数値を使用し、それからツールのダイアログで **隅からの初期嵌め込み（*LH* 側）** パラメータとして記述します。

このツールは、ポイントの長方形のグリッドを生成するために、林分レイヤの全体の範囲を使用していることに気づきます。しかし、お使いの森林区域の内側に実際にある点のみに関心があります（下の画像を参照）：



1. プロセシングツールボックスから GDAL *GDAL* -> **ベクタ・ジオプロセシング** -> **マスクレイヤで切り抜く** を開きます。
2. Select `systematic_plots` as the *Input layer*.
3. Set `forest_stands_2012` as the *Mask layer*.
4. **切り抜く（マスク）** 結果を `forestry\sampling\` フォルダに `systematic_plots_clip.shp` として保存します。
5. **アルゴリズムの終了後に出力ファイルを開く** をチェックします。
6. **実行** を押します。

これで、野外チームが設計されたサンプルプロットの場所に移動するために使用するポイントができました。野外作業のためにより便利になるように、これらのポイントをさらに準備できます。少なくとも、ポイントのために意味のある名前を追加し、GPS デバイスで使用できる形式にエクスポートする必要はあるでしょう。

まずは、サンプルプロットの命名から始めましょう。森林区域の内側のプロットの **属性テーブル** を確認すると、 **規則的点群** ツールで自動的に生成されたデフォルトの *id* フィールドがあることがわかります。ポイントにラベルを付けて地図上で確認し、サンプルプロットの命名の一部としてこれらの番号を使用できるかどうかを検討します:

1. `systematic_plots_clip` レイヤの **レイヤプロパティ** **ラベル** を開きます。
2. トップメニューを **単一定義** に変えます。
3. **値** 項目で `id` 属性を選びます。

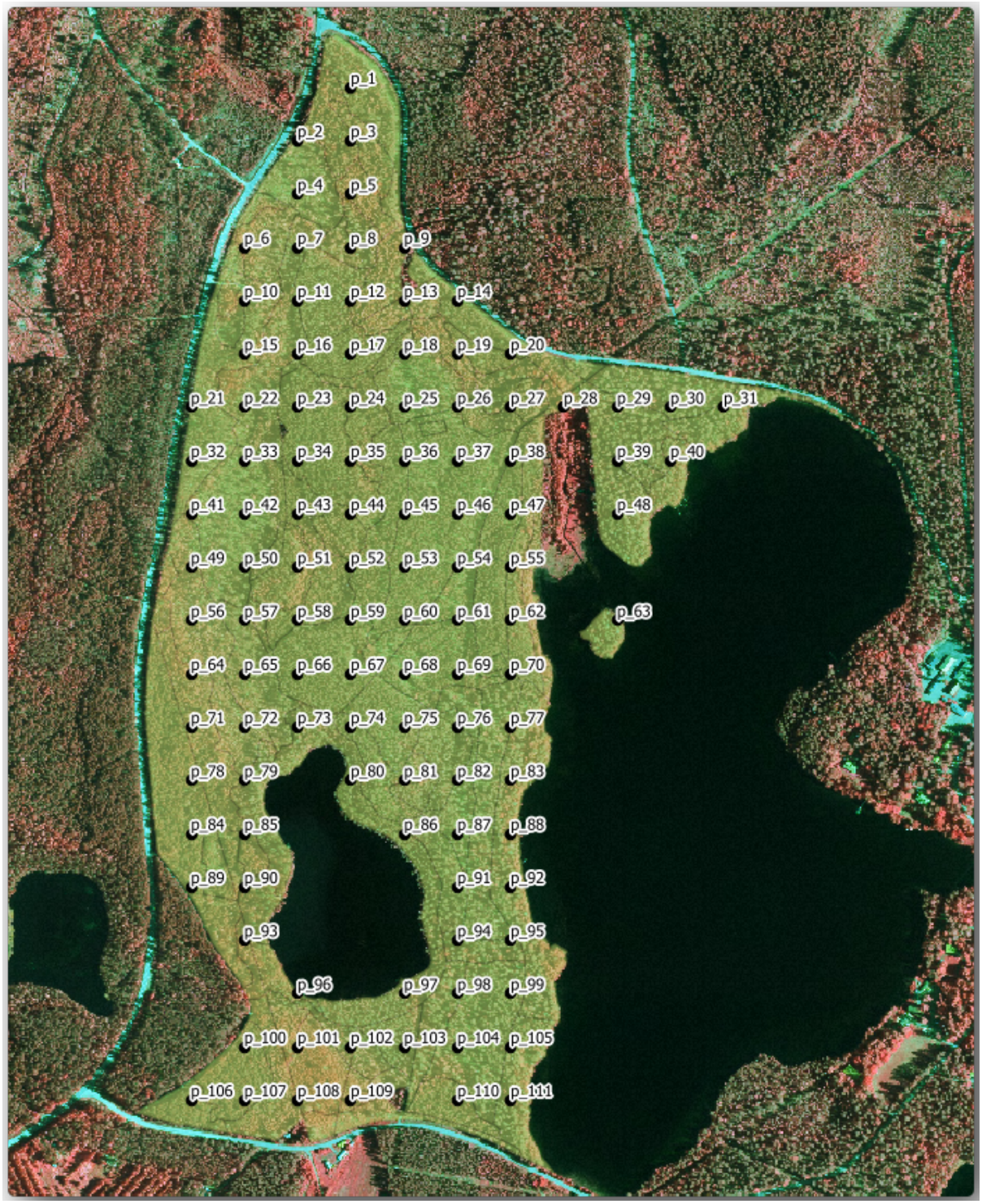
4. **バッファ** タブを選び、**テキストバッファを描画** をチェックし、バッファの **大きさ** を 1 にします。
5. *OK* をクリックします。

ここで地図上のラベルを見てください。ポイントが作成され、最初に西から東、それから北から南へと番号付けられていることがわかります。再び属性テーブルを見れば、テーブル内の順序もそのパターンに従っていることがわかります。異なる方法でサンプルプロットに名前付けする理由がない限り、名前を西東/南北様式で名前付けすることは論理的な順序に従っており、良いオプションです。

それにしても、`id` フィールドの数値はあまりよくありません。`p_1`、`p_2...` のような命名が良いでしょう。`systematic_plots_clip` レイヤに新しい列を作成することができます:

1. `systematic_plots_clip` の **属性テーブル** に移動します。
2.  編集モードを有効にします。
3.  フィールド計算機 を開きます:
    1. **新規フィールドを作成** をチェックします
    2. **出力する属性（フィールド）の名前** に `Plot_id` と入力します
    3. **フィールド型** に **テキスト** *(string)* を設定します。
    4. **式** フィールドに、この式 `concat('P_', @row_number )` を書くか、コピーするか、構成します。**関数リスト** の中の要素をダブルクリックすることもできることを思い出してください。`concat` 関数は :guilabel:`文字列 (String)`の下に、``@row_number``は :guilabel:`変数と値`グループの下にあります。
4. *OK* をクリックします。
5. 編集モードを無効にし、変更を保存します。

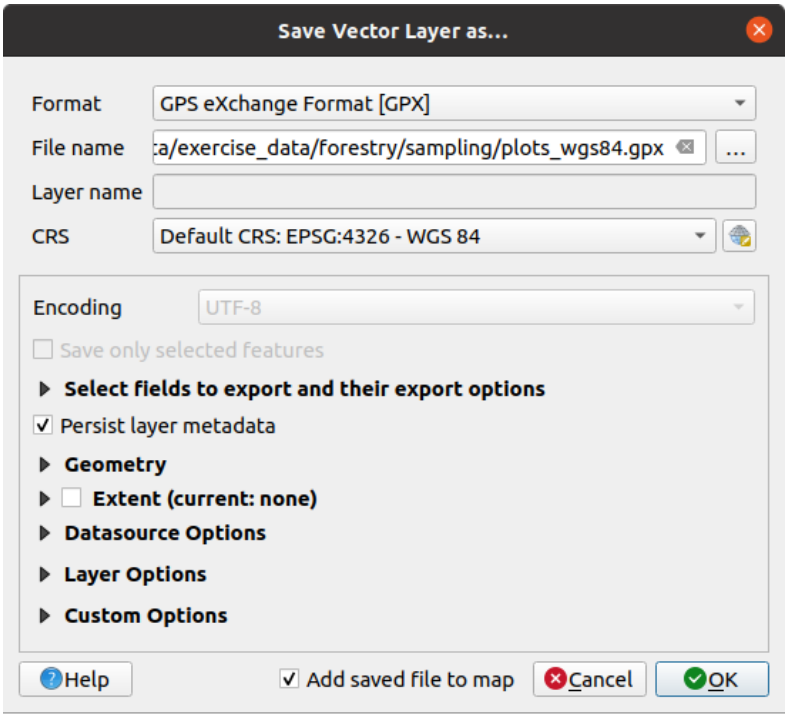
これで、あなたにとって意味のあるプロット名を持つ新しい列ができました。`systematic_plots_clip` レイヤで、ラベル付けに使用しているフィールドを新しい `Plot_id` フィールドに変更します。


p_1
p_2 p_3
p_4 p_5
p_6 p_7 p_8 p_9
p_10 p_11 p_12 p_13 p_14
p_15 p_16 p_17 p_18 p_19 p_20
p_21 p_22 p_23 p_24 p_25 p_26 p_27 p_28 p_29 p_30 p_31
p_32 p_33 p_34 p_35 p_36 p_37 p_38 p_39 p_40
p_41 p_42 p_43 p_44 p_45 p_46 p_47 p_48
p_49 p_50 p_51 p_52 p_53 p_54 p_55
p_56 p_57 p_58 p_59 p_60 p_61 p_62 p_63
p_64 p_65 p_66 p_67 p_68 p_69 p_70
p_71 p_72 p_73 p_74 p_75 p_76 p_77
p_78 p_79 p_80 p_81 p_82 p_83
p_84 p_85 p_86 p_87 p_88
p_89 p_90 p_91 p_92
p_93 p_94 p_95
p_96 p_97 p_98 p_99
p_100 p_101 p_102 p_103 p_104 p_105
p_106 p_107 p_108 p_109 p_110 p_111

14.5.3 ★☆☆ （初級レベル） 理解しよう: GPX 形式としてサンプルプロットを書き出す

野外チームは、おそらく GPS デバイスを使用して、あなたが計画したサンプルプロットの位置を特定することになるでしょう。次のステップは、作成したポイントを GPS が読み取れる形式にエクスポートすることです。QGIS では、点と線のベクタデータを GPS eXchange Format (GPX) で保存することができます。これは、ほとんどの専門ソフトウェアで読むことができる標準の GPS データ形式です。データを保存する際に CRS を選択するのには注意が必要です:

1. `systematic_plots_clip` レイヤを右クリックし、**エクスポート　新規ファイルに地物を保存...** を選びます。



2. **形式** で *GPS* **交換フォーマット** *[GPX]* を選択します。
3. 出力を `forestry\sampling\` フォルダに `plots_wgs84.gpx` という **ファイル名** で保存します。
4. *CRS* 中で **選択された** *CRS* を選択します。
5. *EPSG:4326 - WGS 84* をブラウズします。

> **注釈**
>
> GPX 形式はこの CRS しか受け付けません。違うものを選ぶと、QGIS はエラーを返しませんが、空のファイルができます。

6. *OK* をクリックします。
7. 開いたダイアログで、`waypoints` レイヤだけを選びます（レイヤの残りは空です）。

目録サンプルプロットは、ほとんどの GPS ソフトウェアで管理できる標準形式になっています。野外チームは、サンプルプロットの位置をデバイスにアップロードできるようになりました。そのためには、各自のデバイスのソフトウェアと、保存した `plots_wgs84.gpx` ファイルを使用することになります。他の方法としては、*GPS Tools* プラグインを使用することもできますが、その場合、特定の GPS デバイスで動作するようにツールを設定する必要があります。もしあなたが自分のデータで作業をしていて、このツールがどのように動作するかを確認したい場合は、**QGIS ユーザーマニュアル** の working_gps のセクションで情報を得ることができます。

ここで QGIS プロジェクトを保存します。

14.5.4 結論

森林目録に使用する体系的なサンプリングの設計を作成する方法を簡単に見たところです。サンプリング設計の他のタイプを作成するには、QGIS 内のさまざまなツールやスプレッドシートを使用したりスクリプトを書いてサンプルプロットの座標を計算することが入ってきますが、一般的な考え方は同じままになります。

14.5.5 次は?

次のレッスンでは、野外チームが担当のサンプルプロットに到達するため使用している詳細地図を自動で作成するために、QGIS の地図帳機能をどう使用するかを見ていきましょう。

14.6 レッスン: 地図帳ツールで詳細な地図を作る

体系的なサンプリングの設計が準備でき、野外チームがナビゲーションデバイスに GPS 座標をロードしました。彼らはまた、すべてのサンプルプロットで測定された情報を収集するための野外データフォームを持っています。野外チームは、すべてのサンプルプロットへ行く道が簡単に見つかるよう、いくつかの地上情報がサンプルプロットのより小さいサブセットや地図エリアに関する情報と一緒にはっきり見える大量の詳細地図を要求してきました。地図帳ツールを使用することで、大量の地図を共通のフォーマットで自動的に作成できます。

このレッスンの目標： QGIS で地図帳ツールを使用して、野外目録作業を支援するための詳細な印刷可能な地図を作成することを学びます。

14.6.1 ★☆☆ （初級レベル） 理解しよう: 印刷レイアウトを準備する

森林地域と私たちのサンプリングプロットの詳細な地図を自動化するには、まず野外作業のために有用と思われる要素がすべて入った地図テンプレートを作成する必要があります。もちろん、最も重要なのは、前に見てきたように、スタイルを適切に設定することですが、印刷された地図を完成する他の多くの要素を追加することも必要です。

1. 前回のレッスンで作成した QGIS プロジェクト `forest_inventory.qgs` を開いてください。少なくとも以下のレイヤがあるはずです：
   - `forest_stands_2012` (with a 50% transparency, green fill and darker green border lines)
   - `systematic_plots_clip`
   - `rautjarvi_aerial`
2. プロジェクトを新しい名前 `map_creation.qgs` として保存します。

印刷可能な地図を作成するには、 *Layout Manager*: を使うことを思い出してください:

1. Open **プロジェクト**  **レイアウトマネージャ**... を開きます。
2. **レイアウトマネージャ** ダイアログで:
   1. **テンプレートから新規作成** の下の **空のレイアウト** の隣にある **作成**... ボタンを押します
   2. 印刷レイアウトに `forest_map` という名前を付けます。
   3. *OK* を押します。新しい印刷レイアウトが作られ、空白の用紙が開きます。
3. 印刷レイアウトウィンドウで、プロパティが A4 紙にセットされていることを確認します:
   1. 紙の上で右クリックし、**ページのプロパティ** を選択します。レイアウトの右に **ページプロパティ** パネルが開きます。
   2. **ページサイズ** が *A4* になっていることを確認します。
   3. **方向** が **横** になっていることを確認します。
4. **ページプロパティ** パネルの隣にある **レイアウト** タブを開き、**エクスポート解像度** を `300 dpi` にセットします。

キャンバスグリッドを使用してさまざまな要素を配置すると、地図の作成が簡単になります。レイアウトグリッドの設定を確認します：

1. **レイアウト** タブで **ガイドとグリッド** を開きます。
2. **グリッド間隔** が `10 mm`、**スナップ許容量** が `5 px` になっていることを確認します。

グリッドの使用を有効にする必要があります。

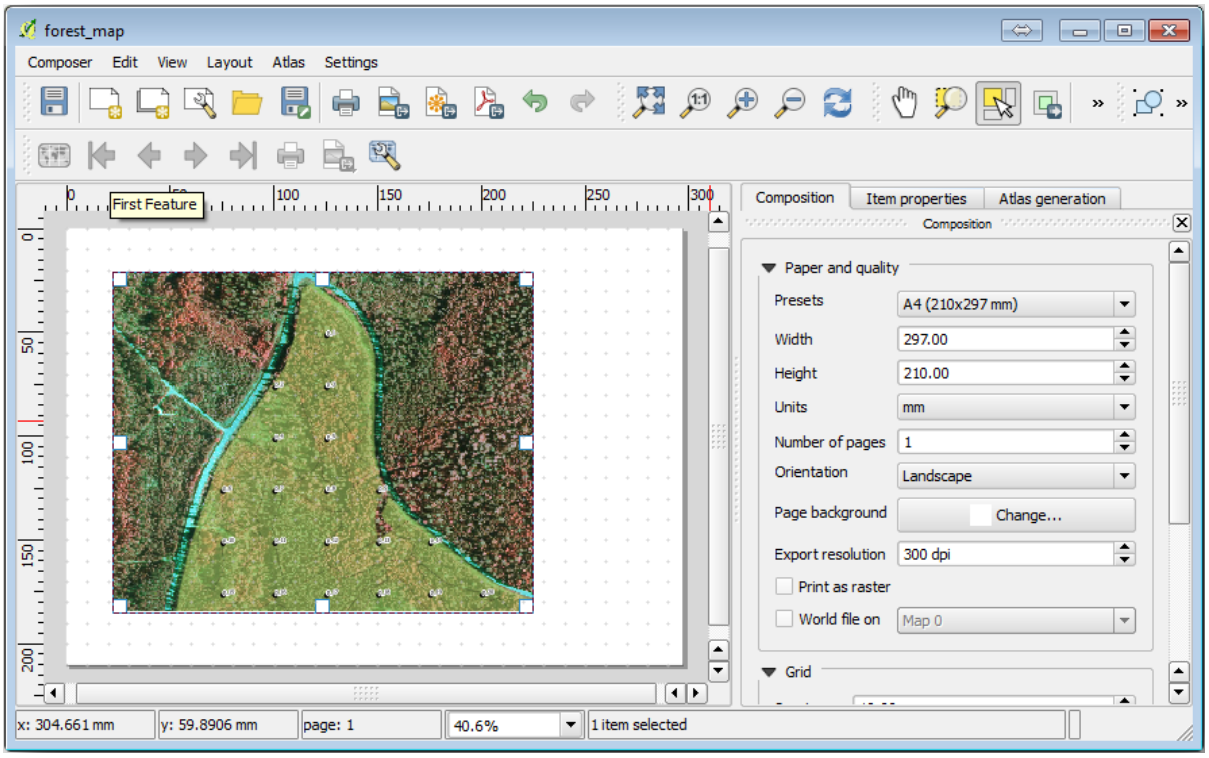
1. **ビュー** メニューを開きます。
2. **グリッドを表示** をチェックします。
3. **グリッドにスナップ** をチェックします。

4. **ガイド** を使うオプションが既定でチェックされていることに気づくでしょう。これによってレイアウト内で要素を動かす際にガイド線が現れるようになります。

5. これで、レイアウトへの要素の追加を開始できます。最初に地図要素を追加して、レイヤーのシンボルを変更すると地図キャンバスがどのように見えるかを確認できるようにします。

   1. 地図を追加 ボタンをクリックします。

   2. 地図がそれのほとんどを占めるようにキャンバス上でクリックして矩形にドラッグします。



マウスカーソルがキャンバスのグリッドにスナップされるに注意してください。他の要素を追加したときに、この機能を使用してください。より精度を高くしたい場合は、グリッド **間隔** 設定を変更してください。何らかの理由でいくつかの点でグリッドにスナップしたくない場合は、常に **ビュー** メニュー中でいつでもそのチェックを切り替えできます。

14.6.2 ★☆☆ （初級レベル） 理解しよう: 背景地図を追加する

レイアウトを開いたままマップに戻ります。地図の内容ができるだけ明確になるようにいくつかの背景データを加えてスタイルを作りましょう。

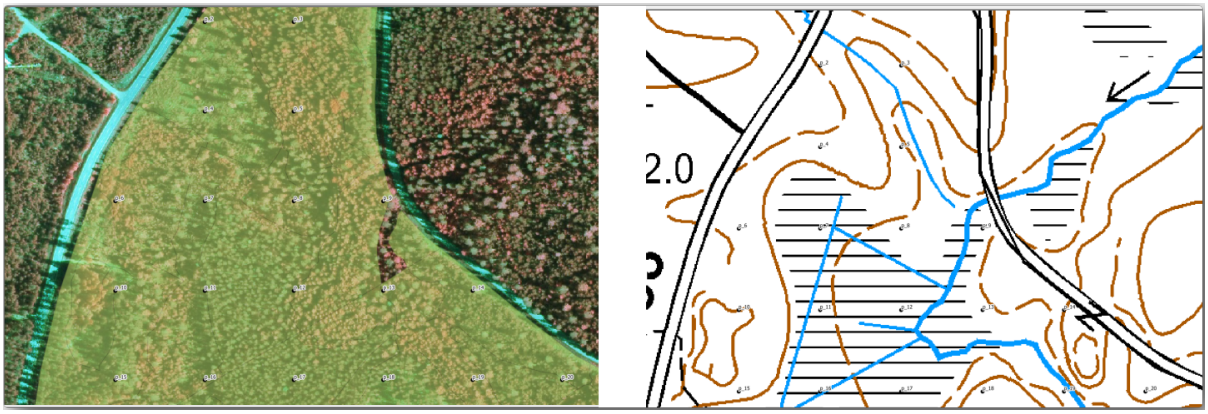
1. `exercise_data\forestry\` フォルダにある背景ラスタ `basic_map.tif` を追加します。

2. 入力を要求されたときは、このラスタに *ETRS89 / ETRT-TM35FIN* CRS を選択してください。

   ご覧のように背景地図にはすでにスタイルが付いています。このタイプの使用準備ができている地図作成ラスターは非常に一般的です。それは、ベクターデータから作成された標準形式でスタイルとラスタとして格納されていますので、いくつかのベクタレイヤにスタイル付けする手間や良い結果を得られているか心配する必要がありません。

3. 今、プロットの約 4 または 5 行を見ることができるようにサンプルプロットにズームします。

サンプルプロットの現在のスタイルは最良ではありません:



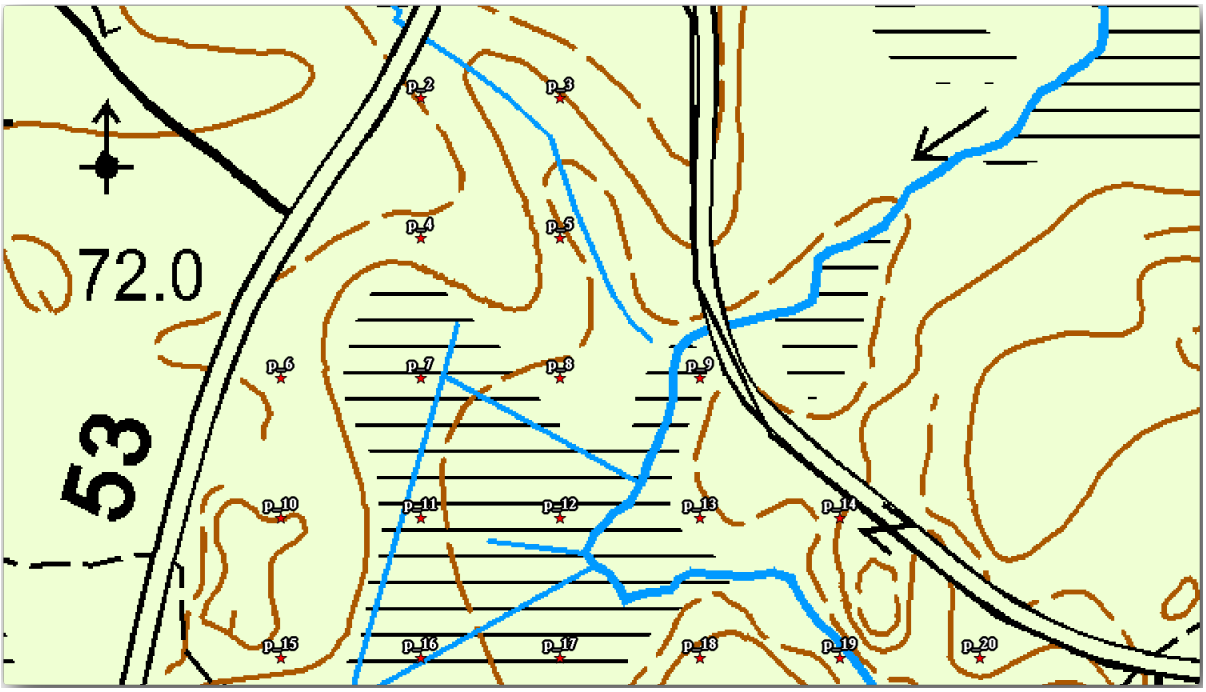
最後の練習では、この白いバッファは空中写真の上では OK でしたが、今は背景イメージがほとんど白くラベルがほぼ見えません。印刷レイアウトではどう見えるでしょう？確認してみましょう:

1. 印刷レイアウトウィンドウに移動します。
2.  Select/Move item ボタンを使ってレイアウトにあるマップ要素を選択します。
3. **アイテムプロパティ** タブに移動します。
4.  **キャンバスの範囲に地図の範囲を合わせる** をクリックします。
5. 要素をリフレッシュするときは  **地図のプレビューの更新** をクリックします。

明らかにこれは十分ではありません；野外チームのためにプロット番号をできるだけはっきりと見えるようにしたいと思います。

14.6.3 ★☆☆ （初級レベル） 自分でやってみよう: レイヤのシンボロジを変更する

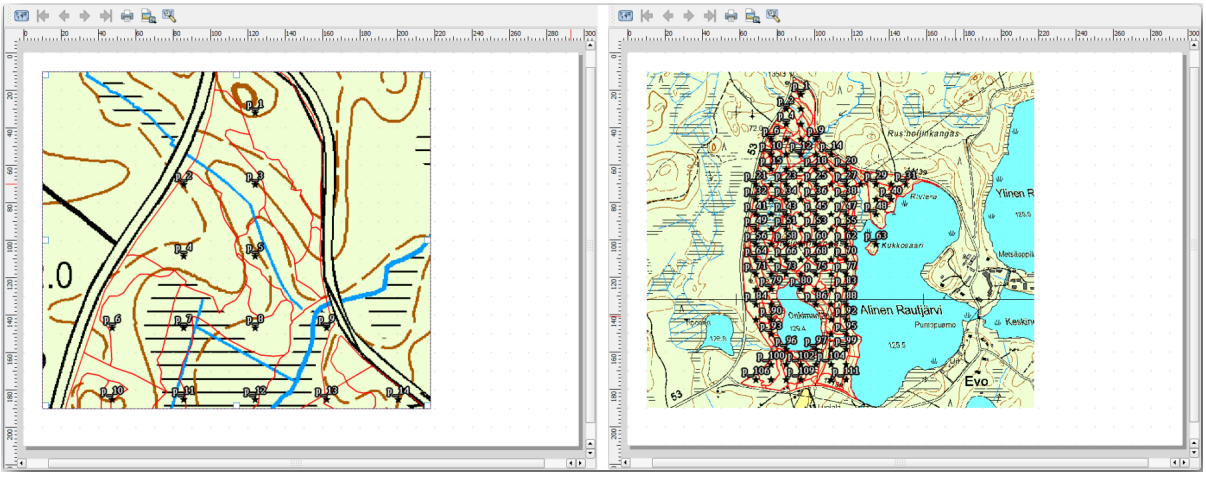
モジュール: 基本的な地図を作成して探索する ではシンボロジを、 **モジュール: ベクタデータを分類する** ではラベルを作業してきました。利用可能なオプションとツールのいくつかについて、記憶を呼び起こす必要があるときは、それらのモジュールに戻ってください。目標はプロットの位置と名前ができるかぎりはっきりと見えるように、しかし常に背景地図要素が見えるようにすることです。この画像からいくつかの指針を取ることができます:



後で `forest_stands_2012` レイヤの緑色のスタイリングを使うことになります。これを維持したまま、林分の境界線だけを表示するビジュアライゼーションにするためです:

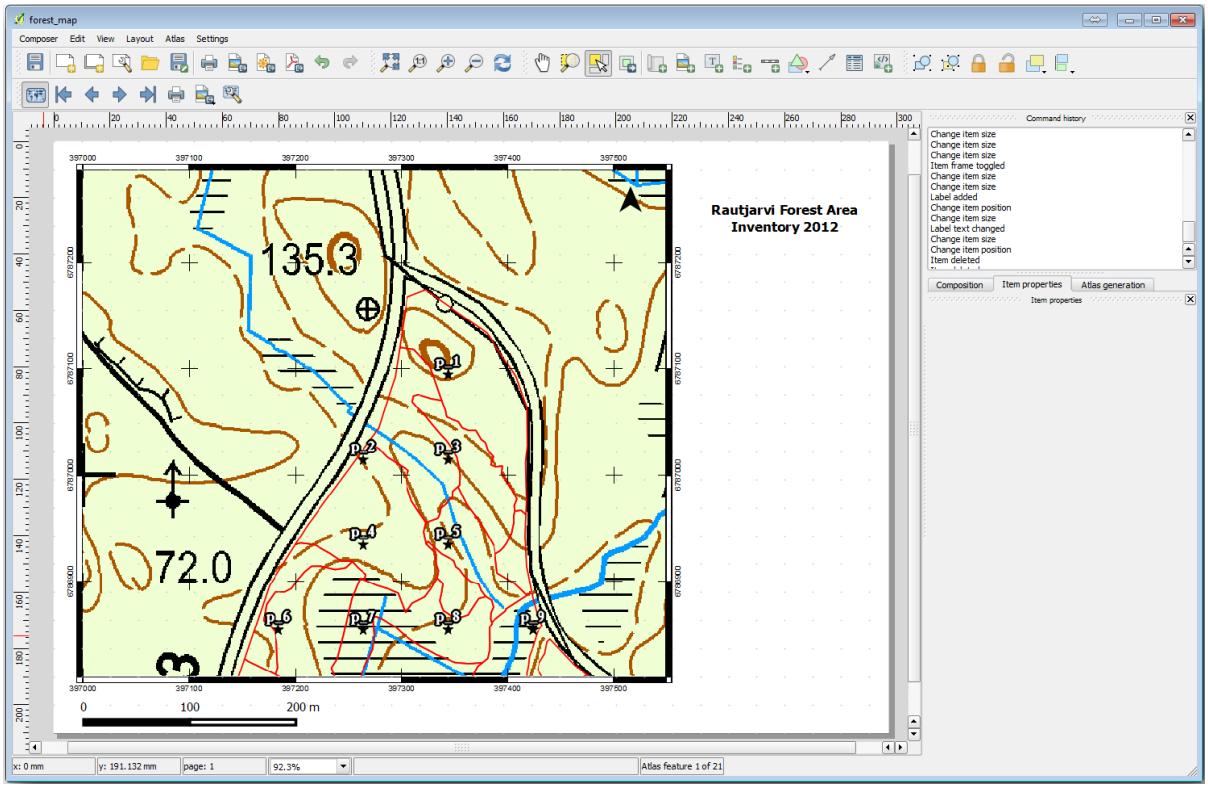
1. Right click on `forest_stands_2012` and select *Duplicate*
2. `forest_stands_2012 copy` という名前の新しいレイヤが作成され、これを使って異なるスタイル、例えば、塗りなし、赤い縁取り、を定義することができます。

   今、林分にはの二つの異なる視覚化があり、詳細地図にどちらを表示するか決定できます。
3. 印刷レイアウトのウィンドウに頻繁に戻って、地図がどのように見えるかを確認します。詳細な地図を作成する目的では、森林地帯全体の縮尺（下の左の画像）ではなく、より大きな縮尺（下の右の画像）できれいに見えるシンボロジを探しているのでしょう。地図のズームやレイアウトを変更するときは、  地図の preview の更新 と  **キャンバスの範囲に地図の範囲を合わせる** を忘れずに使ってください。



14.6.4 ★☆☆ （初級レベル） 自分でやってみよう: 基本地図テンプレートを作る

1. 満足のいくシンボロジができたら、印刷する地図にさらに情報を追加する準備が整いました。少なくとも次の要素を追加してください:
   - タイトル。
   - スケールバー。
   - 地図のグリッドフレーム。
   - グリッドの両側の座標。
2. **モジュール: 地図をレイアウトする** ですでに同様のレイアウトを作成しています。必要に応じて、そのモジュールに戻りましょう。参考にこのサンプル画像を見ることができます:



3. 地図を画像として書き出し、それを見てください。
   1. **レイアウト　画像としてエクスポート**...。
   2. *JPG format* の例を使います。

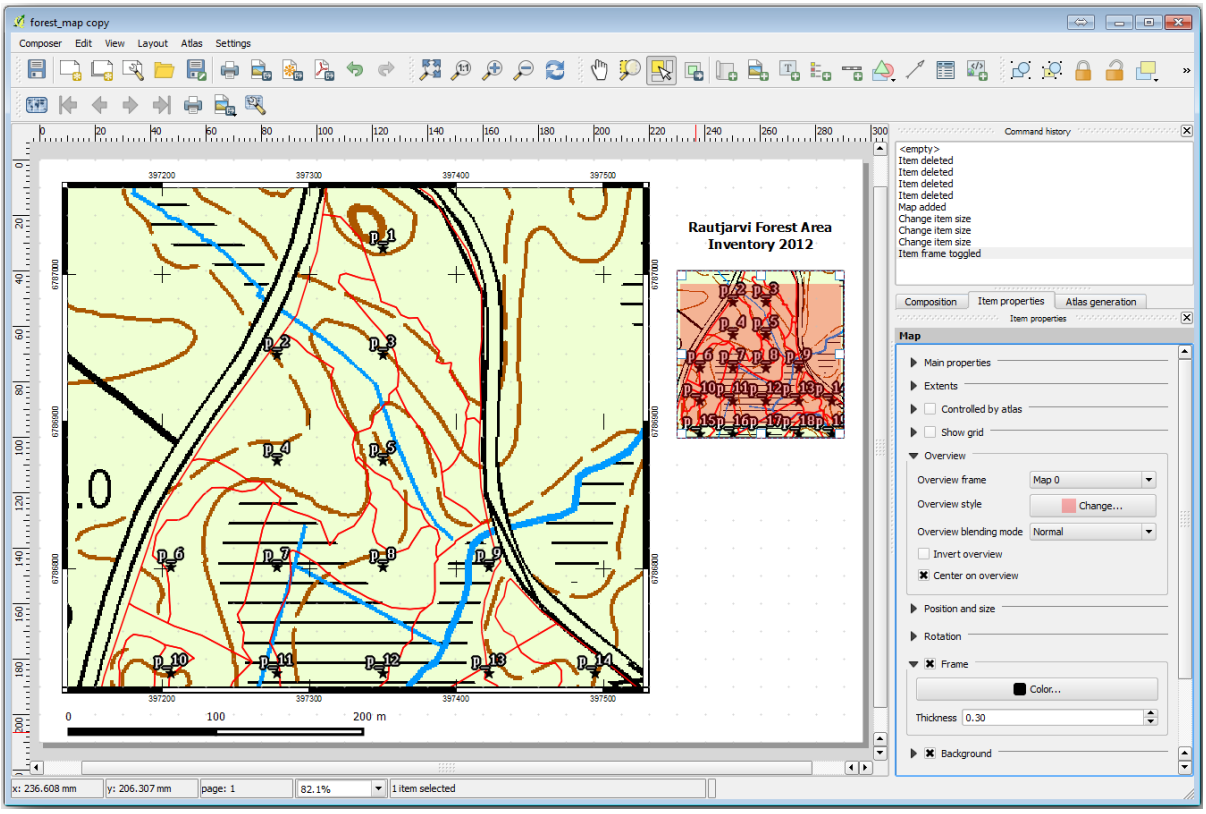
これは、印刷されたときに、それがどのように見えるかです。

14.6.5 ★☆☆ （初級レベル） 理解しよう: 印刷レイアウトにさらに要素を追加する

提案された地図テンプレート画像でおそらくお気づきのように、キャンバスの右側には余白がたくさんあります。他に何をそこに置けるか見てみましょう。この地図の目的に本来凡例は必要ありませんが、全体図といくつかのテキストボックスで、地図に価値を加えることができます。

全体図は、野外チームが、一般的な森林地域内の詳細図を置くのに役立ちます。

1. タイトルテキストのすぐ下で、キャンバスに別の地図要素を追加します。
2. **アイテムのプロパティ** タブで、 **全体図** ドロップダウンを開きます。
3. **概要フレーム** を地図 *Map 0* に設定してください。これは小さい地図上に大きい地図に表示された範囲を表す影の矩形を作成します。
4. **フレーム** オプションをチェックし、色を黒に **太さ** を `0.30` にします。



この全体図では本当に望んでいる森林地域の概観になっていないことに注意してください。やりたいことは、この地図が森林地域全体を表現し、それが背景地図と `forest_stands_2012` レイヤだけを表示して、サンプルプロットを表示しないことです。そして、もうレイヤの可視性や順序を変更しても変更されないよう、そのビューをロックすることです。

1. 地図に戻りますが、 **印刷レイアウト** は閉じないでください。
2. Right click the `forest_stands_2012` layer and click on *Zoom to Layer Extent*.
3. Deactivate all layers except for `basic_map` and `forest_stands_2012`.
4. **レイヤ** パネルの 地図テーマを管理 ツールを展開し、 **テーマを追加** を選択します。

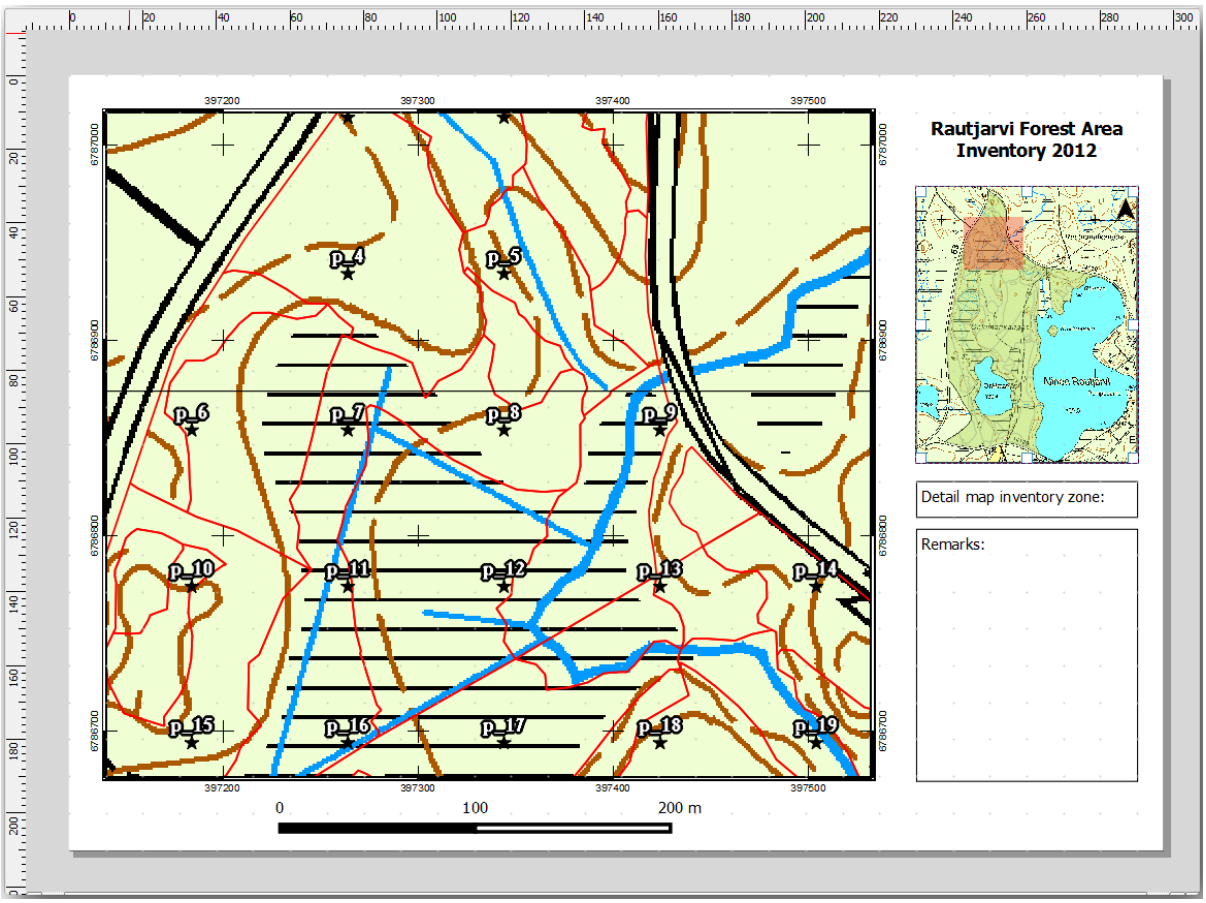
5. その名前を basic_overview にします。
6. 印刷レイアウトに戻ります。
7. 小さな地図を選択した状態で、**キャンバスの範囲に地図の範囲を合わせる** をクリックし、マップウィンドウに表示される範囲にその範囲を設定します。
8. **地図テーマに従う** をチェックし、**メインプロパティ** で basic_overview を選択して、全体図のビューをロックします。

これで、全体図は期待通りのものになり、そのビューはもう変わることはないでしょう。しかし、当然ながら、詳細地図には林分境界線もサンプルプロットも表示されなくなりました。これを修正しましょう：

1. 再びマップウィンドウに移動し、表示したいレイヤ（systematic_plots_clip、forest_stands_2012 copy そして Basic_map）を選択します。
2. サンプルプロットの数ラインだけ見えるように再び拡大します。
3. **印刷レイアウト** ウィンドウに戻ります。
4. レイアウトの大きい方の地図を選択します。
5. **アイテムプロパティ** の **ビューを更新** と **キャンバスの範囲に地図の範囲を合わせる** をクリックします。

大きな地図だけが現在のマップビューを表示し、小さな全体図はそれをロックしたときと同じ表示を保っていることに注意してください。

全体図は、詳細地図に示されている範囲の陰影付の枠を表示していることにも注意してください。



テンプレート地図はほぼ準備ができています。ここで 2 つのテキストボックスを地図の下に追加します、一方はテキスト「詳細地図ゾーン：」を、もう一方は「備考：」が入っています。それらが上の画像に見えるように配置してください。

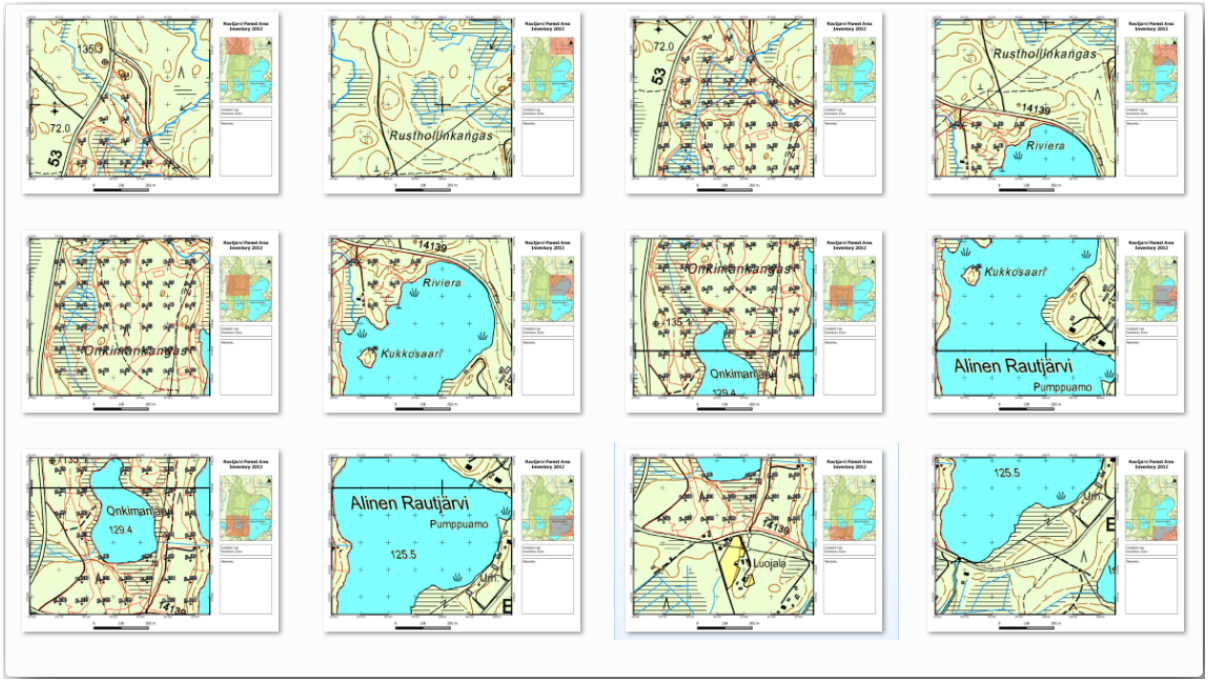
また、全体図に方位記号を追加できます。

1. 方位記号を追加 ツールを選びます。
2. 全体図の右上の角で矩形をクリック＆ドラッグします。
3. **アイテムプロパティ** で *SVG* **イメージ** をチェックします
4. *SVG* **ブラウザ**　*SVG* **グループ** を閲覧して矢印の画像を探します。
5. **画像の回転** の下にある **地図と同期する** をチェックし、*Map 1*（全体図）を選びます。
6. 矢印の画像のサイズを小さな地図上で良く見える大きさに変更します。

基本的な地図レイアウトの準備ができたので、[地図帳] ツールを使用して、必要と思われる数の詳細図をこの形式で生成します。

14.6.6 ★☆☆（初級レベル）理解しよう: 地図帳カバレッジを作る

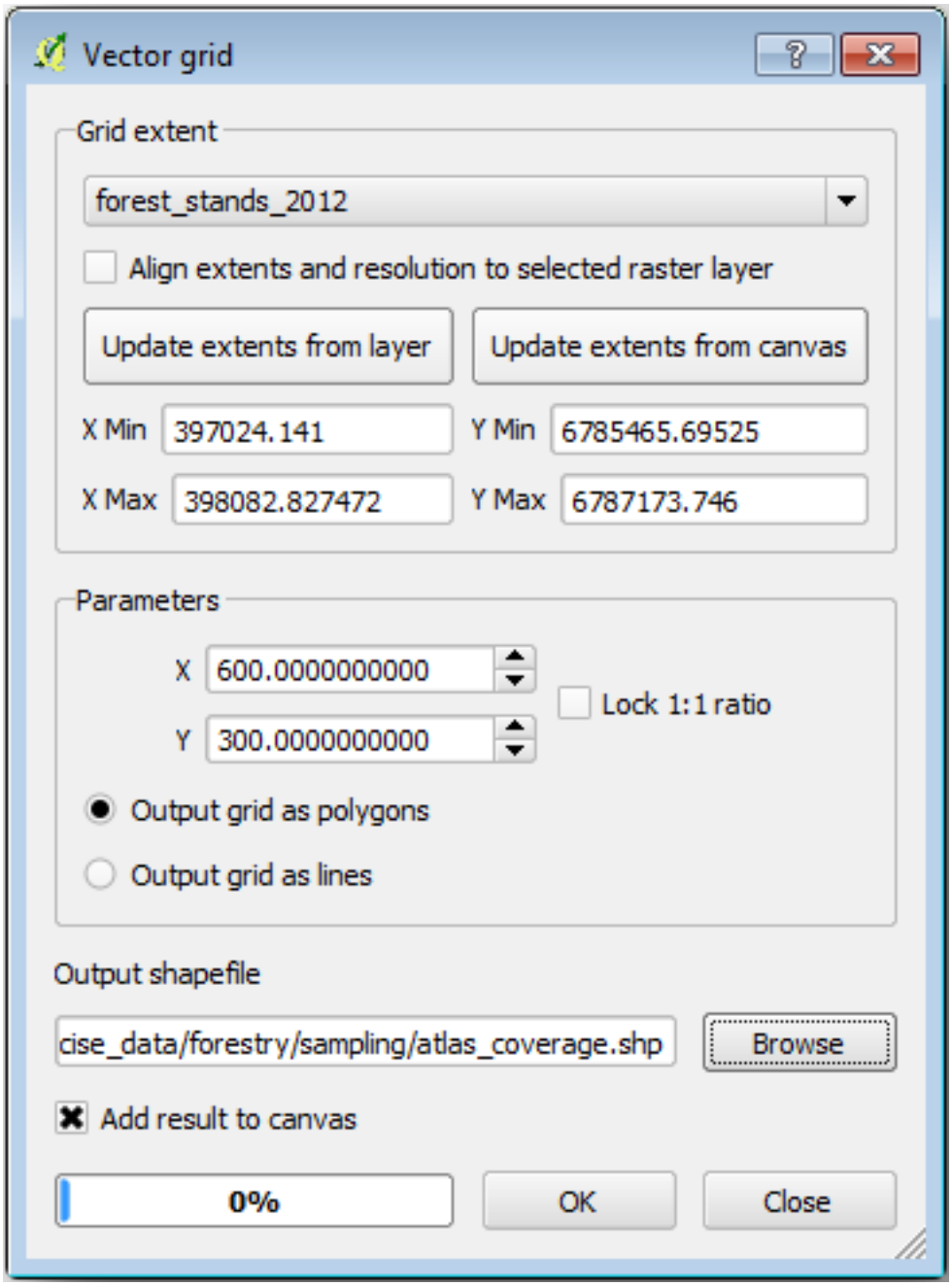
地図帳カバレッジは、カバレッジ内のすべての地物に対して 1 枚の地図という、詳細地図を作成するために使う、ただのベクタレイヤです。次に何をするのかイメージできるように、森林地域のため詳細地図のフルセットをお見せします：



カバレッジは既存のレイヤでも良いのですが、通常はその特定の目的のために 1 つ作成する方が理にかなっています。森林地域を範囲にするポリゴンのグリッドを作成してみましょう：

1. QGIS マップビューで、**ベクタ**　**調査ツール**　**グリッドを作成** を開きます。

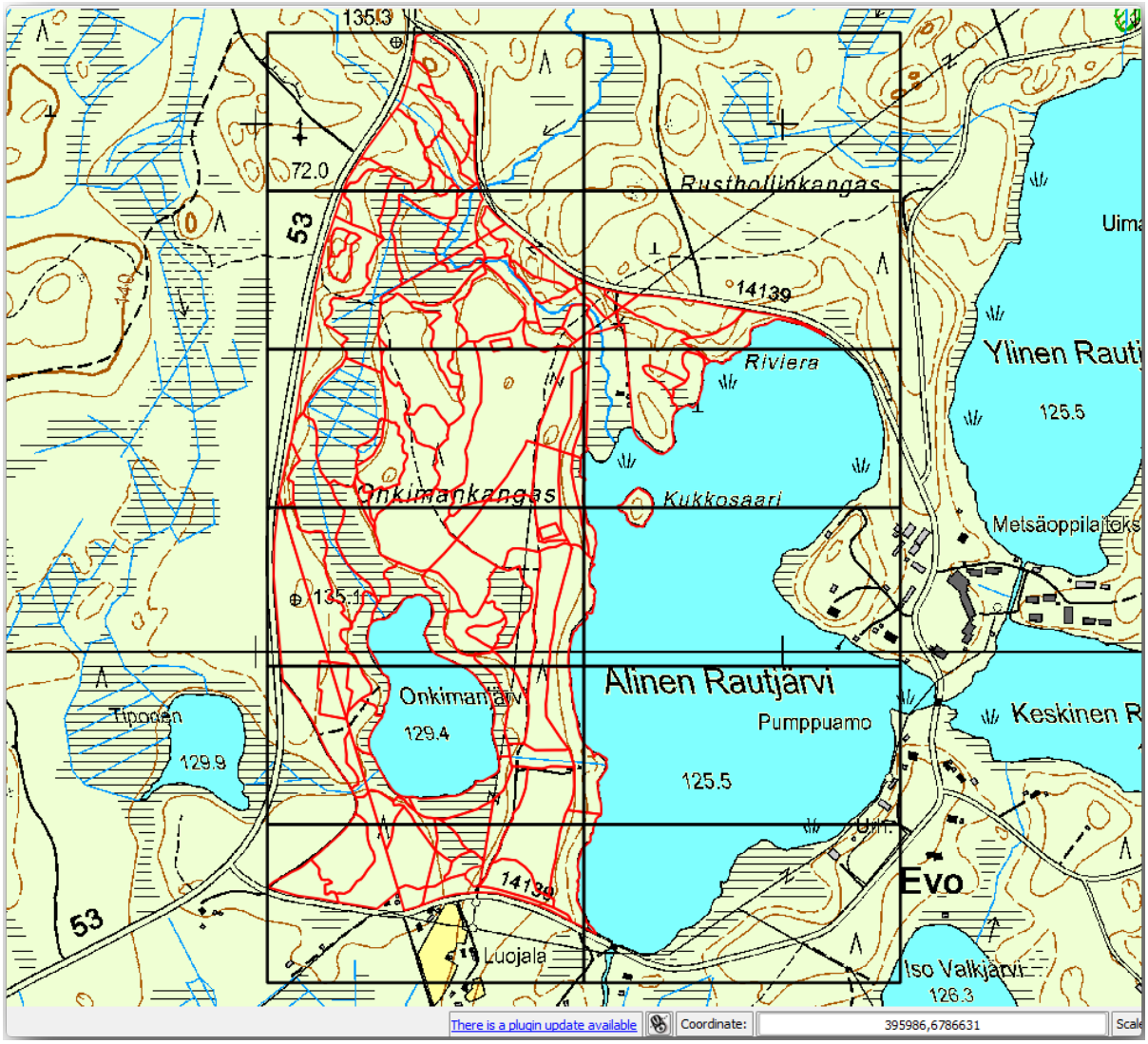
2. この画像に示すようにツールを設定します。



3. 出力を `atlas_coverage.shp` として保存します。

4. 新しいレイヤ `atlas_coverage` のスタイルをポリゴンが塗りつぶされないものにします。

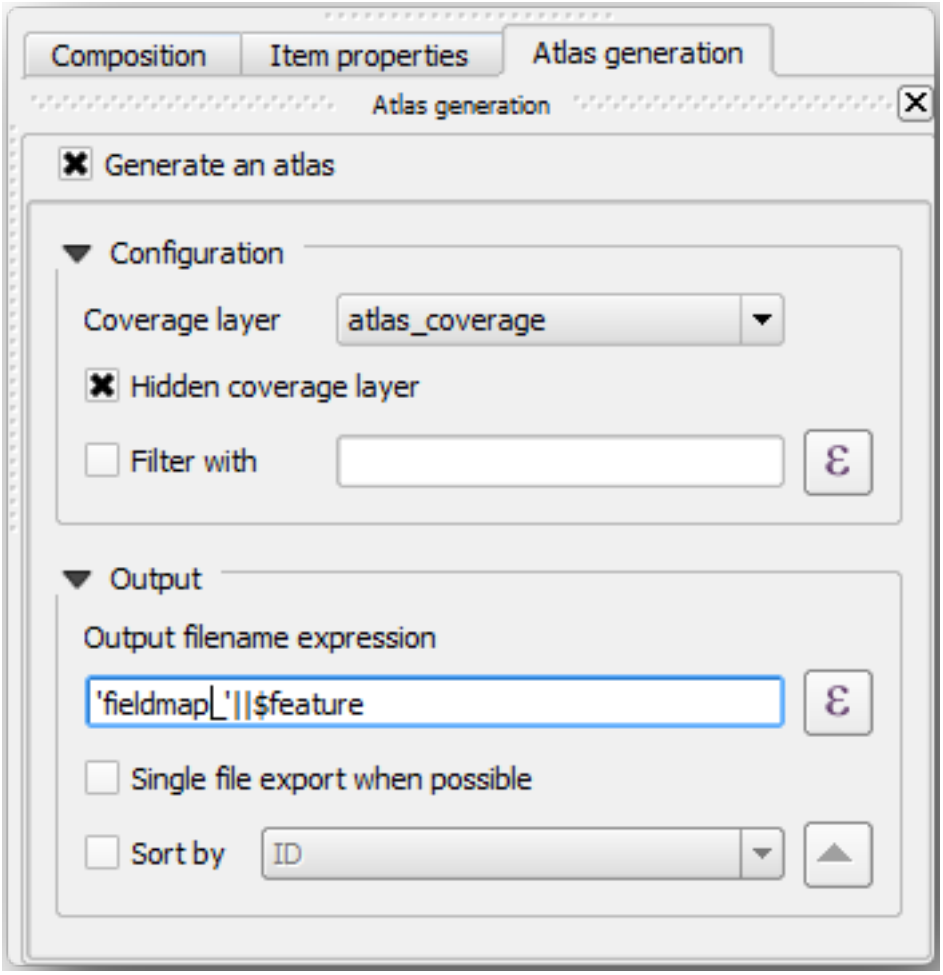
新しいポリゴンは全体の森林地域をカバーしていて、それらは（各ポリゴンから作成された）各地図に何が含まれているかを教えてくれます。



14.6.7 ★☆☆ （初級レベル） 理解しよう: 地図帳ツールを設定する

最後のステップは、地図帳ツールを設定することです：

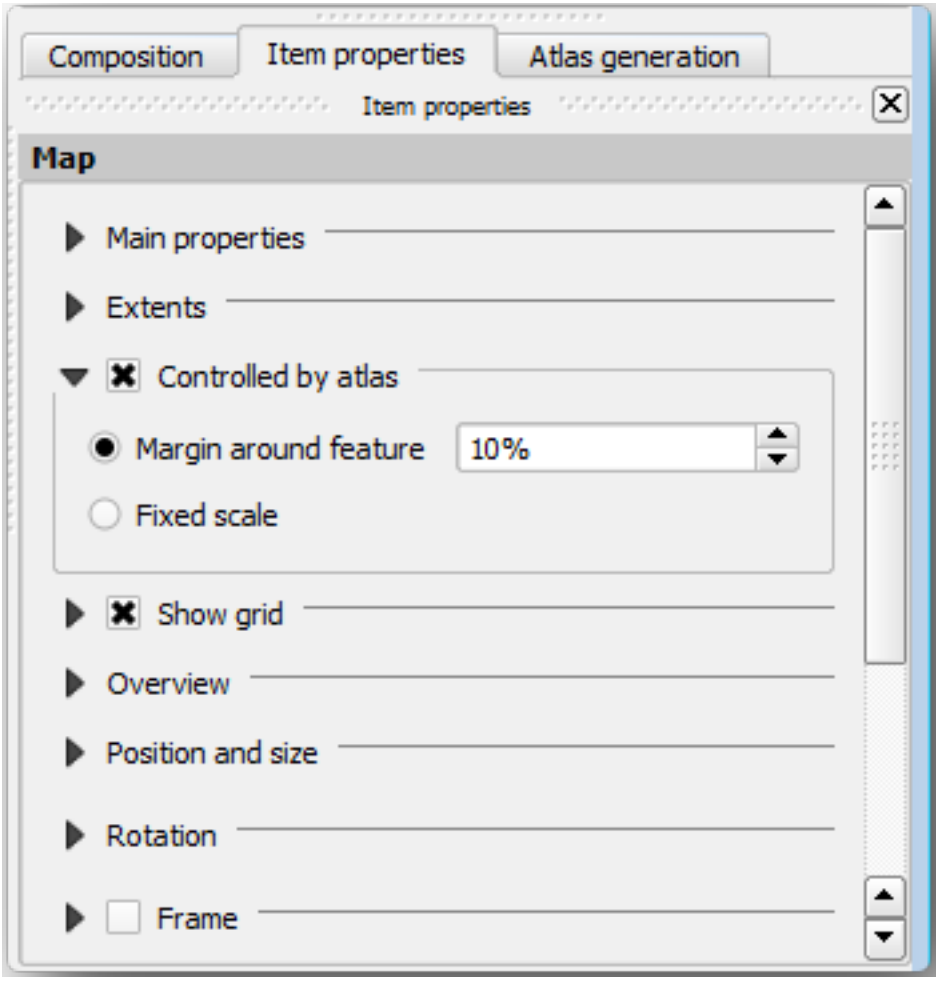
1. 印刷レイアウトに戻ります。
2. 右側のパネルで **地図帳** タブに行きます。
3. 次のようにオプションを設定します。



これは、地図帳ツールに `atlas_coverage` 内部の地物（ポリゴン）をそれぞれの詳細地図の焦点として使うように指示します。レイヤ内の各地物に対して 1 つの地図を出力します。**カバレッジレイヤを隠す** は、出力地図にポリゴンを表示しないよう地図帳に指示します。

もうひとつしておくことがあります。すべての出力地図に対してどの地図要素が更新されるかを地図帳ツールに伝える必要があります。もうお分かりだと思いますが、地物ごとに変更する地図は、サンプルプロットの詳細ビューを含むように準備したもの、つまりキャンバス内の大きい方の地図要素です：

1. 大きい方の地図要素 (`Map 0`) を選択します。
2. **アイテムプロパティ** タブに移動します。
3. リストで、**地図帳による制御** をチェックします。
4. そして **地物周りの余白** を `10%` に設定します。ビューの範囲はポリゴンより 10%大きくなり、詳細地図は 10%重なることになります。



今、地図がどのように見えるか確認するために地図帳地図のプレビューツールを使用できます。

1. ボタン を使用して、または地図帳ツールバー が表示されていない場合は **地図帳** **地図帳をプレビュー** を通じて、地図帳のプレビューを有効にします。
2. 作成される地図を移動するには、地図帳ツールバーまたは **地図帳** メニューの矢印を使用できます。

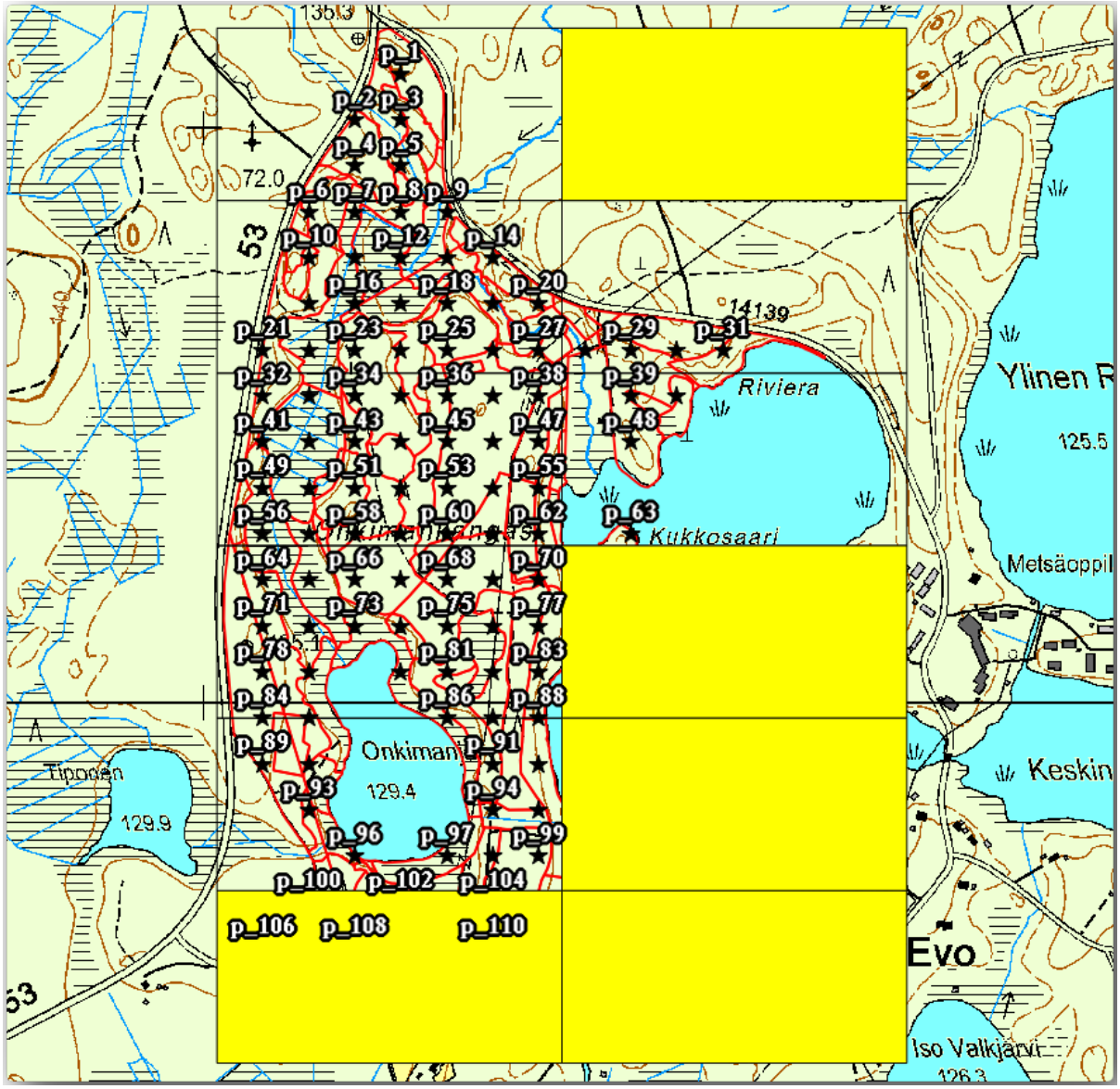
その中には、面白くもない地域をカバーしているものもあることに注意してください。そんな無駄な地図は印刷せず、木を節約しましょう。

14.6.8 ★☆☆ （初級レベル） 理解しよう: カバレッジレイヤを編集する

それらの地域のために関心のないポリゴンを削除するだけでなく、カバレッジ レイヤの **属性テーブル** からのコンテンツを生成するように、地図内のテキストラベルもカスタマイズできます。

1. 地図ビューに戻ります。
2. Enable editing for the `atlas_coverage` layer.
3. 下の画像に（黄色で）選択されたポリゴンを選択します。
4. 選択したポリゴンを削除します。

5. 編集を無効にして、編集内容を保存します。



印刷レイアウトに戻り、地図帳のプレビューがレイヤに残したポリゴンだけを使うことが確認できます。

使用しているカバレッジレイヤは、まだ地図内のラベルの内容をカスタマイズするために使用できる有用な情報を持っていません。最初のステップはそれらを作成すること、たとえばポリゴン領域のゾーンコードや野外チームが考慮すべき備考を記載したフィールドを追加することができます：

1. Open the *Attribute table* for the `atlas_coverage` layer.
2. 編集を有効にします。
3.  計算機を使用して、以下の二つのフィールドを作成、入力します。
4. `Zone` という名前で **整数** *(integer)* 型のフィールドを作ります。
5. **式** ボックスに `@row_number` を書く／コピー／組み立てます。
6. `Remarks` という名前の **テキスト** *(string)* 型で長さが `255` のフィールドを作ります。

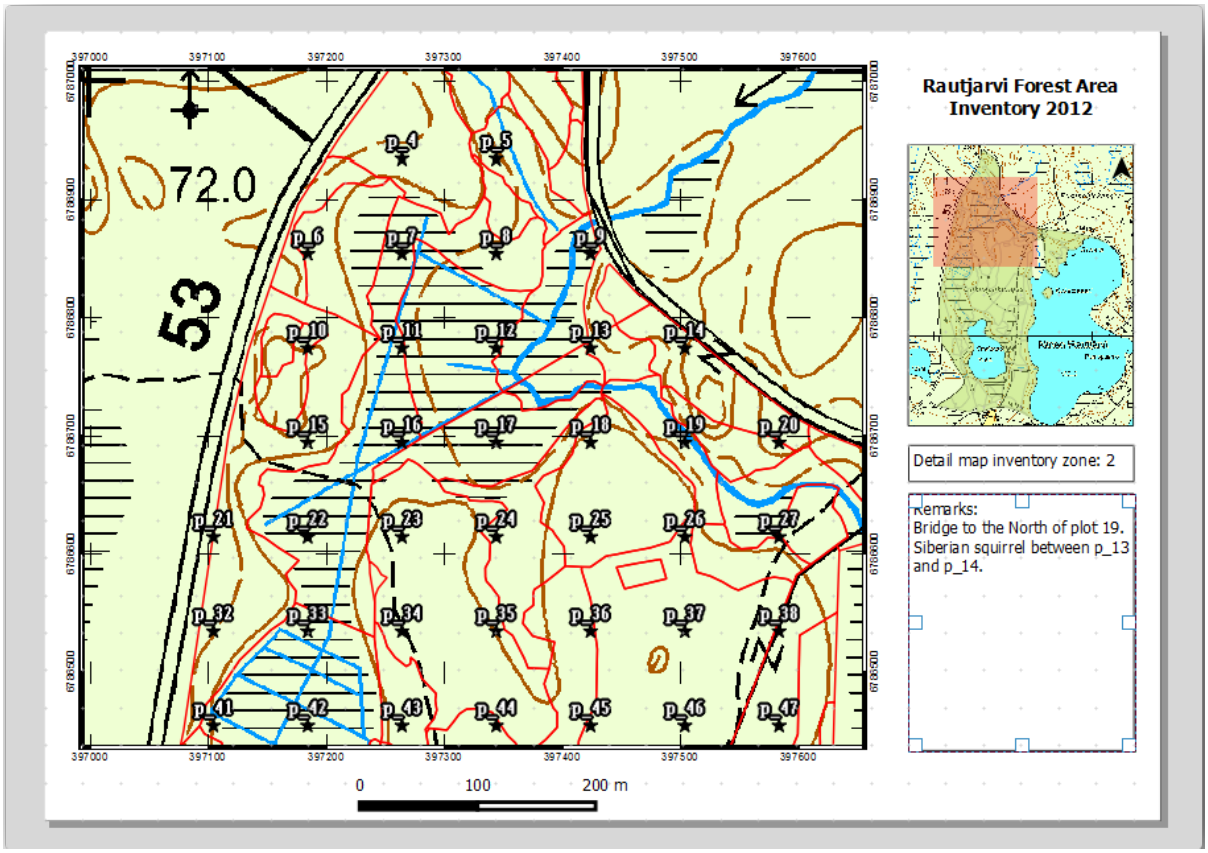
7. **式** ボックスに `'No remarks.'` と書きます。これですべてのポリゴンに規定値を設定します。

The forest manager will have some information about the area that might be useful when visiting the area. For example, the existence of a bridge, a swamp or the location of a protected species. The `atlas_coverage` layer is probably in edit mode still, add the following text in the *Remarks* field to the corresponding polygons (double click the cell to edit it):

- *Zone* 2: `plot 19` の北に橋。`P_13` と `p_14` の間にエゾモモンガ。
- For the *Zone* 6: `Difficult to transit in swamp to the North of the lake.`
- *Zone* 7: `p_94` の南東にエゾモモンガ。
- 編集を無効にして、編集内容を保存します。

Almost ready, now you have to tell the Atlas tool that you want some of the text labels to use the information from the `atlas_coverage` layer's attribute table.

1. **印刷レイアウト** に戻ります。
2. 詳細地図... というテキストラベルを選びます。
3. **フォント** の大きさを 12 にします。
4. ラベル内のテキストの末尾にカーソルを設定します。
5. **アイテムプロパティ** タブで、 **メインプロパティ** の内側で **式を挿入または編集** をクリックしてください。
6. **関数リスト** から **フィールドと値** の下にあるフィールド *Zone* をダブルクリックします。
7. *OK* をクリックします。
8. The text inside the box in the *Item properties* should show `Detail map inventory zone: [% "Zone" %]`. Note that the `[% "Zone" %]` will be substituted by the value of the field `Zone` for the corresponding feature from the layer `atlas_coverage`.
9. 異なる地図帳プレビュー地図を見て、ラベルの内容をテストします。
10. Do the same for the labels with the text `Remarks:` using the field with the zone information. You can leave a break line before you enter the expression. You can see the result for the preview of zone 2 in the image below:



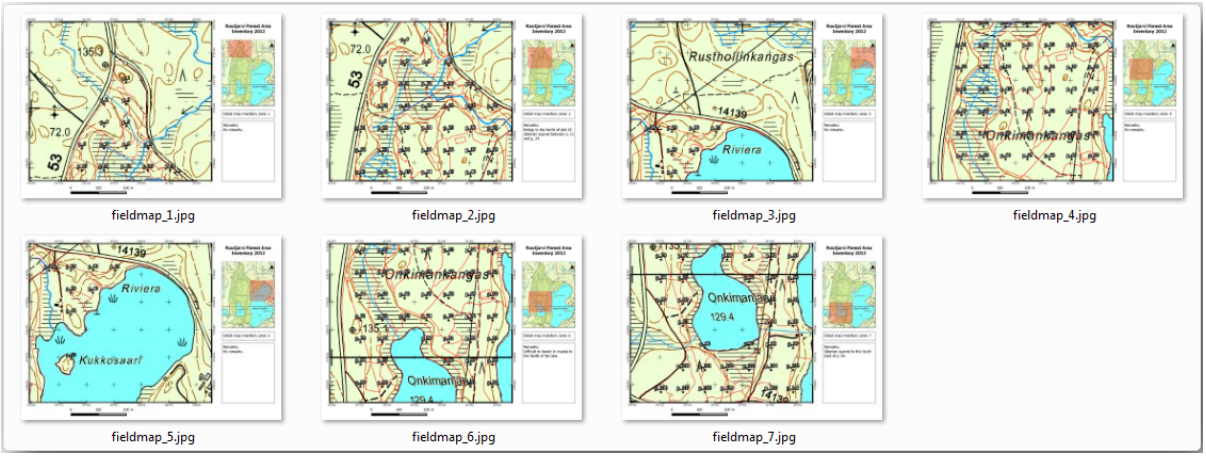
11. 地図帳プレビューを使用して。すぐに作成されているであろうすべての地図を閲覧してお楽しみください！

14.6.9 ★☆☆ （初級レベル） 理解しよう: 地図を印刷する

最後になりましたが、印刷や画像ファイルや PDF ファイルに地図をエクスポートします。 **地図帳　地図帳を *PDF* として書き出し...** や **地図帳　地図帳を画像として書き出し...** を使用できます。現在、SVG のエクスポート形式は正常に動作せず、貧しい結果が得られます。

現地事務所に印刷用に送付できるよう、単一の PDF として地図を印刷しましょう：

1. 右パネルの **地図帳作成** タブに移動します。
2. **出力** の下で **可能ならば単一ファイルに書き出し** をチェックします。これはすべての地図を 1 つの PDF ファイルにまとめます。このオプションがチェックされていない場合はすべての地図に 1 つずつファイルが得られます。
3. **レイアウト　*PDF* として書き出し...** を開きます。
4. PDF ファイルを `inventory_2012_maps.pdf` として `exercise_data\forestry\sampling\map_creation\` フォルダに保存します。
5. すべてが期待どおりに行ったことを確認するために PDF ファイルを開きます。
6. 同じように簡単にすべての地図に対して別々の画像を作成できます（単一ファイルの作成をオフにすることを忘れないでください）、作成されるであろう画像のサムネイルをここで見ることができます：



7. **印刷レイアウト** で 保存 を押し、印刷レイアウトの変更をプロジェクトに保存します。これにより、プロジェクトファイルも保存されます。いつでもプロジェクトを開き、地図帳を実行したり編集したりすることができます。

   また、作成したマップをレイアウトテンプレートとして `forestry_atlas.qpt` として `exercise_data` フォルダに保存することもできます。**レイアウト　テンプレートとして保存** を使用します。このテンプレートは、他のプロジェクトで何度も使用することができるようになります。

8. **印刷レイアウト** とプロジェクトを閉じます。

14.6.10 結論

別のプロットに移動するのを助けるために野外で使用される詳細地図を自動作成するのに使用できるテンプレート地図の作成にこぎつけました。お気づきのように、これは簡単な作業ではありませんでしたが、他の地域についても同様の地図を作成する必要があるときに利益がもたらされ、先ほど保存したテンプレートを使用できます。

14.6.11 次は?

次のレッスンでは、LIDAR データを使って DEM を作る方法、そしてそれを使ってデータと地図の見え方を高める方法を見てゆきます。

14.7 レッスン: 森林パラメータを計算する

森林のパラメータを推定することは、森林調査の目標です。前のレッスンからの例を続け、野外で収集された調査情報を使い、最初に森林全体に対して、次に前にデジタイズした林分に対して、森林のパラメーターを計算しましょう。

このレッスンの目標： 森林パラメーターを全般レベルと林分レベルで計算する。

14.7.1 ★☆☆ （初級レベル） 理解しよう: 調査結果を追加する

野外チームは森林を訪れ、あなたが提供した情報の助けを借りて、すべてのサンプルプロットでの森林に関する情報を集めました。

多くの場合、情報は野外で紙の様式に収集され、その後でスプレッドシートに入力されます。サンプルプロットの情報は、QGIS で簡単に開けるように `.csv` ファイルに凝縮されています。

目録の設計に関するレッスンで使用した QGIS プロジェクト（ `forest_inventory.qgs` という名前をつけた筈です）を続けます。

まず、QGIS プロジェクトにサンプルプロットの測定を追加します。

1. **レイヤ**　**レイヤを追加**　*CSV* **テキストレイヤを追加**... に移動します。
2. `exercise_data/forestry/results/` にある `systematic_inventory_results.csv` というファイルをブラウズします。
3. **ポイント座標** オプションがチェックされていることを確認します。
4. 座標を表すフィールドを *X* と *Y* に設定します。
5. *OK* をクリックします。
6. プロンプトが表示されたら、CRS として *ETRS89 / ETRS-TM35FIN* を選択します。
7. 新しいレイヤーの **属性テーブル** を開いてデータを見てみましょう。

サンプルプロットの測定結果に含まれるデータの型は、 `exercise_data/forestry/results/` フォルダにあるテキストファイル `legend_2012_inventorydata.txt` で読み取ることができます。

先ほど追加した `systematic_inventory_results` レイヤは、実際には `.csv` ファイル内のテキスト情報を仮想的に表現したものに過ぎません。続行する前に、目録結果を実際の空間データセットに変換してください。

1. `systematic_inventory_results` レイヤを右クリックします。
2. `exercise_data/forestry/results/` フォルダをブラウズします。
3. そのファイルに `sample_plots_results.shp` という名前をつけます。
4. **保存したファイルを地図に追加** をチェックします。
5. プロジェクトから `systematic_inventory_results` レイヤを削除します。

14.7.2 ★☆☆ （初級レベル） 理解しよう: 森林全体のパラメータの推定

目録作成結果から、1 ヘクタールあたりの体積や幹数などの興味深いパラメータについて、この森林エリア全体の平均を計算することができます。系統的なサンプルプロットは同じ面積を表しているので、`sample_plots_results` レイヤからヘクタールあたりの体積と幹数の平均を直接計算することができます。

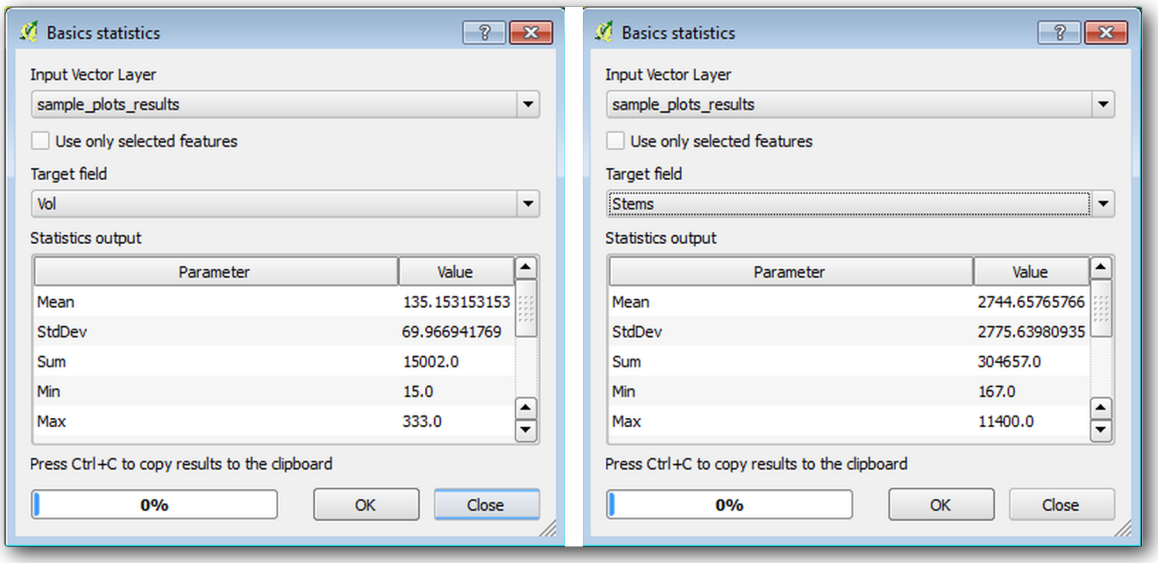
基本統計 ツールを使用して、ベクタレイヤ内のフィールドの平均値を計算できます:

1. **ベクタ**　**解析ツール**　**属性の基本統計量** を開きます。

2. **入力レイヤ** として sample_plots_results を選びます。
3. **統計量を計算する属性（フィールド）** として Vol を選びます。
4. *OK* をクリックします。

森林の平均体積は 135.2 m3/ha です。

幹数の平均も同様に 2745 本/ha と計算できます。



14.7.3 ★☆☆ （初級レベル） 理解しよう: 林分のパラメータを推定する

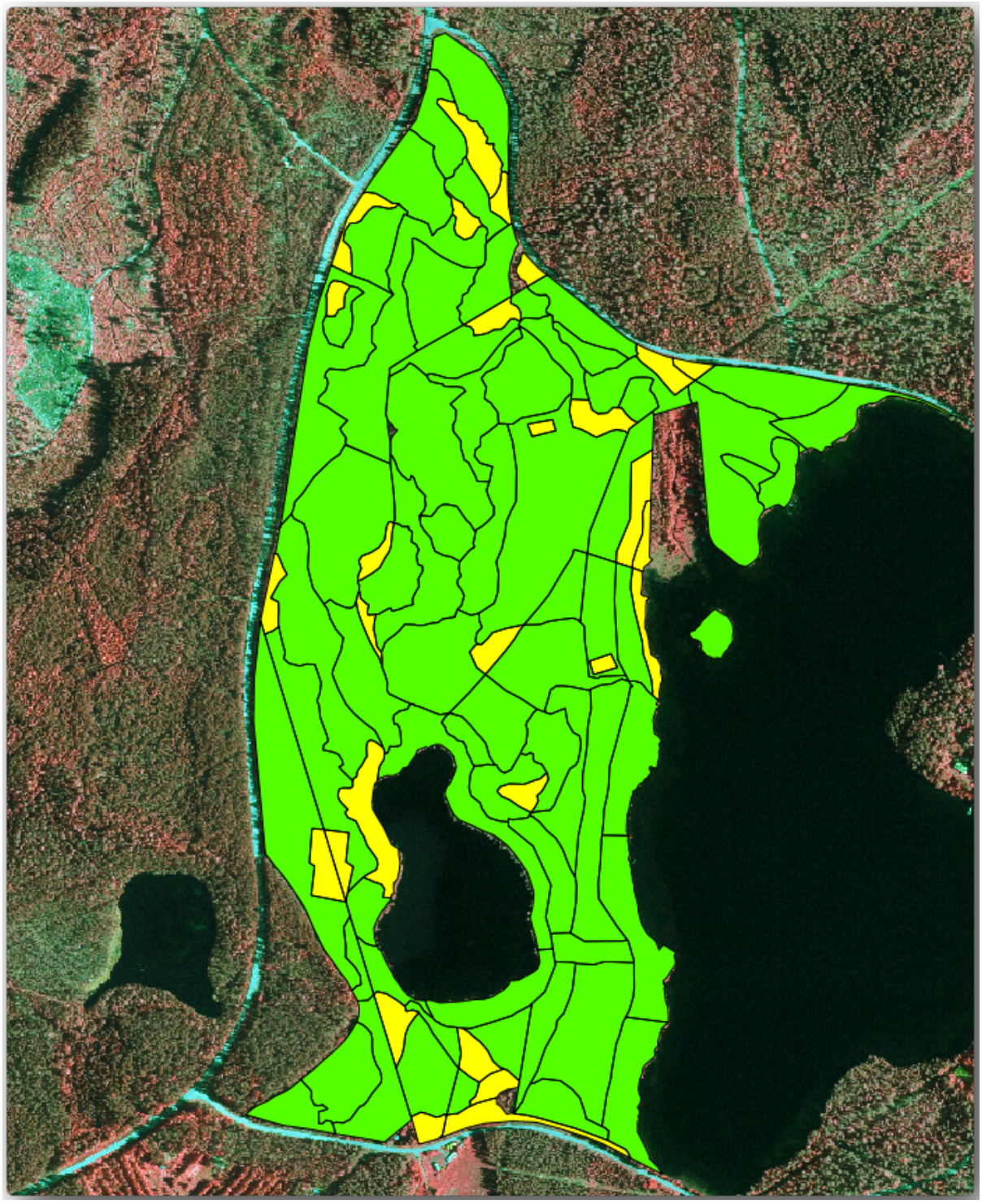
以前にデジタイズした林分の推定値を算出するのに、同じ体系的サンプルプロットを利用できます。林分の一部はどんなサンプルプロットも取得しておらず、それらについては情報を得ることはできません。体系的な目録を計画するときに、野外チームがこの目的のためにいくつかの余分なサンプルプロットを測定するように、いくつかの余分なサンプルプロットを計画することもできたかもしれません。あるいは後で野外チームを送って不足している林分の見積を取得することで林分の調査を完了することもできたかもしれません。それでも、計画されたプロットを使用するだけで林分のかなりの数の情報が得られるでしょう。

必要なのは林分の中にあるサンプルプロットの平均値を取得することです。それらの相対的な位置に基づいて情報を結合したいときは、空間的結合を行います：

1. **ベクタ　データ管理　属性の空間結合** ツールを開きます。
2. **ターゲットベクタレイヤ** に forest_stands_2012 を設定します。結果が欲しいレイヤ。
3. sample_plots_results を **結合するベクタレイヤ** に設定します。推定値を計算したいレイヤ。
4. **交差する地物を集計** をチェックします。
5. **平均** だけ計算するようチェックしてください。
6. 結果を forest_stands_2012_results.shp として exercise_data/forestry/results フォルダーに保存します。

7. 最後に **すべてのレコードを保管**... を選択し、どの林分が情報を取得していないか、後で内容を確認できるようにします。
8. *OK* をクリックします。
9. プロンプトが表示されたら、プロジェクトに新しいレイヤを追加し受け入れます。
10. **位置によって属性を結合** ツールを閉じます。

Open the *Attribute table* for `forest_stands_2012_results` and review the results you got. Note that a number of forest stands have `NULL` as the value for the calculations, those are the ones having no sample plots. Select them all and view them in the map, they are some of the smaller stands:



先ほどと同じように森林全体の平均を計算しましょう。ただし、今回は林分の平均を計算のベースとして使用します。先ほどは、各サンプルプロットが 80x80 m の理論上の林分を表していたことを思い出してください。その代わりに、それぞれの林分の面積を個別に考慮する必要があります。そうすることで、例えば体積を m3/ha で表したパラメータの平均値が、林分の総体積に変換されるのです。

最初に林分に対する面積を計算し、それからそれらの各々に対する合計体積および合計茎数を計算する必要があります：

1. **属性テーブル** 中で編集を有効にします。

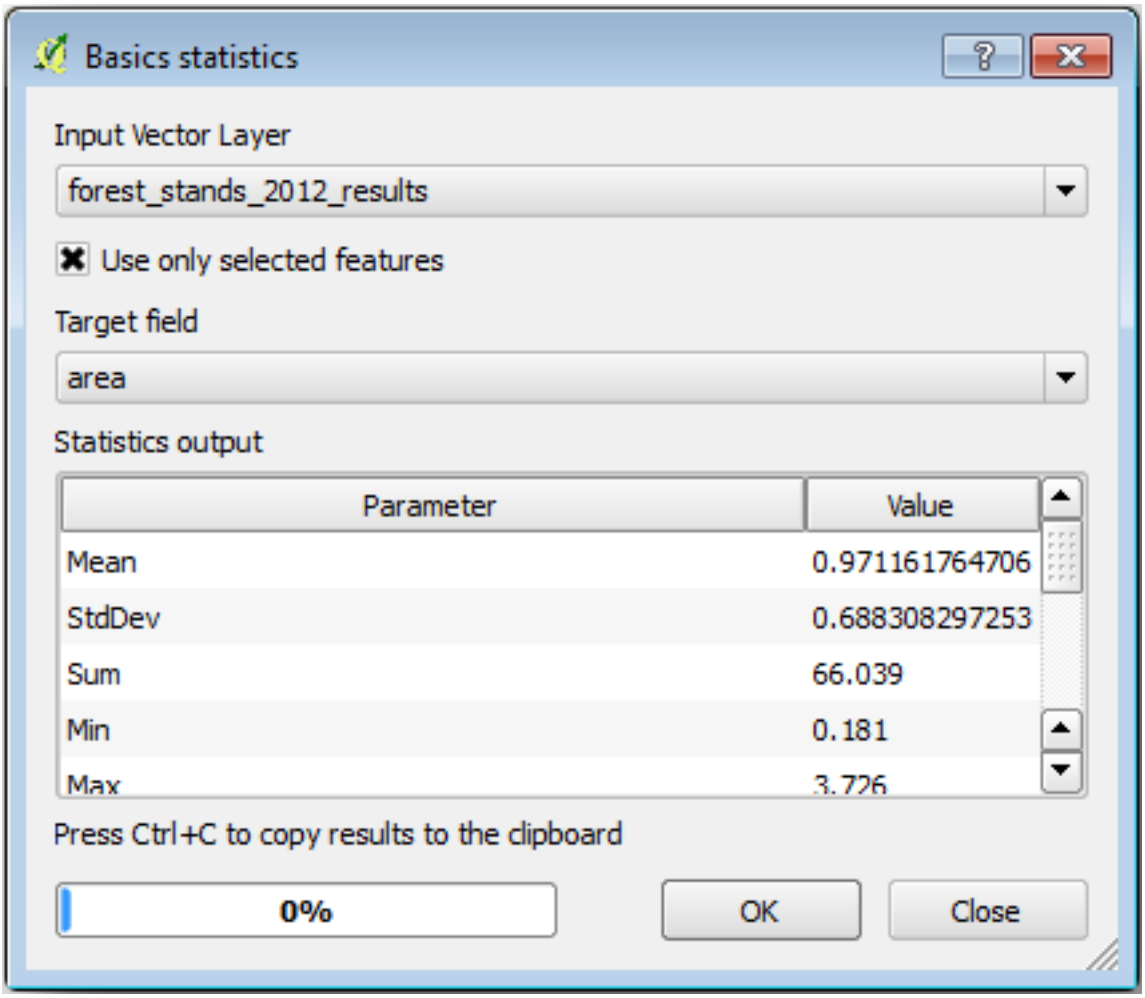
2. **フィールド計算機** を開きます。
3. area という新しいフィールドを作成します。
4. **フィールド型** に 小数点付き数値 (real) を設定します。
5. **精度** に 2 を設定します。
6. **式** ボックスに、 $area / 10000 を入力します。これは林分の面積を ha で計算します。
7. *OK* をクリックします。

ここですべての林分について推定される体積合計および幹数合計を持つフィールドを計算します：

1. そのフィールドを s_vol と s_stem とします。
2. フィールドは整数でもよいし、実数も使用できます。
3. 体積合計と幹数合計にそれぞれ式 "area" * "MEANVol" と "area" * "MEANStems" を使います。
4. 終了したら、編集内容を保存します。
5. 編集を無効にします。

以前の状況では、すべてのサンプルプロットによって表される面積が同じだったので、サンプルプロットの平均値を計算すれば十分でした。今は見積もりを計算するためには、林分の体積または幹数の合計を必要な情報を含む林分の面積の合計で割る必要があります。

1. forest_stands_2012_results レイヤの **属性テーブル** で情報を持っている全ての林分を選択します。
2. **ベクタ　解析ツール　属性の基本統計量** を開きます。
3. **入力レイヤ** に forest_stands_2012_results を選択します。
4. **統計量を計算する属性** に area を設定する。
5. **選択した地物のみ** をチェックする
6. *OK* をクリックします。



ここに見えるように、林分面積の合計は 66.04 ha です。抜けている林分の面積が 7 ha に過ぎないことに注意してください。

同じ方法で これらの林分の体積の合計が 8908 m3/ha、幹数の合計が 179594 stems であることが計算できます。

サンプルプロットから、下記の平均推定値を与えることを使用して林分からの情報を使用して、代わりに直接：

- 184.9 m3/ha と
- 2719 stems/ha。

QGIS プロジェクト forest_inventory.qgs を保存します。

14.7.4 結論

自分の体系的なサンプルプロットからの情報を使用して、最初は森林の特性を考慮することなしで、そしてまた林分への空撮画像の解釈を使用して、森林全体に対して森林の推定値を計算できました。また特定の林分に関するいくつかの貴重な情報も得ましたが、これは今後森林の管理を計画するために使用できることでしょう。

14.7.5 次は?

次のレッスンでは、まずレーザー測量データセットから陰影起伏の背景を作成します。これはたった今計算した森林の結果で地図プレゼンテーションを準備するために使用するしょう。

14.8 レッスン: レーザー測量データからの DEM

さまざまな背景画像を使用することで、地図の外観を改善できます。基本地図も以前に使用してきた空撮画像も使用できますが、地形の陰影起伏ラスターが見栄えを良くする状況もあるでしょう。

LAStools を使用してレーザー測量データセットから DEM を抽出し、その後、地図のプレゼンテーションで使用する陰影起伏ラスタを作成します。

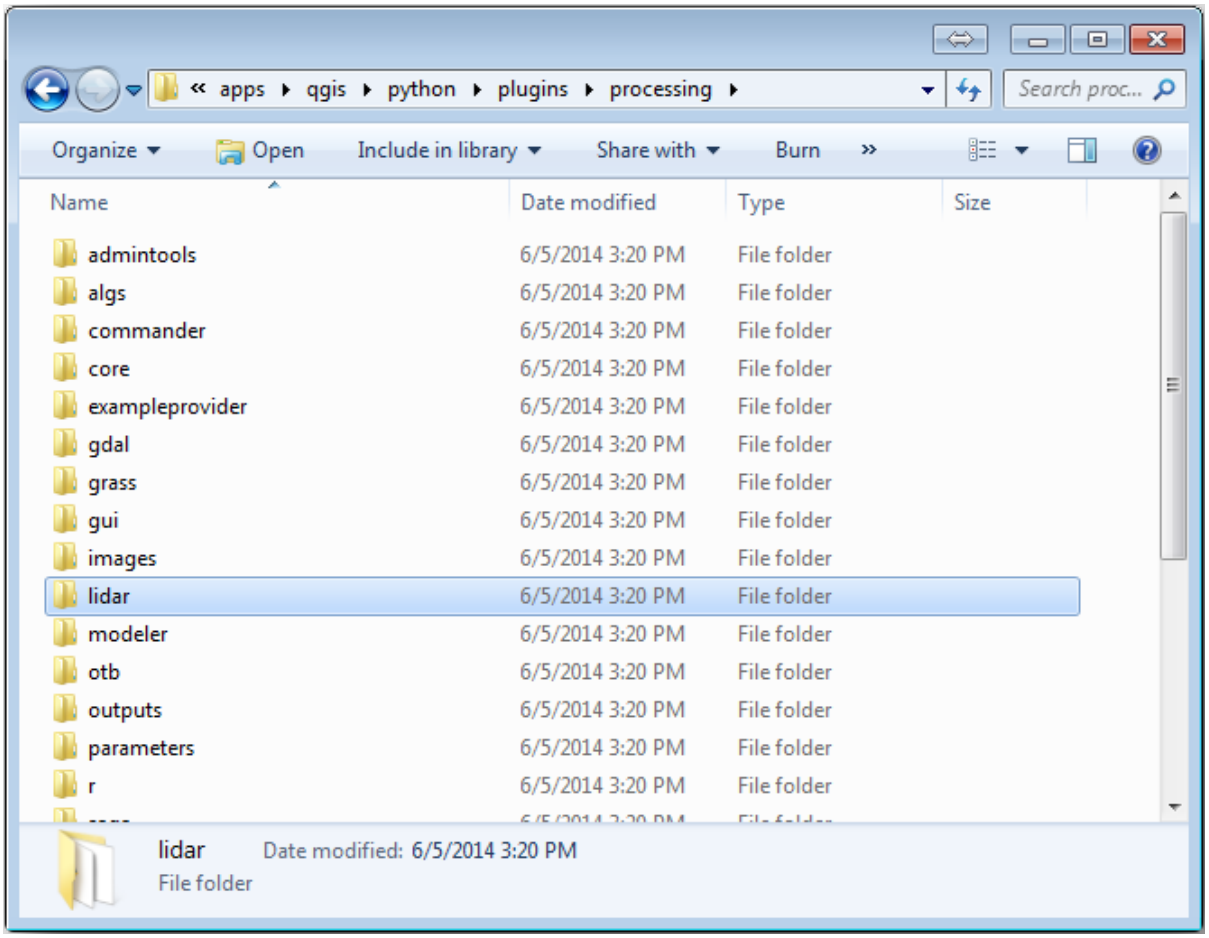
このレッスンの目標： LAStools をインストールし、レーザー測量データと陰影起伏ラスタから DEM を計算します。

14.8.1 ★☆☆ （初級レベル） 理解しよう: Lastools をインストールする

LiDAR のデータを QGIS で管理するには、プロセシングフレームワークと LAStools が提供するアルゴリズムが必要です。

レーザー測量点群から数値標高モデル（DEM）を取得して、プレゼンテーション目的のために、視覚的により直感的な陰影起伏ラスタを作成できます。最初に、**プロセシング** フレームワークの設定を、LAStools で動作するように適切に設定する必要があります。

- QGIS をすでに開始している場合は閉じます。
- 古いライダープラグインは、フォルダ `C:/Program Files/QGISValmiera/apps/qgis/python/plugins/processing/` 内のシステムにデフォルトでインストールされる可能性があります。
- If you have a folder named `lidar`, delete it. This is valid for some installations of QGIS 2.2 and 2.4.



- Go to the `exercise_data\forestry\lidar\` folder, there you can find the file `QGIS_2_2_toolbox.zip`. Open it and extract the `lidar` folder to replace the one you just deleted.
- 異なる QGIS のバージョンを使っている場合は、このチュートリアル にインストールの説明が見つかります。

次に、LAStools をコンピュータにインストールする必要があります。最新の *lastools* バージョンを ここ から入手し、 `lastools.zip` ファイルの内容をシステム内のフォルダー、例えば `C:\lastools\` に展開します。file:*lastools* フォルダーのパスには、スペースや特殊文字は使用できません。

注釈

Read the `LICENSE.txt` file inside the `lastools` folder. Some of the LAStools are open source and other are closed source and require licensing for most commercial and governmental use. For education and evaluation purposes you can use and test LAStools as much as you need to.

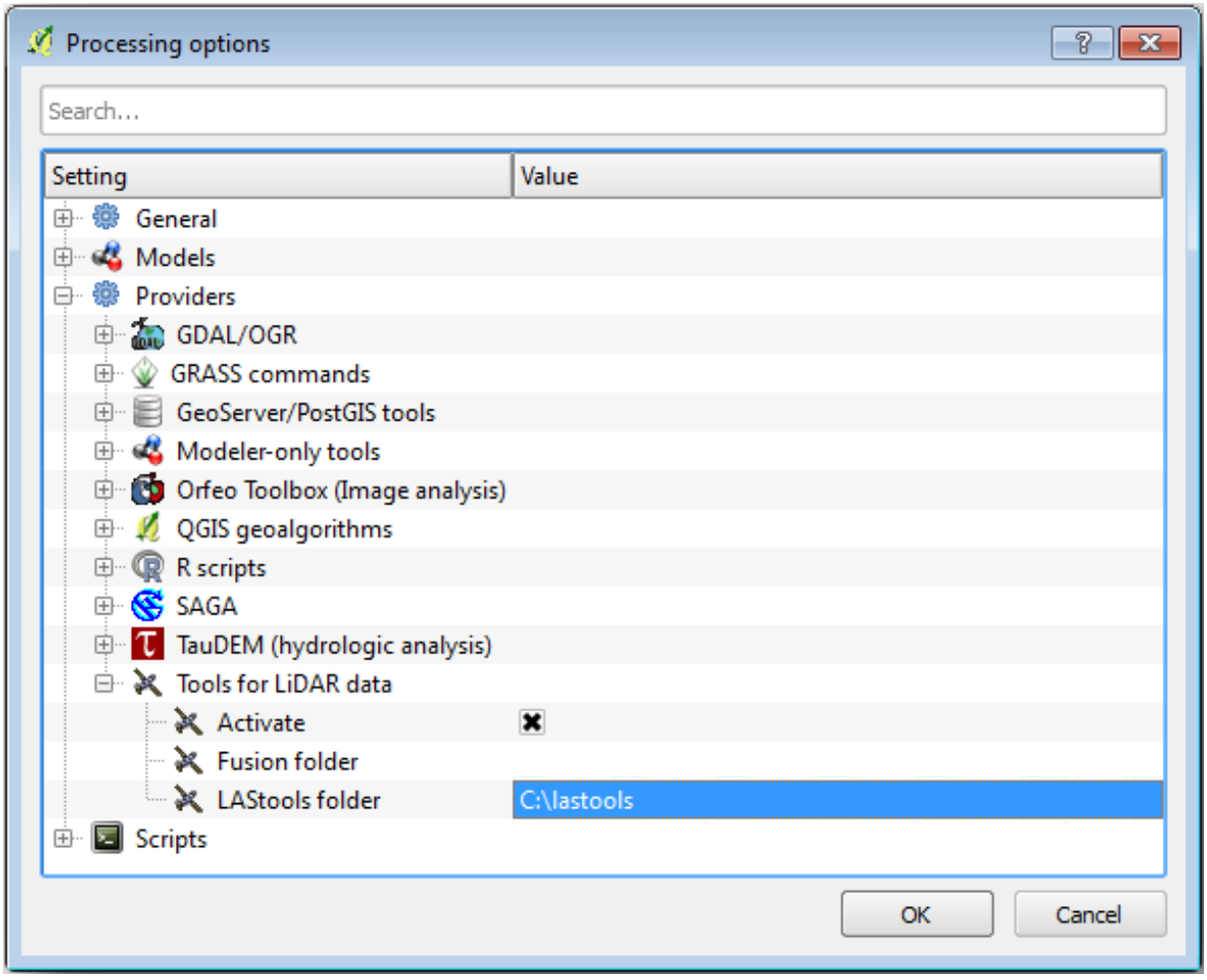
今プラグインと実際のアルゴリズムはお手元のコンピュータにインストールされ、使用する準備がほぼできていますので、それらの使用を開始するためにプロセシングフレームワークを設定することが必要であるだけです。

- QGIS で新しいプロジェクトを開きます。
- Set the project's CRS to *ETRS89 / ETRS-TM35FIN*.

- Save the project as `forest_lidar.qgs`.

QGIS で LAStools を設定するには：

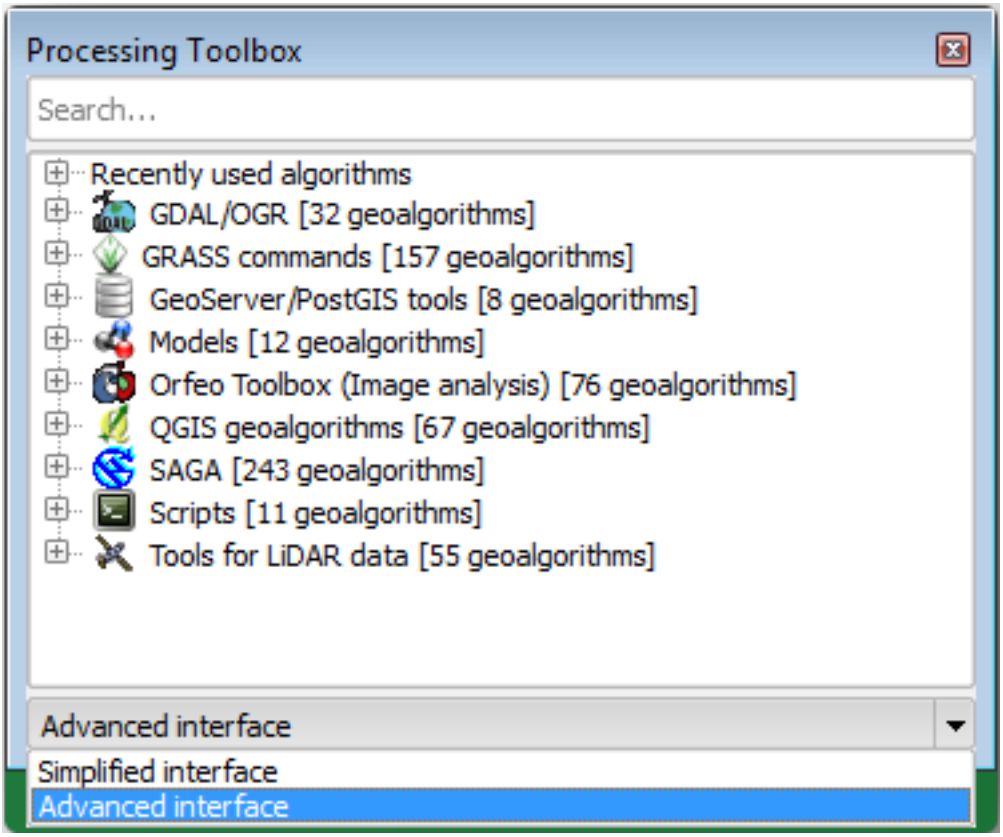
- **プロセシング** → **オプションと構成** に行きます。
- **プロセシングオプション** ダイアログで、 **プロバイダ** に、その後 *LiDAR* **データのためのツール** に行きます。
- *Activate* をチェックします。
- For *LAStools folder* set `c:\lastools\` (or the folder you extracted LAStools to).



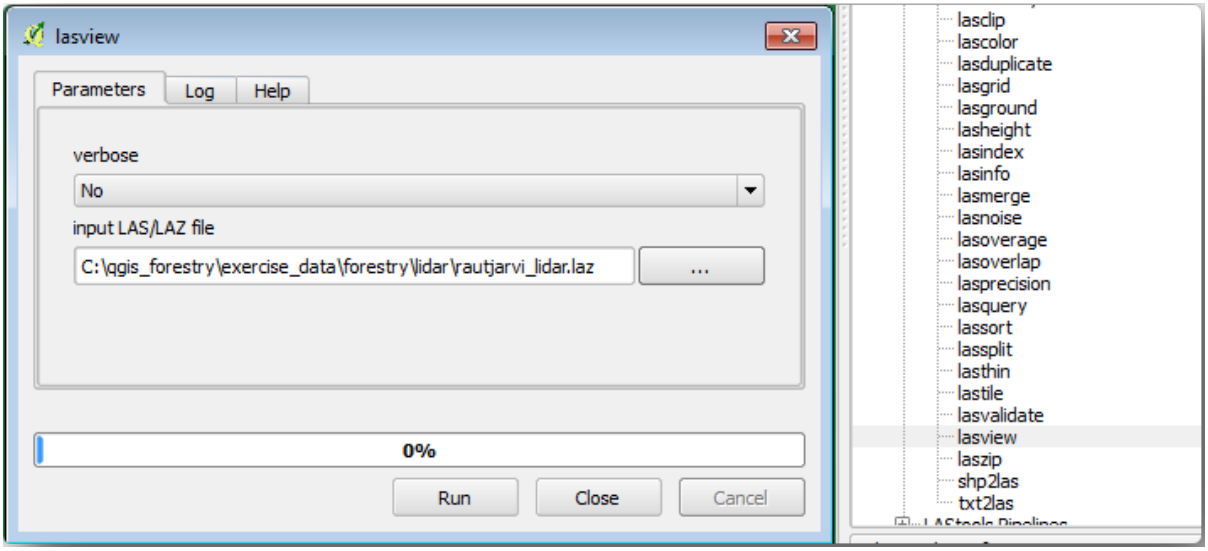
14.8.2 ★☆☆ （初級レベル） 理解しよう: LAStools で DEM を計算する

いくつかの SAGA アルゴリズムを実行するために、 レッスン: 空間統計 中の **プロセシング** ツールボックスはすでに使用してきました。ここでは LAStools プログラムを実行するためにそれを使用しようとしています。

- **プロセシング** → **ツールボックス** を開きます。
- 一番下にあるドロップダウンメニューで、**高度なインターフェイス** を選択します。
- **レーザー測量データ** カテゴリのツールが表示されるはずです。

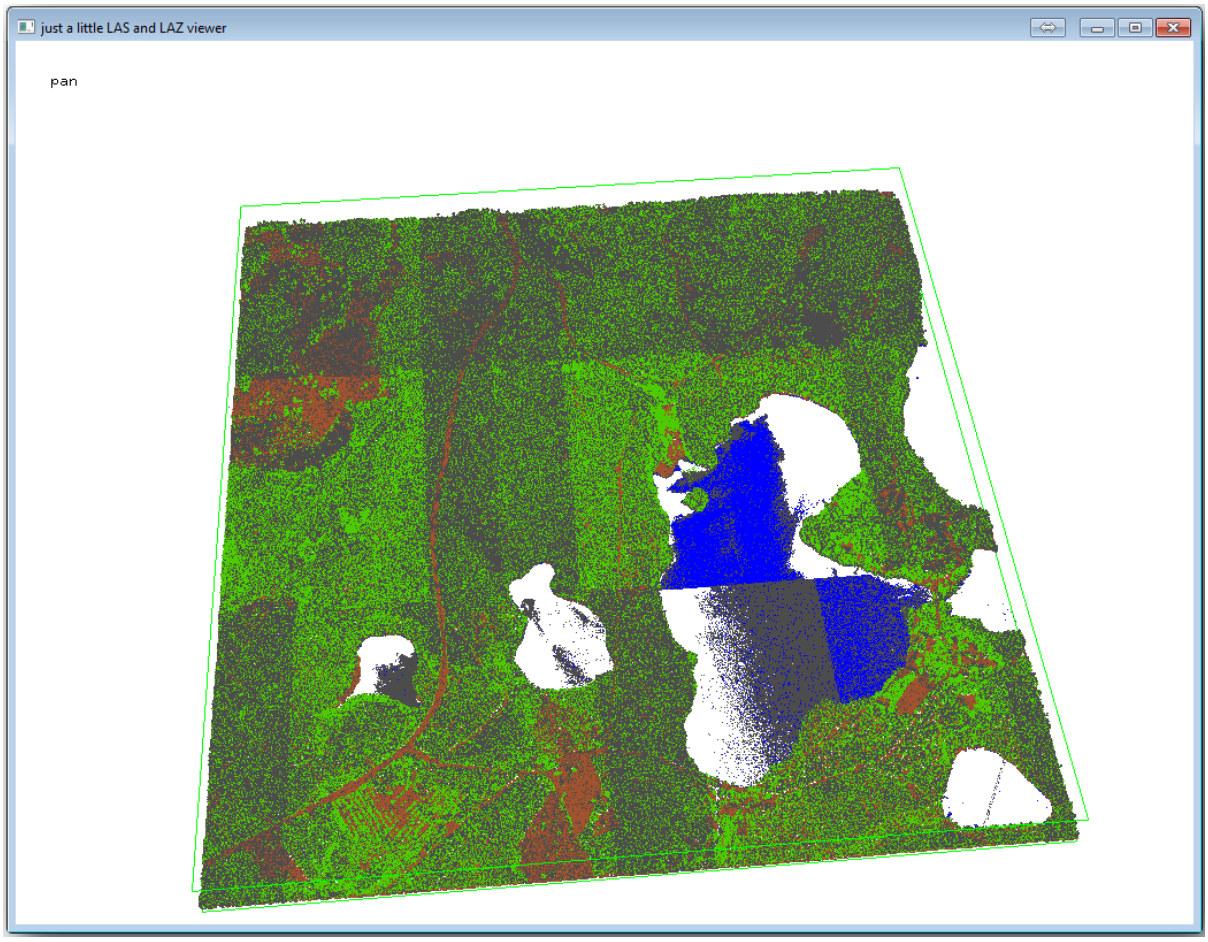


- 利用可能なツールが見えるようにそれを展開し、さらに *LAStools* カテゴリを展開します（アルゴリズムの数は変化する場合があります）。
- *lasview* アルゴリズムが見つかるまで下にスクロールし、見つかったらダブルクリックして開きます。
- **入力** *LAS/LAZ* **ファイル** で `exercise_data\forestry\lidar\` を探し、`rautjarvi_lidar.laz` ファイルを選択します。



- **実行** をクリックします。

今、 **ほんの少し** *LAS* **と** *LAZ* **ビューア** ダイアログウィンドウ内に LiDAR データを見ることができます：



このビューア内ではいろいろなことができますが、今はビューアをクリックおよびドラッグして LiDAR 点群にパンし、それがどのように見えるかを確認できます。

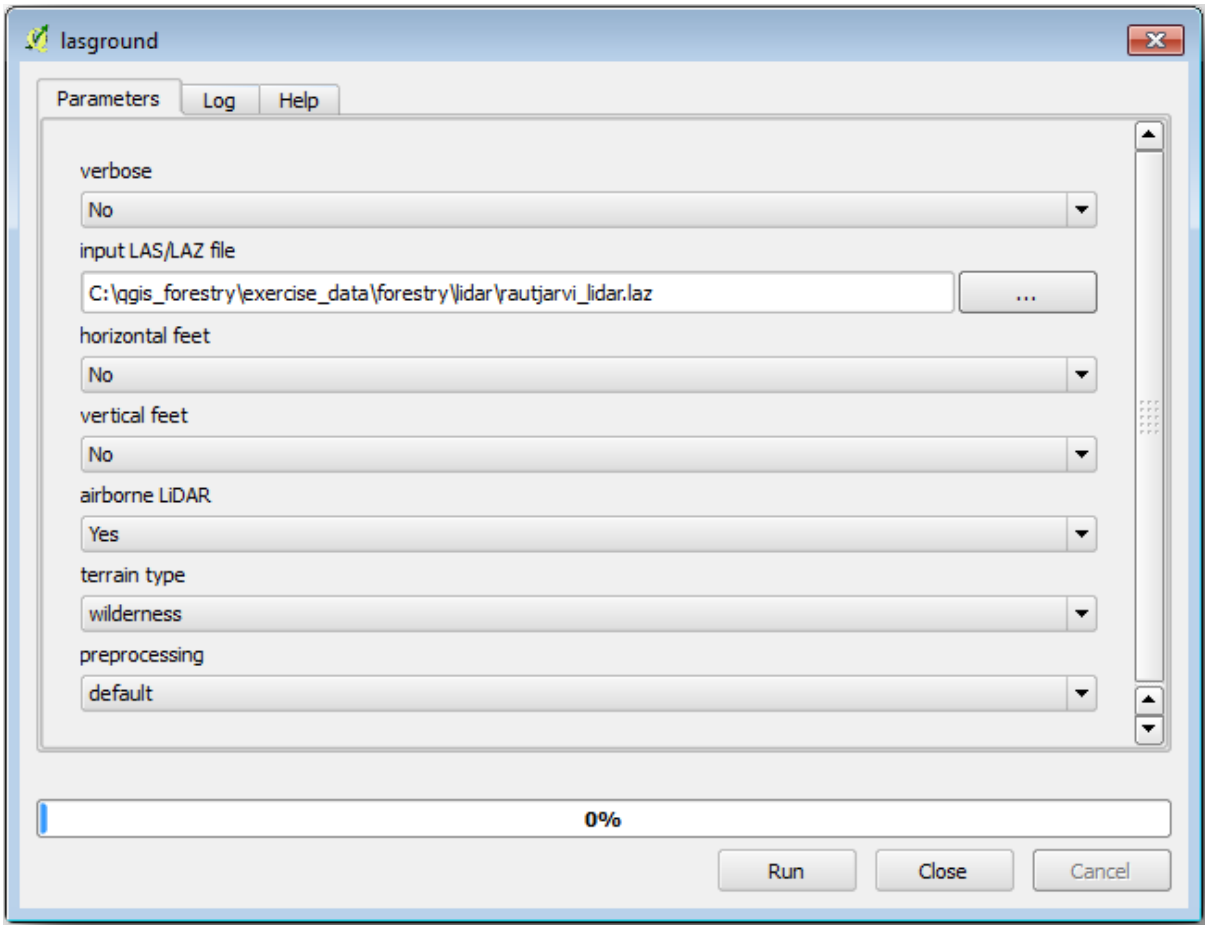
注釈

LAStools がどのように動くのか詳細が知りたいときは、`C:\lastools\bin\` フォルダーにある 各ツールに関する `README` テキストファイルを読むことができます。チュートリアルやその他の資料は、`Rapidlasso のウェブページ <https://rapidlasso.de/>_ で入手できます。

- 準備ができたら、ビューアを閉じます。

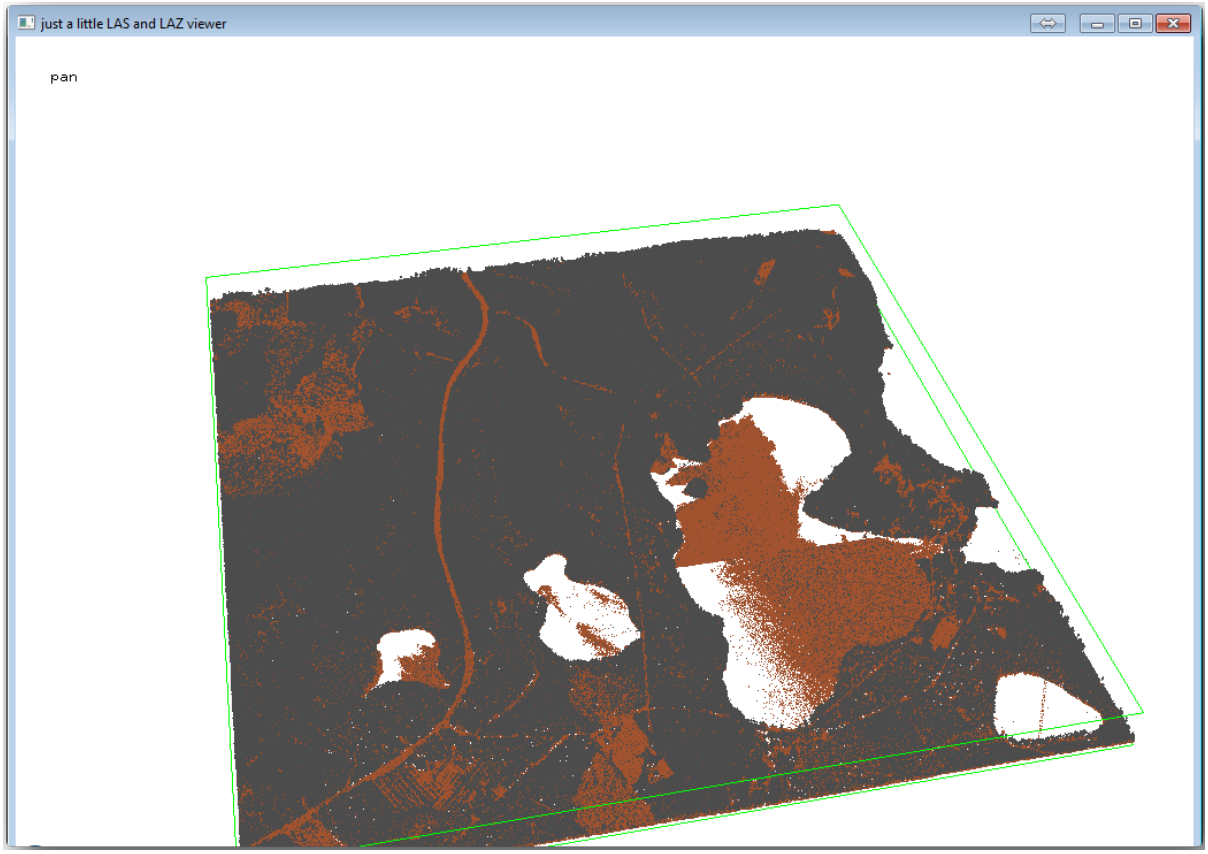
Creating a DEM with LAStools can be done in two steps, first one to classify the point cloud into *ground* and *no ground* points and then calculating a DEM using only the *ground* points.

- **処理ツールボックス** に戻ります。
- Note the *Search...* box, write `lasground`.
- ダブルクリックして *lasground* ツールを開き、この画像のように設定します：



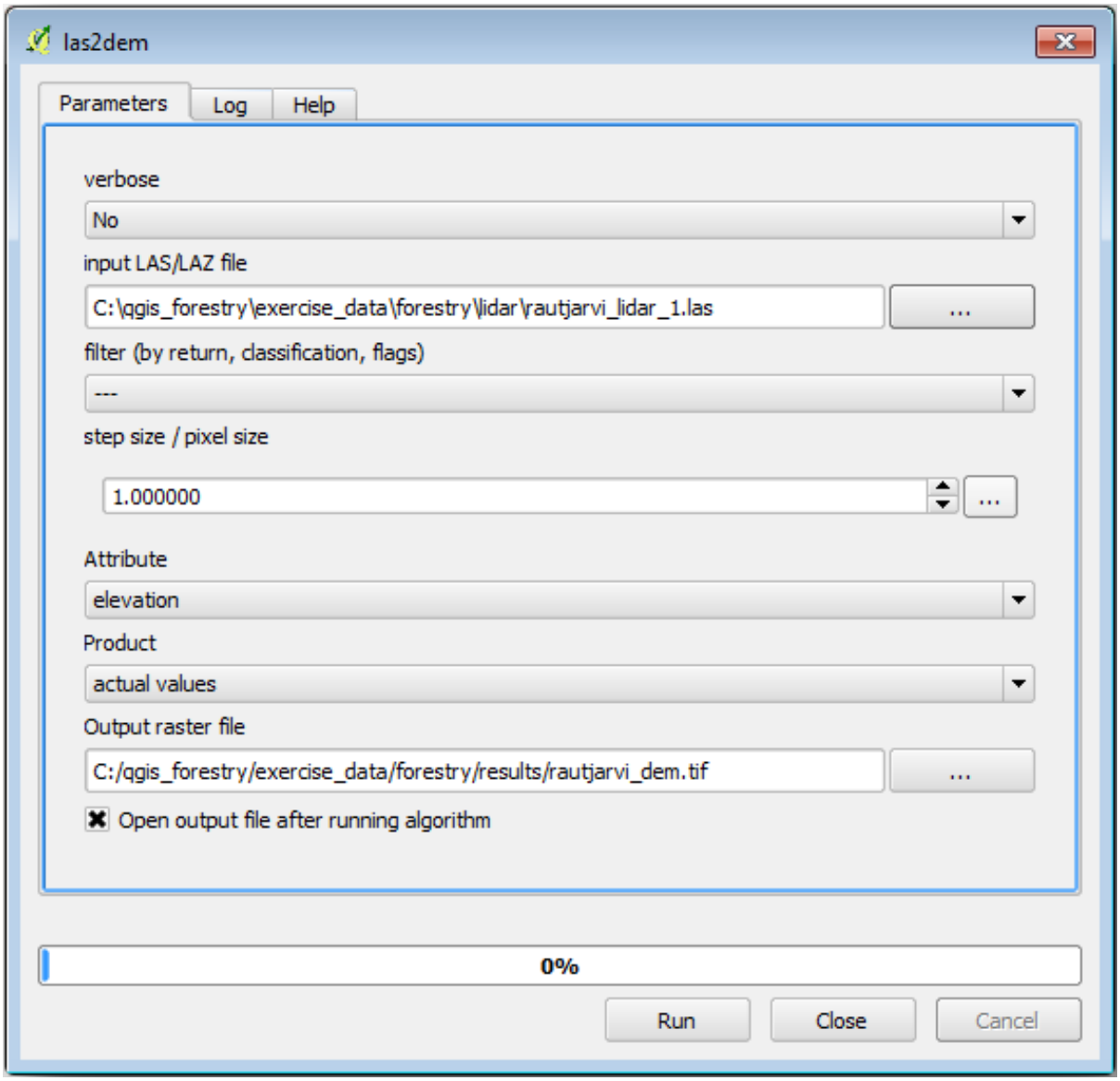
- 出力ファイルが `rautjarvi_lidar.laz` があるのとと同じフォルダーに `rautjarvi_lidar_1.las` という名前で保存されます。

それを確認したい場合は、*lasview* で開くことができます。



茶色の点は地上に分類された点で、灰色の点はそれ以外の点です。地上の点だけを表示するには `g` という文字を、分類されていない点だけを表示するには `u` という文字をクリックします。kbd:*a* の文字をクリックすると、すべてのポイントを再び見ることができます。その他のコマンドについては `lasview_README.txt` ファイルを参照してください。もし興味があれば、LiDAR ポイントの手動編集に関する チュートリアル もご覧ください。このチュートリアルでは、ビューアでのさまざまな操作を紹介しています。

- 再びビューアを閉じます。
- In the *Processing Toolbox*, search for `las2dem`.
- *las2dem* ツールを開き、この画像に示すように設定します。



The result DEM is added to your map with the generic name *Output raster file*.

注釈

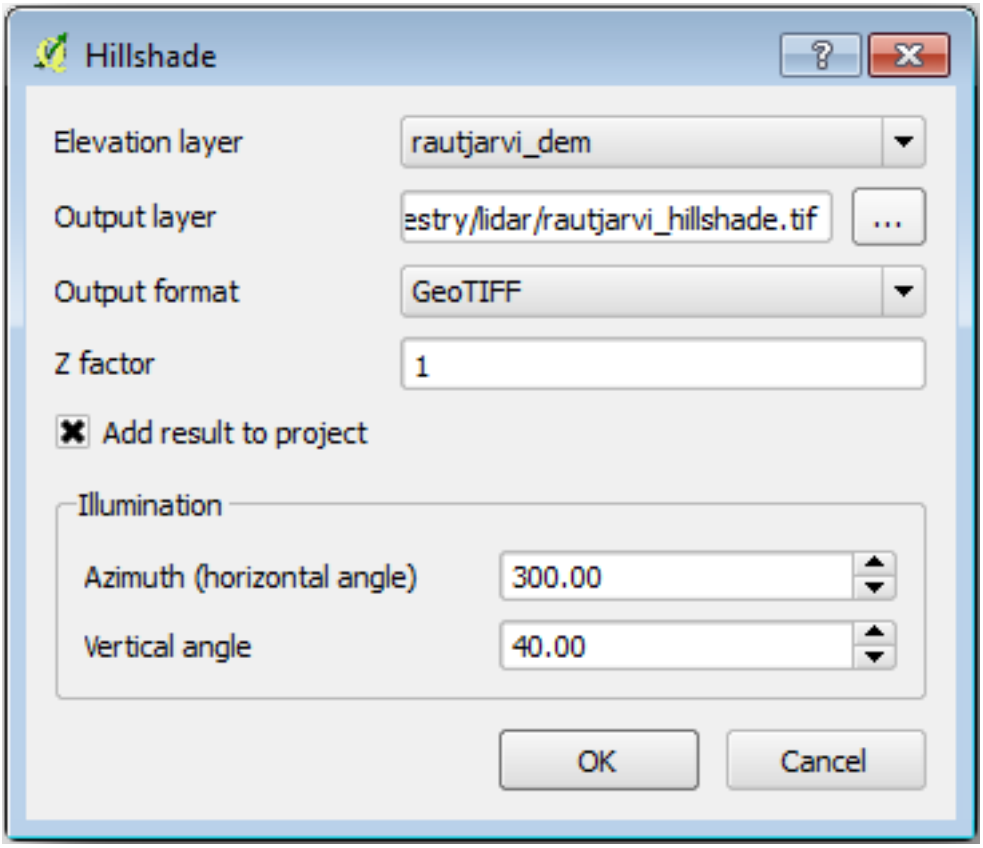
lasground と *las2dem* ツールはライセンスが必要です。ライセンス・ファイルに示されるように、unilicense のツールを使用できますが、画像結果にはっきりわかる対角線が出ます。

14.8.3 ★☆☆ （初級レベル） 理解しよう: 地形陰影起伏を作る

可視化の目的には DEM から生成される陰影起伏がより良い

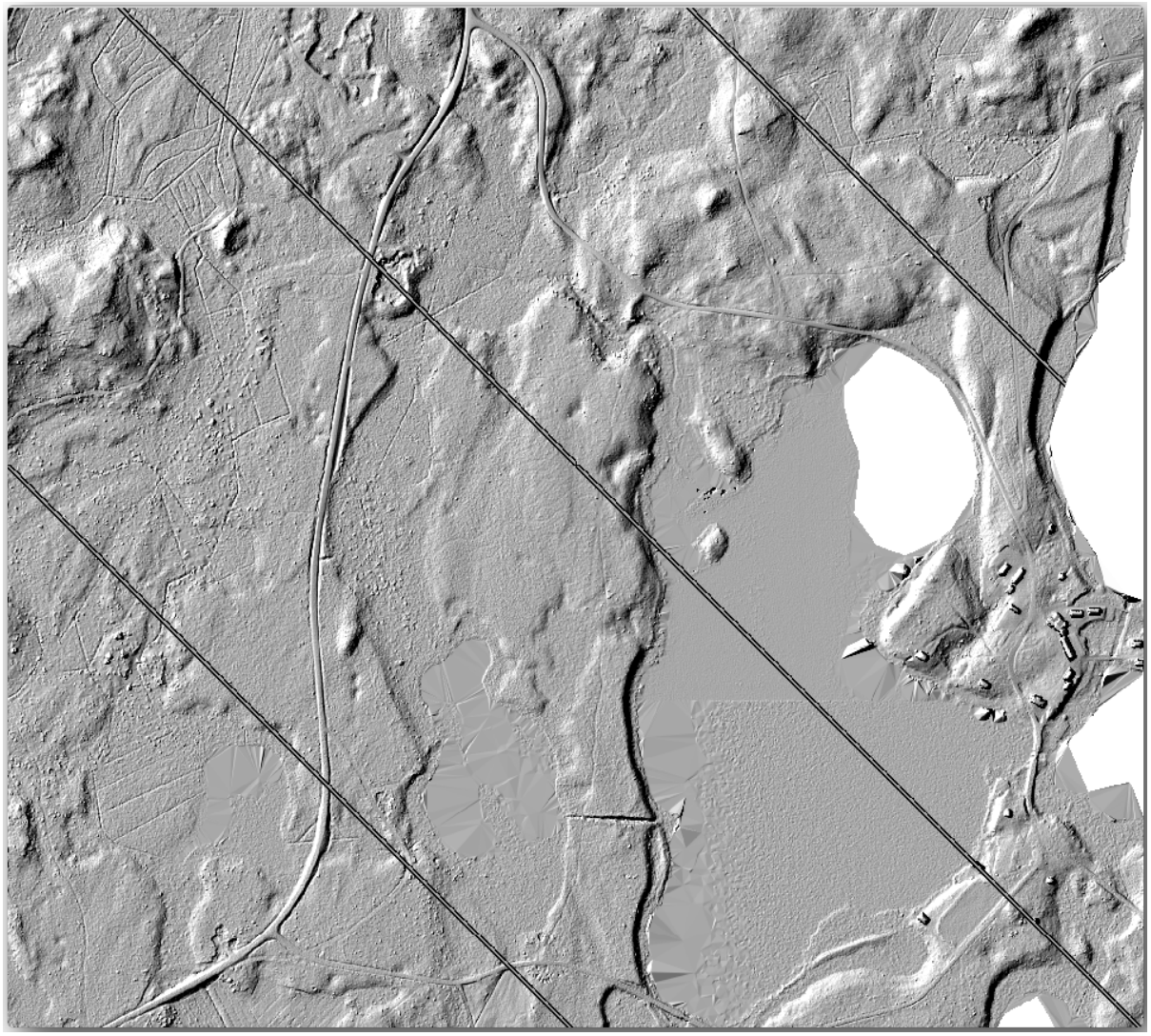
地形の可視化:

- **ラスタ** **地形分析** **陰影** を開きます。
- **出力レイヤ** には、`exercise_data\forestry\lidar\` を選択し `hillshade.tif` というファイル名にします。
- デフォルトの設定でパラメーターの残りの部分を残します。



- Select *ETRS89 / ETRS-TM35FIN* as the CRS when prompted.

陰影起伏ラスター結果に残っている対角線にもかかわらず、明らかに地域の正確な起伏を見ることができます。森に掘られている異なる土壌の排水も見ることができます。



14.8.4 結論

LiDAR データを使用して DEM を取得すると、特に森林地帯では、それほど労力をかけずに良い結果が得られます。また、LiDAR で取得した DEM や、SRTM 9m 解像度 DEM のような他のソースを使用することもできます。いずれにせよ、地図プレゼンテーションで使用する陰影起伏ラスタを作成するために、それらを使用することができます。

14.8.5 次は?

次の、そしてこのモジュールの最後のステップの、レッスンでは、陰影起伏ラスターと森林目録の結果を使用して、結果の地図のプレゼンテーションを作成します。

14.9 レッスン: 地図プレゼンテーション

以前のレッスンでは、GIS プロジェクトとして古い森林調査をインポートし、それを現在の状況に更新し、森林調査を設計し、フィールドワークの地図を作成し、フィールド測定から森林のパラメーターを算出しました。

GIS プロジェクトの結果で地図を作成することはしばしば重要です。森林調査の結果を提示する地図によって、特定の数字を見なくても、結果が何であるか誰にでも一目でわかりやすくなるでしょう。

このレッスンの目標： 調査結果を提示する地図を、陰影起伏ラスターを背景に使用して作成します。

14.9.1 ★☆☆ （初級レベル） 理解しよう: 地図データを準備する

Open the QGIS project from the parameters calculations lesson, `forest_inventory.qgs`. Keep at least the following layers:

- `forest_stands_2012_results`.
- `basic_map`.
- `rautjarvi_aerial`.
- `lakes` (if you don't have it, add it from the `exercise_data\forestry\` folder).

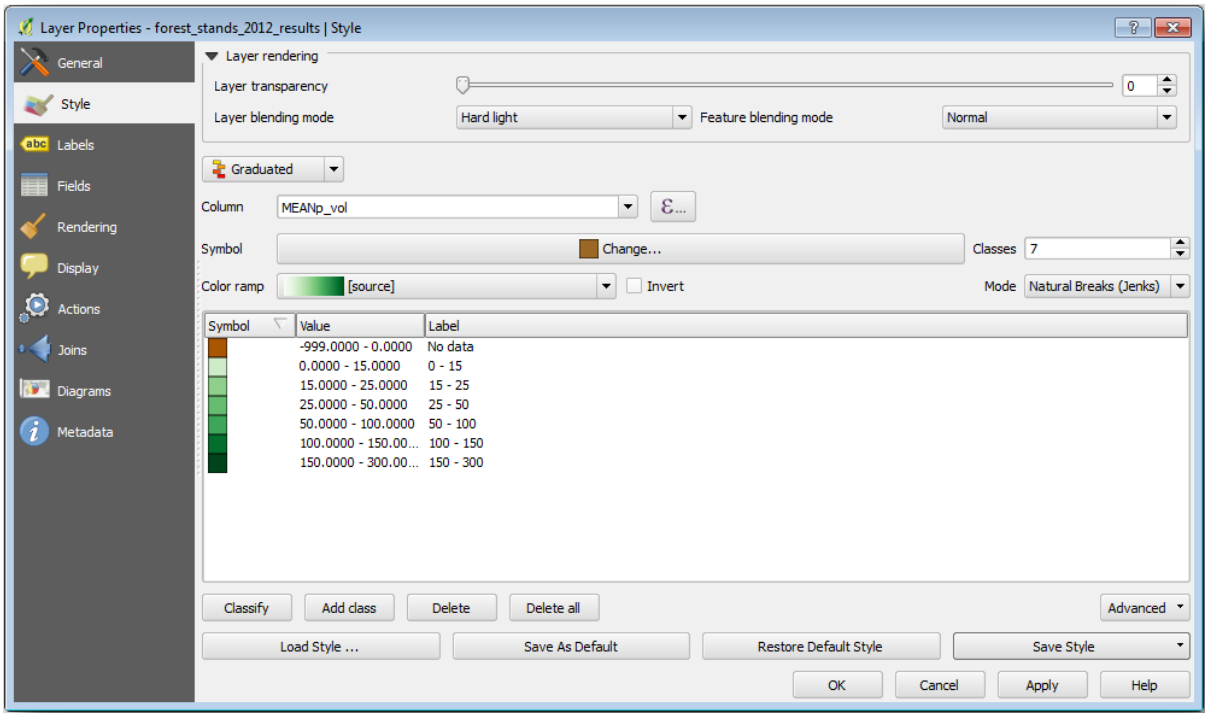
You are going to present the average volumes of your forest stands in a map. If you open the *Attribute table* for the `forest_stands_2012_results` layer, you can see the `NULL` values for the stands without information. To be able to get also those stands into your symbology you should change the `NULL` values to, for example, `-999`, knowing that those negative numbers mean there is no data for those polygons.

For the `forest_stands_2012_results` layer:

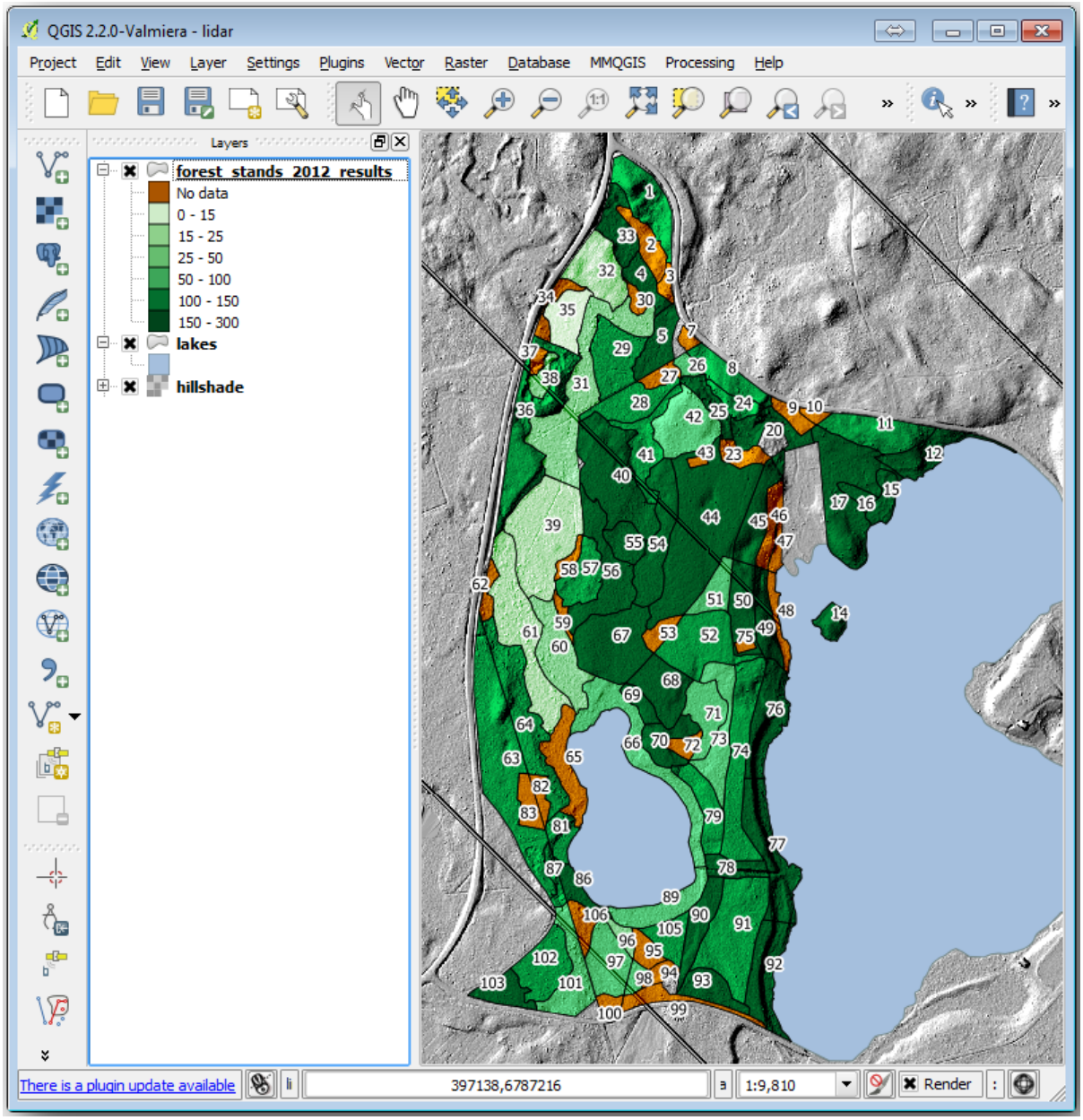
- **属性テーブル** を開き編集を可能にします。
- Select the polygons with `NULL` values.
- Use the calculator to update the values of the `MEANVol` field to `-999` only for the selected features.
- 編集を無効にし、変更を保存します。

今、保存されたスタイルをこのレイヤーに使用できます。

- **シンボロジ** タブに移動します。
- **スタイル　スタイルを読み込む**...。
- Select the `forest_stands_2012_results.qml` from the `exercise_data\forestry\results\` folder.
- *OK* をクリックします。

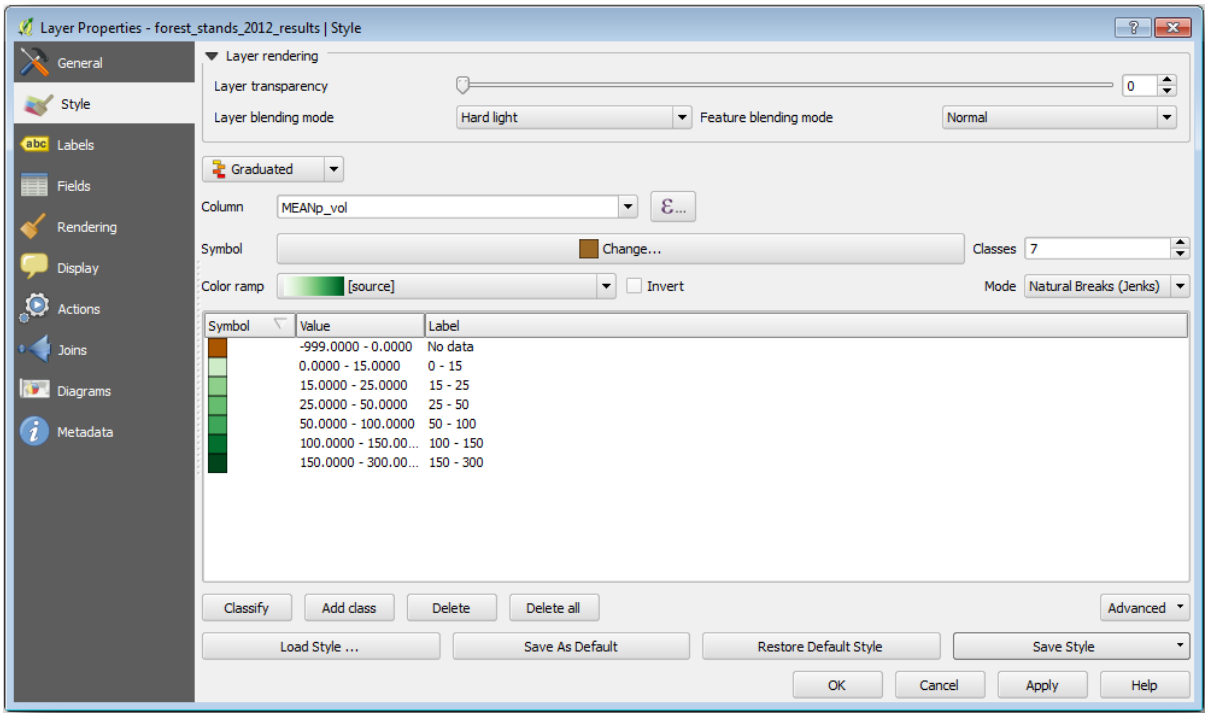


地図は次のようになります。



14.9.2 ★☆☆ （初級レベル） 自分でやってみよう: さまざまなブレンドモードを試す

読み込んだスタイル:

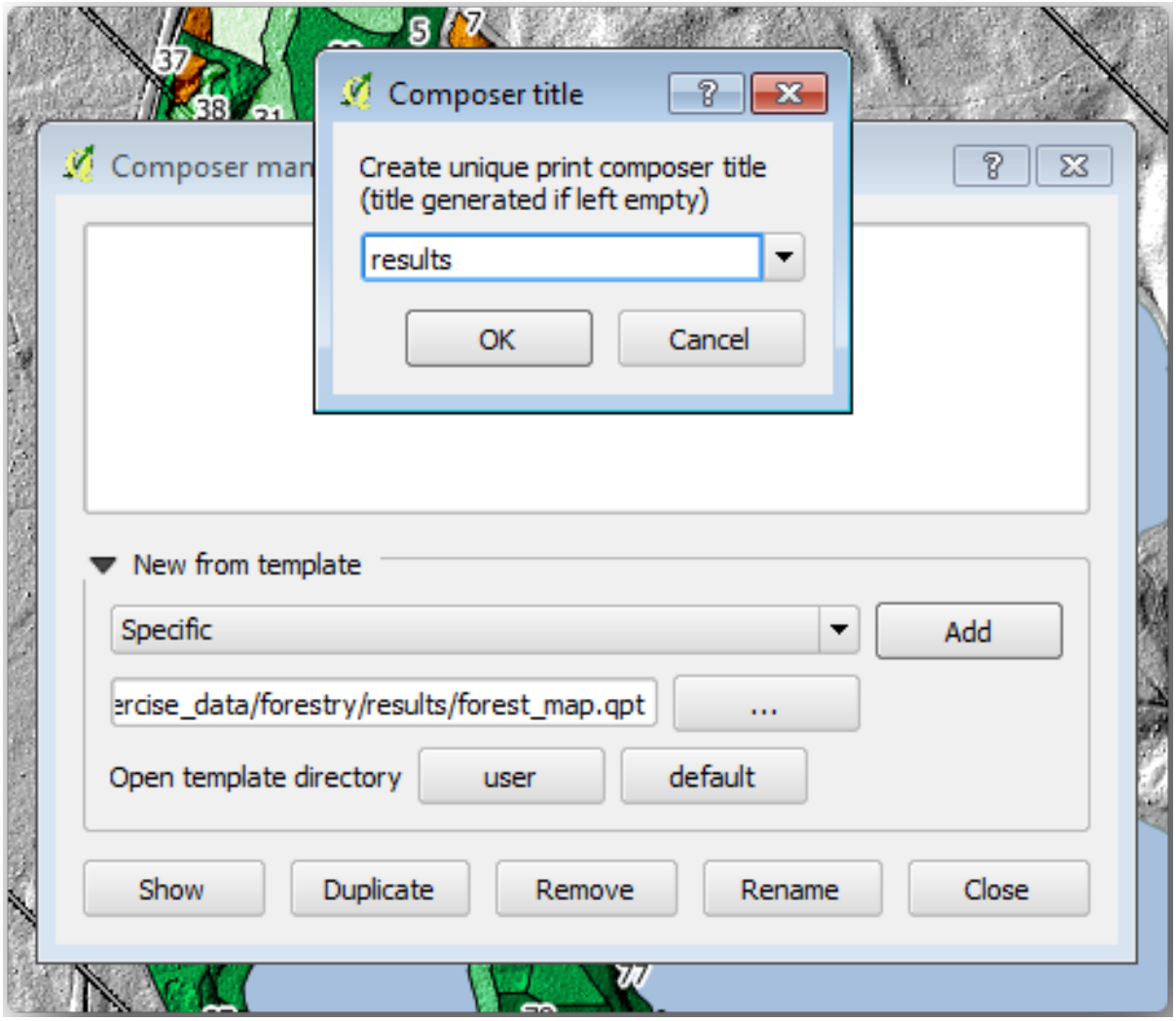


is using the *Hard light* mode for the *Layer blending mode*. Note that the different modes apply different filters combining the underlying and overlying layers, in this case the hillshade raster and your forest stands are used. You can read about these modes in the User Guide.

異なるモードで試してみて、地図の違いを見てください。それから、より好きな方を最終的な地図として選択してください。

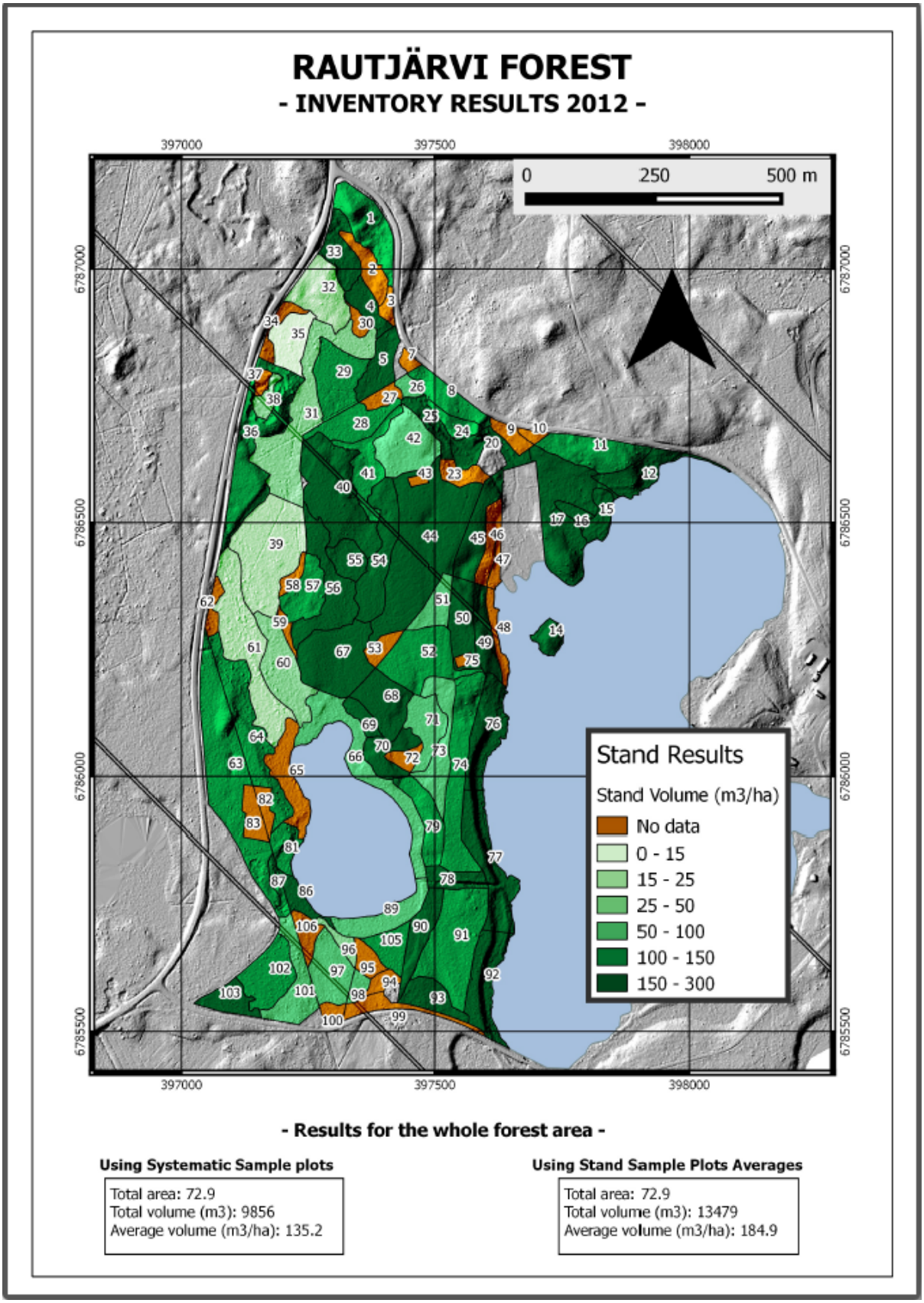
14.9.3 ★☆☆ （初級レベル） 自分でやってみよう: レイアウトテンプレートを使って地図の結果を作る

Use a template prepared in advanced to present the results. The template `forest_map.qpt` is located in the `exercise_data\forestry\results\` folder. Load it using the *Project Layout Manager...* dialog.



印刷レイアウトを開き、最終的なマップを編集して納得のいく仕上がりにします。

使用している地図テンプレートは、これと同じ地図を提供します：



将来の参照のため QGIS プロジェクトを保存します。

14.9.4 結論

このモジュールを通して、基本的な森林調査を計画し、QGIS で提示する方法を見てきました。アクセスできるさまざまなツールでさらに多くの森林の分析が可能ですが、このマニュアルがご自身が必要とする結果を達成する方法を探るための良い出発点になればと思います。

第15章　モジュール: PostgreSQLでのデータベース概念

リレーショナル・データベースは GIS システムの重要な部分です。リレーショナルデータベース管理システム（RDBMS）の概念について学び、データを格納するための新しいデータベースを作成するために PostgreSQL を使用してだけでなく、他の一般的な RDBMS の機能について学習します。

15.1 レッスン: データベースの概要

PostgreSQL のを使用する前に、一般的なデータベース理論をさらうことによって私たちの根拠を確認してみましょう。サンプルコードはどれも入力する必要はありません。それは説明目的のためだけにあります。

このレッスンの目標： 基本的なデータベースの概念を理解します。

15.1.1 データベースとは何ですか？

データベースは、典型的にはデジタル形式の、1 つ以上の用途のための組織化されたデータの集合からなる。 -**ウィキペディア**

データベース管理システム（DBMS）は、データベースを操作し、ストレージ、アクセス、セキュリティ、バックアップなどの機能を提供するソフトウェアで構成されています。 -**ウィキペディア**

15.1.2 表

リレーショナルデータベースとフラットファイルデータベースにおいてテーブルは、（名前で識別される）縦の列と横の行のモデルを使って構成されたデータ要素（値）の集合です。テーブルの列の数は指定されますが、行の数は任意です。各行は、特定の列の部分集合に現れる候補キーとして識別された値によって識別されます。 -**ウィキペディア**

```
 id | name  | age
----+-------+-----
  1 | Tim   |  20
  2 | Horst |  88
(2 rows)
```

SQL データベースではテーブルは **関係** としても知られています。

15.1.3 列/フィールド

列とは、特定の単純型のデータ値の集合であり、テーブルの各行に対して 1 つずつ存在します。列は、行を構成するための構造を提供します。フィールドという用語はしばしば列と互換的に使われますが、1 つの行と 1 つの列の交点に存在するひとつの項目を指すときは、フィールド（またはフィールド値）を使う方が正しいと考える人も多い。 -**ウィキペディア**

列：

```
| name  |
+-------+
| Tim   |
| Horst |
```

フィールド：

```
| Horst |
```

15.1.4 レコード

レコードは、テーブル行に格納されている情報です。各レコードには、テーブル内の各列のフィールドがあります。

```
2 | Horst |  88  <-- one record
```

15.1.5 データ型

データ型は、列に格納できる情報の種類を制限します。* - ティムとホルスト*

データ型には多くの種類があります。最も一般的なものに焦点を当ててみましょう：

- `String` - 自由形式のテキストデータを格納します
- `Integer` - 整数を格納します
- `Real` - 小数を保存します
- `Date` - 誰も忘れないよう、ホルストの誕生日を格納します
- `Boolean` - 単純な真/偽の値を格納します

フィールドに何も保存しないようにデータベースに指示することができます。フィールドに何もない場合、フィールドの中身は **'null' 値** と呼ばれます:

```
insert into person (age) values (40);

select * from person;
```

結果:

```
id | name  | age
---+-------+-----
 1 | Tim   |  20
 2 | Horst |  88
 4 |       |  40  <-- null for name
(3 rows)
```

使えるデータ型には更に多くがありますー PostgreSQL マニュアルを確認してください

15.1.6 住所データベースをモデル化

データベースが構築されるかを確認するために、単純なケーススタディを使ってみましょう。住所のデータベースを作成したいとします。

★☆☆ （初級レベル） 自分でやってみよう:

簡単な住所を構成し、そしてデータベースに格納される、プロパティを書き出します。

答え

私たちの理論上の住所テーブルの場合、次のようなプロパティを保存しておくとよいでしょう:

```
House Number
Street Name
Suburb Name
City Name
Postcode
Country
```

住所オブジェクトを表すテーブルを作成するとき、これらのプロパティのそれぞれを表す列を作成し、SQL に準拠したできるだけ短い名前を付けます:

```
house_number
street_name
suburb
city
postcode
country
```

住所の構造

住所を記述するプロパティは列です。各列に格納される情報のタイプは、そのデータ型です。次のセクションでは、概念的な住所テーブルを分析して、それをより良くする方法を見てみましょう。

15.1.7 データベース理論

データベースを作成するプロセスには、現実世界のモデルを作成することが含まれます。現実世界の概念を取り入れ、エンティティとしてデータベースに表現します。

15.1.8 正規化

データベースの主なアイデアの 1 つは、データの重複/冗長性を避けることです。データベースから冗長性を除去するプロセスを正規化といいます。

正規化は、データベース構造が汎用的な照会に適しており、挿入、更新、および削除の異常（データの整合性が失われる可能性がある）などの望ましくない特性がないことを確実にする体系的な方法です。 -**ウィキペディア**

正規「形」には様々な種類があります。

簡単な例を見てみましょう:

```
Table "public.people"

Column    |          Type          |                Modifiers
----------+------------------------+-----------------------------------
 id       | integer                | not null default
          |                        | nextval('people_id_seq'::regclass)
          |                        |
 name     | character varying(50)  |
 address  | character varying(200) | not null
 phone_no | character varying      |
Indexes:
 "people_pkey" PRIMARY KEY, btree (id)
```

```
select * from people;

id |     name      |           address            |  phone_no
---+---------------+------------------------------+-------------
 1 | Tim Sutton    | 3 Buirski Plein, Swellendam | 071 123 123
 2 | Horst Duester | 4 Avenue du Roix, Geneva     | 072 121 122
(2 rows)
```

友達が同じ名前の通りや都市に多いとしましょう。このデータが複製されるたびに、領域が消費されます。さらに悪いことに、都市の名称が変わった場合は、データベースを更新するために多くの作業が必要になります。

15.1.9 ★☆☆ （初級レベル） 自分でやってみよう:

Redesign the theoretical `people` table above to reduce duplication and to normalise the data structure.

データベースの正規化については ここ に読み物があります

答え

The major problem with the `people` table is that there is a single address field which contains a person's entire address. Thinking about our theoretical `address` table earlier in this lesson, we know that an address is made up of many different properties. By storing all these properties in one field, we make it much harder to update and query our data. We therefore need to split the address field into the various properties. This would give us a table which has the following structure:

```
id |     name      | house_no |  street_name   |    city    |   phone_no
 --+---------------+----------+----------------+------------+-----------------
 1 | Tim Sutton    |      3   | Buirski Plein  | Swellendam | 071 123 123
 2 | Horst Duester |      4   | Avenue du Roix | Geneva     | 072 121 122
```

次のセクションでは、データベースの構造をさらに改善するために、この例で使用することができる外部キー関係について学びます。

15.1.10 索引

データベース索引は、データベース表のデータ検索操作の速度を向上させるデータ構造です。 -**ウィキペディア**

たとえば教科書を読んで、ある概念の説明を探しているが、その教科書には索引がなかったとします。表紙から読み始め、必要な情報が見つかるまで、本全体を通して作業を進めなければなりません。教科書の裏にある索引は、関連情報を持つページに素早くジャンプするのに役立ちます:

```
create index person_name_idx on people (name);
```

名前の検索が高速になります:

```
Table "public.people"

Column   |          Type          |                Modifiers
----------+------------------------+-----------------------------------
 id       | integer                | not null default
          |                        | nextval('people_id_seq'::regclass)
          |                        |
```

(次のページに続く)

(前のページからの続き)

```
 name     | character varying(50)  |
 address  | character varying(200) | not null
 phone_no | character varying      |
Indexes:
 "people_pkey" PRIMARY KEY, btree (id)
 "person_name_idx" btree (name)
```

15.1.11 シーケンス

シーケンスは、一意の番号ジェネレータです。通常、テーブル内の列の一意の識別子を作成するために使用されます。

この例では、ID はシーケンスで、その数はレコードがテーブルに追加されるたびに 1 つ増えます:

```
id |     name     |           address           |  phone_no
---+--------------+-----------------------------+-------------
 1 | Tim Sutton   | 3 Buirski Plein, Swellendam | 071 123 123
 2 | Horst Duster | 4 Avenue du Roix, Geneva    | 072 121 122
```

15.1.12 エンティティ・リレーションシップ図の作成

In a normalised database, you typically have many relations (tables). The entity-relationship diagram (ER Diagram) is used to design the logical dependencies between the relations. Consider our non-normalised `people` table from earlier in the lesson:

```
select * from people;

 id |     name     |           address           |  phone_no
----+--------------+-----------------------------+-------------
 1  | Tim Sutton   | 3 Buirski Plein, Swellendam | 071 123 123
 2  | Horst Duster | 4 Avenue du Roix, Geneva    | 072 121 122
(2 rows)
```

ちょっとした作業で 2 つのテーブルに分割でき、同じ通りに住む人のために通りの名前を繰り返す必要がなくなります:

```
select * from streets;

 id |     name
----+--------------
 1  | Plein Street
(1 row)
```

および:

```
select * from people;

 id |     name      | house_no | street_id |  phone_no
----+---------------+----------+-----------+-------------
  1 | Horst Duster |        4 |         1 | 072 121 122
(1 row)
```

We can then link the two tables using the 'keys' `streets.id` and `people.streets_id`.

この 2 つのテーブルのための ER 図を描く場合は、次のようになります。



ER 図は、関係「一対多」を表現する助けになります。この場合、矢印記号は、1 つの通りに対して住んでいる人々は何人もいることがあると示しています。

★★☆ （中級レベル） 自分でやってみよう:

この *people* モデルにはまだいくつかの正規化の問題があります - さらに正規化して、ER 図を用いて自分の考えを示すことができるかどうか確認してみてください。

答え

Our `people` table currently looks like this:

```
id |     name      | house_no | street_id |  phone_no
---+---------------+----------+-----------+-------------
 1 | Horst Duster |        4 |         1 | 072 121 122
```

The *street_id* column represents a 'one to many' relationship between the people object and the related street object, which is in the `streets` table.

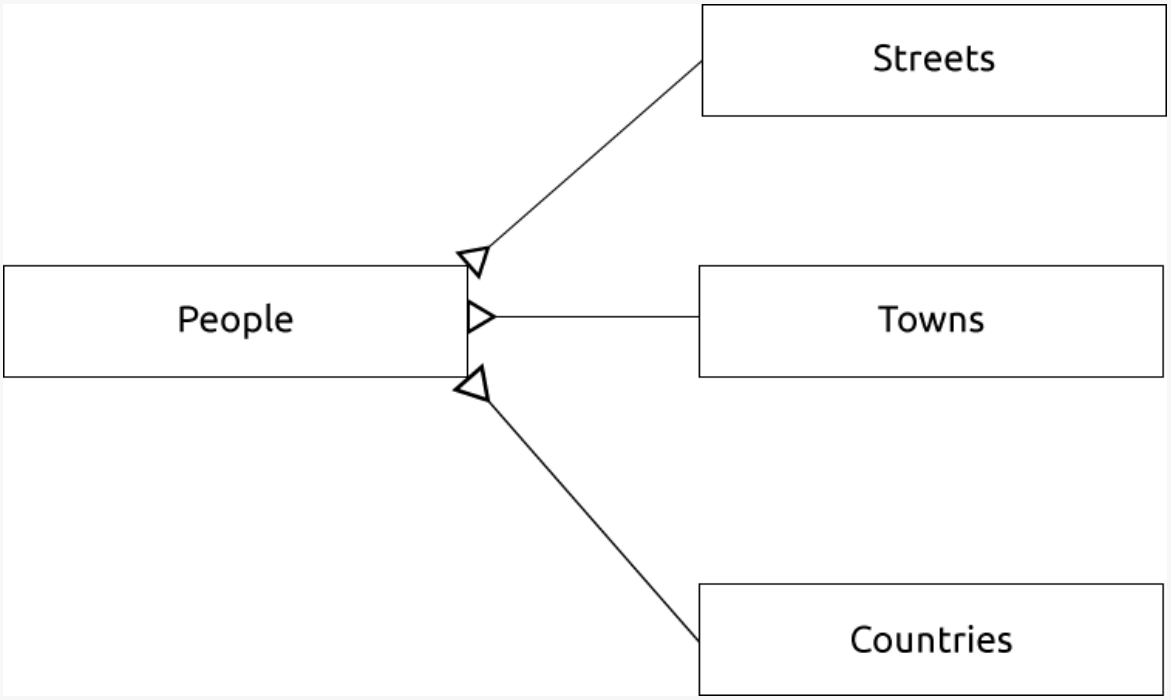
テーブルをさらに正規化する方法のひとつは、名前フィールドを *first_name* と *last_name* に分割することです:

```
id | first_name | last_name  | house_no | street_id |  phone_no
---+------------+------------+----------+-----------+-------------
 1 |    Horst   |   Duster   |     4    |     1     | 072 121 122
```

We can also create separate tables for the town or city name and country, linking them to our `people` table via 'one to many' relationships:

```
id | first_name | last_name | house_no | street_id | town_id | country_id
---+------------+-----------+----------+-----------+---------+------------
 1 |    Horst   |   Duster  |     4    |     1     |    2    |     1
```

これを表す ER 図は次のようになるでしょう:



15.1.13 制約、主キーと外部キー

A database constraint is used to ensure that data in a relation matches the modeller's view of how that data should be stored. For example a constraint on your postal code could ensure that the number falls between `1000` and `9999`.

主キーは、レコードを一意にする 1 つ以上のフィールドの値です。通常、主キーは id というシーケンスです。

外部キーは、（他のテーブルの主キーを使用して）別のテーブルに一意のレコードを参照するために使用されます。

ER 図では、テーブル間の結合は、通常、主キーにリンクする外部キーに基づいています。

ここでの people の例を見てみると、テーブルの定義によれば、street 列は streets テーブルの主キーを参照する外部キーです:

```
Table "public.people"

 Column    |         Type          |  Modifiers
-----------+-----------------------+-------------------------------------
 id        | integer               | not null default
           |                       | nextval('people_id_seq'::regclass)
 name      | character varying(50) |
 house_no  | integer               | not null
 street_id | integer               | not null
 phone_no  | character varying     |
```

(次のページに続く)

(前のページからの続き)

```
Indexes:
"people_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
"people_street_id_fkey" FOREIGN KEY (street_id) REFERENCES streets(id)
```

15.1.14 トランザクション

データベース内のデータを追加、変更、または削除するときは、何か問題が生じた場合にデータベースを良好な状態に保つことが常に重要です。ほとんどのデータベースは、トランザクションサポートと呼ばれる機能を提供します。トランザクションを使用すると、データベースへの変更が計画通りに実行されなかった場合に戻ることができるロールバック位置を作成できます。

会計システムを持っているというシナリオを取ります。1 つの口座から資金を転送し、他にそれらを追加する必要があります。一連のステップは次のように進むでしょう。

- Joe から R20 を削除
- Anne を R20 に追加

処理の間に何か問題（例えば停電）が発生した場合、トランザクションはロールバックされます。

15.1.15 結論

データベースを使用すると、簡単なコードの構造を使用して構造化された方法でデータを管理できます。

15.1.16 次は?

これでデータベースが理論的にどのように動作するか見てしまいましたので、カバーしてきた理論を実装する新しいデータベースを作成してみましょう。

15.2 レッスン: データモデルの実装

すべての理論をカバーしたところで新しいデータベースを作ってみましょう。このデータベースは後に続くレッスンの実習で使います。

このレッスンの目標: 必要なソフトウェアをインストールしてサンプルデータベースの実装に使用します。

15.2.1 PostgreSQL のインストール

> **注釈**
>
> お使いのオペレーティングシステムに対応した PostgreSQL のパッケージとインストール方法は、https://www.postgresql.org/download/ でご覧いただけます。なお、このドキュメントでは、ユーザーが Ubuntu で QGIS を実行していることを想定しています。

Ubuntu で:

```
sudo apt install postgresql-9.1
```

このようなメッセージを取得するはずです:

```
[sudo] password for qgis:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
postgresql-client-9.1 postgresql-client-common postgresql-common
Suggested packages:
oidentd ident-server postgresql-doc-9.1
The following NEW packages will be installed:
postgresql-9.1 postgresql-client-9.1 postgresql-client-common postgresql-common
0 upgraded, 4 newly installed, 0 to remove and 5 not upgraded.
Need to get 5,012kB of archives.
After this operation, 19.0MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Y と Enter キーを押し、ダウンロードとインストールが完了するまで待ちます。

15.2.2 ヘルプ

PostgreSQL にはとても良い オンライン ドキュメントがあります。

15.2.3 データベースユーザーの作成

Ubuntu で:

インストールが完了したらこのコマンドを実行して postgres ユーザーになり、新しいデータベースユーザーを作成します:

```
sudo su - postgres
```

入力を求められたら通常のログインパスワードを入力します (sudo 権限を持っている必要があります)。

では、postgres ユーザーでの bash プロンプトでデータベースユーザーを作成します。ユーザー名は unix ログイン名と一致させて下さい。そうするとログインする時に postgres が自動的に認証するのでいろいろと楽になります:

```
createuser -d -E -i -l -P -r -s qgis
```

入力を求められたらパスワードを入力します。ログインパスワードとは異なるパスワードを使用するべきです。

これらのオプションはどういう意味ですか?

```
-d, --createdb     role can create new databases
-E, --encrypted    encrypt stored password
-i, --inherit      role inherits privileges of roles it is a member of (default)
-l, --login        role can login (default)
-P, --pwprompt     assign a password to new role
-r, --createrole   role can create new roles
-s, --superuser    role will be superuser
```

今、入力することにより、postgres ユーザーの bash シェル環境を残す必要があります:

```
exit
```

15.2.4 新しいアカウントの確認

```
psql -l
```

このように返されるはずです:

```
Name      |  Owner   | Encoding | Collation  |   Ctype    |
----------+----------+----------+------------+------------+
postgres  | postgres | UTF8     | en_ZA.utf8 | en_ZA.utf8 |
template0 | postgres | UTF8     | en_ZA.utf8 | en_ZA.utf8 |
template1 | postgres | UTF8     | en_ZA.utf8 | en_ZA.utf8 |
(3 rows)
```

Q を入力して終了します。

15.2.5 データベースの作成

createdb コマンドは新しいデータベースを作成するのに使います。これは bash シェルプロンプトから実行しましょう:

```
createdb address -O qgis
```

このコマンドを使用して新しいデータベースの存在を確認できます:

```
psql -l
```

このように返されるはずです:

```
Name      |  Owner   | Encoding | Collation  |   Ctype    |   Access privileges
----------+----------+----------+------------+------------+-----------------------
address   | qgis     | UTF8     | en_ZA.utf8 | en_ZA.utf8 |
postgres  | postgres | UTF8     | en_ZA.utf8 | en_ZA.utf8 |
template0 | postgres | UTF8     | en_ZA.utf8 | en_ZA.utf8 | =c/postgres: postgres=CTc/
↪postgres
template1 | postgres | UTF8     | en_ZA.utf8 | en_ZA.utf8 | =c/postgres: postgres=CTc/
↪postgres
(4 rows)
```

Q を入力して終了します。

15.2.6 データベースのシェルセッションの開始

このようにして簡単にデータベースに接続することができます:

```
psql address
```

psql データベースシェルを終了するには:

```
\q
```

シェルのヘルプを見るには:

```
\?
```

sql コマンドのヘルプを見るには:

```
\help
```

特定のコマンドのヘルプを表示するには (例):

```
\help create table
```

Psql cheat sheet も参照してください。

15.2.7 SQL でテーブルを作成する

いくつかのテーブルを作ってみましょう! ガイドとして ER 図を使用します。まず、address（住所）データベースに接続します:

```
psql address
```

Then create a `streets` table:

```
create table streets (id serial not null primary key, name varchar(50));
```

`serial` and `varchar` are **data types**. `serial` tells PostgreSQL to start an integer sequence (auto-number) to populate the `id` automatically for every new record. `varchar(50)` tells PostgreSQL to create a character field of 50 characters in length.

You will notice that the command ends with a `;` - all SQL commands should be terminated this way. When you press `Enter`, psql will report something like this:

```
NOTICE:  CREATE TABLE will create implicit sequence "streets_id_seq"
         for serial column "streets.id"
NOTICE:  CREATE TABLE / PRIMARY KEY will create implicit index
         "streets_pkey" for table "streets"
CREATE TABLE
```

That means your table was created successfully, with a primary key `streets_pkey` using `streets.id`.

Note: If you hit return without entering a `;`, then you will get a prompt like this: `address-#`. This is because PostgreSQL is expecting you to enter more. Enter `;` to run your command.

テーブルのスキーマを表示するにはこうします:

```
\d streets
```

このように表示されるはずです:

```
Table "public.streets"
Column  |         Type          |            Modifiers
--------+-----------------------+-------------------------------------
 id     | integer               | not null default
        |                       | nextval('streets_id_seq'::regclass)
 name   | character varying(50) |
Indexes:
  "streets_pkey" PRIMARY KEY, btree (id)
```

テーブルの内容を表示するにはこうします:

```
select * from streets;
```

このように表示されるはずです:

```
id | name
---+------
(0 rows)
```

ご覧のようにテーブルは現在空です。

やってみよう: ★★☆ （中級レベル）

上記のアプローチを使用して people（人々）というテーブルを作成します:

電話番号、自宅住所、名前などのフィールドを追加します。上記と同じデータ型の ID 列も作ったか確認して下さい。

答え

正しい people テーブルを作る SQL は次の通り:

```
create table people (id serial not null primary key,
                     name varchar(50),
                     house_no int not null,
                     street_id int not null,
                     phone_no varchar null );
```

テーブルのスキーマ（\\d people を入力）は次のようなものです:

```
Table "public.people"

Column     |         Type          |                     Modifiers
-----------+-----------------------+-------------------------------------
id         | integer               | not null default
           |                       | nextval('people_id_seq'::regclass)
name       | character varying(50) |
house_no   | integer               | not null
street_id  | integer               | not null
phone_no   | character varying     |
Indexes:
  "people_pkey" PRIMARY KEY, btree (id)
```

説明のために、あえて fkey 制約を省略しています。

15.2.8 SQL でキーを作成する

上記のソリューションの問題はデータベースが people と streets に論理的な関係があることを知らないことです。この関係を表現するには、streets テーブルの主キーを指す外部キーを定義する必要があります。



これを行うには 2 つの方法があります:

- テーブル作成後にキーを追加する
- テーブル作成時にキーを定義する

テーブルは既に作成されているので最初の方法を採ります:

```
alter table people
  add constraint people_streets_fk foreign key (street_id) references streets(id);
```

That tells the `people` table that its `street_id` fields must match a valid street `id` from the `streets` table.

より一般的には制約の作成はテーブルの作成時に行います:

```
create table people (id serial not null primary key,
                     name varchar(50),
                     house_no int not null,
                     street_id int references streets(id) not null,
                     phone_no varchar null);

\d people
```

制約を追加した後、テーブルのスキーマはこのようになります:

```
Table "public.people"

  Column   |         Type          |            Modifiers
-----------+-----------------------+--------------------------------
 id        | integer               | not null default
           |                       | nextval('people_id_seq'::regclass)
 name      | character varying(50) |
 house_no  | integer               | not null
 street_id | integer               | not null
 phone_no  | character varying     |
Indexes:
  "people_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
  "people_streets_fk" FOREIGN KEY (id) REFERENCES streets(id)
```

15.2.9 SQL でインデックスを作成する

人の名前をすばやく検索できるようにするには people（人々）テーブルの name（名前）列にインデックスを作成します:

```
create index people_name_idx on people(name);

\d people
```

その結果:

```
Table "public.people"

  Column   |         Type          |                     Modifiers
-----------+-----------------------+-----------------------------------
 id        | integer               | not null default nextval
           |                       | ('people_id_seq'::regclass)
 name      | character varying(50) |
 house_no  | integer               | not null
 street_id | integer               | not null
 phone_no  | character varying     |
Indexes:
 "people_pkey" PRIMARY KEY, btree (id)
 "people_name_idx" btree (name)    <-- new index added!
Foreign-key constraints:
 "people_streets_fk" FOREIGN KEY (id) REFERENCES streets(id)
```

15.2.10 SQL でテーブルを削除する

If you want to get rid of a table you can use the DROP command:

```
drop table streets;
```

現在の例では、上記のコマンドは機能しません。なぜダメなのでしょう？

答え

The reason the DROP command would not work in this case is because the `people` table has a Foreign Key constraint to the `streets` table. This means that dropping (or deleting) the `streets` table would leave the `people` table with references to non-existent `streets` data.

It is possible to 'force' the `streets` table to be deleted by using the `CASCADE` command, but this would also delete the `people` and any other table which had a relationship to the `streets` table. Use with caution!

If you used the same `drop table` command on the `people` table, it would be successful:

```
drop table people;
```

> **注釈**
>
> If you actually did enter that command and dropped the `people` table, now would be a good time to rebuild it, as you will need it in the next exercises.

15.2.11 pgAdmin III について一言

データベースについて学ぶために非常に有効な方法なので *psql* プロンプトから SQL コマンドを入力しています。しかし、より早くより簡単に行う方法があります。pgAdmin III をインストールすると GUI 上のクリック操作でテーブルの create, drop, alter 等を行うことができます。

Ubuntu ではこのようにインストールします:

```
sudo apt install pgadmin3
```

pgAdmin III は別のモジュールで詳しく取り上げます。

15.2.12 結論

真新しいデータベースを完全にゼロから作成する方法を見てきました。

15.2.13 次は?

次は DBMS を使用して新しいデータを追加する方法を学びます。

15.3 レッスン: モデルにデータを追加する

作成したモデルには、今、含まれることを意図されるデータが投入される必要があります。

このレッスンの目標： データベースモデルに新しいデータを挿入する方法を学習します。

15.3.1 insert 文

How do you add data to a table? The sql INSERT statement provides the functionality for this:

```
insert into streets (name) values ('High street');
```

注意すべきいくつかの事：

- After the table name (streets), you list the column names that you will be populating (in this case only the name column).
- After the values keyword, place the list of field values.
- 文字列は単一引用符で囲む必要があります。
- Note that we did not insert a value for the id column; this is because it is a sequence and will be auto-generated.
- If you do manually set the id, you may cause serious problems with the integrity of your database.

You should see INSERT 0 1 if it is successful.

テーブル内のすべてのデータを選択して、挿入アクションの結果を見ることができます:

```
select * from streets;
```

結果:

```
select * from streets;
 id |    name
----+-------------
  1 | High street
(1 row)
```

やってみよう: ★☆☆ （初級レベル）

Use the INSERT command to add a new street to the streets table.

答え

使うべき SQL コマンドはこのようなものです（街路の名称はあなたが選んだもので置き換えることができます）：

```
insert into streets (name) values ('Low Road');
```

15.3.2 制約に従ってデータの追加を順序付けする

15.3.3 やってみよう: ★★☆ （中級レベル）

Try to add a person object to the `people` table with the following details:

```
Name: Joe Smith
House Number: 55
Street: Main Street
Phone: 072 882 33 21
```

注釈

この例では、文字列ではなく整数として電話番号を定義したことを思い出してください。

At this point, you should have an error report if you try to do this without first creating a record for Main Street in the `streets` table.

以下のことに気づいたはずです：

- その名前を使用して街路を追加できません
- You can't add a street using a street `id` before first creating the street record on the `streets` table

2 つのテーブルが主キー/外部キーのペアを介して結合していることに注意してください。これは、有効な人は有効な対応する街路レコードも存在していなければ作成できないことを意味します。

上記の知識を使用して、データベースに新しい人を追加します。

答え

正しい SQL 文は次のとおりです:

```
insert into streets (name) values('Main Road');
insert into people (name,house_no, street_id, phone_no)
  values ('Joe Smith',55,2,'072 882 33 21');
```

（先ほどと同じように select 文を使って）もう一度街路テーブルを見てみると、`Main Road` のエントリの *id* が *2* であることがわかるでしょう。

そのため、上記のように単に数字の `2` を入力することができます。上のエントリで `Main Road` が完全に書き込まれていなくても、データベースはそれを *street_id* の値 *2* と関連付けることができるのです。

すでに新しい街路オブジェクトを追加している場合、その新しい `Main Road` は *2* ではなく *3*の *id* を持っていることに気づくでしょう。

15.3.4 データを選択

レコードを選択するための構文はすでに示しました。さらにいくつかの例を見てみましょう:

```
select name from streets;
```

```
select * from streets;
```

```
select * from streets where name='Main Road';
```

後のセッションでは、データを選択してフィルタリングする方法について詳細に見て行きます。

15.3.5 データを更新

何か既存のデータに変更を加えたい場合は？例えば、街路名が変更されます:

```
update streets set name='New Main Road' where name='Main Road';
```

Be very careful using such update statements - if more than one record matches your `WHERE` clause, they will all be updated!

よりよい解決策は、テーブルの主キーを使用して変更するレコードを参照することです:

```
update streets set name='New Main Road' where id=2;
```

It should return `UPDATE 1`.

> **注釈**
>
> The `WHERE` statement criteria are case sensitive, `Main Road` is not the same as `Main road`.

15.3.6 データを削除

In order to delete an object from a table, use the `DELETE` command:

```
delete from people where name = 'Joe Smith';
```

今度は人々のテーブルを見てみましょう:

```
address=# select * from people;

 id | name | house_no | street_id | phone_no
----+------+----------+-----------+----------
(0 rows)
```

15.3.7 やってみよう: ★★★ （上級レベル）

学んだスキルを使用して、データベースに新しい友達を何人か追加してください:

```
      name       | house_no | street_id |    phone_no
-----------------+----------+-----------+--------------
Joe Bloggs       |        3 |         2 | 072 887 23 45
Jane Smith       |       55 |         3 | 072 837 33 35
Roger Jones      |       33 |         1 | 072 832 31 38
Sally Norman     |       83 |         1 | 072 932 31 32
```

15.3.8 結論

以前に作成した既存のモデルに新しいデータを追加する方法がわかりましたね。データの新しい種類を追加したい場合は、そのデータを格納する新しいモデルを変更かつ／または作成したいこともあることを忘れないでください。

15.3.9 次は?

データを追加してしまったので、クエリを使用してさまざまな方法でこのデータにアクセスする方法を学びましょう。

15.4 レッスン: クエリ

When you write a `SELECT ...` command it is commonly known as a query - you are interrogating the database for information.

このレッスンの目的： 有用な情報を返すクエリを作成する方法を学習します。

> **注釈**
>
> If you did not do so in the previous lesson, add the following people objects to your `people` table. If you receive any errors related to foreign key constraints, you will need to add the 'Main Road' object to your streets table first

```
insert into people (name,house_no, street_id, phone_no)
          values ('Joe Bloggs',3,2,'072 887 23 45');
insert into people (name,house_no, street_id, phone_no)
          values ('Jane Smith',55,3,'072 837 33 35');
insert into people (name,house_no, street_id, phone_no)
```

(次のページに続く)

(前のページからの続き)

```
          values ('Roger Jones',33,1,'072 832 31 38');
insert into people (name,house_no, street_id, phone_no)
          values ('Sally Norman',83,1,'072 932 31 32');
```

15.4.1 結果を並べ替える

自分の家の番号順に並べられた人々のリストを検索してみましょう:

```
select name, house_no from people order by house_no;
```

結果:

```
     name      | house_no
---------------+----------
 Joe Bloggs    |        3
 Roger Jones   |       33
 Jane Smith    |       55
 Sally Norman  |       83
(4 rows)
```

結果の並べ替えは複数の列の値によってもできます:

```
select name, house_no from people order by name, house_no;
```

結果:

```
     name      | house_no
---------------+----------
 Jane Smith    |       55
 Joe Bloggs    |        3
 Roger Jones   |       33
 Sally Norman  |       83
(4 rows)
```

15.4.2 フィルタリング

だいたいはデータベース内のすべてのレコード一つ一つを見たいとは思わないでしょう。特に何千ものレコードがあり、1 つか 2 つを見たいだけの場合は 。

これは house_no が 50 未満であるオブジェクトのみを返す数値フィルタの例です:

```
select name, house_no from people where house_no < 50;
```

(次のページに続く)

(前のページからの続き)

```
     name     | house_no
--------------+----------
 Joe Bloggs   |        3
 Roger Jones  |       33
(2 rows)
```

フィルタ（WHERE 句を使用して定義される）はソート（ORDER BY 句を使用して定義される）と組み合わせることができます:

```
select name, house_no from people where house_no < 50 order by house_no;

     name     | house_no
--------------+----------
 Joe Bloggs   |        3
 Roger Jones  |       33
(2 rows)
```

テキストデータに基づいてもフィルタできます:

```
select name, house_no from people where name like '%s%';

     name     | house_no
--------------+----------
 Joe Bloggs   |        3
 Roger Jones  |       33
(2 rows)
```

Here we used the LIKE clause to find all names with an s in them. You'll notice that this query is case-sensitive, so the Sally Norman entry has not been returned.

If you want to search for a string of letters regardless of case, you can do a case in-sensitive search using the ILIKE clause:

```
select name, house_no from people where name ilike '%r%';

      name     | house_no
---------------+----------
 Roger Jones   |       33
 Sally Norman  |       83
(2 rows)
```

That query returned every **people** object with an r or R in their name.

15.4.3 結合

ID の代わりに人の詳細とその通りの名前を確認したい場合は？そのためには、単一のクエリ中で 2 つのテーブルを結合する必要があります。例を見てみましょう:

```
select people.name, house_no, streets.name
from people,streets
where people.street_id=streets.id;
```

注釈

結合によって、情報が由来する 2 つのテーブルを、この場合は people と streets ですが、常に宣言することになります。また、どの 2 つのキー（外部キー＆主キー）が一致しなければならないかを指定する必要があります。それを指定しない場合は、people と streets のすべての可能な組み合わせの一覧が得られますが、誰が実際にその通りに住んでいるか知るすべはありません！

正しい出力はこのように見えるでしょう:

```
    name     | house_no |    name
-------------+----------+-------------
 Joe Bloggs  |        3 | Low Street
 Roger Jones |       33 | High street
 Sally Norman|       83 | High street
 Jane Smith  |       55 | Main Road
(4 rows)
```

結合については、後でより複雑なクエリを作成するときに再訪します。それらは二つ以上のテーブルからの情報を組合わせるための簡単な方法を提供するとだけ覚えておいてください。

15.4.4 副選択

副選択は外部キー関係を介して連結されている別のテーブルからのデータに基づいて一つのテーブルからオブジェクトを選択できます。この場合は特定の街路に住む人々を見つけたいです。

まず、データをわずかに微調整しましょう:

```
insert into streets (name) values('QGIS Road');
insert into streets (name) values('OGR Corner');
insert into streets (name) values('Goodle Square');
update people set street_id = 2 where id=2;
update people set street_id = 3 where id=3;
```

それらの変更の後でデータを簡単に見てみましょう：前のセクションのクエリを再利用できます:

```
select people.name, house_no, streets.name
from people,streets
where people.street_id=streets.id;
```

結果:

```
     name     | house_no |    name
--------------+----------+-------------
 Roger Jones  |       33 | High street
 Sally Norman |       83 | High street
 Jane Smith   |       55 | Main Road
 Joe Bloggs   |        3 | Low Street
(4 rows)
```

Now let's show you a sub-selection on this data. We want to show only people who live in `street_id` number 1:

```
select people.name
from people, (
    select *
    from streets
    where id=1
  ) as streets_subset
where people.street_id = streets_subset.id;
```

結果:

```
     name
--------------
 Roger Jones
 Sally Norman
(2 rows)
```

これは非常に単純な例でありこの小さなデータセットでは不要ですが、大規模かつ複雑なデータセットを照会する際に有用かつ重要な副選択をする方法を示しています。

15.4.5 クエリの集約

データベースの強力な機能の 1 つは、そのテーブル内のデータを要約する能力です。これらの要約は集計クエリと呼ばれます。これは、people オブジェクトが people テーブルに何人いるかを教えてくれる代表的な例です:

```
select count(*) from people;
```

結果:

```
 count
-------
     4
(1 row)
```

人数を街路名で要約したい場合は、こうすることができます:

```
select count(name), street_id
from people
group by street_id;
```

結果:

```
 count | street_id
-------+-----------
     2 |         1
     1 |         3
     1 |         2
(3 rows)
```

注釈

Because we have not used an `ORDER BY` clause, the order of your results may not match what is shown here.

やってみよう: ★★☆ （中級レベル）

通りの名前で人を要約し、street_ids の代わりに実際の通りの名前を表示してください。

答え

使うべき正しい SQL 文はこちらです:

```
select count(people.name), streets.name
from people, streets
where people.street_id=streets.id
group by streets.name;
```

結果:

```
count |    name
------+-------------
    1 | Low Street
    2 | High street
    1 | Main Road
(3 rows)
```

フィールド名の前にテーブル名を付けていることにお気づきでしょうか（people.name や street.name など）。これは、フィールド名があいまいな場合（つまり、データベース内のすべてのテーブルで一意でない場合）に使う必要があります。

15.4.6 結論

クエリを使用して、データベース内の有用な情報を抽出できるような形でデータを返す方法を見てきました。

15.4.7 次は?

次は今書いたクエリからビューを作成する方法について説明します。

15.5 レッスン: ビュー

クエリを記述するときはそれを考案するのに多くの時間と労力が必要です。ビューを使えば SQL クエリの定義を再利用可能な「仮想テーブル」に保存できます。

このレッスンの目標: クエリをビューとして保存します。

15.5.1 ビューの作成

ビューはテーブルのように扱うことができますが、そのデータはクエリから供給されます。上記に基いて単純なビューを作りましょう:

```
create view roads_count_v as
  select count(people.name), streets.name
  from people, streets where people.street_id=streets.id
  group by people.street_id, streets.name;
```

As you can see the only change is the `create view roads_count_v as` part at the beginning. We can now select data from that view:

```
select * from roads_count_v;
```

結果:

```
 count |    name
-------+-------------
     1 | Main Road
     2 | High street
```

(次のページに続く)

(前のページからの続き)

```
     1 | Low Street
(3 rows)
```

15.5.2 ビューの変更

ビューは固定されておらず、「実データ」を持ちません。つまりデータベースの中のデータに影響を与えることなく簡単に変更できます:

```
CREATE OR REPLACE VIEW roads_count_v AS
  SELECT count(people.name), streets.name
  FROM people, streets WHERE people.street_id=streets.id
  GROUP BY people.street_id, streets.name
  ORDER BY streets.name;
```

(また、この例はすべての SQL キーワードには大文字を使用する最良慣行を示しています。)

You will see that we have added an `ORDER BY` clause so that our view rows are nicely sorted:

```
select * from roads_count_v;

 count |     name
-------+-------------
     2 | High street
     1 | Low Street
     1 | Main Road
(3 rows)
```

15.5.3 ビューの削除

不要になったビューはこのように削除できます:

```
drop view roads_count_v;
```

15.5.4 結論

ビューを用いてクエリを保存し、テーブルであるかのようにその結果へアクセスできます。

15.5.5 次は?

データを変更する時に、変更がデータベースの中の他の場所へ影響を及ぼすのが望ましい場合があります。次のレッスンではこの方法を紹介します。

15.6 レッスン: ルール

ルールは "クエリ木" に書き換えることができます。一つの一般的な使用法は、更新可能なビューなど、ビューを実装することです。 *Wikipedia* **より**

このレッスンの目標: データベースの新しいルールを作成する方法を学習する.

15.6.1 ログに記録するルールを作る

Say you want to log every change of `phone_no` in your `people` table in to a `people_log` table. So you set up a new table:

```
create table people_log (name text, time timestamp default NOW());
```

In the next step, create a rule that logs every change of a `phone_no` in the `people` table into the `people_log` table:

```
create rule people_log as on update to people
  where NEW.phone_no <> OLD.phone_no
  do insert into people_log values (OLD.name);
```

ルールが正しく機能することを確認するには, 電話番号を変更してみましょう :

```
update people set phone_no = '082 555 1234' where id = 2;
```

Check that the `people` table was updated correctly:

```
select * from people where id=2;

 id |    name     | house_no | street_id |   phone_no
----+-------------+----------+-----------+--------------
  2 | Joe Bloggs |        3 |         2 | 082 555 1234
(1 row)
```

Now, thanks to the rule we created, the `people_log` table will look like this:

```
select * from people_log;

    name     |            time
-------------+----------------------------
```

(次のページに続く)

(前のページからの続き)

```
 Joe Bloggs | 2014-01-11 14:15:11.953141
(1 row)
```

注釈

The value of the `time` field will depend on the current date and time.

15.6.2 結論

ルールを使用すると、データベースの他の部分の変更を反映するために、自動的にデータベース内でデータを追加または変更できます。

15.6.3 次は?

The next module will introduce you to Spatial Database using PostgreSQL with PostGIS extension, which takes these database concepts and applies them to GIS data.

第16章　Module: Spatial Database Concepts with PostGIS / PostgreSQL

Spatial Databases allow the storage of the geometries of records inside a Database as well as providing functionality for querying and retrieving the records using these Geometries. In this module we will use PostGIS, an extension to PostgreSQL, to learn how to setup a spatial database, import data into the database and make use of the geographic functions that PostGIS offers.

このセクションで作業している間、 BostonGIS ユーザーグループ から入手できる **PostGIS チートシート** のコピーを保持することをお勧めします。もう 1 つの便利なリソースは、 オンライン PostGIS ドキュメントです。

また、Boundless が作成した PostGIS と空間データベースに関するいくつかのより広範なチュートリアルが、PostGIS ウェブサイトで公開されています:

- Introduction to PostGIS（PostGIS 入門）
- PostGIS Database Tips and Tricks（PostGIS データベースのヒントとコツ

PostGIS In Action も参照。

16.1 レッスン: PostGIS の設定

PostGIS の関数を設定することで, PostgreSQL の中から空間関数にアクセス可能になります.

このレッスンの目的: 空間関数をインストールし, それらの効果を簡単にデモする.

> **注釈**
>
> この演習では、PostGIS バージョン 2.1 以降を使用することを想定しています。古いバージョンでは、インストールとデータベース設定が異なりますが、このモジュールの残りの部分はそのまま使用できます。インストールとデータベース設定のヘルプについては、お使いのプラットフォームのドキュメントを参照してください。

16.1.1 Ubuntu でのインストール

PostGIS は apt から簡単にインストールできます.

```
$ sudo apt install postgresql
$ sudo apt install postgis
```

本当に簡単です...

> **注釈**
>
> インストールされる正確なバージョンは、使用している Ubuntu のバージョンと、設定したリポジトリに依存します。インストール後、psql や他のツールで `select PostGIS_full_version();` クエリを発行して、バージョンを確認することができます。

特定のバージョン（例えば、PostgreSQL バージョン 13 と PostGIS 3）をインストールする場合は、以下のコマンドを使用します。

```
$ sudo apt install wget ca-certificates
$ sudo lsb_release -a
$ wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key␣
↪add -
$ sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb_release -cs`-
↪pgdg main" >> /etc/apt/sources.list.d/pgdg.list'
$ sudo apt-get update
$ sudo apt install postgis postgresql-13-postgis-3
```

16.1.2 Windows でのインストール

Windows へのインストールは、バイナリパッケージから通常の Windows のインストールダイアログを使って行うことができます。

まず、 ダウンロードページ にアクセスします。次に、このガイド に従ってください。

Windows へのインストールに関する詳しい情報は、PostGIS ウェブサイト に掲載されています。

16.1.3 その他のプラットフォームへのインストール

PostGIS ウェブサイトのダウンロード には、macOS を含む他のプラットフォームや他の Linux ディストリビューションへのインストールについての情報があります

16.1.4 PostGIS を使うためにデータベースを設定する

一度 PostGIS がインストールされたら、拡張機能を使用するようにデータベースを設定する必要があります。PostGIS のバージョン> 2.0 をインストールしている場合、これが前の演習からアドレスデータベースを使用して psql で次のコマンドを発行するのと同じくらい簡単です。

```
$ psql -d address -c "CREATE EXTENSION postgis;"
```

注釈

バージョンによっては、データベースを空間的に有効にする方法について、https://postgis.net/docs/postgis_administration.html#create_spatial_db で多くの説明を見つけることができるでしょう。

16.1.5 インストールされた PostGIS 関数を見る

PostGIS は、空間データを扱えるように PostgreSQL のコア機能を拡張する、データベース機能の集まりと考えることができます。「扱える」とは、格納、検索、クエリおよび操作を意味します。これを行うために、多数の機能が、データベースにインストールされています。

私たちの PostgreSQL の address 住所 データベースは PostGIS のおかげで地理空間的に有効になりました。次のセクションではこれについてもっと深く掘り下げていきますが、ここでちょっと味見しましょう。テキストからポイントを作成したいとしましょう。最初に、psql コマンドを使ってポイントに関連する関数を見つけます。address 住所 データベースにまだ接続していない場合は、今すぐ実行してください。次に、

```
\df *point*
```

This is the command we're looking for: `st_pointfromtext`. To page through the list, use the down arrow, then press `Q` to quit back to the psql shell.

次のコマンドを実行してみてください：

```
select st_pointfromtext('POINT(1 1)');
```

結果:

```
st_pointfromtext
--------------------------------------------
0101000000000000000000F03F000000000000F03F
(1 row)
```

注目すべき 3 点:

- We defined a point at position 1,1 (EPSG:4326 is assumed) using `POINT(1 1)`,
- SQL 文を実行しましたが、どのテーブル上でもなく、SQL プロンプトから入力されたデータで,
- 結果の行はあまり意味がありません。

得られた行は、「よく知られているバイナリ」（WKB）と呼ばれる OGC フォーマットです。私たちは、次のセクションで詳細にこのフォーマットを見ていきます。

結果をテキストとして戻すために、テキストを返す何かを探して関数リストをすばやくスキャンできます：

```
\df *text
```

私たちが探しているクエリは st_astext です。以前のクエリと組み合わせてみましょう：

```
select st_astext(st_pointfromtext('POINT(1 1)'));
```

結果:

```
 st_astext
------------
  POINT(1 1)
  (1 row)
```

Here, we entered the string POINT(1,1), turned it into a point using st_pointfromtext(), and turned it back into a human-readable form with st_astext(), which gave us back our original string.

実際に PostGIS の使い方の詳細に入る前の最後の例は：

```
select st_astext(st_buffer(st_pointfromtext('POINT(1 1)'),1.0));
```

それは何をしましたか？それは私たちのポイントを中心に 1 度のバッファを作成し、テキストとして結果を返しました。

16.1.6 空間参照系

PostGIS の機能だけでなく、この拡張は欧州石油調査グループ（EPSG）によって定義された空間参照システム（SRS）の定義のコレクションを含んでいます。これらは、座標参照系（CRS）変換などの操作中に使用されます。

通常のデータベーステーブルに格納されているので、データベース中のこれらの SRS の定義を調べることができます。

まず、psql プロンプトで次のコマンドを入力して、テーブルのスキーマを見てみましょう。

```
\d spatial_ref_sys
```

結果はこうなります：

```
Table "public.spatial_ref_sys"
   Column   |          Type           | Modifiers
 -----------+-------------------------+-----------
  srid      | integer                 | not null
  auth_name | character varying(256)  |
```

(次のページに続く)

(前のページからの続き)

```
 auth_srid | integer                  |
 srtext    | character varying(2048) |
 proj4text | character varying(2048) |
 Indexes:
"spatial_ref_sys_pkey" PRIMARY KEY, btree (srid)
```

このテーブルを表示および操作するために、（入門セクションから学んだように）標準の SQL クエリを使用できます - 何をしているのか分からなければ、任意のレコードを更新または削除するのは良いアイデアではありません。

興味のある SRID は、EPSG：4326 - WGS 84 楕円体を使用した地理/緯度経度参照系です。それを見てみましょう：

```
select * from spatial_ref_sys where srid=4326;
```

結果:

```
srid      | 4326
auth_name | EPSG
auth_srid | 4326
srtext    | GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS
84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],TOWGS84[0,
0,0,0,0,0,0],AUTHORITY["EPSG","6326"]],PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],UNIT["degree",0.0174532925199433,
AUTHORITY["EPSG","9122"]],AUTHORITY["EPSG","4326"]]
proj4text | +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
```

The `srtext` is the projection definition in well known text (you may recognise this from .prj files in your shapefile collection).

16.1.7 結論

今 PostGIS の機能は PostgreSQL のコピーにインストールされています。これにより PostGIS のさまざまな空間関数を使用できるでしょう。

16.1.8 次は?

次はデータベースにおける空間地物の表現方法について学習しましょう。

16.2 レッスン: 単純地物モデル

データベースの中にどのように地物を保存し、表現できるでしょうか? このレッスンでは OGC によって定義されている単純地物モデルを見ていきます。

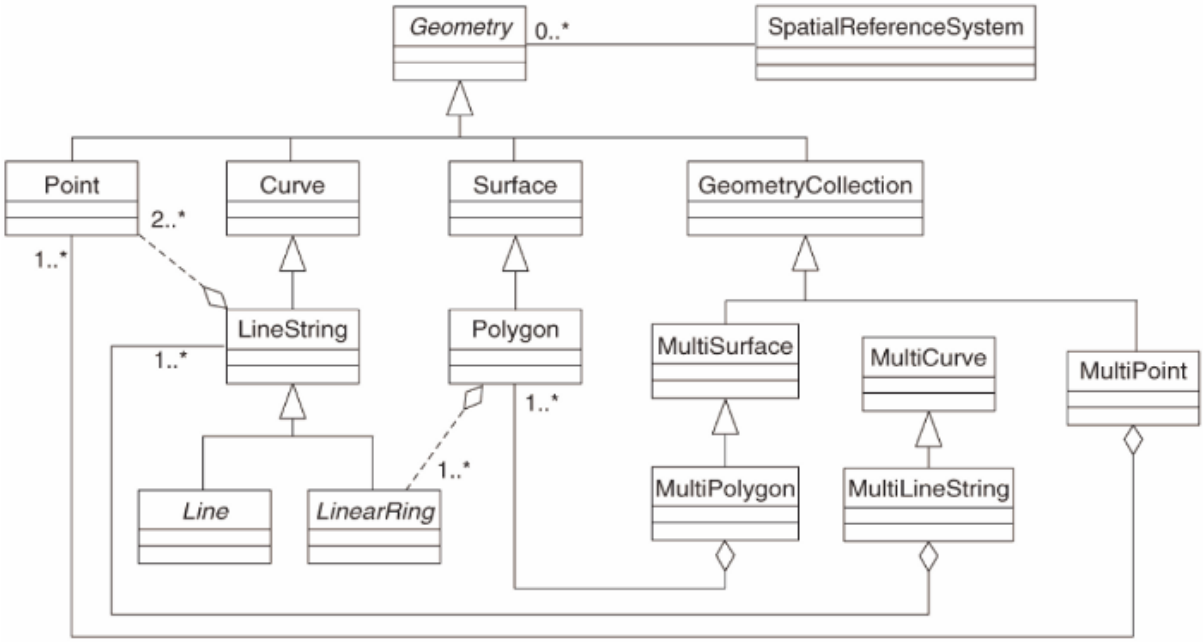
このレッスンの目標: SFS モデルとは何か、それをどうやって使うかを学習します。

16.2.1 OGC とは

Open Geospatial Consortium (OGC) は、1994 年に発足した国際的な民間コンセンサス標準団体です。OGC では、世界中の 370 以上の企業、政府、非営利組織そして研究機関が協力し、地理空間コンテンツとサービス、GIS データの解析と交換のための標準の開発と実装を行っています。 - *Wikipedia*

16.2.2 SFS モデルとは

SQL 用単純地物 (SFS) モデルとはデータベースに地理空間データを格納する **非トポロジ的** な方法で、データへのアクセス、操作、構築のための関数を定義しています。



このモデルは地理空間データをポイント、ラインストリング及びポリゴン型（そしてそれらの集合）で定義しています。

For further information, have a look at the OGC Simple Feature for SQL standard.

16.2.3 ジオメトリフィールドをテーブルに追加する

people（人）テーブルにポイントフィールドを追加しましょう：

```
alter table people add column geom geometry;
```

16.2.4 ジオメトリタイプに基づく制約を追加する

ジオメトリフィールドタイプは、フィールドのジオメトリの **タイプ** を暗黙に指定していないことに気づくでしょう。そのために制約が必要です:

```
alter table people
add constraint people_geom_point_chk
    check(st_geometrytype(geom) = 'ST_Point'::text
          OR geom IS NULL);
```

これはポイントジオメトリまたは null 値だけを受け入れる制約をテーブルに追加します。

16.2.5 ★★★ （上級レベル） 自分でやってみよう:

cities（都市）という新しいテーブルを作成して、それに適切な列を追加します。それにはポリゴン (市の境界) を格納するジオメトリフィールドを含めて、ジオメトリをポリゴンに制限する制約を追加して下さい。

答え

16.2.6 geometry_columns テーブルの設定

この時点で、 `geometry_columns` テーブルにエントリを追加する必要があります：

```
insert into geometry_columns values
  ('','public','people','geom',2,4326,'POINT');
```

Why? `geometry_columns` is used by certain applications to be aware of which tables in the database contain geometry data.

注釈

上記の `INSERT` 文でエラーが発生した場合は、まずこのクエリを実行してください：

```
select * from geometry_columns;
```

If the column `f_table_name` contains the value `people`, then this table has already been registered and you don't need to do anything more.

値「2」は次元の数を示します。この場合、 **X** と **Y** の 2 つです。

The value `4326` refers to the projection we are using; in this case, WGS 84, which is referred to by the number 4326 (refer to the earlier discussion about the EPSG).

★☆☆ （初級レベル） 自分でやってみよう:

Add an appropriate `geometry_columns` entry for your new cities layer

答え

```
insert into geometry_columns values
      ('','public','cities','geom',2,4326,'POLYGON');
```

16.2.7 SQL を使用してテーブルにジオメトリレコードを追加する

テーブルが地理的に有効になったので、そこにジオメトリを格納することができます：

```
insert into people (name,house_no, street_id, phone_no, geom)
        values ('Fault Towers',
                  34,
                  3,
                  '072 812 31 28',
                  'SRID=4326;POINT(33 -33)');
```

注釈

上記の新しいエントリには使用する投影法 (SRID) を指定する必要があります。これはプレーンテキストを用いて新しいポイントのジオメトリを入力すると正しい投影法の情報が自動的に付加されないためです。新しいポイントはデータセットと同じ SRID を使用する必要がありますのでそれを指定しなければいけません。

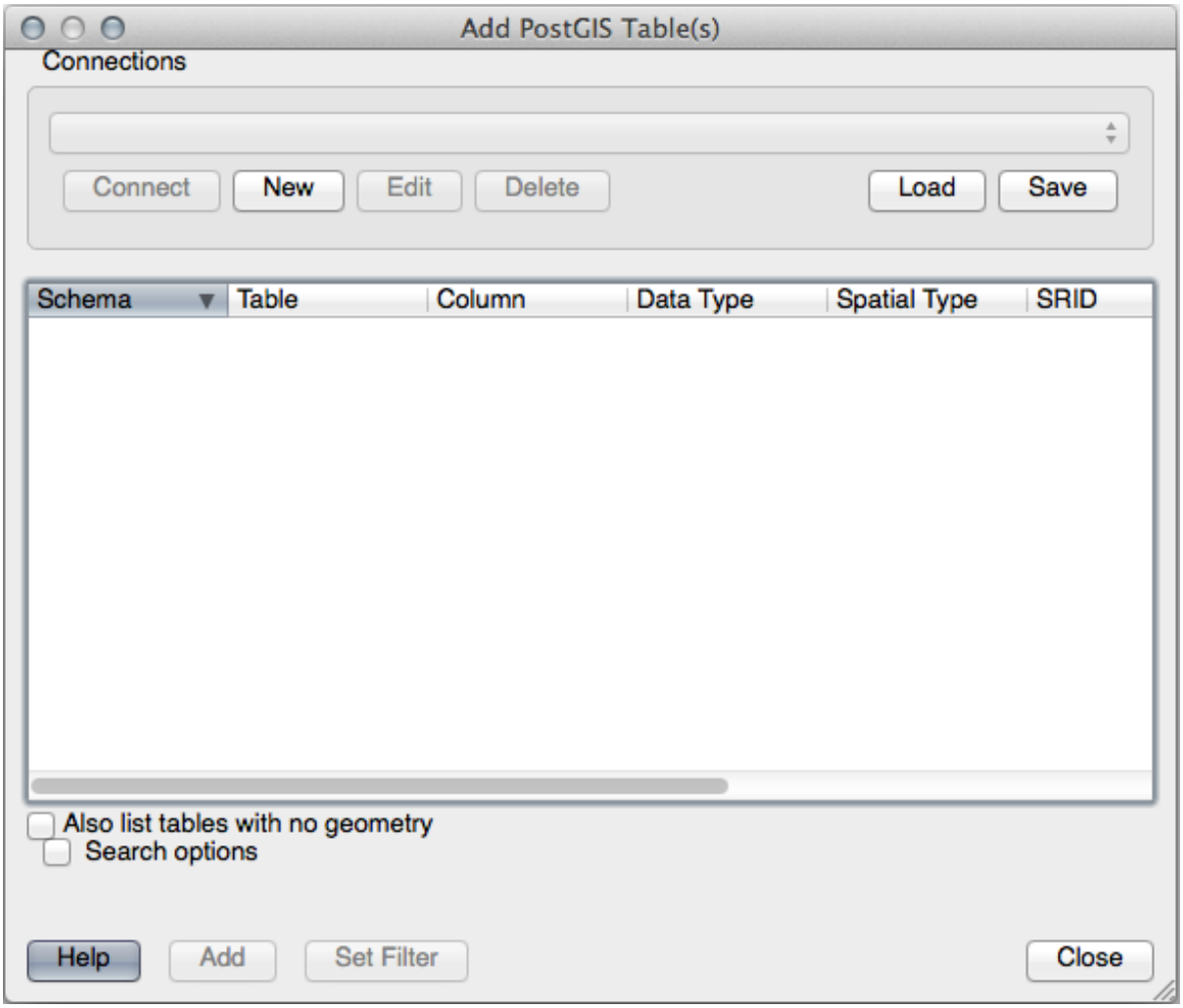
もしグラフィカルなインターフェイスを使用していれば、たとえば、各ポイントの投影法は自動で指定されます。つまり以前行ったようにデータセットに投影法を指定しておけば、すべてのポイントに対して正しい投影法を指定しなくてもよいのです。

Now is probably a good time to open QGIS and try to view your `people` table. Also, we should try editing / adding / deleting records and then performing select queries in the database to see how the data has changed.

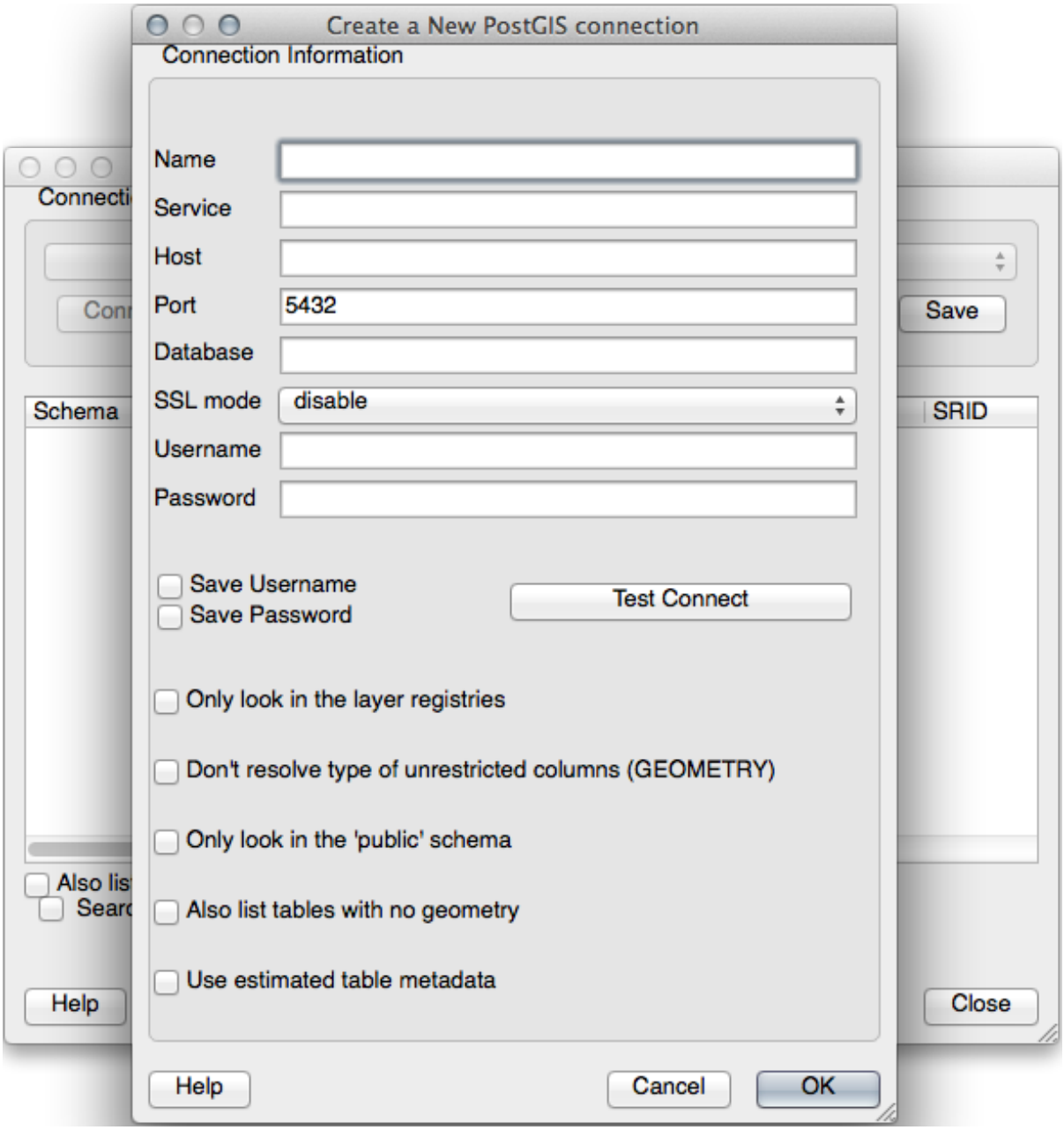
To load a PostgreSQL layer in QGIS, use the *Layer* *Add PostgreSQL Layers* menu option or toolbar button:



ダイアログが表示されます:



新規 ボタンをクリックしてこのダイアログを開きます:



新しい接続を定義します。例えば:

```
Name: myPG
Service:
Host: localhost
Port: 5432
Database: address
User:
Password:
```

To see whether QGIS has found the `address` database and that your username and password are correct, click *Test Connect*. If it works, check the boxes next to *Save Username* and *Save Password*. Then click *OK* to create this connection.

Back in the *Add PostgreSQL Layers* dialog, click *Connect* and add layers to your project as usual.

★★☆ （中級レベル） 自分でやってみよう:

人の名前と街路の名前、位置 (geom 列) をプレーンテキストとして表示するクエリを作成して下さい。

答え

```
select people.name,
       streets.name as street_name,
       st_astext(people.geom) as geometry
from   streets, people
where  people.street_id=streets.id;
```

結果:

```
     name      | street_name |   geometry
---------------+-------------+---------------
 Roger Jones   | High street |
 Sally Norman  | High street |
 Jane Smith    | Main Road   |
 Joe Bloggs    | Low Street  |
 Fault Towers  | Main Road   | POINT(33 -33)
(5 rows)
```

ご覧のとおり、この制約により、データベースに null を追加できます。

16.2.8 結論

空間オブジェクトをデータベースに追加して GIS ソフトウェアで表示する方法を見てきました。

16.2.9 次は?

次はデータベースへデータをインポートする方法、およびデータベースからデータをエクスポートする方法を見ていきます。

16.3 レッスン: インポートとエクスポート

Of course, a database with no easy way to migrate data into it and out of it would not be of much use. Fortunately, there are a number of tools that will let you easily move data into and out of PostgreSQL.

16.3.1 shp2pgsql

shp2pgsql is a commandline tool to import ESRI Shapefile to the database. Under Unix, you can use the following command for importing a new PostgreSQL table:

```
shp2pgsql -s <SRID> -c -D -I <path to shapefile> <schema>.<table> | \
  psql -d <databasename> -h <hostname> -U <username>
```

Windows では, 2 ステップでインポート処理を実行します :

```
shp2pgsql -s <SRID> -c -D -I <path to shapefile> <schema>.<table> > import.sql
psql psql -d <databasename> -h <hostname> -U <username> -f import.sql
```

次のようなエラーが発生することがあります :

```
ERROR:  operator class "gist_geometry_ops" does not exist for access method
"gist"
```

これは、インポートするデータの空間インデックスを **その場** で作成することに関する既知の問題です。このエラーを回避するには、-I パラメータを除外してください。この場合、空間インデックスは直接作成されないので、データをインポートした後にデータベースで作成する必要があります。(空間インデックスの作成については、次のレッスンで説明します)。

16.3.2 pgsql2shp

pgsql2shp is a commandline tool to export PostgreSQL Tables, Views or SQL select queries. To do this under Unix:

```
pgsql2shp -f <path to new shapefile> -g <geometry column name> \
  -h <hostname> -U <username> <databasename> <table | view>
```

クエリを使用してデータをエクスポートするには :

```
pgsql2shp -f <path to new shapefile> -g <geometry column name> \
  -h <hostname> -U <username> "<query>"
```

16.3.3 ogr2ogr

ogr2ogr is a very powerful tool to convert data into and from PostgreSQL to many data formats. ogr2ogr is part of the GDAL library and has to be installed separately. To export a table from PostgreSQL to GML, you can use this command:

```
ogr2ogr -f GML export.gml PG:'dbname=<databasename> user=<username>
        host=<hostname>' <Name of PostgreSQL-Table>
```

16.3.4 DB Manager

You may have noticed another option in the *Database* menu labeled *DB Manager*. This is a tool that provides a unified interface for interacting with spatial databases including PostgreSQL. It also allows you to import and export from databases to other formats. Since the next module is largely devoted to using this tool, we will only briefly mention it here.

16.3.5 結論

データベースとの間でデータをインポート及びエクスポートは, 多様な方法で行うことができます. 異なるデータソースを使用する場合は特に, この機能 (またはこの機能に似た機能) を使用します.

16.3.6 次は?

次に, 私たちが以前に作成したデータを参照する方法を見ていきます.

16.4 レッスン: 空間クエリ

空間クエリは、その他のデータベースのクエリと変わりありません。ジオメトリ列を他のデータベースの列と同じように使うことができます。PostGIS をデータベースにインストールすることでデータベースのクエリの機能が追加されます。

このレッスンの目的： 空間関数が、空間関数でない一般の関数と同様に導入できることを明らかにすること。

16.4.1 空間演算子

ある地点 (X,Y) から距離が 2 度内の地点を特定したい場合以下の操作ができます

```
select *
from people
where st_distance(geom,'SRID=4326;POINT(33 -34)') < 2;
```

結果:

```
 id |     name      | house_no | street_id |   phone_no    |   geom
----+---------------+----------+-----------+---------------+----------------
  6 | Fault Towers |       34 |         3 | 072 812 31 28 | 01010008040C0
(1 row)
```

注釈

上記の geom 値はこのページ上では当サイトのスペースを残すため削除されました。人が読める座標を確認したい場合、上記の「WKT としてポイントを表示」セクションと類似の操作で確認できます。

上述のクエリが 2 **度** という空間内にある地点をすべて返すということはどうやって判るでしょうか？なぜ 2 **メートル**、あるいはその他の単位ではないのでしょうか？

答え

レイヤが使用している CRS は WGS 84 であるため、このクエリで使用されている単位は度です。これは地理 CRS で、単位は度であることを意味します。UTM 図法のような投影 CRS は、単位がメートルです。

クエリを書くときには、レイヤの CRS がどの単位にあるのかを知る必要があることを忘れないでください。そうすることで、期待通りの結果を返すクエリを書くことができるようになります。

16.4.2 空間索引

空間索引も定義できます。空間索引を使用すると、空間クエリをより迅速に作成できます。ジオメトリ列に空間索引を作成するには、次のようにします。

```
CREATE INDEX people_geo_idx
  ON people
  USING gist
  (geom);

\d people
```

結果:

```
Table "public.people"
   Column   |         Type          |                Modifiers
 -----------+-----------------------+-----------------------------------------
  id        | integer               | not null default
            |                       | nextval('people_id_seq'::regclass)
  name      | character varying(50) |
  house_no  | integer               | not null
  street_id | integer               | not null
  phone_no  | character varying     |
  geom      | geometry              |
Indexes:
  "people_pkey" PRIMARY KEY, btree (id)
  "people_geo_idx" gist (geom)  <-- new spatial key added
  "people_name_idx" btree (name)
Check constraints:
  "people_geom_point_chk" CHECK (st_geometrytype(geom) = 'ST_Point'::text
  OR geom IS NULL)
Foreign-key constraints:
  "people_street_id_fkey" FOREIGN KEY (street_id) REFERENCES streets(id)
```

16.4.3 やってみよう: ★★☆ （中級レベル）

都市のテーブルを、そのジオメトリ列が空間索引付けされるように変更します。

答え

```
CREATE INDEX cities_geo_idx
  ON cities
  USING gist (geom);
```

16.4.4 PostGIS 空間関数デモ

PostGIS の空間関数のデモを行うため、いくつかの（架空の）データを含む新しいデータベースを作成します。

まず、新しいデータベースを作成します（まず psql シェルを終了します）。

```
createdb postgis_demo
```

Postgis 拡張機能をインストールすることを忘れないでください：

```
psql -d postgis_demo -c "CREATE EXTENSION postgis;"
```

Next, import the data provided in the `exercise_data/postgis/` directory. Refer back to the previous lesson for instructions, but remember that you'll need to create a new PostGIS connection to the new database. You can import from the terminal or via DB Manager. Import the files into the following database tables:

- `points.shp` into `building`
- `lines.shp` into `road`
- `polygons.shp` into `region`

Load these three database layers into QGIS via the *Add PostGIS Layers* dialog, as usual. When you open their attribute tables, you'll note that they have both an `id` field and a `gid` field created by the PostGIS import.

テーブルはインポートされていますので、データを照会するための PostGIS を使用できます。端末（コマンドライン）に戻って以下を実行することにより psql のプロンプトを入力してください:

```
psql postgis_demo
```

QGIS でそれらを開き、結果を見ることができるように、それらからのビューを作成することによってこれらの select 文の一部をデモします。

場所による選択

クワズール地域のすべての建物を手に入れよう：

```
SELECT a.id, a.name, st_astext(a.geom) as point
  FROM building a, region b
    WHERE st_within(a.geom, b.geom)
    AND b.name = 'KwaZulu';
```

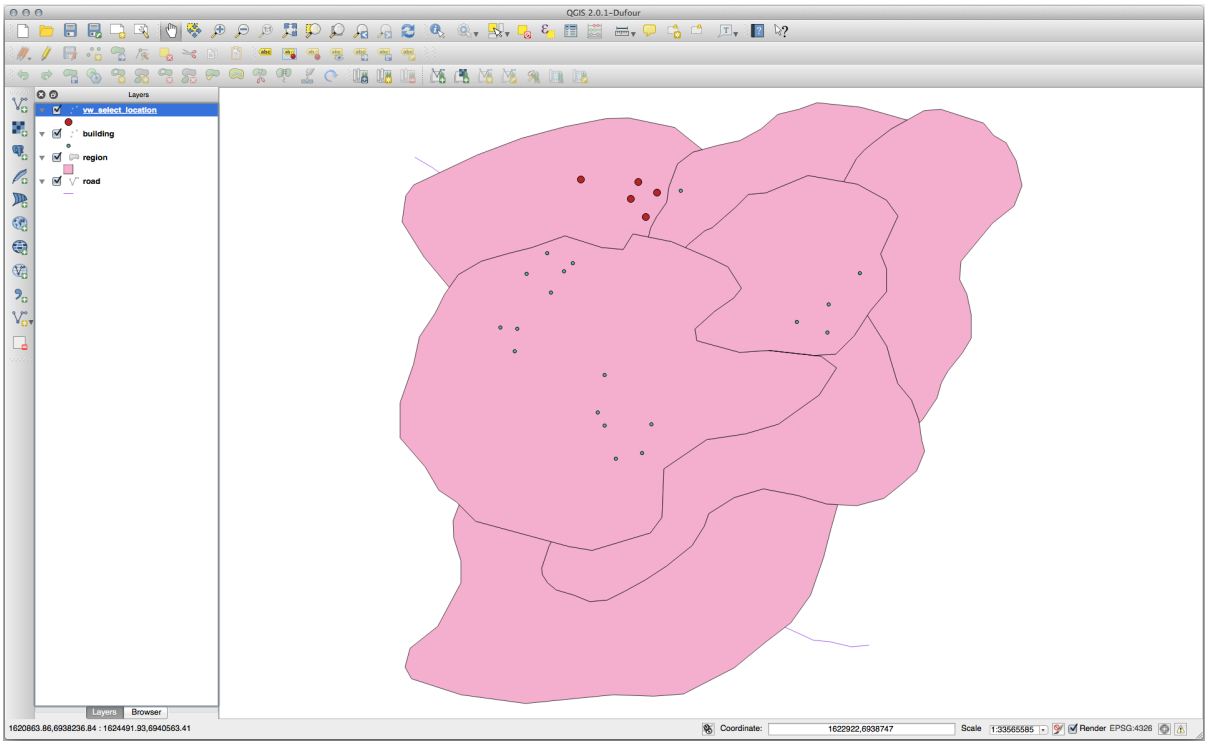
結果:

```
 id | name |                  point
----+------+-----------------------------------------
 30 | York | POINT(1622345.23785063 6940490.65844485)
 33 | York | POINT(1622495.65620524 6940403.87862489)
 35 | York | POINT(1622403.09106394 6940212.96302097)
 36 | York | POINT(1622287.38463732 6940357.59605424)
 40 | York | POINT(1621888.19746548 6940508.01440885)
(5 rows)
```

または、そこからビューを作成する場合は、次のようにします。

```
CREATE VIEW vw_select_location AS
  SELECT a.gid, a.name, a.geom
    FROM building a, region b
      WHERE st_within(a.geom, b.geom)
      AND b.name = 'KwaZulu';
```

レイヤとしてビューを追加し、QGIS で表示：



近傍の選択

北海道地域に隣接する地域のすべての名前のリストを表示する：

```
SELECT b.name
  FROM region a, region b
    WHERE st_touches(a.geom, b.geom)
    AND a.name = 'Hokkaido';
```

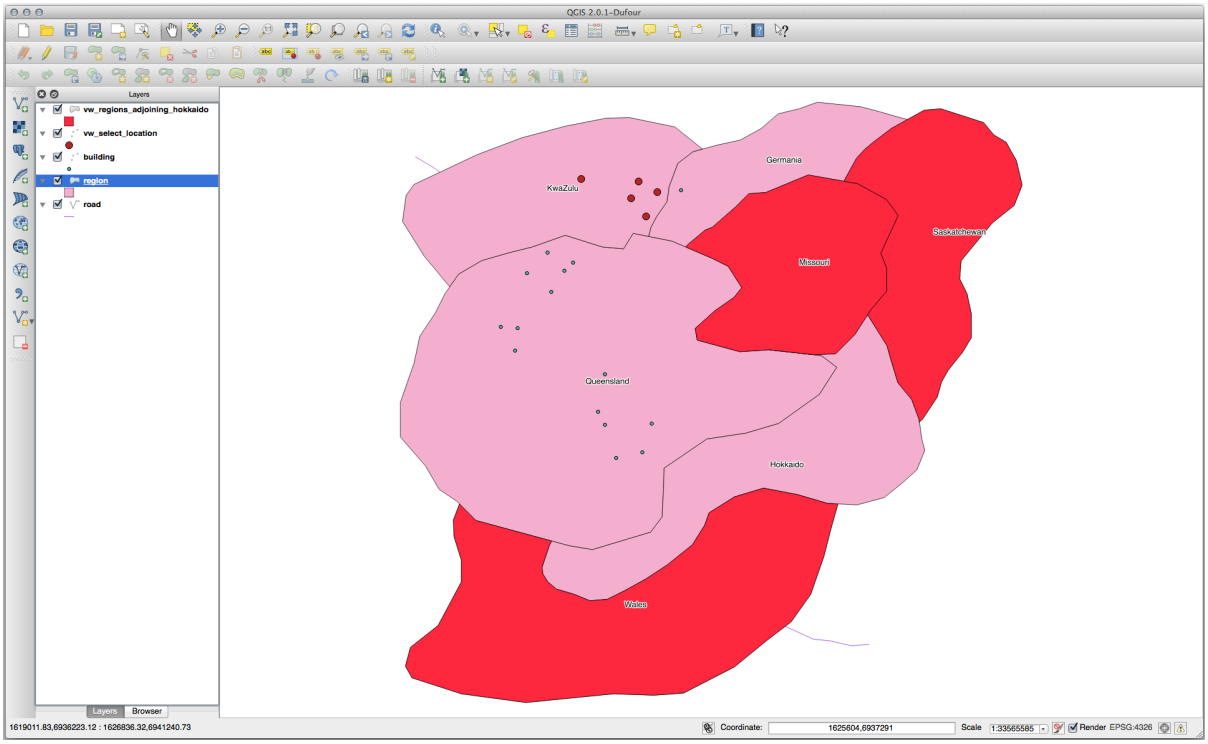
結果:

```
    name
--------------
 Missouri
 Saskatchewan
 Wales
(3 rows)
```

ビューとして：

```
CREATE VIEW vw_regions_adjoining_hokkaido AS
  SELECT b.gid, b.name, b.geom
    FROM region a, region b
      WHERE st_touches(a.geom, b.geom)
      AND a.name = 'Hokkaido';
```

QGIS では:

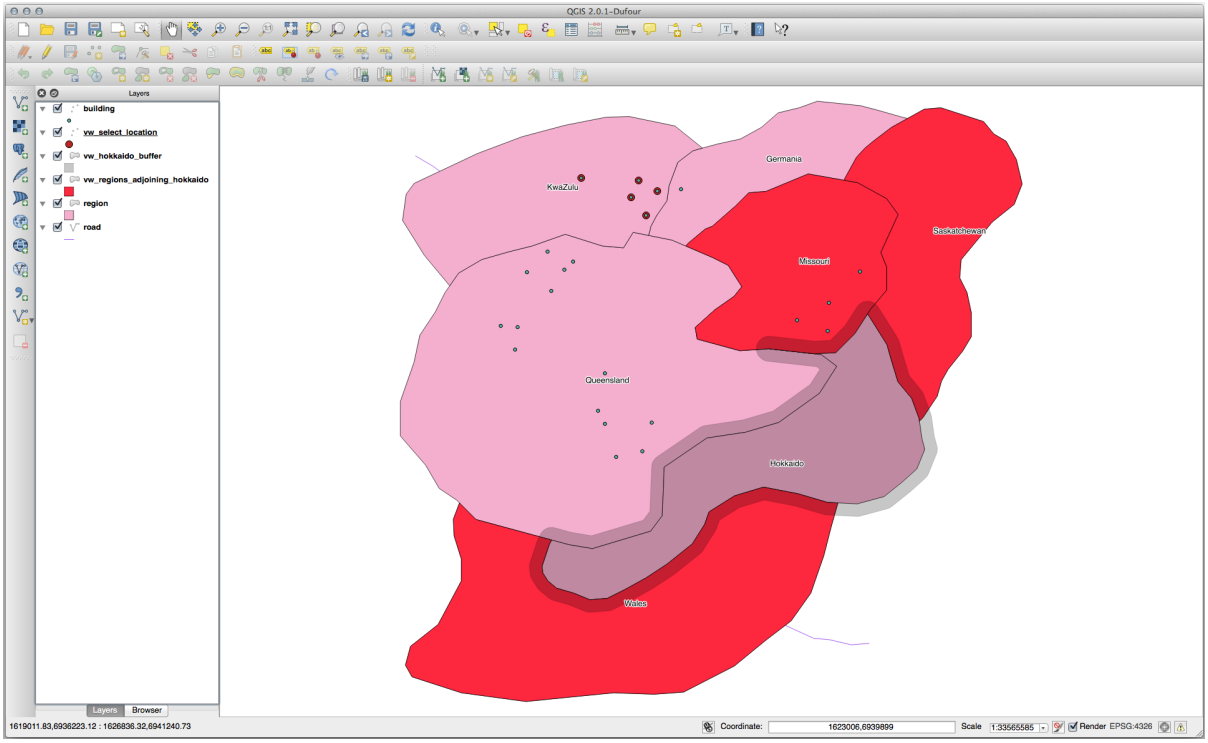


不足している地域（クイーンズランド州）に注意してください。これはトポロジエラーが原因である可能性があります。このようなアーティファクトによって、データの潜在的な問題を警告できます。データが持つ異常に巻き込まれることなくこの謎を解決するために、代わりにバッファ交差を使用できます：

```
CREATE VIEW vw_hokkaido_buffer AS
  SELECT gid, ST_BUFFER(geom, 100) as geom
    FROM region
      WHERE name = 'Hokkaido';
```

北海道の周囲に 100m のバッファを作成します。

暗いエリアがバッファです:

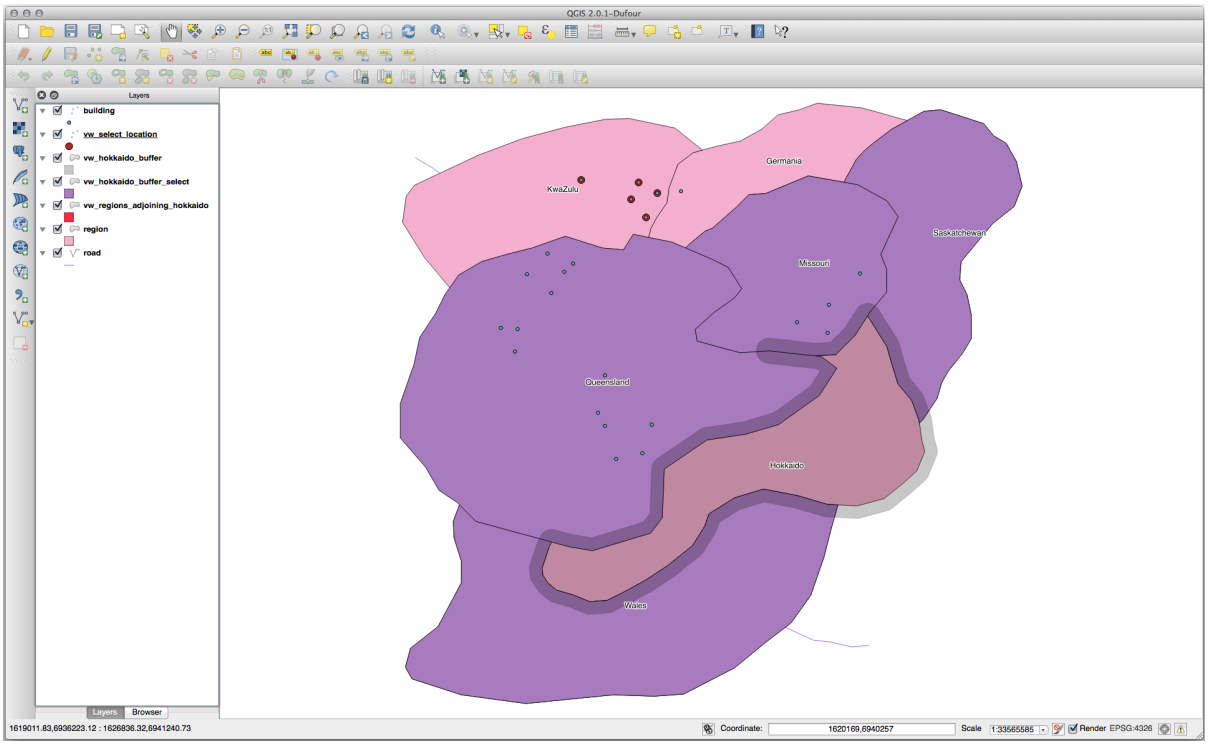


バッファを使用して選択：

```
CREATE VIEW vw_hokkaido_buffer_select AS
  SELECT b.gid, b.name, b.geom
    FROM
    (
      SELECT * FROM
        vw_hokkaido_buffer
    ) a,
    region b
    WHERE ST_INTERSECTS(a.geom, b.geom)
    AND b.name != 'Hokkaido';
```

In this query, the original buffer view is used as any other table would be. It is given the alias `a`, and its geometry field, `a.geom`, is used to select any polygon in the `region` table (alias `b`) that intersects it. However, Hokkaido itself is excluded from this select statement, because we don't want it; we only want the regions adjoining it.

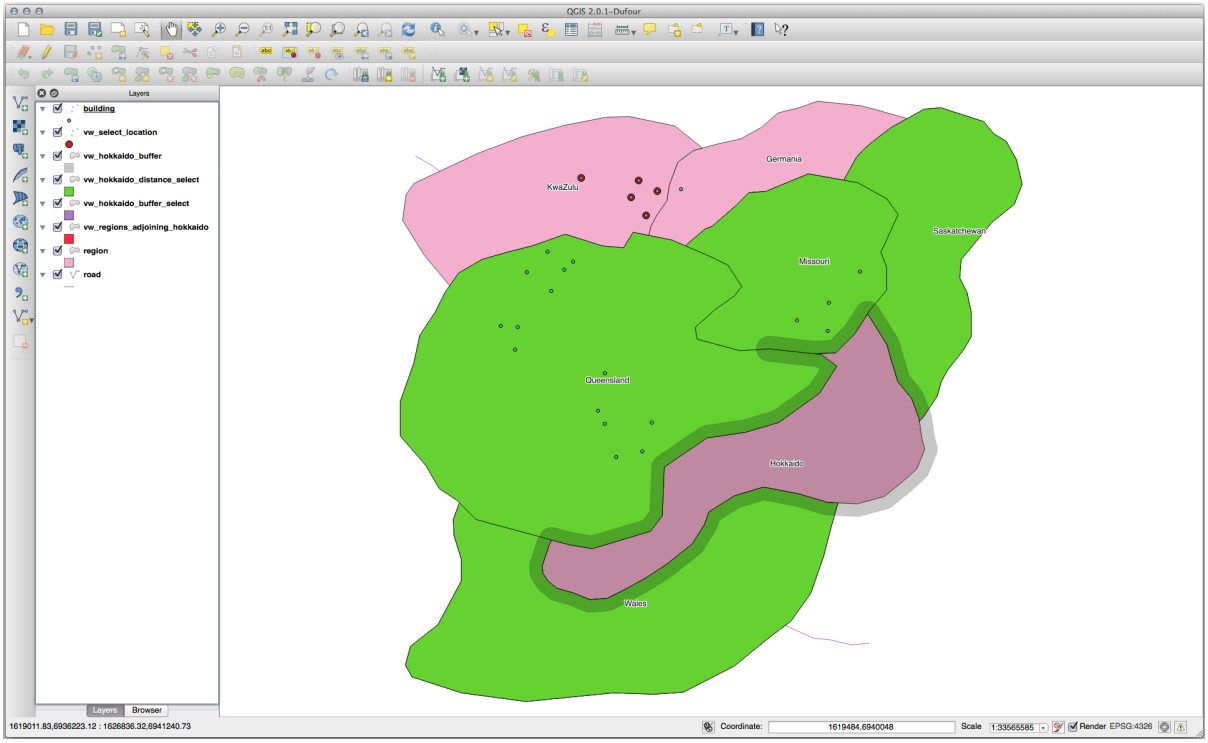
QGIS では:



バッファを作成する追加のステップを行うことなしに、指定された距離内のすべてのオブジェクトを選択することもできます：

```
CREATE VIEW vw_hokkaido_distance_select AS
  SELECT b.gid, b.name, b.geom
    FROM region a, region b
      WHERE ST_DISTANCE (a.geom, b.geom) < 100
      AND a.name = 'Hokkaido'
      AND b.name != 'Hokkaido';
```

これは、中間バッファステップを必要とせずに、同じ結果を達成します：



一意の値を選択する

クイーンズランド州のすべての建物に固有の町名のリストを表示します：

```
SELECT DISTINCT a.name
  FROM building a, region b
    WHERE st_within(a.geom, b.geom)
    AND b.name = 'Queensland';
```

結果:

```
  name
---------
 Beijing
 Berlin
 Atlanta
(3 rows)
```

その他の事例

```
CREATE VIEW vw_shortestline AS
  SELECT b.gid AS gid,
        ST_ASTEXT(ST_SHORTESTLINE(a.geom, b.geom)) as text,
        ST_SHORTESTLINE(a.geom, b.geom) AS geom
    FROM road a, building b
      WHERE a.id=5 AND b.id=22;

CREATE VIEW vw_longestline AS
  SELECT b.gid AS gid,
         ST_ASTEXT(ST_LONGESTLINE(a.geom, b.geom)) as text,
         ST_LONGESTLINE(a.geom, b.geom) AS geom
    FROM road a, building b
      WHERE a.id=5 AND b.id=22;
```

```
CREATE VIEW vw_road_centroid AS
  SELECT a.gid as gid, ST_CENTROID(a.geom) as geom
    FROM road a
      WHERE a.id = 1;

CREATE VIEW vw_region_centroid AS
  SELECT a.gid as gid, ST_CENTROID(a.geom) as geom
    FROM region a
      WHERE a.name = 'Saskatchewan';
```

```
SELECT ST_PERIMETER(a.geom)
  FROM region a
    WHERE a.name='Queensland';

SELECT ST_AREA(a.geom)
  FROM region a
    WHERE a.name='Queensland';
```

```
CREATE VIEW vw_simplify AS
  SELECT gid, ST_Simplify(geom, 20) AS geom
    FROM road;

CREATE VIEW vw_simplify_more AS
  SELECT gid, ST_Simplify(geom, 50) AS geom
    FROM road;
```

```
CREATE VIEW vw_convex_hull AS
  SELECT
    ROW_NUMBER() over (order by a.name) as id,
```

(次のページに続く)

(前のページからの続き)

```
    a.name as town,
    ST_CONVEXHULL(ST_COLLECT(a.geom)) AS geom
    FROM building a
    GROUP BY a.name;
```

16.4.5 結論

PostGIS からの新しいデータベース機能を使用して、空間オブジェクトをクエリする方法を見てきました。

16.4.6 次は?

次は、より複雑な幾何形状の構造と、それらを PostGIS を使用して作成する方法を調査するつもりです。

16.5 レッスン: ジオメトリの構成

このセクションでは単純なジオメトリが SQL 内でどのように構成されるかを少し掘り下げます。実際には、複雑なジオメトリをデジタル化ツールを使用して作るには、QGIS のような GIS を使用するでしょう。しかし、それらがどのように形作られているかを知ることは、クエリを書いたりデータベースがどのように作られているかを理解するのに役立ちます。

The goal of this lesson: To better understand how to create spatial entities directly in PostgreSQL.

16.5.1 ラインストリングの作成

Going back to our `address` database, let's get our streets table matching the others; i.e., having a constraint on the geometry, an index and an entry in the `geometry_columns` table.

16.5.2 やってみよう: ★★☆ （中級レベル）

- Modify the `streets` table so that it has a geometry column of type ST_LineString.
- ジオメトリ列のテーブルに伴う更新を行うことを忘れないでください！
- また、ラインストリングでないか null であるようなジオメトリが追加されることを防ぐための制約を追加します。
- 新しいジオメトリ列に空間インデックスを作成します。

答え

```
alter table streets add column geom geometry;
alter table streets add constraint streets_geom_point_chk check
     (st_geometrytype(geom) = 'ST_LineString'::text OR geom IS NULL);
insert into geometry_columns values ('','public','streets','geom',2,4326,
     'LINESTRING');
create index streets_geo_idx
  on streets
  using gist
  (geom);
```

次に、street テーブルにラインストリングを挿入しましょう。この場合、既存の street レコードを更新します：

```
update streets
set geom = 'SRID=4326;LINESTRING(20 -33, 21 -34, 24 -33)'
where streets.id=2;
```

結果を QGIS で確認してみます。(' レイヤ' パネルの streets レイヤを右クリックし、' レイヤの領域にズームする' を選択する必要があるかもしれません。)

いくつかは QGIS から、いくつかはコマンドラインから street のエントリをもう少し追加します。

16.5.3 ポリゴンの作成

ポリゴンの作成も簡単です。覚えておくべきことの 1 つは、ポリゴンには少なくとも 4 つの頂点があり、最後と最初のものが同じ場所にあることです。

```
insert into cities (name, geom)
values ('Tokyo', 'SRID=4326;POLYGON((10 -10, 5 -32, 30 -27, 10 -10))');
```

注釈

ポリゴンは、その座標リストの周りに二重括弧を必要とします。これにより、複数の連結していない領域を持つ複雑なポリゴンを追加できるようなります。例えば

```
insert into cities (name, geom)
values ('Tokyo Outer Wards',
        'SRID=4326;POLYGON((20 10, 20 20, 35 20, 20 10),
                           (-10 -30, -5 0, -15 -15, -10 -30))'
        );
```

この手順に従った場合、QGIS に cities データセットをロードし、属性テーブルを開き、新しいエントリを選択することで、何をしたか確認できます。二つの新しいポリゴンが 1 つのポリゴンのように振る舞うか

に注意してください。

16.5.4 練習:Cities を People にリンクする

この演習では次のことを行う必要があります。

- people テーブルからすべてのデータを削除します。
- cities テーブルの主キーを参照する people への外部キー列を追加します。
- QGIS を使っていくつかの cities をキャプチャします。
- SQL を使って、それぞれが関連した street と city を持った、新しい people レコードをいくつか挿入してください。

更新された people のスキーマは、次のようになります:

```
\d people

Table "public.people"
   Column   |         Type          |                      Modifiers
 -----------+-----------------------+-----------------------------------------------
  id        | integer               | not null
            |                       | default nextval('people_id_seq'::regclass)
  name      | character varying(50) |
  house_no  | integer               | not null
  street_id | integer               | not null
  phone_no  | character varying     |
  geom      | geometry              |
  city_id   | integer               | not null
Indexes:
  "people_pkey" PRIMARY KEY, btree (id)
  "people_name_idx" btree (name)
Check constraints:
  "people_geom_point_chk" CHECK (st_geometrytype(geom) =
                        'ST_Point'::text OR geom IS NULL)
Foreign-key constraints:
  "people_city_id_fkey" FOREIGN KEY (city_id) REFERENCES cities(id)
  "people_street_id_fkey" FOREIGN KEY (street_id) REFERENCES streets(id)
```

答え

```
delete from people;
alter table people add column city_id int not null references cities(id);
```

（QGIS で cities をキャプチャする）

```
insert into people (name,house_no, street_id, phone_no, city_id, geom)
   values ('Faulty Towers',
```

```
            34,
            3,
            '072 812 31 28',
            1,
            'SRID=4326;POINT(13 -15)');

insert into people (name,house_no, street_id, phone_no, city_id, geom)
   values ('IP Knightly',
            32,
            1,
            '071 812 31 28',
            1,
            'SRID=4326;POINT(18 -24)');

insert into people (name,house_no, street_id, phone_no, city_id, geom)
   values ('Rusty Bedsprings',
            39,
            1,
            '071 822 31 28',
            1,
            'SRID=4326;POINT(22 -25)');
```

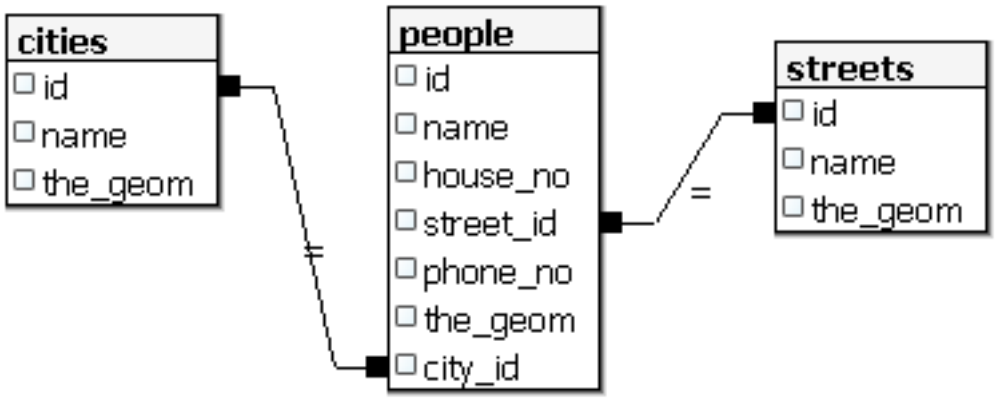
次のエラーメッセージが出た場合:

```
ERROR:  insert or update on table "people" violates foreign key constraint
        "people_city_id_fkey"
DETAIL: Key (city_id)=(1) is not present in table "cities".
```

then it means that while experimenting with creating polygons for the cities table, you must have deleted some of them and started over. Just check the entries in your cities table and use any `id` which exists.

16.5.5 スキーマに着目する

スキーマはこのように見えるべきです:



16.5.6 やってみよう: ★★★ （上級レベル）

その都市のすべてのアドレスの最小凸包を計算し、その領域の周りのバッファを計算することにより、都市の境界線を作成します。

答え

- 'Tokyo Outer Wards' 市に何人か加えます

```
INSERT INTO people (name, house_no, street_id, phone_no, city_id, geom)
   VALUES ('Bad Aboum',
           57,
           2,
           '073 712 31 21',
           2,
           'SRID=4326;POINT(22 18)');

INSERT INTO people (name, house_no, street_id, phone_no, city_id, geom)
   VALUES ('Pat Atra',
           59,
           2,
           '074 712 31 25',
           2,
           'SRID=4326;POINT(23 14)');

INSERT INTO people (name, house_no, street_id, phone_no, city_id, geom)
   VALUES ('Kat Herin',
           65,
           2,
           '074 722 31 28',
           2,
           'SRID=4326;POINT(29 18)');
```

- myPolygonTable テーブルを生成します

```
CREATE TABLE myPolygonTable (
  id serial NOT NULL PRIMARY KEY,
  city_id int NOT NULL REFERENCES cities(id),
  geom geometry NOT NULL
);

ALTER TABLE myPolygonTable
  ADD CONSTRAINT myPolygonTable_geom_polygon_chk
  CHECK (st_geometrytype(geom) = 'ST_Polygon'::text );
```

- 凸包を作成して読み込みます

```
INSERT INTO myPolygonTable (city_id, geom)
  SELECT * FROM
  (
    SELECT
      ROW_NUMBER() over (order by city_id)::integer AS city_id,
      ST_CONVEXHULL(ST_COLLECT(geom)) AS geom
        FROM people
        GROUP BY city_id
  ) convexHulls;
```

16.5.7 サブオブジェクトへのアクセス

With the SFS-Model functions, you have a wide variety of options to access sub-objects of SFS Geometries. When you want to select the first vertex point of every polygon geometry in the table `myPolygonTable`, you have to do this in this way:

- ポリゴン境界をラインストリングに変換する：

```
select st_boundary(geom) from myPolygonTable;
```

- 結果のラインストリングの最初の頂点を選択します。

```
select st_startpoint(myGeometry)
from (
    select st_boundary(geom) as myGeometry
    from myPolygonTable) as foo;
```

16.5.8 データプロセシング

PostGIS では、すべての OGC SFS/MM 標準準拠関数をサポートしています。これらの関数はすべて ST_ で始まります。

16.5.9 クリッピング

データのサブパートをクリップするには、 ST_INTERSECT() 関数を使用します。空のジオメトリを避けるには、次のようにします。

```
where not st_isempty(st_intersection(a.geom, b.geom))
```

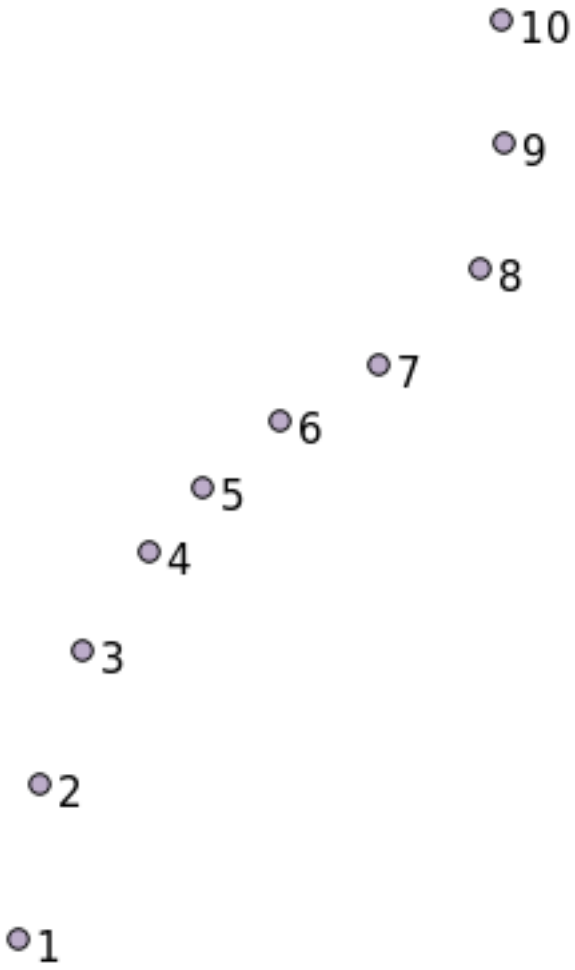


```
select st_intersection(a.geom, b.geom), b.*
from clip as a, road_lines as b
where not st_isempty(st_intersection(st_setsrid(a.geom,32734),
  b.geom));
```



16.5.10 ジオメトリを他のジオメトリから構築する

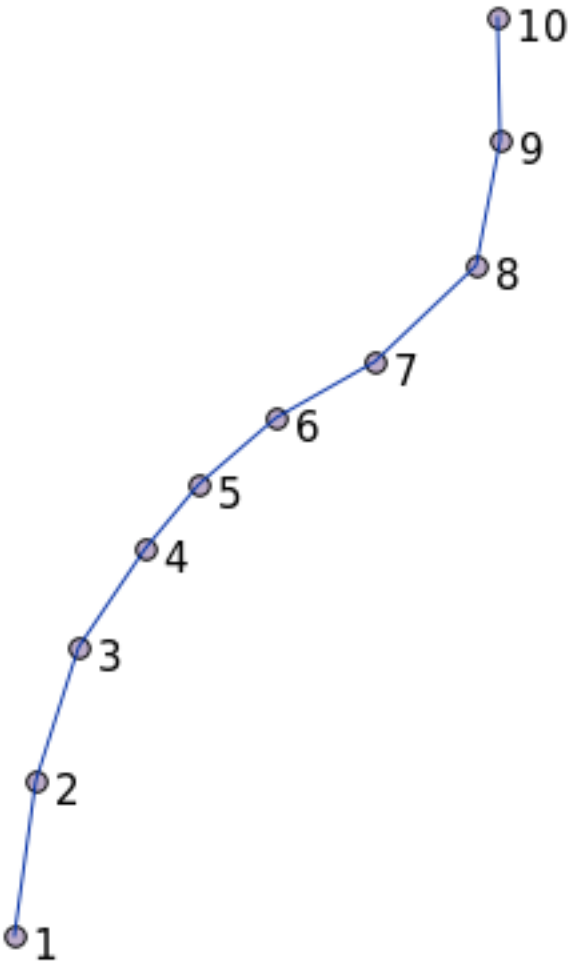
From a given point table, you want to generate a linestring. The order of the points is defined by their `id`. Another ordering method could be a timestamp, such as the one you get when you capture waypoints with a GPS receiver.



'points' という新しいポイントレイヤからラインストリングを作成するには、次のコマンドを実行します:

```
select ST_LineFromMultiPoint(st_collect(geom)), 1 as id
from (
  select geom
  from points
  order by id
) as foo;
```

どのように機能するかを新しいレイヤを作らずに確認するために、'people' レイヤにこのコマンドを実行することもできます。もちろん、これを行うことに現実世界の意味はほとんどないでしょうが。



16.5.11 ジオメトリクリーニング

このトピックについては このブログエントリ にもっと情報があります。

16.5.12 テーブル間の差

同じ構造を持つ 2 つのテーブルの差を検出するには、PostgreSQL キーワード EXCEPT を使用します：

```
select * from table_a
except
select * from table_b;
```

As the result, you will get all records from `table_a` which are not stored in `table_b`.

16.5.13 表領域

Postgres がディスク上にデータを格納する場所を定義するには、表領域を作成します。

```
CREATE TABLESPACE homespace LOCATION '/home/pg';
```

データベースを作成するときは、例えばどの表領域を使用するか指定できます:

```
createdb --tablespace=homespace t4a
```

16.5.14 結論

PostGIS の文を使用して、より複雑なジオメトリを作成する方法を学びました。これは GIS のフロントエンドを通じて地理対応データベースを操作するときは、暗黙知を向上させることがほとんどであることに留意してください。これらのステートメントを実際に手動で入力する必要は普通はありませんが、その構造について一般的な知識を持っておくと、GIS を使用するときに、特にそうでなければ不可解と思うであろうエラーが発生したときに、役に立ちます。

第17章　QGISプロセシングガイド

このモジュールは Victor Olaya 氏と Paolo Cavallini による貢献です。

内容:

17.1 はじめに

このガイドでは、QGIS プロセシングフレームワークの使い方について説明します。このガイドでは、プロセシングフレームワークやそれが依存するアプリケーションに関する事前知識は必要ありません。QGIS の基本的な知識は前提としています。スクリプトに関する章では、Python とおそらく QGIS Python API の基本的な知識を持っていることを想定しています。

このガイドは自習用又はプロセシングワークショップを実行するために設計されています。

このガイドの例は、QGIS 3.4 を使用しています。それ以外のバージョンでは動作しない、または利用できない可能性があります。

このガイドは、徐々に複雑になっていく小さな演習のセットで構成されています。もし、あなたがプロセシングフレームワークを使ったことがないのであれば、一番最初から始めてください。もし、ある程度の経験があるのであれば、自由にレッスンを飛ばしてください。各章は多かれ少なかれ独立しており、各章のタイトルや冒頭の短い紹介文にあるように、それぞれ新しい概念や新しい要素を導入しています。そのため、特定のトピックを扱っているレッスンを見つけるのは簡単です。

すべてのフレームワークコンポーネントとその使用法についてのより体系的な説明については、ユーザーマニュアルの対応する章を確認することが推奨されます。このガイドと一緒にサポートテキストとして使用してください。

このガイドのすべての演習は、トレーニングマニュアルで使用され、セクション **データ** で参照されるものと同じフリーデータセットを使用します。ダウンロードする ZIP ファイルには、このガイドの各レッスンに対応するいくつかのフォルダが含まれています。それぞれのフォルダには、QGIS のプロジェクトファイルがあります。それを開くだけで、レッスンを始めることができます。

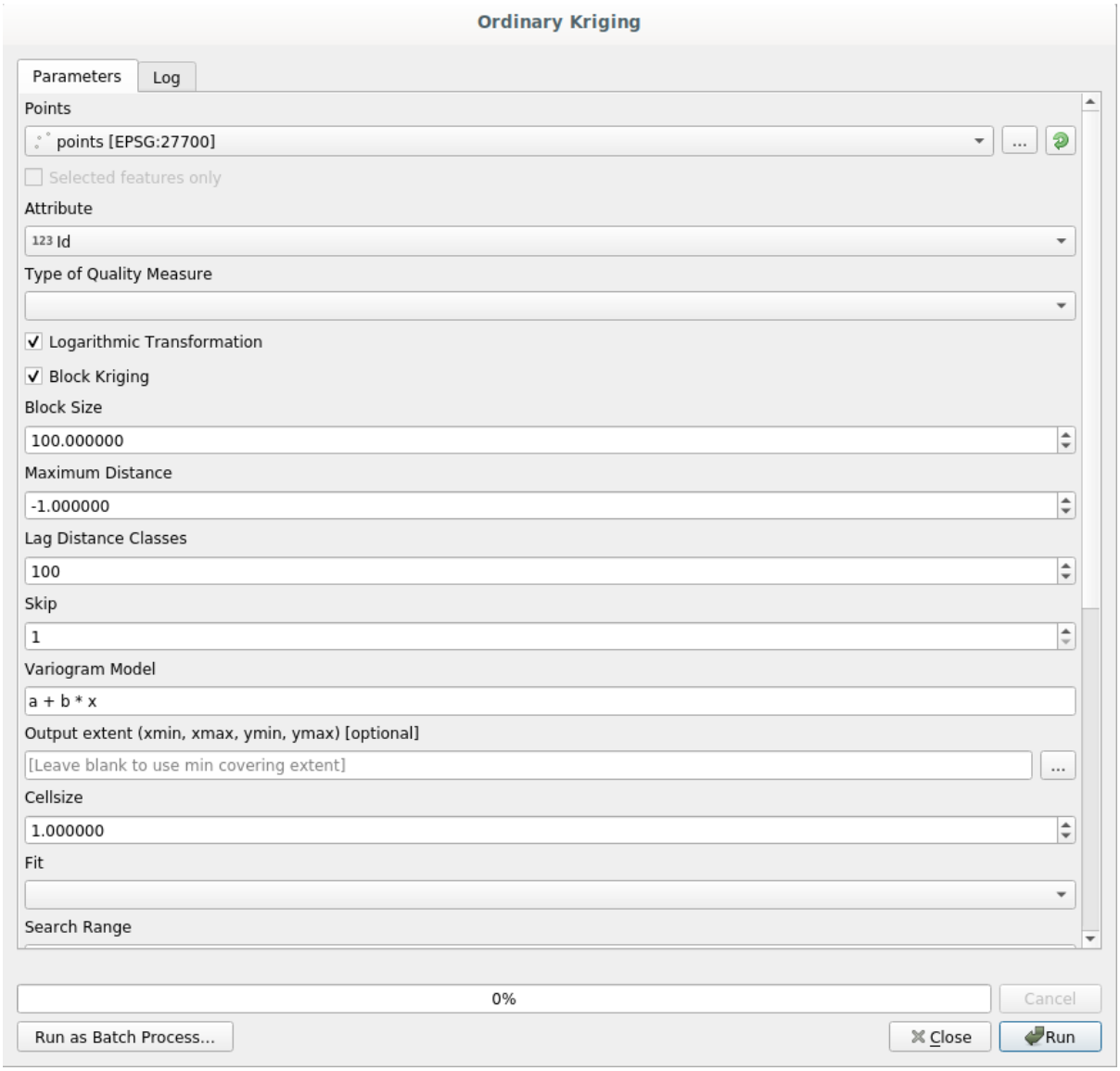
楽しんでください!

17.2 始める前の重要な警告

ワードプロセッサのマニュアルが小説や詩を書く方法を教えてくれないように、CAD チュートリアルが建物の梁のサイズを計算する方法を示してくれないように、このガイドでは空間分析を教えません。代わりに、空間分析を行うための強力なツールである QGIS プロセシングフレームワークを使う方法を説明します。その種の分析を理解するために必要な概念を学ぶかどうかはあなた次第です。試してみたくなるかもしれませんが、それらなしでは、フレームワークとそのアルゴリズムを使っても何にもなりません。

例を挙げてもっと明確に説明しましょう。

与えられた点の集合と、各点における与えられた変数の値から **クリギング** ジオアルゴリズムを使ってラスタレイヤを計算できます。そのモジュールのパラメータダイアログは、次のようなものです。



それは複雑に見えます、よね？

このマニュアルを読むことで、そのモジュールの使い方、バッチ処理で数百のポイントレイヤから一度にラスタレイヤを作成する方法、入力レイヤに選択されているポイントがある場合はどうなるかなど、さまざまなことを学ぶことができます。ただし、パラメータそのものは説明されていません。地理統計学に精

通した経験豊富な分析者であれば、これらのパラメータを理解するのに何の問題もないでしょう。もしあなたがそうではなく、**シル**、**レンジ**、または**ナゲット**という概念に馴染みがなければ、**クリギング**モジュールを使うべきではありません。さらに言えば、**クリギング**モジュールを使うには、空間自己相関やセミバリオグラムといった、おそらく聞いたことがないか、十分に勉強していない概念について学ぶ必要があるので、使う準備ができているとは言えません。まずはそれらを学習し理解した上で、QGIS に戻って実際に実行し、分析を行うべきです。この順序を無視すると、間違った結果や貧弱な（そしてほとんどの場合役に立たない）分析結果を招くことになります。

全てのアルゴリズムがクリギングのように複雑ではありません（しかし、さらに複雑なものもあります！）が、ほとんどはそれが基盤としている基本的な分析の考え方を理解することが必要です。その知識がなく使えば、ほぼ確実に悪い結果を招くでしょう。

空間分析の基礎知識を持たずにジオアルゴリズムを使うことは、文法や構文の知識がなく、物語の構成についても何も知らない状態で小説を書こうとするようなものです。結果は得られるかもしれませんが、その価値はほとんどないでしょう。どうか自分を欺かず、このガイドを読んだだけで空間分析を行って健全な結果が得られるようになると考えないでください。空間分析についても学ぶ必要があります。

ここに、空間データ分析についての詳細を学ぶために読むことができる良い参考文献があります。

地理空間分析（第 *3* 版）：原則、テクニックやソフトウェアツールへの総合ガイド　マイケル・ジョン・デ・スミス、マイケル・F・グッドチャイルド、ポール・A・ロングリー

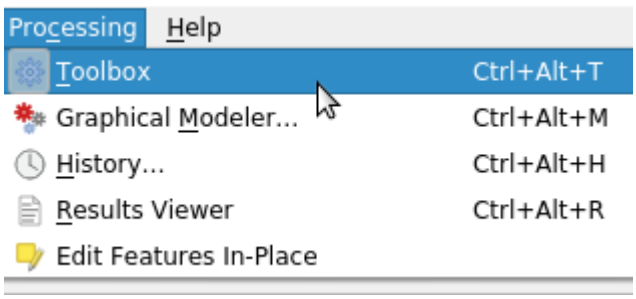
それは ここ でオンラインで入手できます

17.3 プロセシングフレームワークの準備をする

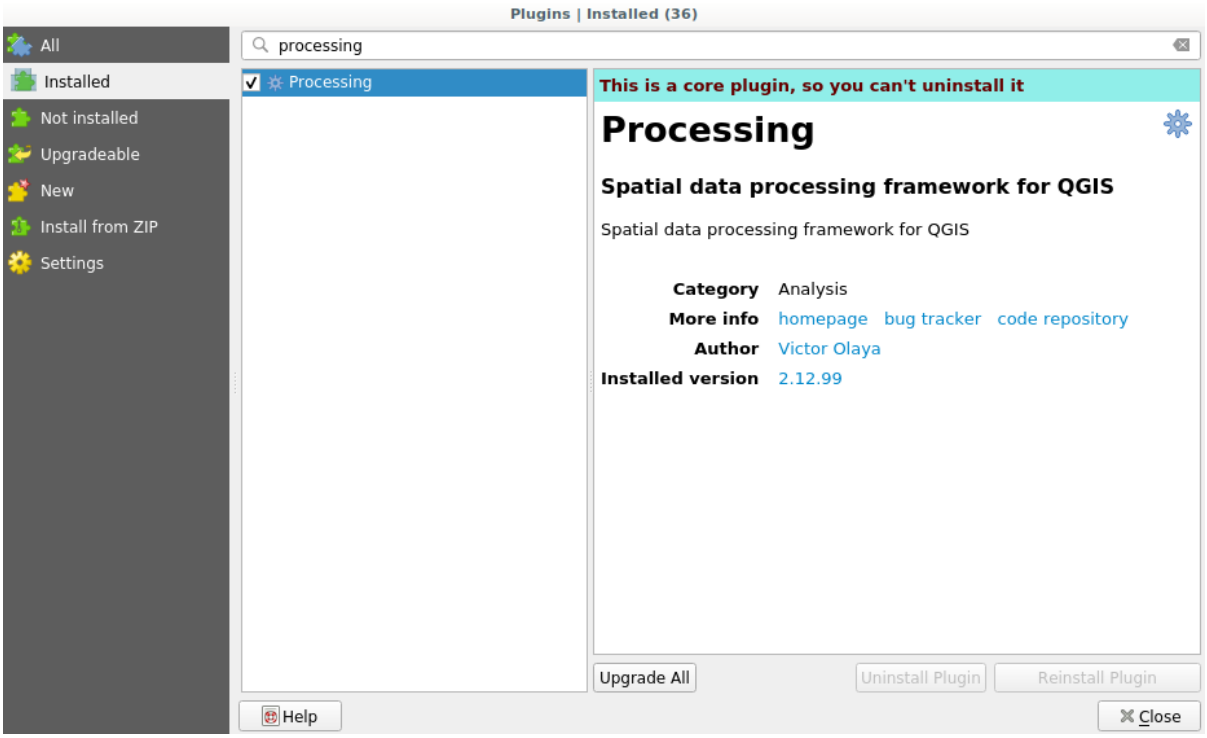
プロセシングフレームワークを使う前に、まず環境設定をする必要があります。設定項目はそれほど多くないので簡単な作業です。

後で利用可能なアルゴリズムのリストを拡張するために使う外部アプリケーションの構成方法を説明しますが、今はこのフレームワークだけで作業します。

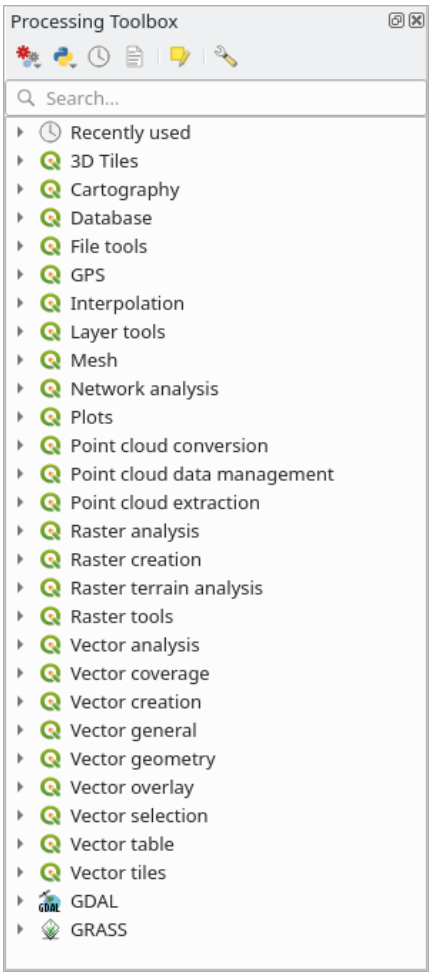
プロセシングフレームワークは QGIS のコアプラグインであり、QGIS に標準で含まれているため、システムに既にインストールされているはずです。それが有効になっている場合、メニューバーに**プロセシング**というメニューが表示されます。そこからフレームワークの全てのコンポーネントにアクセスできます。



そのメニューが見つからない場合は、プラグインマネージャに移動し、プラグインを有効にする必要があります。



作業に使う主な要素はツールボックスです。対応するメニュー項目をクリックすると、QGIS ウィンドウの右側にツールボックスがドッキングされます。



ツールボックスには、利用可能なすべてのアルゴリズムのリストが **プロバイダ** と呼ばれるグループに分類されて格納されています。プロバイダは、 **設定->オプション->プロセシング** で有効（無効）にできます。このダイアログについては、このマニュアルの後半で説明します。

デフォルトでは、サードパーティ製のアプリケーションに依存しないプロバイダだけ（つまり、実行するのに QGIS 要素だけを必要とするもの）が有効です。外部アプリケーションを必要とするアルゴリズムには、追加の設定が必要になる場合があります。プロバイダの設定は、このマニュアルの後の章で説明されています。

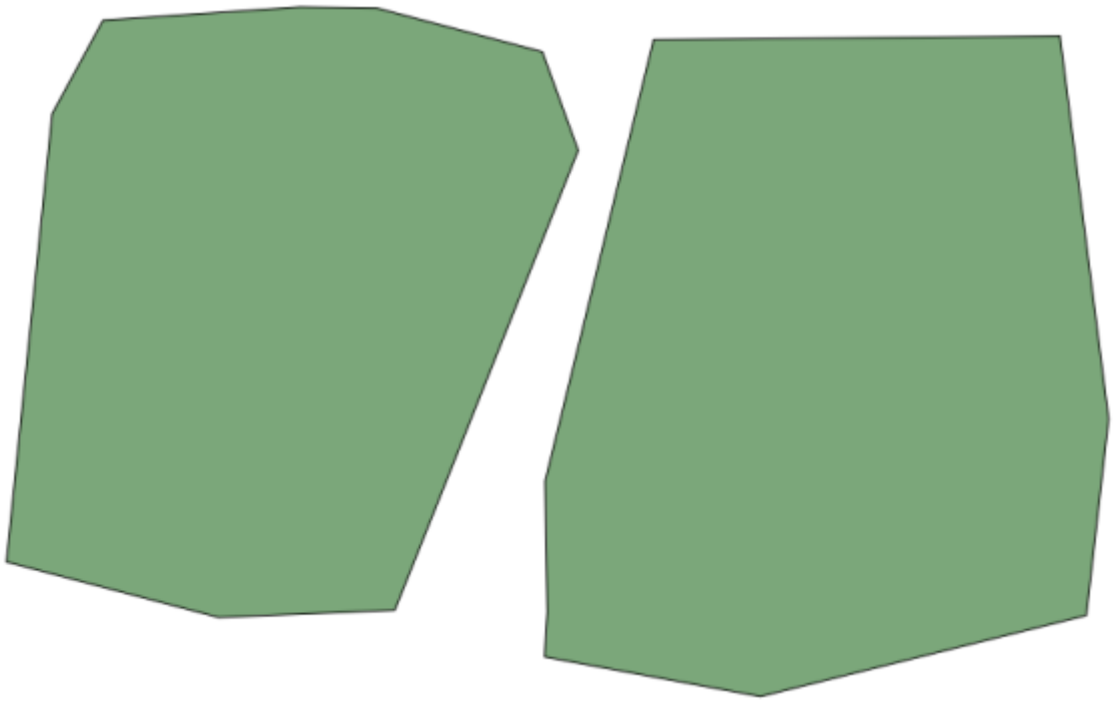
ここまで来ていればジオアルゴリズムを使う準備は整っています。現時点では他に何も設定する必要はありません。次のレッスンで行う最初のアルゴリズムの実行ができます。

17.4 最初のアルゴリズムを実行する・ツールボックス

> **注釈**
>
> このレッスンでは、最初のアルゴリズムを実行し、それから、最初の結果を取得します。

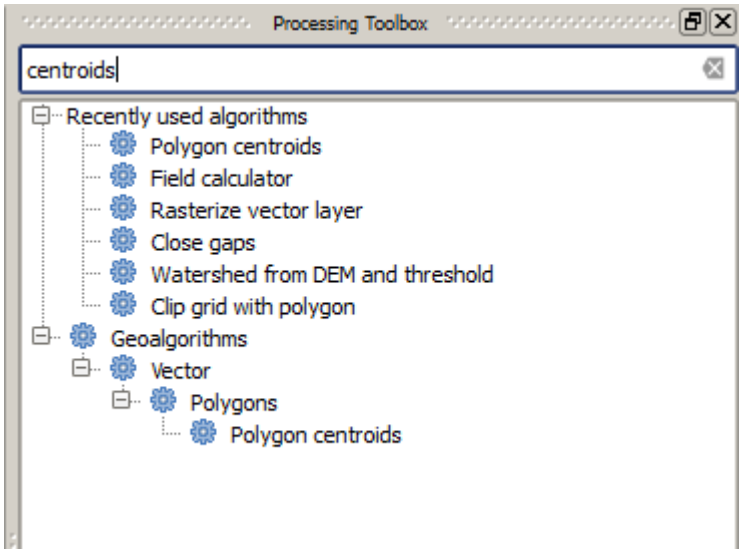
すでに述べたように、プロセシングフレームワークは他のアプリケーションのアルゴリズムを実行できますが、外部ソフトウェアを必要としないで実行できるネイティブのアルゴリズムも含んでいます。プロセシングフレームワークの探求を始めるため、ネイティブのアルゴリズムのひとつを実行します。具体的には、ポリゴンの集合の重心を計算します。

まず、このレッスンに対応する QGIS プロジェクト (first_alg) を開きます。これには、2 つのポリゴンを持つ単一のレイヤだけが含まれています

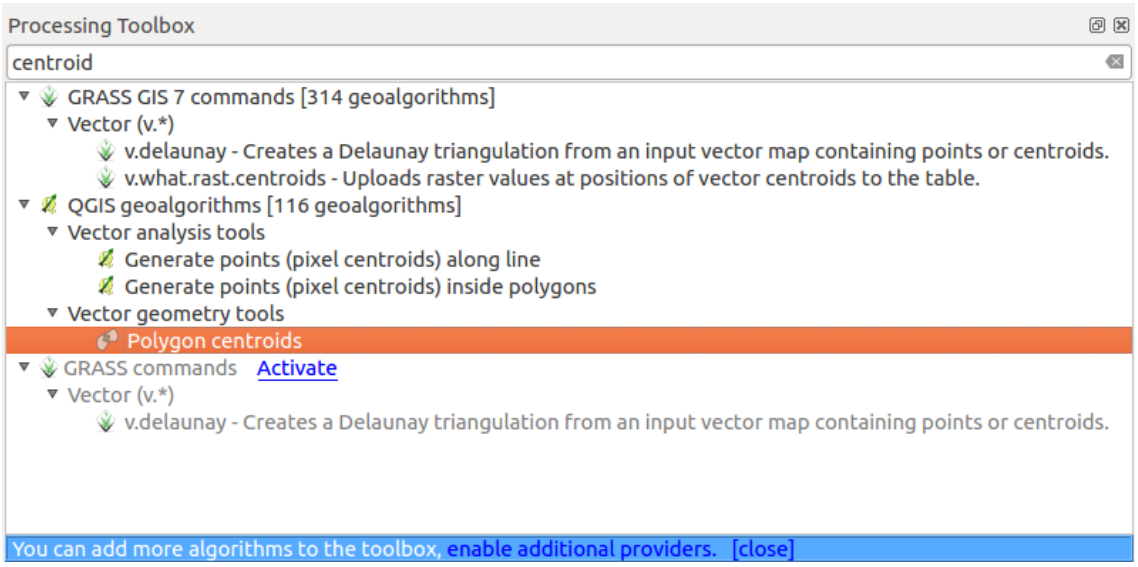


ツールボックスの上部にあるテキストボックスに移動してください。これが検索ボックスで、テキストを入力すると、入力されたテキストを含むアルゴリズムだけがリストからフィルタリングされて表示されます。検索に一致するアルゴリズムがあって、そのプロバイダがアクティブでない場合、ツールボックスの下部に追加のラベルが表示されます。

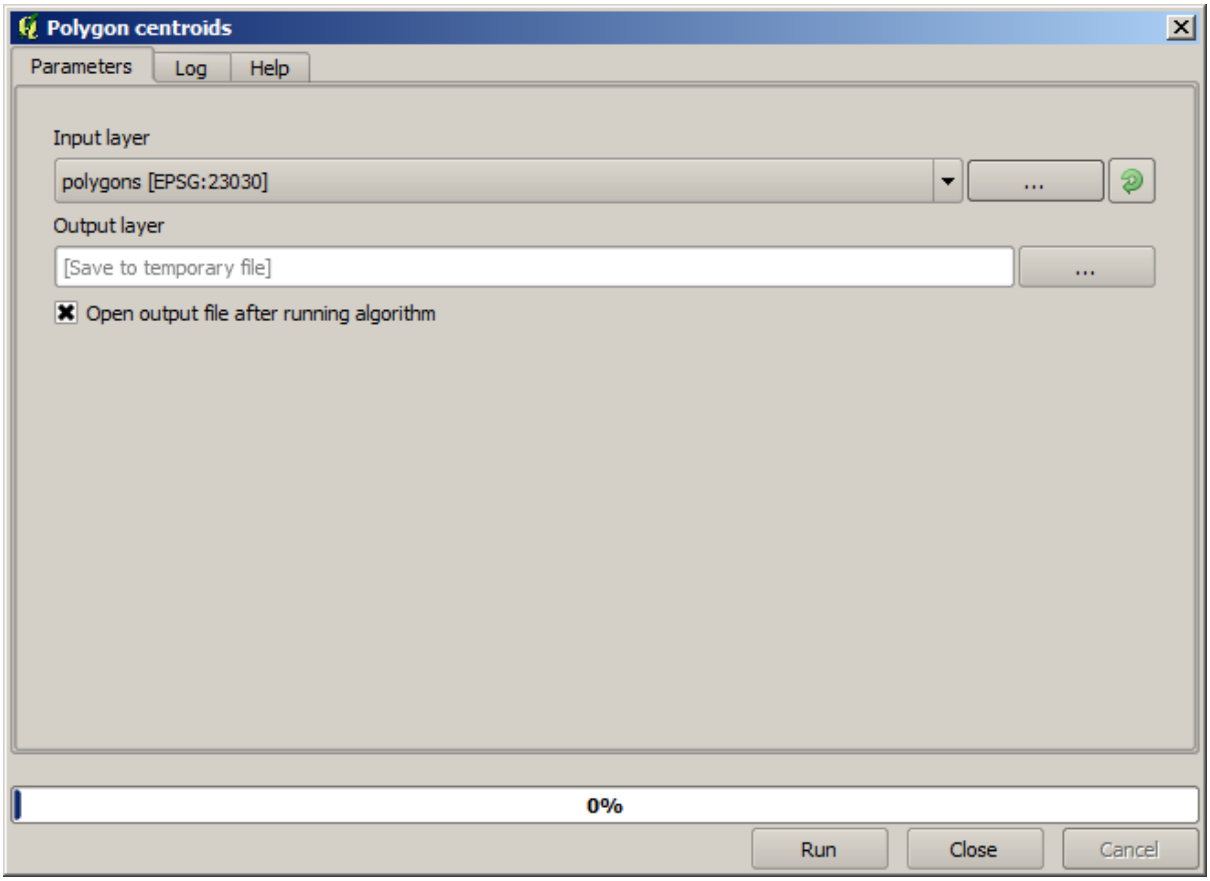
`centroids` と入力すると次のように見えるはずです。



検索ボックスは、探しているアルゴリズムを見つけるのにとても実用的な方法です。ダイアログの下部には、検索条件に一致するがアクティブではないプロバイダに属するアルゴリズムがあることを示す追加のラベルが表示されます。そのラベルのリンクをクリックすると、アルゴリズムのリストにアクティブでないプロバイダの結果も明るいグレーで表示されます。アクティブでないプロバイダをアクティブにするリンクもそれぞれに表示されます。その他のプロバイダをアクティブにする方法については、後で説明します。



To execute an algorithm, you just have to double-click on its name in the toolbox. When you double-click on the *Centroids* algorithm, you will see the following dialog.



どのアルゴリズムも似たようなインターフェイスを持っていて、基本的には、入力の必要な入力パラメータと、保存する場所を選択するアウトプットが含まれています。今回の場合、唯一の入力はポリゴンを持ったベクタレイヤです。

Select the `Polygons` layer as input. This also can be done by dragging and dropping the layer from the layers panel to the input box. The algorithm has a single output, which is the centroids layer. There are two options to define where a data output is saved: enter a filepath or save it to a temporary filename.

保存先を設定してかつ一時ファイルに結果を保存しないようにしたい場合には、出力の形式は、ファイル名の拡張子によって定義されます。形式を選択するには、単に対応するファイルの拡張子を選択します（または、直接ファイルパスを入力している場合は、代わりにそれを追加します）。入力したファイルパスの拡張子は、サポートされているもののいずれかと一致しない場合は、デフォルトの拡張子（通常はテーブルに対して `.dbf` 、ラスタレイヤに対して `.tif` ベクタレイヤに対して `.shp` ）がファイルパスに追加され、その拡張子に対応するファイル形式がレイヤまたはテーブルを保存するために使用されます。

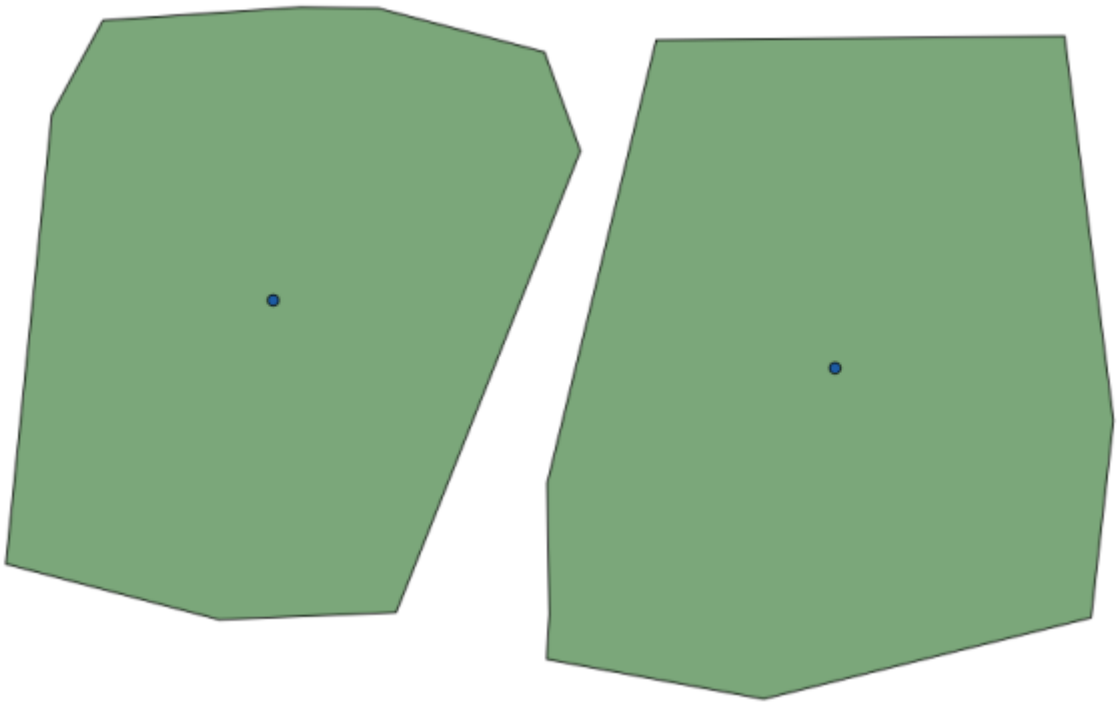
このガイドの全ての演習では、後で使うために保存する必要が無いので、結果は一時ファイルに保存しています。恒久的な場所に保存しても構いません。

警告

一時ファイルは QGIS を閉じると削除されます。一時的な出力として保存された出力を持つプロジェクトを作成した場合、後でそのプロジェクトを開き直そうとすると、出力ファイルが存在しないので、QGIS は文句を言うでしょう。

アルゴリズムダイアログを設定し終えたら **実行** を押してアルゴリズムを実行します。

次のような出力が得られます。



出力は入力と同じ CRS を持ちます。ジオアルゴリズムは、全ての入力レイヤが同じ CRS を共有していることを前提とし、再投影しません。（再投影アルゴリズムなど）特別なアルゴリズムを除き、出力も同じ CRS を持ちます。これについてはすぐに詳しく説明します。

自分でさまざまなファイル形式（例えば、拡張子として `shp` と `geojson` を使用）を使って保存してみてください。また、レイヤの生成後に QGIS にそれを読み込ませたくない場合は、出力パスボックスの下にあるチェックボックスのチェックを外してください。

17.5 さらなるアルゴリズムとデータタイプ

注釈

このレッスンでは、さらに 3 つのアルゴリズムを実行し、他の入力タイプを使用する方法を学習し、出力が指定したフォルダに自動的に保存されるように設定します。

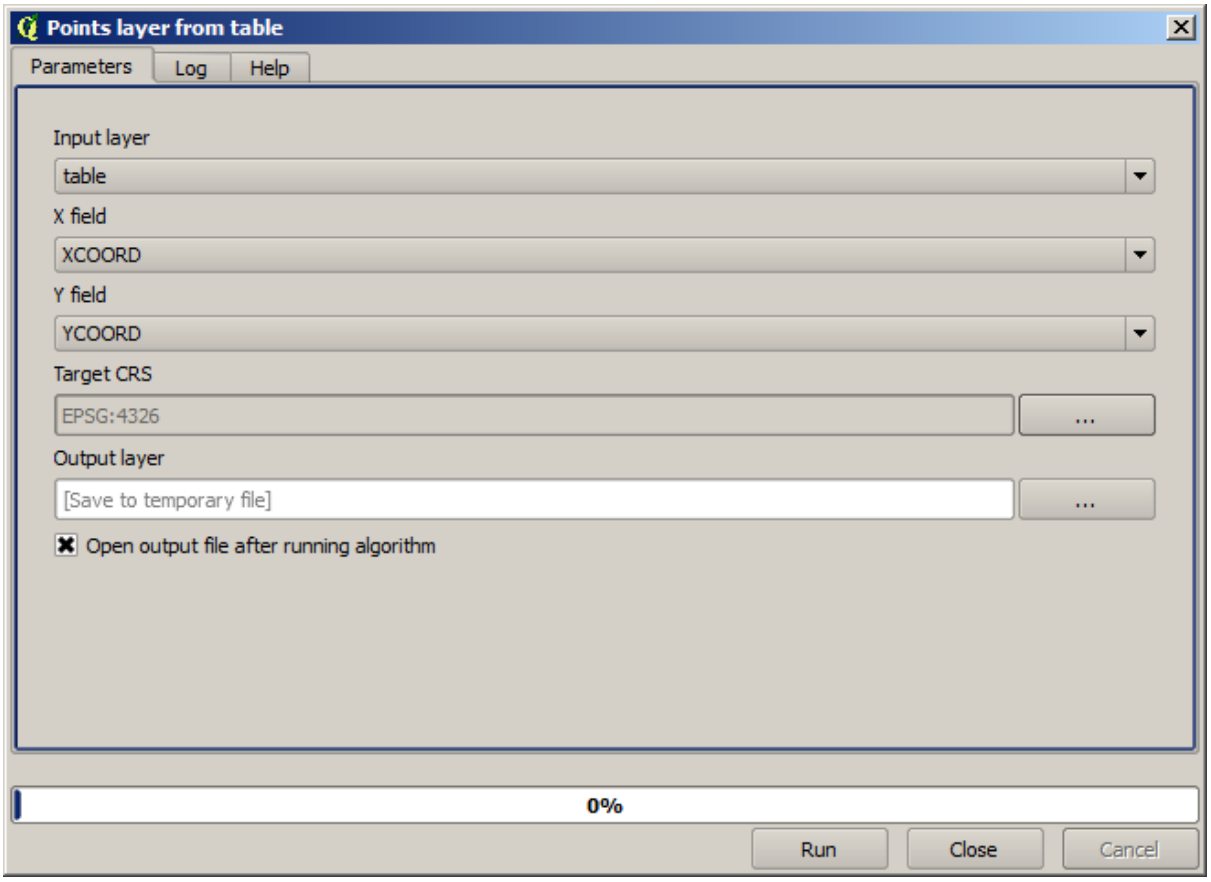
このレッスンにはひとつのテーブルとひとつのポリゴンレイヤが必要です。テーブル内の座標に基づいてポイントレイヤを作成し、各ポリゴン内のポイントの数を数えます。このレッスンに対応する QGIS プロジェクト (second_alg) を開くと、X と Y 座標を持ったテーブルがありますが、ポリゴンレイヤはありませんが心配しないでください。これからプロセシング・ジオアルゴリズムを使って作成します。

The first thing we are going to do is to create a points layer from the coordinates in the table, using the *Create points layer from table* algorithm. You now know how to use the search box, so it should not be hard for you to find it. Double-click on it to run it and get to its following dialog.

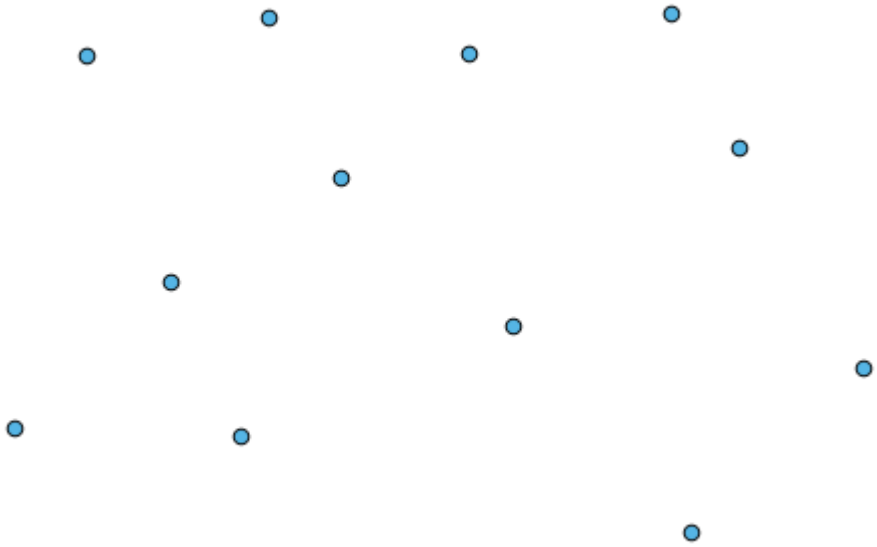
このアルゴリズムは、前のレッスンのようにただ 1 つの出力を生成し、また、3 つの入力を持っています：

- **テーブル** ：座標を持つテーブル。ここでレッスンのデータからテーブルを選択する必要があります。
- *X and Y fields*: these two parameters are linked to the first one. The corresponding selector will show the name of those fields that are available in the selected table. Select the `XCOORD` field for the *X* parameter, and the `YYCOORD` field for the *Y* parameter.
- *CRS* ：このアルゴリズムでは入力レイヤに何も取らないので、それに基づいた CRS を出力レイヤへ割り当てることができません。代わりに、テーブルの座標に使われている CRS を手動で選択するように求められます。左側のボタンをクリックして QGIS CRS セレクタを開き、出力 CRS として EPSG:4326 を選択してください。テーブル内の座標がその CRS なので、この CRS を使用しています。

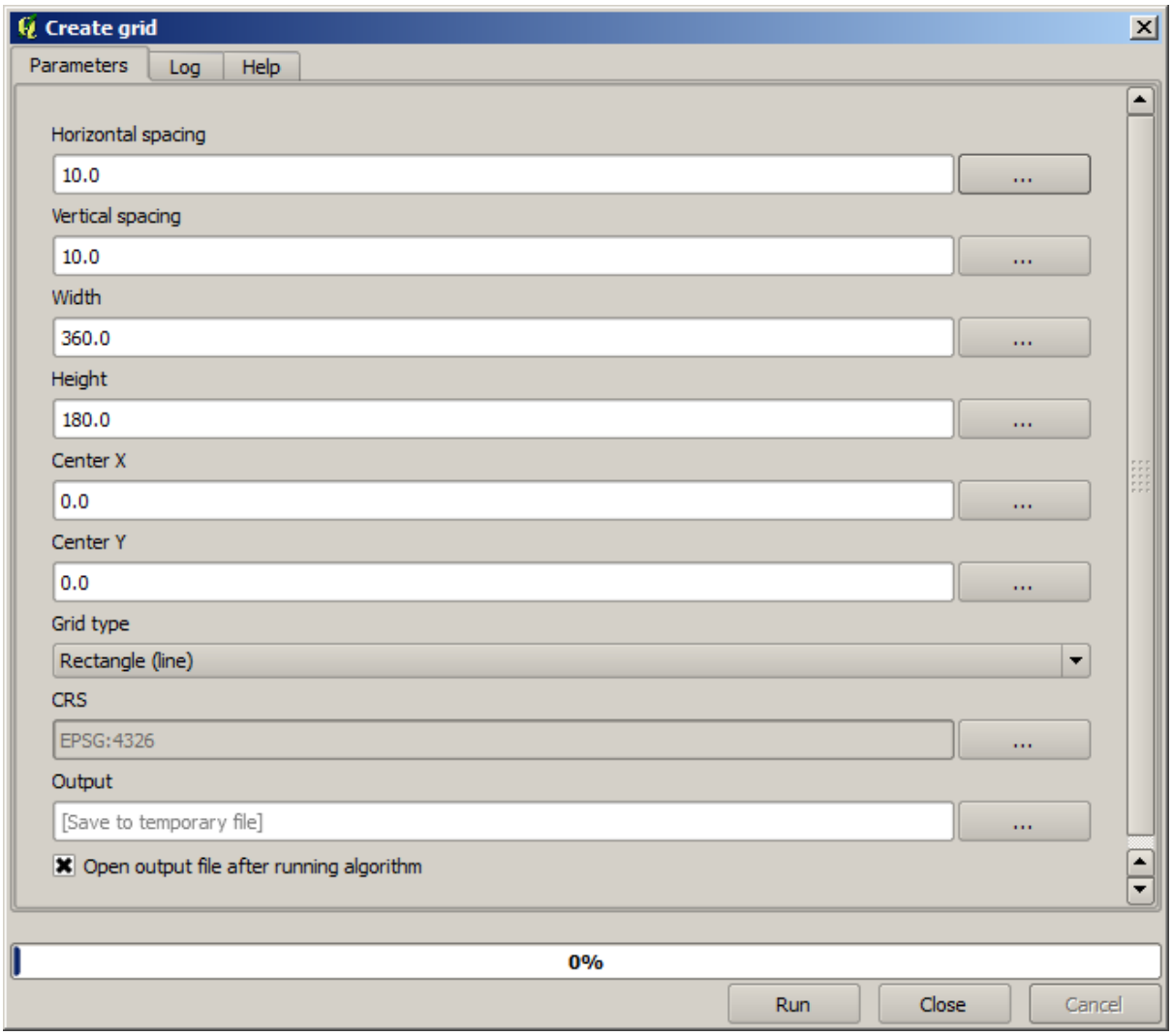
ダイアログは次のようになります。



Now press the *Run* button to get the following layer (you may need to zoom full to reenter the map around the newly created points):



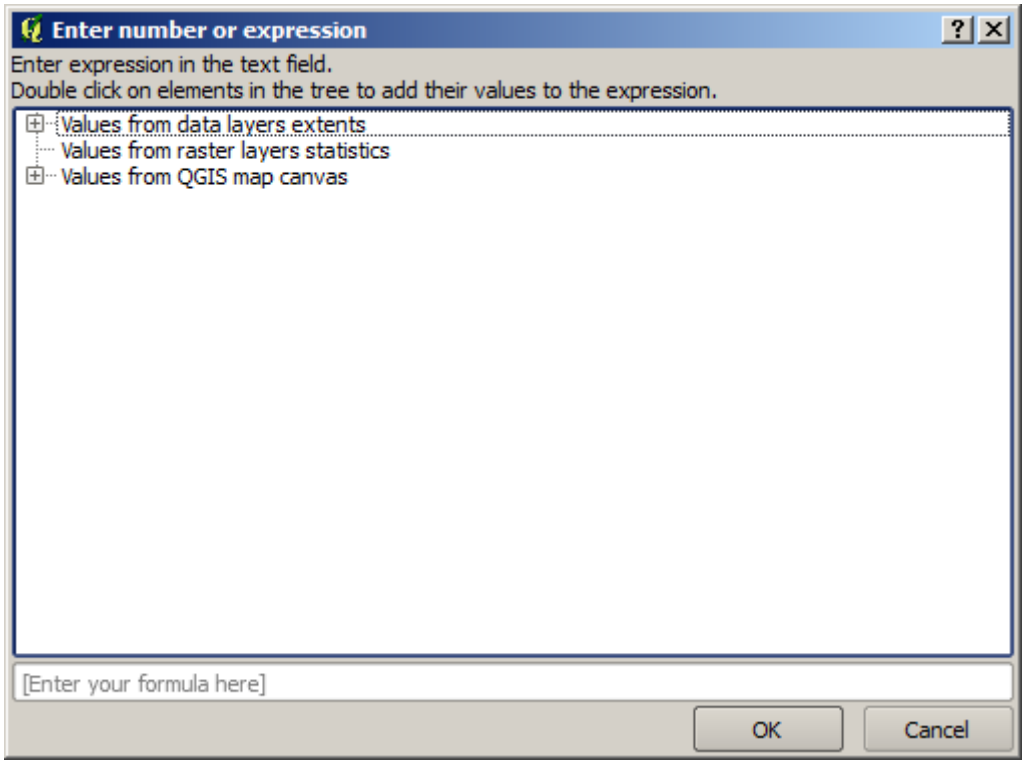
The next thing we need is the polygon layer. We are going to create a regular grid of polygons using the *Create grid* algorithm, which has the following parameters dialog.



警告

QGIS の最近のバージョンではオプションが単純になりました。X と Y の最小値と最大値を入力するだけです（推奨値：-5.696226, -5.695122, 40.24742, 40.248171）

グリッドを作成するために必要な入力はすべて数値です。数値を入力する必要がある場合、次の 2 つのオプションがあります: 対応するボックスに直接入力するか、右にあるボタンをクリックして次のようなダイアログボックスを表示させます。



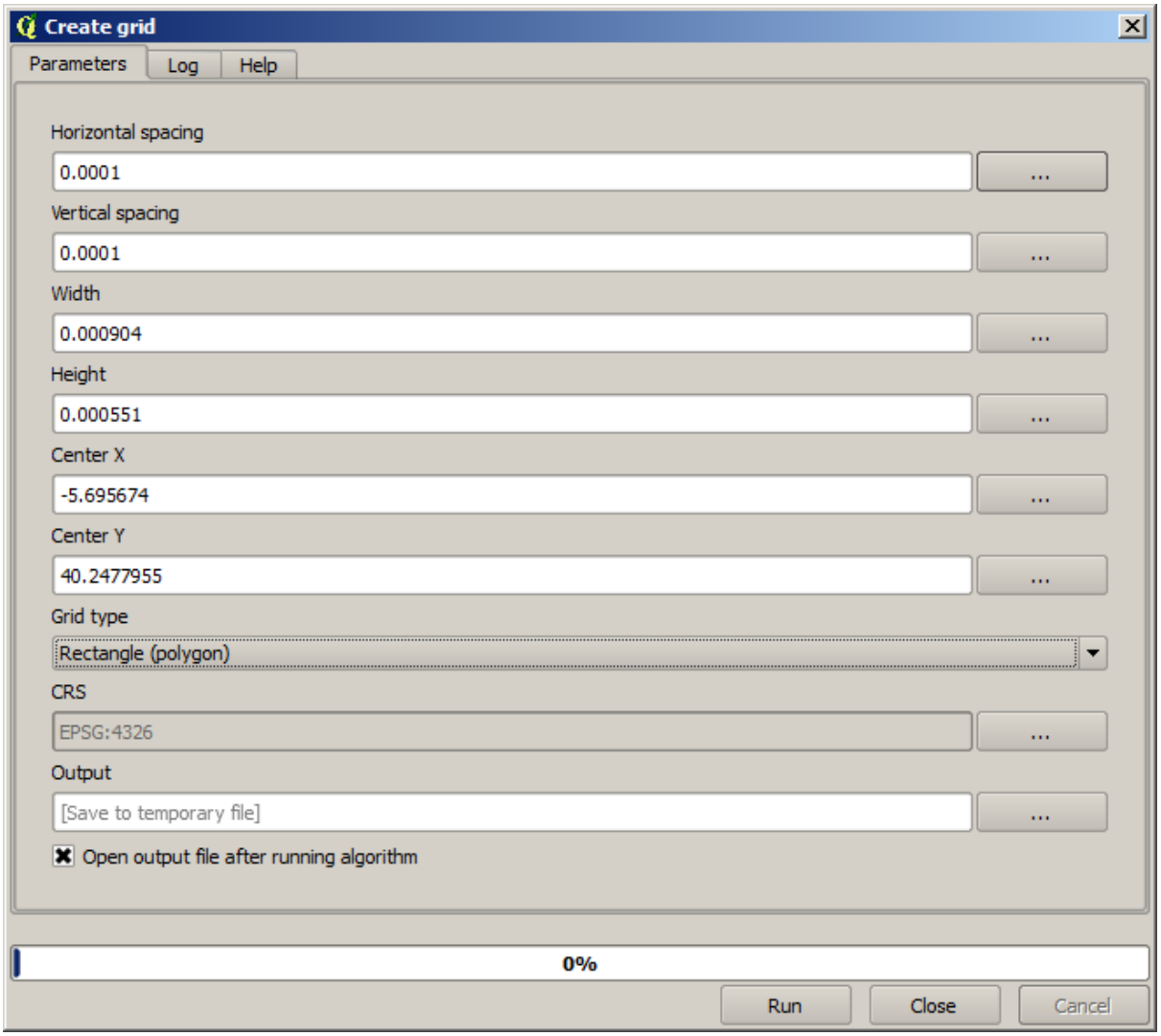
このダイアログには簡単な計算機が含まれており、`11 * 34.7 + 4.6` のような式を入力すると、計算結果がパラメータダイアログの対応するテキストボックスに入ります。また、それには定数や利用可能な他のレイヤの値も含んでいます。

ここでは入力ポイントレイヤの範囲を覆うグリッドを作成したいので、それらの座標を使い、アルゴリズムがグリッドを作成するのに要するパラメータであるグリッドの中心座標と幅と高さを計算する必要があります。少し数学が必要ですが、計算機ダイアログと入力ポイントレイヤの定数を使って自分でやってみましょう。

Select *Rectangles (polygons)* in the *Grid type* field.

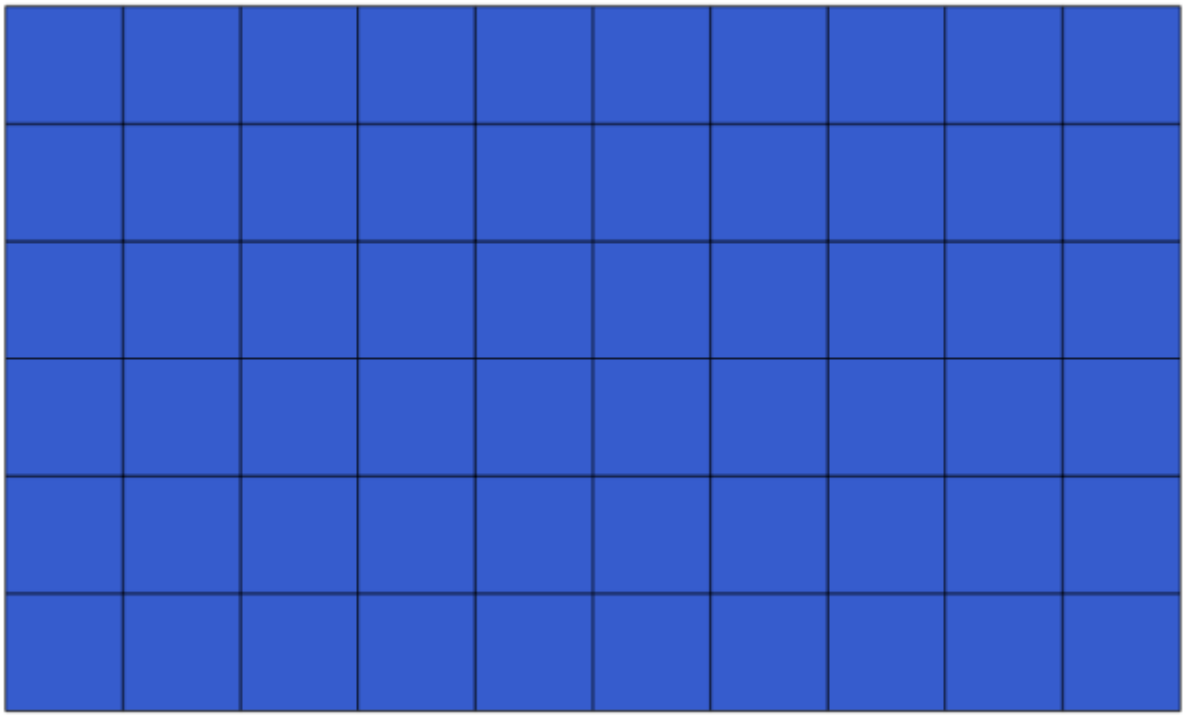
最後のアルゴリズムの場合のように、ここにも CRS を入力する必要があります。前に行ったように、ターゲット CRS として EPSG:4326 を選択します。

最終的に、次のようなパラメータダイアログになるはずです:

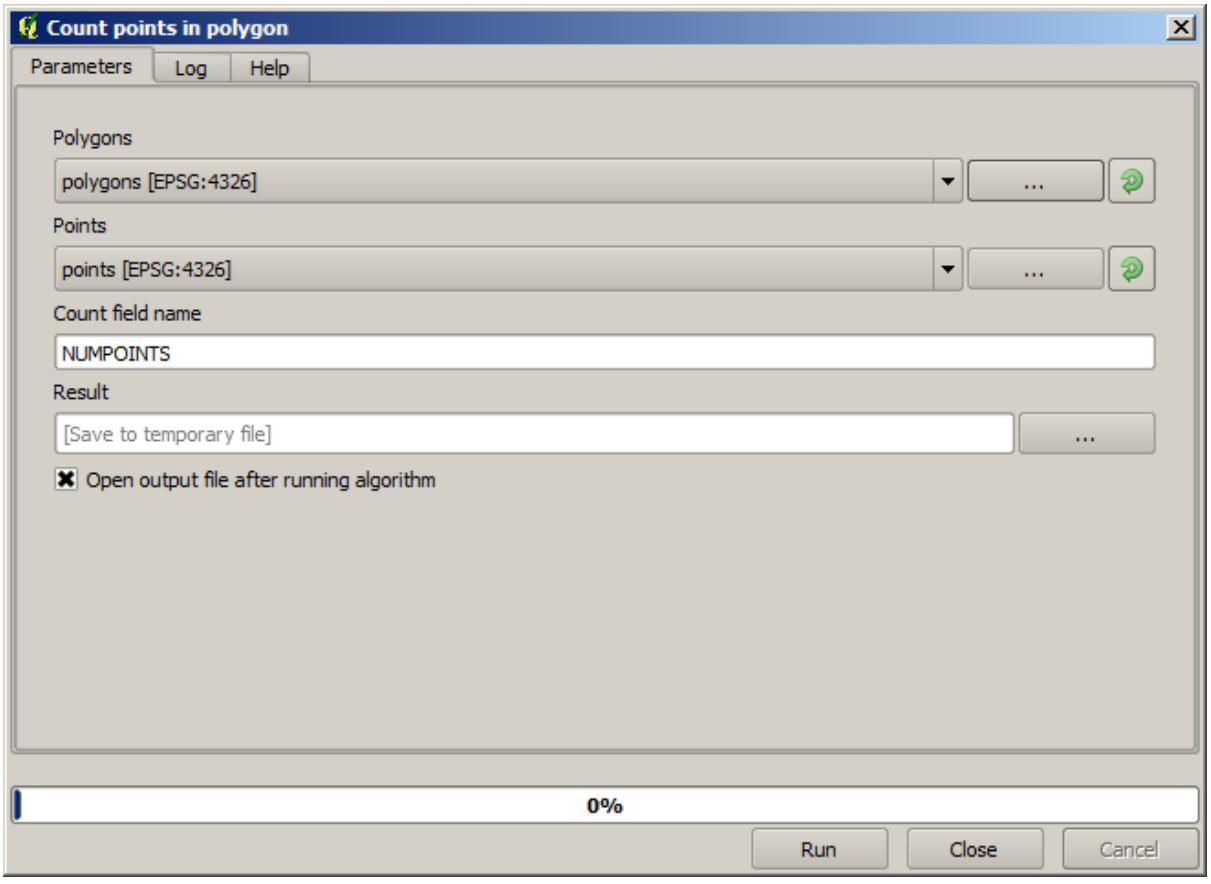


(Better add one spacing on the width and height: *Horizontal spacing*: `0.0001`, *Vertical spacing*: `0.0001`, *Width*: `0.001004`, *Height*: `0.000651`, *Center X*: `-5.695674`, *Center Y*: `40.2477955`) The case of X center is a bit tricky, see: -5.696126+((-5.695222+ 5.696126)/2)

Press *Run* and you will get the graticule layer.

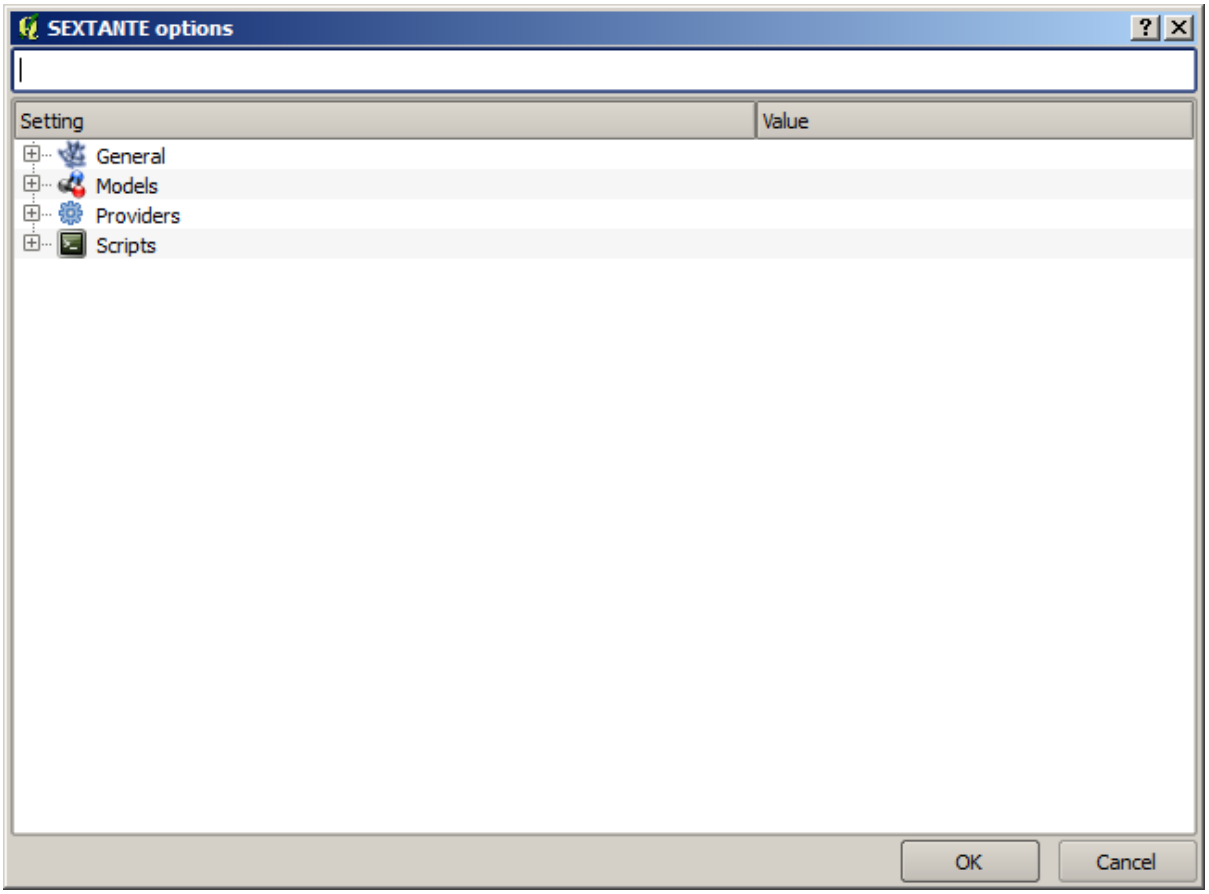


The last step is to count the points in each one of the rectangles of that graticule. We will use the *Count points in polygons* algorithm.

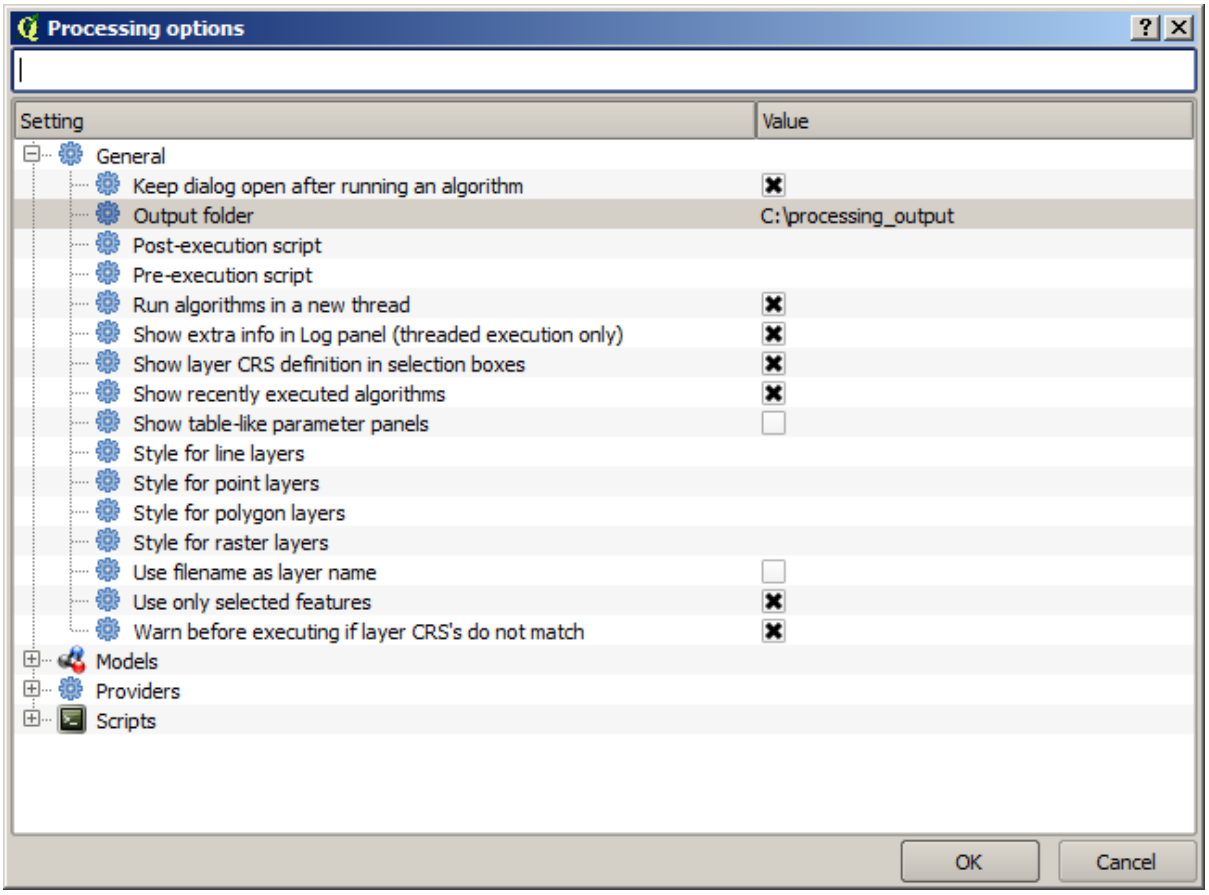


これで求めていた結果が得ることができました。

このレッスンを終える前に、データを永続的に保存したい場合に役立つ簡単なヒントを紹介します。全ての出力ファイルを特定のフォルダに保存したい場合、フォルダ名を毎回入力する必要はありません。代わりに、プロセシングメニューに移動し、 **オプションと設定** を選択します。すると設定ダイアログが開きます。



In the *Output folder* entry that you will find in the *General* group, type the path to your destination folder.



Now when you run an algorithm, just use the filename instead of the full path. For instance, with the configuration shown above, if you enter `graticule.shp` as the output path for the algorithm that we have just used, the result will be saved in `D:processing_outputgraticule.shp`. You can still enter a full path in case you want a result to be saved in a different folder.

Try yourself the *Create grid* algorithm with different grid sizes, and also with different types of grids.

17.6 CRS・再投影

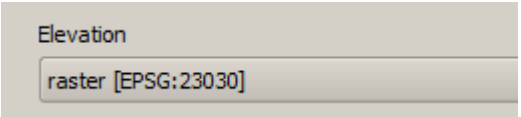
注釈

この課題で、私たちはプロセシングがどのように CRS を使うかを説明します。私たちはまた、再投影というとても便利なアルゴリズムについて見てみます。

CRS は QGIS プロセシングのユーザーにとって大いなる混乱の源です。そこで、新しいレイヤを作成するときにジオアルゴリズムがそれをどのように扱うかについて、いくつかの一般的なルールを紹介します。

- 入力レイヤが複数存在する場合は、最初のレイヤの CRS が使用されます。全ての入力レイヤの CRS は同じ必要があり、これがその CRS であると想定されます。CRS が一致していないレイヤを使うと、

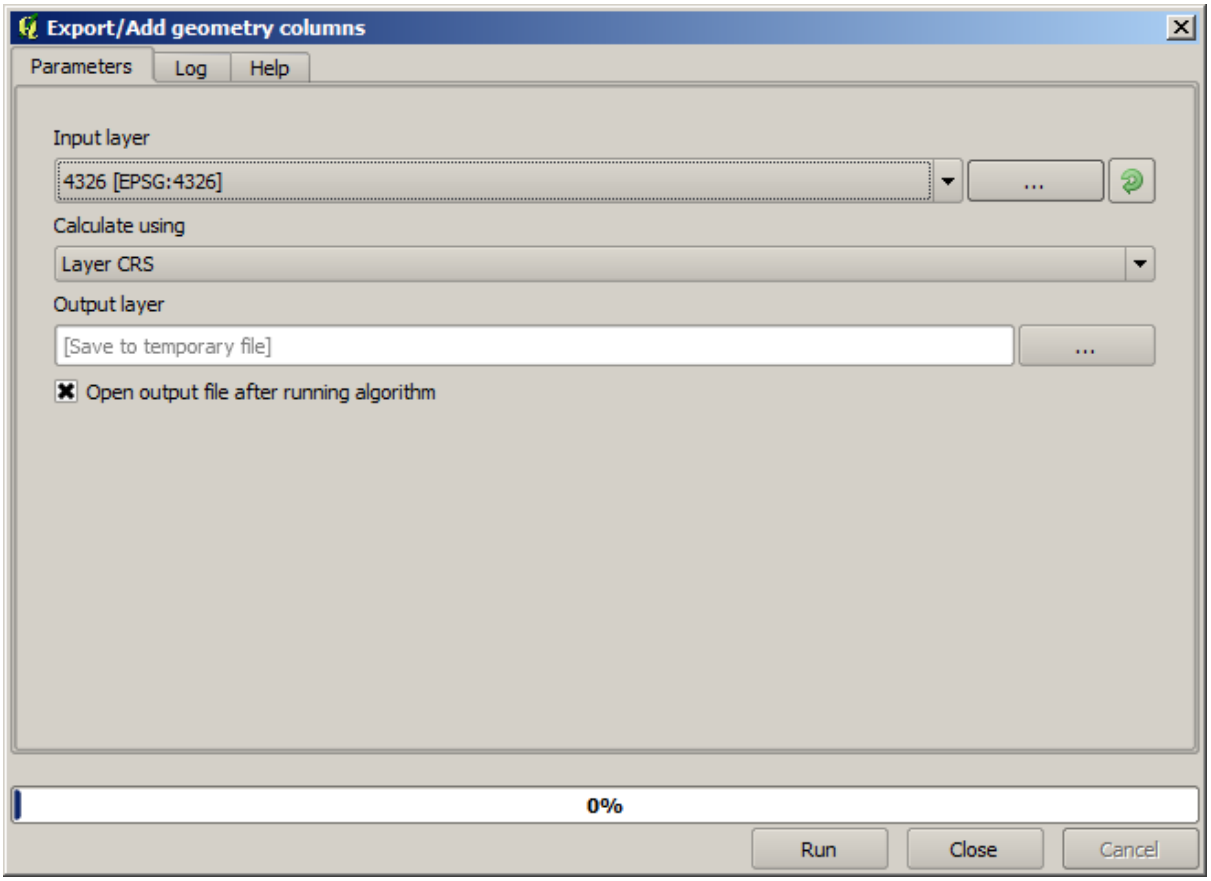
QGIS がそれを警告します。入力レイヤの CRS はパラメータダイアログ中にその名前と一緒に表示されていることに注意してください。



- 入力レイヤがない場合はプロジェクトの CRS が使用されますが、アルゴリズムに具体的な CRS フィールドが含まれていれば、(経緯線網アルゴリズムを使った前回のレッスンのように) それが使われます

Open the project corresponding to this lesson and you will see two layers named `23030` and `4326`. They both contain the same points, but in different CRSs (`EPSG:23030` and `EPSG:4326`). They appear in the same place because QGIS is reprojecting on the fly to the project CRS (`EPSG:4326`), but they are not actually the same layer.

Open the *Add geometry attributes* algorithm.



このアルゴリズムは、ベクタレイヤの属性テーブルに新しい列を追加します。列の内容は、レイヤのジオメトリの種類によって異なります。ポイントの場合、各点の X 及び Y 座標で新しい列を追加します。

入力レイヤのフィールドにある使用可能なレイヤのリストには、各レイヤとそれに対応する CRS が表示されています。それは、キャンバスの同じ場所に表示されても異なる方法で処理されることを意味します。4326 レイヤを選択してください。

The other parameter of the algorithm allows to set how the algorithm uses coordinates to calculate the new value that it will add to the resulting layers. Most algorithms do not have an option like that, and just use the coordinates

directly. Select the *Layer CRS* option to just use coordinates as they are. This is how almost all geoalgorithms work.

他の 2 つのレイヤとまったく同じポイントを持つ新しいレイヤが作成されるはずです。レイヤの名前を右クリックしてプロパティを開くと、入力レイヤと同じ CRS、つまり EPSG:4326 を共有していることが分かります。このレイヤが QGIS に読み込まれる際、レイヤの CRS は QGIS がすでに知っているので、それを入力するよう求められることはありません。

新しいレイヤの属性テーブルを開くと、それが各ポイントの X と Y 座標を持つ 2 つの新しいフィールドが含まれていることが分かります。

	ID	PT_NUM_A	PT_ST_A	xcoord	ycoord
0	1	1.100000	a	-5.695426	40.248071
1	2	2.200000	b	-5.695885	40.247622
2	3	3.300000	c	-5.695406	40.247520
3	4	4.400000	a	-5.695222	40.247694
4	5	5.500000	b	-5.695642	40.248030
5	6	6.600000	a	-5.695855	40.248067
6	7	7.700000	b	-5.696049	40.248028
7	8	8.800000	c	-5.696126	40.247629
8	9	9.900000	a	-5.695961	40.247786
9	10	11.000000	b	-5.695353	40.247929
10	11	12.100000	a	-5.695595	40.247739
11	12	13.200000	b	-5.695779	40.247896

それらの座標値は、そうするオプションを選んだので、そのレイヤの CRS で与えられています。しかし他のオプションを選んでも、入力 CRS は出力レイヤの CRS を設定するのに使われるので、レイヤの出力 CRS は同じであったはずです。他のオプションを選ぶと、値は異なりますが、結果の点が変わったり、出力レイヤの CRS が入力レイヤの CRS と異なったりすることはありません。

もう一方のレイヤを使って、同じ計算をしてみてください。出来上がったレイヤは他のレイヤと全く同じ場所にレンダリングされ、入力レイヤと同じ EPSG:23030 CRS を持つはずです。

その属性テーブルに行くと、最初に作成したレイヤのものとは異なる値が表示されるでしょう。

	ID	PT_NUM_A	PT_ST_A	xcoord	ycoord
0	1	1.100000	a	270839.655869	4458983.162670
1	2	2.200000	b	270799.116425	4458934.552874
2	3	3.300000	c	270839.468187	4458921.978139
3	4	4.400000	a	270855.745301	4458940.799487
4	5	5.500000	b	270821.164389	4458979.173980
5	6	6.600000	a	270803.157564	4458983.848803
6	7	7.700000	b	270786.542791	4458980.047841
7	8	8.800000	c	270778.601980	4458935.968837
8	9	9.900000	a	270793.142411	4458952.931700
9	10	11.000000	b	270845.414756	4458967.311298
10	11	12.100000	a	270824.166376	4458946.784250
11	12	13.200000	b	270809.035643	4458964.649799

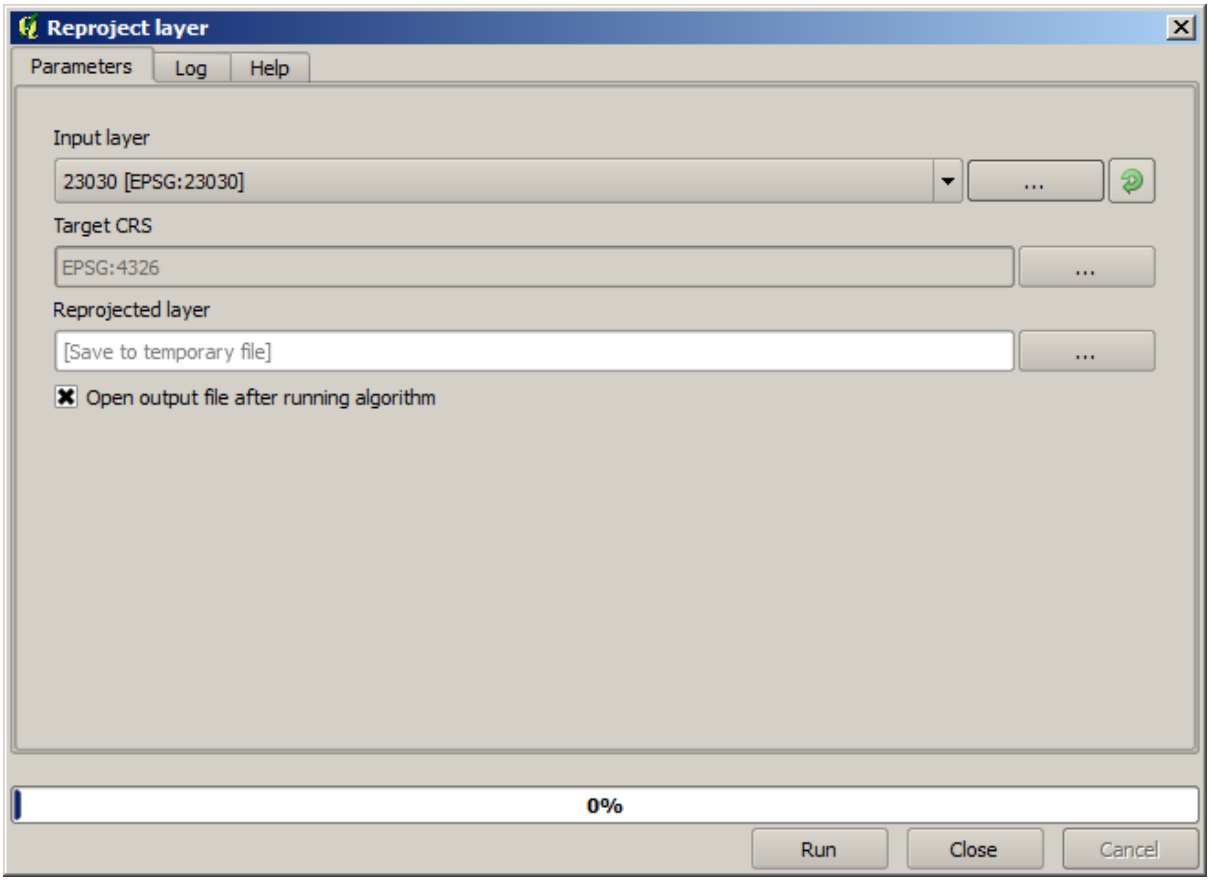
これは元のデータが異なる（異なる CRS を使用している）ためであり、座標はそれから取られています。

ここから何を学ぶべきでしょうか？これらの例の背後にある重要な考えは、地理アルゴリズムは元のデータソースのレイヤをそのまま使用し、QGIS でレンダリングする前に行われているかもしれない再投影は完全に無視しているということです。言い換えれば、キャンバスに見えているものは信用せず、常に元のデータが使われることを念頭に置いてください。ここでは一度に 1 つのレイヤだけを使用しているのでそれほど重要ではありませんが、複数のレイヤを必要とするアルゴリズム（例えばクリップアルゴリズム）では、一致したり重なったりするように見えるレイヤが、異なる CRS を持っていて、互いに非常に離れていることがあります。

アルゴリズムは（次に見る再投影アルゴリズムを除いて）再投影を行わないので、レイヤの CRS が一致していることを確認することは自分の責任です。

CRS を扱う興味深いモジュールのひとつが再投影モジュールです。それが特殊なケースを代表するのは、入力レイヤ（再投影の対象となるもの）を持つが、その CRS を出力レイヤに使わないからです。

Open the *Reproject layer* algorithm.



Select any of the layers as input, and select EPSG:23029 as the destination CRS. Run the algorithm and you will get a new layer, identical to the input one, but with a different CRS. It will appear on the same region of the canvas, like the other ones, since QGIS will reproject it on the fly, but its original coordinates are different. You can see that by running the *Add geometry attributes* algorithm using this new layer as input, and verifying that the added coordinates are different to the ones in the attribute tables of both of the two layers that we had computed before.

17.7 選択

注釈

このレッスンでは、プロセシングアルゴリズムが入力として使用されるベクタレイヤの選択範囲をどのように扱うか、そしてどのようにアルゴリズムの特定のタイプを使用して選択範囲を作成するかについて説明します。

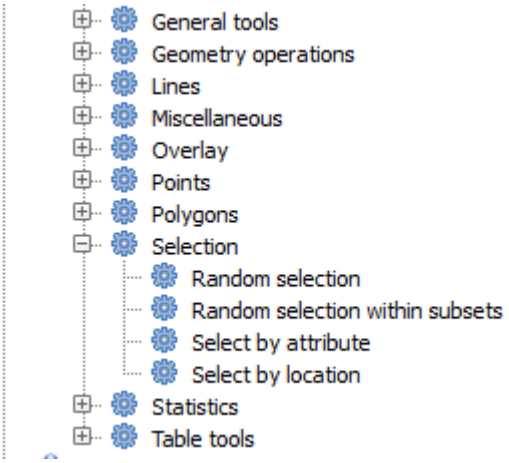
QGIS の他の分析プラグインと異なり、プロセシング・地理アルゴリズムには「選択した地物だけを使用する」チェックボックスや同様のオプションは存在しません。選択範囲に関する動作は、各アルゴリズムの実行ごとに設定されるものではなく、プラグイン全体とその全てのアルゴリズムに対して設定されます。アルゴリズムがベクタレイヤを使うとき、次の単純な規則に従います。

- レイヤに選択範囲がある場合は、選択されている地物のみが使われます。
- 選択範囲がない場合は、全ての地物が使われます。

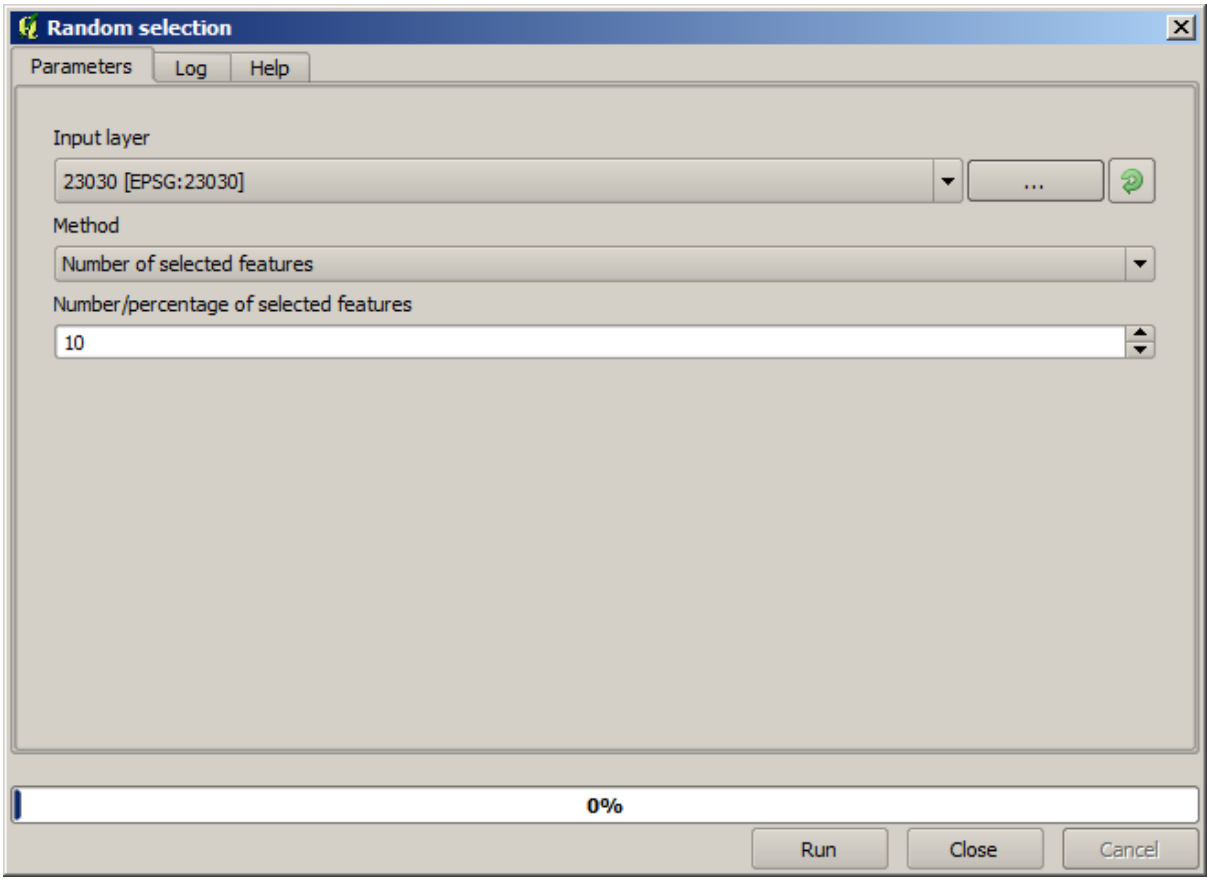
プロセシング　オプション　一般 メニュー中の関連するオプションを選択解除すると、この動作を変更できるので、注意してください。

それは、前の章で使用されるレイヤのいずれかでポイントをいくつか選択し、それらに再投影アルゴリズムを実行することでテストできます。得られる再投影レイヤには選択されたポイントのみが含まれます。ただし何も選択がない場合は、結果レイヤには、元のレイヤの全てのポイントが含まれることになります。

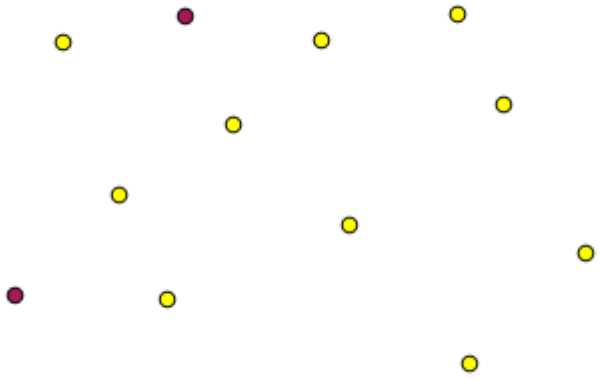
選択範囲を作るには、QGIS で利用可能な方法やツールのどれでも使えます。そのために地理アルゴリズムを使うこともできます。選択範囲を作るためのアルゴリズムは、ツールボックスの **ベクタ選択** にあります



Open the *Random selection* algorithm.



デフォルト値のままにすると、現在のレイヤから 10 点が選択されます。



このアルゴリズムは何の出力も作らず、入力レイヤ（レイヤ自体ではなく、その選択範囲）が変更されることが分かるでしょう。他の全てのアルゴリズムは新たなレイヤを作成し、入力レイヤを変更しないので、これは珍しい動作です。

選択範囲はデータ自体の一部ではなく、QGIS 内にのみ存在するものなので、これらの選択アルゴリズムは、QGIS で開いているレイヤを選択するためにのみ使われ、対応するパラメータ値ボックスにあるファイル選択オプションでは使用できません。

先ほど作った選択範囲は、他の選択アルゴリズムによって作成されるものと同じように、QGIS から手動で行うこともできるので、なぜアルゴリズムを使う必要があるのか疑問に思うかもしれません。今はあまりピンとこないかもしれませんが、後でモデルとスクリプトを作成する方法を説明します。（プロセシングワークフローを定義する）モデルの途中で選択範囲を作りたい場合、モデルに追加できるのは地理アルゴリズムだけで、その他の QGIS 要素と操作は追加できません。これが一部のプロセシングアルゴリズムが他の QGIS 要素でも利用できる機能を重複している理由です。

今のところはプロセシング・地理アルゴリズムを使って選択範囲を作れること、アルゴリズムは、選択範囲があるときは選択された地物だけを、それ以外は全ての地物を使うことだけを覚えておいてください。

17.8 外部アルゴリズムを実行する

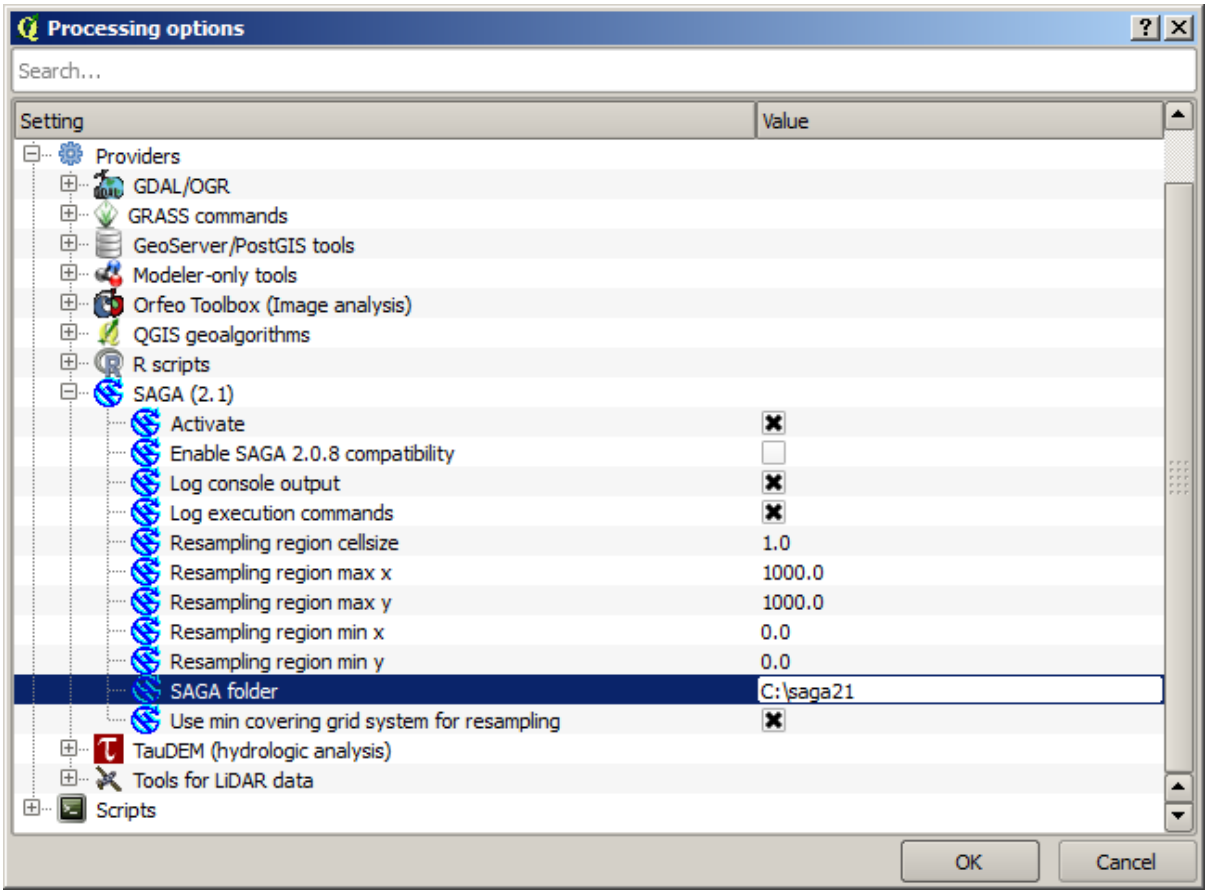
注釈

このレッスンではサードパーティのアプリケーション、特に主要なアルゴリズムプロバイダーの一つである SAGA に依存するアルゴリズムを使用する方法について説明します。

これまで実行したすべてのアルゴリズムは、プロセシングフレームワークの一部です。つまり、プラグインに実装された **ネイティブ** なアルゴリズムであり、プラグイン自体が実行されるのと同じように QGIS によって実行されます。しかし、プロセシングフレームワークの最大の特徴は、外部アプリケーションのアルゴリズムを使用し、それらのアプリケーションの可能性を拡張することができるということです。そのようなアルゴリズムはラップされてツールボックスに含まれているため、QGIS から簡単に利用することができ、QGIS のデータを使って実行することができます。

単純化されたビューに表示されるアルゴリズムには、サードパーティ製のアプリケーションがシステムにインストールされていることが必要なものがあります。特に注目すべきアルゴリズムプロバイダの一つが SAGA（自動地理空間分析のためのシステム）です。まず、QGIS から SAGA を正しく呼び出せるように、全ての設定を行う必要があります。これは難しい作業ではありませんが、仕組みを理解することが重要です。外部アプリケーションにはそれぞれ独自の設定があり、このマニュアルの後半で他についても説明しますが、SAGA は主要なバックエンドになるものなので、ここでは詳細に説明します。

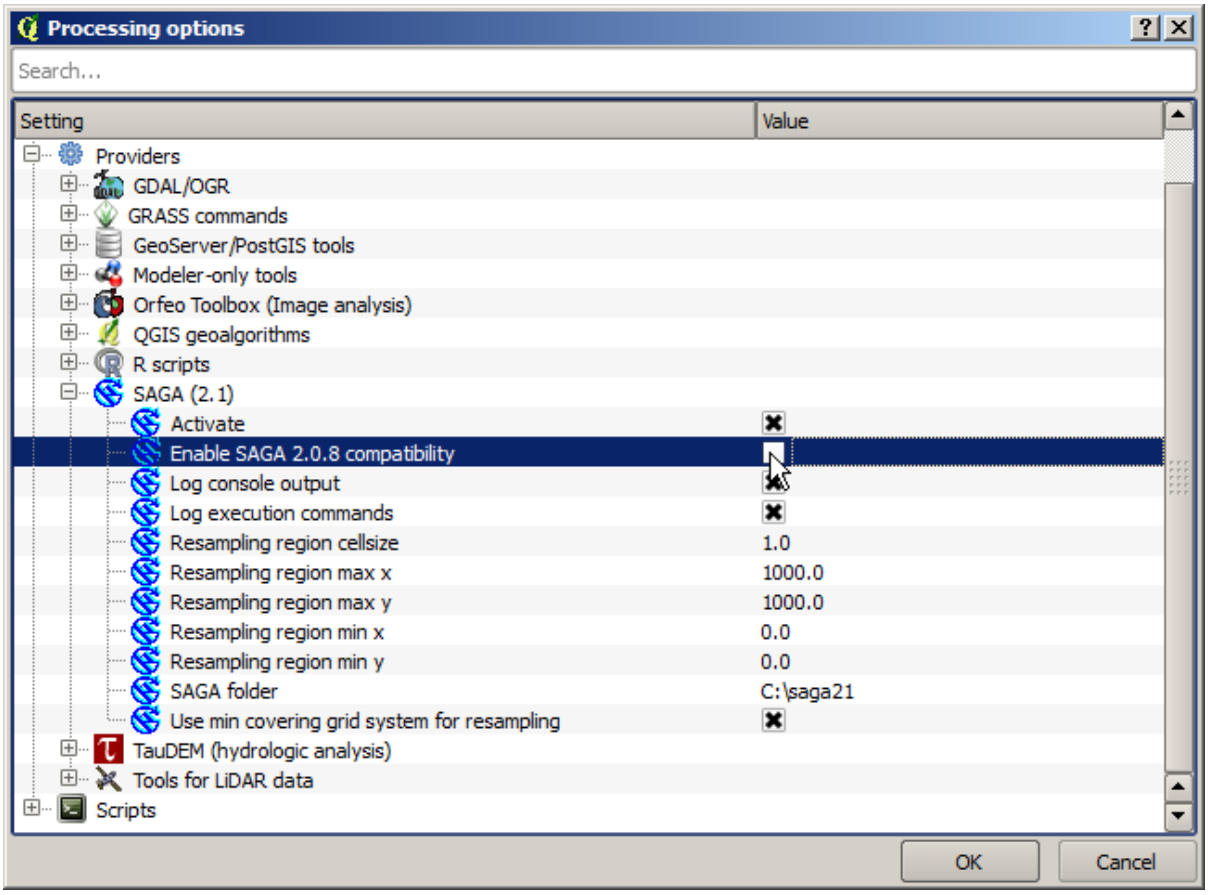
Windows を使っている場合は、外部アルゴリズムを使う最良の方法は、スタンドアロンのインストーラを使って QGIS をインストールすることです。これにより、SAGA を含む必要な全ての依存関係が自動的にインストールされるので、他に作業はありません。設定ダイアログを開き、 **プロバイダ**/*SAGA* グループに移動してください。



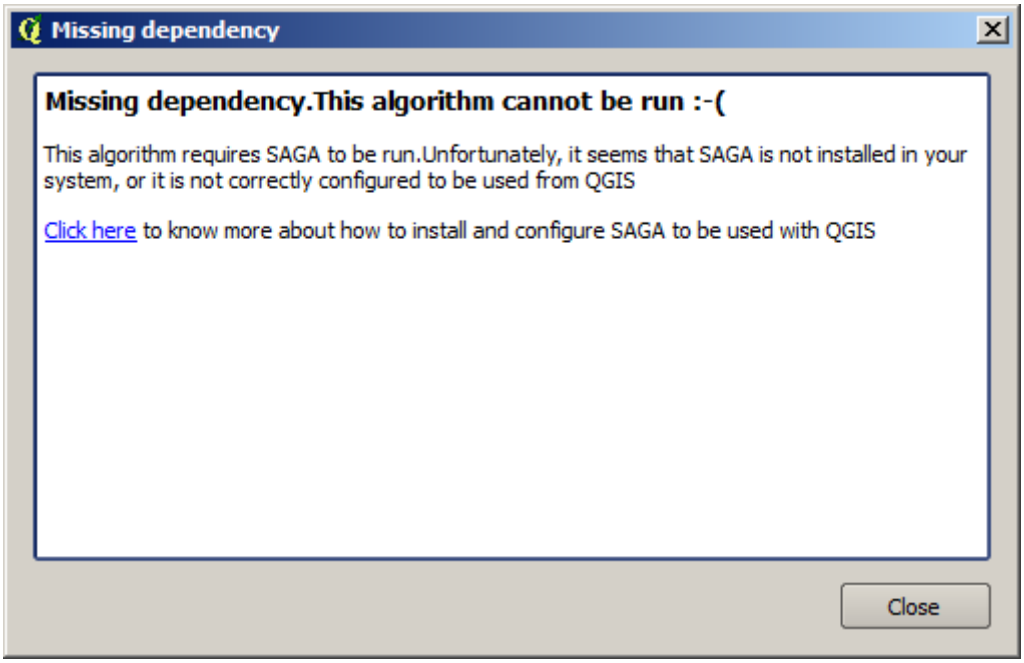
SAGA パスは既に設定され、SAGA がインストールされているフォルダを指しているようにしてください

スタンドアロンのインストーラを使用しないで QGIS をインストールした場合は、そこに（別個にインストールした）SAGA のインストールへのパスを入力する必要があります。必要なバージョンは、SAGA 2.1 [これは SAGA のリリースに応じて変化します] です。

Linux を使用している場合は、プロセシング構成で SAGA のインストールへのパスを設定する必要はありません。代わりに、SAGA をインストールし、それがコンソールから呼び出すことができるよう SAGA フォルダが、PATH にあることを確認する必要があります（それを確認するため、コンソールを開いて `saga_cmd` を入力するだけ）。Linux では、SAGA のターゲットバージョンも 2.1 ですが、（OSGeo ライブ DVD など）一部のインストールでは 2.0.8 だけが利用できる可能性があります。2.1 パッケージが利用可能なものがありますが、それらは一般的にインストールされておらず、いくつか問題があるかもしれないので、より一般的で安定した 2.0.8 を使いたい場合は、 *SAGA* グループ下の設定ダイアログで 2.0.8 の互換性を可能にすることによって、それをすることができます。

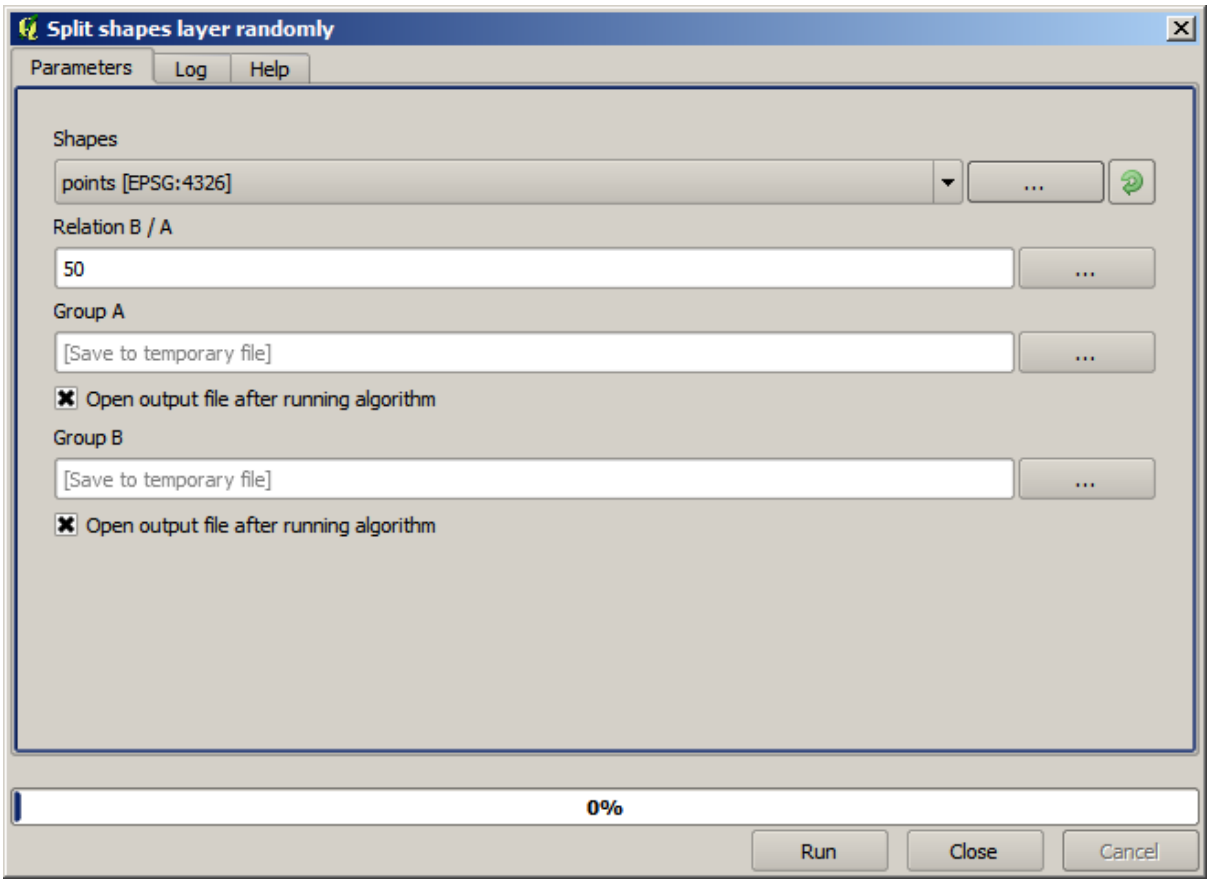


一度 SAGA をインストールすると、他のアルゴリズムと同様に、その名前をダブルクリックすることで SAGA アルゴリズムを実行できます。単純化されたインターフェイスを使用しているので、どのアルゴリズムが SAGA を基盤としているか、または別の外部アプリケーションにあるのかは分かりませんが、そのうちのひとつをたまたまダブルクリックして、対応するアプリケーションがインストールされていない場合、次のようなものが表示されます。

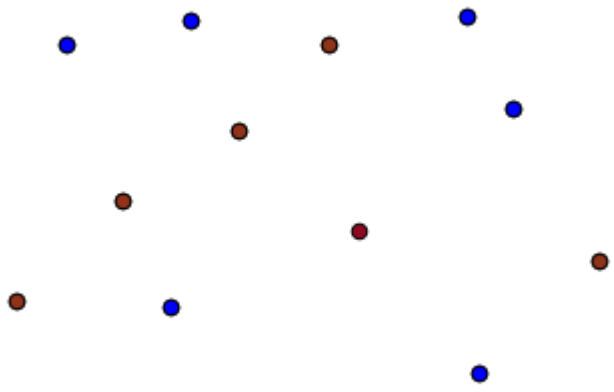


この場合、SAGA が正しくインストールされ構成されていれば、このウィンドウは表示されず、代わりにパラメータダイアログが表示されます。

SAGA ベースのアルゴリズムを **シェープレイヤーをランダムに分割** というもので試してみましょう。



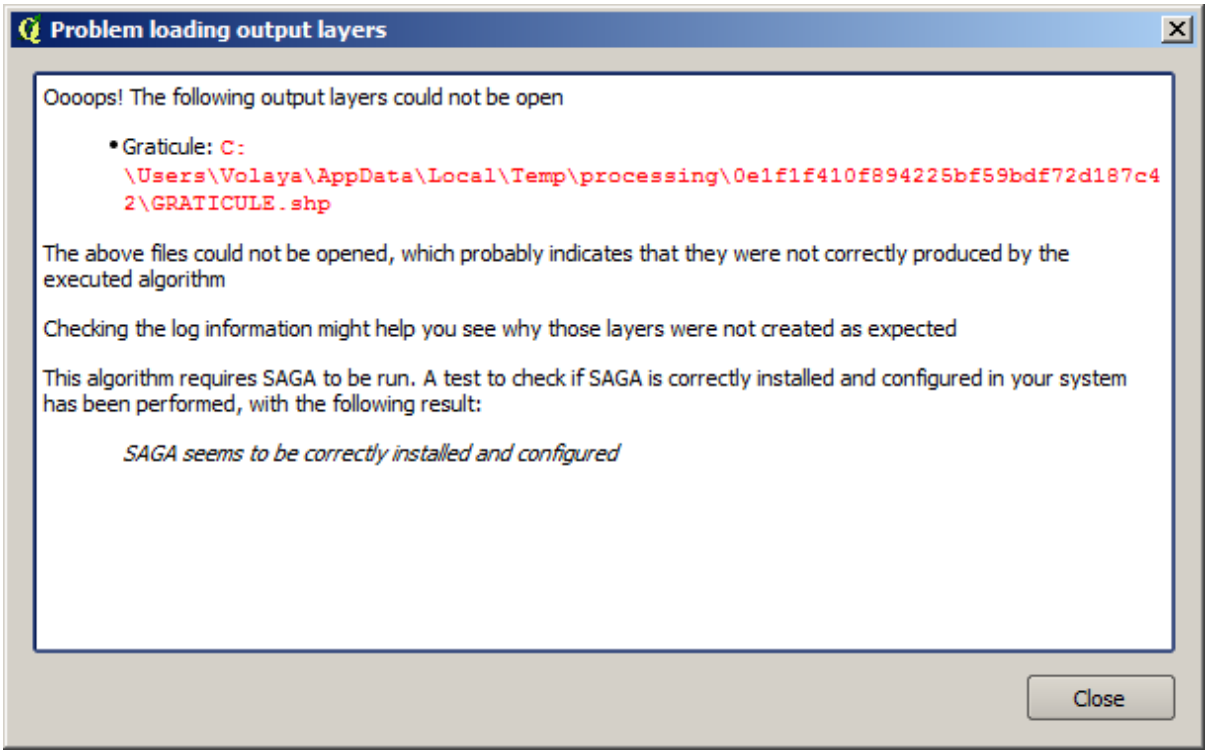
このレッスンに対応したプロジェクト内のポイントレイヤを入力として使用し、デフォルトのパラメータ値を使用すると、次のような結果が得られます（分割がランダムであるので、結果は異なる場合があります）。



入力レイヤは、それぞれに同じ数のポイントを持つ 2 つのレイヤに分割されました。この結果は、SAGA

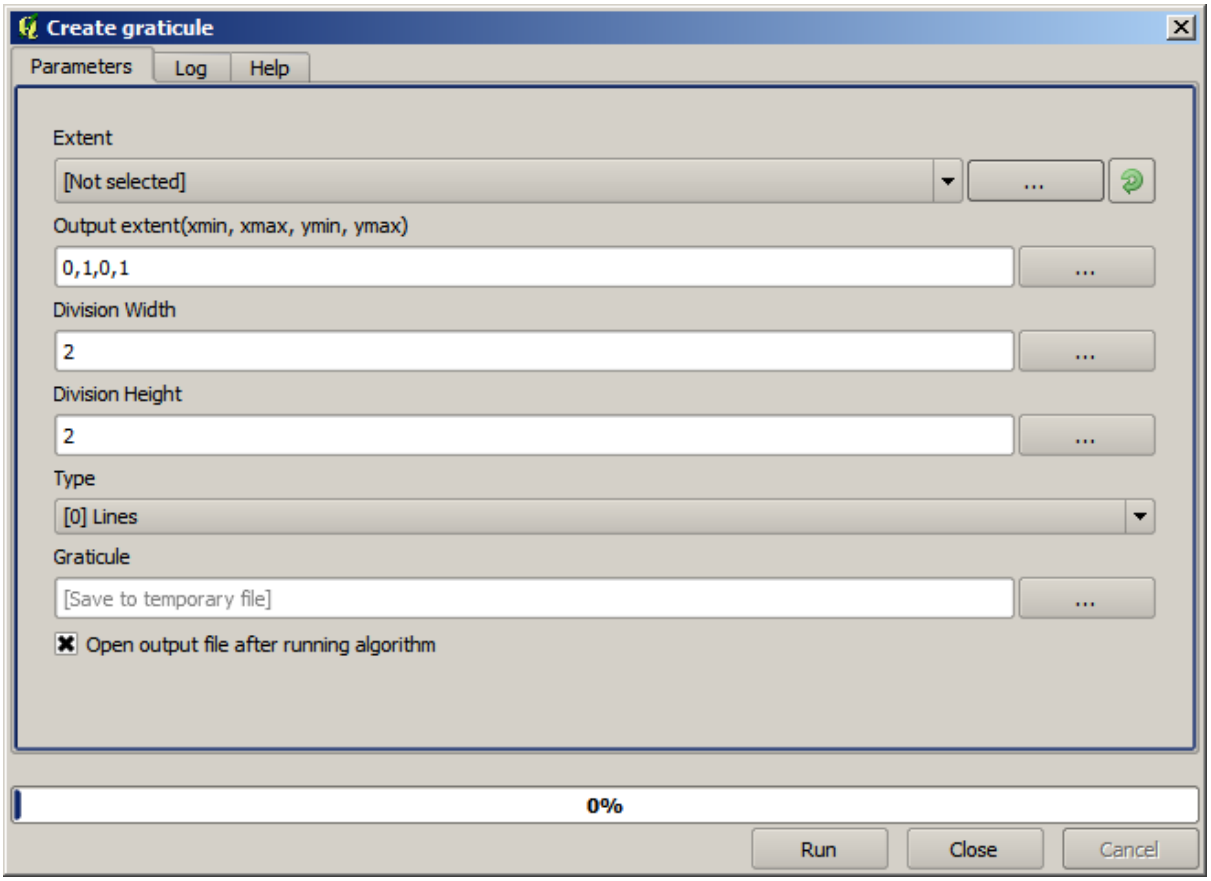
によって計算され、その後に QGIS に取り込まれ、QGIS プロジェクトに追加されました。

全てがうまくいけば、この SAGA ベースのアルゴリズムと、以前に実行をした他のアルゴリズムとの間の違いに気付かないでしょう。しかし SAGA が何らかの理由で、結果を出せず、QGIS が期待するファイルが生成できない場合があります。その場合、結果を QGIS プロジェクトに追加するときに問題が発生し、次のようなエラーメッセージが表示されます。



この種の問題は、SAGA（またはプロセシングフレームワークから呼び出している他のアプリケーション）が正しくインストールされていても起こるもので、それらに対処する方法を知っておくことが重要です。これらのエラーメッセージの１つを出してみましょう。

経緯線網を作成 アルゴリズムを開き、次の値を使用します。



指定された範囲よりも大きい幅と高さの値を使用しているので、SAGA は何も出力を生成できません。言い換えれば、パラメータ値が誤っていても、SAGA がそれらを取得して経緯線網を作成しようとするまではチェックされません。それを作成できないので、期待されたレイヤを生成せず、上記のようなエラーメッセージが表示されます。

注釈

SAGA >= 2.2.3 では、コマンドが自動的に誤った入力データを調整するため、エラーは発生しません。エラーを発生させるには、割り算に負の値を使用してください。

この種の問題を理解することは、それらを解決し、何が起こっているかの理由を見つけるのに役立ちます。エラーメッセージに見られるように、SAGA との接続が正しく機能していることを確認するためのテストが行われ、アルゴリズムが実行された方法に問題があるかもしれないことを示しています。これは SAGA だけでなく、他の外部アプリケーションにも適用されます。

次のレッスンでは、ジオアルゴリズムによって実行されるコマンドに関する情報が保持される処理ログを紹介し、このような問題が現れたときに詳細を確認する方法について学びます。

17.9 プロセシングログ

注釈

このレッスンではプロセシングログを説明します。

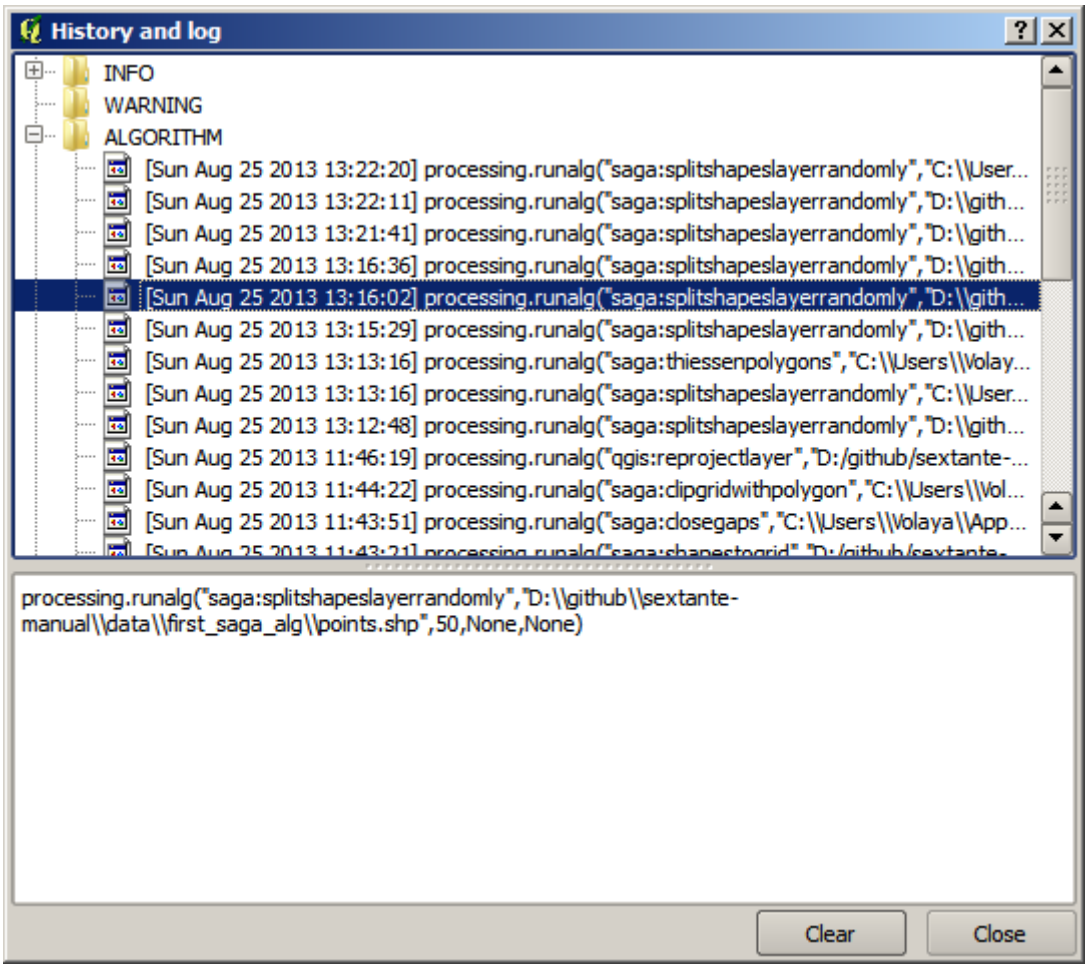
プロセシングフレームワークで実行されるすべての分析は、QGIS のログシステムに記録されます。これによってプロセシングツールによって行われたことの詳細を知るとか、問題が発生したときにそれを解決することができます。またログシステムもいくつかの対話を実装しているので、以前の操作の再実行もできます。

ログを開くには、QGIS のステータスバー右下のバルーンをクリックしてください。いくつかのアルゴリズムは、ここではそれらの実行に関する情報を残すことがあります。例えば、外部アプリケーションを呼び出すアルゴリズムは、通常、そのアプリケーションのコンソール出力をこのエントリにログ出力します。それを見ると、私たちが実行した（そして入力データが正しくなかったので実行に失敗した）ばかりの SAGA アルゴリズムの出力がここに格納されていることが分かります。

これは、何が起こっているか理解するのに便利です。上級ユーザーは、その出力を分析して、アルゴリズムが失敗した理由を探せるでしょう。上級ユーザーでなければ、持っている問題を診断するのを他の人に助けてもらうために有用でしょう。外部のソフトウェアのインストール中の問題であるかもしれないし、提供されたデータの問題であるかもしれません。

アルゴリズムには、そのアルゴリズムが実行できたとしても結果が正しくないかもしれない場合に警告を残すものがあります。例えば、非常に少ない点で補間アルゴリズムを実行するとき、アルゴリズムは実行できて結果は作成されますが、より多くの点が使用されなければならないので、それは正しくない可能性が高いです。指定されたアルゴリズムのいくつかの側面に確信がない場合は、定期的にこのタイプの警告をチェックすることをお勧めします。

GUI や（このマニュアルの後半で説明する）コンソールから実行されたかに関係なく、**プロセシング　履歴...** メニューから実行された **アルゴリズム** を見つけることができます。実行内容はコンソール呼び出しとしてこのダイアログに保存されます。つまり、アルゴリズムを実行するたびに、コンソールコマンドがログに追加され、作業セッションの完全な履歴が残ります。その履歴は以下のようなものです:



これはコンソールを使い始めるときに、アルゴリズムの構文を学ぶのに非常に便利です。コンソールから分析コマンドを実行する方法について説明するときにも使います。

17.9.1 ★☆☆ （初級レベル） 理解しよう

履歴は対話的でもあり、そのエントリをダブルクリックするだけで、以前のアルゴリズムを実行できます。これは、以前に行なった作業を繰り返す簡単な方法です。

例として、次のように試してみて下さい:

1. このマニュアルの第 1 章に対応するデータを開き、そこで説明されているアルゴリズムを実行します。
2. 次に履歴ダイアログに行き、今実行したアルゴリズムに対応した最後のアルゴリズムをリストから探します。
3. それをダブルクリックすると、通常のダイアログを使ってツールボックスから呼び出したときと同じように、新しい結果が生成されるはずです。

17.9.2 ★★☆ （中級レベル） 理解しよう

そのアルゴリズムに手を加えることもできます。

1. アルゴリズムの呼出をコピーします
2. **プラグイン** *Python* **コンソール** を開きます
3. コピーを貼り付けて分析を実行します； パラメータは自由に変更します。
4. 結果のファイルを表示するには、貼り付けたコマンドの `processing.run` を `processing.runAndLoadResults` に置き換えるとよいでしょう。

17.10 ラスタ計算機。No-data 値

注釈

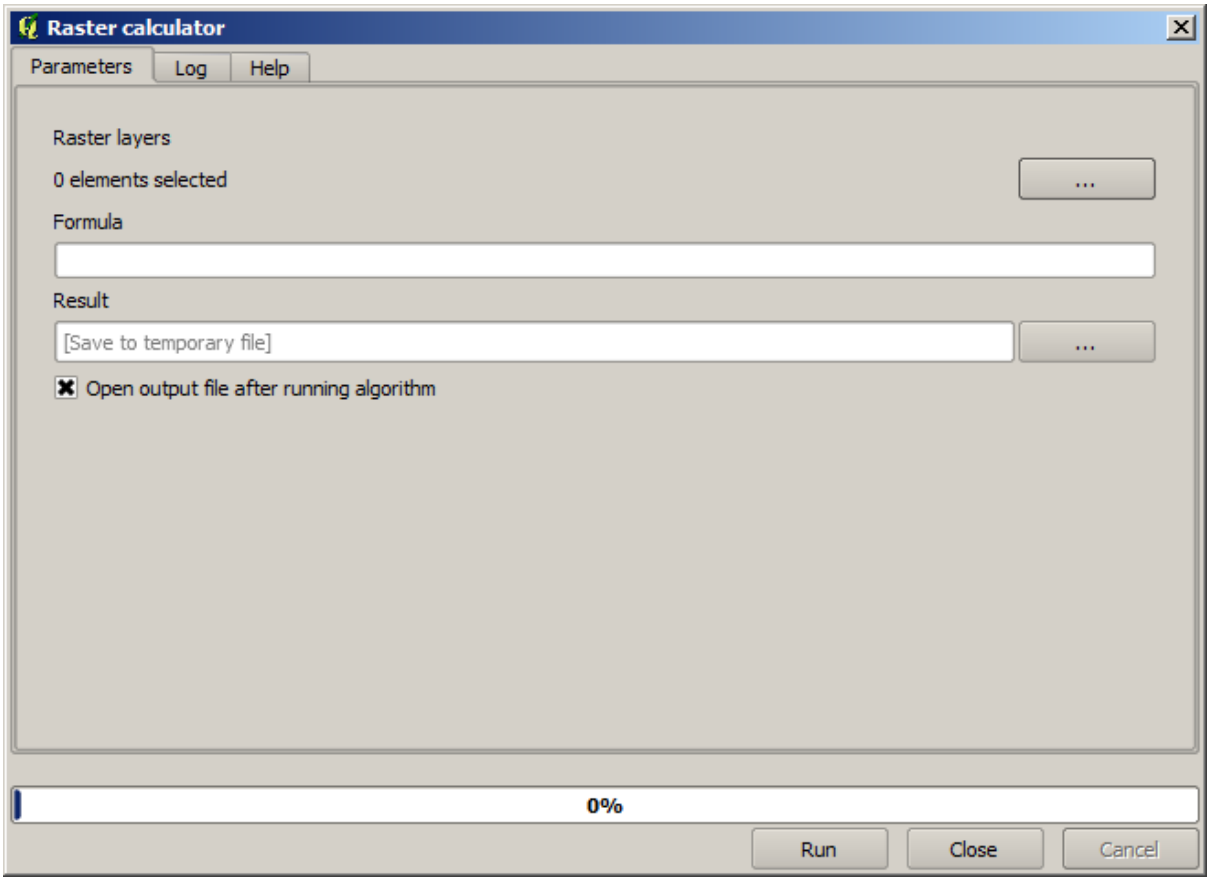
このレッスンでは、ラスタ計算機を使ってラスタレイヤに対していくつかの操作を行う方法を紹介します。また、no--data 値とは何か、計算機や他のアルゴリズムがどのようにそれを扱うかについても説明します。

ラスタ計算機は最も強力なアルゴリズムの一つです。様々な計算に使える非常に柔軟で汎用性の高いアルゴリズムであり、すぐにあなたのツールボックスの重要な一部になることでしょう。

このレッスンでは、ラスタ計算機を使っていくつか計算を行います。そのほとんどは簡単な計算です。ここでは、ラスタ計算機の使い方や、特殊な状況に対応する方法を学びます。これを理解することは、後で計算機を使うときに期待通りの結果を得るために重要であり、また計算機を使って一般的に適用される特定のテクニックを理解するために重要です。

このレッスンに対応する QGIS プロジェクトを開きます。それには複数のラスタレイヤが含まれていることが分かります。

次にツールボックスを開き、ラスタ計算機に対応するダイアログを開きます。



注釈

インターフェイスは、最近のバージョンで異なります。

2 つのパラメータがダイアログに含まれています。

- 分析に使用するレイヤ。これは複数入力であり、望むだけ多くのレイヤを選択することができます。右側のボタンをクリックし、表示されるダイアログで使用したいレイヤを選択します。
- 適用する式。式は、上記パラメータで選択され、変数名としてアルファベット文字 (`a`, `b`, `c...`) または `g1`, `g2`, `g3...` で命名されたレイヤを使用します。すなわち、式 `a + 2 * b` は `g1 + 2 * g2` と同じであり、第一レイヤの値と第二レイヤの値の 2 倍の合計を計算します。レイヤの順序付けは選択ダイアログに表示されるのと同じ順序です。

警告

計算機では、大文字と小文字が区別されます。

手始めに、メートルからフィートに DEM の単位を変更します。必要な計算式は次のものです。

```
h' = h * 3.28084
```

レイヤフィールド内で DEM を選択し、式フィールドに `a * 3.28084` とタイプします。

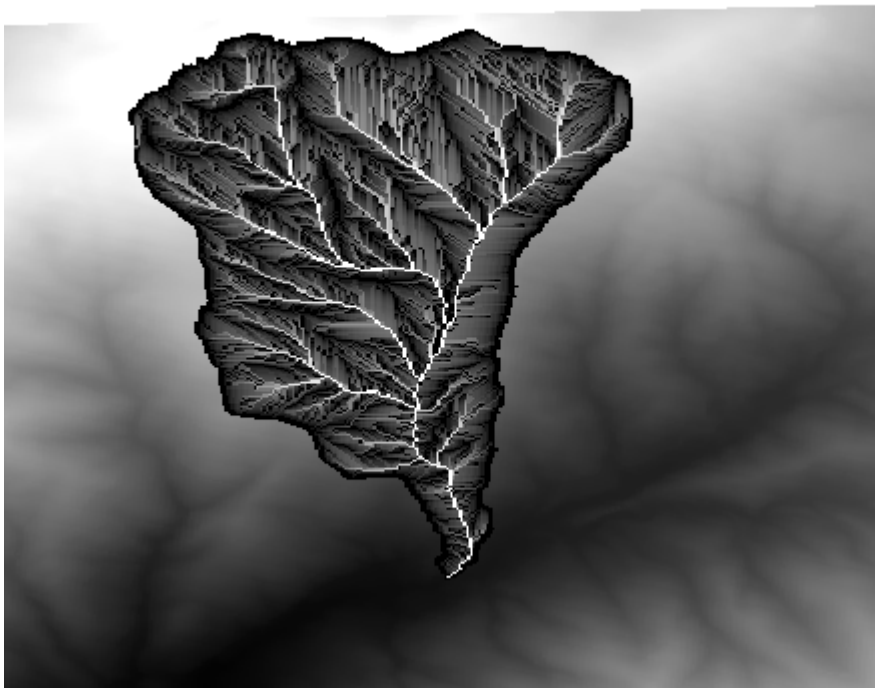
> **警告**
>
> 英語以外のユーザーの場合：常に「,」ではなく「.」を使用してください。

実行 をクリックしてアルゴリズムを実行してください。入力レイヤの同じ外観ですが、異なった値を持つレイヤが得られます。使用した入力レイヤは、その全てのセルに有効な値を持っているので、最後のパラメータはまったく効果がありません。

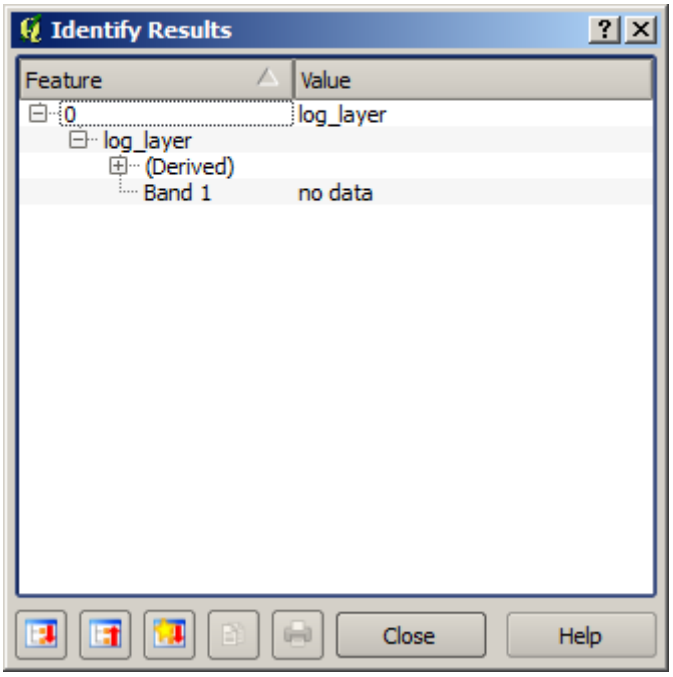
今度は、*accflow* レイヤを使って、別の計算をしてみましょう。このレイヤは、水文学的なパラメータである累積流量の値を含んでいます。流域の範囲内にのみその値があり、その外は no--data 値です。見ての通り、値が分散しているため、レンダリングはあまり有益ではありません。流量の蓄積の対数を使用すると、より情報量の多い表現になります。ラスタ計算機を使ってそれを計算することができます。

再びアルゴリズムダイアログを開き、入力レイヤとして *accflow* レイヤだけを選択し、次の式を入力します： `log(a)` 。

これが得られるレイヤです。



与えられた点でのレイヤの値を知るために **識別** ツールを選択し、たった今作成したレイヤを選択し、流域の外でポイントをクリックすると、no--data 値が含まれていることがわかるでしょう。



次の練習では、1 つのレイヤの代わりに 2 つのレイヤを使用し、2 番目のレイヤで定義された盆地内のみ有効な標高値を持つ DEM を取得するつもりです。計算機ダイアログを開き、入力レイヤフィールドでプロジェクトの両方のレイヤを選択します。対応するフィールドに次の数式を入力します:

```
a/a * b
```

a は、積算流量のレイヤを指し（リストに表示される最初のものであるため)、 b は DEM を指します。外部データ値 - ここで式の最初の部分でしていることはそれ自体で積算流量レイヤを割ることです。この結果は流域内で値 1、外でデータなし値です。それから DEM を掛け、流域内でそれらのセルの標高値 （ `DEM * 1 = DEM`)、外でデータなし値（ `DEM * no_data = no_data` ）を得ます

結果のレイヤはこれです。



この技術は、ラスタレイヤ内で値を **マスク** するために頻繁に使用され、ラスタレイヤによって使用される任意の矩形領域でない領域のための計算を実行したいときはいつでも有用です。例えば、ラスタレイヤの標高ヒストグラムはあまり意味を持ちません。代わりそれが流域に対応する値のみを使用して（上記の場合のように）計算される場合、得られる結果は実際に流域の構成についての情報を与える有意味なものです。

実行してきたこのアルゴリズムについては、「データなし」値とそれらがどのように処理されるかを別にしても、他に興味深いものがあります。私たちが掛け算しているレイヤの範囲を見る場合（それは目次のレイヤの自分の名前をダブルクリックしてそのプロパティを見ることで行うことができます)、同じでないことがわかります、流量蓄積レイヤによって覆われる範囲は完全 DEM の範囲より小さいので。

それが意味するのは、これらのレイヤが一致しないこと、1 つまたは両方のレイヤをリサンプリングすることでそれらのサイズと範囲を同じに揃えなければ直接掛け算はできないことです。しかし、私たちは何もしませんでした。QGIS ではこのような状況の面倒を見てくれて、必要なときに自動的に入力レイヤをリサンプリングします。出力範囲は、入力レイヤから計算された最小の被覆範囲、およびそれらセルサイズの最小セルサイズです。

この場合（そしてほとんどの場合)、これは所望の結果を生成しますが、どんな操作が追加で行われているかには常に注意しておく必要があります。なぜならそれらは結果に影響を与える可能性があるからです。この動作が希望されない場合がある場合には、手動のリサンプリングが事前に適用されるべきです。後の章では、複数のラスタレイヤを使用したときのアルゴリズムの動作について詳細に見るでしょう。

このレッスンを別のマスク作成の練習で終えましょう。私たちは標高が 1000 メートルと 1500 メートルの間のすべての地域で傾きを計算しようとしています。

この場合は、マスクとして使用するレイヤはありませんが、計算機を使用して作成できます。

唯一の入力レイヤとして DEM を、そして次式を使用して計算を実行します

```
ifelse(abs(a-1250) < 250, 1, 0/0)
```

おわかりのように、簡単な代数演算を行うためだけでなく、計算機を使用でき、また上記のような条件文を含むより複雑な計算を実行します。

結果は、私たちが作業をしたい範囲内で値 1、およびそれ以外のセルでデータなしを持っています。



データなし値は 0/0 の式から来ています。それは未定の値であるので、SAGA は、実際にはデータ値として扱う NaN（非数）の値を、追加します。この小さなトリックを使えば、セルのデータなし値が何か知らなくても、データなし値を設定できます。

今、プロジェクトに含まれる傾斜レイヤを掛ける必要があるだけで、希望の結果が得られるでしょう。

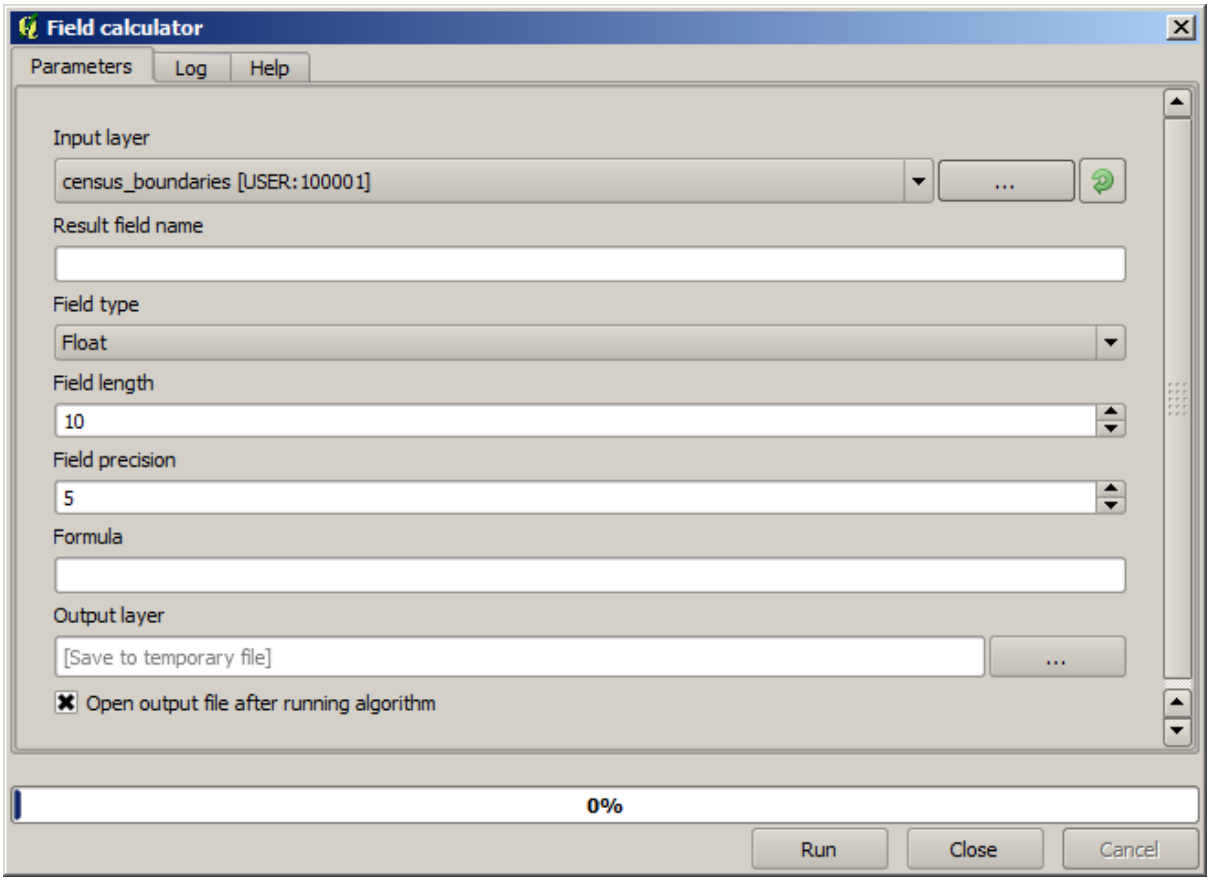
すべてのことは、計算機で、単一の操作で行うことができます。これは読者の練習として残しておきます。

17.11 ベクター計算機

注釈

このレッスンでは、ベクター演算を使用して、数式に基づいてベクターレイヤーに新しい属性を追加する方法について説明します。

We already know how to use the raster calculator to create new raster layers using mathematical expressions. A similar algorithm is available for vector layers, and generates a new layer with the same attributes of the input layer, plus an additional one with the result of the expression entered. The algorithm is called *Field calculator* and has the following parameters dialog.



注釈

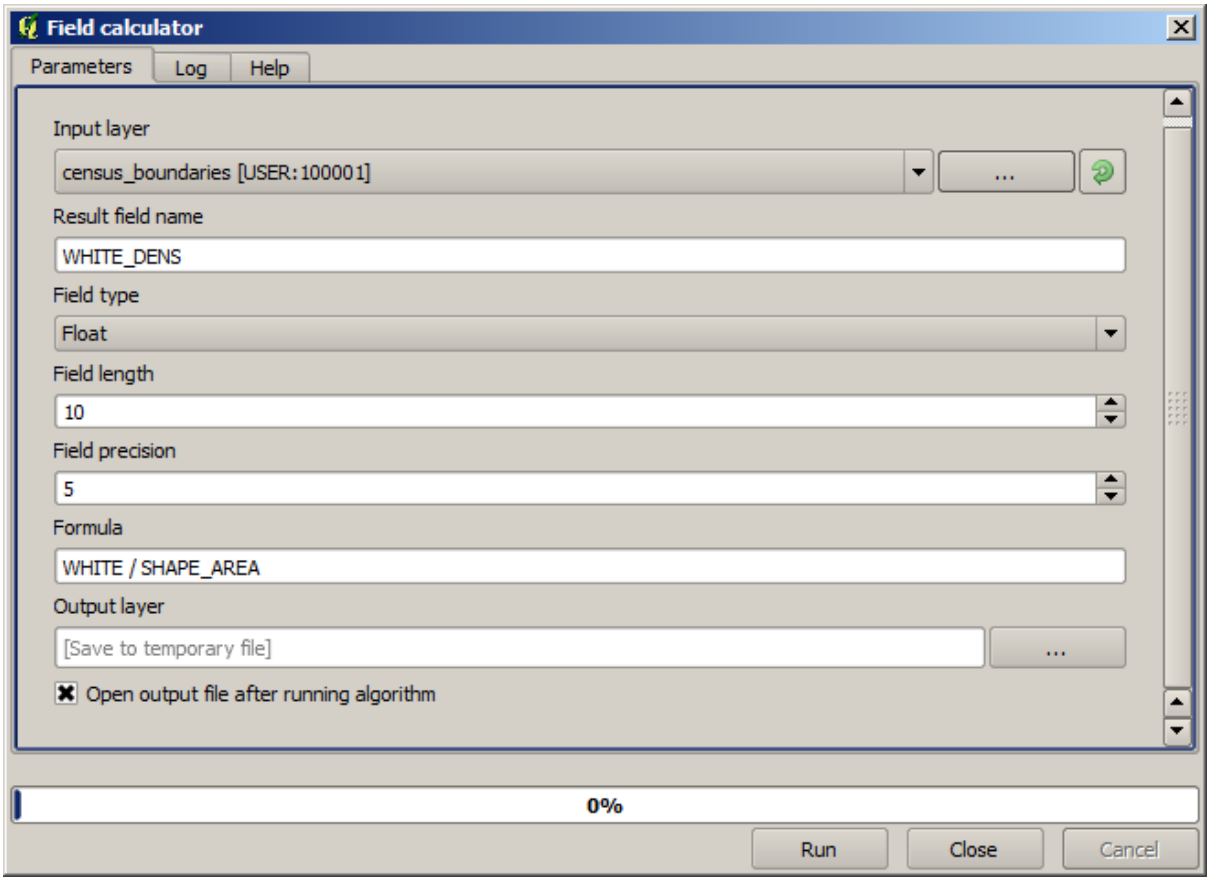
インターフェイスが大幅に変更されたプロセッシングの新しいバージョンでは、より強力で使いやすいです。

ここでは、そのアルゴリズムを使用してのいくつかの例があります。

まずは、国勢調査を表し、各ポリゴン中の白人の人口密度を計算してみましょう。属性テーブル内の二つのフィールド、すなわち WHITE と SHAPE_AREA がそのために使用できます。それらを割り算し、百万で乗算する（平方キロメートルあたりの密度にするため）必要があるだけなので、対応するフィールドに次の式を使用できます

```
( "WHITE" / "SHAPE_AREA" ) * 1000000
```

以下に示すようにパラメーターダイアログが満たされる必要があります。



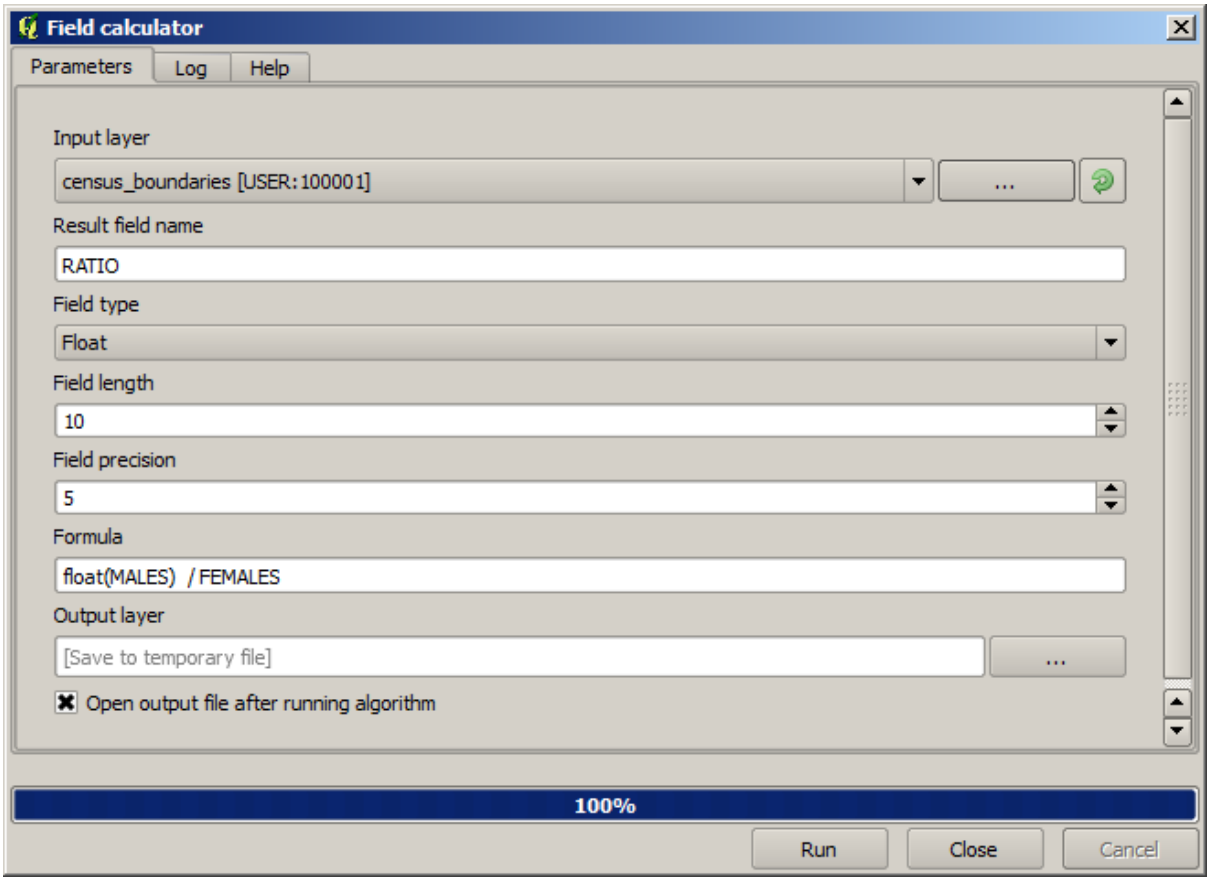
これは WHITE_DENS という名前の新しいフィールドを生成します。

今度は、男性人口が女性人口に比して数値的に優位にあるかどうかを示す新しいものを作成するために、MALES と FEMALES フィールド間の比率を計算してみましょう。

次の式を入力します。

```
"MALES" / "FEMALES"
```

This time the parameters window should look like this before pressing the *OK* button.

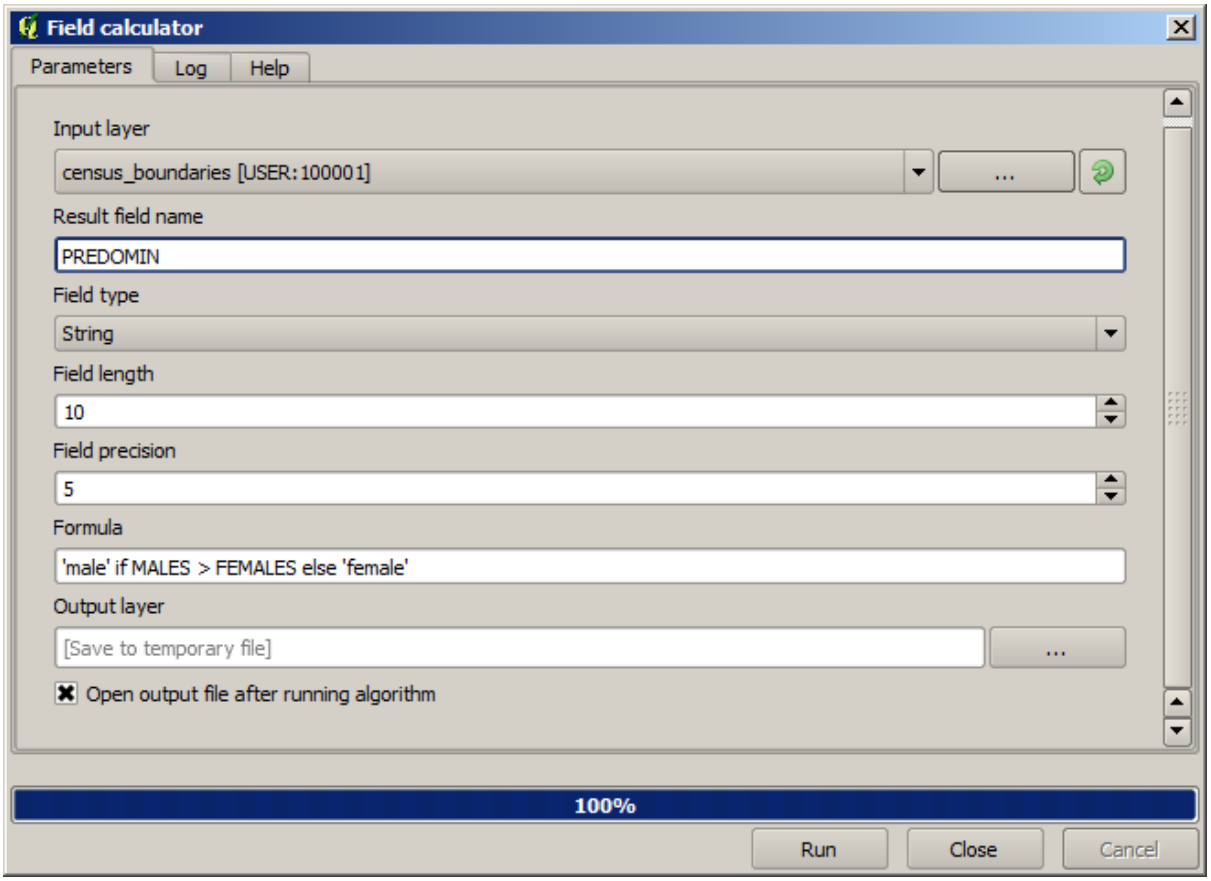


両方のフィールドが整数型であるため、以前のバージョンでは、結果が整数に切り捨てられることになります。この場合、式は次のようになります。私たちは浮動小数点数に結果を望んでいることを示すために `1.0 * "MALES" / "FEMALES"`。

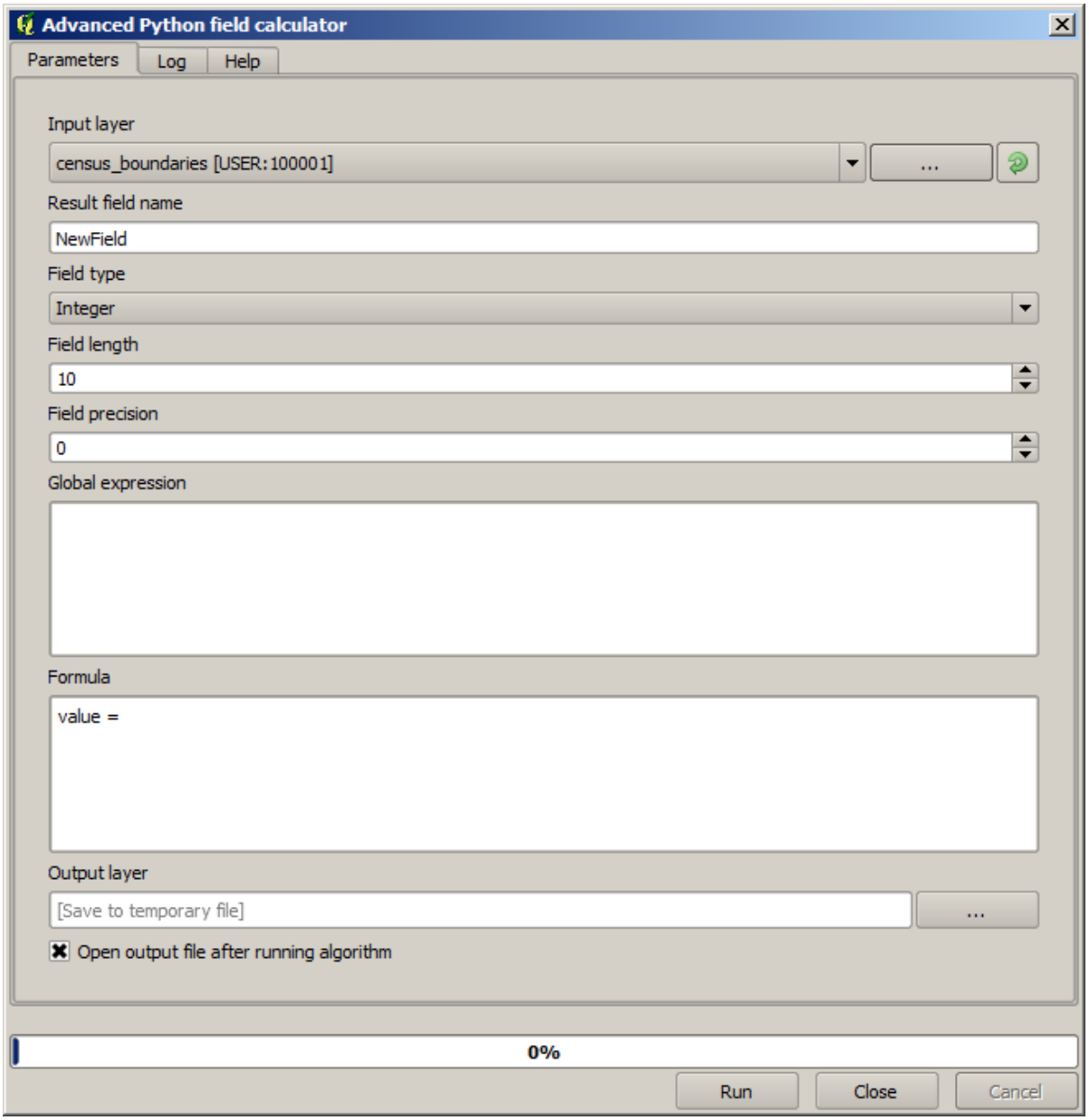
次の式を使用して、`male` または `female` テキスト文字列の代わりに、それらの比の値で新しいフィールドを持つように、条件付きの関数を使用できます:

```
CASE WHEN  "MALES" > "FEMALES"  THEN 'male' ELSE 'female' END
```

パラメーターウィンドウは、次のようになります。



A python field calculator is available in the *Advanced Python field calculator*, which will not be detailed here



17.12 範囲を定義する

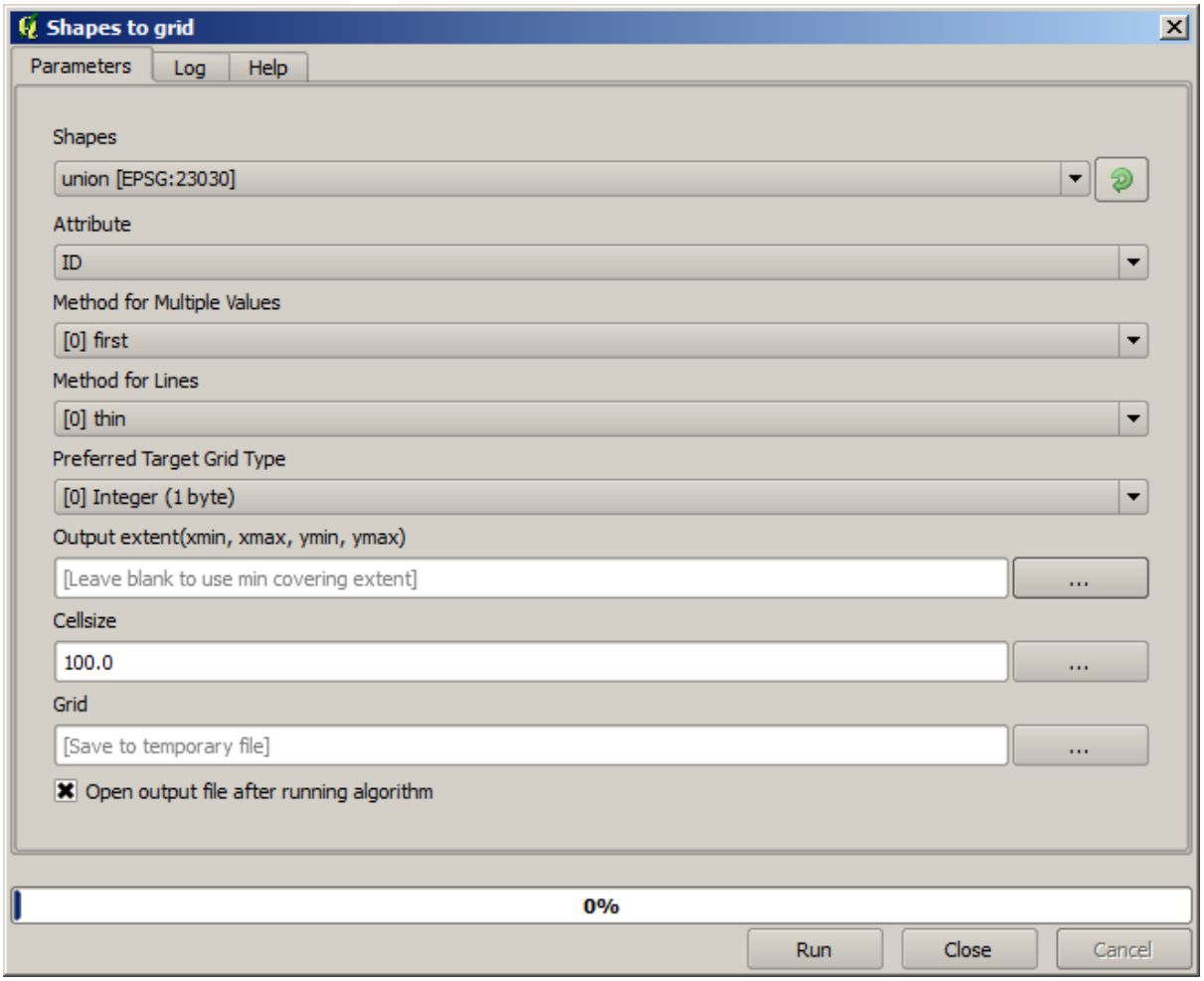
注釈

このレッスンでは、いくつかのアルゴリズムによって必要とされている範囲、特にラスタの範囲を定義する方法について説明します。

いくつかのアルゴリズムは、それらが行う解析によってカバーされる区域を定義する範囲、そしてたいていは結果のレイヤの区域を定義する範囲を必要とします。

範囲が必要とされる場合は、それを定義する 4 つの値（最小 X、最小 Y、最大 X、最大 Y）を入力することによって手動で定義できますが、他にもそれを同様に行うより実用的でより興味深い方法があります。このレッスンではそれらのすべてを見ることができます。

First, let's open an algorithm that requires an extent to be defined. Open the *Rasterize (vector to raster)* algorithm, which creates a raster layer from a vector layer.

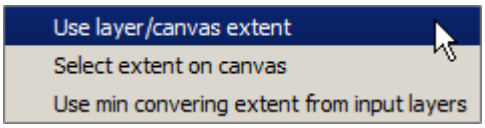


最後の 2 つのパラメータを除くすべてのパラメータは、ラスタ化するレイヤを定義し、ラスタ化処理の方法を設定するために使用されます。一方、最後の 2 つのパラメータは、出力レイヤの特性を定義するものです。つまり、カバーする区域（入力ベクタレイヤがカバーする区域と同じとは限らない）と、解像度/セルサイズ（ベクタレイヤにはセルサイズがないため、ベクタレイヤから推測することはできない）を定義する。

まず行うことができることは、前に説明した定義する 4 つの値を、カンマで区切って入力することです。

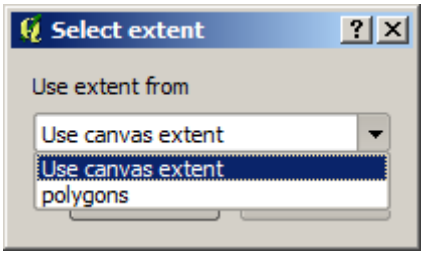


それは余分な説明を必要としません。これは最も柔軟なオプションですが、場合によってはあまり実用的でないこともあり、それが他のオプションが実装されている理由です。それらにアクセスするには、範囲テキストボックスの右側のボタンをクリックする必要があります。



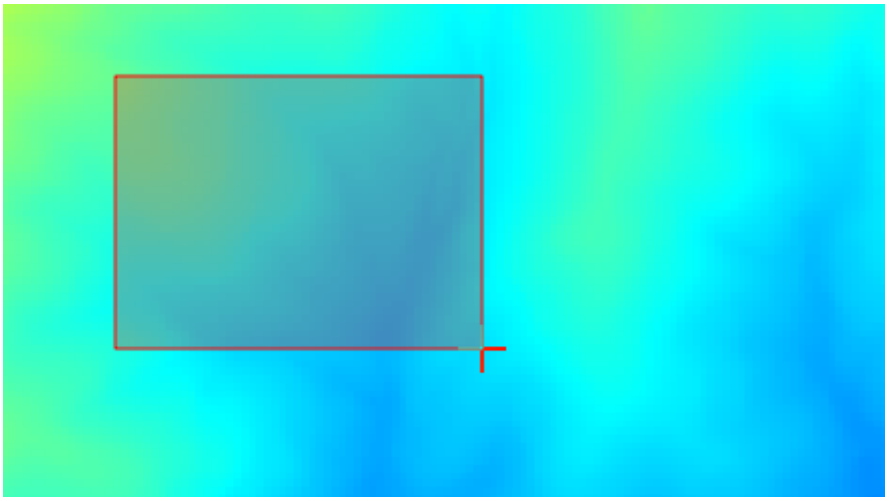
それらのそれぞれが何をするか見てみましょう。

The first option is *Use layer/canvas extent*, which will show the selection dialog shown below.



ここでは、キャンバスの範囲（現在のズームでカバーされている範囲）、または利用可能ないずれかのレイヤの範囲を選択できます。それを選択して *OK* をクリックすると、テキストボックスが自動的に対応する値で満たされます。

The second option is *Select extent on canvas*. In this case, the algorithm dialog disappears and you can click and drag on the QGIS canvas to define the desired extent.

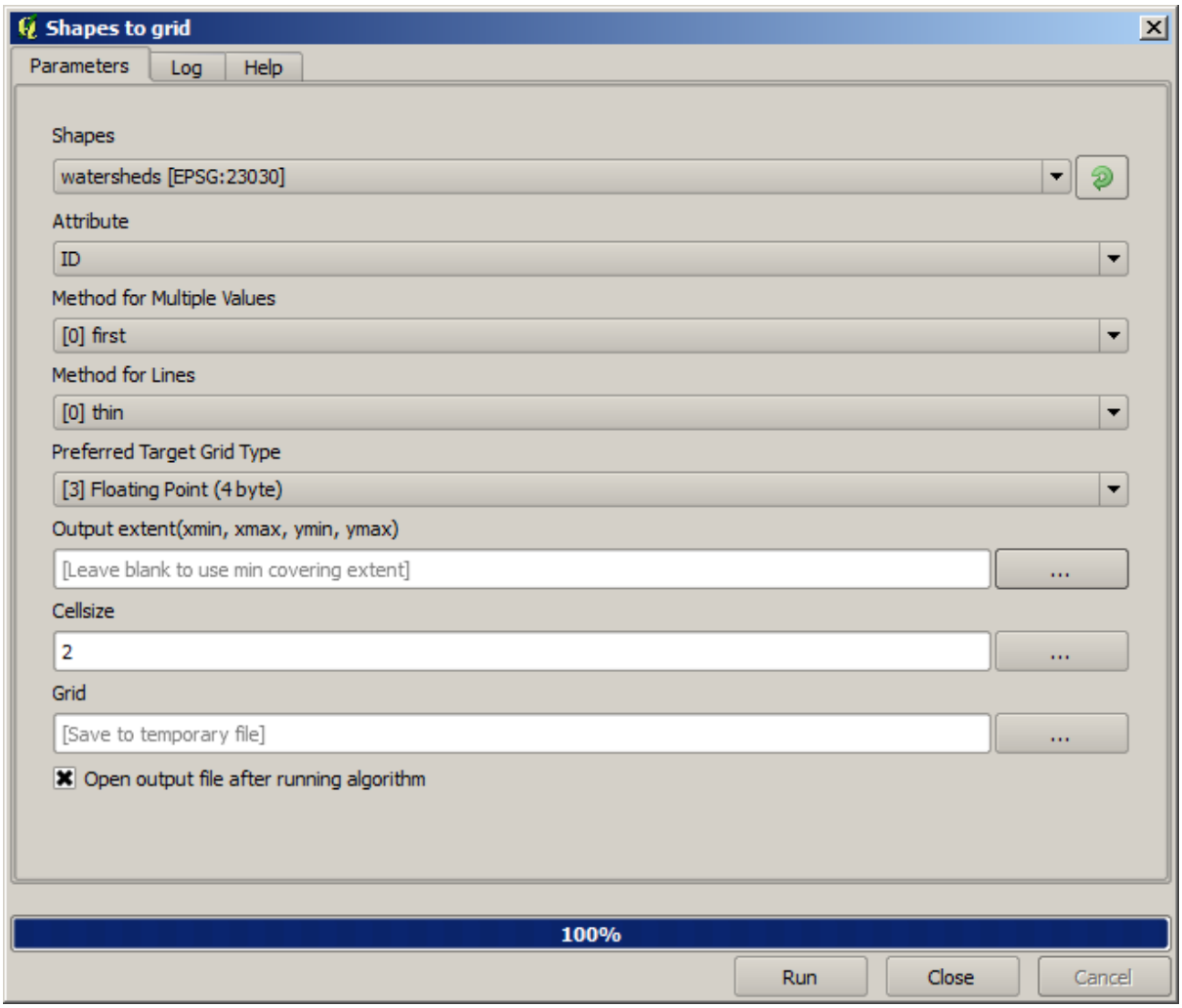


マウスボタンを放すとダイアログが再び現れ、そのテキストボックスには定義された範囲に対応する値がすでに入っています。

The last option is *Use min covering extent from input layers*, which is the default option. This will compute the min covering extent of all layers used to run the algorithm, and there is no need to enter any value in the text box. In the case of a single input layer, as in the algorithm we are running, the same extent can be obtained by selecting that same input layer in the *Use layer/canvas extent* that we already saw. However, when there are several input layers, the min covering extent does not correspond to any of the input layer extent, since it is computed from all of them together.

ラスタ化アルゴリズムを実行するために、この最後のメソッドを使用します。

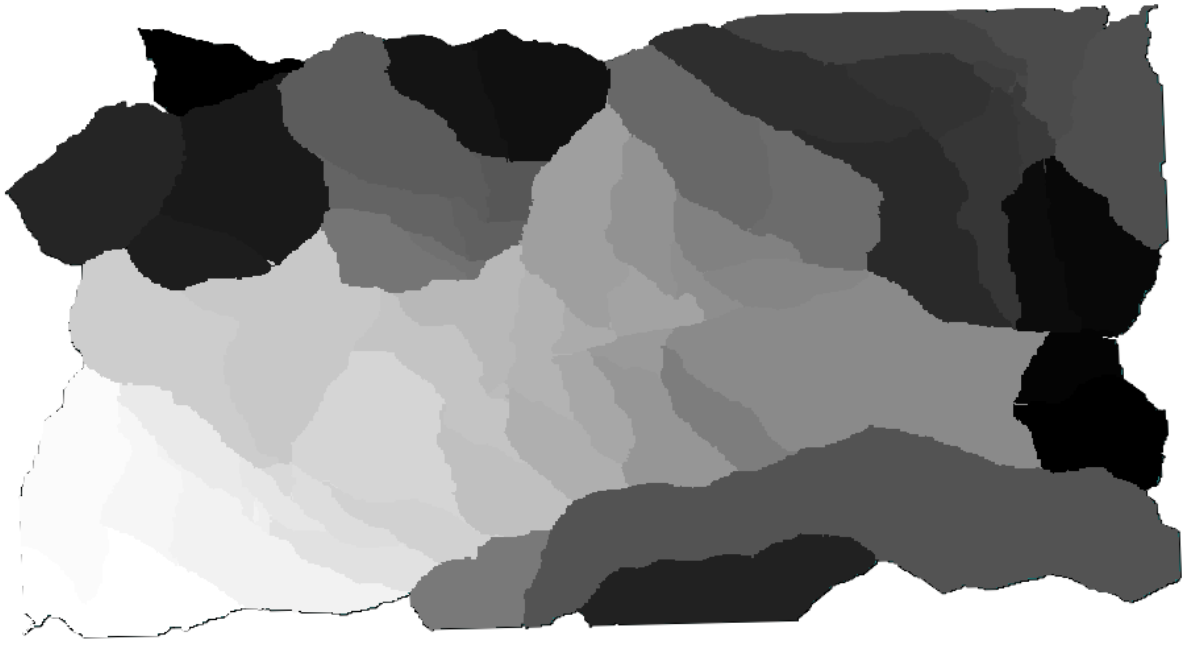
Fill the parameters dialog as shown next, and press *OK*.



注釈

In this case, better use an `Integer (1 byte)` instead of a `Floating point (4 byte)`, since the `NAME` is an integer with maximum value=64. This will result in a smaller file size and faster computations.

元のベクタレイヤがカバーする区域を正確にカバーするラスタ化レイヤが得られるでしょう。



In some cases, the last option, *Use min covering extent from input layers*, might not be available. This will happen in those algorithm that do not have input layers, but just parameters of other types. In that case, you will have to enter the value manually or use any of the other options.

選択が存在する場合、レイヤの範囲は地物の全体集合のことであり、たとえラスタ化が選択された項目に対してのみ実行されても、選択は範囲を計算するために使用されませんのでご注意ください。その場合は、実際に選択から新しいレイヤを作成しそれを入力として使用したいかもしれません。

17.13 HTML 出力

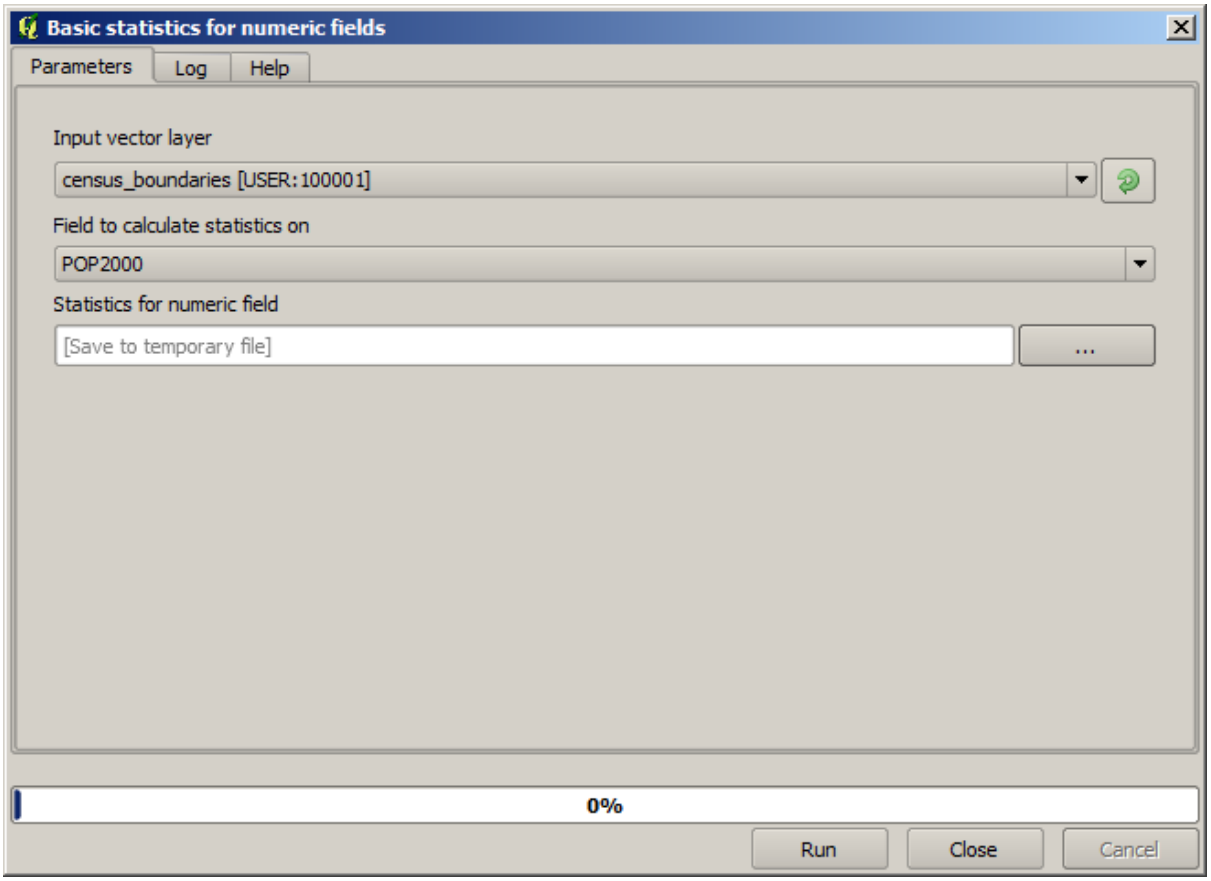
注釈

このレッスンでは、QGIS は、テキスト出力とグラフを生成するために使用されている HTML 形式の出力を、どのように処理するかを学びます。

All the outputs we have produced so far were layers (whether raster or vector). However, some algorithms generate outputs in the form of text and graphics. All this outputs are wrapped in HTML files and displayed in the so--called *Results viewer*, which is another element of the processing framework.

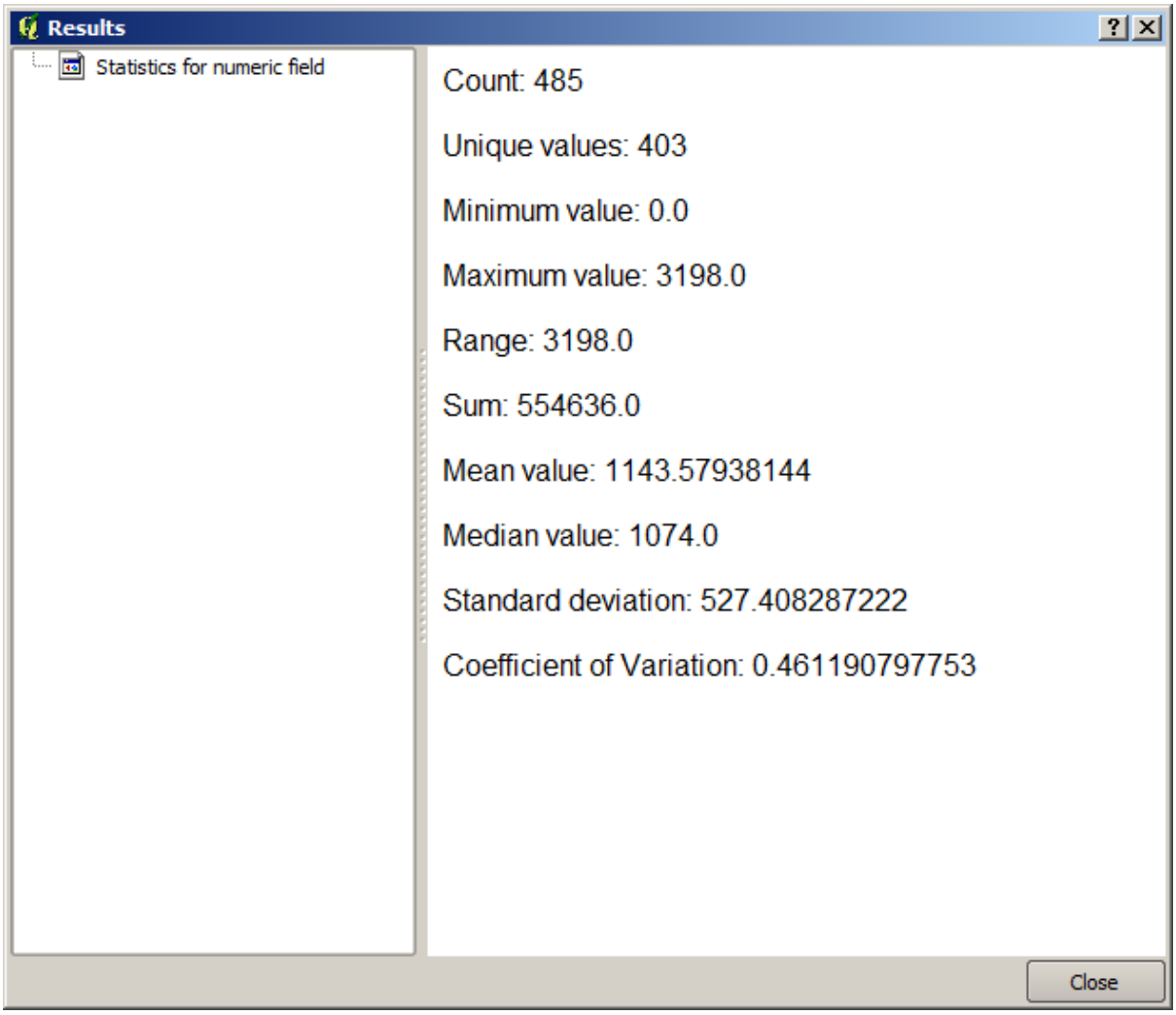
どのように機能するかを理解するため、これらのアルゴリズムの１つを見てみましょう。

Open the project with the data to be used in this lesson and then open the *Basic statistics for fields* algorithm.



アルゴリズムはかなり単純で、使用するレイヤーおよびそのフィールド（数値フィールド）の１つを選択する必要があるだけです。出力のタイプは HTML ですが、対応するボックスは正確にラスターまたはベクター出力の場合に見つけることができるもののように動作します。ファイルパスを入力できますし、一時ファイルに保存する場合は空白のままにもできます。この場合はしかし、 `html` と `htm` 拡張子だけが許可されているので、別の拡張子を使用して出力形式を変更する方法はありません。

Run the algorithm selecting the only layer in the project as input, and the `POP2000` field, and a new dialog like the one shown next will appear once the algorithm is executed and the parameters dialog is closed.



This is the *Results viewer*. It keeps all the HTML result generated during the current session, easily accessible, so you can check them quickly whenever you need it. As it happens with layers, if you have saved the output to a temporary file, it will be deleted once you close QGIS. If you have saved to a non-temporary path, the file will remain, but it will not appear in the *Results viewer* the next time you open QGIS.

いくつかのアルゴリズムは、他のより詳細な出力に分けることができないテキストを生成します。例えば、アルゴリズムで外部プロセスからのテキスト出力を取り込む場合がそうです。他の場合には、出力はテキストとして提示されますが、内部的には通常は数値の形で、いくつかの小さな出力に分割されます。私たちが実行したアルゴリズムはそのうちの一つです。これらの値の各々は、単一の出力として扱われ、変数に格納されます。これは今はまったく重要性を持ちませんが、モデルデザイナーに移ったときに、これらの値が他のアルゴリズムの数値入力として使用できるようになっていることがわかるでしょう。

17.14 最初の分析例

注釈

このレッスンでは、プロセシング・フレームワークの要素により精通できるよう、ツールボックスだけを使用していくつかの実際の分析を実行します。

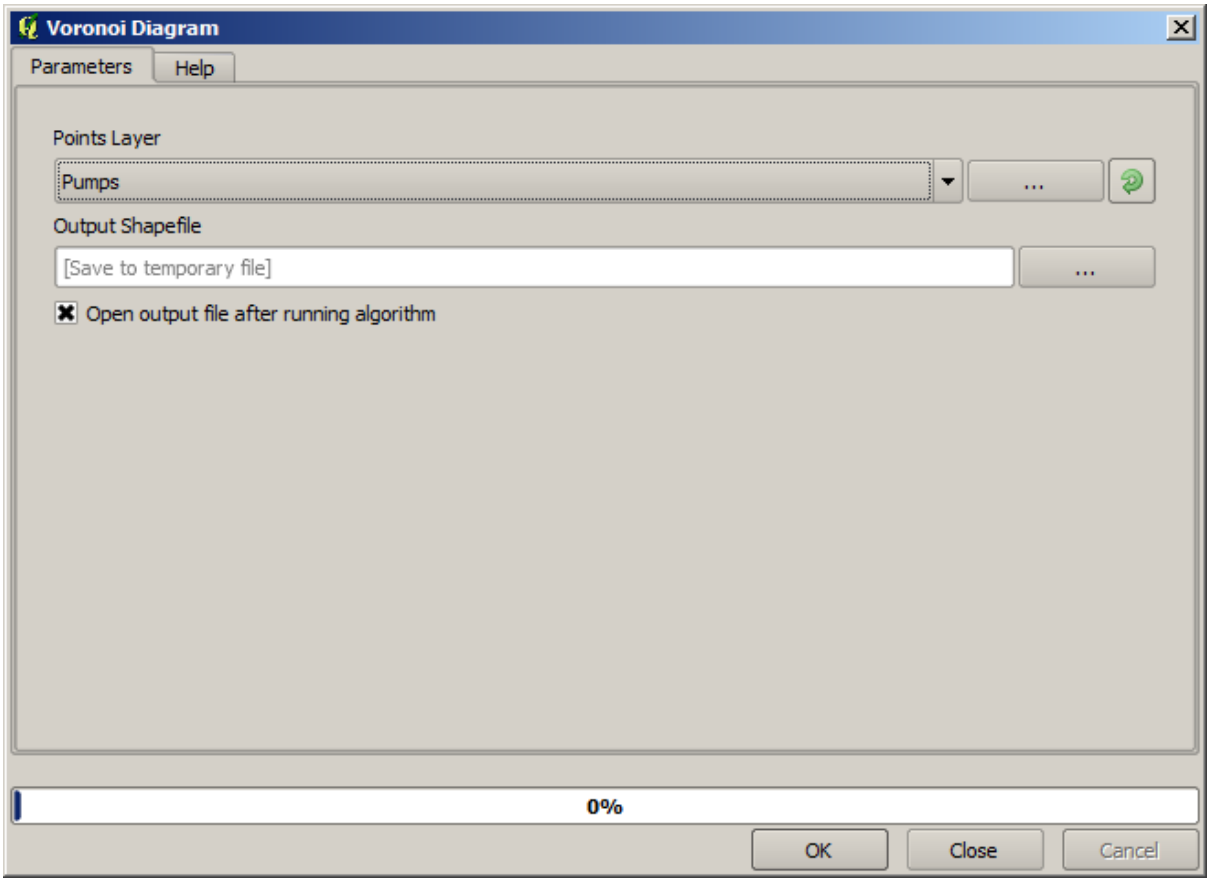
今やすべて設定されていて外部アルゴリズムを使用できますので、空間分析を実行するための非常に強力なツールを持っています。何か実世界のデータでより大規模な練習に取り掛かる時です。

我々は、John Snow が 1854 年に彼の画期的な仕事 (https://en.wikipedia.org/wiki/John_Snow_%28physician%29) で使用した有名なデータセットを使用し、いくつかの興味深い結果を得ることになります。このデータセットの分析は非常に明白であり、良い結果と結論に至るために高度な GIS 技術を必要としませんが、このような空間的問題が、異なる処理ツールを使うことによってどのように分析され解決されるかを示す良い方法であるといえるでしょう。

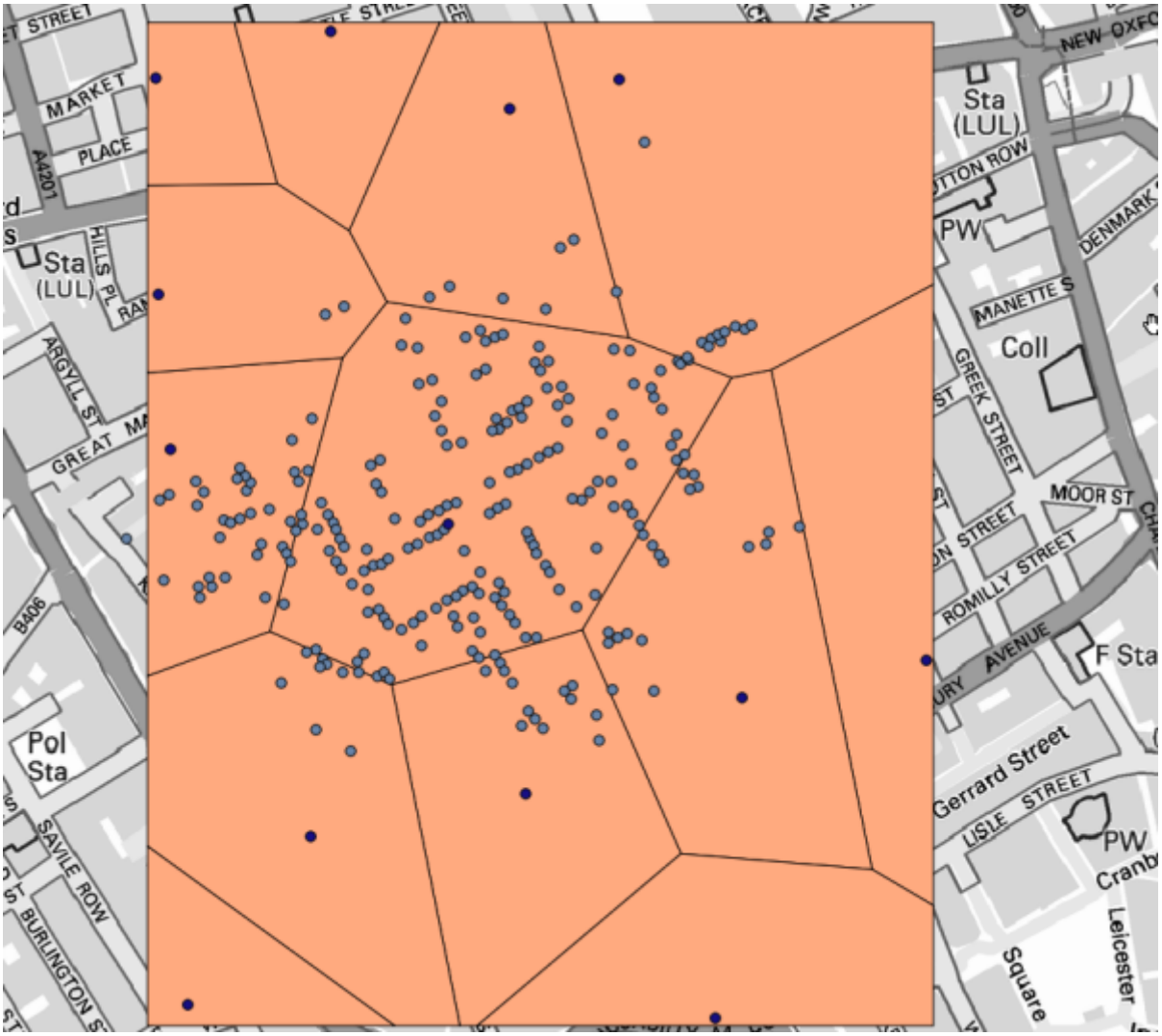
データセットには、コレラによる死者の位置とポンプの位置のシェープファイル、および TIFF フォーマットでの OSM レンダリングされた地図が含まれています。このレッスンのための対応 QGIS プロジェクトを開きます。



The first thing to do is to calculate the Voronoi diagram (a.k.a. Thiessen polygons) of the pumps layer, to get the influence zone of each pump. The *Voronoi polygons* algorithm can be used for that.

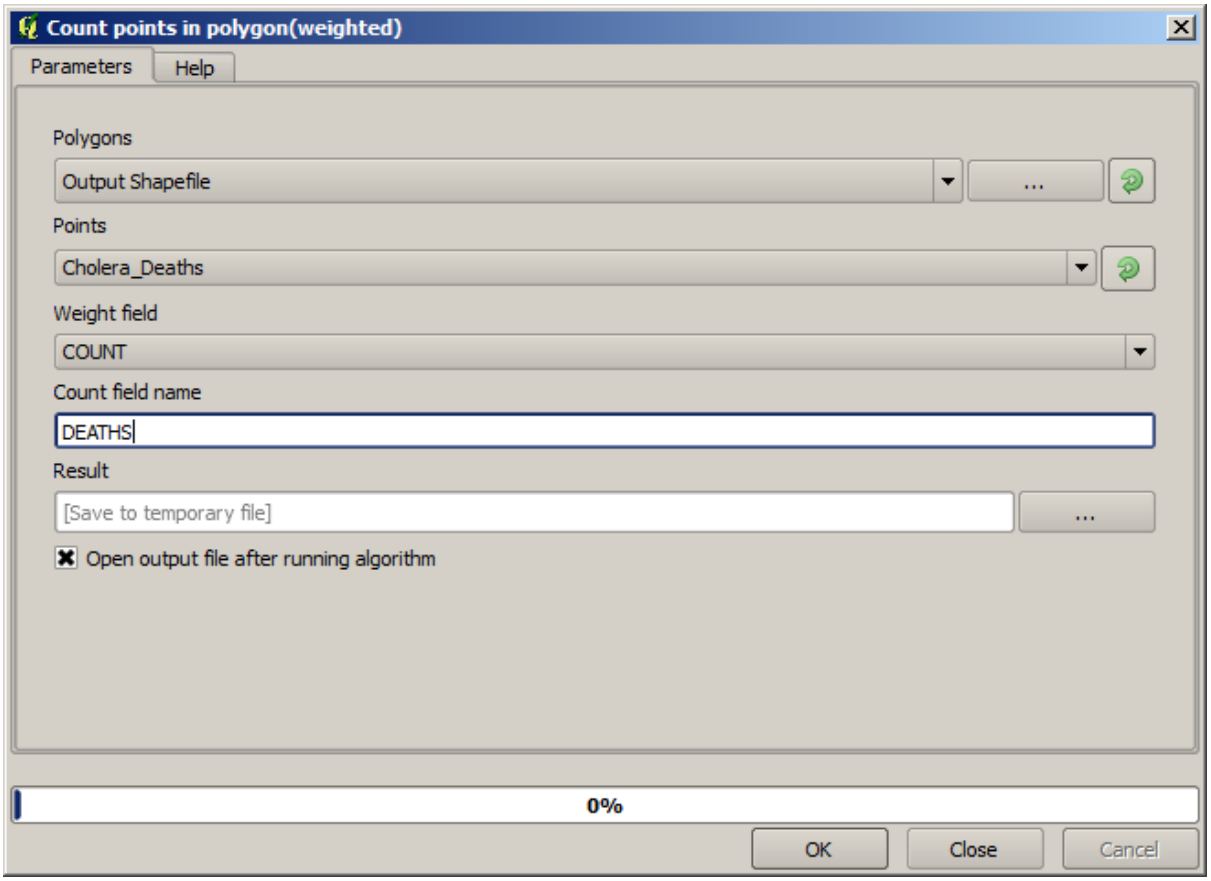


かなり簡単ですが、それですでに興味深い情報が得られます。

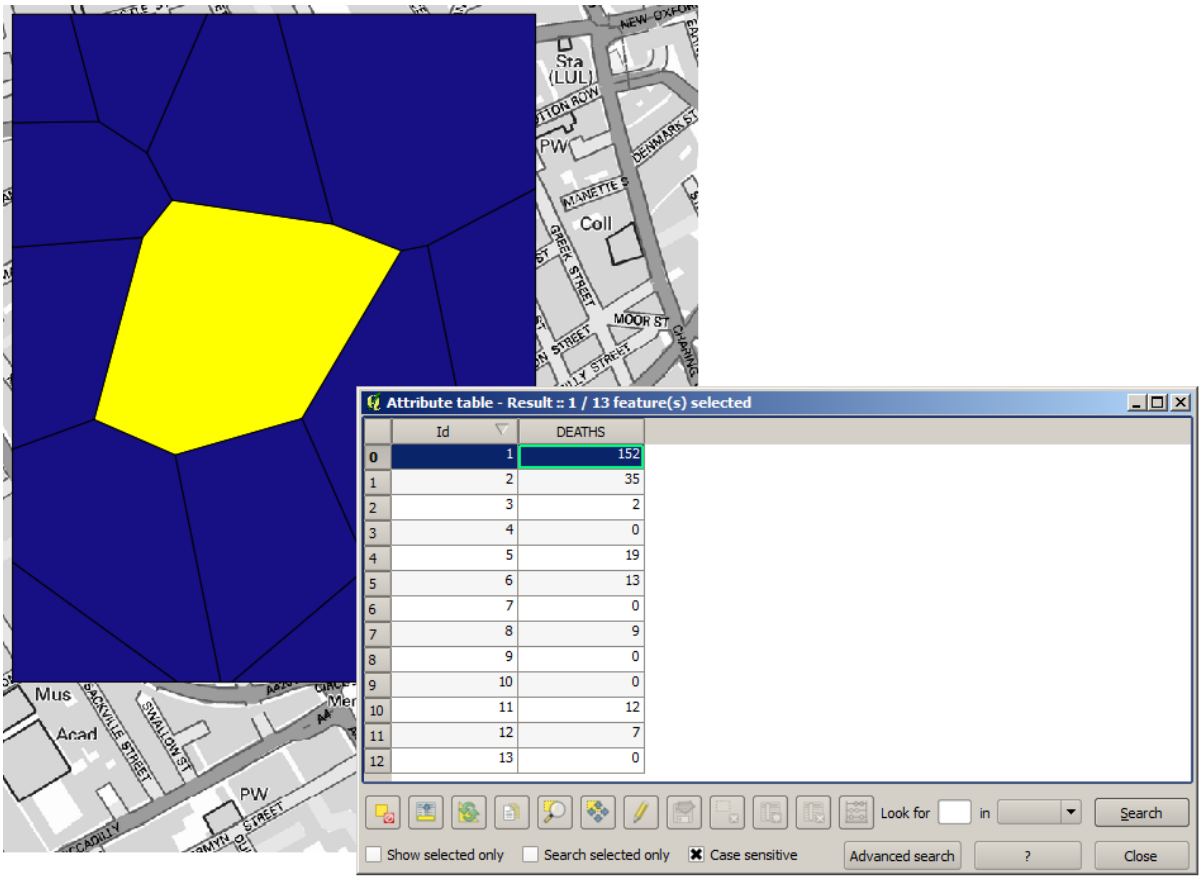


明らかに、ほとんどの患者がポリゴンの 1 つの範囲内にあります

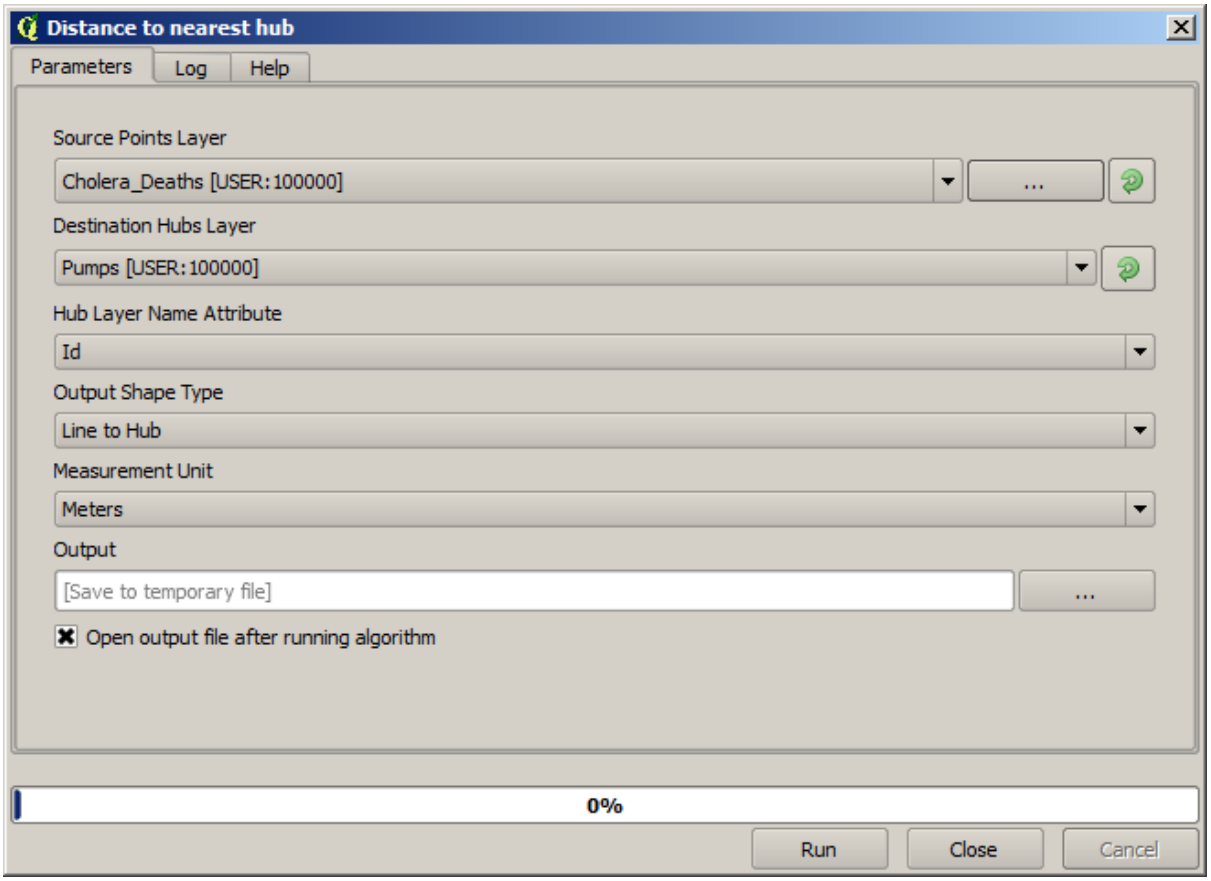
To get a more quantitative result, we can count the number of deaths in each polygon. Since each point represents a building where deaths occured, and the number of deaths is stored in an attribute, we cannot just count the points. We need a weighted count, so we will use the *Count points in polygon* tool.



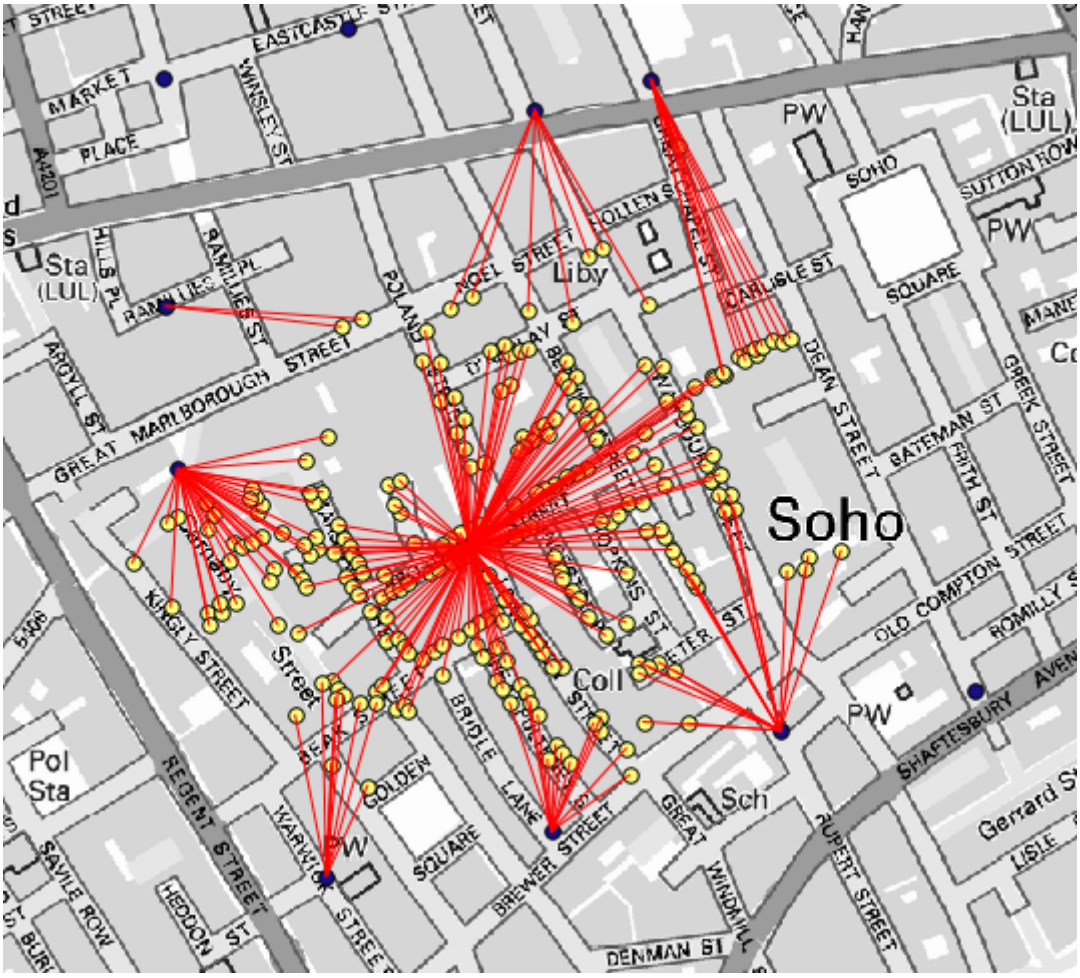
The new field will be called `DEATHS`, and we use the `COUNT` field as weighting field. The resulting table clearly reflects that the number of deaths in the polygon corresponding to the first pump is much larger than the other ones.



Another good way of visualizing the dependence of each point in the `Cholera_deaths` layer with a point in the `Pumps` layer is to draw a line to the closest one. This can be done with the *Distance to nearest hub* tool, and using the configuration shown next.

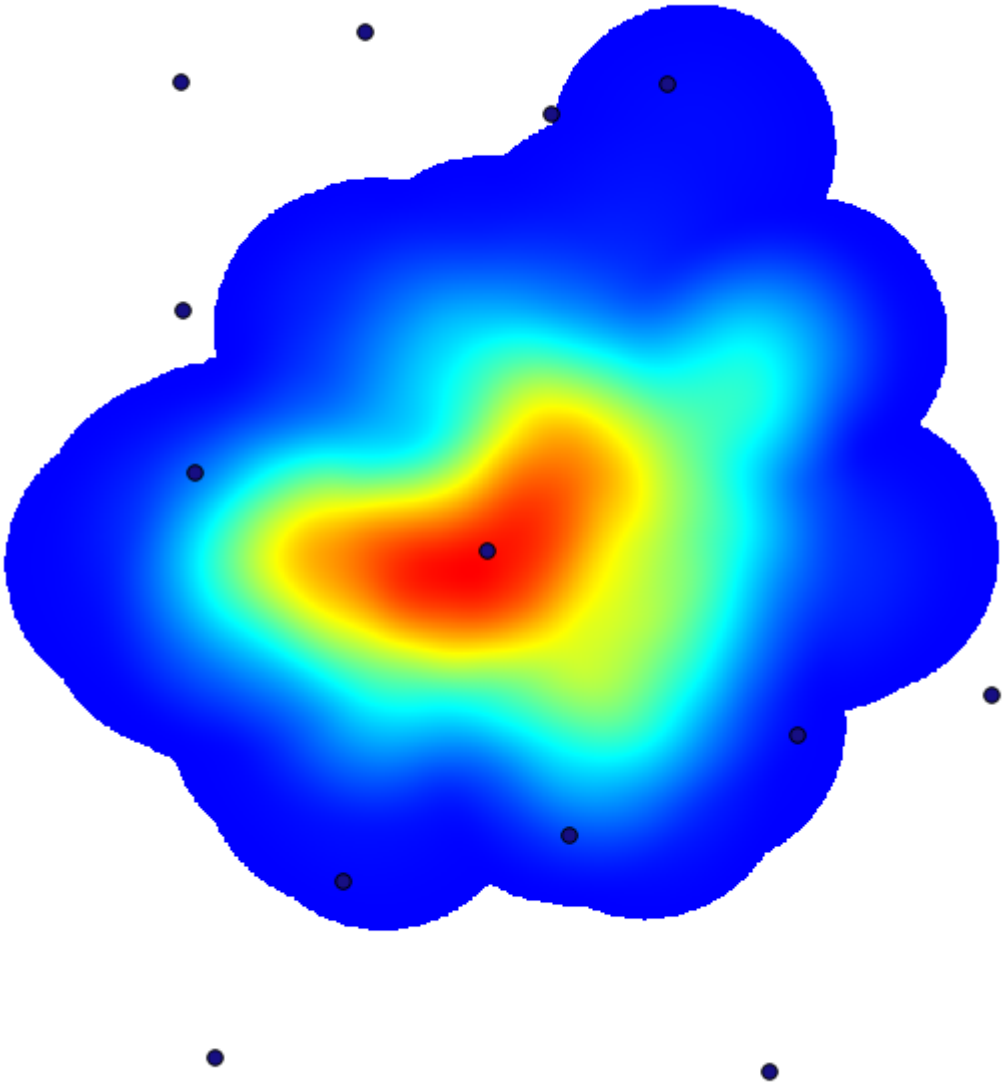


これの結果は次の通りです:

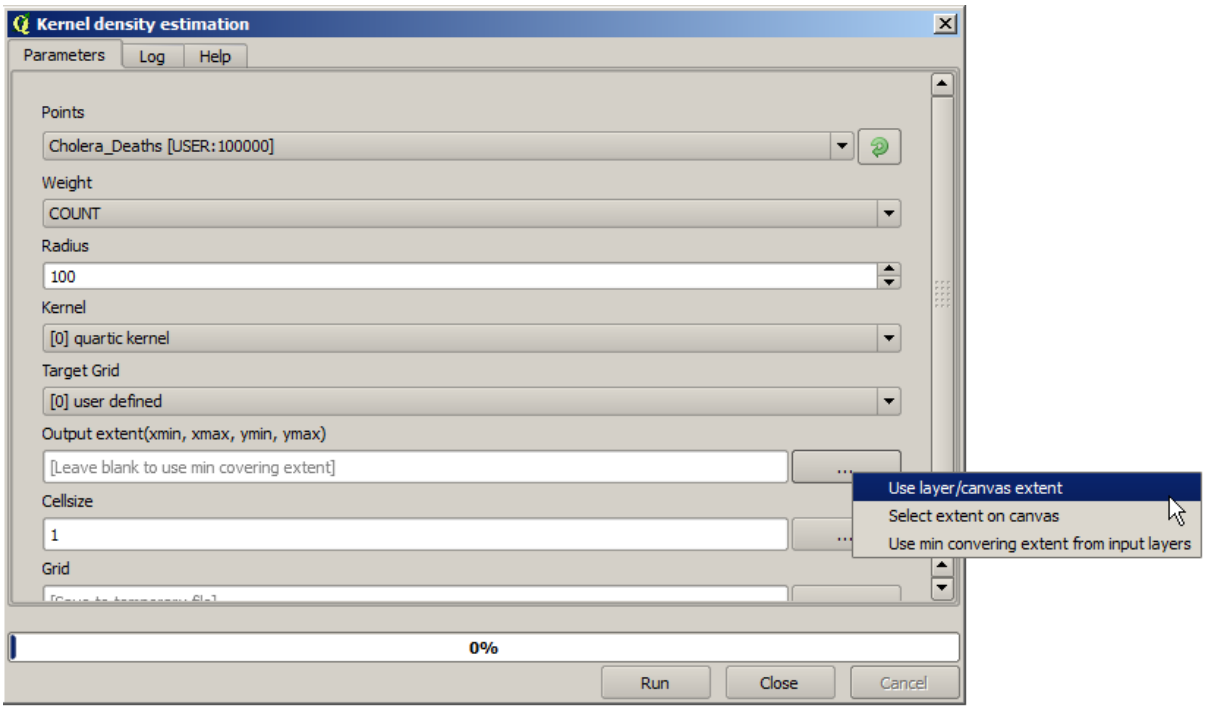


線の数は中央のポンプの方が多いですが、これは死者数ではなくコレラ患者が発見された場所の数を表していることを忘れないでください。それは代表的なパラメータですが、ある場所では他の場所よりも多くの患者が発生している可能性が考慮されていません。

A density layer will also give us a very clear view of what is happening. We can create it with the *Heatmap (Kernel density estimation)* algorithm. Using the `Cholera_deaths` layer, its `COUNT` field as weight field, with a radius of 100, the extent and cell size of the streets raster layer, we get something like this.



Remember that, to get the output extent, you do not have to type it. Click on the button on the right-hand side and select *Use layer/canvas extent*.



街路ラスタレイヤを選択すると、その範囲が自動的にテキストフィールドに追加されます。セルサイズでも同じことを行い、そのレイヤのセルサイズを選択する必要があります。

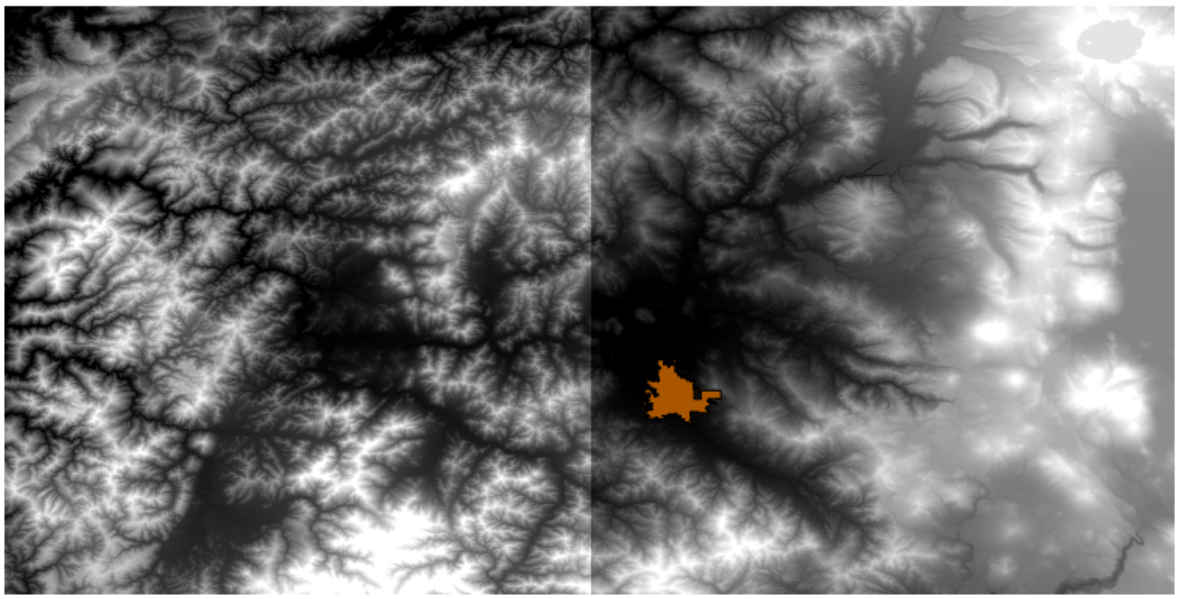
ポンプレイヤと組み合わせると、死亡例の密度が最大となるホットスポットには明らかにポンプが 1 台あることが分かります。

17.15 ラスタレイヤをクリップしてマージする

注釈

このレッスンでは、現実の世界のシナリオで地理アルゴリズムを継続して使用する、空間データの準備の別の例が表示されます。

このレッスンでは、ポリゴン 1 つだけのベクタレイヤによって与えられる市街地を囲む領域に対して傾斜レイヤを計算しようとしています。ベース DEM は 2 つのラスタレイヤに分割され、併せると作業したい都市の周りのものよりはるかに大きい領域をカバーしています。このレッスンに対応したプロジェクトを開くと、次のように表示されます。



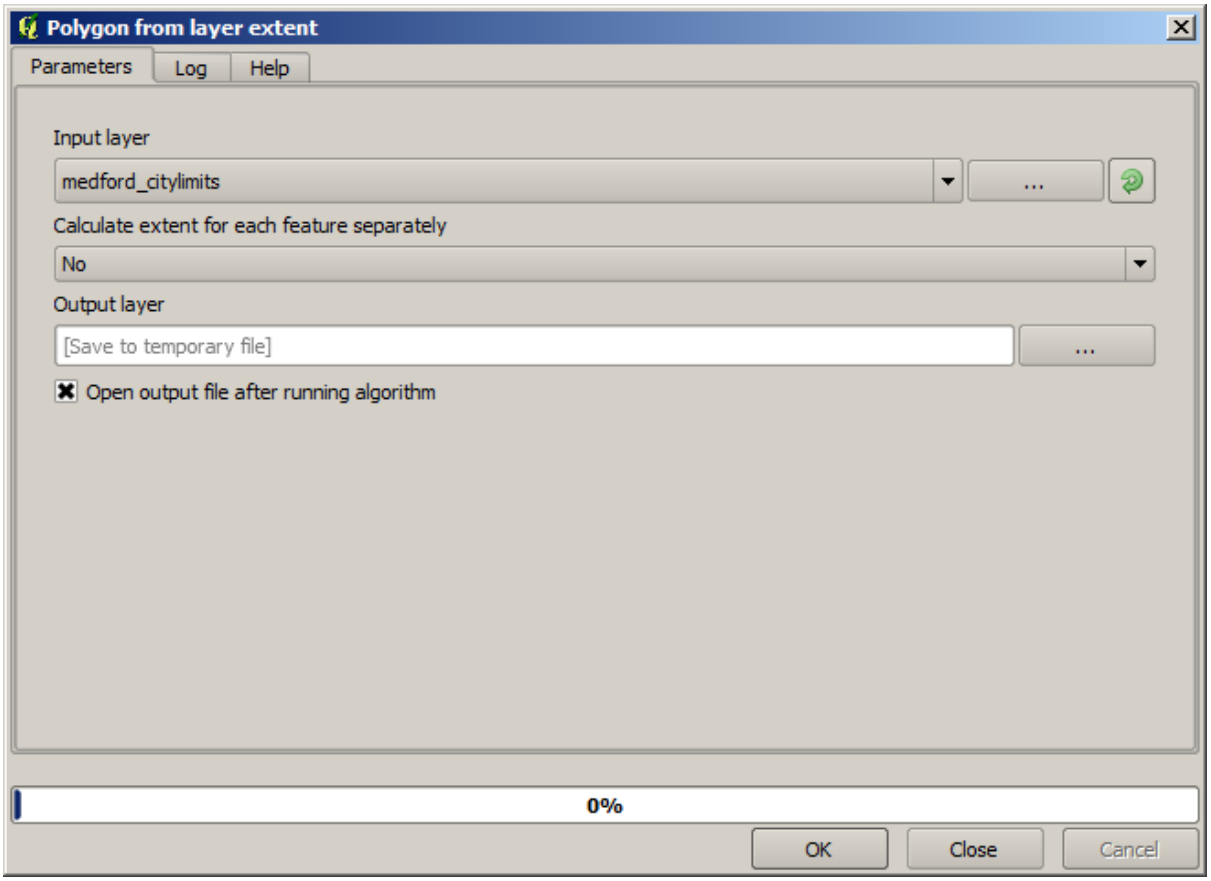
これらのレイヤには二つの問題があります。

- それらは希望するよりずっと大きい領域をカバーしています（興味があるのは市内中心部の周りのより小さな領域）
- それらは 2 つの異なるファイルにあります（市域は 1 つだけのラスタレイヤに入るが、言われているように、その周りにいくつかの余分な面積が欲しい）。

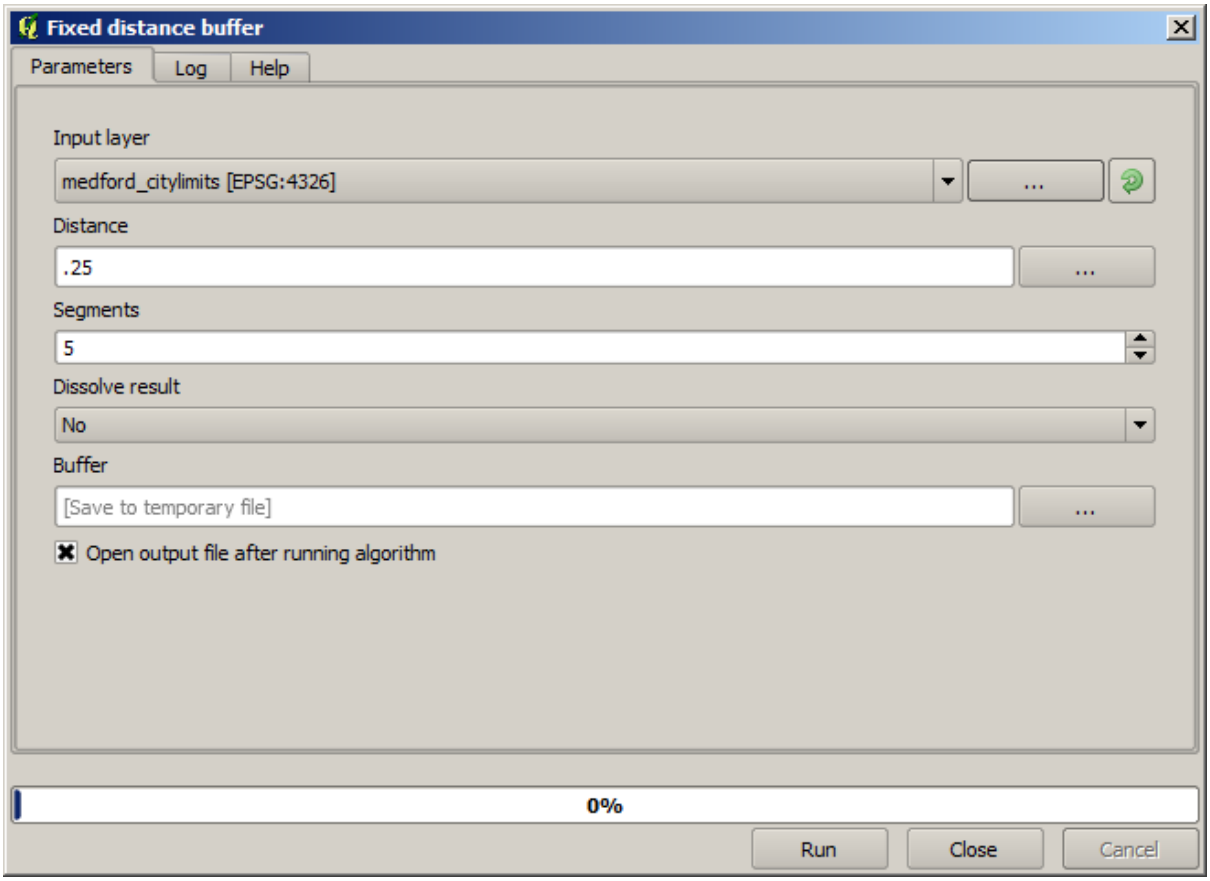
それらの両方が適切な地理アルゴリズムで簡単に解決できます。

まず、望む領域を定義する矩形を作成します。これを行うには、市の面積の限界を有するレイヤのバウンディングボックスを含むレイヤを作成し、それから厳密に必要であるよりもう少しカバーするラスタレイヤを有するように、それをバッファリングします。

To calculate the bounding box , we can use the *Extract layer extent* algorithm



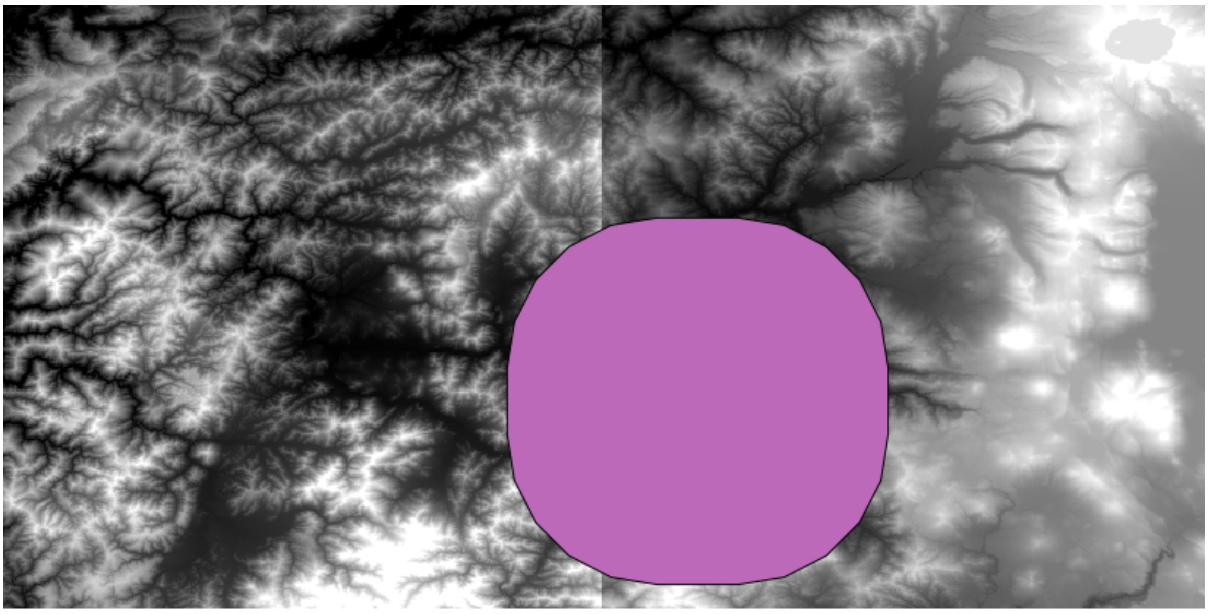
To buffer it, we use the *Buffer* algorithm, with the following parameter values.



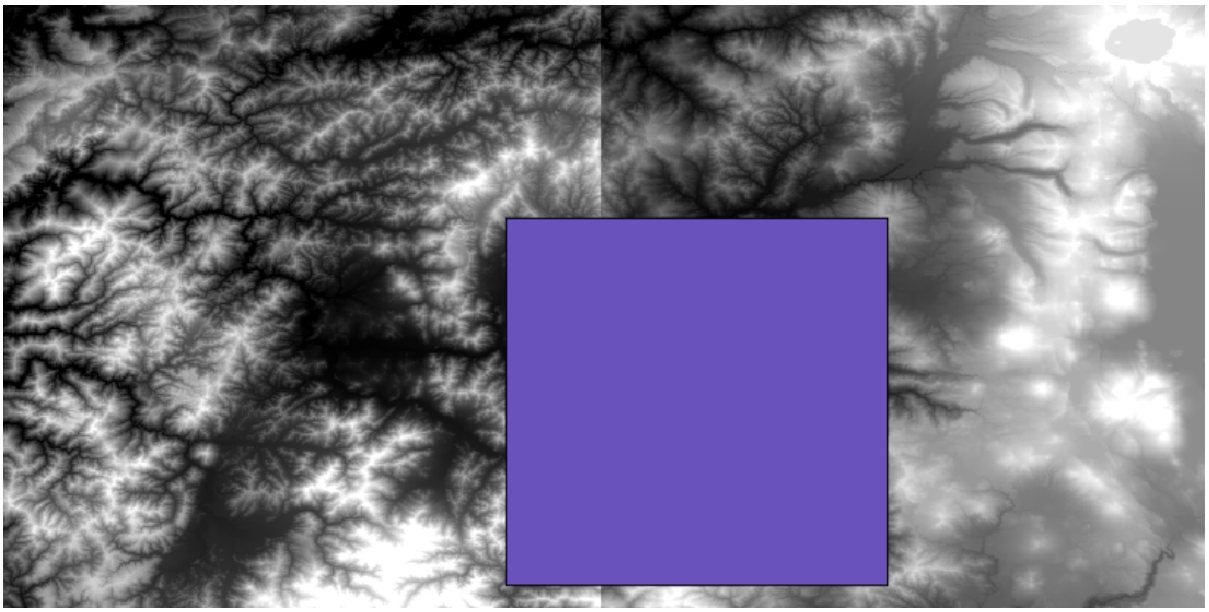
警告

構文は最近のバージョンで変更されました；距離とアークの頂点の両方を 0.25 に設定します

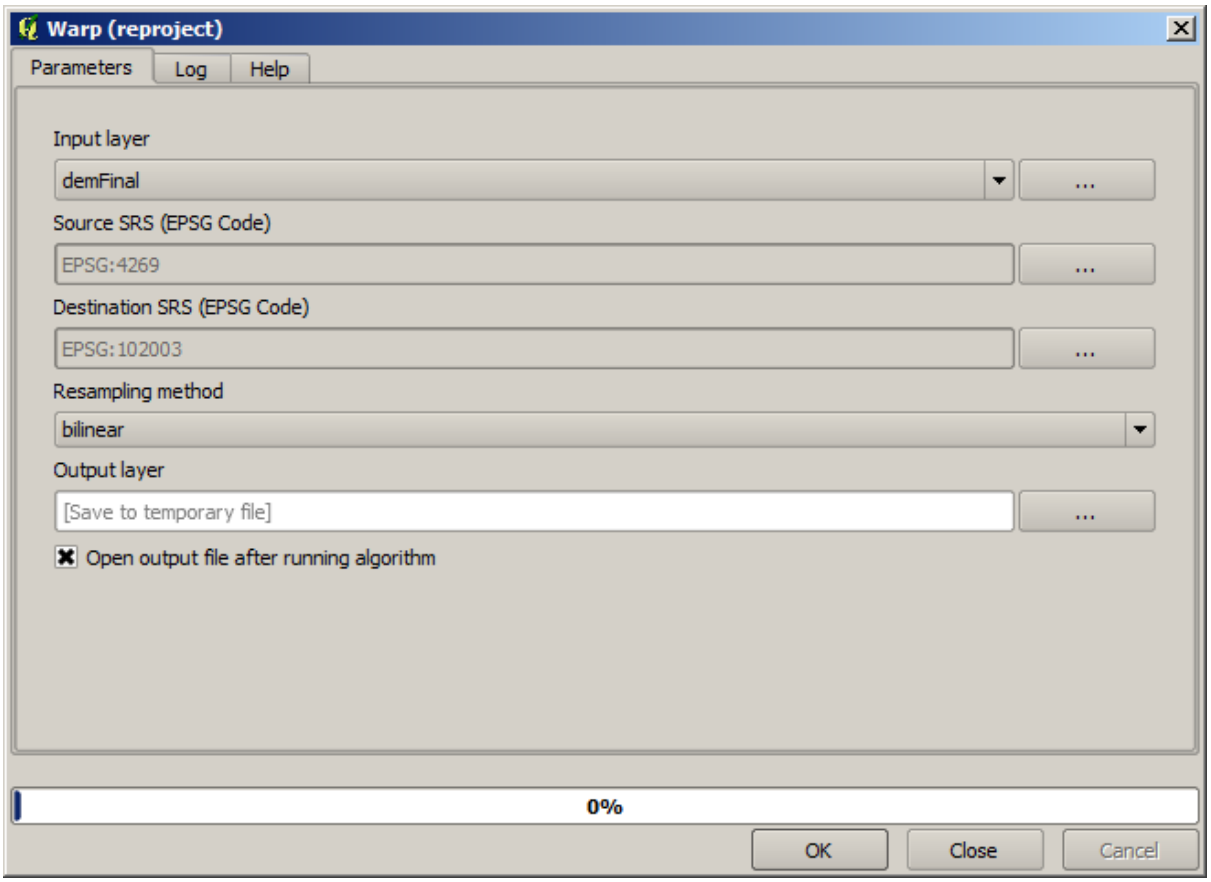
これが上に示したパラメーターを用いて得られた結果のバウンディングボックスです



It is a rounded box, but we can easily get the equivalent box with square angles, by running the *Extract layer extent* algorithm on it. We could have buffered the city limits first, and then calculate the extent rectangle, saving one step.

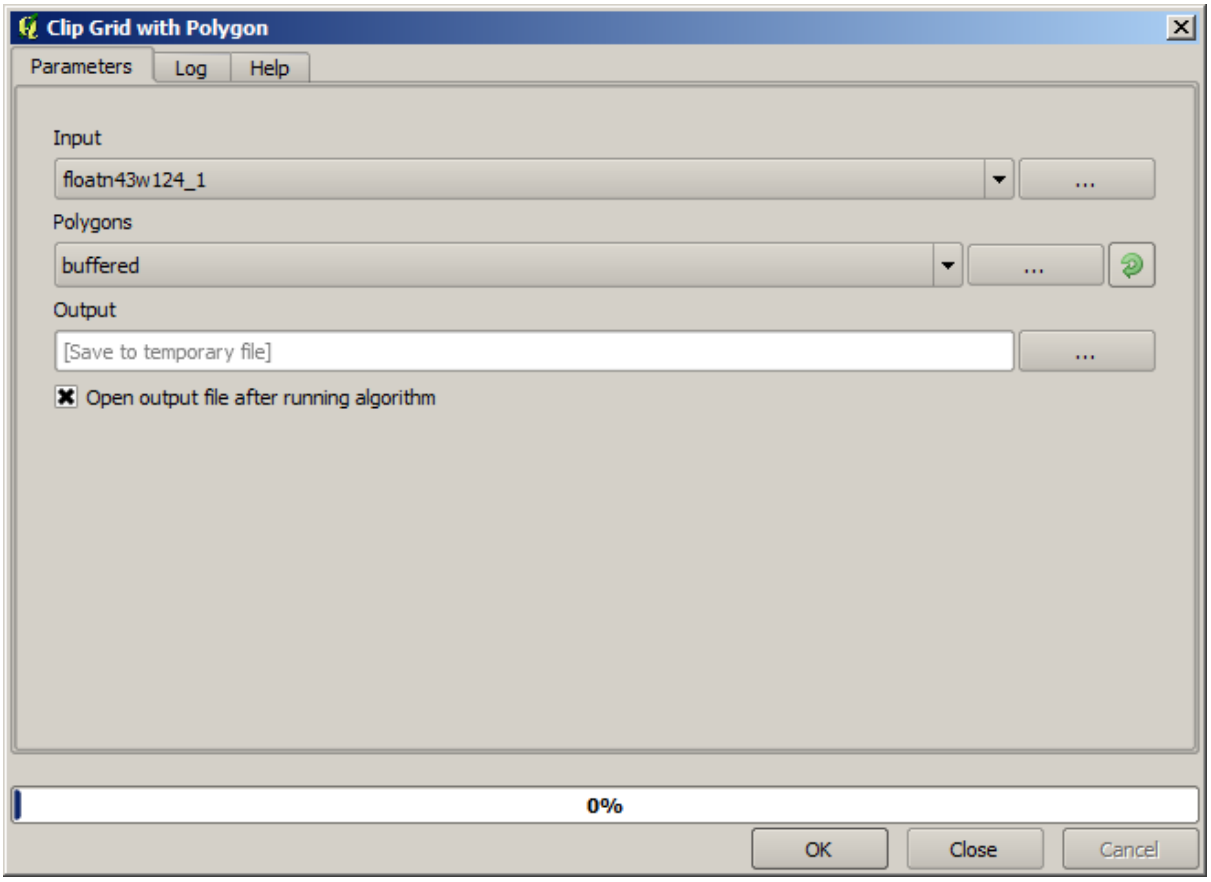


You will notice that the rasters has a different projection from the vector. We should therefore reproject them before proceeding further, using the *Warp (reproject)* tool.

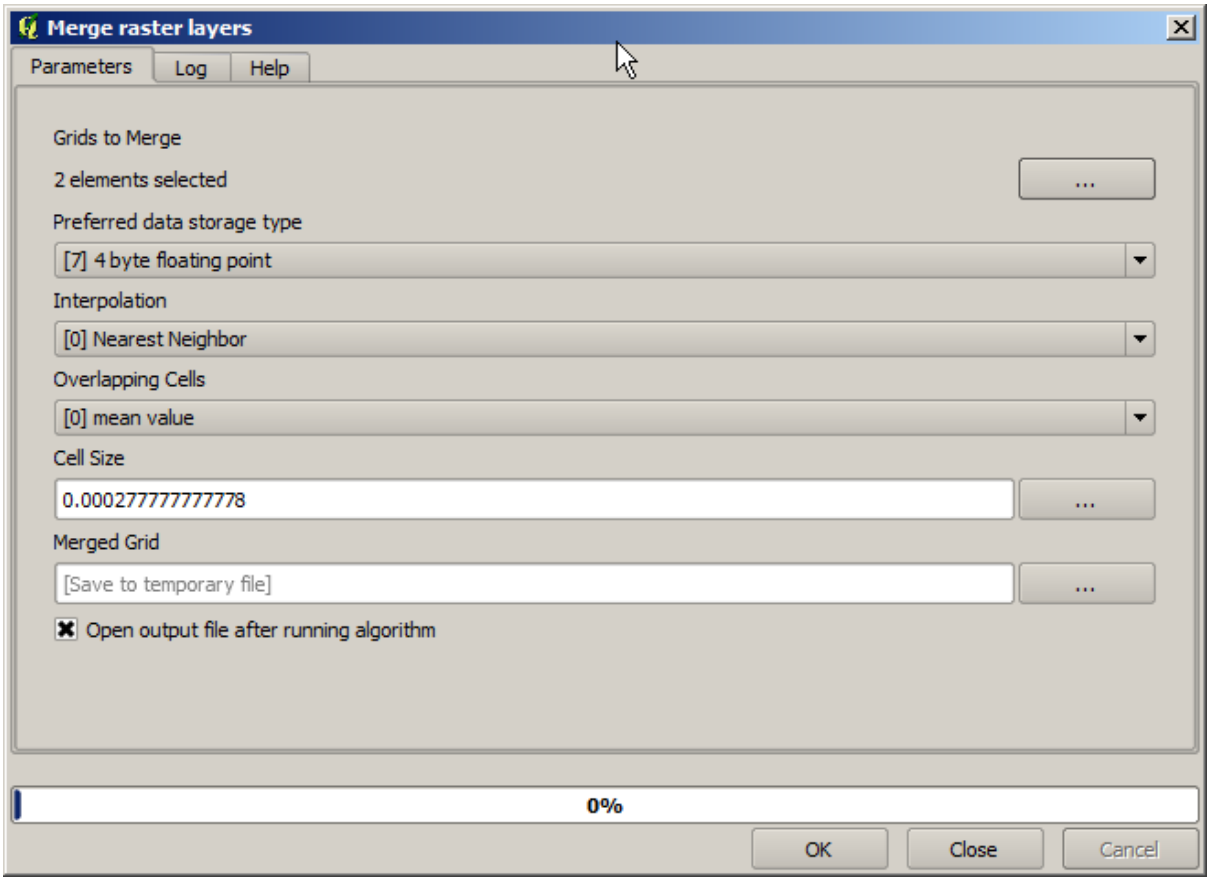


> **注釈**
>
> 最近のバージョンではより複雑なインターフェイスになっています。少なくとも 1 つの圧縮方式が選択されていることを確認します。

With this layer that contains the bounding box of the raster layer that we want to obtain, we can crop both of the raster layers, using the *Clip raster by mask layer* algorithm.



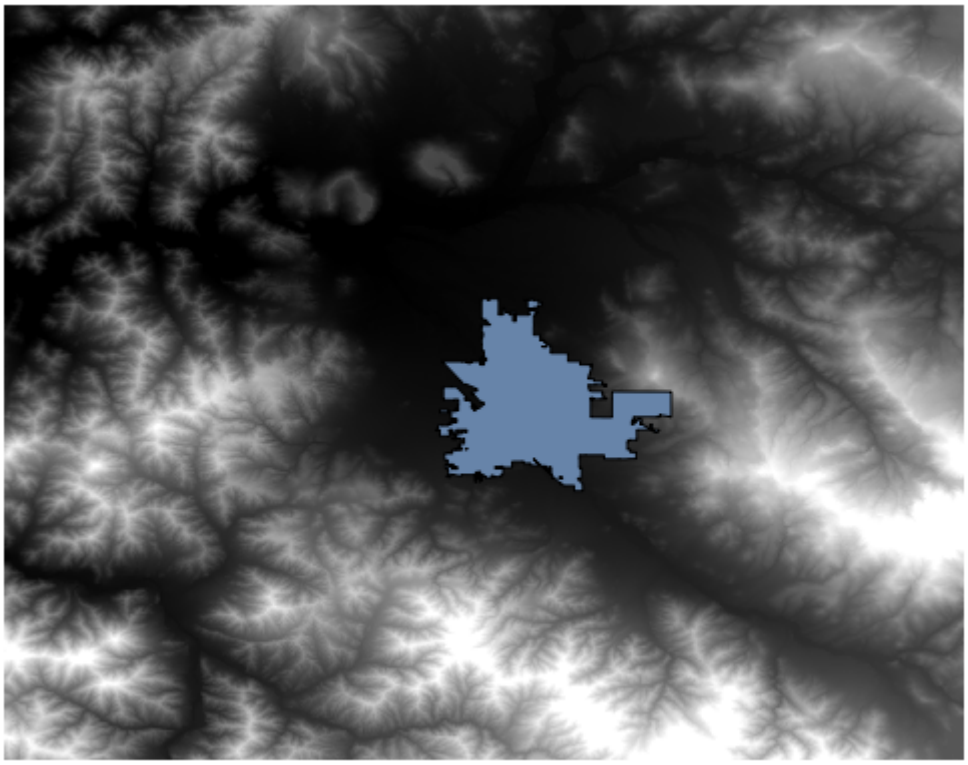
Once the layers have been cropped, they can be merged using the SAGA *Mosaic raster layers* algorithm.



注釈

最初にマージしてからトリミングすると時間を節約でき、二回クリッピングアルゴリズムを呼び出さずにすむでしょう。しかしながら、マージする複数のレイヤがあってそれらがかなり大きなサイズを持っている場合、それが後工程に処理が困難であるよりも大きなレイヤになってしまいます。その場合はクリッピングアルゴリズムを数回呼び出す必要があります。時間がかかるかもしれませんが、心配しないで。その操作を自動化するためにいくつかの追加のツールがあることがすぐにわかりますから。この例では、レイヤは 2 つだけなので、今それを心配することはありません。

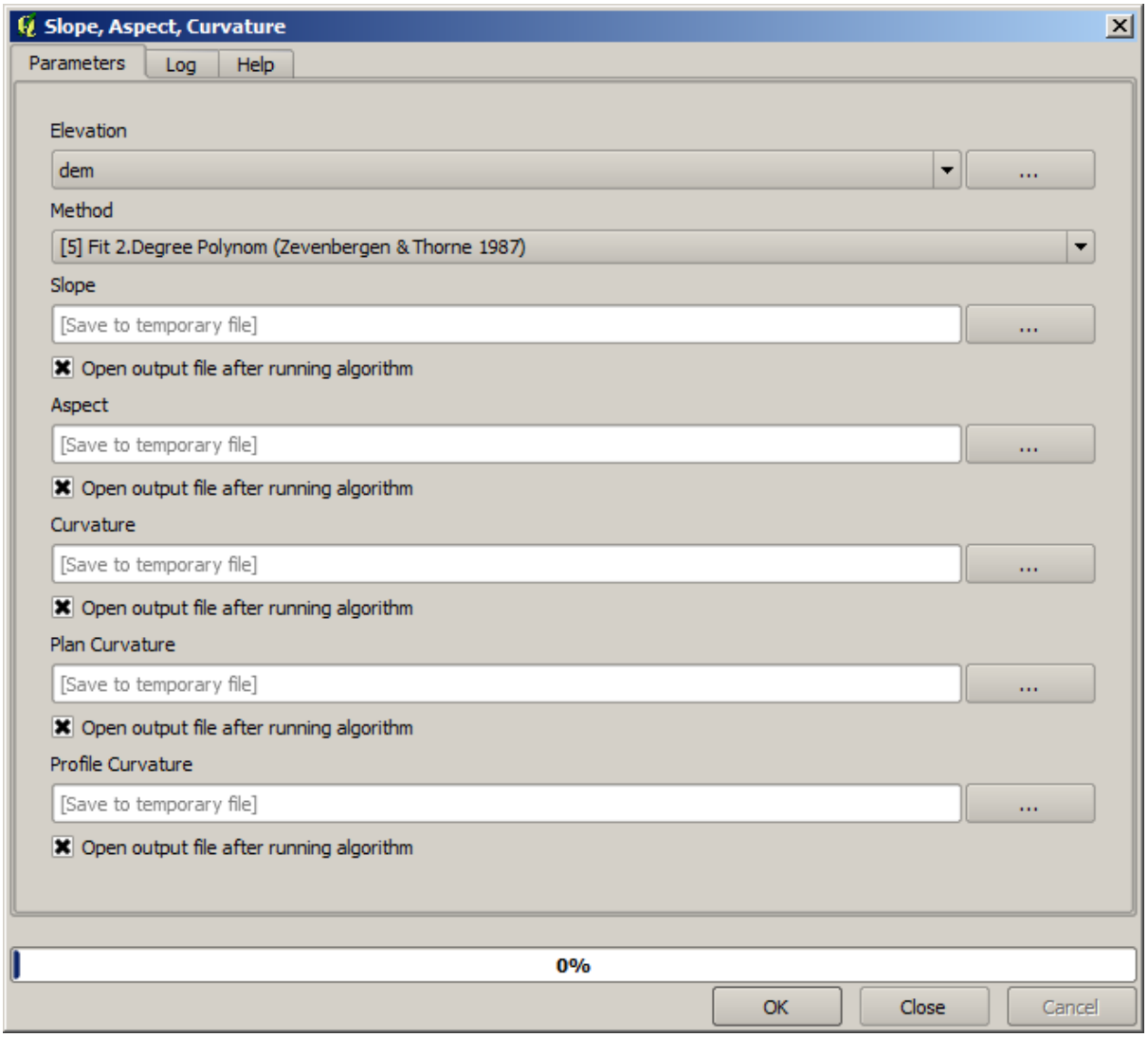
それによって、私たちが望む最後の DEM が得られます。



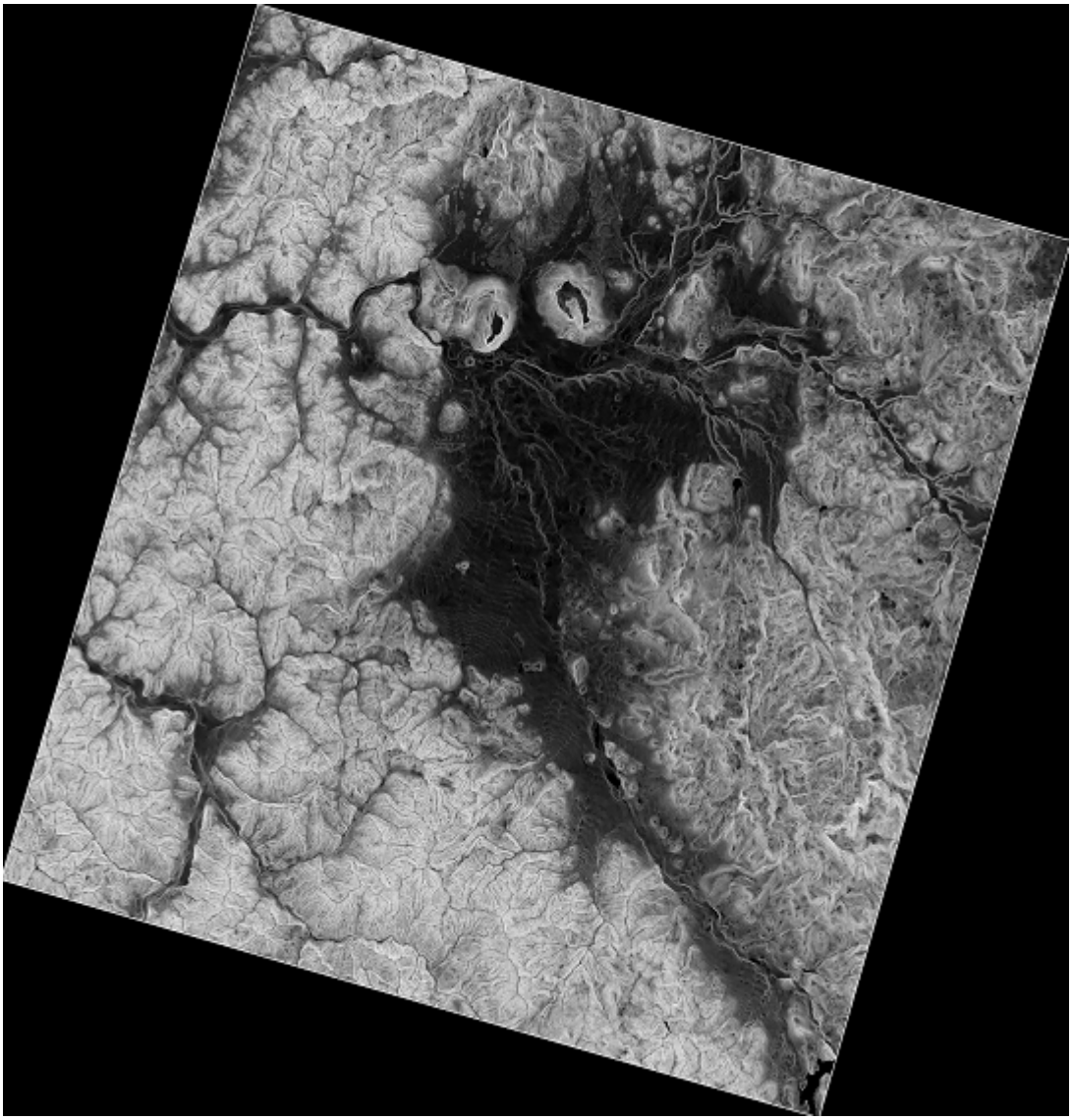
では傾斜レイヤを計算しましょう。

A slope layer can be computed with the *Slope* algorithm, but the DEM obtained in the last step is not suitable as input, since elevation values are in meters but cellsize is not expressed in meters (the layer uses a CRS with geographic coordinates). A reprojection is needed. To reproject a raster layer, the *Warp (reproject)* algorithm can be used again. We reproject into a CRS with meters as units (e.g. 3857), so we can then correctly calculate the slope, with either SAGA or GDAL.

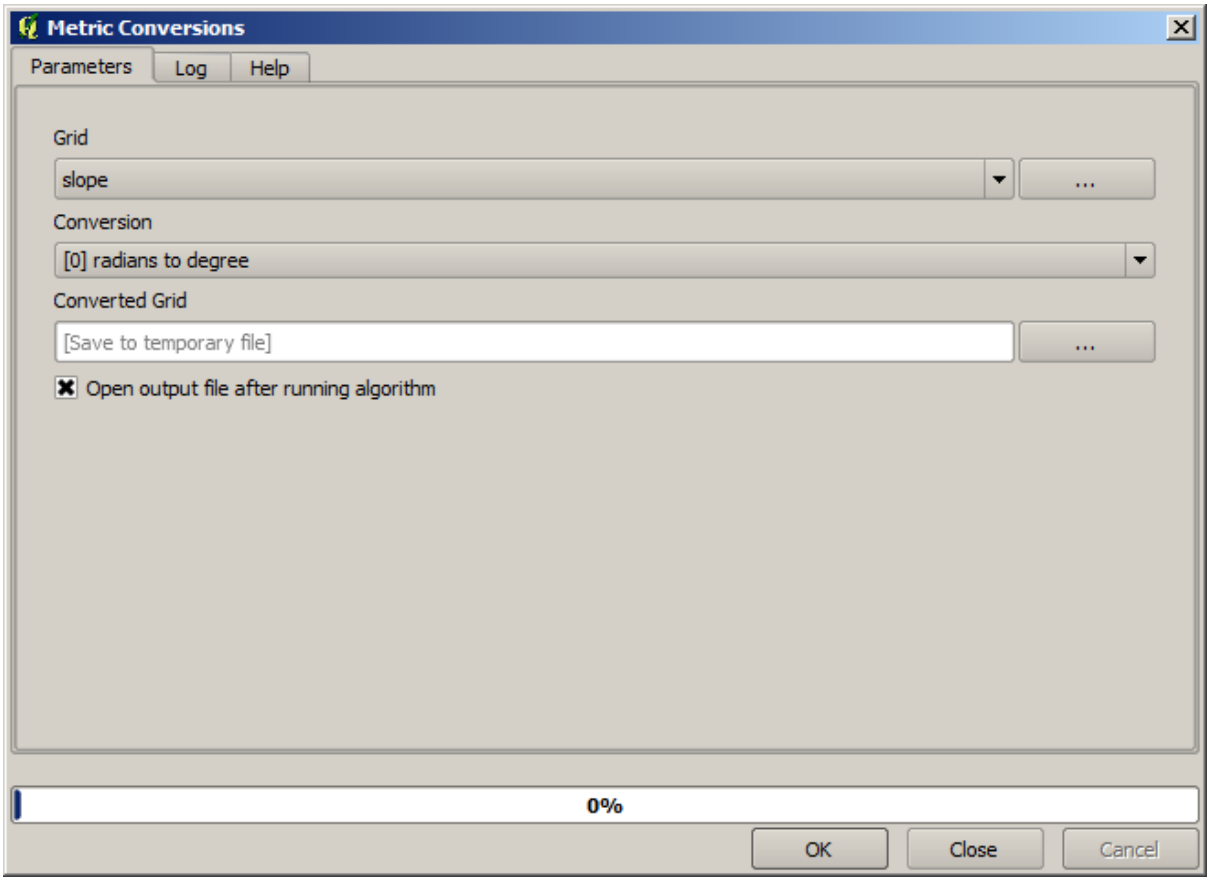
新 DEM では、傾きが計算できるようになりました。



そして、これが結果の傾斜レイヤです。



The slope produced by the *Slope* algorithm can be expressed in degrees or radians; degrees are a more practical and common unit. In case you calculated it in radians, the *Metric conversions* algorithm will help us to do the conversion (but in case you didn't know that algorithm existed, you could use the raster calculator that we have already used).



Reprojecting the converted slope layer back with the *Reproject raster layer*, we get the final layer we wanted.

再投影プロセスでは、最初のステップの 1 つで計算されたバウンディングボックス外のデータを最終レイヤが格納するようにしている可能性があります。これは、ベース DEM を得るためにしたのと同じように、それを再びクリッピングすることによって解決できます。

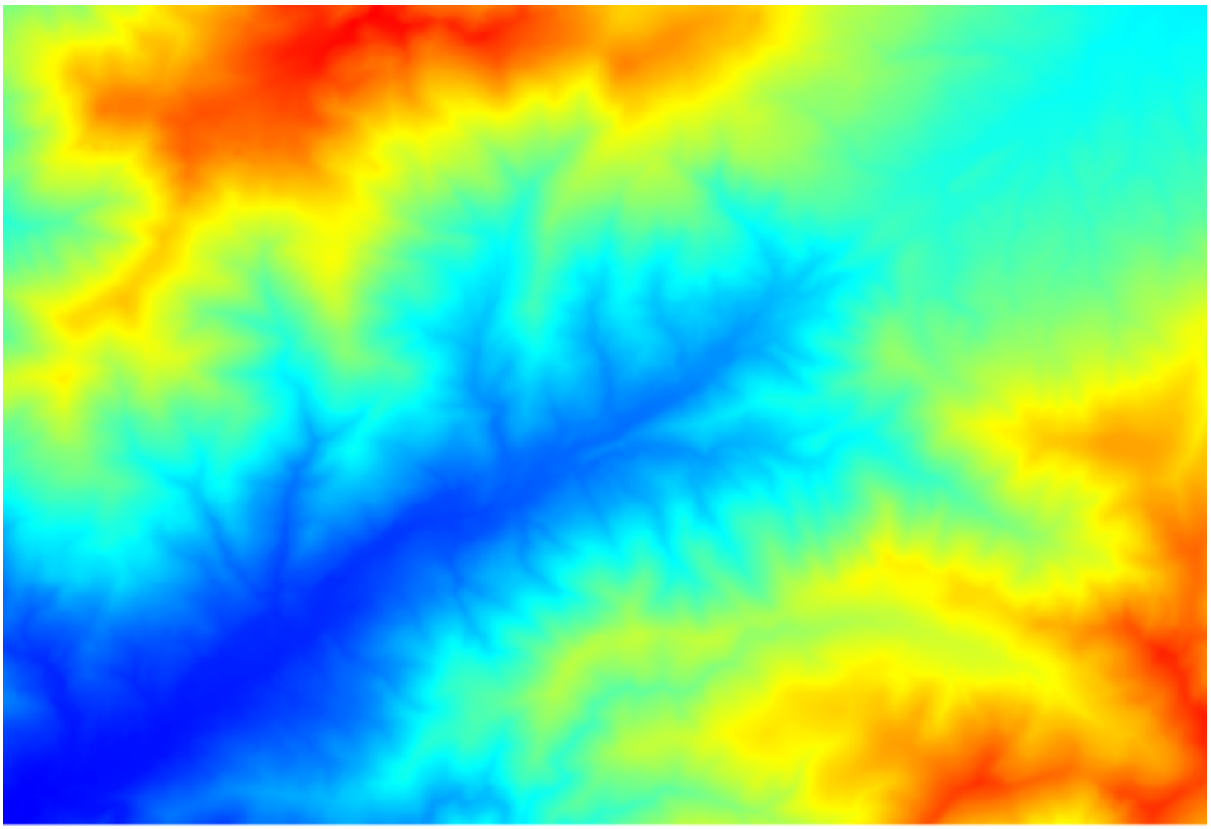
17.16 水文解析

注釈

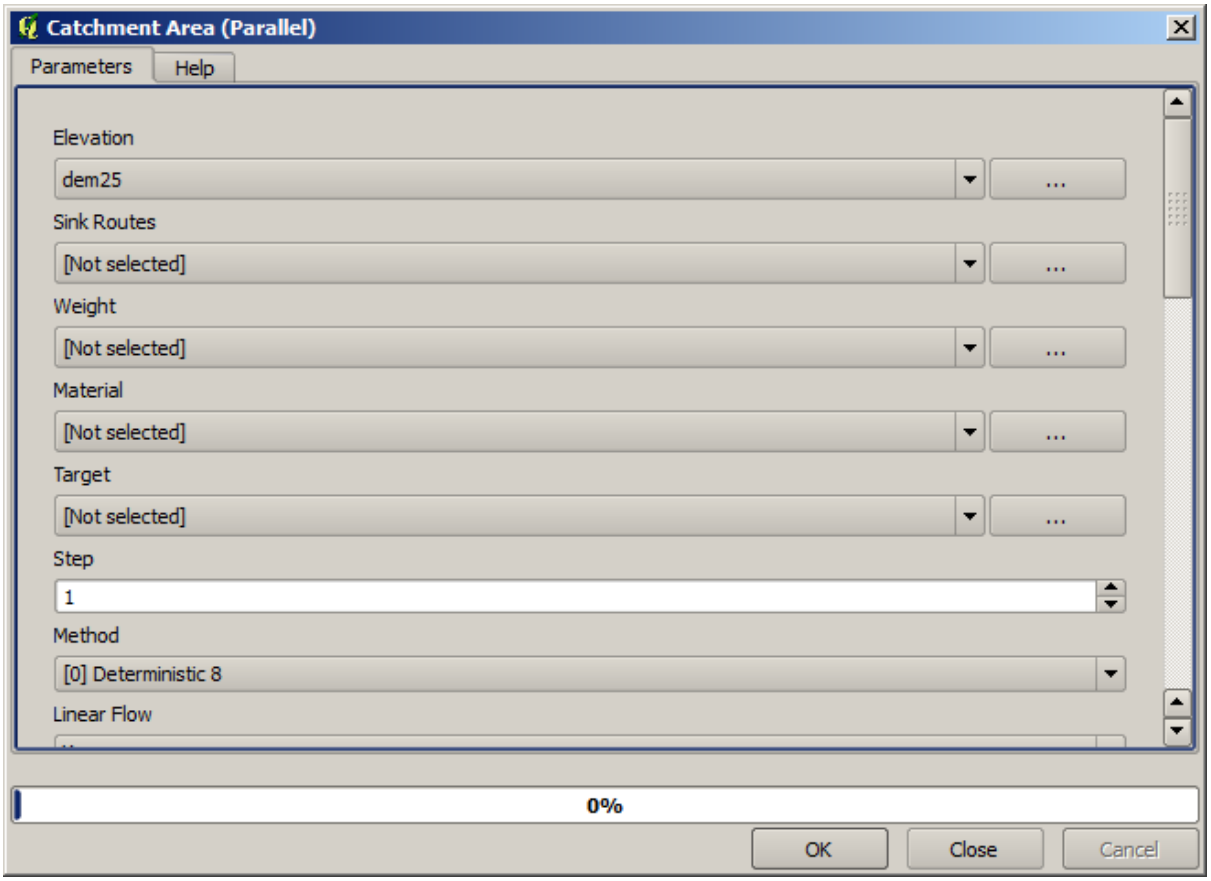
このレッスンでは、ちょっとした水文解析を実行します。この分析は解析ワークフローの非常に良い例を構成しているので、後のいくつかのレッスンの一部に使用されます。そして、いくつかの高度な機能を発揮するためにそれを使用します。

目的：DEM から始めて、水路網を抽出し、流域を描写し、いくつかの統計を計算します。

1. 最初に DEM だけが含まれているレッスンデータを持つプロジェクトを読み込みます。

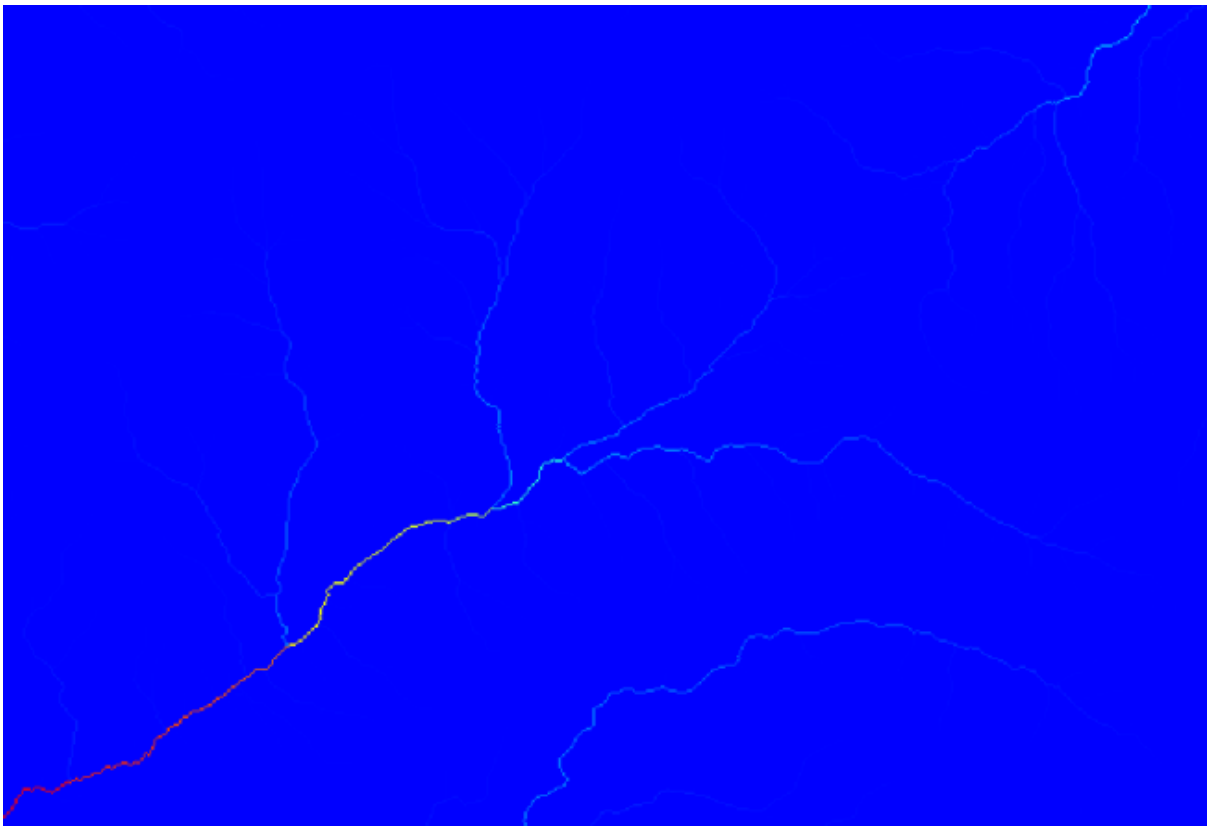


2. The first module to execute is *Catchment area* (in some SAGA versions it is called *Flow accumulation (Top Down)*). You can use any of the others named *Catchment area*. They have different algorithms underneath, but the results are basically the same.
3. **標高** フィールドで DEM を選択し、残りのパラメータはデフォルト値のままにしておきます。

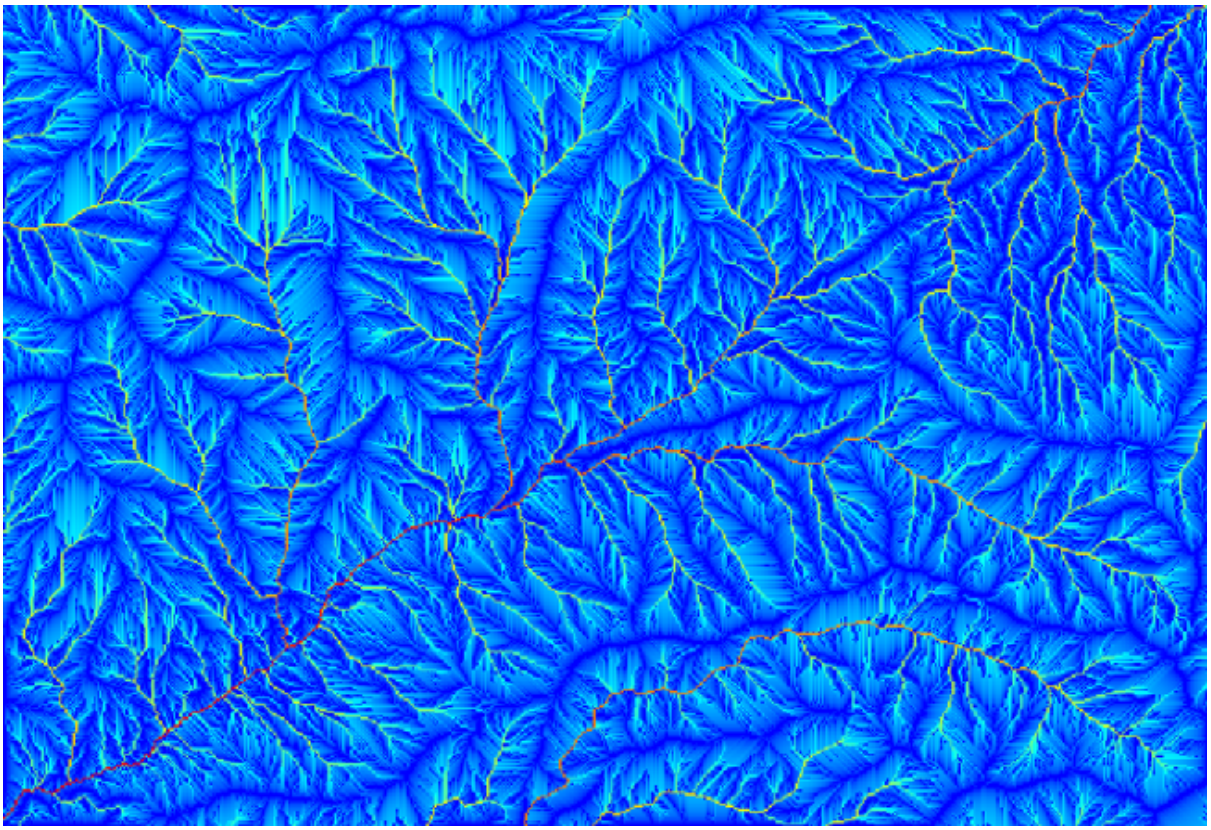


Some algorithms calculate many layers, but the *Catchment Area* layer is the only one we will be using. You can get rid of the other ones if you want.

レイヤのレンダリングは非常に有益ではありません。

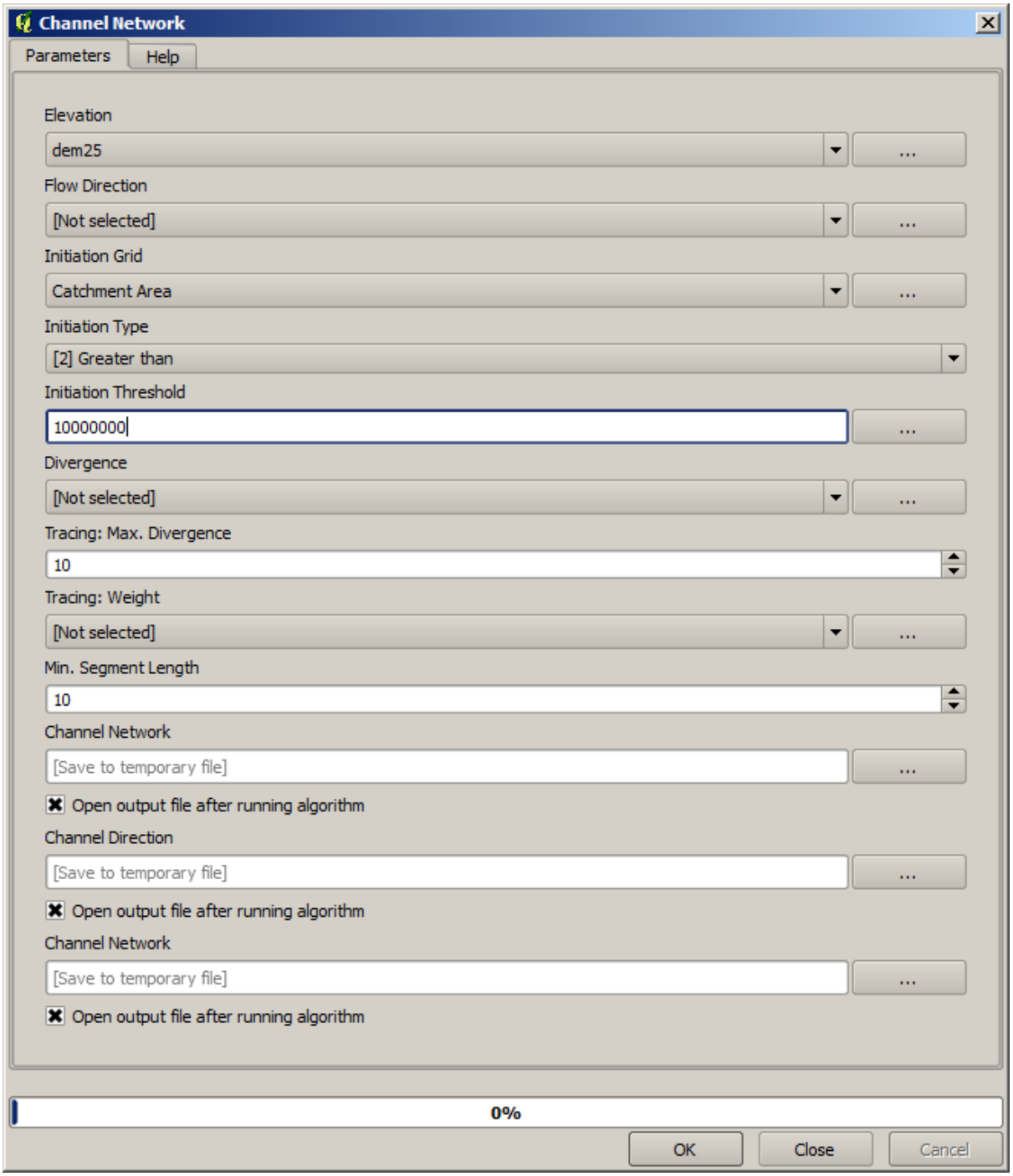


理由を知るために、ヒストグラムを見ると、値が均等に分散されていないことがわかります（非常に高い値のセルがいくつかあります。水路ネットワークに対応するものです）。 **ラスター計算機** アルゴリズムを使用して、集水域の値の領域の対数を計算すると、より多くの情報を含むレイヤーが得られます。

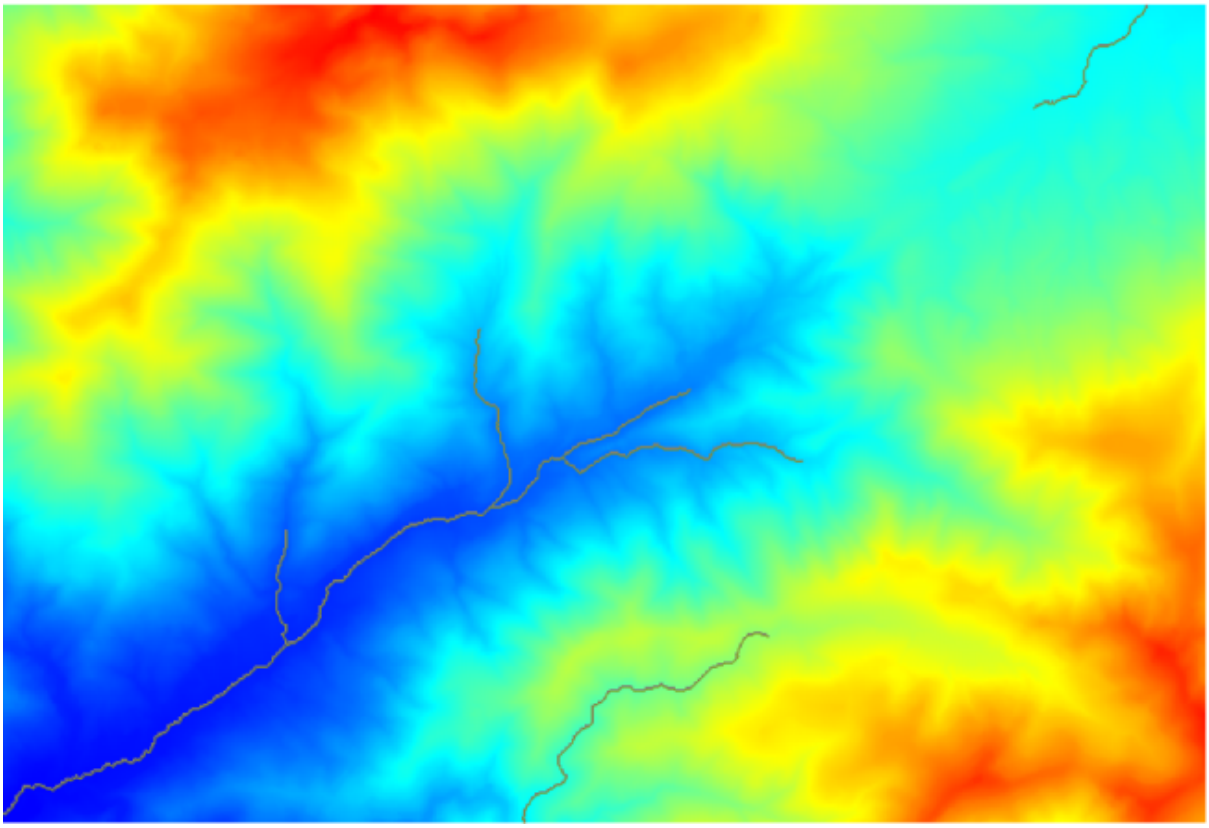


4. 集水域（累積流量としても知られます）は、水路の開始の閾値を設定するために使用することができます。これは **水路網** アルゴリズムを使用して行うことができます。

   - *Initiation grid*: 対数ではなく、集水域のレイヤを使用します。
   - *Initiation threshold*: `10.000.000`
   - *Initiation type*: *Greater than*

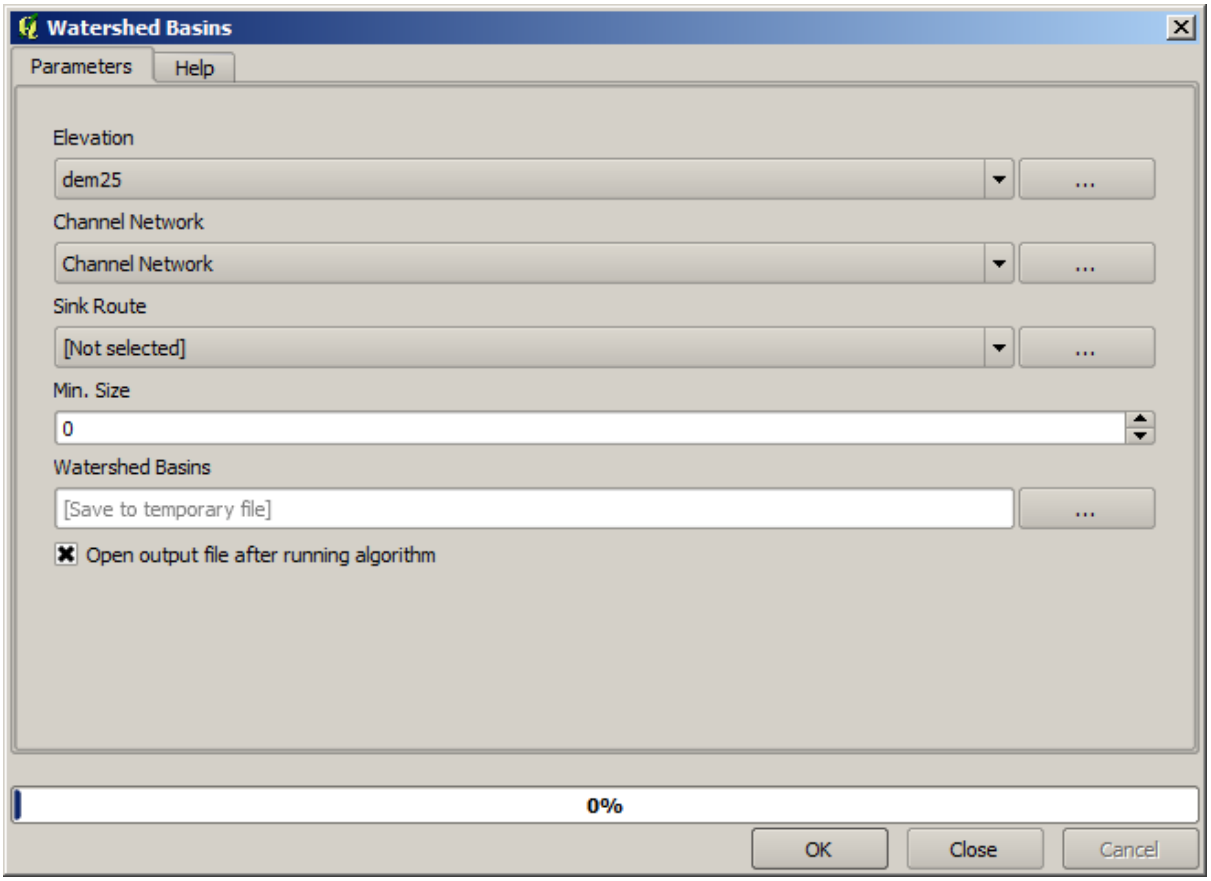


Initiation threshold の値を大きくすると、より疎な水路網が得られます。値を小さくすると、より密な網になります。提案された値では、このような結果になります。

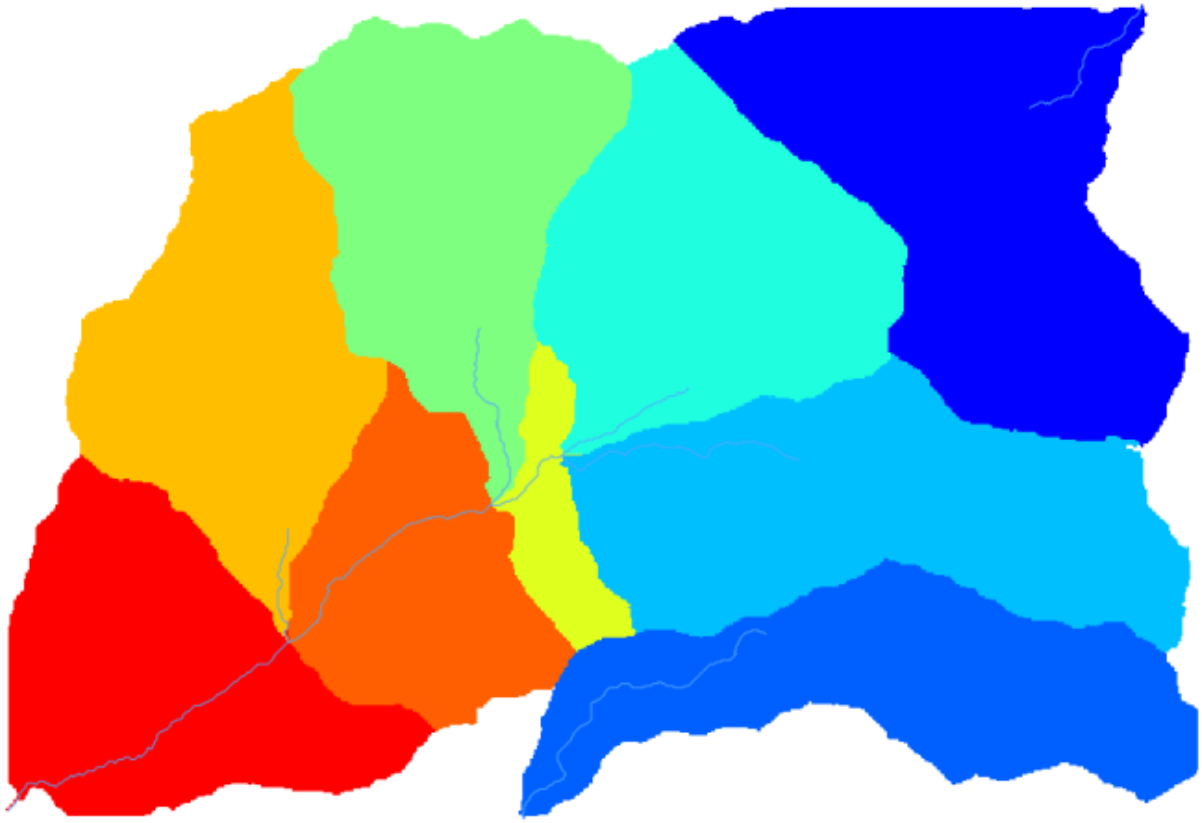


上の画像は、出来上がったベクタレイヤと DEM だけですが、同じ水路網を持つラスタレイヤも存在するはずです。このラスタレイヤが、これから使用するレイヤとなります。

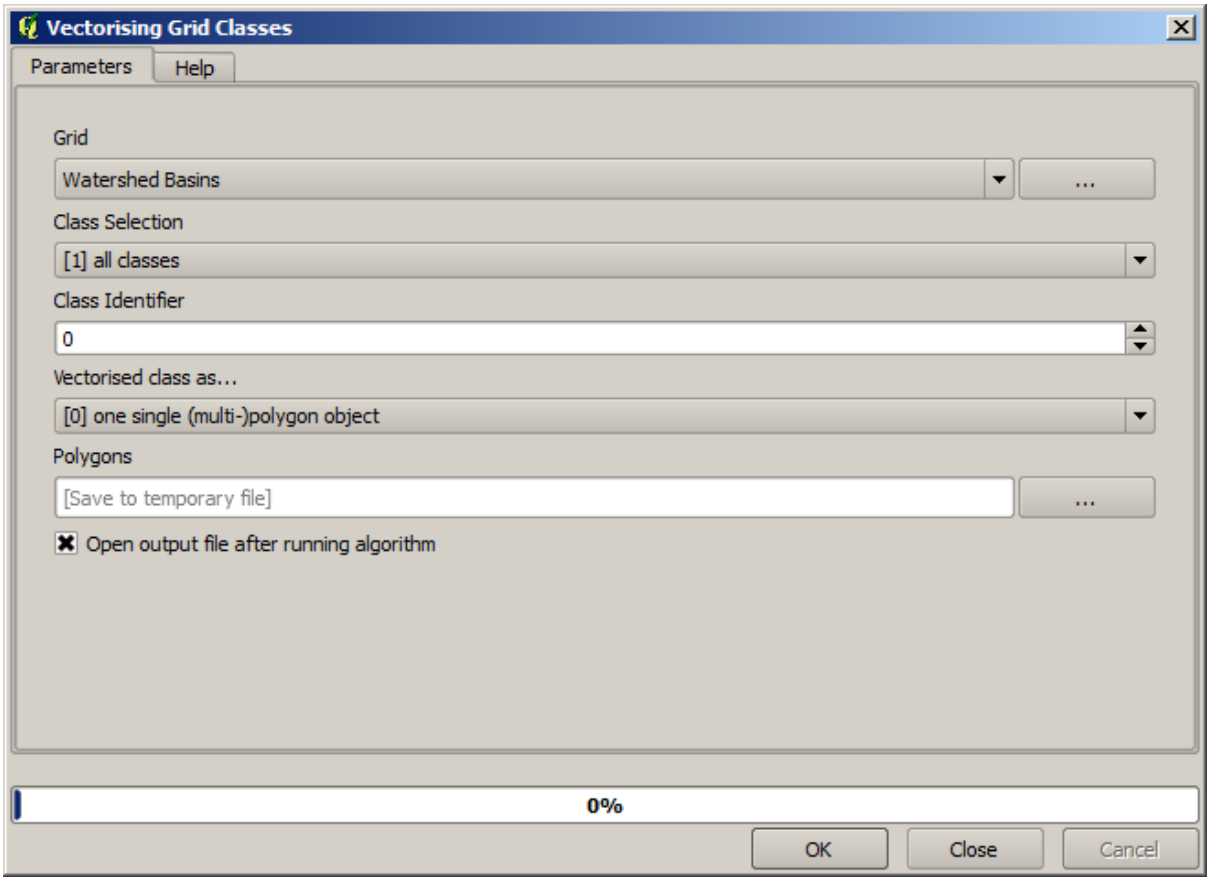
5. 次に、*Watersheds basins* アルゴリズムを使って、水路網に対応する第二次流域を、その中のすべてのジャンクションを出口点として定義します。ここでは、対応するパラメータをダイアログで設定する必要があります。

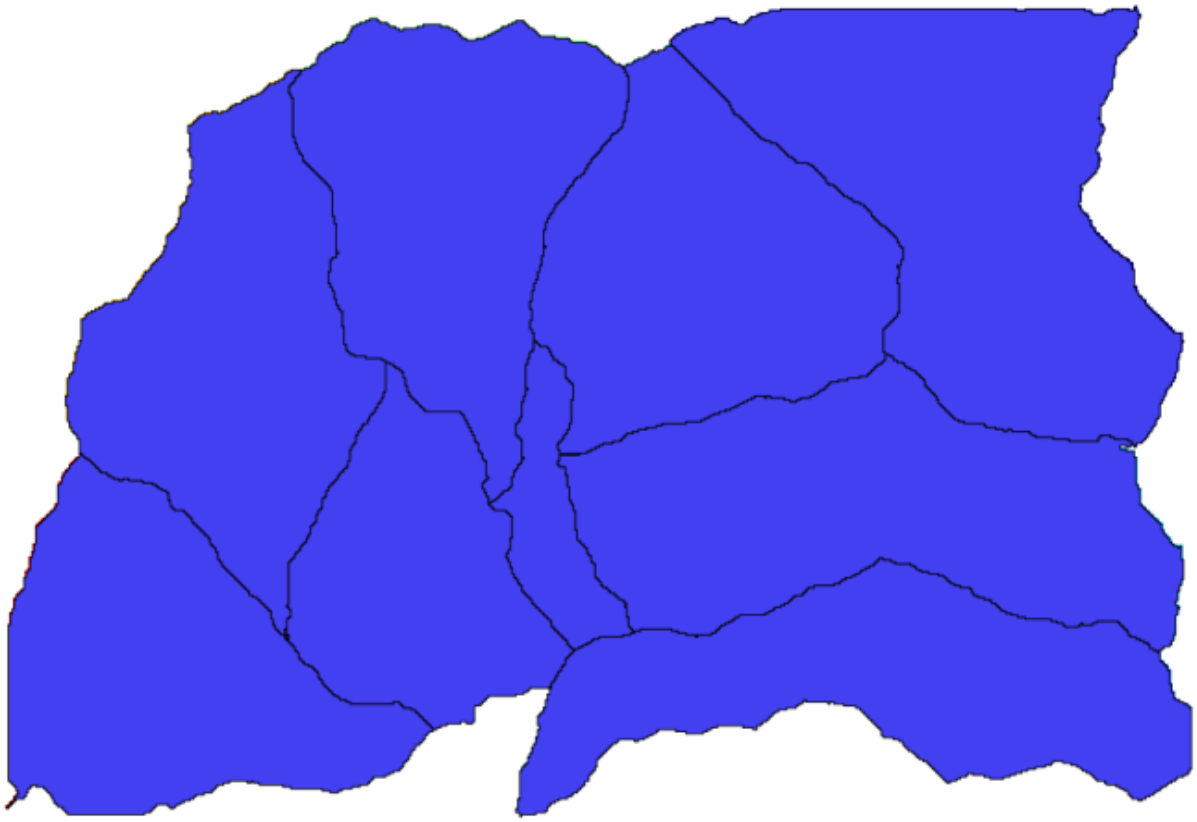


そして、得られるものはこれです。



6. これはラスタの結果です。*Vectorising grid classes* アルゴリズムを使ってベクタ化できます。

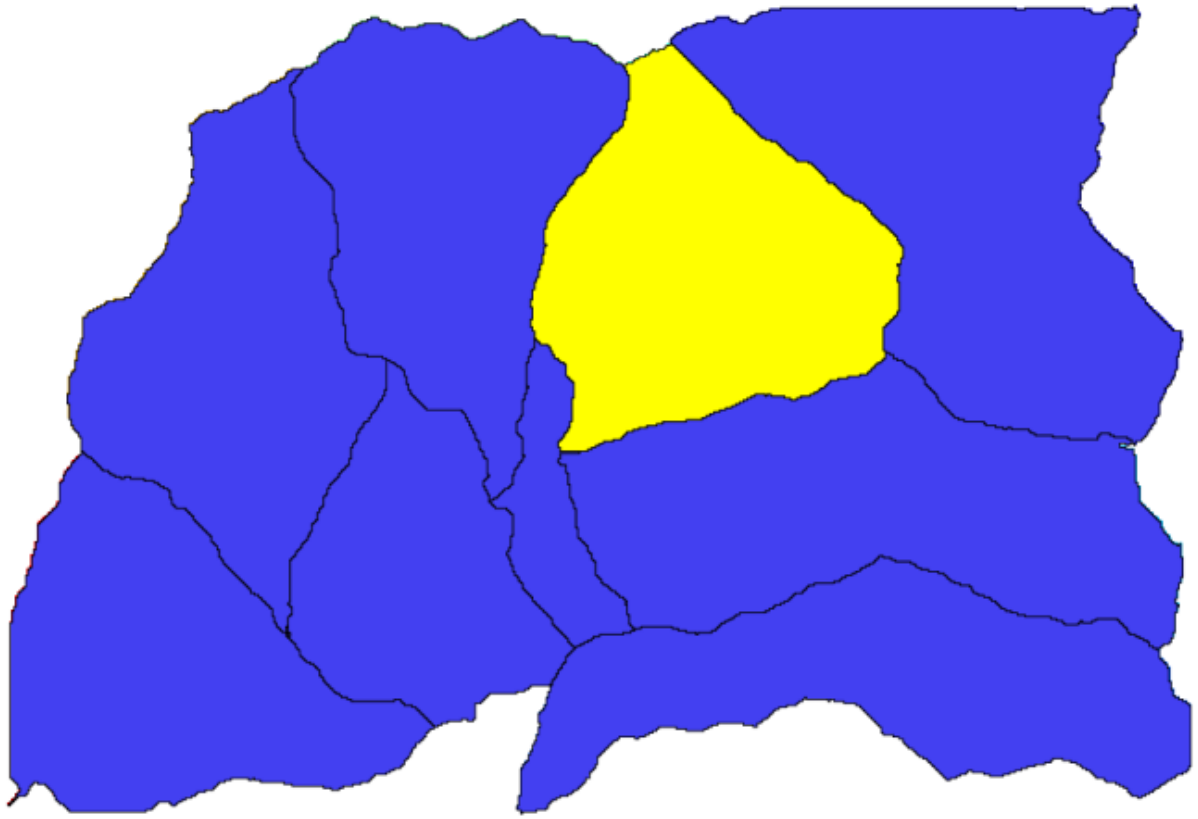

Vectorising Grid Classes
Parameters
Help
Grid
Watershed Basins
...
Class Selection
[1] all classes
Class Identifier
0
Vectorised class as...
[0] one single (multi-)polygon object
Polygons
[Save to temporary file]
...
Open output file after running algorithm
0%
OK
Close
Cancel



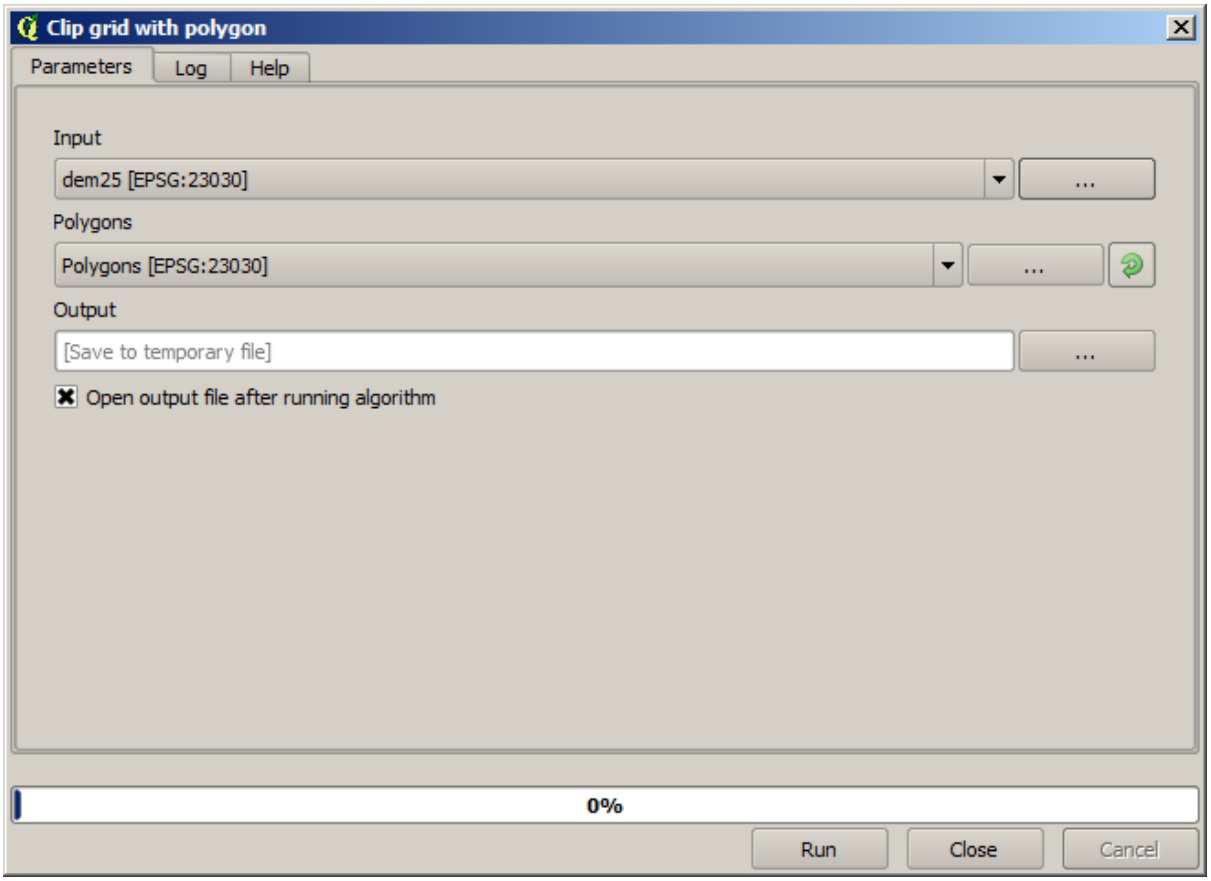
さて、第二次流域の一つで標高値についての統計を計算してみましょう。考え方は、ちょうどその第二次流域内だけの標高を表しているレイヤを得て、それをそれらの統計を計算するモジュールに渡すことです。

1. まず、第二次流域を表すポリゴンを使用して元の DEM をクリップします。 **ポリゴンでラスタをクリップ** アルゴリズムを使用します。単一の第二次流域ポリゴンを選択してからクリッピングアルゴリズムを呼び出すと、アルゴリズムが選択を認識しているため、DEM をそのポリゴンでカバーされる領域にクリップできます。

    1. ポリゴンを選びます

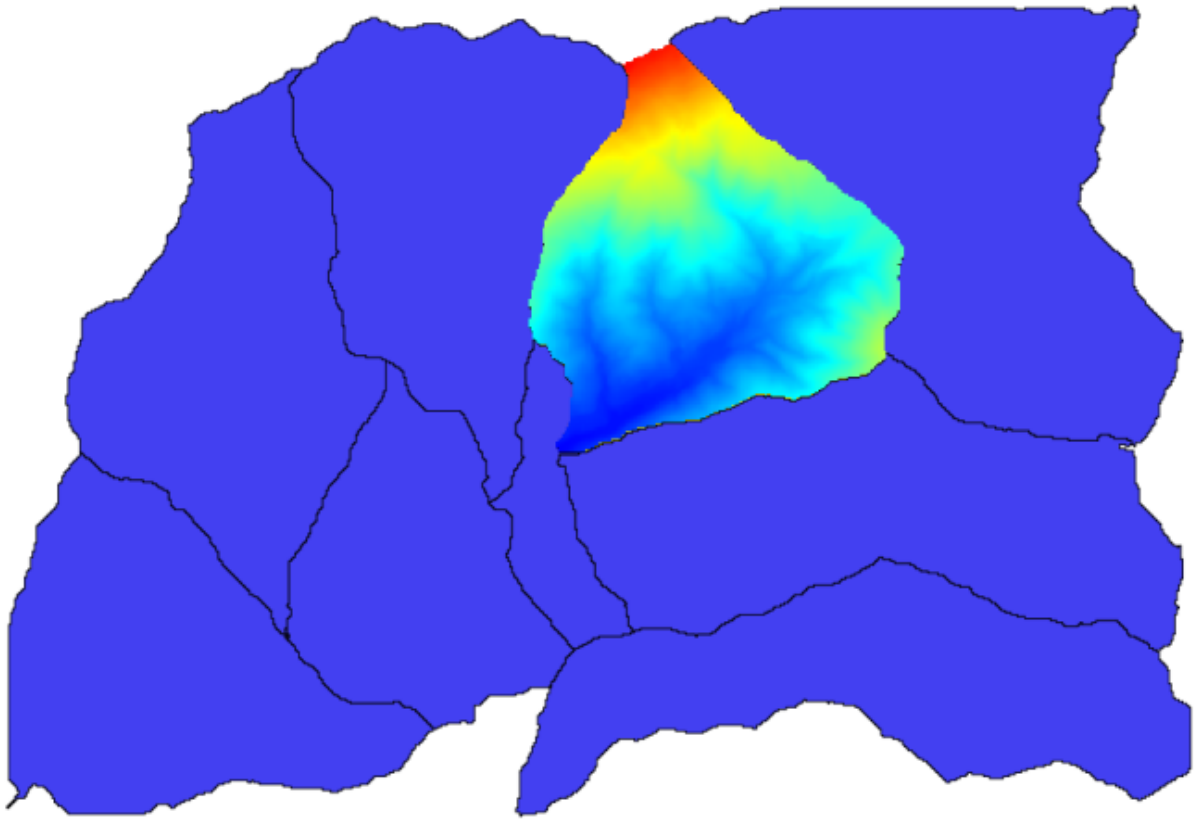


    2. 次のパラメータでクリッピングアルゴリズムを呼び出します：

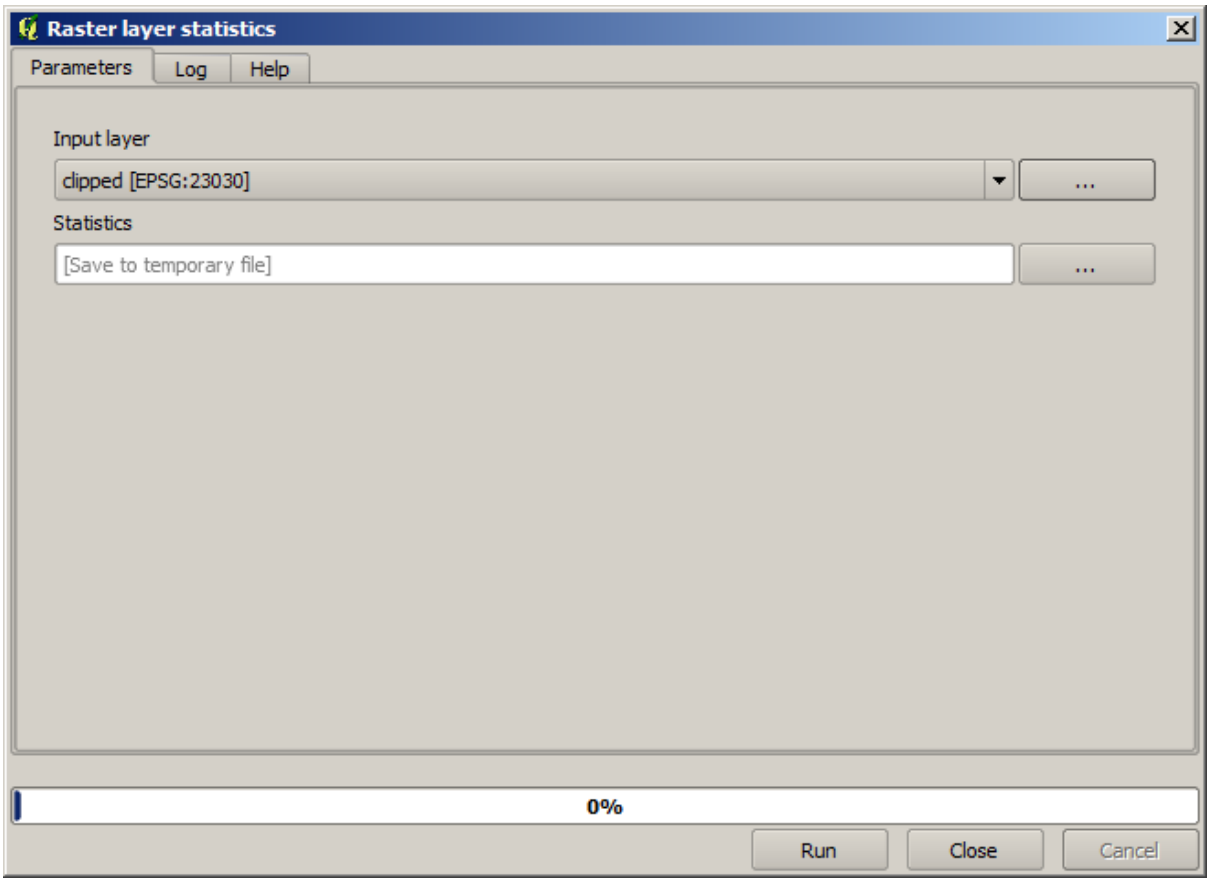


入力フィールドで選択された要素は、もちろん、クリップしたい DEM です。

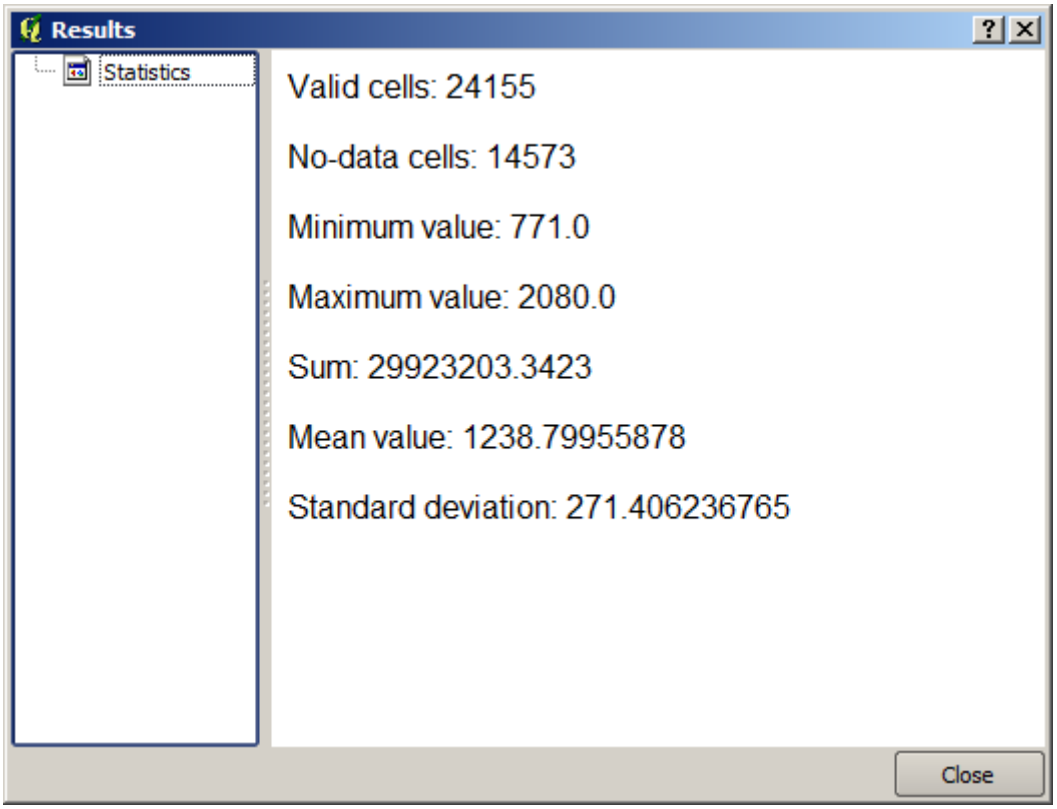
このようなものが得られます。



2. このレイヤは **ラスタレイヤの統計量** アルゴリズムで使う準備ができています。



結果の統計は以下のものです。



他のレッスンでは流域の計算手順および統計計算の両方を使用するでしょう。そして他の要素がそれらの両方を自動化しより効率的に作業するのにどのように役立ちうるかを見ます。

17.17 モデルデザイナーから始める

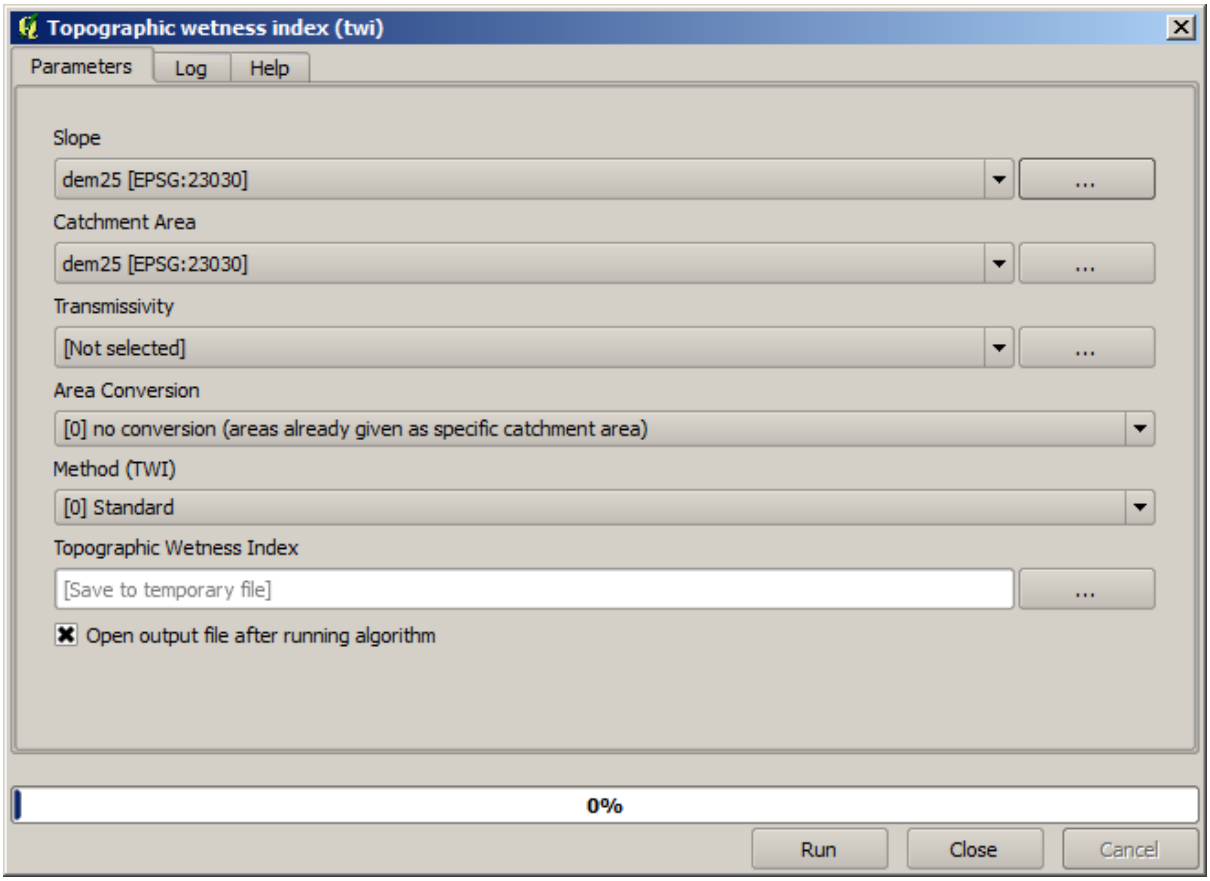
注釈

このレッスンでは、ワークフローを定義してアルゴリズムの連鎖を実行するために使用できる強力なコンポーネントである、モデルデザイナーを使います。

プロセシングツールによる通常のセッションは、単一のアルゴリズムを実行する以上のものを含んでいます。通常、結果を得るためには複数のアルゴリズムが実行され、そしていくつかのアルゴリズムの出力は、他のいくつかのアルゴリズムの入力として使われます。

モデルデザイナーを使うと、ワークフローを１つのモデルに入れることができ、一回の実行で必要なすべてのアルゴリズムが実行されるので、全体のプロセスが簡素化され自動化されます。

このレッスンの始めに、地形湿潤指数というパラメータを計算することにします。それを計算するアルゴリズムは *Topographic wetness index (twi)* と呼ばれています。



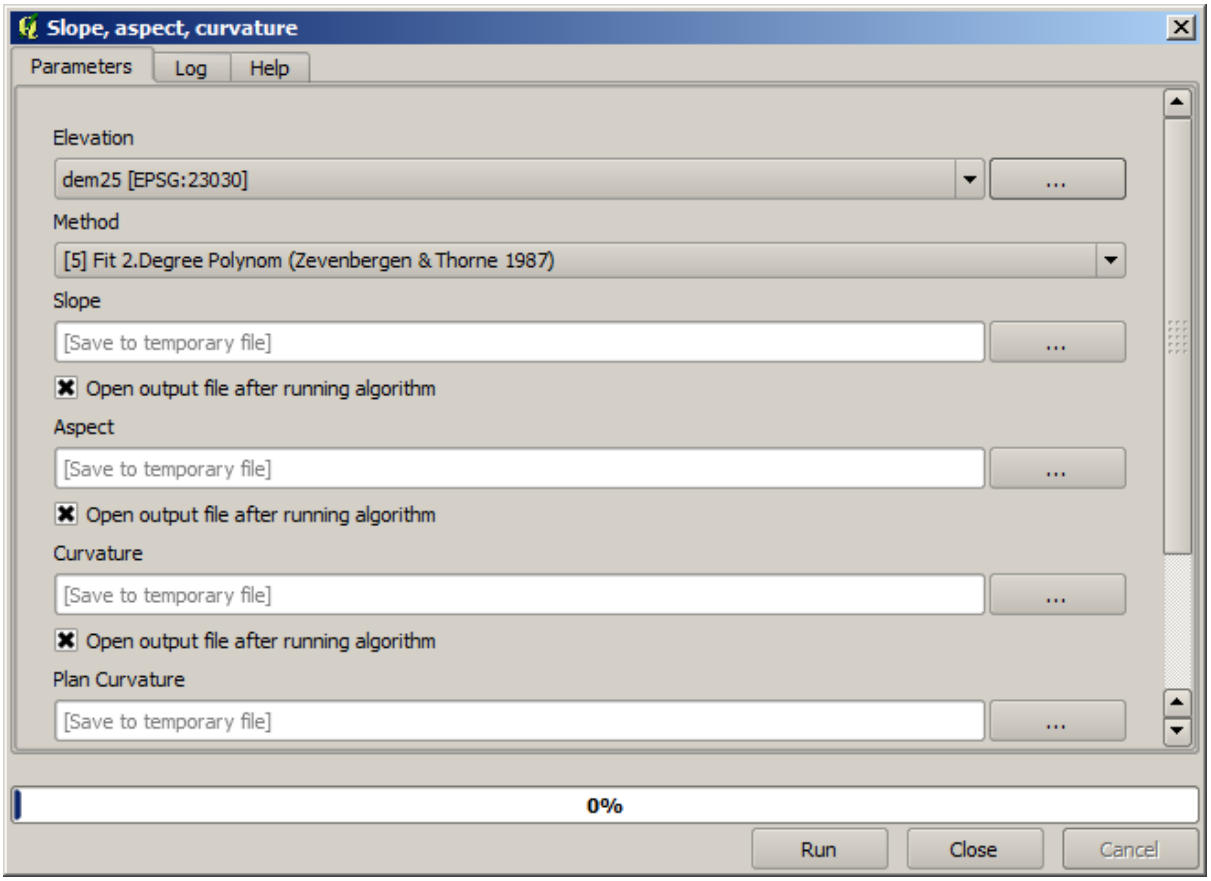
As you can see, there are two mandatory inputs: *Slope* and *Catchment Area*. There is also an optional input, but we will not be using it, so we can ignore it.

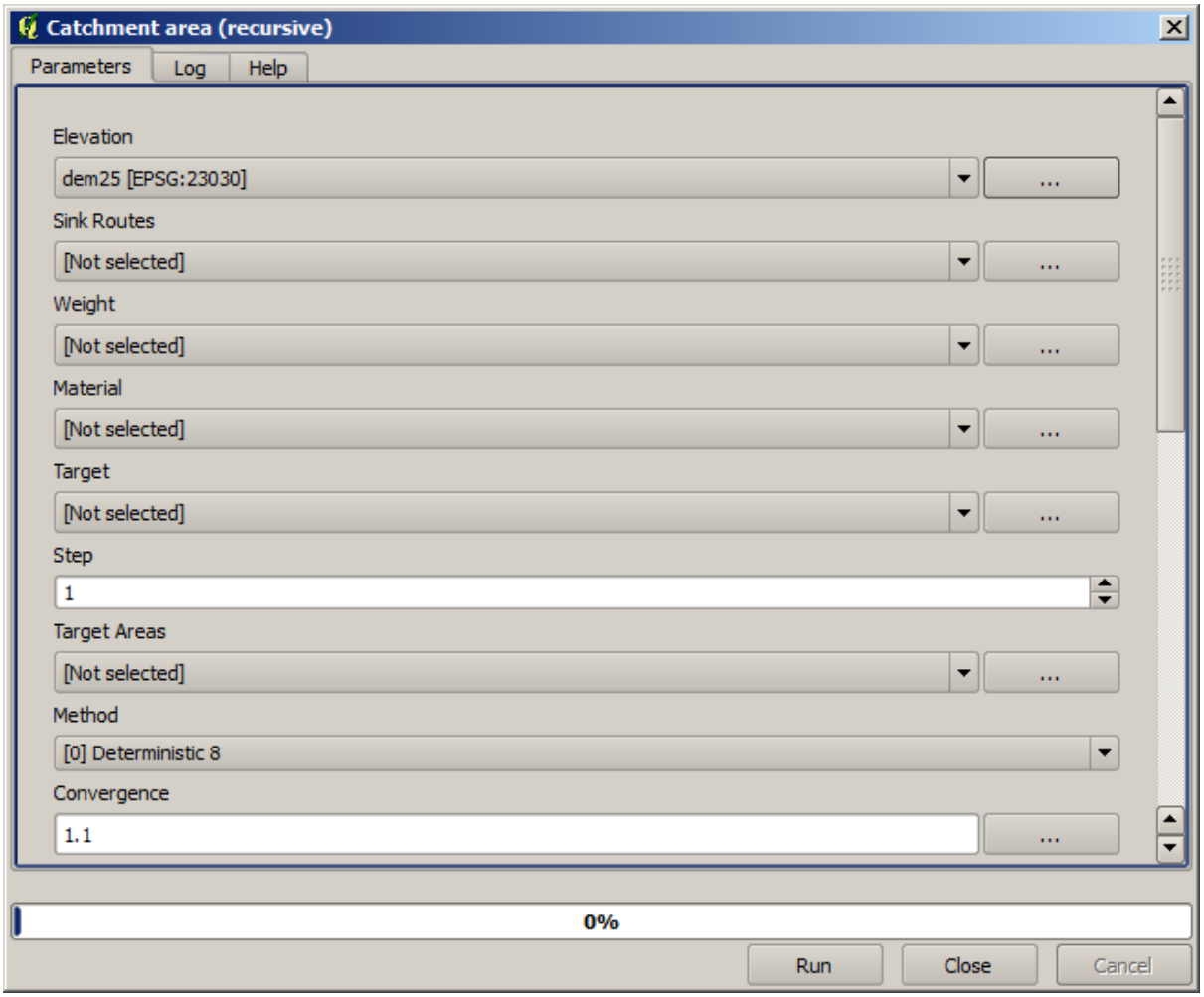
このレッスン用のデータには DEM しか入っていませんから、必要な入力は何もありません。しかし、傾きと集水域面積を計算するアルゴリズムをすでに見ていますので、その DEM からそれらの両方を計算する方法はわかっています。だから最初にこれらのレイヤを計算すると、それらを TWI アルゴリズムのために使用できます。

これが 2 つの中間レイヤを計算するために使用すべきパラメータのダイアログです。

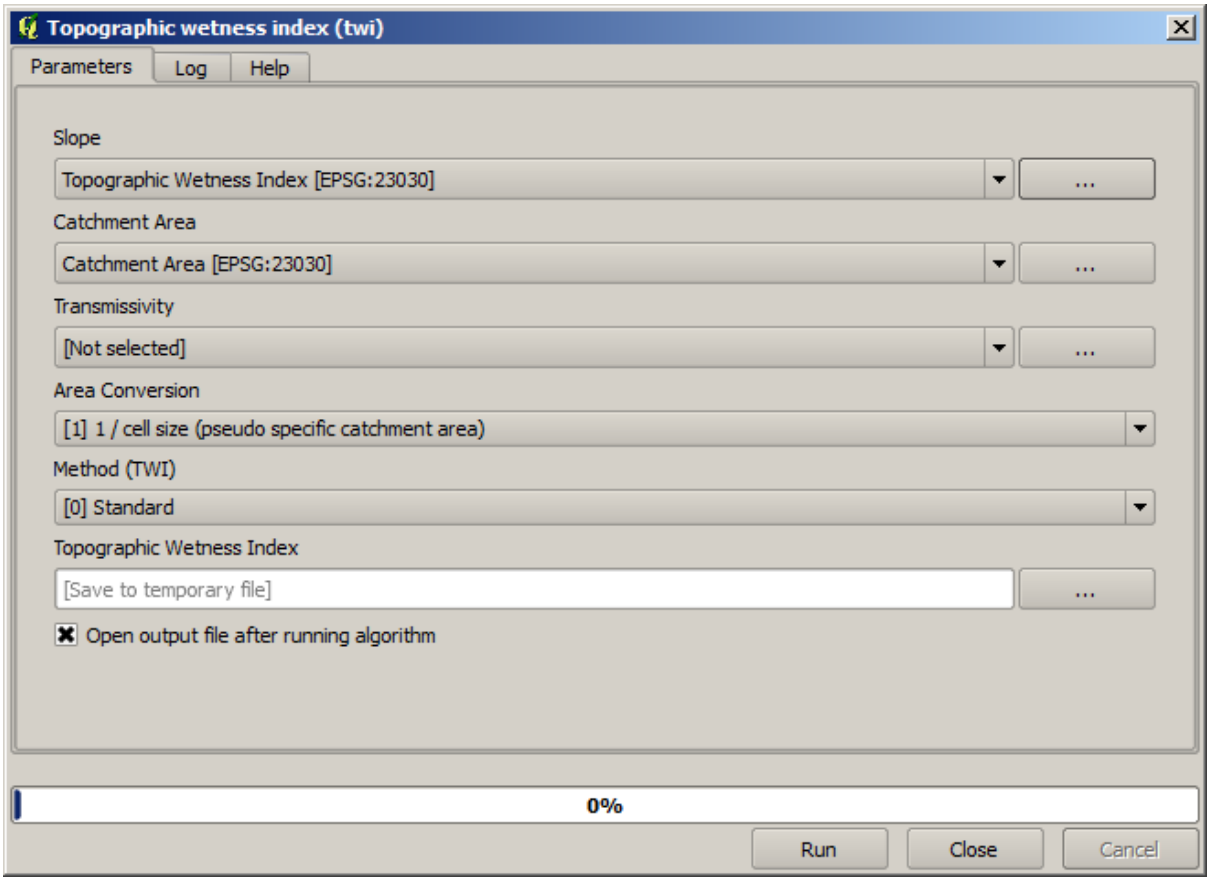
注釈

傾斜はラジアンではなく度で計算しなければなりません。

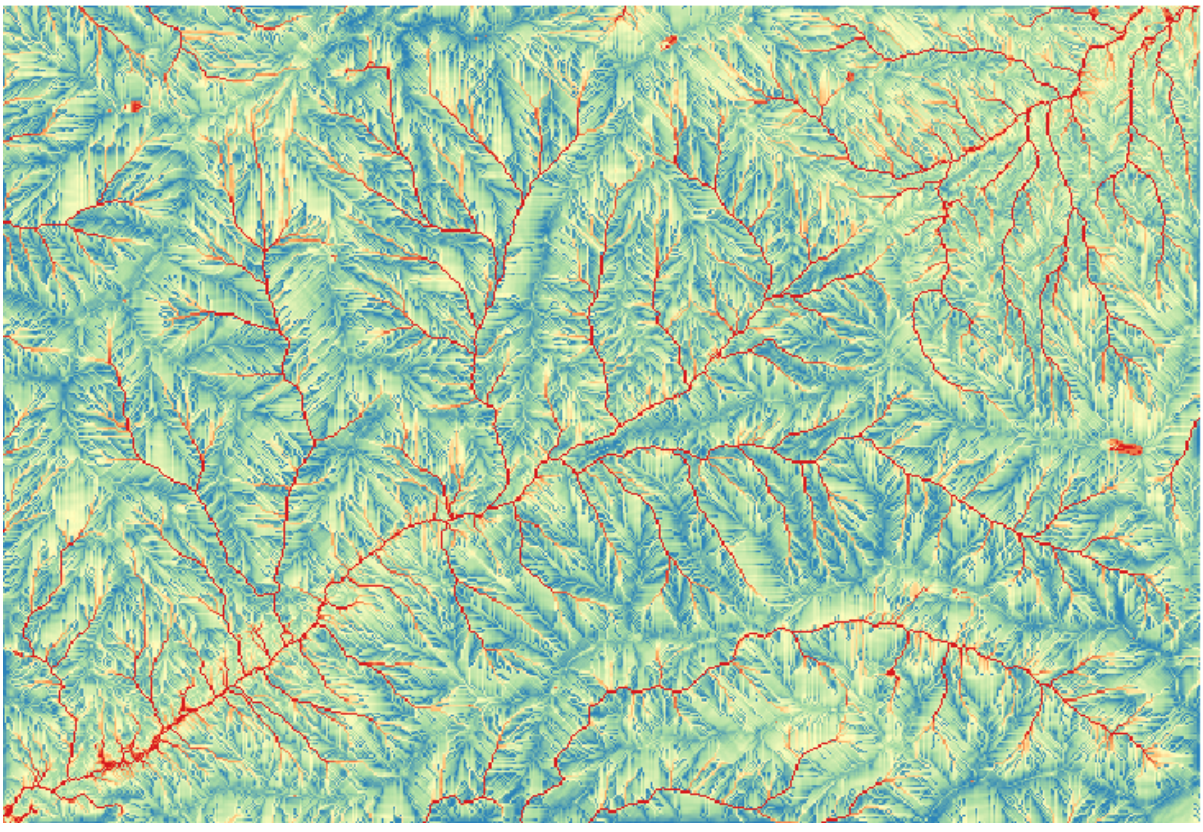

Slope, aspect, curvature
Parameters
Log
Help
Elevation
dem25 [EPSG:23030]
...
Method
[5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
Slope
[Save to temporary file]
...
Open output file after running algorithm
Aspect
[Save to temporary file]
...
Open output file after running algorithm
Curvature
[Save to temporary file]
...
Open output file after running algorithm
Plan Curvature
[Save to temporary file]
...
0%
Run
Close
Cancel



そしてこれは TWI アルゴリズムのパラメータダイアログボックスをどのように設定する必要があるかです。

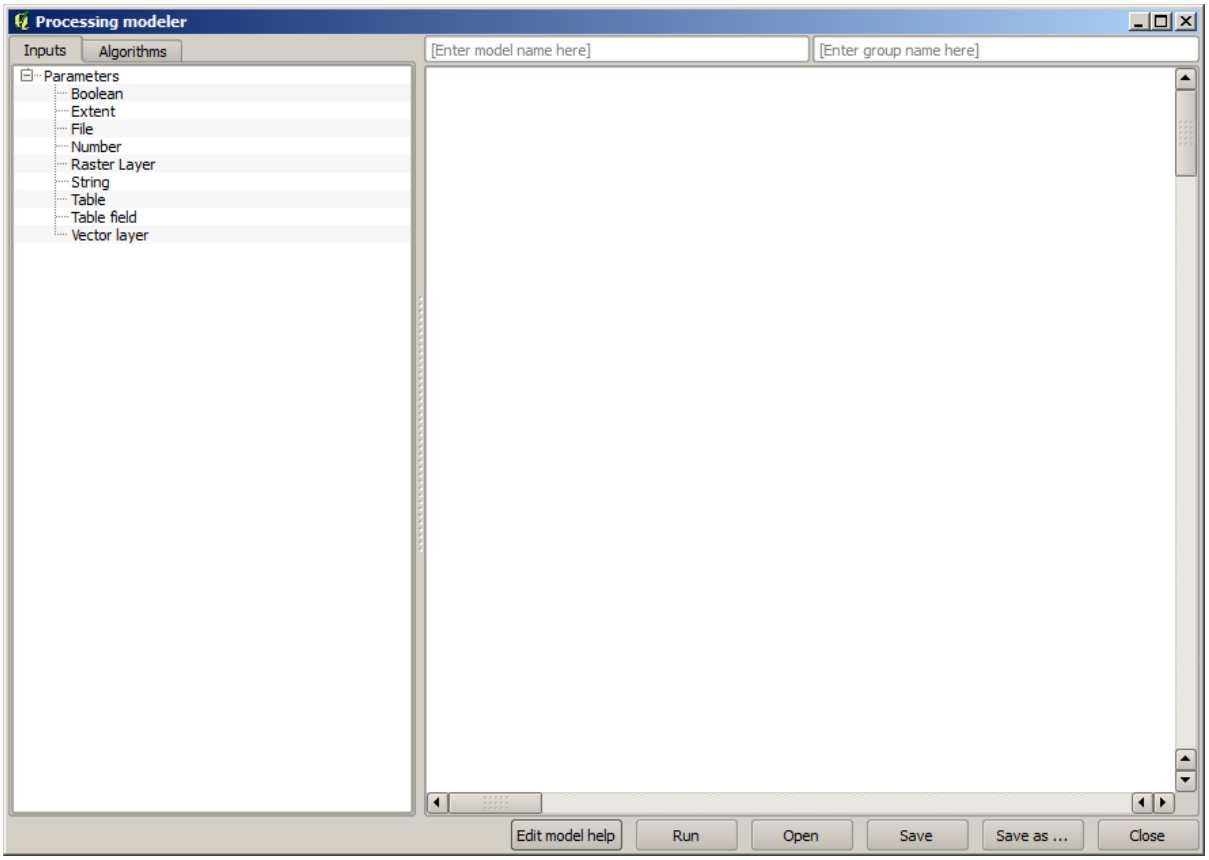


このような結果が得られます（レンダリングにはデフォルトの単バンド疑似カラー反転パレットが使われています）。提供された `twi.qml` スタイルを使用することができます。



ここでは、DEM から TWI を 1 ステップで計算するアルゴリズムを作成することを目指します。これにより、後で別の DEM から TWI レイヤを計算する必要が生じた場合、上記の 3 つのステップではなく、たった 1 つのステップで計算できるため、作業を軽減することができます。必要な処理はすべてツールボックスに用意されているので、あとはそれらを包むワークフローを定義するだけです。そこで登場するのがモデルデザイナーです。

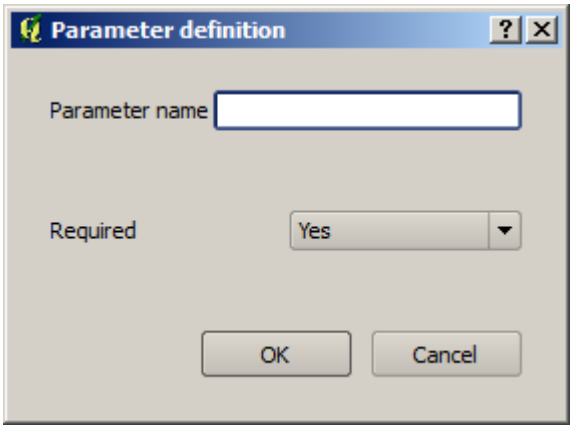
1. プロセシングメニューで、メニュー項目を選択することにより、モデラーを開きます。



   モデルを作成するには 2 つのことが必要です：必要な入力を設定することと、そのモデルに含まれるアルゴリズムを定義することです。この 2 つは、モデラーウィンドウの左側にある 2 つのタブ、 **入力** と **アルゴリズム** から要素を追加することで実現できます。

2. 入力から始めましょう。このケースでは追加することがあまりありません。DEM を持つラスタレイヤが必要なだけで、それが唯一の入力データとなります。

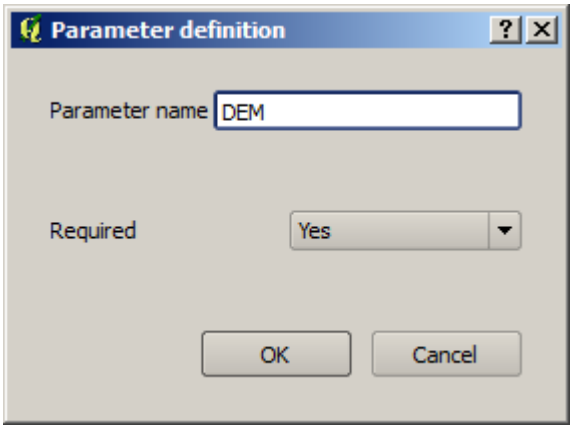
3. **ラスタレイヤ** 入力でダブルクリックすると次のダイアログが表示されます。



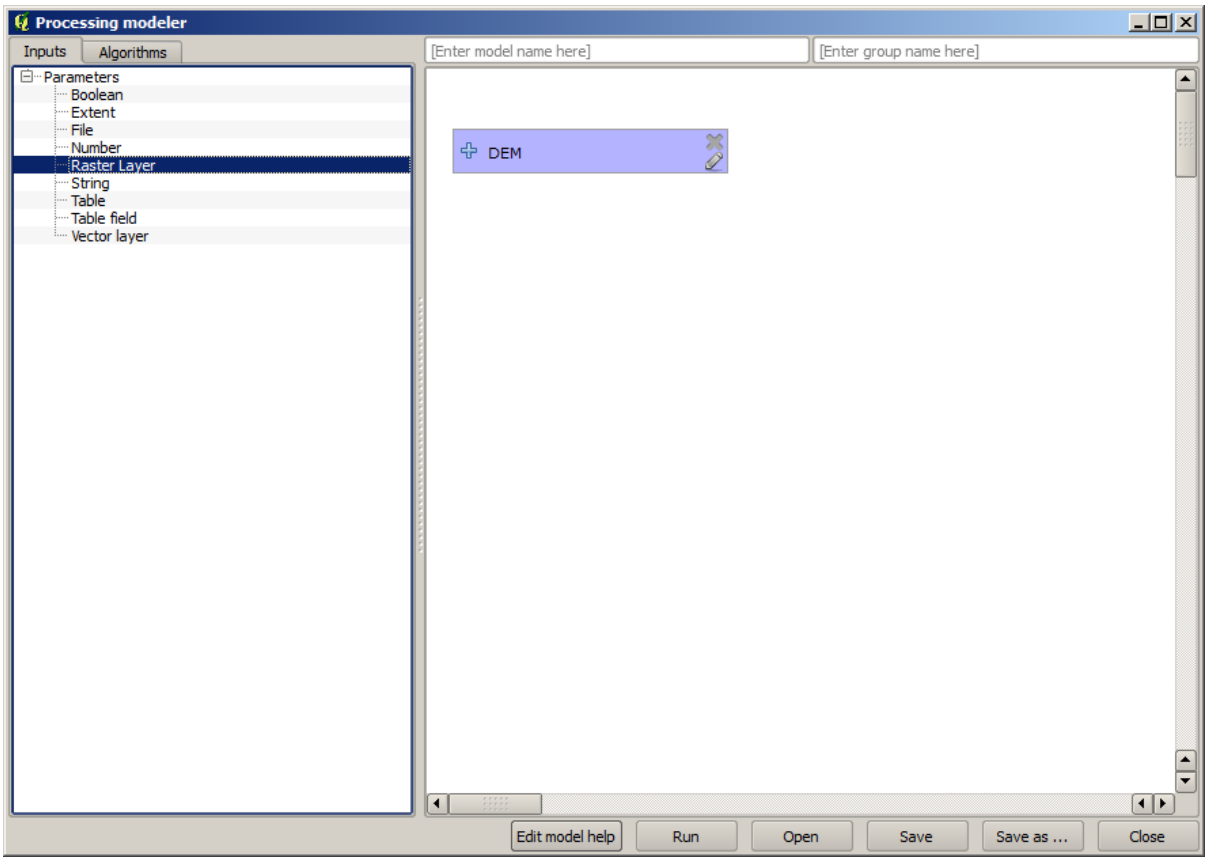
4. ここでは、必要な入力を定義する必要があります：

   1. このラスタレイヤは DEM であることが期待されるので、`DEM` と呼ぶことにします。これは、モデルを実行する際に、モデルのユーザーが見ることができる名前です。
   2. そのレイヤがないと動かないので、必須レイヤとして定義します。

   これがダイアログを設定する方法です

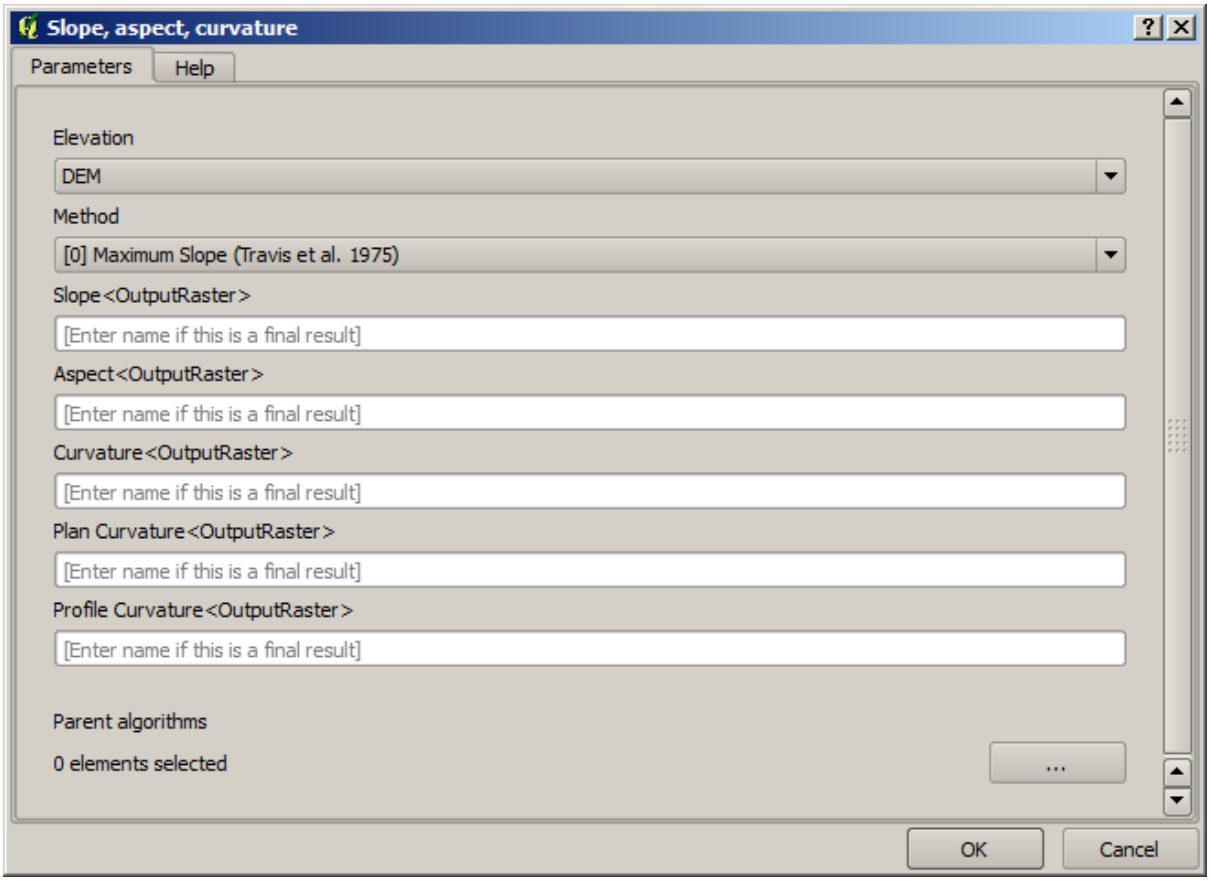


5. *OK* をクリックするとモデラーキャンバスに入力が現れます。



6. 次に **アルゴリズム** タブに移りましょう。

7. 最初に実行するアルゴリズムは、 *Slope, aspect, curvature* アルゴリズムです。アルゴリズムリストでこれを探し、ダブルクリックすると、以下のようなダイアログが表示されます。

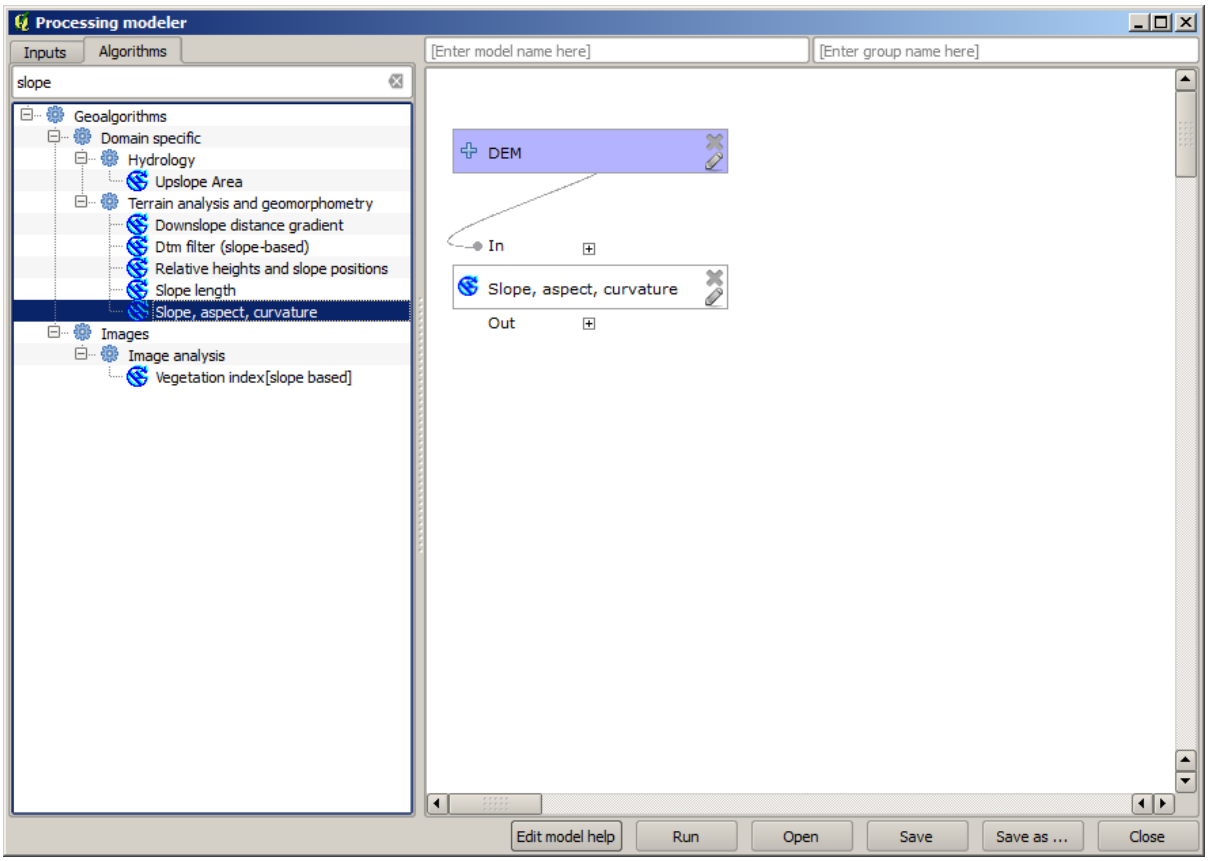


This dialog is very similar to the one that you can find when running the algorithm from the toolbox, but the element that you can use as parameter values are not taken from the current QGIS project, but from the model itself. That means that, in this case, we will not have all the raster layers of our project available for the *Elevation* field, but just the ones defined in our model. Since we have added just one single raster input named `DEM`, that will be the only raster layer that we will see in the list corresponding to the *Elevation* parameter.

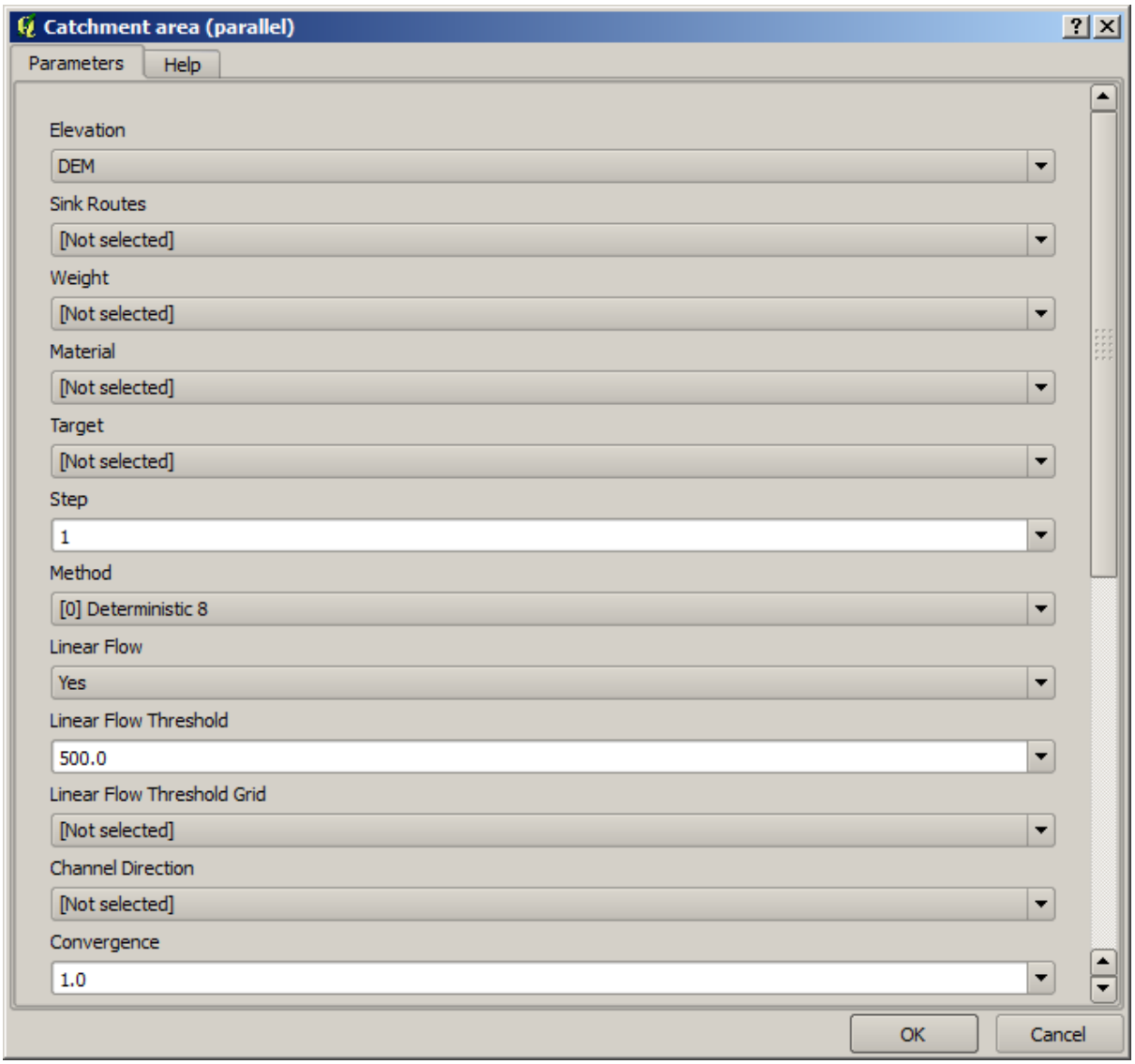
アルゴリズムにより生成された出力は、アルゴリズムがモデルの一部として使用される場合、少し異なって処理されます。各出力を保存するファイルパスを選択する代わりに、その出力が中間層であるか（そしてモデルの実行後に保存しないようにするか)、または最終出力であるかを指定する必要があります。この場合、このアルゴリズムによって生成されるすべてのレイヤは中間です。そのうちの 1 つ（傾斜レイヤ）のみを使用しますが、取得したい最終結果である TWI レイヤを計算するために必要なだけなので、それを保存する必要はありません。

最終的な結果ではないレイヤの場合、対応するフィールドは残しておかなければなりません。そうしないと、後でモデルを実行したときに表示されるパラメータダイアログボックスでレイヤを識別するために使用される名前を入力する必要があります。

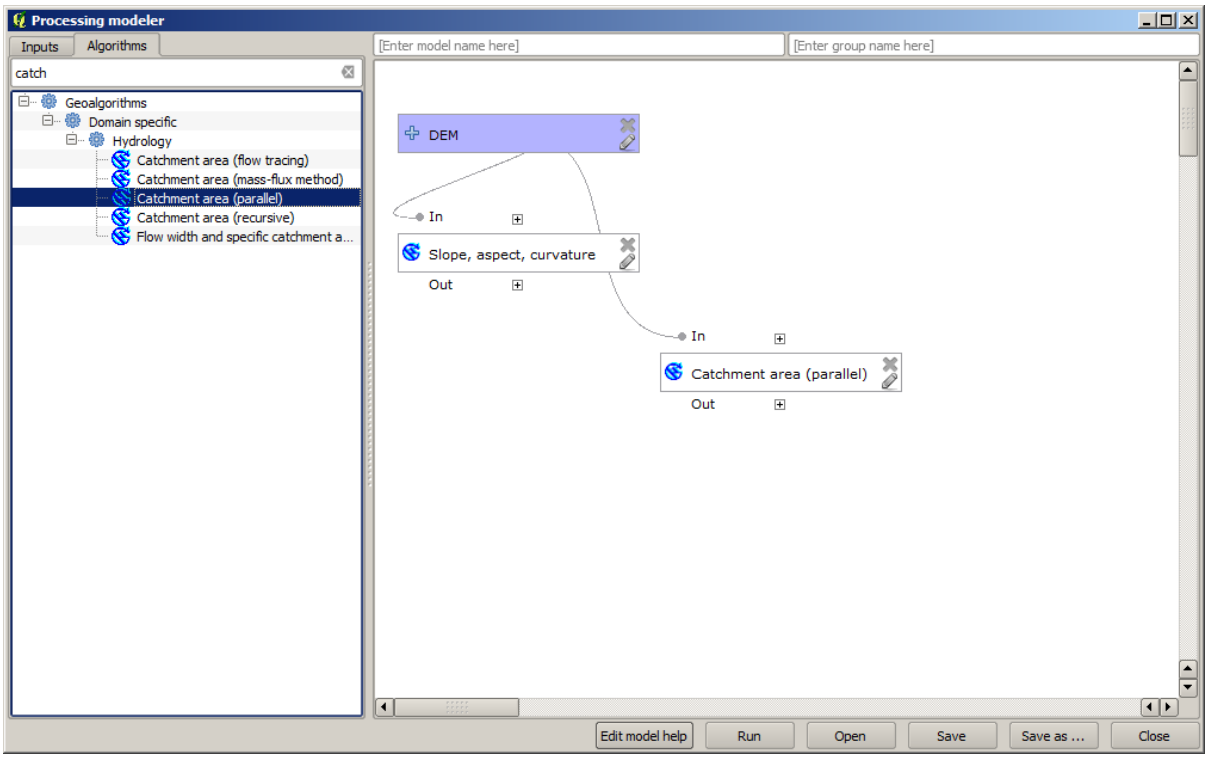
8. この最初のダイアログで選択することはあまりありません。なぜなら、このモデルには 1 つのレイヤ（作成した DEM 入力）しかありません。この場合、ダイアログのデフォルト設定が正しいので、 *OK* を押すだけでよいのです。これがモデラーキャンバスに表示されるものです。



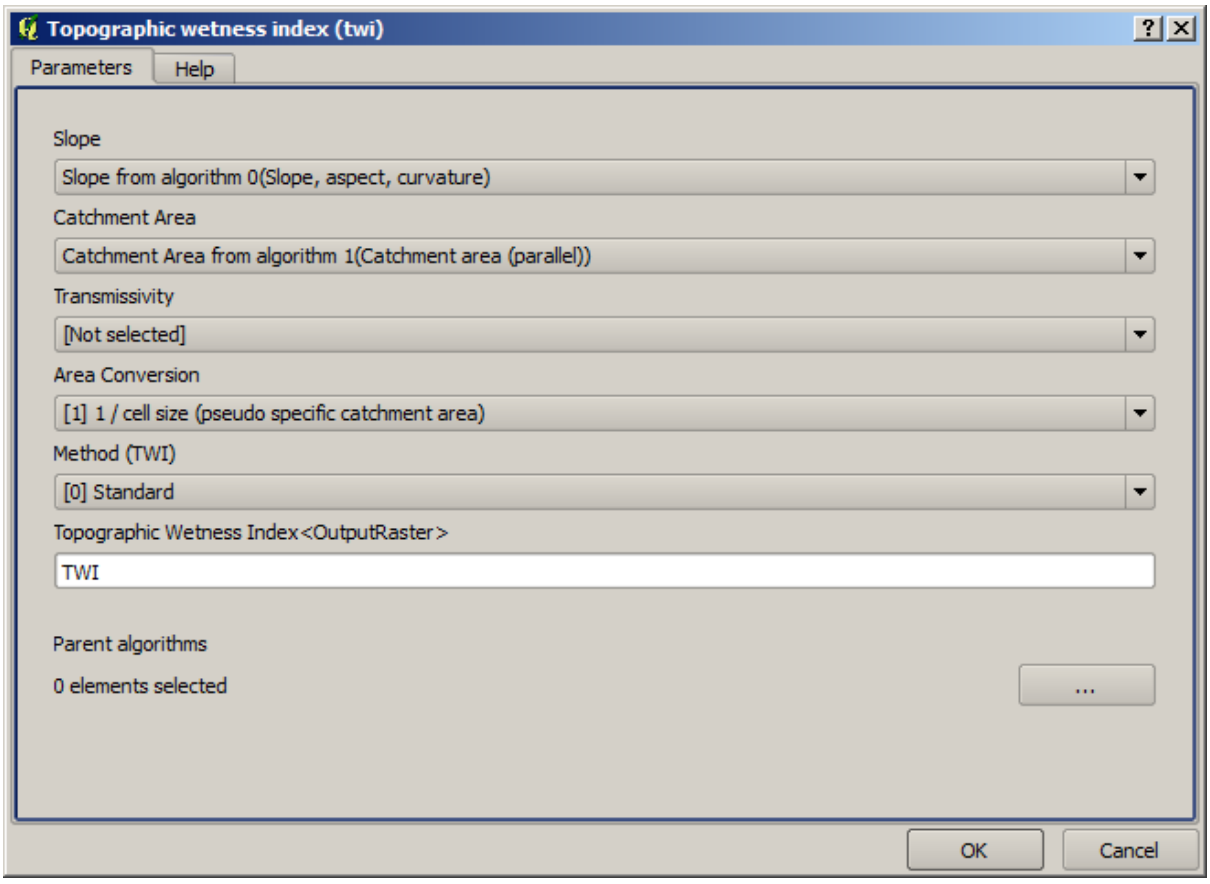
9. The second algorithm we have to add to our model is the catchment area algorithm. We will use the algorithm named *Catchment area (Paralell)*. We will use the DEM layer again as input, and none of the ouputs it produces are final, so here is how you have to fill the corresponding dialog.



これでモデルは次のようになります。



10. 最後に *Topographic wetness index* アルゴリズムを追加し、以下の構成とします。

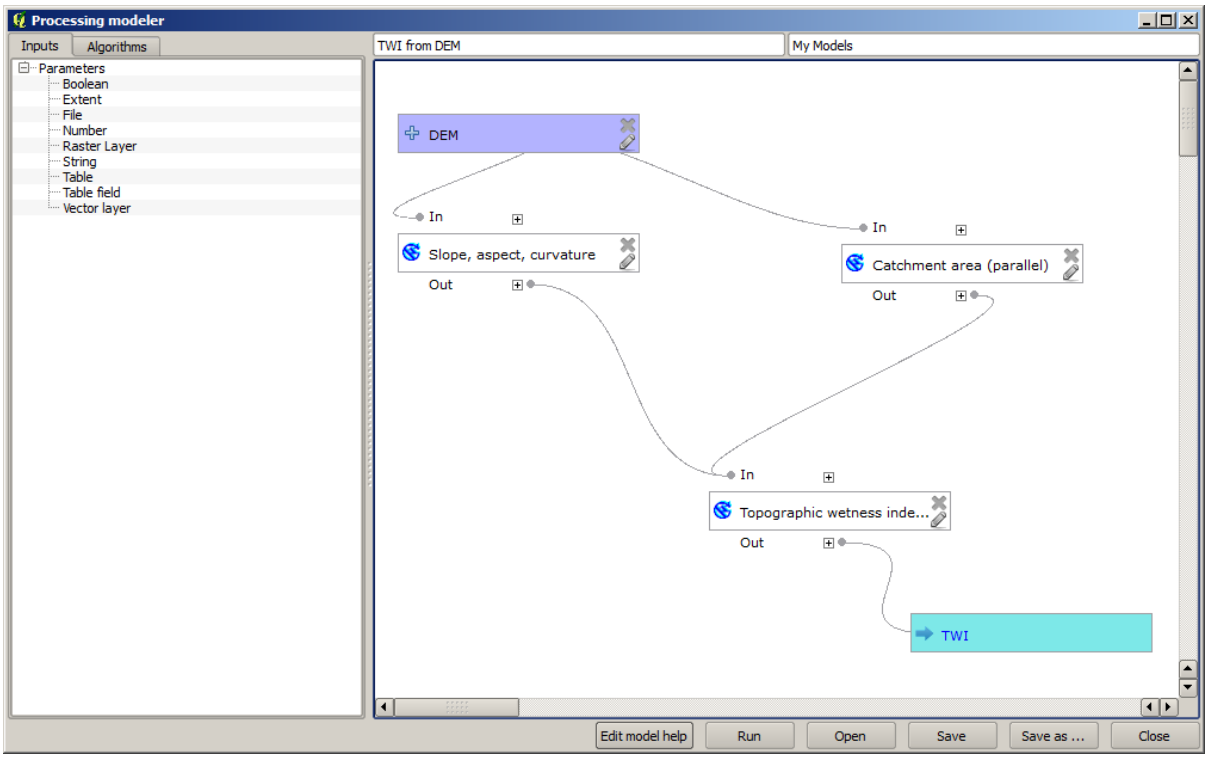


この場合、入力として DEM を使用しない代わりに、以前に追加アルゴリズムによって計算される傾

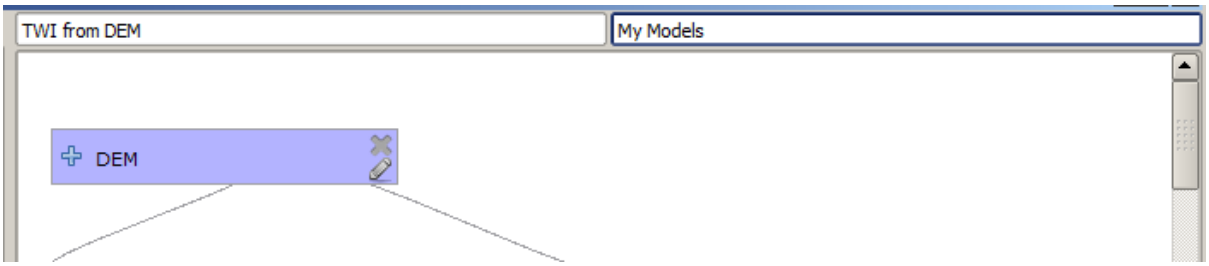
斜および集水域レイヤを使用します。新しいアルゴリズムを追加すると、それらが作り出す出力が他のアルゴリズムのために利用可能になり、それらを使用してアルゴリズムをリンクし、ワークフローを作成します。

11. この場合、出力される TWI レイヤは最終レイヤであるため、そのように表示する必要があります。対応するテキストボックスに、この出力に表示させたい名前を入力します。

    今、モデルが完成すると、それは次のようになります。

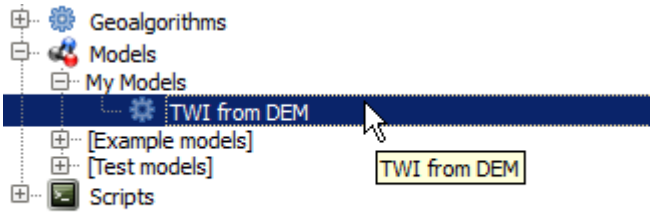


12. モデルウィンドウの上部に名前とグループ名を入力します。

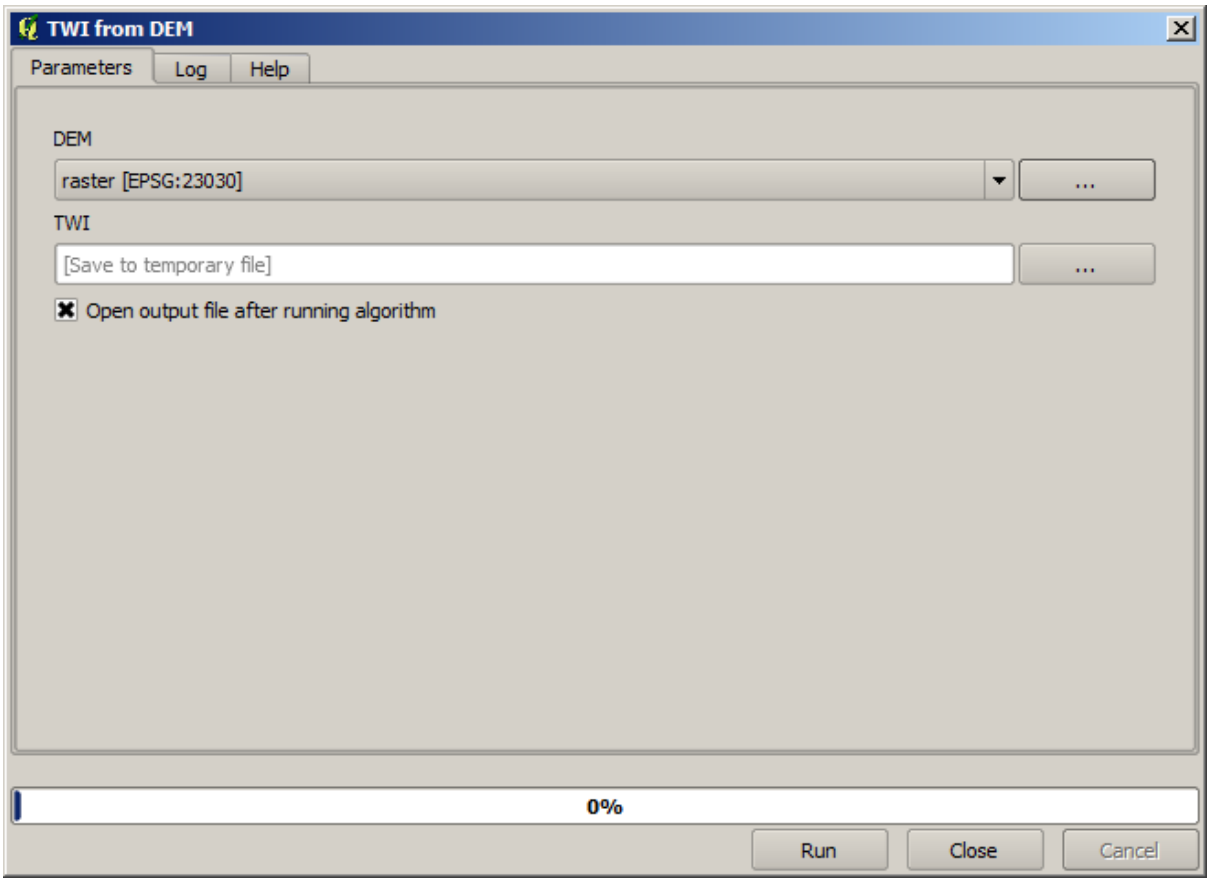


13. **モデルを保存** ボタンをクリックして保存します。好きな場所に保存して後で開くこともできますが、モデルフォルダ（ファイル保存ダイアログが表示されたときに表示されるフォルダです）に保存すると、あなたのモデルはツールボックスでも利用可能になります。ですから、そのフォルダにとどまって、お好みのファイル名でモデルを保存してください。

14. モデラーダイアログを閉じ、ツールボックスに移動します。**モデル** 要素の中にあなたのモデルが見つかります。



15. 通常のアルゴリズムと同じように、ダブルクリックで実行することができます。



ご覧のように、パラメータダイアログは、対応するアルゴリズムを追加するときに最終的な出力として設定した出力と共に、モデルに追加の入力を含みます。

16. 入力として DEM を使用して、それを実行すると、1 つのステップだけで TWI レイヤが得られるでしょう。

17.18 より複雑なモデル

注釈

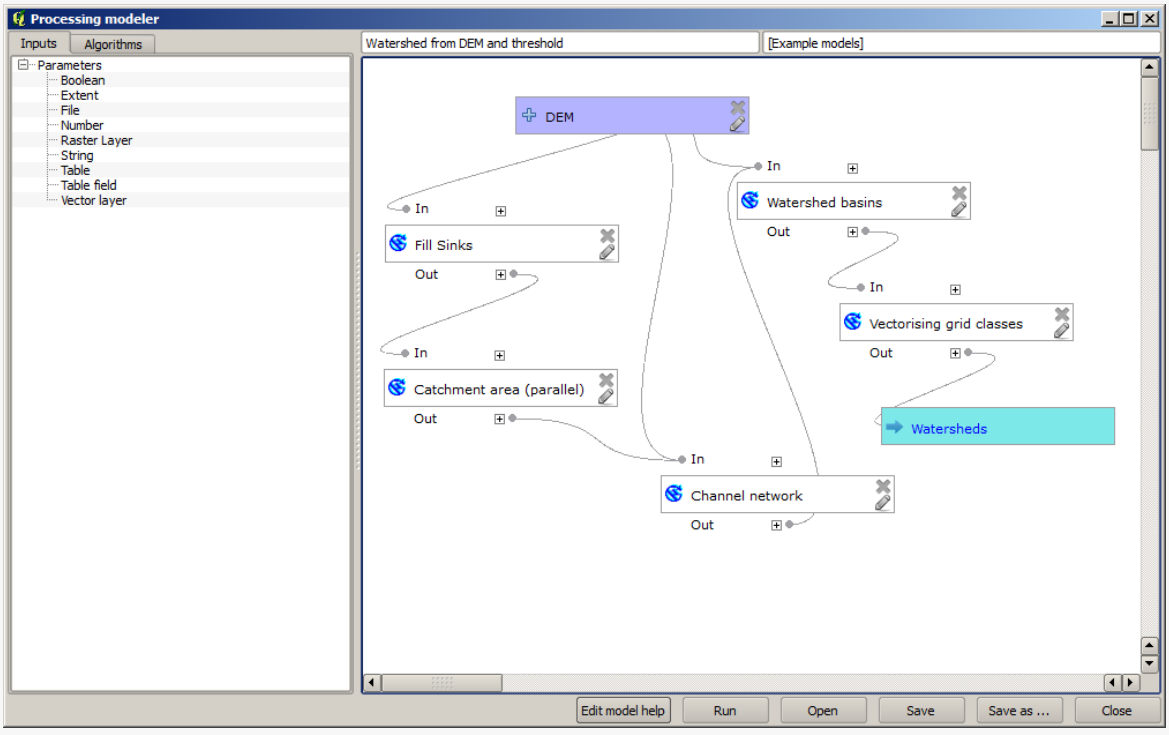
このレッスンでは、モデルデザイナーでより複雑なモデルを扱います。

前章で作成した最初のモデルは、1 つの入力と 3 つのアルゴリズムだけという、非常にシンプルなものでした。異なるタイプの入力とより多くのステップを含んだ、より複雑なモデルを作成することができます。この章では、DEM としきい値に基づいて流域のベクタレイヤを作成するモデルを扱います。このモデルは、異なる閾値に対応する複数のベクタレイヤを計算するのに非常に便利で、各ステップを毎回繰り返す必要がありません。

このレッスンでは、モデルの作成方法についての説明はありません。必要な手順はすでに知っているし（**前のレッスン** で見ています）、モデラーについての基本的な考え方も知っているので、自分でやってみるのがいいでしょう。数分かけてモデルを作ってみてください。間違うことを心配する必要はありません。思い出してください：まず入力を追加し、次にそれを使ってワークフローを作成するアルゴリズムを追加します。

注釈

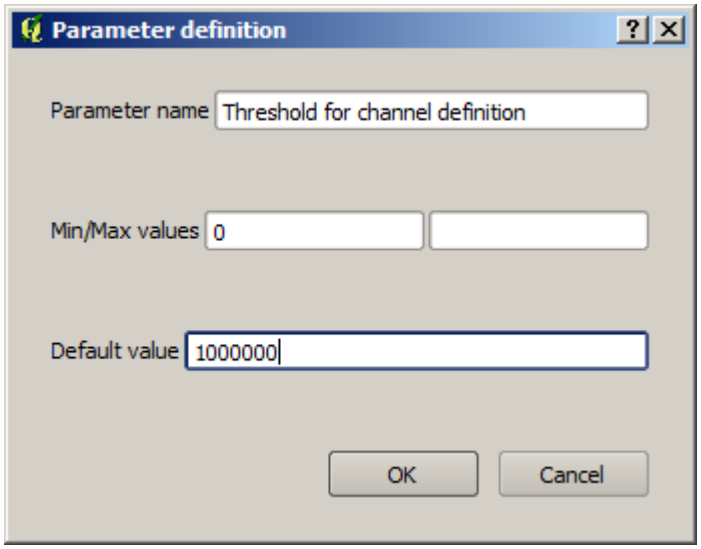
完全なモデルをご自分で作れず、いくらかの助けを必要とする場合、このレッスンに対応するデータフォルダにその「ほとんど」完成版が含まれています。モデラーを開き、データフォルダにあるモデルファイルを開きます。このようなものが表示されるはずです。



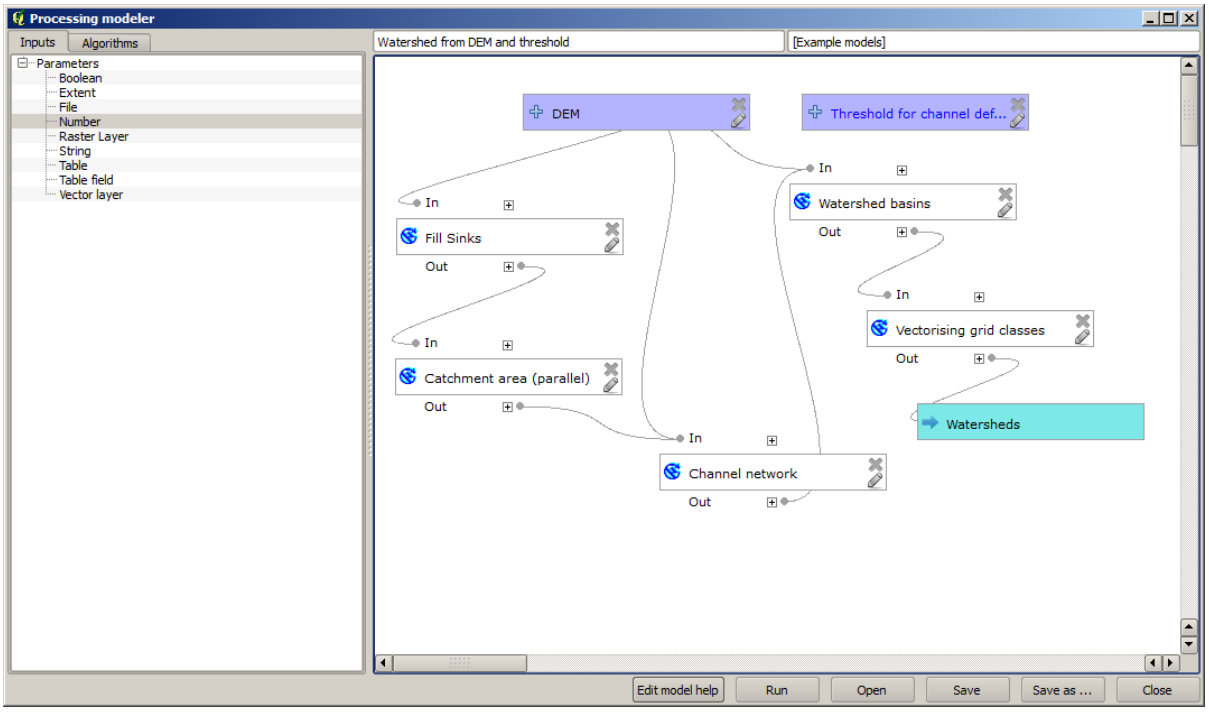
このモデルは、計算を完了するために必要なすべてのステップを含んでいますが、入力は DEM の 1 つだけです。つまり、水路決定のしきい値が固定値であるため、このモデルはまだあまり役に立ちません。し

かし、このモデルは編集することができ、これからそれをするので、問題ありません。

1. まず、数値入力を追加してみましょう。これはユーザーに数値入力を求めるもので、このモデルに含まれるアルゴリズムのいずれかにそのような値が必要な場合に使用することができます。
2. **入力** ツリーの **数値** エントリーをクリックすると、対応するダイアログが表示されます。
3. それに次の値を入力します。
   - **パラメータ名**: 水路決定のしきい値
   - **デフォルト値**: `1,000,000`



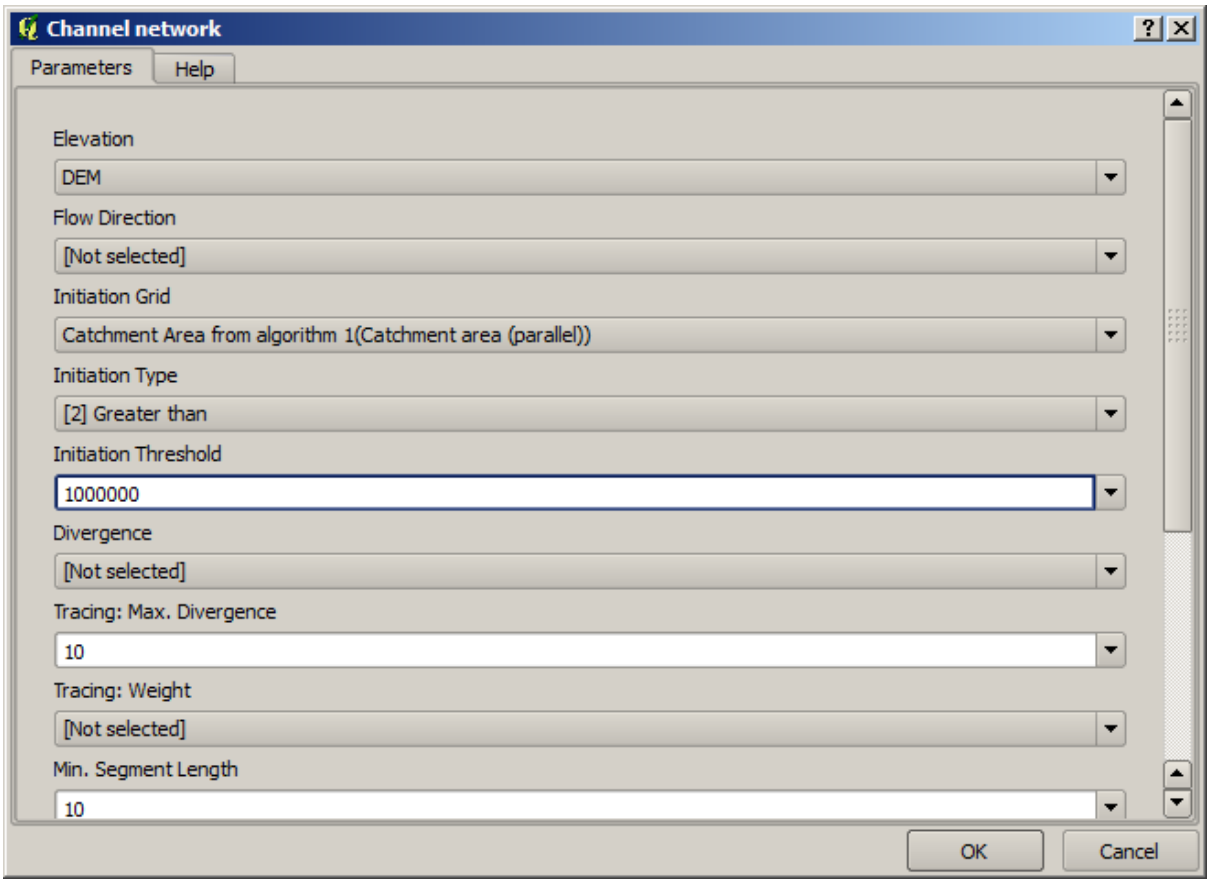
これでモデルは次のようになります。



The input that we have just added is not used, so the model hasn't actually changed. We have to link that

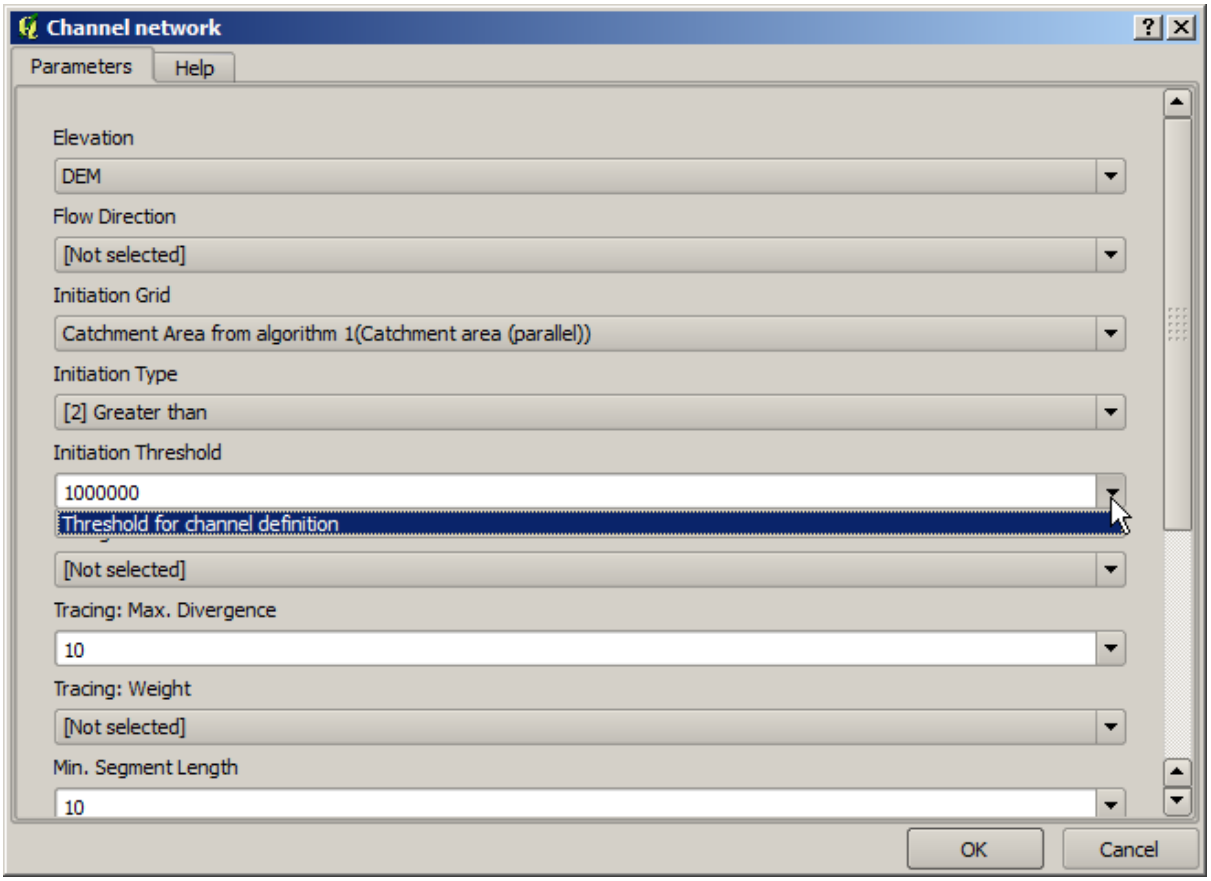
input to the algorithm that uses it, in this case the *Channel network* one. To edit an algorithm that already exists in the modeler, just click on the pen icon on the corresponding box in the canvas.

4. *Channel network* アルゴリズムをクリックすると、次のようになります。



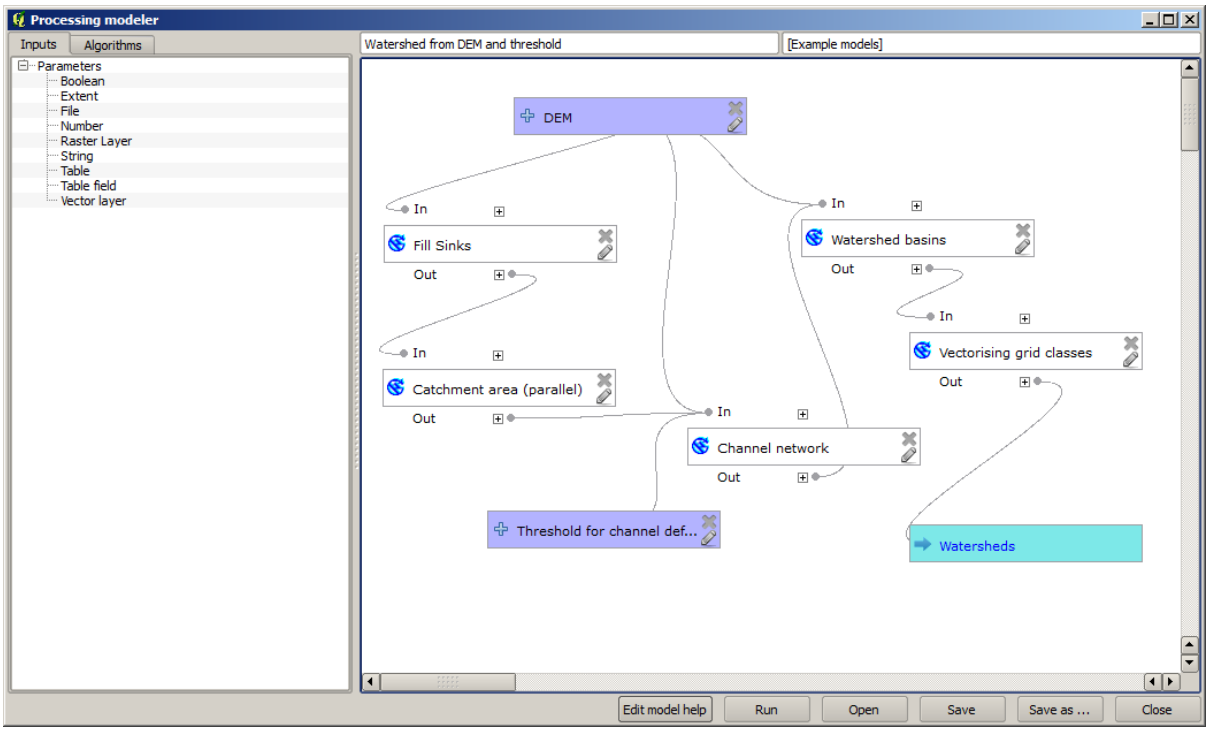
ダイアログには、アルゴリズムが使用する現在の値が入力されています。**開始しきい値** パラメータが 1,000,000 という固定値を持っていることがわかります（これはアルゴリズムのデフォルト値でもありますが、他のどんな値でも入れることができます）. しかし、このパラメータは一般的なテキストボックスではなく、ドロップダウンメニューに入力されていることにお気づきでしょうか？

5. しきい値パラメータメニューを展開すると、このように表示されます。



追加した入力がそこにあり、それを選択できます。モデルでアルゴリズムが数値を必要とするたびに、それをハードコーディングしそれを直接入力することも、使用可能な入力と値のいずれかを使用することもできます（いくつかのアルゴリズムは単一の数値を生成することを思い出してください。これについての詳細はすぐに見るでしょう）。文字列パラメーターの場合も文字列の入力が表示され、それらのいずれかを選択することも、所望の固定値を入力することもできます。

6. **開始しきい値** パラメータで、水路決定のしきい値 の入力を選択します。
7. *OK* をクリックすると、モデルに変更が適用されます。これで、モデルのデザインはこのようになります。



8. これでモデルは完成です。これまでのレッスンで使用した DEM を使用し、しきい値を変えて実行してみてください。

   以下は、異なる値で得られた結果のサンプルです。デフォルトの値の結果と比較できます。これは、hydrological analysis lesson で得られたものです。

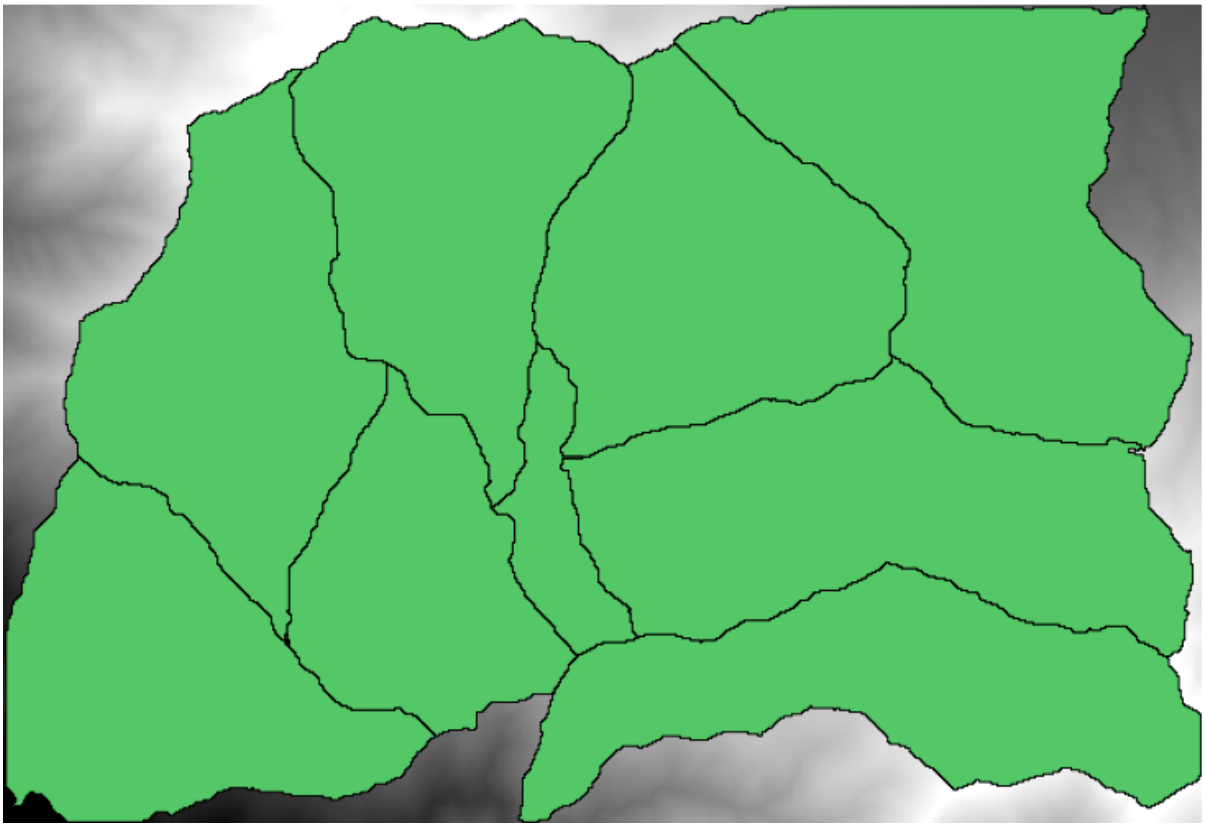


図 17.1: しきい値= 100,000

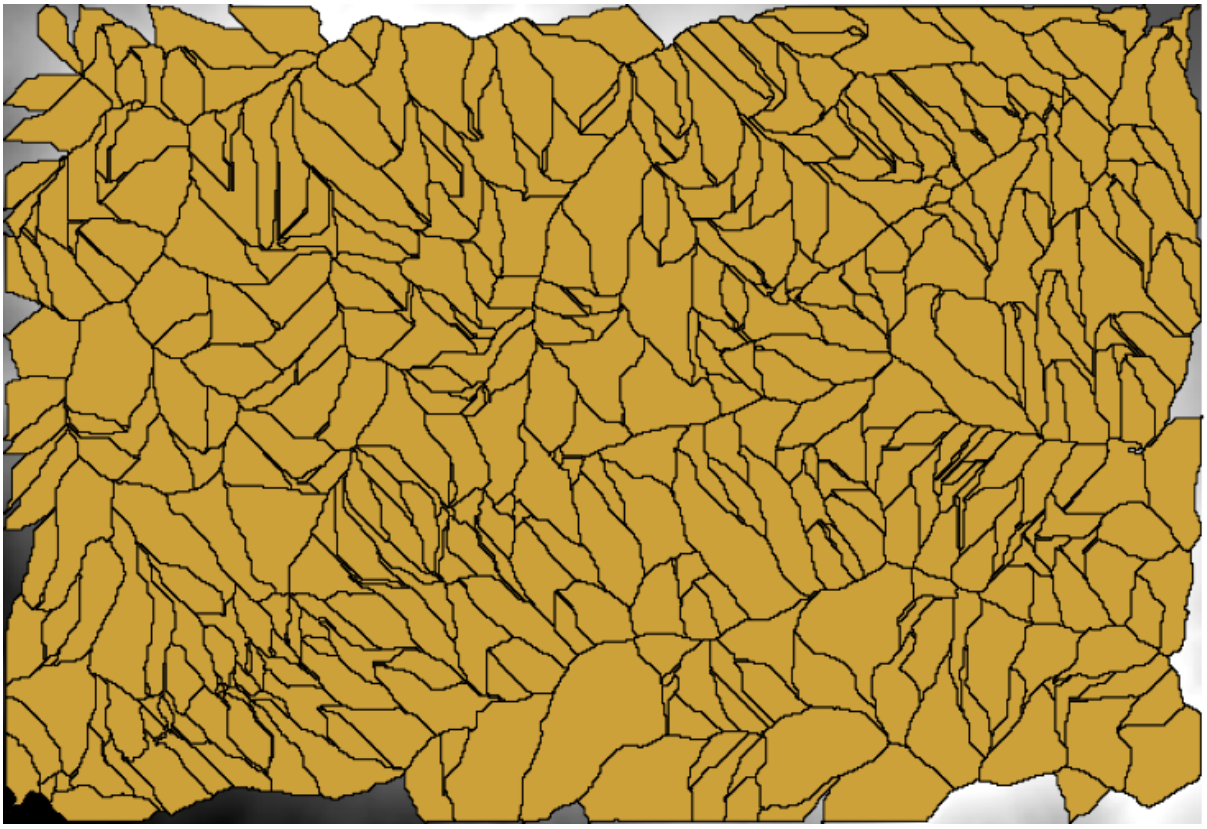


図 17.2: しきい値= 1,0000,000

17.19 モデラーでの数値計算

警告

注意してください。この章はよく検証されていないため、何か問題を見つけたらご報告ください；画像は欠けています

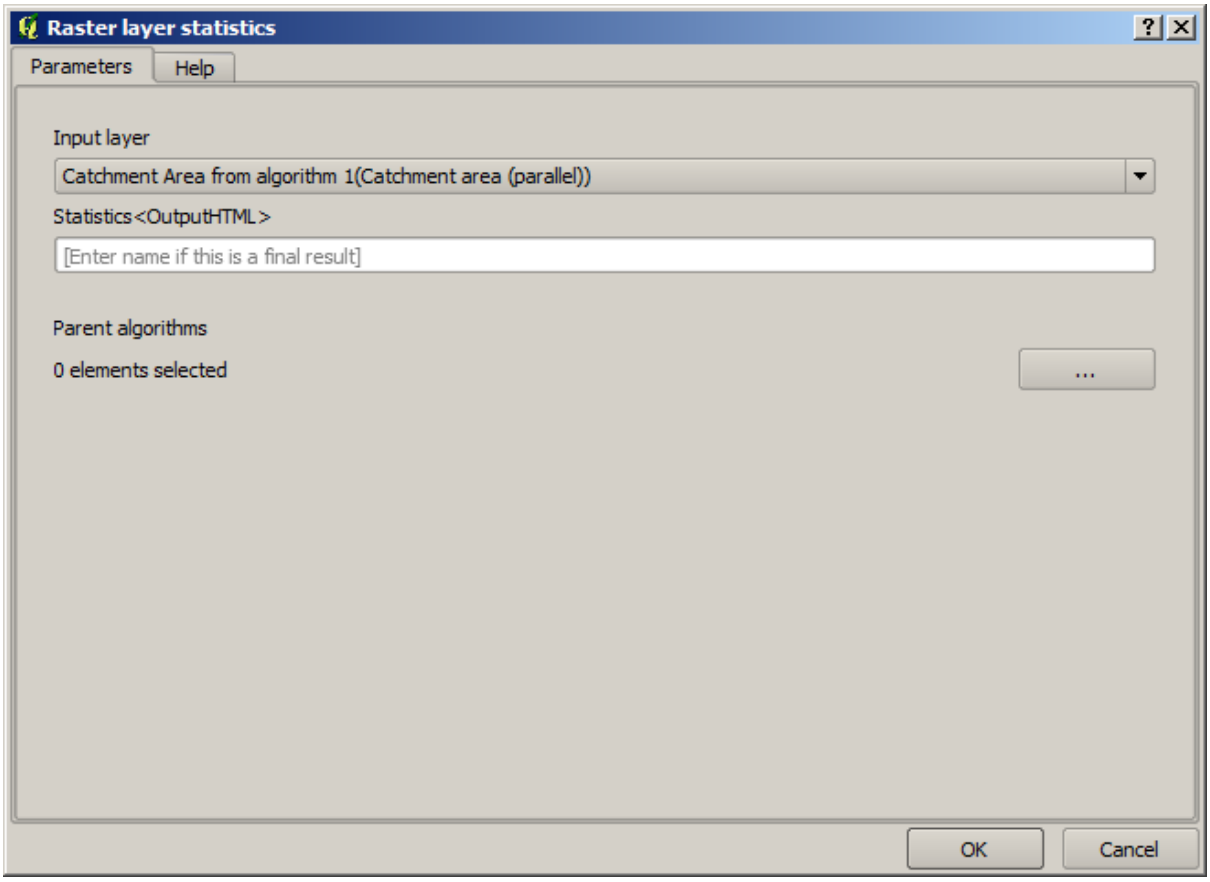
注釈

このレッスンでは、モデラーの数値出力を使用する方法について説明します

このレッスンでは、最後の章（開始前にモデラーで開く）で作成した水文モデルを変更して、有効なしきい値の計算を自動化でき、ユーザーに入力を依頼する必要がないようにしていきます。その値はしきい値ラスタレイヤ内の変数を指しているので、いくつかの単純な統計分析に基づいて、そのレイヤからそれを抽出します。

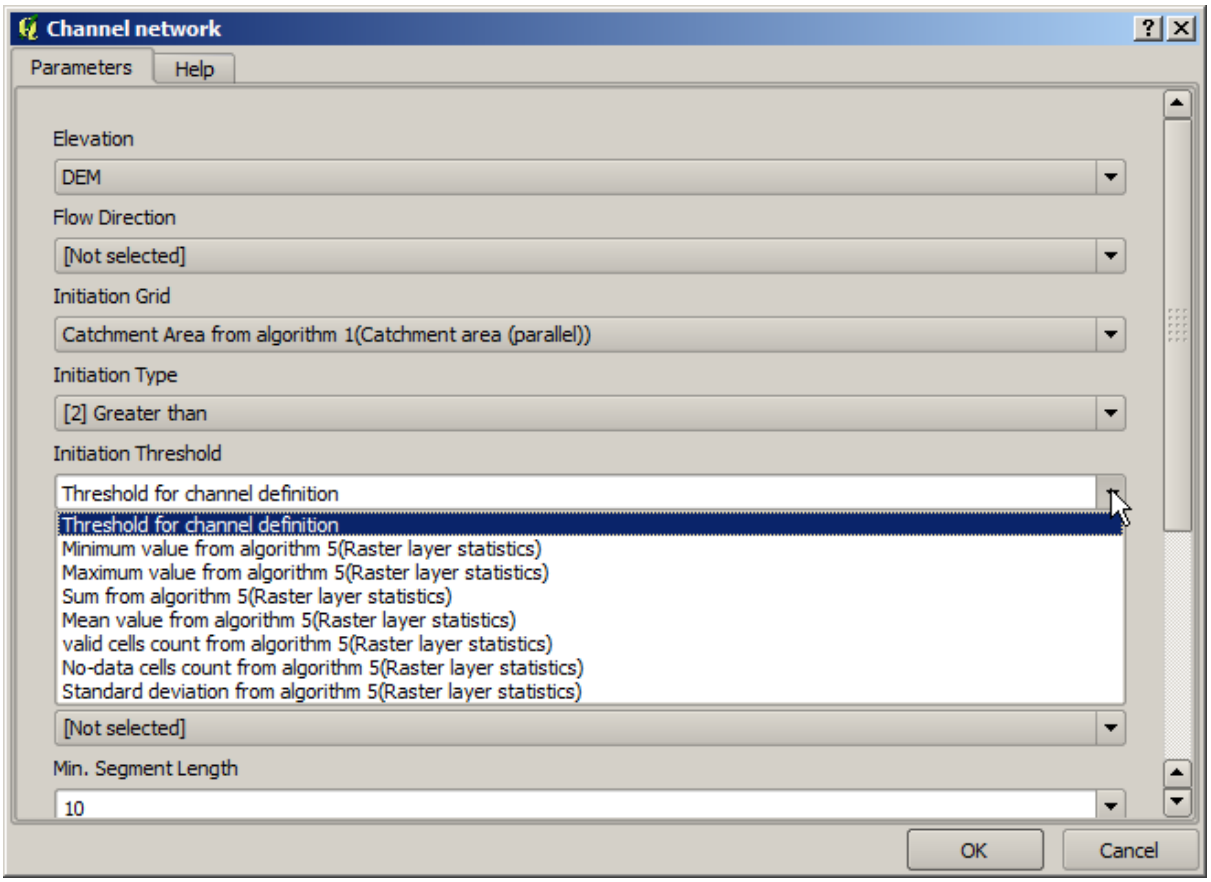
前述のモデルから開始し、以下の修正を行いましょう。

First, calculate statistics of the flow accumulation layer using the *Raster layer statistics* algorithm.



これによって統計値のセットが生成され、他のアルゴリズムの全ての数値フィールドで利用できるようになります。

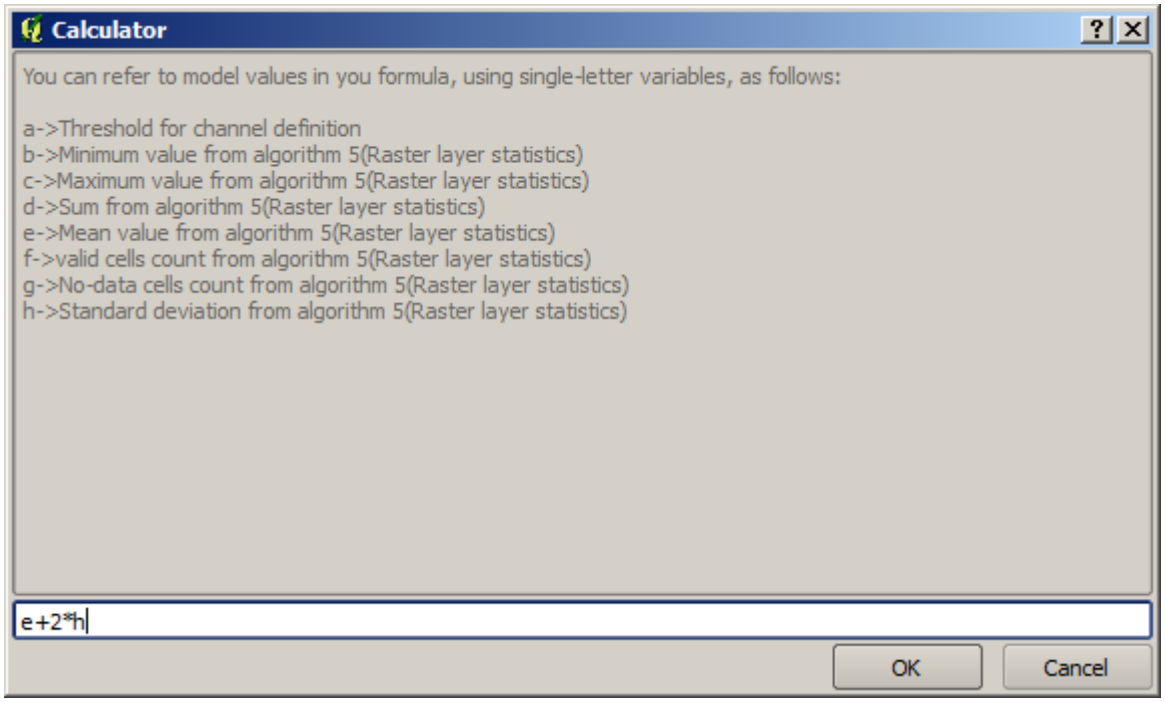
If you edit the *Channel network* algorithm, as we did in the last lesson, you will see now that you have other options apart from the numeric input that you added.



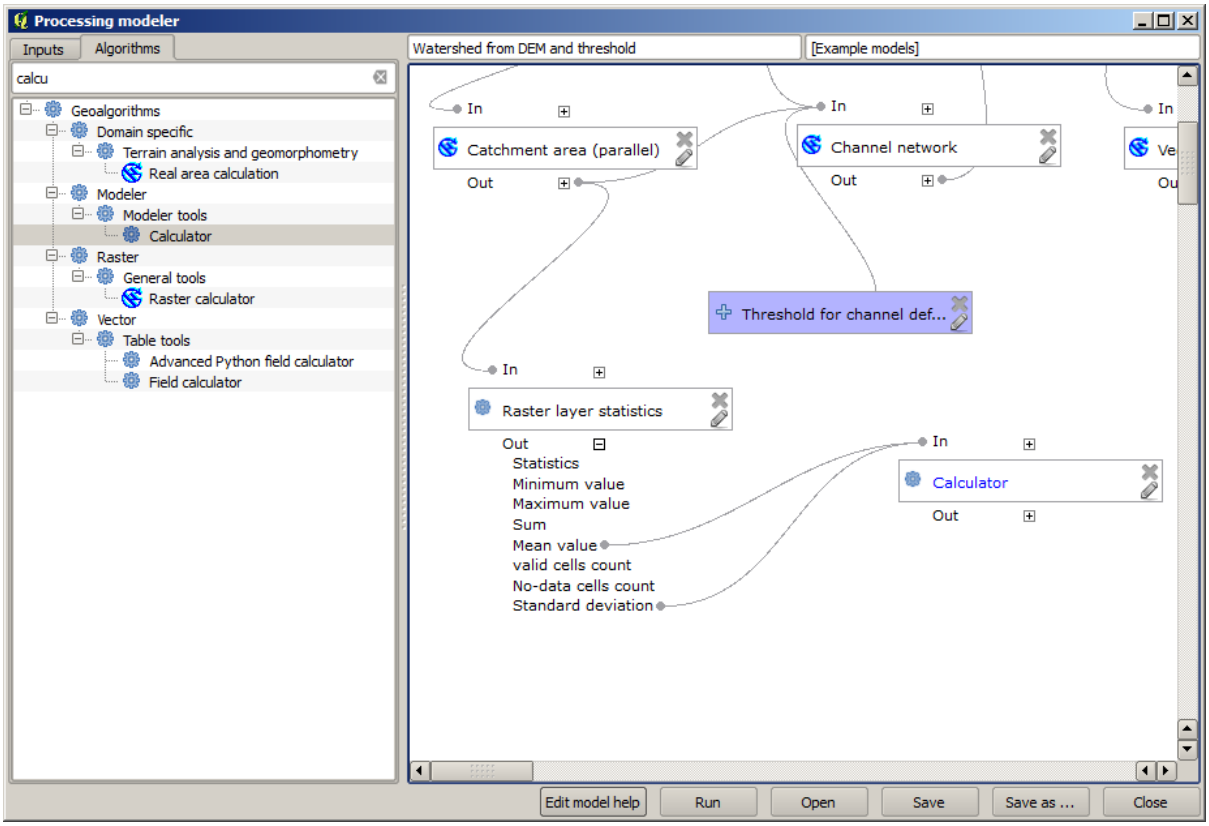
それらは非常に現実的ではない水路ネットワークになりますので、この値はいずれも、有効なしきい値として使用するのに適していません。代わりに、より良い結果を得るために、それらに基づいて、いくつかの新しいパラメーターを導出できます。例えば、平均+標準偏差の 2 倍を使用できます。

その演算処理を追加するには、*Geoalgorithms/modeler/modeler-tools* グループで見つける計算機を使用できます。このグループは、モデラーの外に非常に有用ではないアルゴリズムが含まれていますが、モデルの作成時には、いくつかの有用な機能を提供します。

計算機アルゴリズムのパラメーターダイアログボックスは次のようになります。

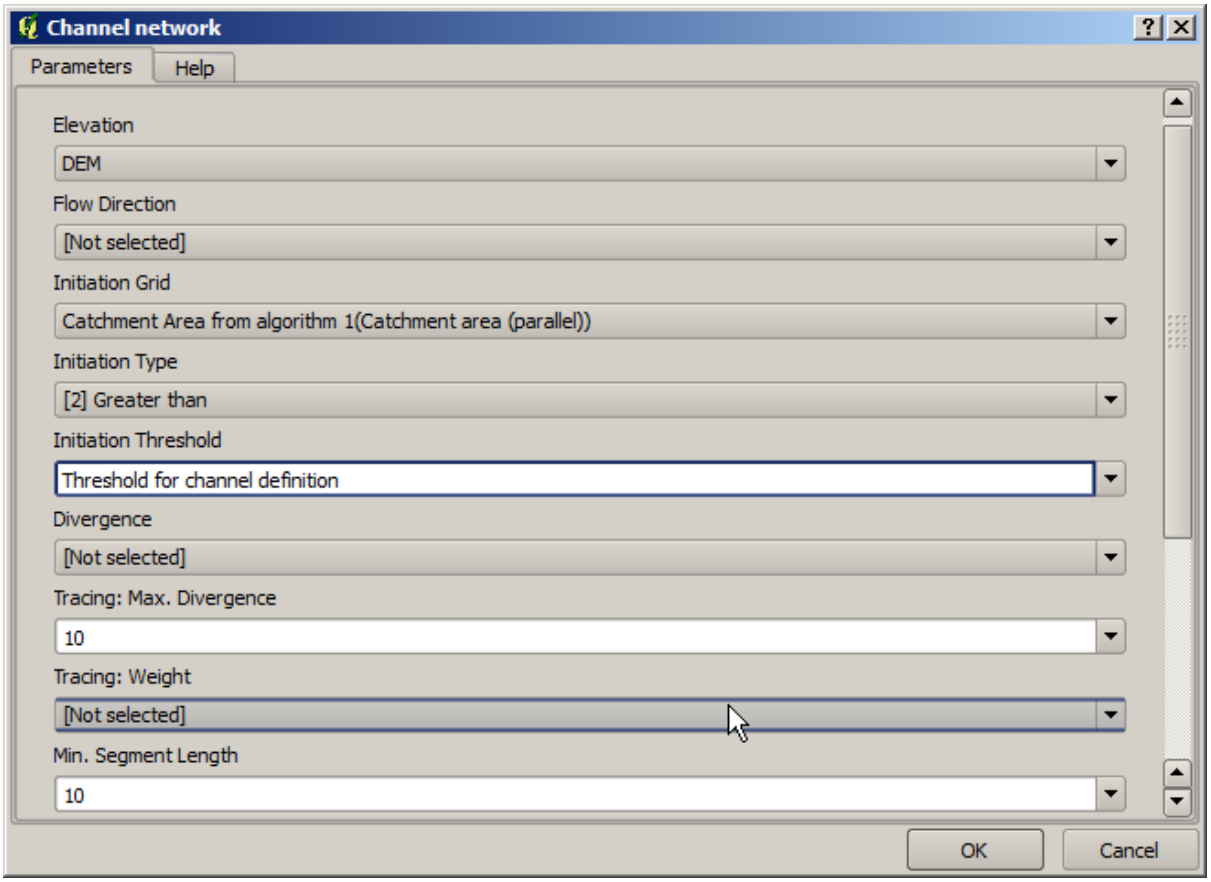


As you can see, the dialog is different to the other ones we have seen, but you have in there the same variables that were available in the `Threshold` field in the *Channel network* algorithm. Enter the above formula and click on *OK* to add the algorithm.

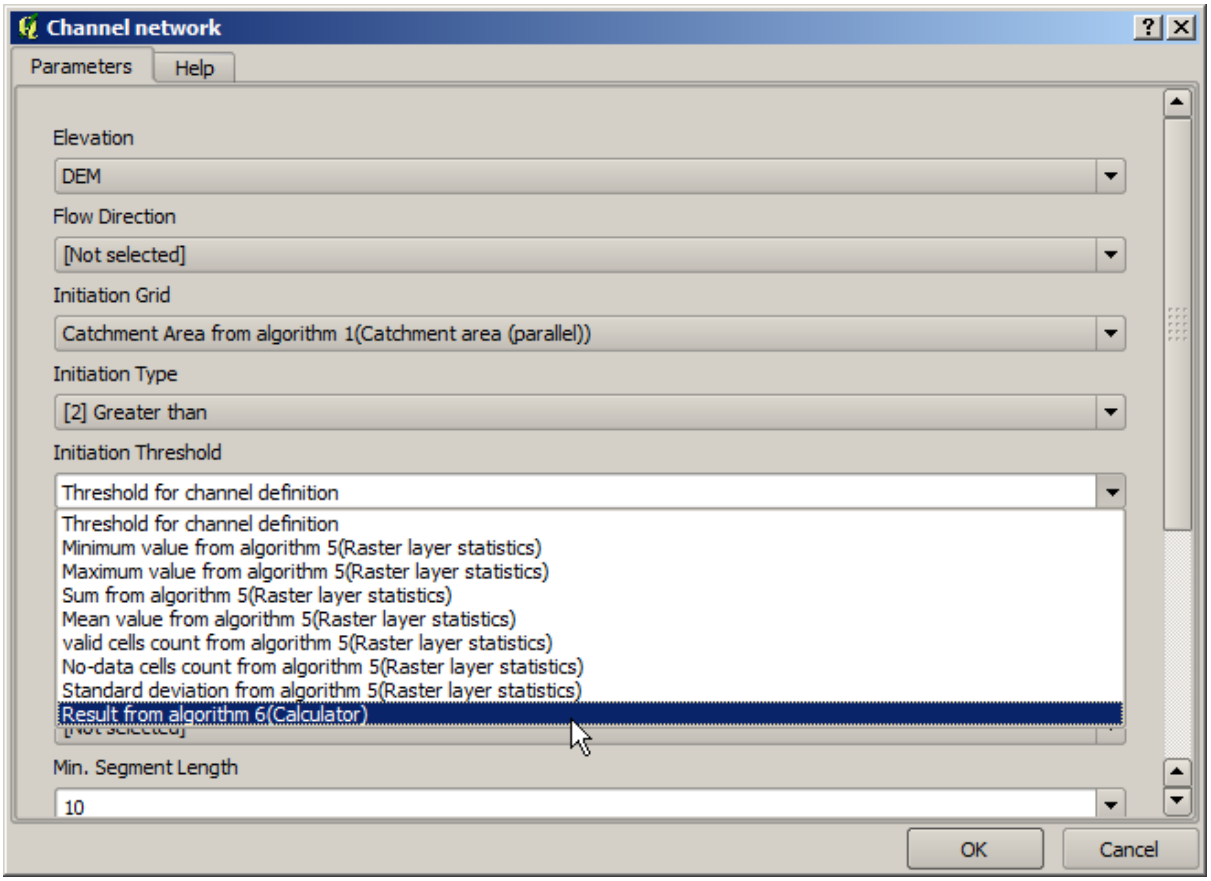


上記のように出力項目を展開すると、モデルが 2 つの値、つまり平均と標準偏差に接続されていることがわかります。これは、数式で使用した値です。

Adding this new algorithm will add a new numeric value. If you go again to the *Channel network* algorithm, you can now select that value in the `Threshold` parameter.



Click on *OK* and your model should look like this.



We are not using the numeric input that we added to the model, so it can be removed. Right-click on it and select *Remove*

> **警告**
>
> TODO：画像を追加します。

新しいモデルが完成しました。

17.20 モデル内のモデル

> **警告**
>
> 注意してください。この章はよく検証されていないため、何か問題を見つけたらご報告ください；画像は欠けています

注釈

このレッスンでは、モデルをより大きなモデル内で使用する方法について説明します。

すでにいくつかのモデルを作成してきましたが、このレッスンでは、大きい方の一つにそれらを組み合わせることができる方法を確認しようとしています。モデルはすでにその後に作成し、別のものの一部として作成したモデルを追加できることを意味し、他のアルゴリズム、同じように動作します。

この場合は、それが結果として生成流域のそれぞれに平均 TWI 値を追加することによって、水文モデルを拡張しようとしています。これを行うには、TWI を計算して、統計を計算する必要があります。すでに DEM から TWI を計算するモデルを作成しているので、代わりにそれが個別に含まれているアルゴリズムを追加することで、そのモデルを再利用することをお勧めします。

最後のレッスンのための出発点として使用するモデルを見てみましょう

警告

TODO：画像を追加します。

まず、TWI モデルを追加します。それが利用可能であるためには、ツールボックスやモデラーにおけるアルゴリズムのリストには表示されませんそうでなければ、それは、モデルフォルダに保存されている必要があります。それが用意されていることを確認します。

それを現在のモデルに追加し、その入力として入力 DEM を使用します。TWI レイヤーで統計情報を計算したいだけなので、出力は一時的なものです。私たちが作成しているこのモデルの唯一の出力は、まだ流域のベクタレイヤになるでしょう。

ここでは、対応するパラメータダイアログは次のとおりです。

警告

TODO：画像を追加します。

今、私たちはそれぞれの流域に対応する TWI の値を含む新しいものを生成するために、流域ベクタレイヤと一緒に使用できる TWI レイヤを持っています。

This calculation is done using the *Grid statistics in polygons* algorithm. Use the layers mentioned above as input, to create the final result.

警告

TODO：画像を追加します。

The output of the *Vectorize grid classes* algorithm was originally our final output, but now we just want it as an intermediate result. To change that, we have to edit the algorithm. Just double--click on it to see its parameters dialog, and delete the name of the output. That will make it a temporary output, as it is by default.

> **警告**
>
> TODO：画像を追加します。

これが最終的なモデルの姿です：

> **警告**
>
> TODO：画像を追加します。

ご覧のとおり、あるモデルを別のモデルで使うことは特別なものではありません。モデルがモデルフォルダに保存され、ツールボックスで使用可能になっているかぎり、ちょうど別のアルゴリズムを追加するように追加できます。

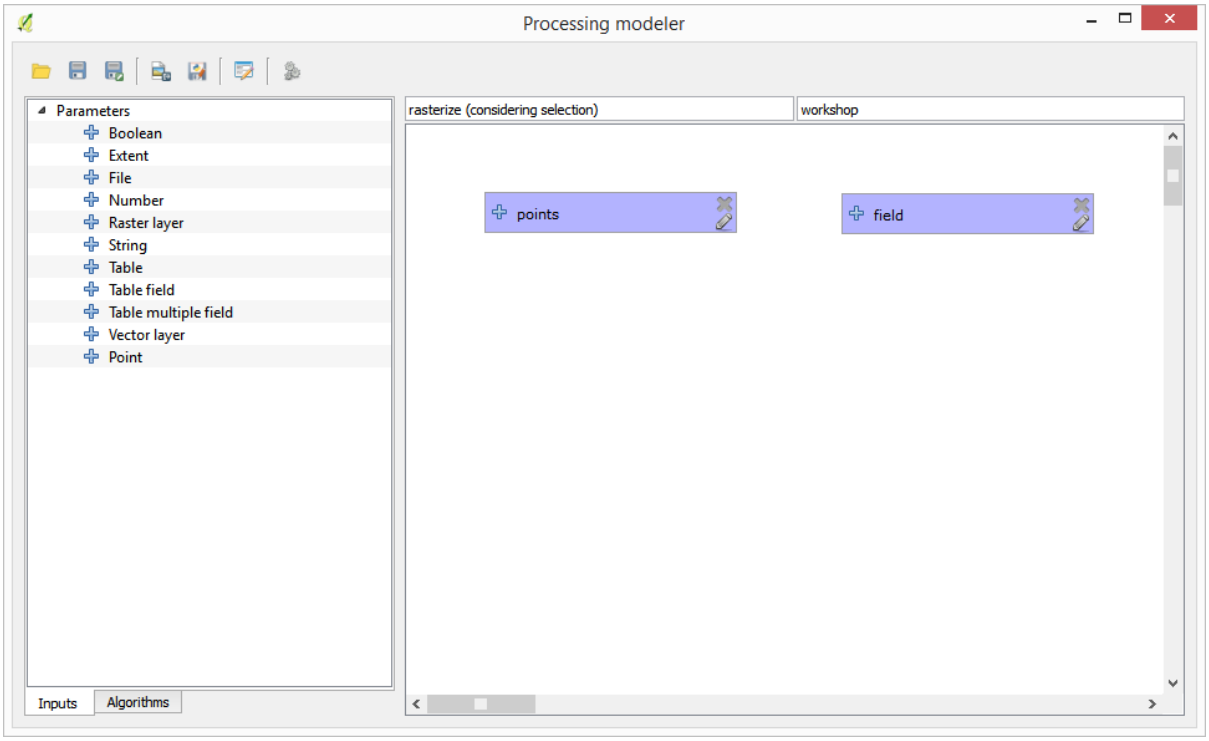
17.21 モデルを作成するためにモデラー専用ツールを使用する

> **注釈**
>
> このレッスンでは、モデルに追加機能を提供するために、モデラーでのみ利用可能ないくつかのアルゴリズムを使用する方法を示しています。

このレッスンの目標は、モデラーを使用して現在の選択を考慮に入れる補間アルゴリズムを作成すること、選択地物だけを使用するのではなく、その選択の範囲を使用して補間されるラスタレイヤを作成することです。

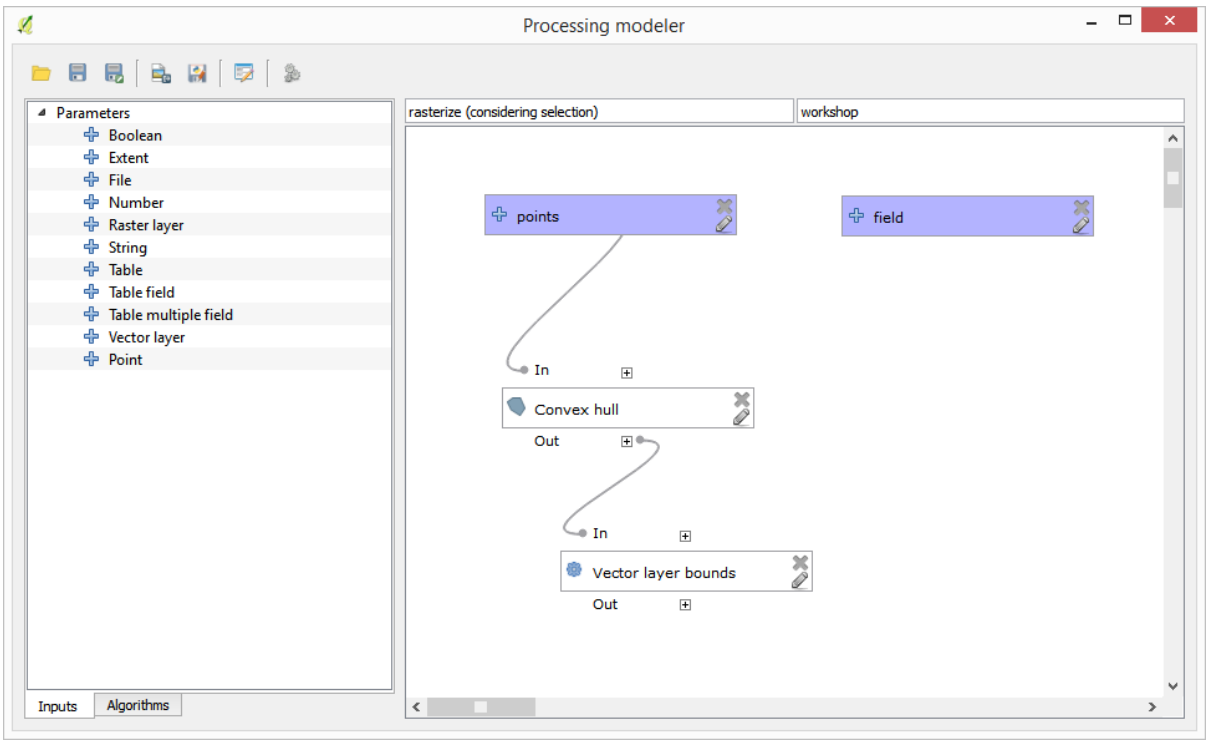
補間処理には、先のレッスンで既に説明したように、2 つのステップが含まれます。ポイントレイヤーをラスタ化し、ラスタ化されたレイヤに表示される no-data（データなし）値を埋めます。ポイントレイヤに選択範囲がある場合、選択されたポイントのみが使用されますが、出力範囲が自動的に調整されるように設定されている場合は、レイヤの全範囲が使用されます。つまり、レイヤの範囲は、選択されたものから計算されたものではなく、常にすべての地物の完全な範囲とみなされます。私たちのモデルにいくつかの追加ツールを使うことによってそれを修正してみます。

モデラーを開き、必要な入力を追加することによって、モデルを開始します。この場合は、ベクタレイヤ（ポイントに制限）と、ラスタ化に使う値を持ったその属性が必要です。

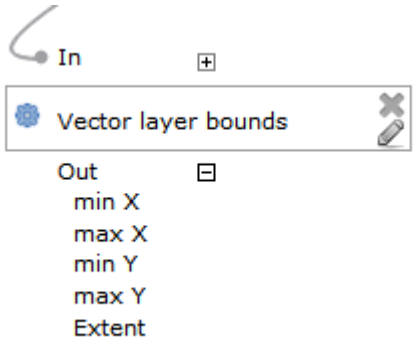


The next step is to compute the extent of the selected features. That's where we can use the model-only tool called *Vector layer bounds*. First, we will have to create a layer that has the extent of those selected features. Then, we can use this tool on that layer.

An easy way of creating a layer with the extent of the selected features is to compute a convex hull of the input points layer. It will use only the selected point, so the convex hull will have the same bounding box as the selection. Then we can add the *Vector layer bounds* algorithm, and use the convex hull layer as input. It should look this in the modeler canvas:

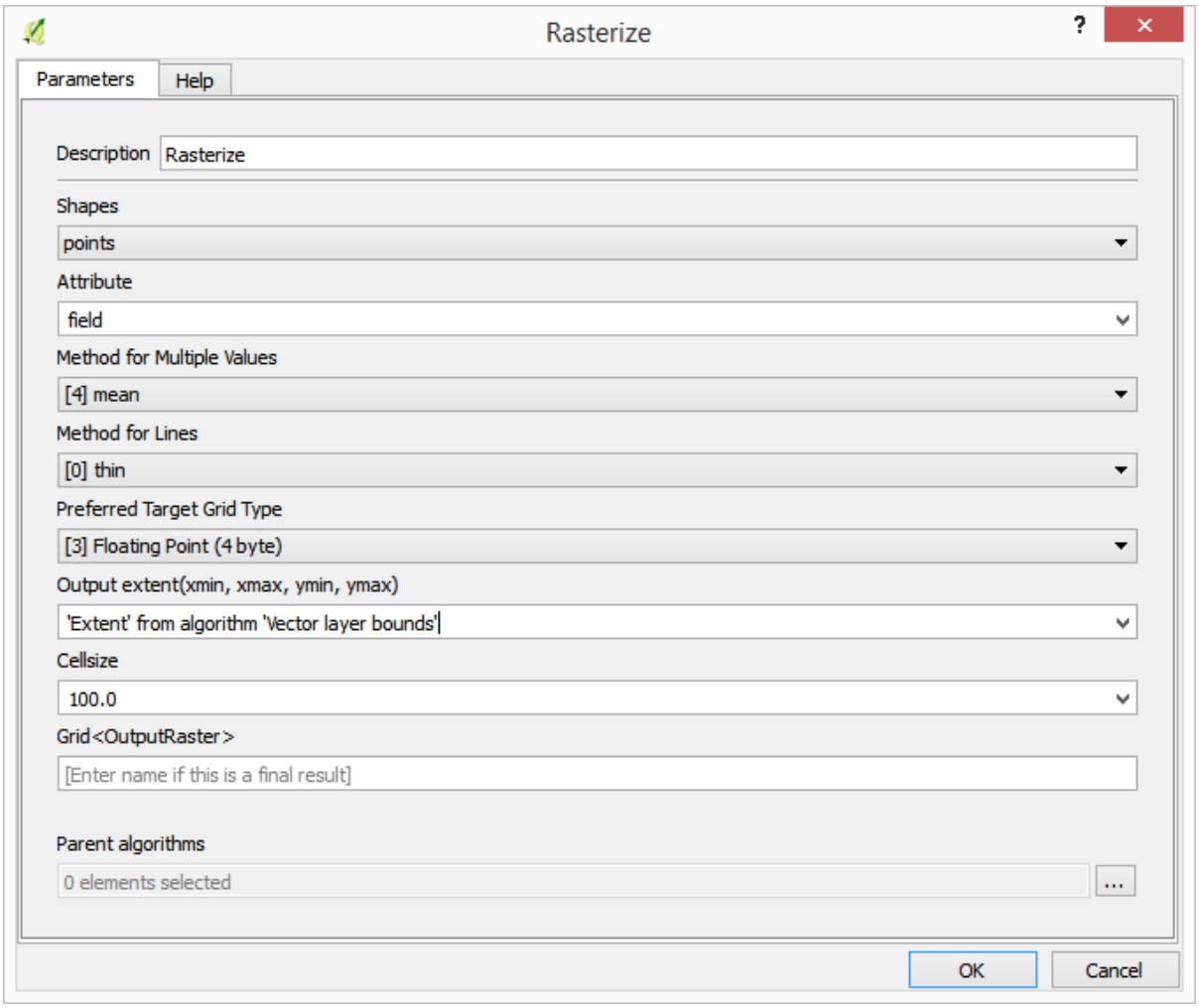


The result from the *Vector layer bounds* is a set of four numeric values and a extent object. We will use both the numeric outputs and the extent for this exercise.

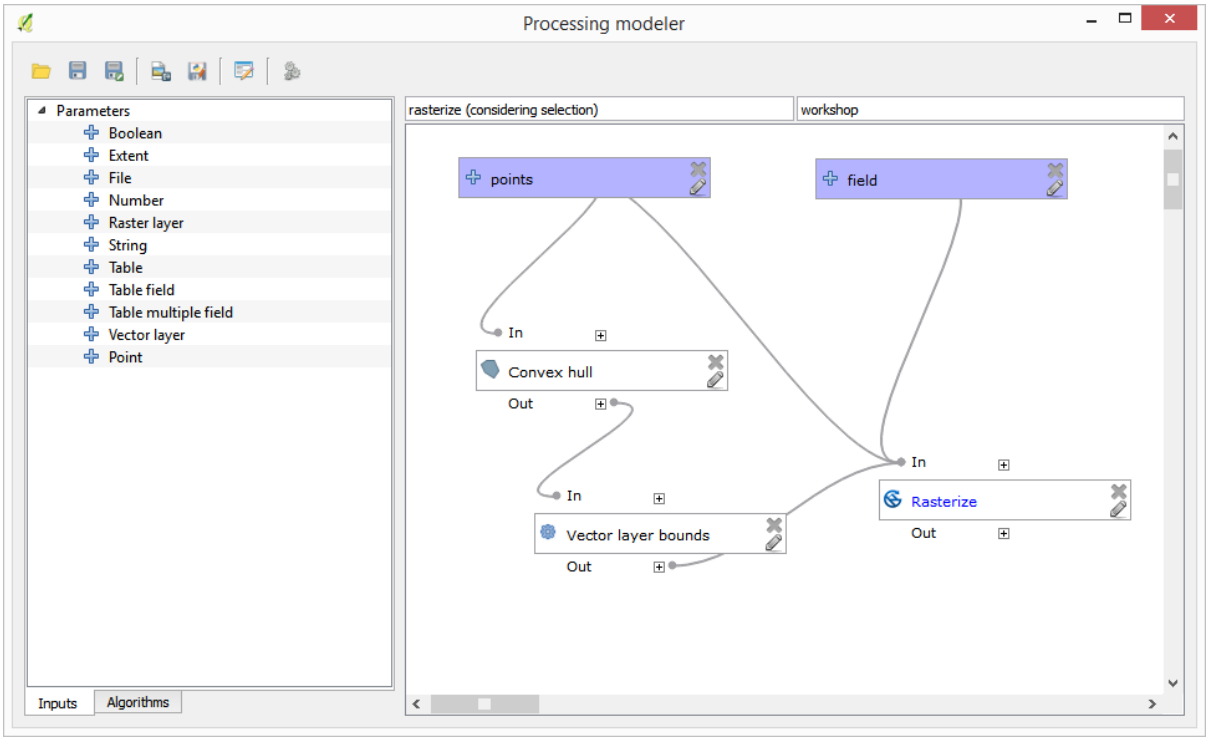


We can now add the algorithm that rasterizes the vector layer, using the extent from the *Vector layer bounds* algorithm as input.

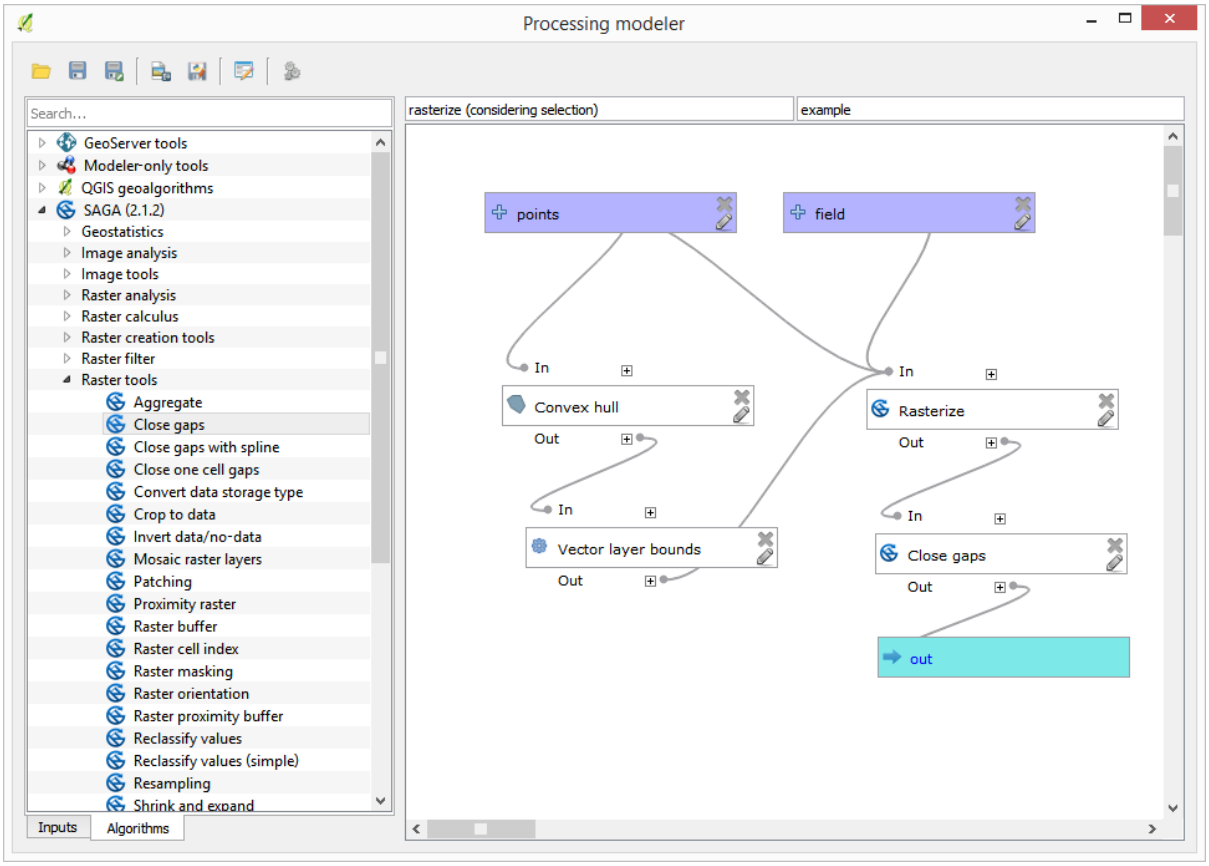
次に示すようにアルゴリズムのパラメータを入力します：



キャンバスはこのようになる筈です。



Finally, fill the no-data values of the raster layer using the *Close gaps* algorithm.

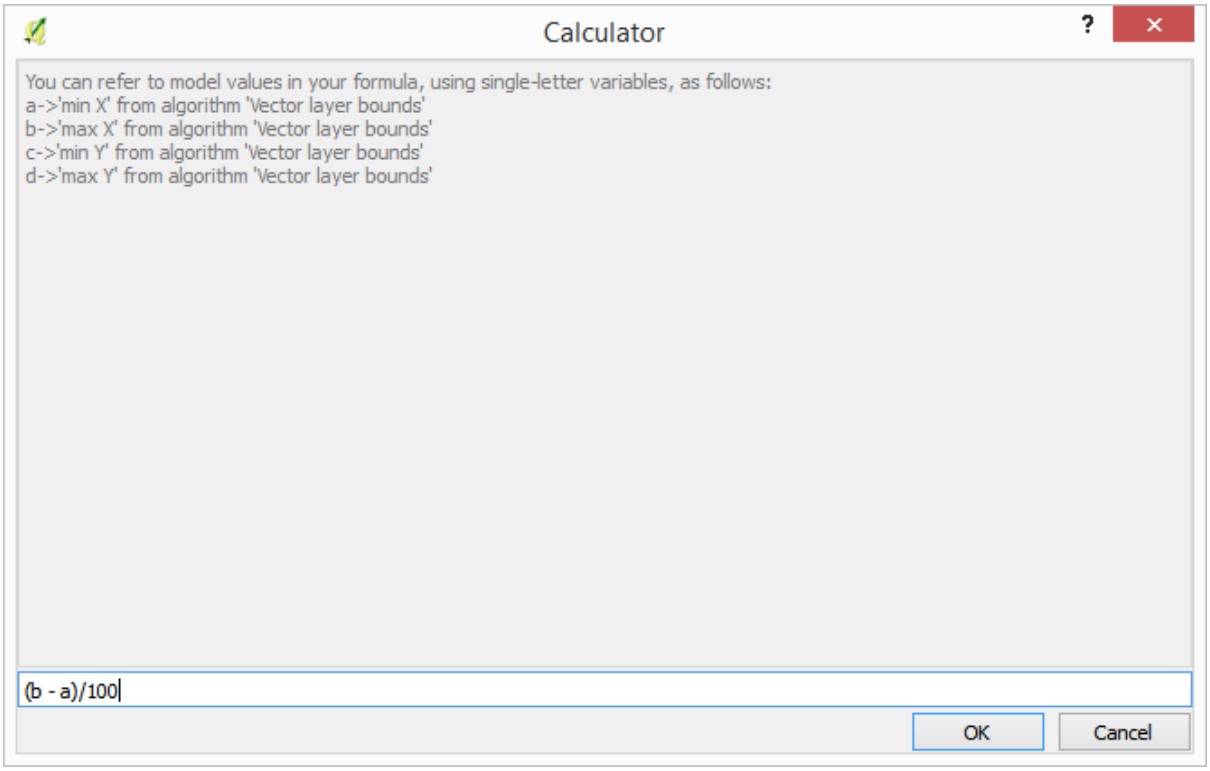


これでアルゴリズムは保存して、ツールボックスに追加する準備が整いました。それを実行すると、入力レイヤで選択されたポイントを補間することでラスタレイヤが作成され、そのレイヤは選択と同じ範囲に

なります。

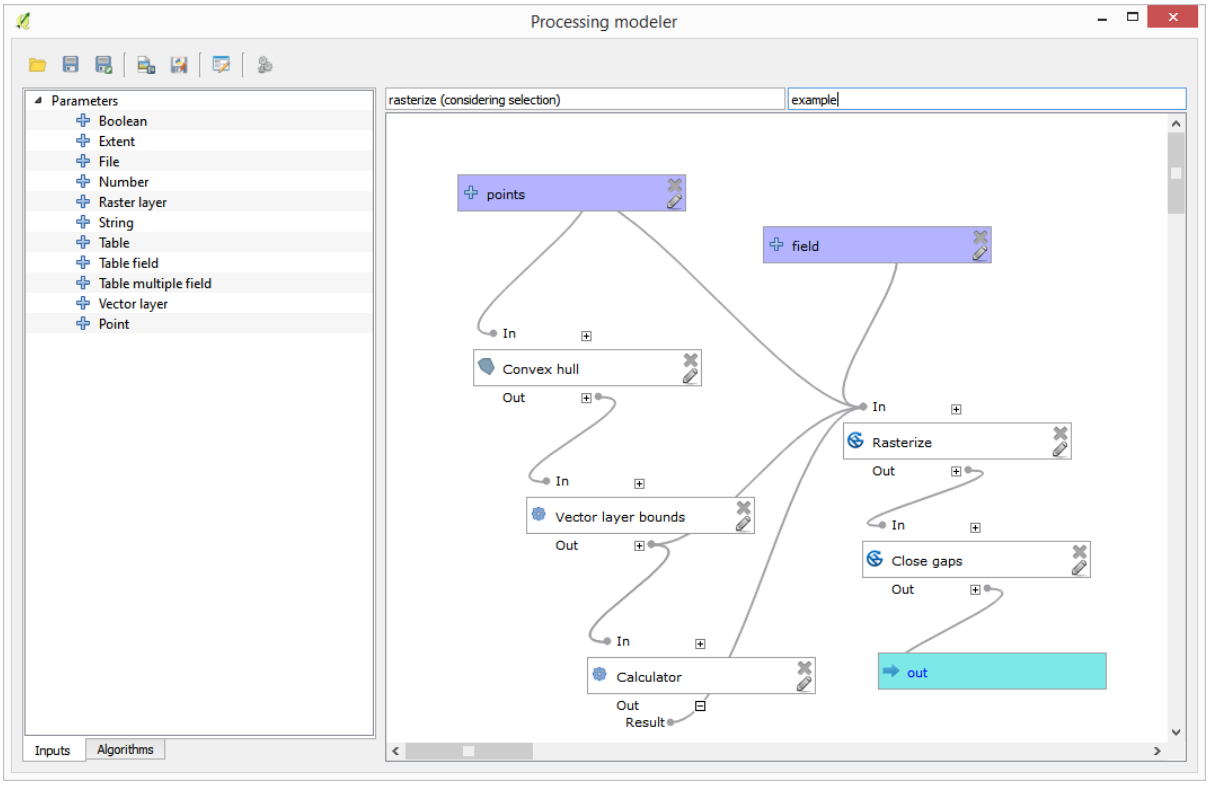
ここでアルゴリズムの改善があります。ラスタ化する時、セルサイズはハードコードされた値を使いました。この値は、ここでのテスト入力レイヤーにとっては良いですが、他の場合ではそうではないかもしれません。新しいパラメータを追加してユーザーが希望する値を入力することもできますが、より良いアプローチは、その値を自動的に計算させることでしょう。

モデラー専用計算機を使用し、範囲座標からその値を計算できます。例えば、固定幅の 100 ピクセルのレイヤーを作成するには、計算機で次の式を使用できます。



ここでラスター化アルゴリズムを、ハードコード値の代わりに計算機の出力を使用するように編集する必要があります。

最終的なアルゴリズムはこうなるはずです：



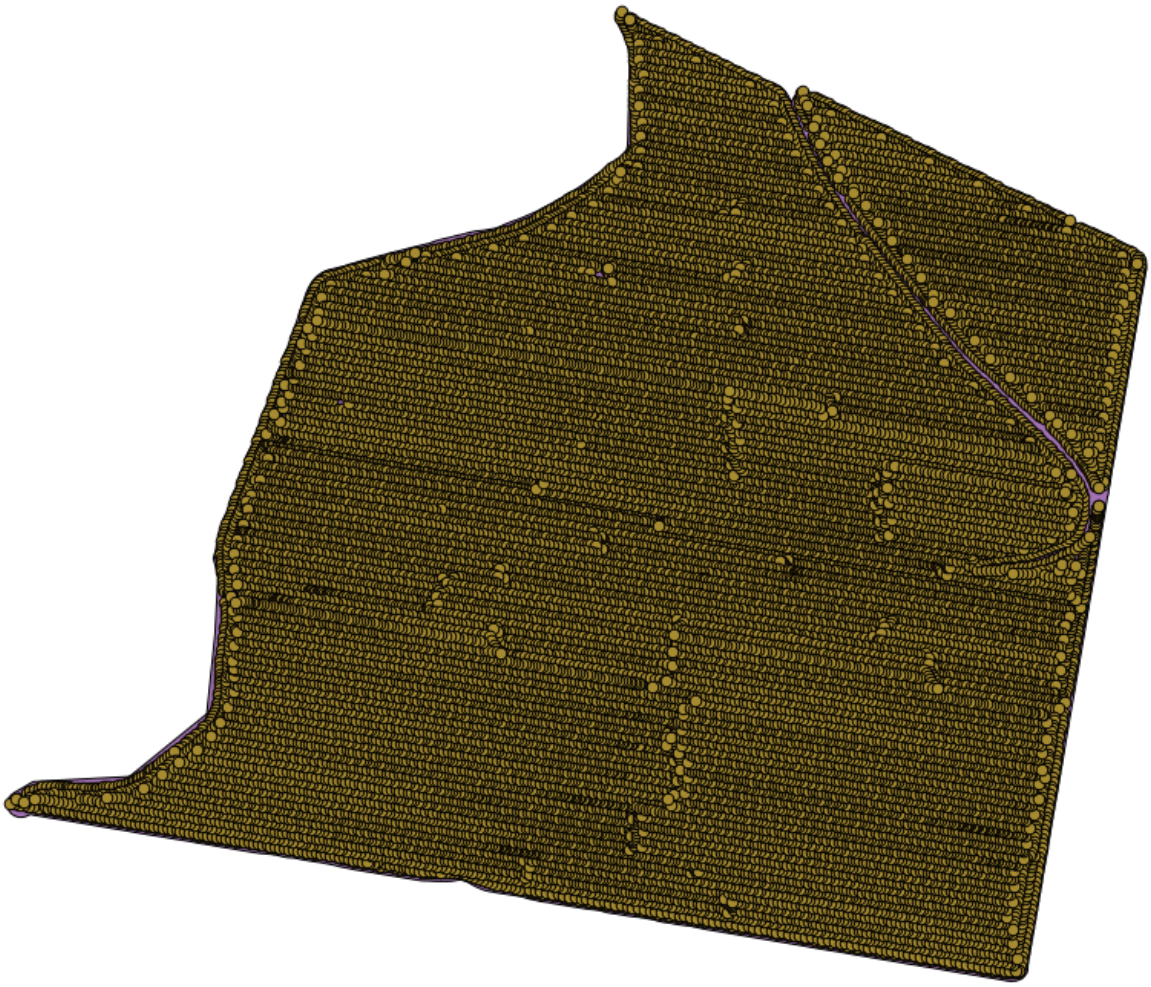
17.22 内挿

注釈

この章では、ポイントデータの補間方法について示し、空間分析の実際の実行例を示します。

このレッスンでは、ポイントデータを補間してラスタレイヤを得ます。それをする前に、若干のデータを準備する必要があります。また補間した後では、結果レイヤを修正する追加処理をいくつか加えるので、完全な分析ルーチンになるでしょう。

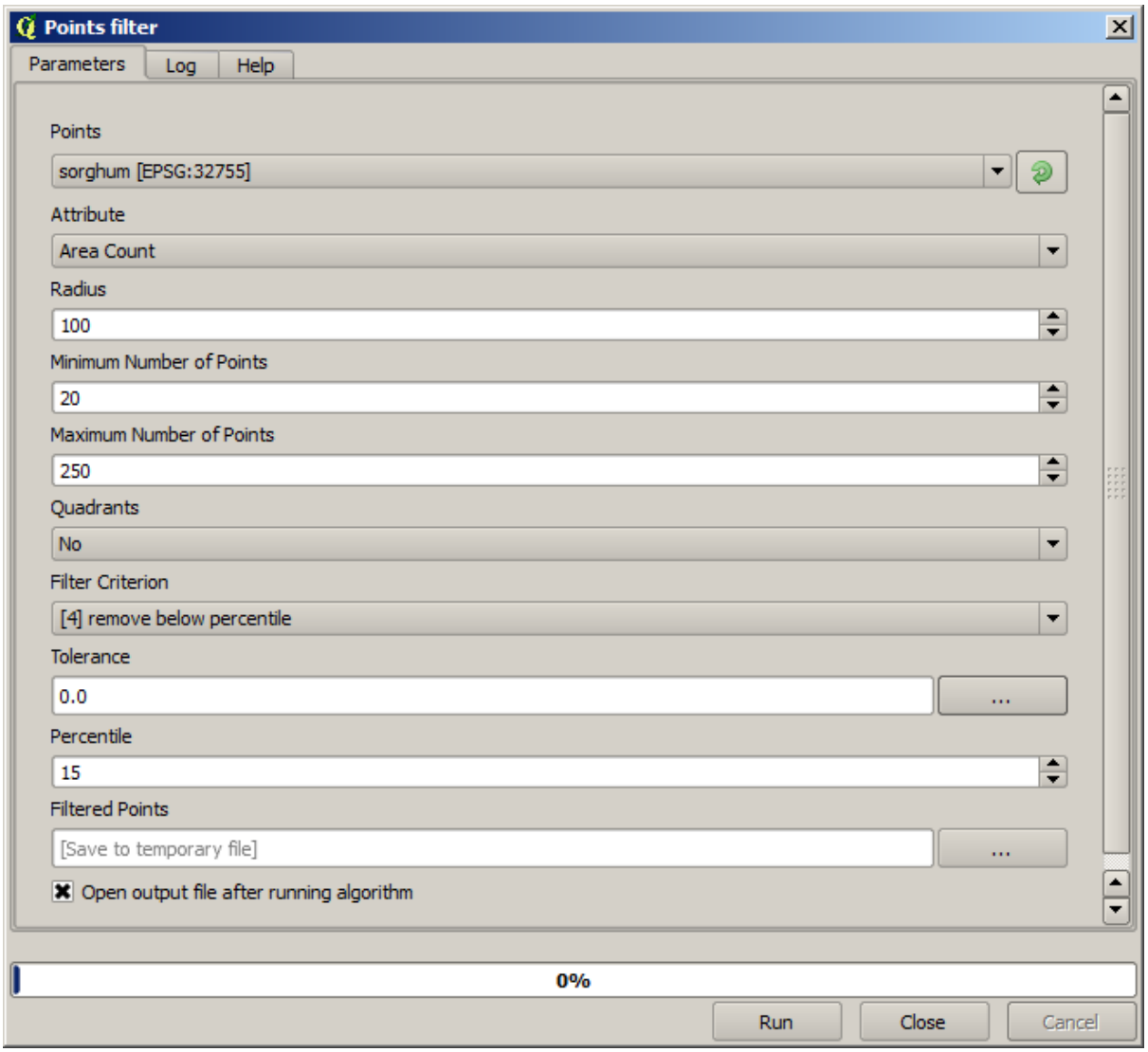
このレッスンの例となるデータを開くと、このように見えます。



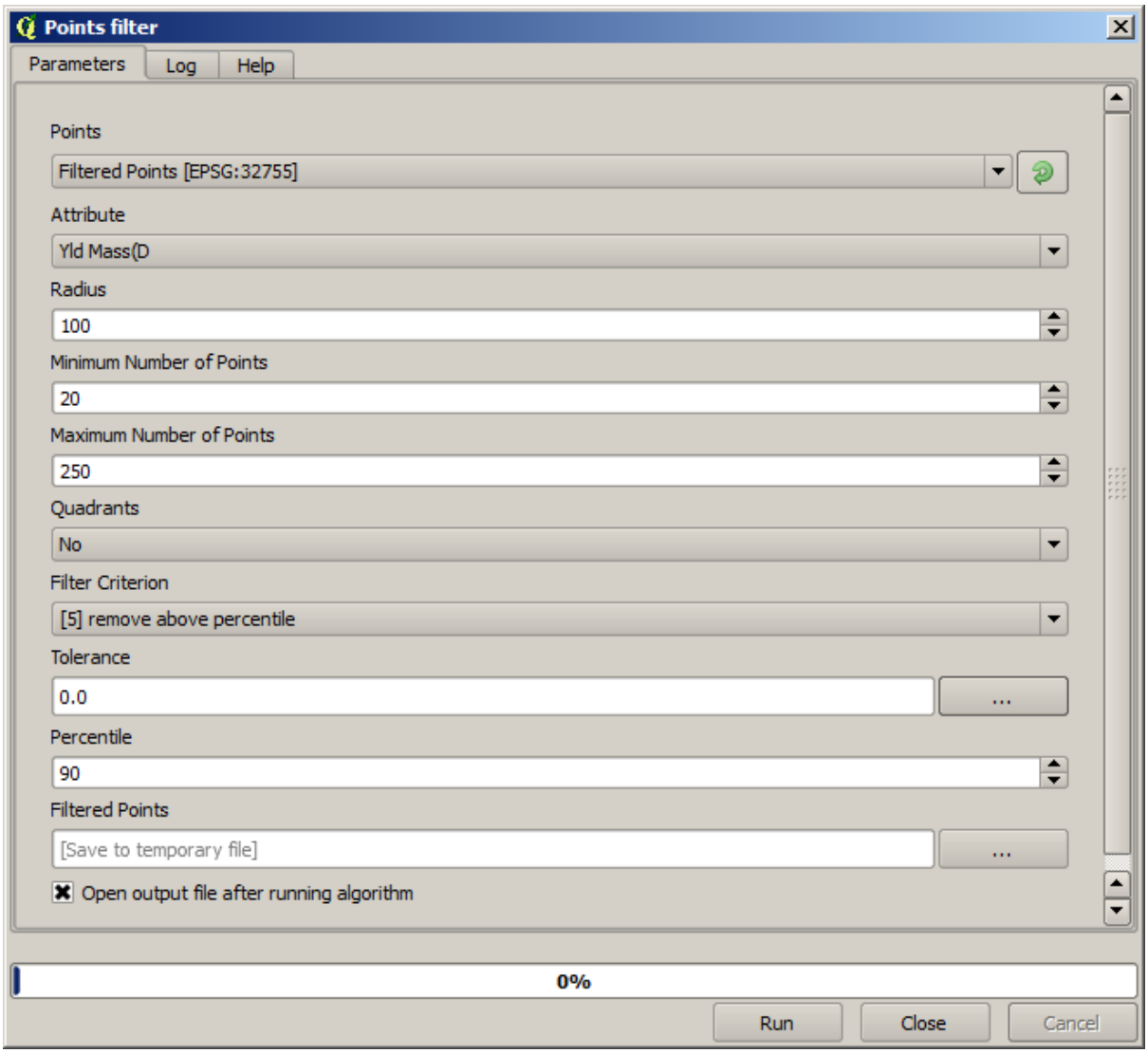
データは、最新の収穫機によって作られるような、収穫産出高データ、に対応します。そして、ここではそれを収穫産出高のラスタレイヤを得るために使います。そのレイヤでさらに何か分析する計画はなく、最も生産的な地域、そして生産性が改善できる地域を簡単に特定するための背景レイヤとしてのみ使用します。

The first thing to do is to clean--up the layer, since it contains redundant points. These are caused by the movement of the harvester, in places where it has to do a turn or it changes its speed for some reason. The *Points filter* algorithm will be useful for this. We will use it twice, to remove points that can be considered outliers both in the upper and lower part of the distribution.

最初の実行には、次のパラメーター値を使います。



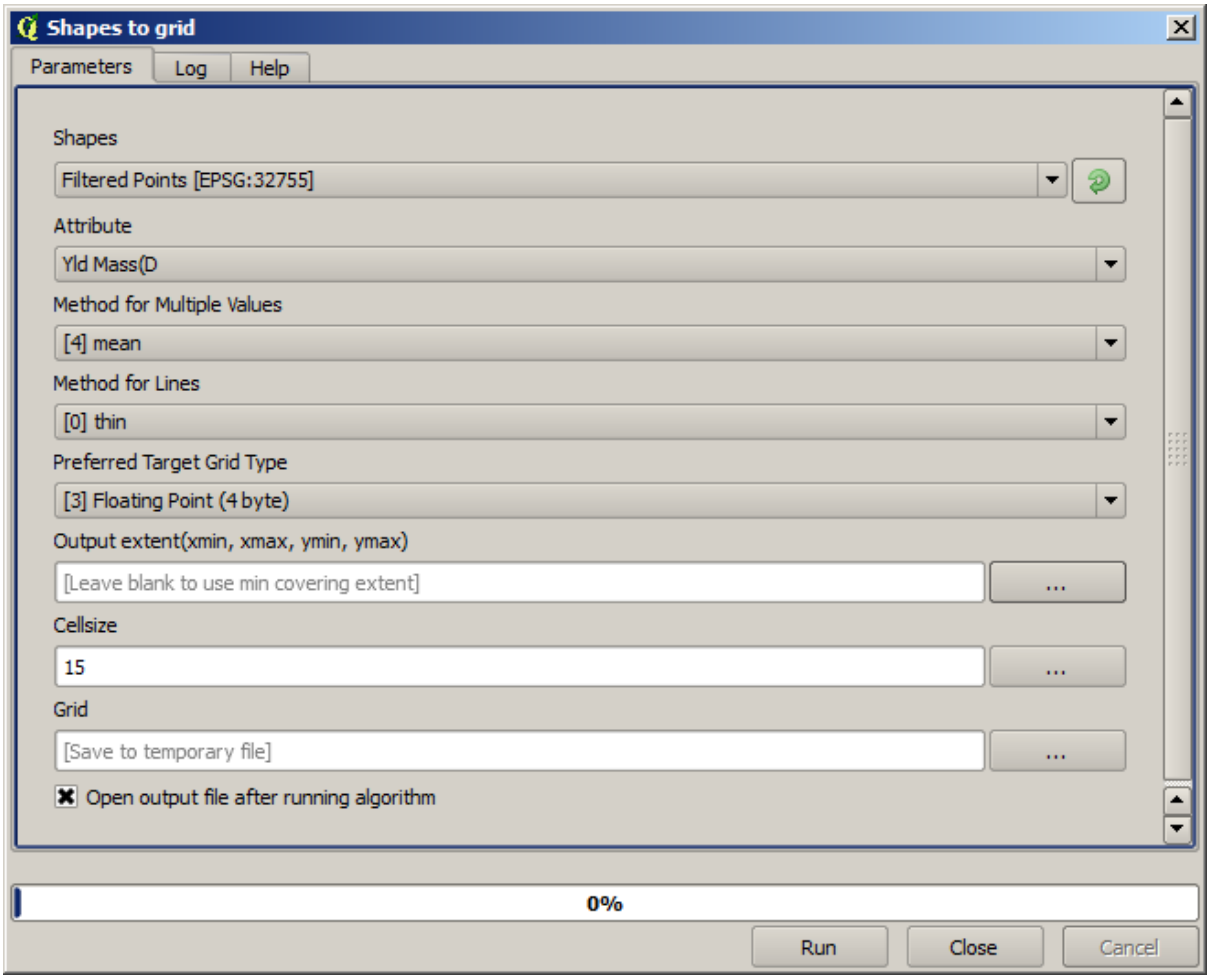
次の実行には、以下に示す設定を使用します。



入力としてオリジナルレイヤーを使用しているのではなく、以前に実行した際の出力を使用している点に注意してください。

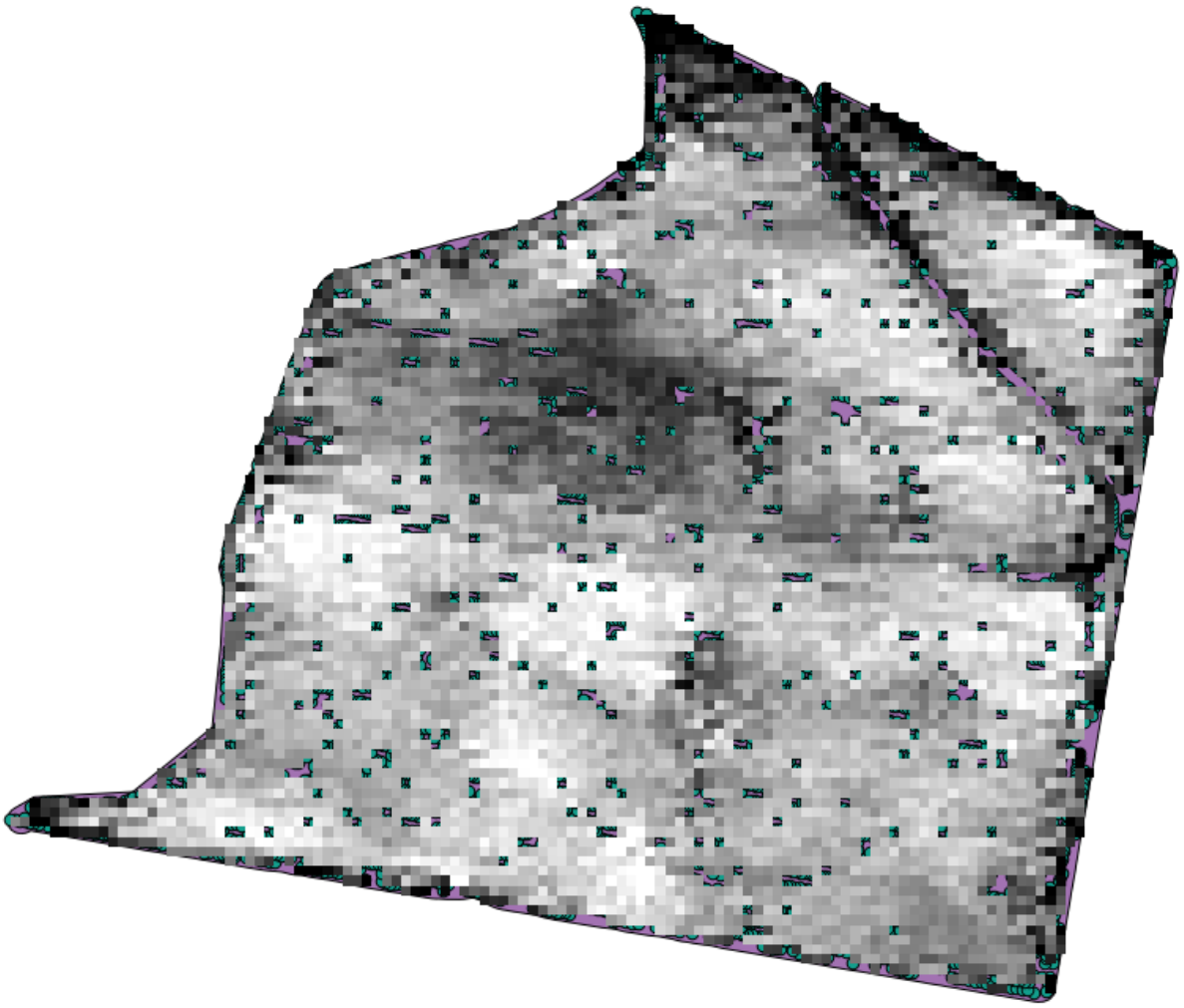
最終的なフィルタレイヤー、減らされたポイントのセットのもの、はオリジナルものと類似して見えるはずですが、含んでいるポイントはより少ないです。それらの属性テーブルで比較すると確認できます。

Now let's rasterize the layer using the *Rasterize* algorithm.

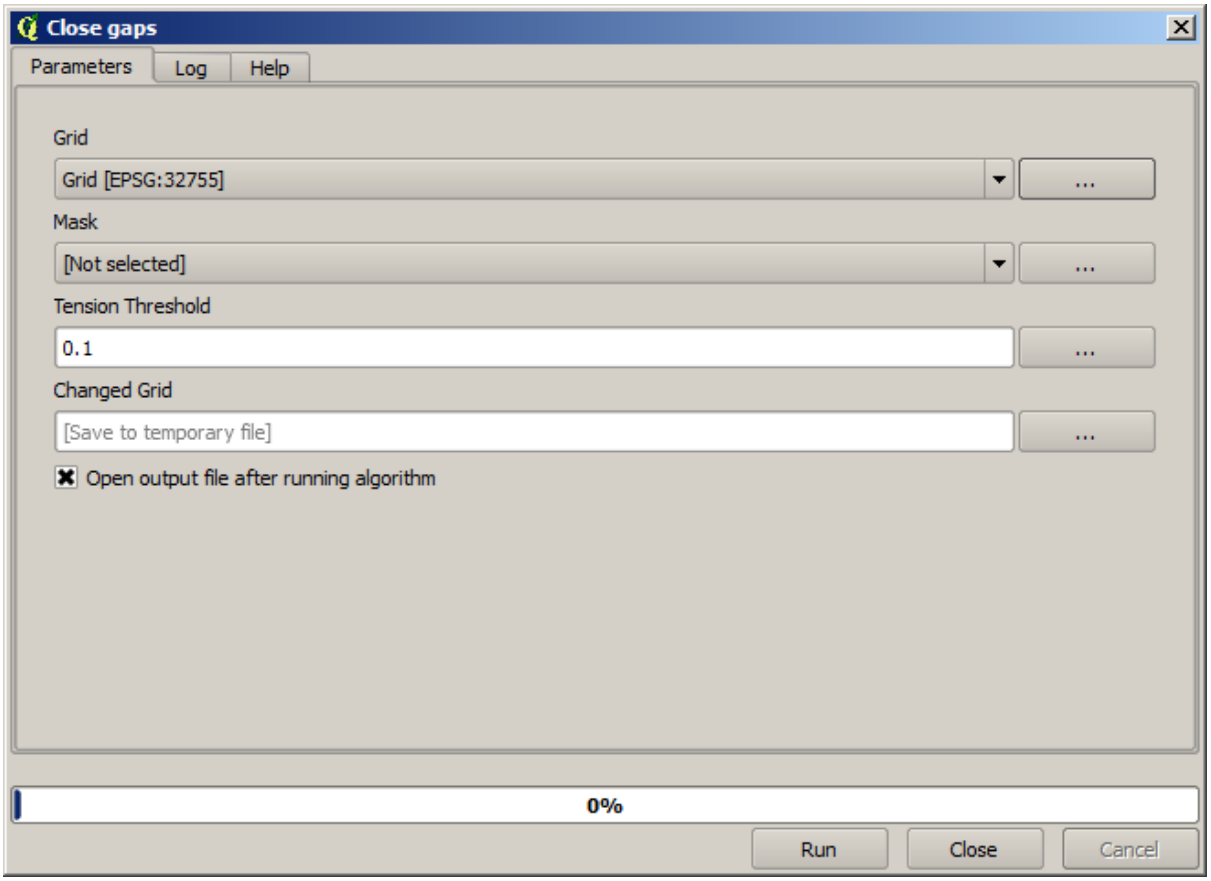


The `Filtered points` layer refers to the resulting one of the second filter. It has the same name as the one produced by the first filter, since the name is assigned by the algorithm, but you should not use the first one. Since we will not be using it for anything else, you can safely remove it from your project to avoid confusion, and leave just the last filtered layer.

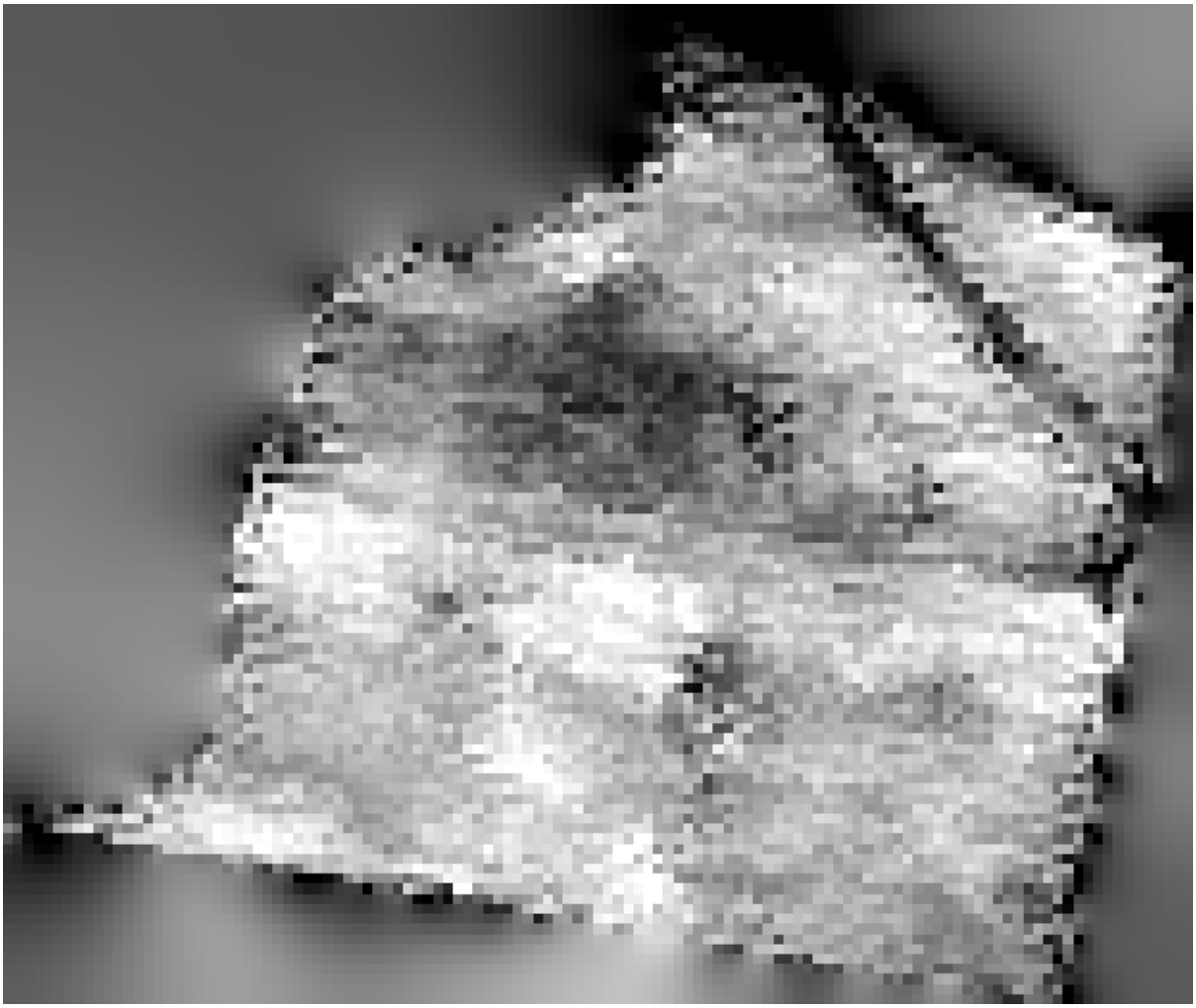
結果のラスターレイヤーはこのように見えます。



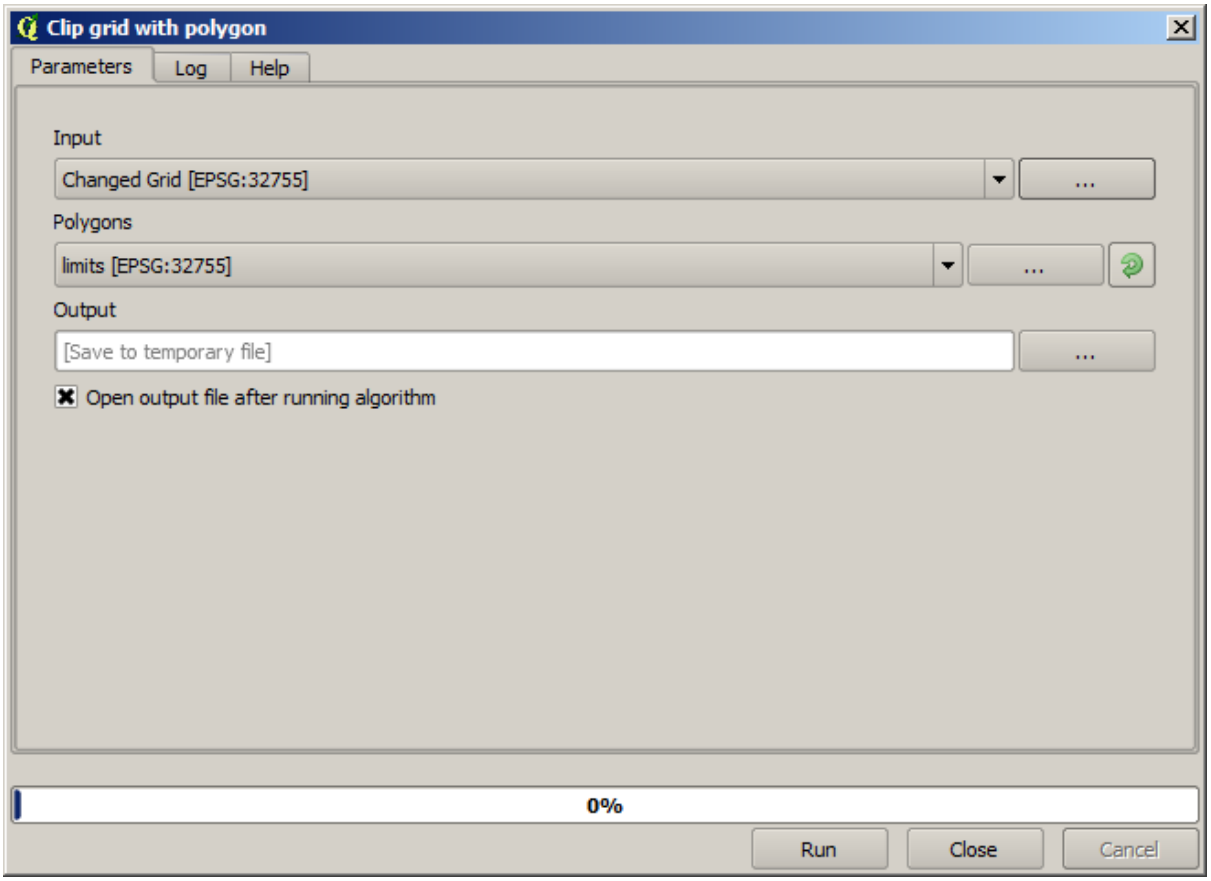
It is already a raster layer, but it is missing data in some of its cells. It only contain valid values in those cells that contained a point from the vector layer that we have just rasterized, and a no--data value in all the other ones. To fill the missing values, we can use the *Close gaps* algorithm.



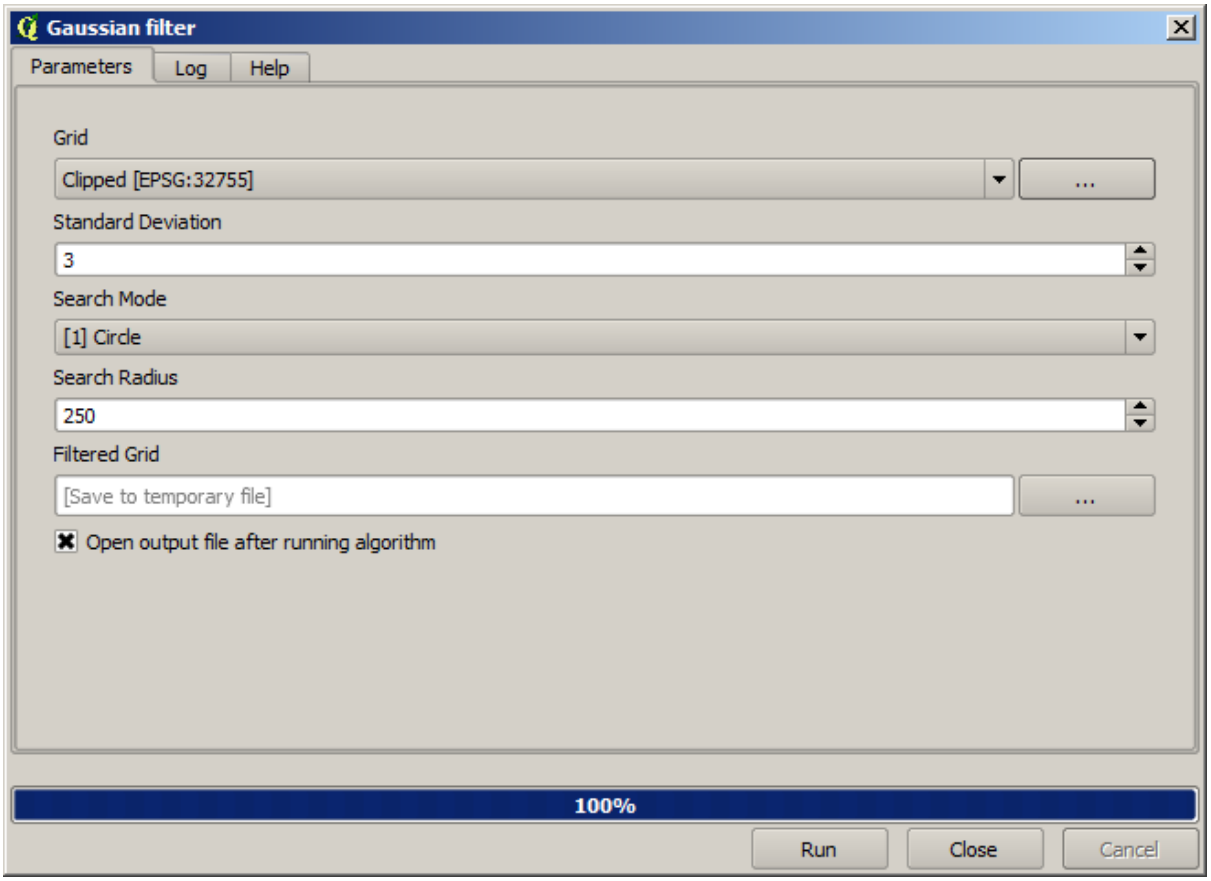
データなし値のないレイヤーはこのように見えます。



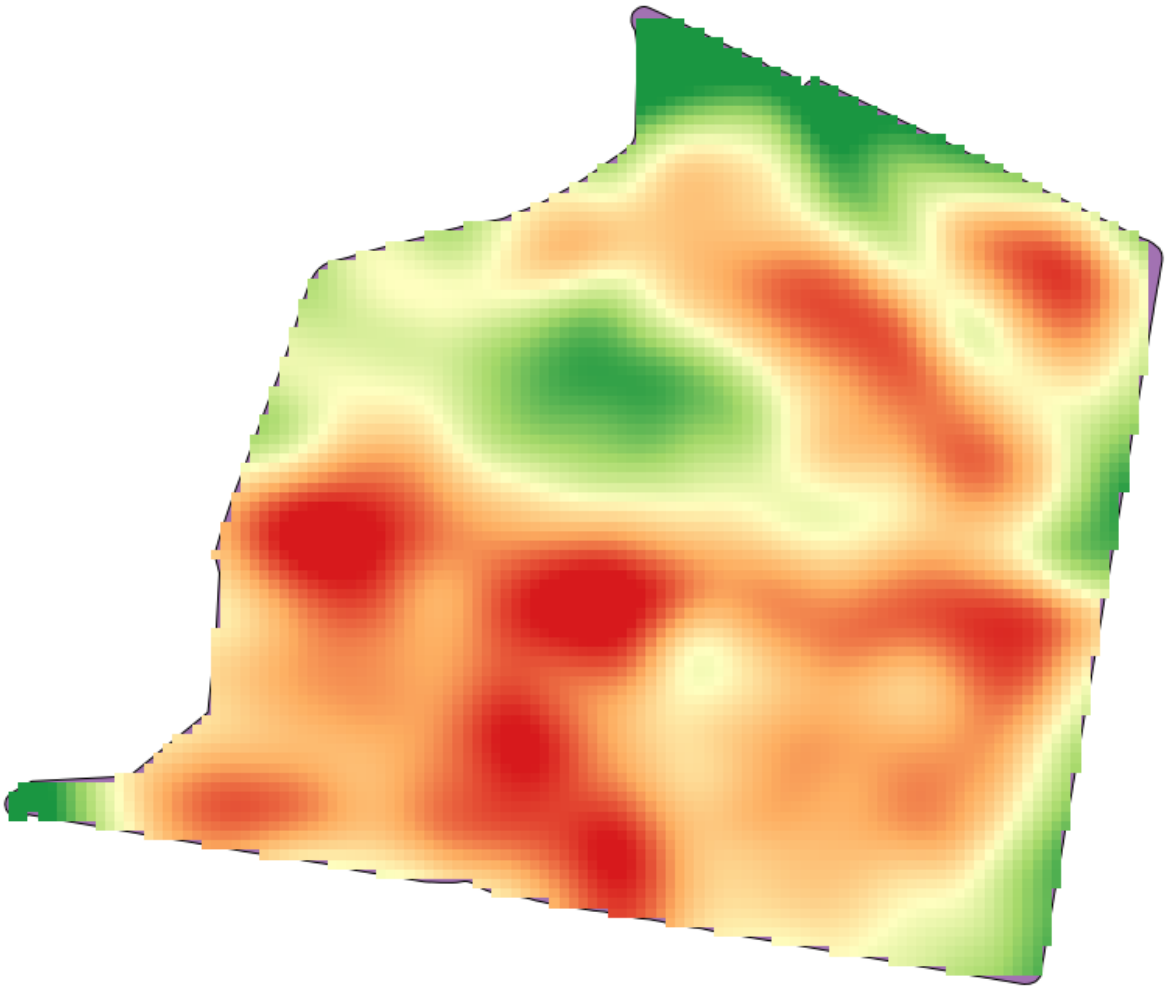
収穫産出高が計られた地域において、データによっておおわれる地域を制限するために、提供された制限レイヤーでラスターレイヤーを切り抜くことができます。



And for a smoother result (less accurate but better for rendering in the background as a support layer), we can apply a *Gaussian filter* to the layer.



上記のパラメーターで、次の結果が得られます。



17.23 補間（続）

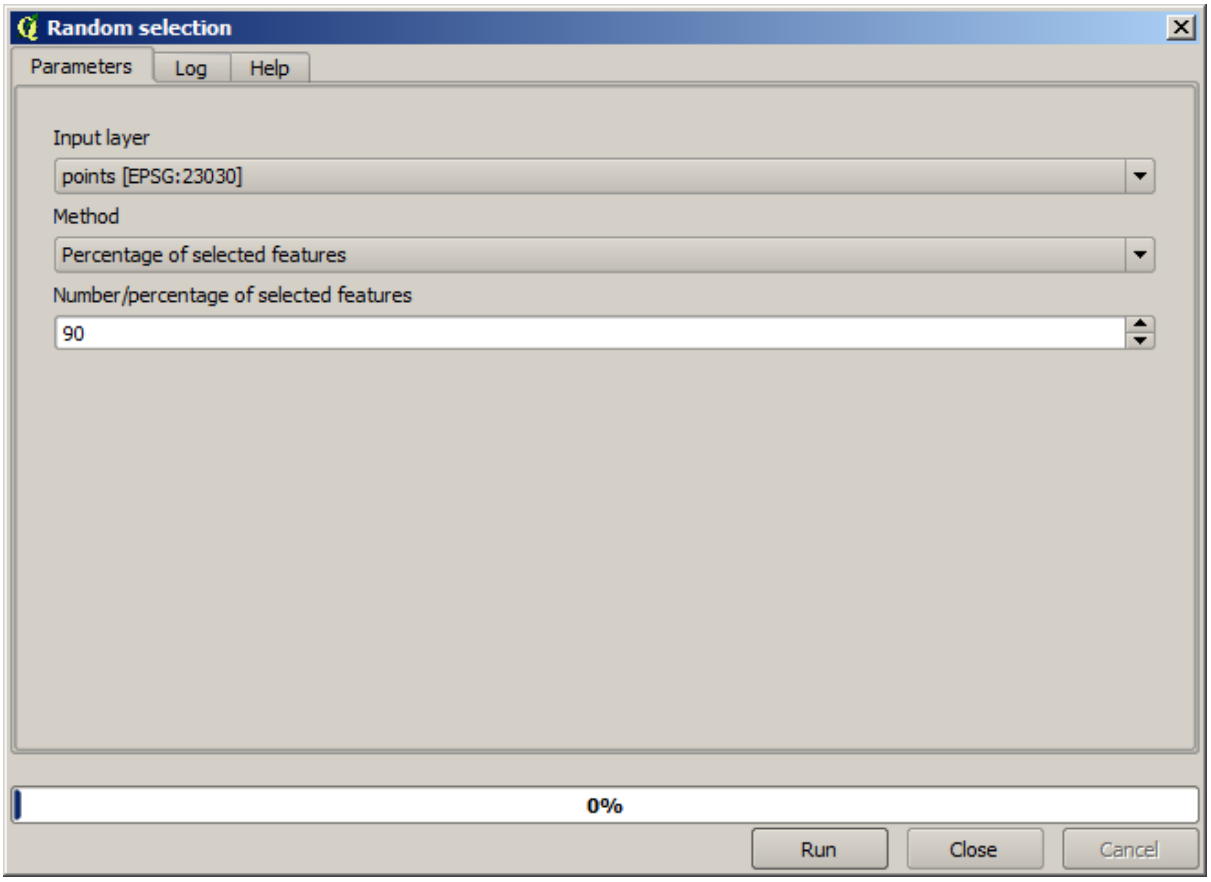
> **注釈**
>
> この章では、補間アルゴリズムが使用される別の実用的な場合を示しています。

補間は一般的な技術であり、QGIS 処理フレームワークを使用して適用できるいくつかの技術を実証するために使用できます。このレッスンでは、すでに導入されたいくつかの補間アルゴリズムを使用しますが、別のアプローチです。

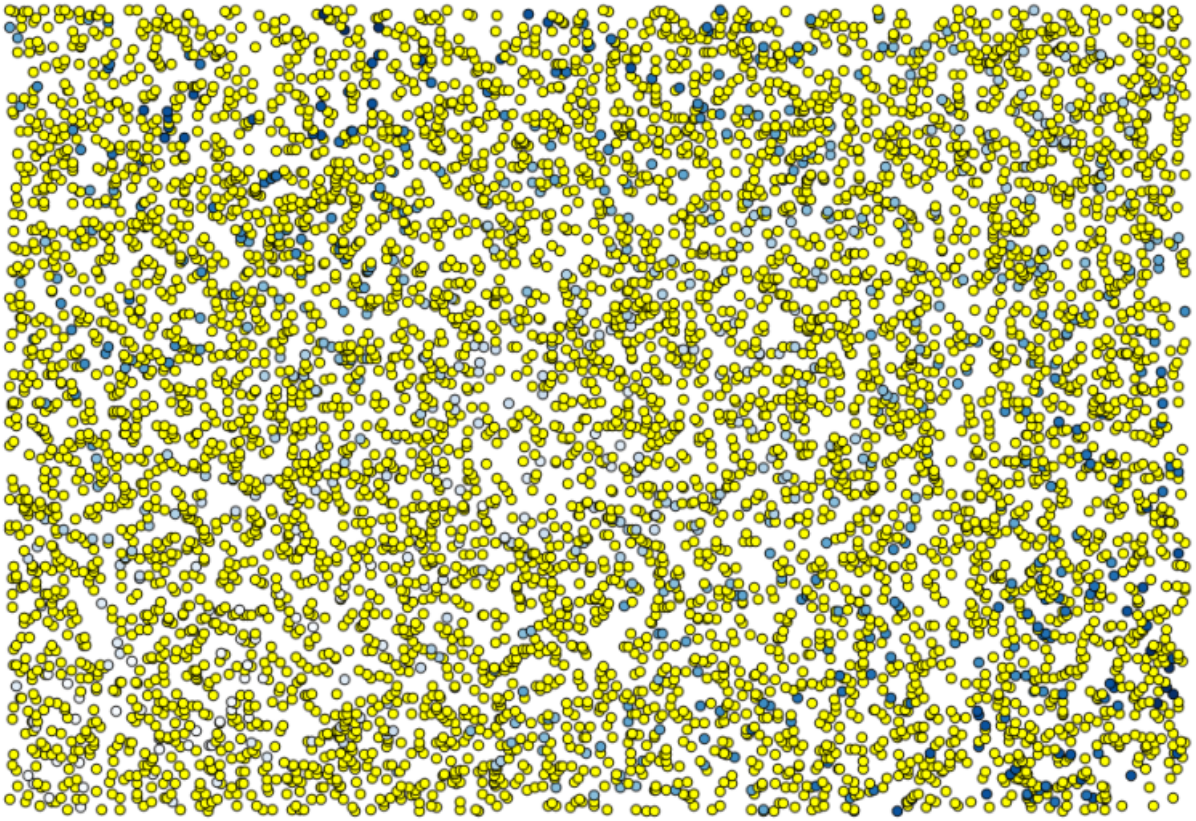
このレッスン用のデータもポイントレイヤーを含んでいますが、この場合は標高データです。前のレッスンで行ったのとずっと同じ方法でそれを補間しようとしているが、今回は元のデータの一部を保存しておいて、補間処理の品質を評価するためにそれを使用します。

First, we have to rasterize the points layer and fill the resulting no--data cells, but using just a fraction of the points in the layer. We will save 10% of the points for a later check, so we need to have 90% of the points ready for the interpolation. To do so, we could use the *Split shapes layer randomly* algorithm, which we have already used in a

previous lesson, but there is a better way to do that, without having to create any new intermediate layer. Instead of that, we can just select the points we want to use for the interpolation (the 90% fraction), and then run the algorithm. As we have already seen, the rasterizing algorithm will use only those selected points and ignore the rest. The selection can be done using the *Random selection* algorithm. Run it with the following parameters.

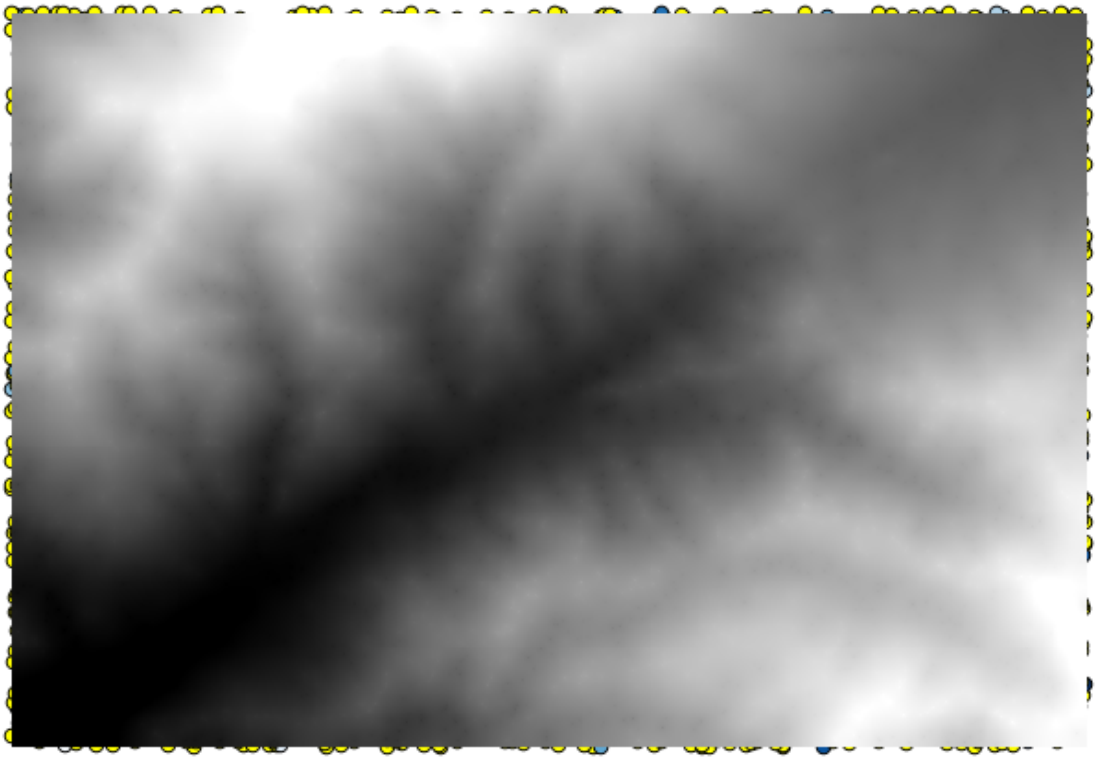


それはラスター化するレイヤー内の点の 90 %を選択します



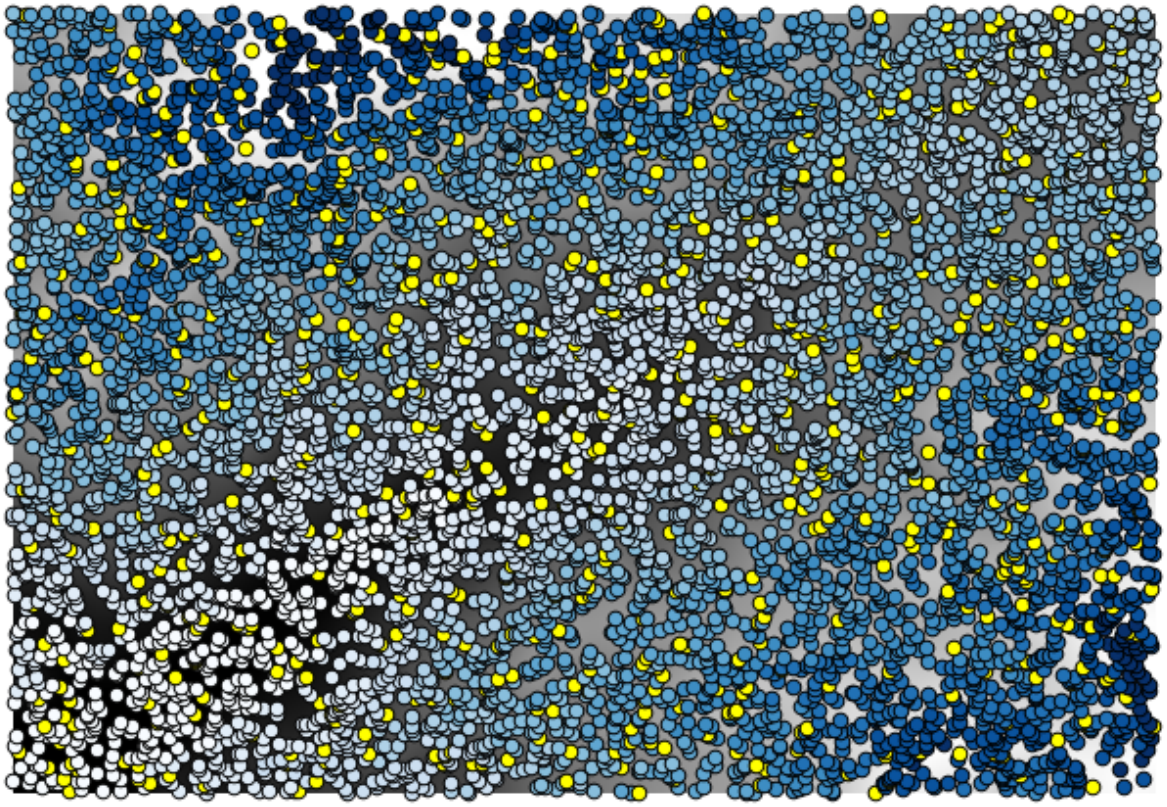
選択はランダムなので、選択が上記の画像に示すような選択とは異なる場合があります。

Now run the *Rasterize* algorithm to get the first raster layer, and then run the *Close gaps* algorithm to fill the no-data cells [Cell resolution: 100 m].

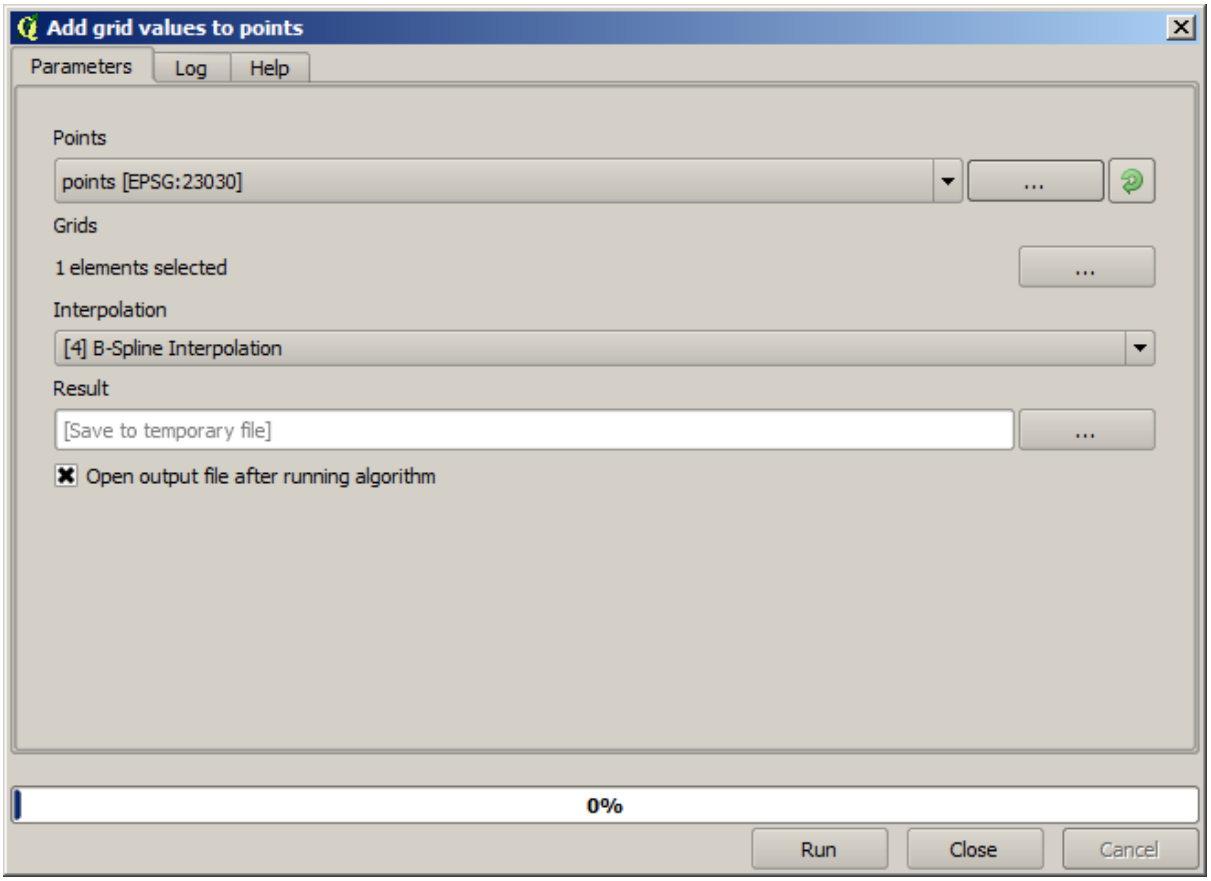


補間の品質を確認するために、今、選択されていないポイントを使用できます。この時点で、実際の標高（ポイントレイヤーにおける値）と補間標高（補間ラスターレイヤーの値を）知っています。これらの値の差を計算することにより、2 つを比較できます

選択されていないポイントを使用しようとしているので、まず、この選択を反転してみましょう。

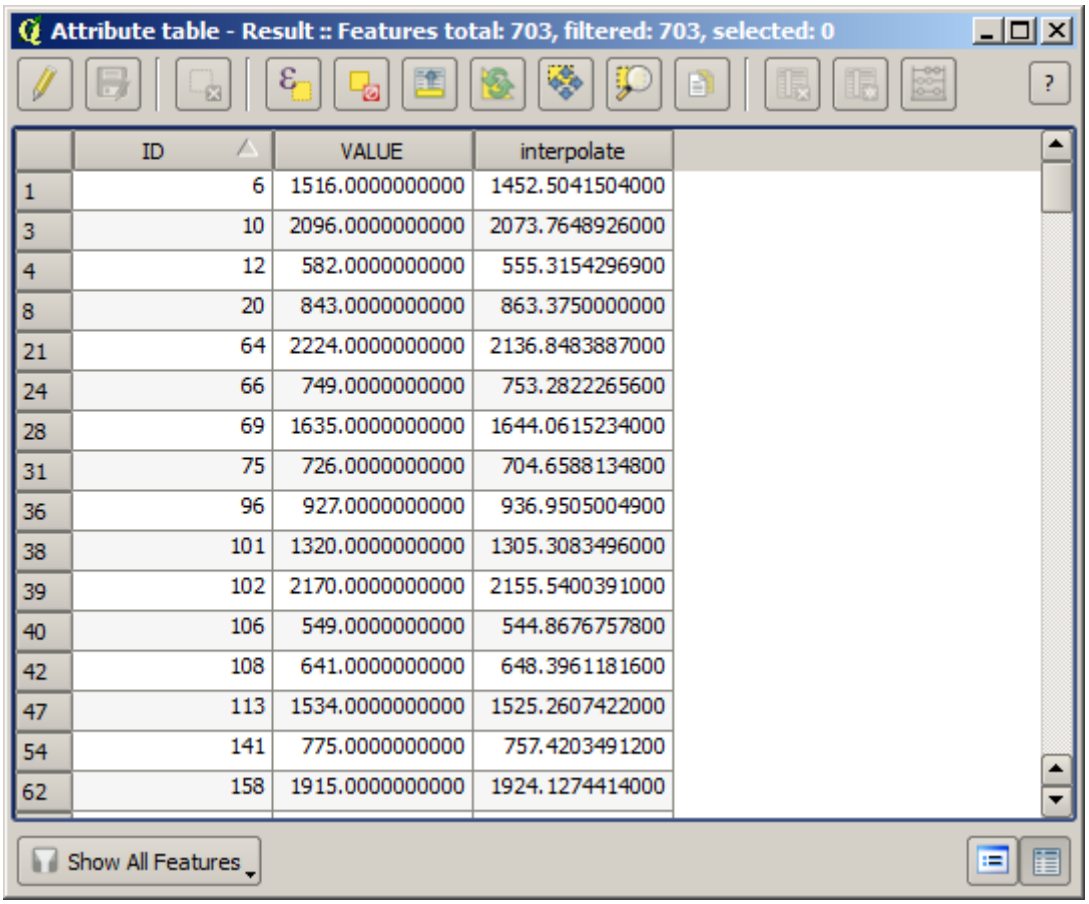


The points contain the original values, but not the interpolated ones. To add them in a new field, we can use the *Add raster values to points* algorithm



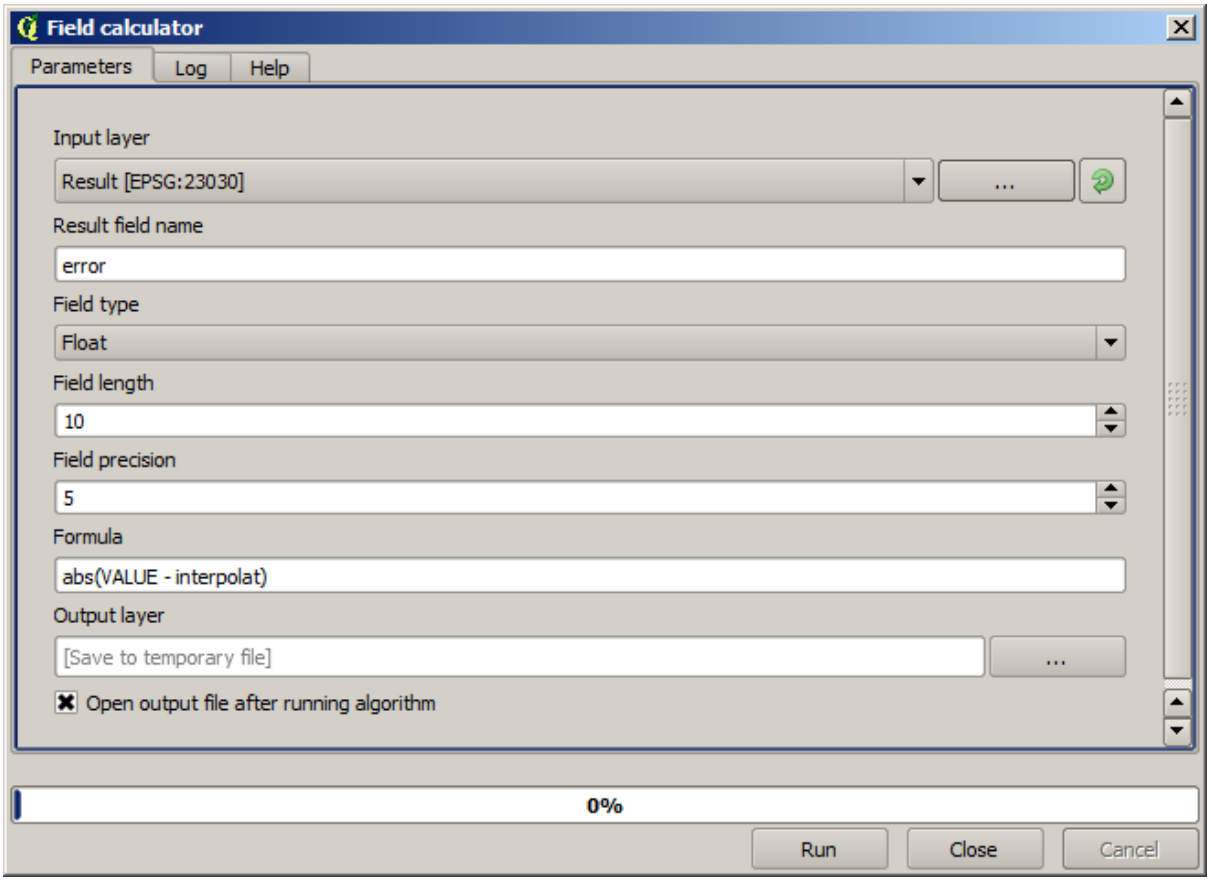
The raster layer to select (the algorithm supports multiple raster, but we just need one) is the resulting one from the interpolation. We have renamed it to `interpolate` and that layer name is the one that will be used for the name of the field to add.

今、補間のために使用されなかったポイントで、両方の値を含むベクターレイヤーを持っています。



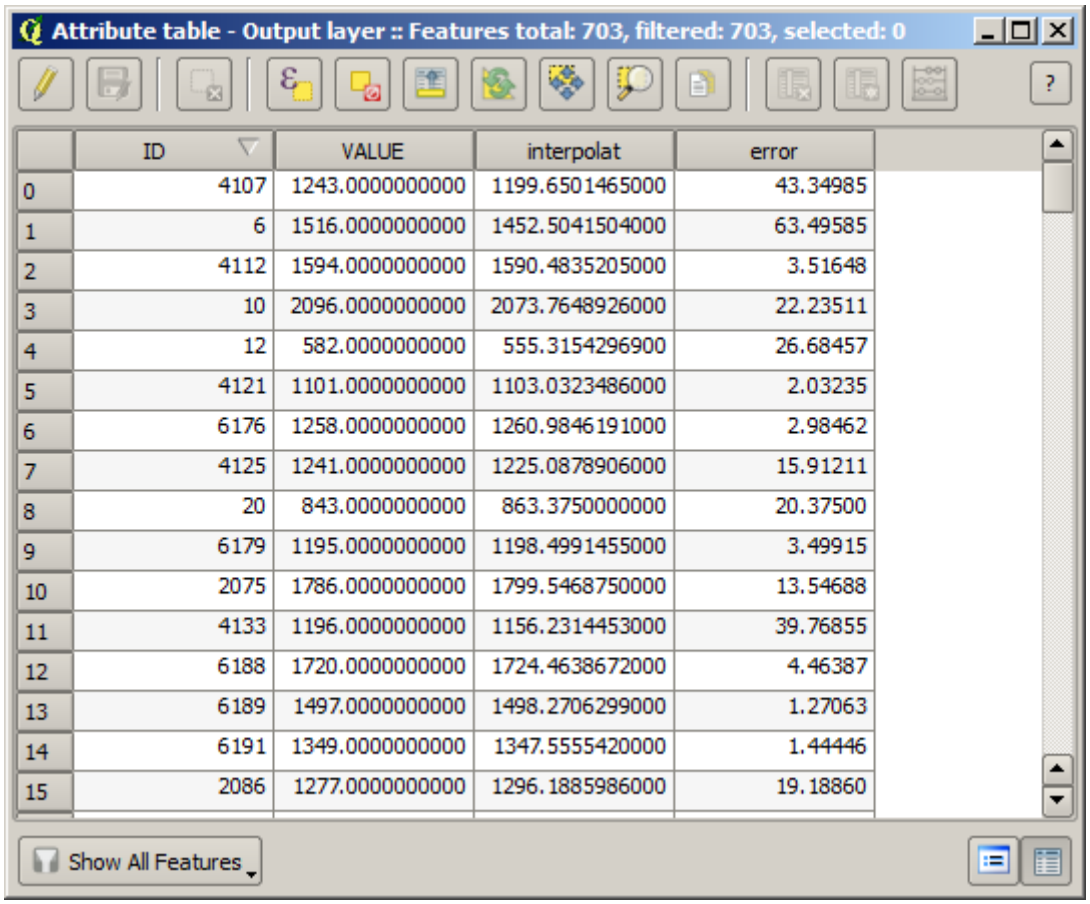
	ID	VALUE	interpolate
1	6	1516.0000000000	1452.5041504000
3	10	2096.0000000000	2073.7648926000
4	12	582.0000000000	555.3154296900
8	20	843.0000000000	863.3750000000
21	64	2224.0000000000	2136.8483887000
24	66	749.0000000000	753.2822265600
28	69	1635.0000000000	1644.0615234000
31	75	726.0000000000	704.6588134800
36	96	927.0000000000	936.9505004900
38	101	1320.0000000000	1305.3083496000
39	102	2170.0000000000	2155.5400391000
40	106	549.0000000000	544.8676757800
42	108	641.0000000000	648.3961181600
47	113	1534.0000000000	1525.2607422000
54	141	775.0000000000	757.4203491200
62	158	1915.0000000000	1924.1274414000

Now, we will use the fields calculator for this task. Open the *Field calculator* algorithm and run it with the following parameters.



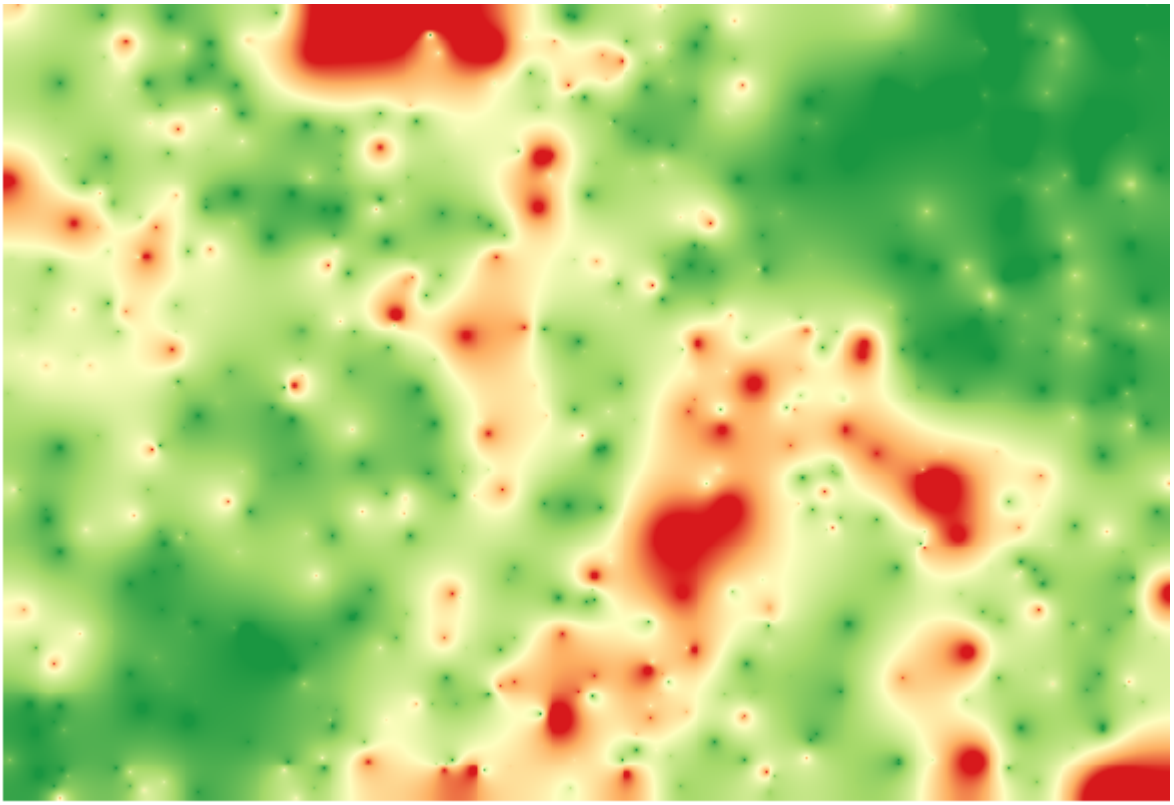
ラスターレイヤーからの値を使用して、フィールドには別の名前を持っている場合は、それに応じて上記の式を変更する必要があります。このアルゴリズムを実行すると、それらのそれぞれが 2 つの標高値の差を含む、補間のために使用していないだけでポイントを持つ新しいレイヤーが得られるでしょう。

その値に従ってそのレイヤーを表現すると最大の不一致がどこに見つかるかの最初のアイデアが得られるでしょう。



	ID	VALUE	interpolat	error
0	4107	1243.0000000000	1199.6501465000	43.34985
1	6	1516.0000000000	1452.5041504000	63.49585
2	4112	1594.0000000000	1590.4835205000	3.51648
3	10	2096.0000000000	2073.7648926000	22.23511
4	12	582.0000000000	555.3154296900	26.68457
5	4121	1101.0000000000	1103.0323486000	2.03235
6	6176	1258.0000000000	1260.9846191000	2.98462
7	4125	1241.0000000000	1225.0878906000	15.91211
8	20	843.0000000000	863.3750000000	20.37500
9	6179	1195.0000000000	1198.4991455000	3.49915
10	2075	1786.0000000000	1799.5468750000	13.54688
11	4133	1196.0000000000	1156.2314453000	39.76855
12	6188	1720.0000000000	1724.4638672000	4.46387
13	6189	1497.0000000000	1498.2706299000	1.27063
14	6191	1349.0000000000	1347.5555420000	1.44446
15	2086	1277.0000000000	1296.1885986000	19.18860

そのレイヤーを補間すると補間された地域のすべての点で推定誤差を持つラスターレイヤーを取得します。



GRASS *v.sample* で直接に同じ情報（元の点の値と補間のものとの差）を得ることもできます。

このレッスンの初めでランダムな選択を実行する際に導入されるランダム成分があるので、結果はこれらのものと異なる場合があります。

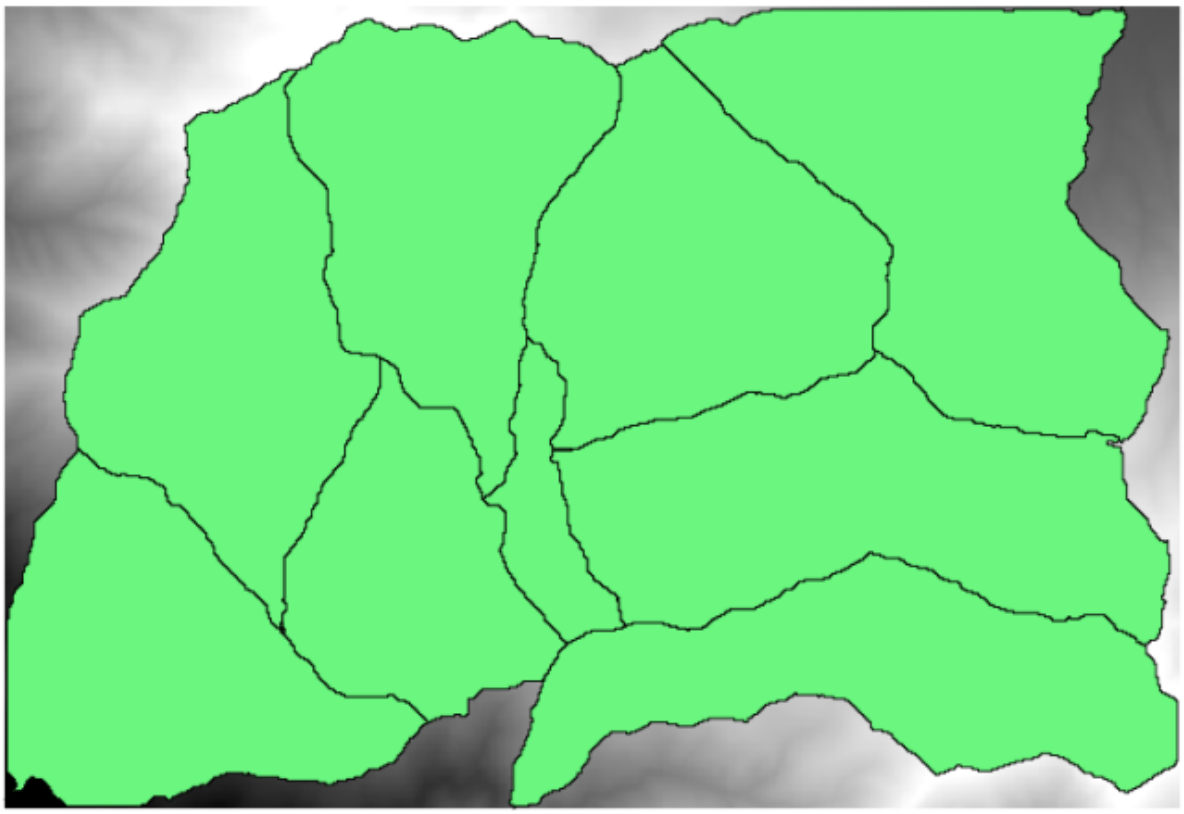
17.24 アルゴリズムの反復実行

注釈

このレッスンでは、アルゴリズムを繰り返し実行して入力ベクタレイヤの地物に対して反復適用するという、ベクタレイヤを使うアルゴリズムを実行する別の方法を示します。

モデルデザイナーが処理タスクを自動化する方法の一つであることは既に分かりました。しかし、状況によっては、モデラーが必要なタスクを自動化するための適切な方法ではないかもしれません。そのような状況のひとつで、異なる機能：アルゴリズムの反復適用を使った、それを簡単に解決する方法を見てみましょう。

この章に対応するデータを開きます。それは次のようになります。

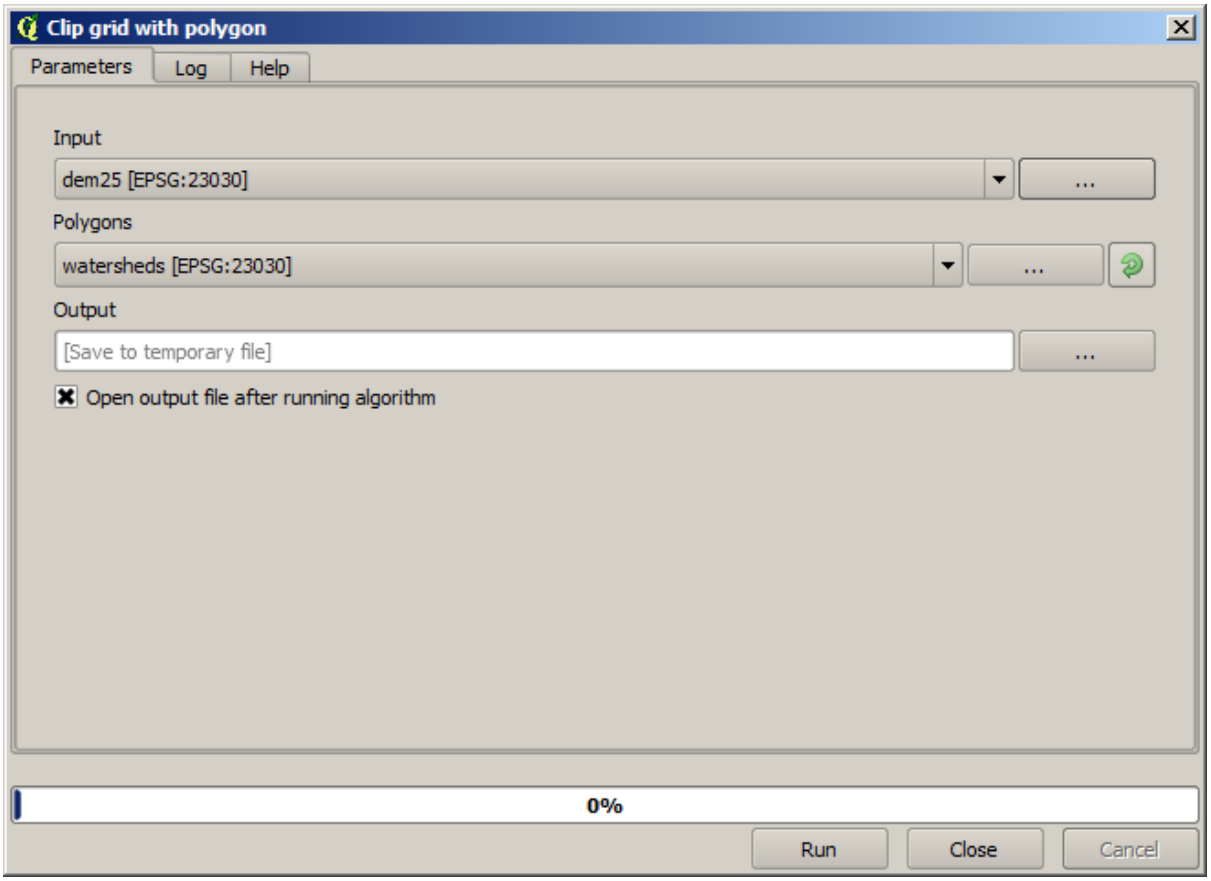


前の章でお馴染みの DEM と、それから抽出された一連の流域を認識することができます。DEM をいくつかの小さなレイヤに分割し、各レイヤに単一の流域に対応する標高データだけが含まれるようにする必要

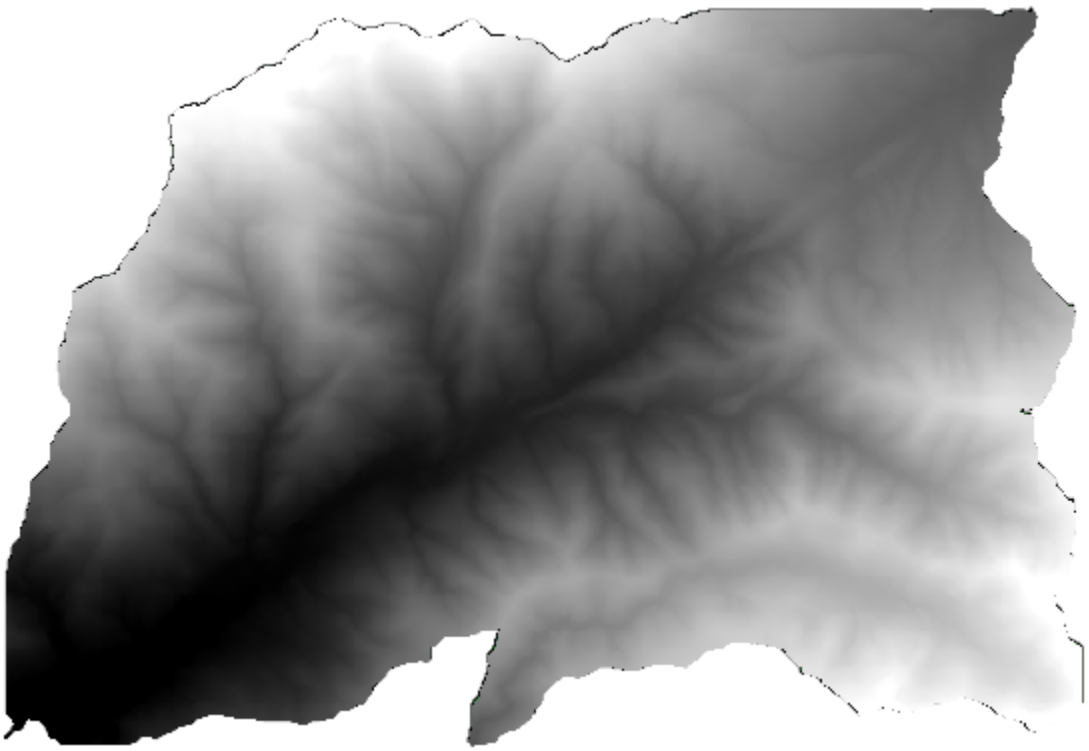
があると想像してください。それは後で各流域に関連する、平均標高や高低曲線などのパラメータを計算したい場合に便利です。

これは、流域の数が多い場合は特に、長くて退屈な作業になることがあります。しかし、それはこれからご覧になるように簡単に自動化できる作業です。

The algorithm to use for clipping a raster layer with a polygon layer is called *Clip raster with polygons*, and has the following parameters dialog.

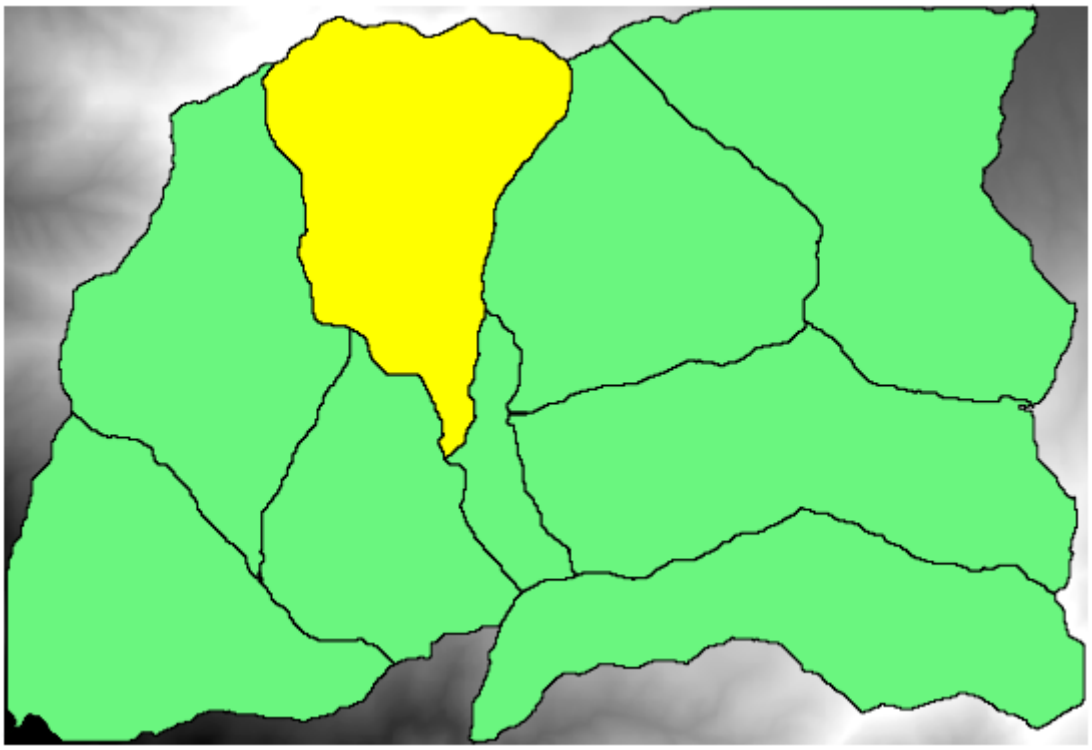


流域レイヤーと DEM を入力として使ってそれを実行できます。そして次の結果が得られます。

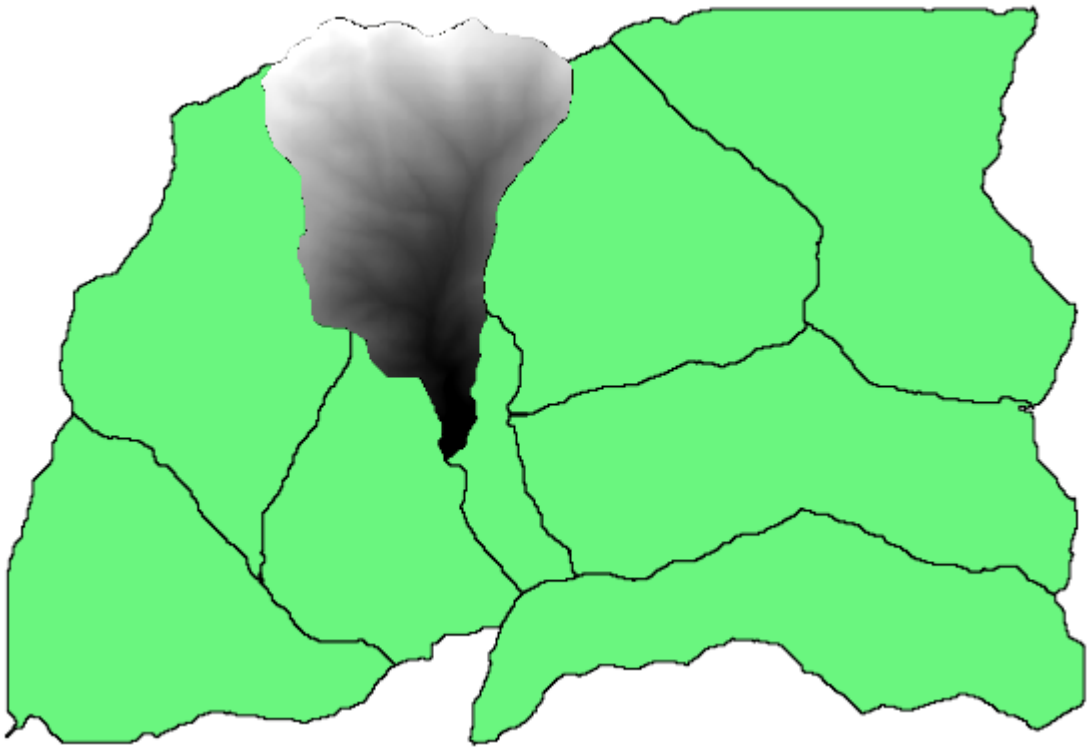


ご覧のように、すべての分水界ポリゴンで覆われた領域が使用されています。

望む流域を選択し、前に行ったアルゴリズムを実行することにより、1 つだけの流域で切り抜かれた DEM が得られます。

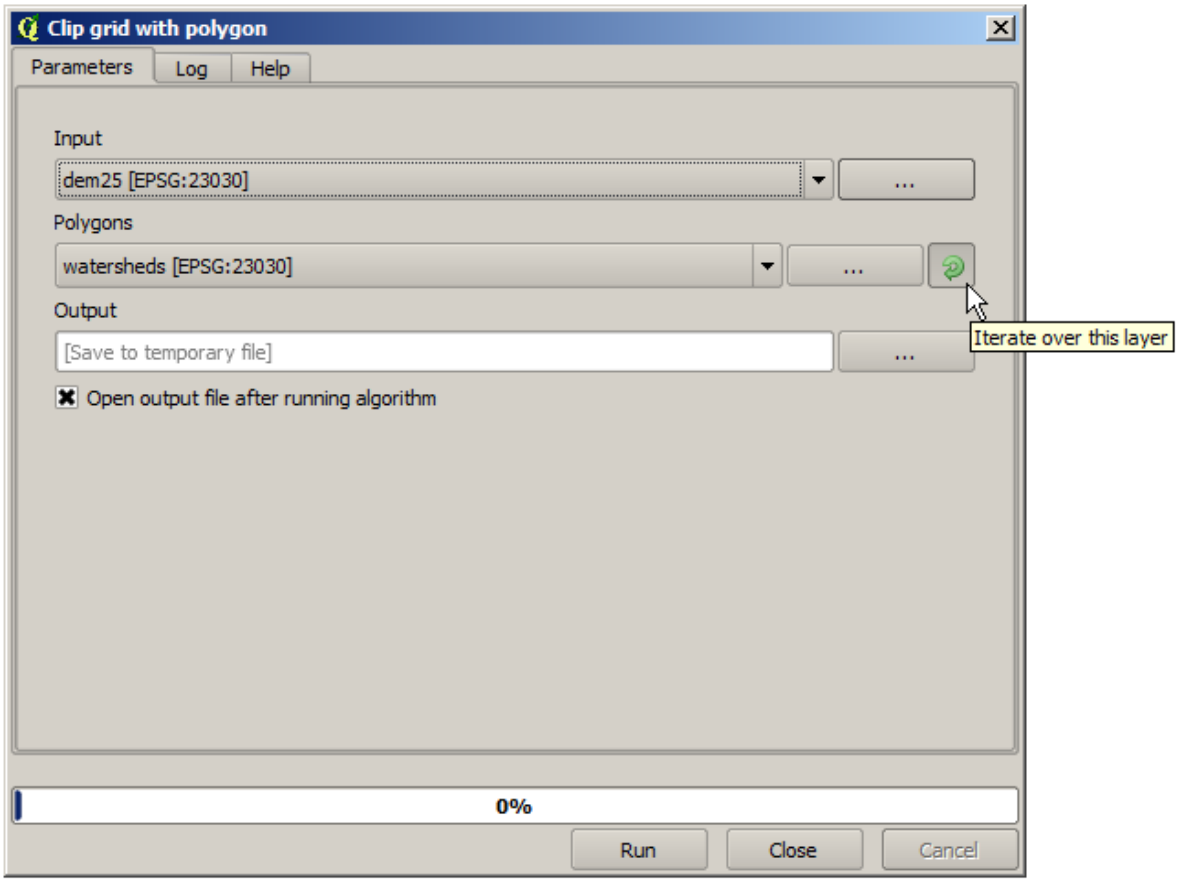


選択した地物だけが使われるので、選択されたポリゴンだけがラスタレイヤを切り抜くために使われます。



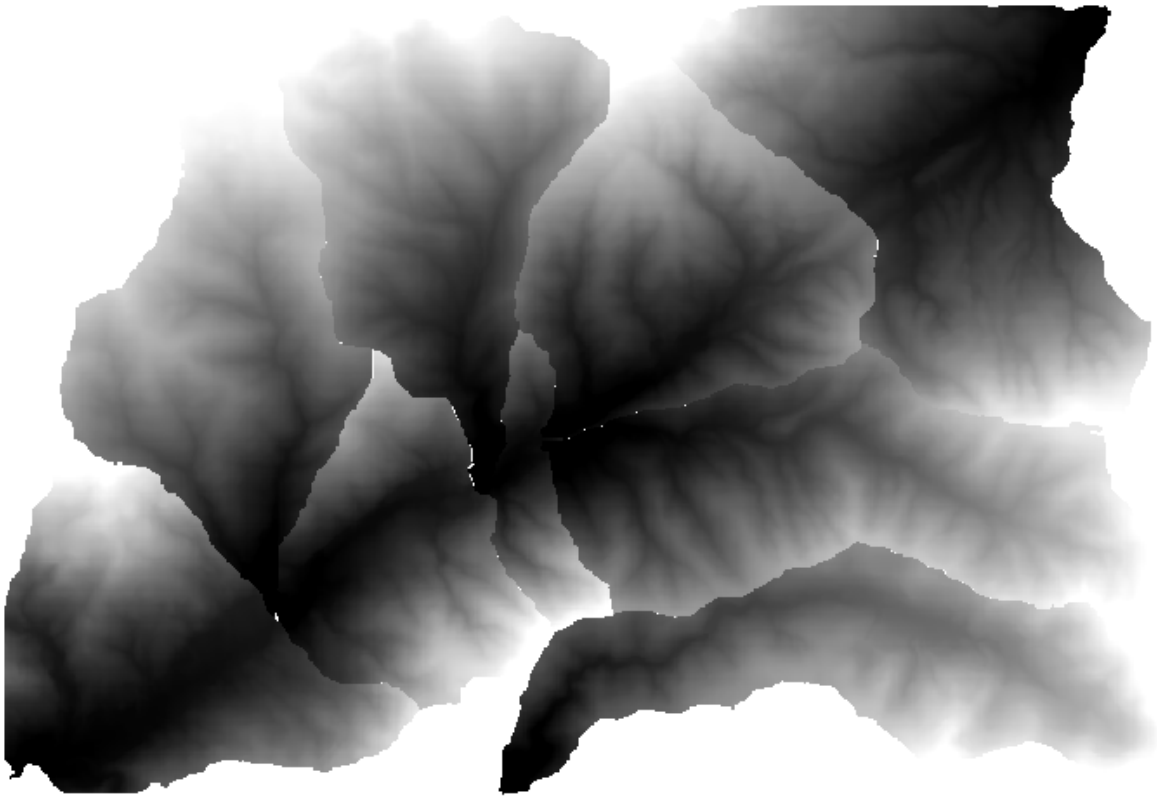
すべての流域についてこれを行えば、求めている結果が生成されますが、あまり現実的な方法とは思えません。その代わりに、その **選択と切り抜き** ルーチンを自動化する方法を見てみましょう。

First of all, remove the previous selection, so all polygons will be used again. Now open the *Clip raster with polygon* algorithm and select the same inputs as before, but this time click on the button that you will find in the right-hand side of the vector layer input where you have selected the watersheds layer.



このボタンを押すと、選択した入力レイヤはそこにある地物の数だけ分割され、ひとつのポリゴンを含んだレイヤに分割されます。そのため、アルゴリズムはそれぞれの単一ポリゴンレイヤに対して繰り返し呼び出されます。このアルゴリズムの場合、結果は 1 つのラスタレイヤではなく、アルゴリズムの実行ごとに対応する複数のラスタレイヤのセットになります。

こちらが説明したようにクリッピングアルゴリズムを実行した場合に得られる結果です。



各レイヤについて、黒と白のカラーパレット（または使用しているパレット）は、最小値から最大値まで異なるように調整されます。そのため、レイヤ間の境界線では、異なるピースと色が一致しないように見えますが、値は一致しています。

出力ファイル名を入力した場合、結果のファイルは、そのファイル名と接尾辞としてそれぞれの反復に対応する番号を使用して名前が付けられます。

17.25 アルゴリズムの反復実行（続）

注釈

このレッスンでは、より一層の自動化をするために、アルゴリズムの反復実行をモデラーと組み合わせる方法を示します。

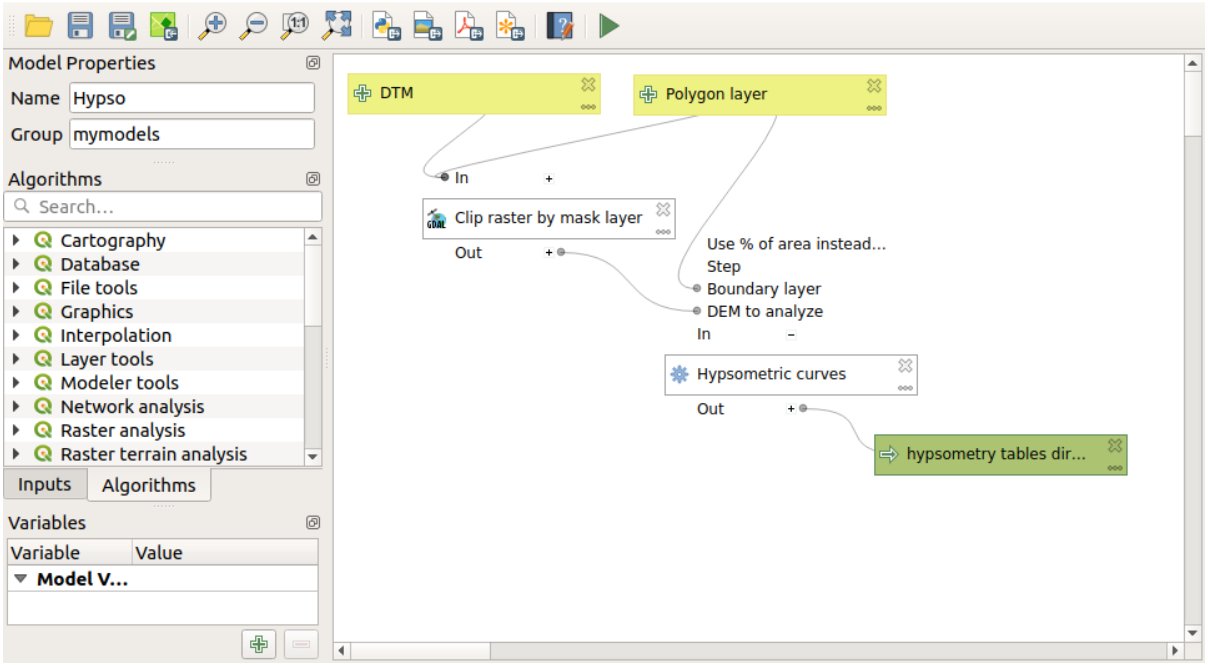
アルゴリズムの反復実行は、内蔵されたアルゴリズムに対してのみでなく、モデルなど、自作できるアルゴリズムに対しても使用可能です。より複雑な結果を簡単に得られるように、モデルとアルゴリズムの反復実行を組み合わせる方法を見ていきます。

このレッスンのために使用しようとしているデータは、すでに直前に使用したのと同じものです。この場合は、各分水界ポリゴンで DEM をクリッピングするだけでなく、いくつかの余分なステップを追加し、各々についてそれぞれの面積高度曲線を計算し、高度が流域内にどう分布するかを考察します。

いくつかのステップ（面積高度曲線を計算+クリッピング）を必要とするワークフローを持っているので、モデラーに移動し、そのワークフローの対応するモデルを作成する必要があります。

このレッスン用のデータフォルダ内に既に作成したモデルが見つかりますが、最初は自分で作成してみると良いでしょう。興味があるのは曲線にだけなので、この場合はクリップされたレイヤーは最終的な結果ではありません。だからこのモデルでは、何もレイヤーを作成せず、曲線データを有するテーブルを作成するだけです。

モデルは次のようになります。

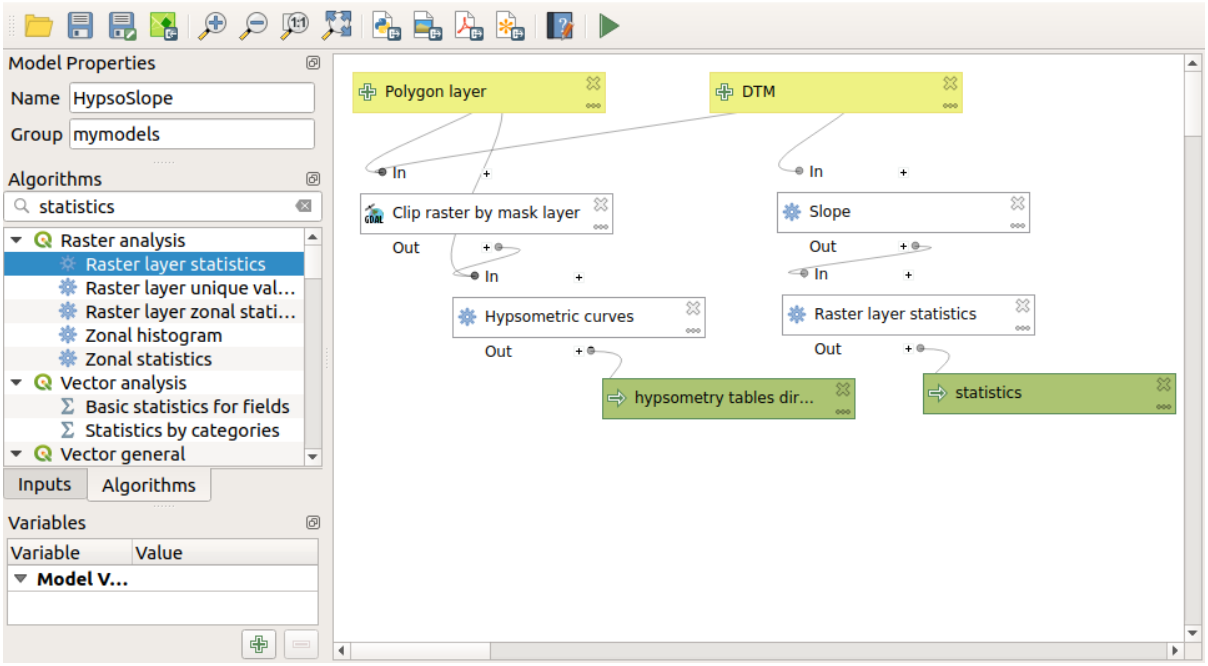


モデルフォルダにこのモデルを追加すると、ツールボックスで使用可能になるので、それを実行します。

DEM および流域を選択します。

アルゴリズムによりすべての盆地のテーブルが作成され、出力ディレクトリに置かれます。

We can make this example more complex by extending the model and computing some slope statistics. Add the *Slope* algorithm to the model, and then the *Raster statistics* algorithm, which should use the slope output as its only input.



ここでモデルを実行すると、テーブルと別に、統計が入ったページのセットが得られるでしょう。これらのページは結果ダイアログで利用できます。

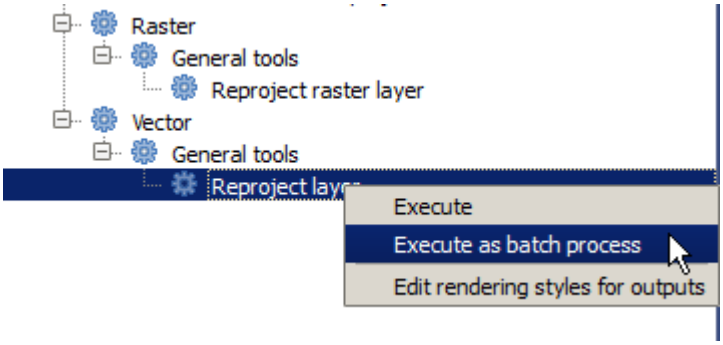
17.26 バッチ処理インターフェイス

注釈

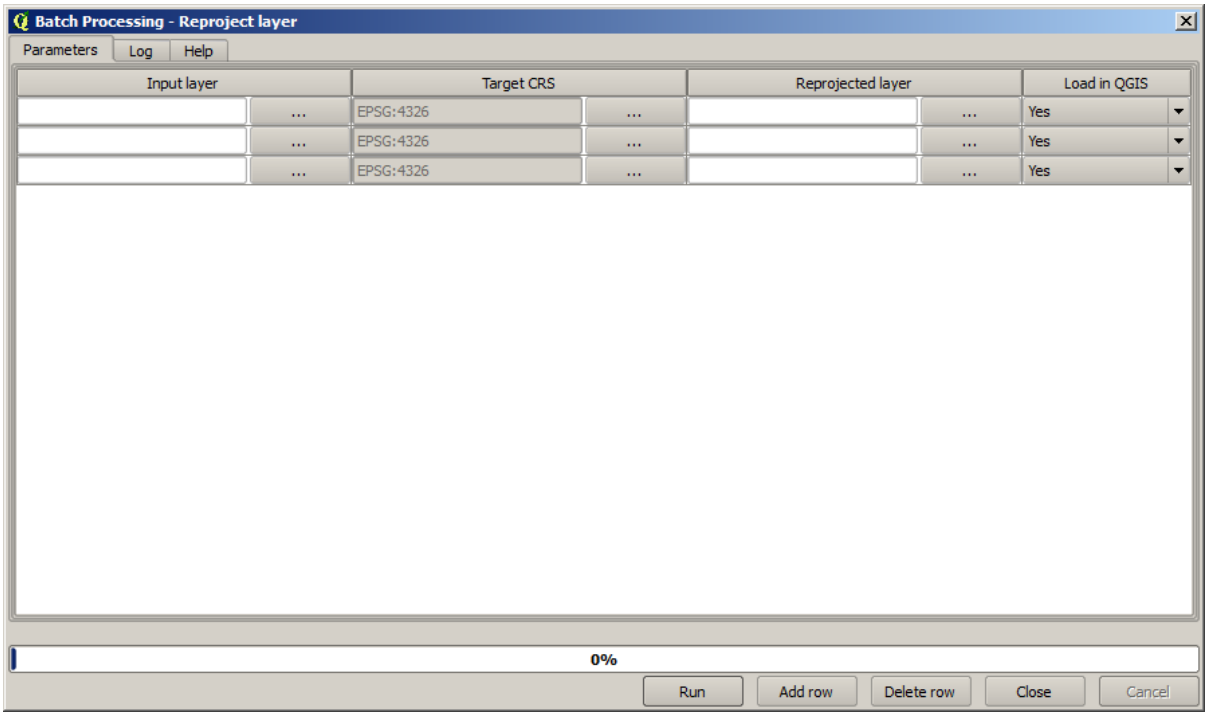
このレッスンではバッチ処理インターフェイスを紹介します。１つのアルゴリズムをさまざまな入力値のセットで実行できます。

あるアルゴリズムを異なる入力で繰り返し実行しなければならないことがあります。例えば、入力ファイルの集合をあるフォーマットから別のフォーマットに変換しなければならない場合や、ある投影法の複数のレイヤを別の投影法に変換しなければならない場合などです。

In that case, calling the algorithm repeatedly on the toolbox is not the best option. Instead, the batch processing interface should be used, which greatly simplifies performing a multiple execution of a given algorithm. To run an algorithm as a batch process, find it in the toolbox, and instead of double-clicking on it, right-click on it and select *Run as batch process*.



For this example, we will use the *Reproject layer* algorithm, so find it and do as described above. You will get to the following dialog.

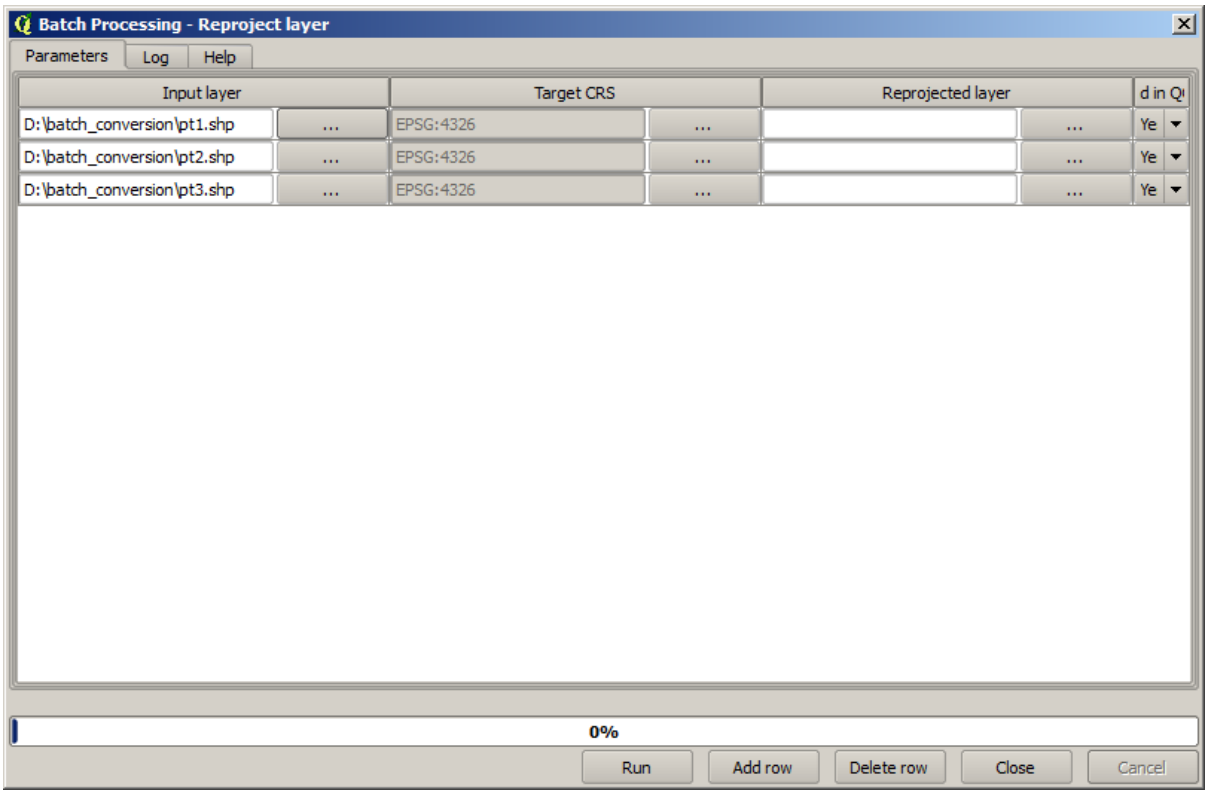


このレッスンのためのデータを見てみると、3 つのシェープファイルのセットは含むが、QGIS プロジェクトファイルが含まれていないことがわかります。アルゴリズムは、バッチプロセスとして実行されると、レイヤの入力は、現在の QGIS プロジェクトまたはファイルのいずれかから選択できるからです。それによって、例えば指定したフォルダ内のすべてのレイヤなど、大量のレイヤを簡単に処理できるようになります。

バッチ処理ダイアログのテーブルの各行は、アルゴリズムの単一の実行を表します。行のセルはアルゴリズムによって必要とされるパラメーターに対応します。これは、通常の単一実行ダイアログの中でのように互いに上下にではなく、その行に水平に配置されます。

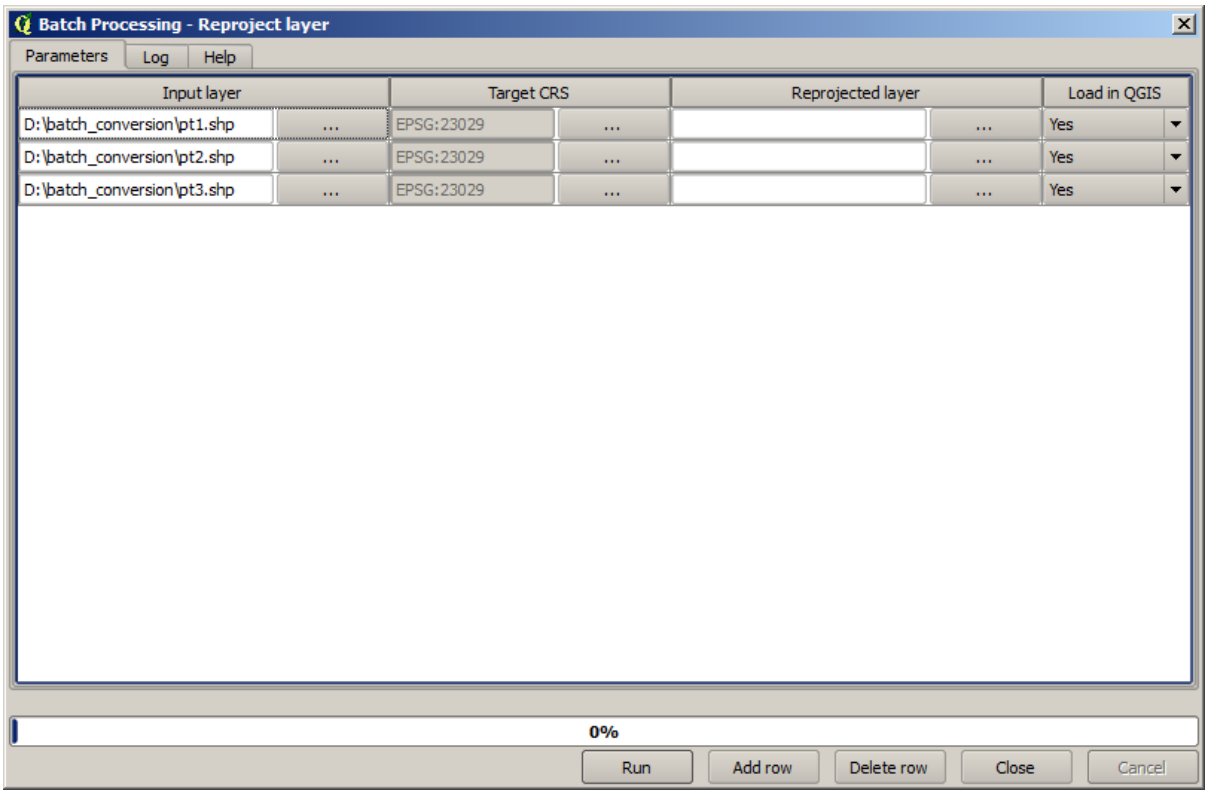
実行するバッチ処理を定義するには、テーブルに対応する値を入力します。ダイアログ自体には、この作業を容易にするためのいくつかのツールが含まれています。

Let's start filling the fields one by one. The first column to fill is the *Input layer* one. Instead of entering the names of each one of the layers we want to process, you can select all of them and let the dialog put one in each row. Click on the button in the upper-left cell, and in the file selection dialog that will popup, select the three files to reproject. Since only one of them is needed for each row, the remaining ones will be used to fill the rows underneath.



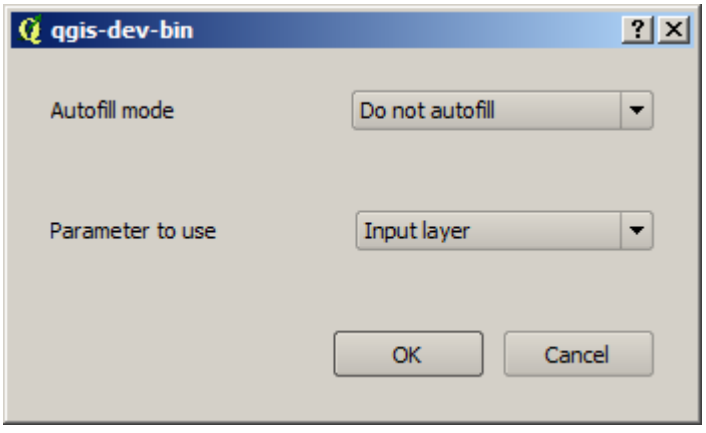
The default number of rows is 3, which is exactly the number of layers we have to convert, but if you select more layers, new rows will be added automatically. If you want to fill the entries manually, you can add more rows using the *Add row* button.

We are going to convert all those layers to the `EPSG:23029` CRS, so we have to select that CRS in the second field. We want the same on for all rows, but we do not have to do it for every single row. Instead, set that CRS for the first row (the one at the top) using the button in the corresponding cell, and then double click on the column header. That causes all the cells in the column to be filled using the value of the top cell.

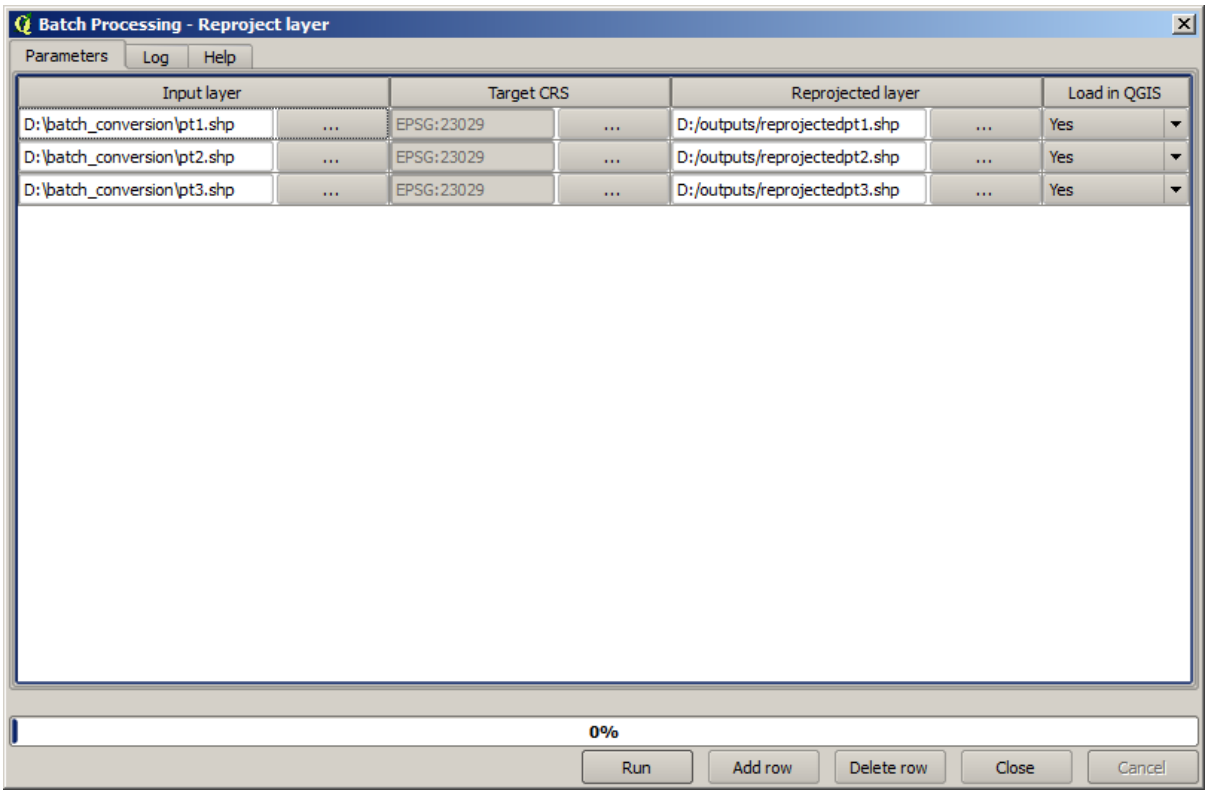


Finally, we have to select an output file for each execution, which will contain the corresponding reprojected layer. Once again, let's do it just for the first row. Click on the button in the upper cell and, in a folder where you want to put your output files, enter a filename (for instance, `reprojected.shp`).

Now, when you click *OK* on the file selection dialog, the file does not automatically gets written to the cell, but an input box like the following one is shown instead.



If you select the first option, only the current cell will be filled. If you select any of the other ones, all the rows below will be filled with a given pattern. In this case, we are going to select the *Fill with parameter value* option, and then the `Input Layer` value in the drop down menu below. That will cause the value in the `Input Layer` (that is, the layer name) to be added to the filename we have added, making each output filename different. The batch processing table should now look like this.



The last column sets whether or not to add the resulting layers to the current QGIS project. Leave the default `Yes` option, so you can see your results in this case.

Click on *OK* and the batch process will be run. If everything went fine, all your layers will have been processed, and 3 new layers would have been created.

17.27 バッチ プロセシング インタフェースのモデル

警告

注意してください。この章はよく検証されていないため、何か問題を見つけたらご報告ください；画像は欠けています

注釈

このレッスンではバッチプロセシングインタフェースのまた別の例を示しますが、今回は内蔵アルゴリズムでなくモデルを使用します

モデルは他のアルゴリズムとまったく同様で、バッチ処理インターフェイスで使用できます。それを実証するための、すでによく知られている水文モデルを使用して行うことができる簡単な例を出します。

モデルがツールボックスに追加した、その後、バッチモードで実行していることを確認します。これは、バッチ処理] ダイアログボックスがどのように見えるかです。

> **警告**
>
> TODO：画像を追加します。

行を合計 5 行になるまで追加します。それらのすべてについて、入力として、このレッスンに対応した DEM ファイルを選択します。それから、次に示すように 5 つの異なるしきい値を入力します。

> **警告**
>
> TODO：画像を追加します。

おわかりのようにバッチ処理インターフェイスは、異なるデータセットに同じプロセスを実行するだけでなく、異なるパラメーターで同じデータセットに同じプロセスを実行するために実行できます。

OK をクリックすると、指定された 5 つのしきい値に対応する流域を持つ 5 つの新しいレイヤーが得られます。

17.28 実行前後のスクリプトのフック

> **注釈**
>
> このレッスンでは、実行前および実行後のフックを使用して、実際の処理の前と後に追加の操作を実行できるようにする方法を示します。

実行前後のフックは、実際のデータ処理が実行される前と後に実行されるプロセッシングのスクリプトです。これは、アルゴリズムが実行されるたびに実行されるべき作業を自動化するために使用できます。

フックの構文は処理スクリプトの構文と同じです。詳細は QGIS ユーザーガイドの対応する 章 を参照してください。

すべてのスクリプト機能に加えて、フックでは `alg` という名前の特別なグローバル変数を使用できます、これはちょうど実行された（またはされようとしている）アルゴリズムを表します。

ここに実行後スクリプトの例があります。デフォルトでは、プロセッシングは一時ファイルでの解析結果を格納します。このスクリプトは、特定のディレクトリに出力をコピーしますので、それらは QGIS を閉じた後に削除されることはありません。

```
import os
import shutil
from processing.core.outputs import OutputVector, OutputRaster, OutputFile
```

(次のページに続く)

(前のページからの続き)

```
MY_DIRECTORY = '/home/alex/outputs'

for output in alg.outputs:
    if isinstance(output, (OutputVector, OutputRaster, OutputFile)):
        dirname = os.path.split(output.value)[0]
        shutil.copytree(dirname, MY_DIRECTORY)
```

例えば、ファイル名を抽出し、ファイルをコピーするような様々なファイルシステム操作のための shutil ---、パス操作のために--- os ：最初の 2 行では、必要な Python パッケージをインポートします。三行目では、処理の出力をインポートします。これは、このレッスンの後半でより詳細に説明します。

それから、分析結果をコピーしたいディレクトリへのパスである MY_DIRECTORY 定数を定義します。

スクリプトの終わりには、メインのフックのコードがあります。このループ中では、アルゴリズムの出力すべてについて反復処理し、この出力がファイルベースの出力でありコピーできるかどうかを確認します。そうである場合は、出力ファイルが置かれる最上位ディレクトリを決定し、そのディレクトリにすべてのファイルをコピーします。

このフックを有効にするには、［プロセッシング］オプションを開き、**一般** グループで *Post-execution script file* という名前のエントリを見つけて、そこにフックスクリプトのファイル名を指定します。指定されたフックは、各プロセッシングアルゴリズムの後に実行されます。

同様にして、実行前フックを実装できます。たとえば、幾何学的エラーがないかどうか入力ベクターをチェックするフックを作成してみましょう。

```
from qgis.core import QgsGeometry, QgsFeatureRequest
from processing.core.parameters import ParameterVector

for param in alg.parameters:
    if isinstance(param, ParameterVector):
        layer = processing.getObject(param.value)
        for f in layer.getFeatures(QgsFeatureRequest().setSubsetOfAttributes([])):
            errors = f.geometry().validateGeometry()
            if len(errors) > 0:
                progress.setInfo('One of the input vectors contains invalid␣
↪geometries!')
```

前の例のように、まず最初に必要な QGIS と処理パッケージをインポートします。

その後、すべてのアルゴリズムパラメーターを反復して ParameterVector パラメーターが見つかった場合、それから、対応するベクターレイヤーオブジェクトを取得します。レイヤーのすべての地物をループし、ジオメトリエラーのためにそれらを確認してください。少なくとも 1 つの地物に無効なジオメトリが含まれている場合、警告メッセージを出力します。

このフックを有効にするには処理の設定ダイアログのオプション **実行前スクリプトファイル** にそのファイル名を入力する必要があります。フックは、任意の処理アルゴリズムを実行する前に実行されます。

17.29 その他のプログラム

モジュール提供：Paolo Cavallini - Faunalia

> **注釈**
>
> この章では、処理の中から、追加のプログラムを使用する方法を示しています。それを完了するには、オペレーティング・システム、関連するパッケージのツールを使用して、インストールしている必要があります。

17.29.1 GRASS: Geographic Resources Analysis Support System、地理的資源分析支援システム

GRASS は、地理空間データの管理と分析、画像処理、グラフィックス、地図制作、空間的モデリング、および視覚化のための、フリーでオープンソースの GIS ソフトウェアスイートです。

これは OSGeo4W スタンドアロンインストーラ（32 ビットおよび 64 ビット）で Windows にデフォルトでインストールされ、そしてそれはすべての主要な Linux ディストリビューション用にパッケージされています。

17.29.2 R

R は、統計計算およびグラフィックス用の、フリーでオープンソースのソフトウェア環境です。

いくつかの必要なライブラリ (**LIST**) とともに、別途インストールする必要があります。QGIS で R を使用できるようにするには、*Processing R Provider* プラグインもインストールする必要があります。

プロセッシングの実装の美点は、ご自身のスクリプトを単純なものも複雑なものも追加できること、それらはその後、より複雑なワークフローにパイプされ、他のモジュールとして使用できること、などです。

R がすでにインストールされている場合（プロセッシングの一般的な設定から R モジュールをアクティブにすることを忘れないでください）、プリインストールされている例のいくつかをテストします。

17.29.3 他

LASTools はレーザー測量データを処理し分析するための、混合されたフリーの独自コマンドのセットです。さまざまなオペレーティングシステムで可用性が様々です。

その他のツールは、例えば追加プラグインを通じて入手できます。

- LecoS：土地被覆統計と景観生態学のスイート
- lwgeom：PostGIS の以前の一部は、このライブラリは、ジオメトリのクリーンアップのためのいくつかの便利なツールを提供します
- Animove：ツールは、動物の家の範囲を分析します。

さらに追加予定。

17.29.4 バックエンドの間での比較

バッファと距離

points.shp をロードしてツールボックスのフィルタで buf とタイプし、それからダブルクリックします：

- **一定距離バッファ**：距離は 10000
- **可変距離バッファ** ：距離フィールドは [サイズ]
- *v.buffer.distance* ：距離は 10000
- *v.buffer.column* ：bufcolumn SIZE
- **シェイプバッファ** ：固定値 10000（ディゾルブするおよびしない）、（縮尺付き）フィールド属性

速度がかなり異なっているか、さまざまなオプションが用意されていることを確認します。

読者のための練習 ：ジオメトリ出力での異なる方法間の差を見つけます。

さて、ラスターバッファとの距離：

- まず、 *GRASS* *v.to.rast.value* でベクター rivers.shp をロードしてラスター化します。 **注意：** セルサイズは 100 メートルに設定する必要があり、そうでなければ計算時間が膨大になります。結果の地図は 1 と NULL を持つことになります
- 同じく、 *SAGA* **シェイプからグリッド** *COUNT* で (結果の地図: 6 to 60)
- 次に、 *proximity*（GRASS の値= 1、SAGA のための河川 ID のリスト）、パラメーター 1000,2000,3000 で *r.buffer r.grow.distance* （２つの地図のうち１つめ；SAGA ラスターで行われる場合は、2 番目は各河川に関連する領域が表示されます）。

融合（dissolve）

共通の属性に基づいて地物をディゾルブします：

- *GRASS* *v.dissolve* municipalities.shp on PROVINCIA
- *QGIS* **ディゾルブ** municipalities.shp on PROVINCIA
- *OGR* **ディゾルブ** municipalities.shp on PROVINCIA
- *SAGA* - >**ポリゴンディゾルブ** municipalities.shp PROVINCIA 上（ **注： 内側境界を保持** を未選択にしてください）

> **注釈**
>
> 最後の一つは SAGA<= 2.10 では壊れています

読者のための練習 ：異なる方法での差（形状と属性）を見つけてください。

17.30 補間と等高線作成

モジュール提供：Paolo Cavallini - Faunalia

注釈

この章では、さまざまな補間を計算するために、異なるバックエンドを使用する方法を示しています。

17.30.1 内挿

このプロジェクトは、南から北へ、降雨量の勾配を示しています。補間のためのさまざまな方法を使用してみましょう、すべてベクター `points.shp` 、パラメーター `RAIN` に基づきます：

警告

Set cell size to `500` for all analyses.

- *GRASS* *v.surf.rst*
- *SAGA* **マルチレベル *B* スプライン補間**
- *SAGA* **逆距離荷重** [べきの逆距離。べき：4。検索半径：グローバル。検索範囲：すべてのポイント]
- *GDAL* **グリッド *(べきの逆距離)*** [べき:4]
- *GDAL* **グリッド *(移動平均)*** [Radius1&2: 50000]

次いで、方法間の分散を測定して点までの距離との相関をとります。

- *GRASS* *r.series* [Unselect Propagate NULLs, Aggregate operation: stddev]
- *GRASS* *v.to.rast.value* on `points.shp`
- *GDAL* **近接**
- *GRASS* *r.covar* で相関マトリックスを表示; 例えば http://vassarstats.net/rsig.html で相関の有意性をチェックします。

このように、ポイントから遠くの領域ではより正確でない補間になります。

17.30.2 等高線

stddev ラスターに等高線を描くための様々な方法 [always step = 10]：

- *GRASS* *r.contour.step*
- *GDAL* **等高線**
- *SAGA* **グリッドからの等高線** [**注意** ：いくつかの古い SAGA のバージョンでは、出力 shp は有効ではなく、既知のバグです]

17.31 ベクターの単純化と平滑化

Module contributed by Paolo Cavallini - Faunalia

注釈

この章では、ベクターを単純化し、鋭い角を滑らかにする方法を示します。

時にベクターの単純な版が必要になることがあります。小さいファイルサイズを持つように、不必要な詳細を取り除きます。多くのツールは非常に粗雑なやり方でこれを行うので、隣接関係を損なったり、多角形が位相的な正しくなくなったりすることもあります。GRASS は、このための理想的なツールです。位相的な GIS ですので、隣接関係および正しさは非常に高い単純化レベルであっても保たれています。私たちのケースでは、ラスターから得られたベクターがあり、これはしたがって境界で「のこぎり」パターンを示します。単純化を適用した結果は直線になります：

- *GRASS* *v.generalize* [最大許容値 30 M]

また、逆の操作を行い、レイヤーをより複雑にして鋭い角を滑らかにできます：

- *GRASS* *v.generalize* [方法：chaiken]

この 2 番目のコマンドを、元のベクターおよび最初の分析からのもの両方に適用してみて、違いを見てください。隣接関係が失われないことに注意してください。

この第 2 のオプションは、例えば、粗いラスターに起因する等高線、頂点が疎な GPS トラック、等に適用できます

17.32 太陽光発電所を計画する

モジュール提供：Paolo Cavallini - Faunalia

> **注釈**
>
> この章では、太陽光発電所を設置するのに適した地域を特定するために、いくつかの基準を使用する方法を示しています

まず第一に、DTM から傾斜方向図を作成します。

- *GRASS* *r.aspect* [データの型：int；セルサイズ：100]

GRASS において、向きは、反時計回りに東から出発して、度で計算されます。南向きの斜面だけを抽出するには（270 度+ - 45)、それを再分類できます：

- *GRASS* *r.reclass*

次のルールで:

```
225 thru 315 = 1 south
* = NULL
```

提供されるテキストファイル `reclass_south.txt` を使用できます。これらの単純なテキストファイルで、非常に複雑な再分類も作成できることに注意してください。

大きな発電所を構築したいので、連続した大きな (> 100 ヘクタール) 領域だけを選択します：

- *GRASS* *r.reclass.greater*

最後に、ベクターに変換します

- *GRASS* *r.to.vect* [地物の種類: エリア; 角を滑らかに: はい]

読者のための練習 ：GRASS コマンドを他のプログラムからの類似の物に置き換えて、分析を繰り返してください。

17.33 プロセシングで R スクリプトを使用する

このモジュールは、Matteo Ghetta が提供し、Scuola Superiore Sant'Anna が資金を提供しました

Processing (with the *Processing R Provider* plugin) makes it possible to write and run R scripts inside QGIS.

> **警告**
>
> コンピュータに R がインストールされ、PATH が正しく設定されている必要があります。さらにプロセシングは外部の R パッケージを呼び出すだけで、それをインストールすることはできません。ですか

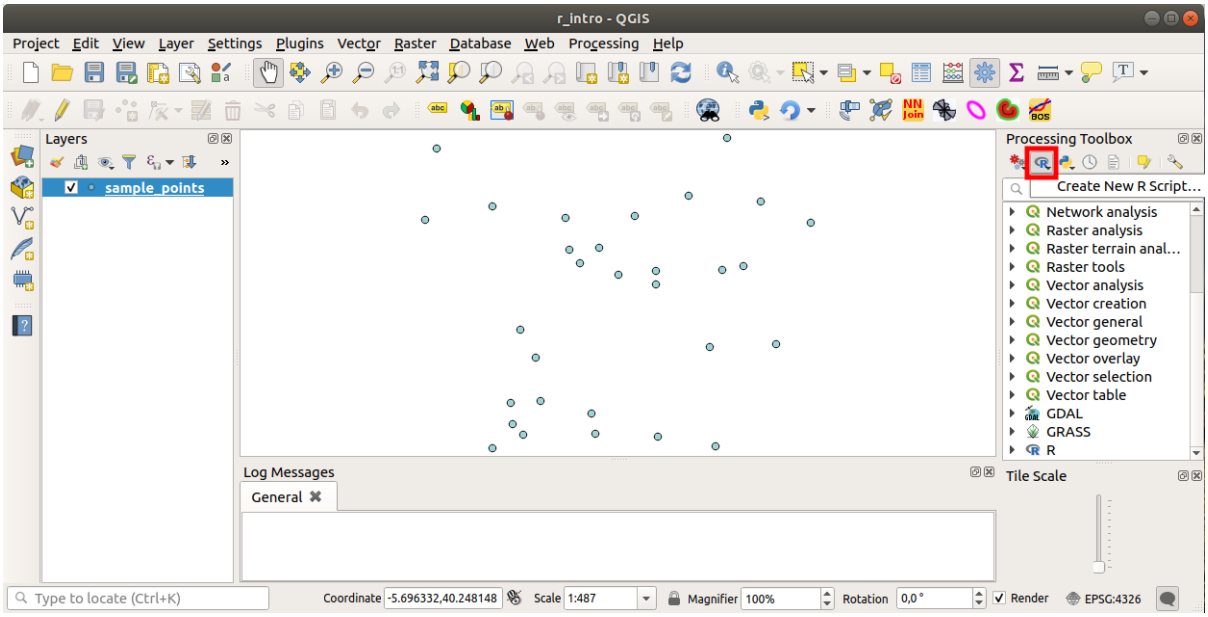
ら外部パッケージは必ず R で直接インストールするようにしてください。ユーザーマニュアルの関連する 章 を参照してください。

注釈

パッケージ の問題がある場合、それは `sp`, `rgdal`, `raster` などのプロセシングで必要な **必須** パッケージが欠けていることに関連しているかもしれません。

17.33.1 スクリプトを追加する

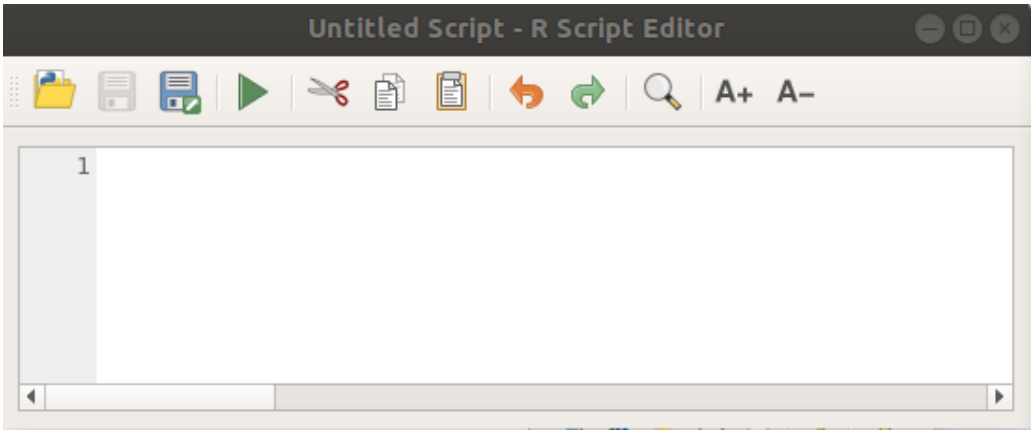
Adding a script is simple. The easiest way is to open the Processing toolbox and choose *Create new R script...* from the R menu (labelled with an R icon) at the top of the Processing Toolbox. You can also create the script in for instance a text editor and save it in your R scripts folder (`processing/rscripts`). When it has been saved there, it will be available for editing by right-clicking on the script name in the processing toolbox and then choose *Edit Script...*).



注釈

プロセシングの中に R が見あたらない場合は、**プロセシング** **オプション** **プロバイダ** を有効にする必要があります

It opens a *R Script Editor* in which you have to specify some parameters before you can add the script body.



17.33.2 プロットを作成する

このチュートリアルでは、ベクタレイヤフィールドの **箱ひげ図** を作成しようとしています。

`exercise_data/processing/r_intro/` フォルダの下にある `r_intro.qgs` QGIS プロジェクトを開きます。

スクリプトのパラメーター

エディタを開いて、それの最初に書き始めます。

スクリプト本体の **前** にいくつかのパラメーターを指定する **必要があります** :

1. スクリプトを置くグループの名前（この場合は *plots*）（グループが存在しない場合は作成されます）:

   ```
   ##plots=group
   ```

   スクリプトはプロセシングツールボックスの **plots** R グループ内にあります。

2. プロットを（この例に）表示したいことをプロセシングに伝える必要があります:

   ```
   ##showplots
   ```

   すると、**結果ビューア** パネルにプロットへのリンクが表示されます（**ビュー　パネル** と **プロセシング　結果ビューア** でオン/オフを切り替えることができます）。

3. また、入力データについてプロセシングに伝える必要があります。この例では、ベクタレイヤのフィールドからプロットを作成したいと思います:

   ```
   ##Layer=vector
   ```

   プロセシングは入力がベクトルであることが判りました。*Layer* という名前は重要ではなく、重要なのは **vector** というパラメータです。

4. 最後に、（上記で指定した名前 **レイヤ** を使って）ベクタレイヤの入力フィールドを指定します:

```
##X=Field Layer
```

これでプロセシングは、*Layer* というフィールドが必要で、それを **X** と呼ぶことを知りました。

5. また、name を使ってスクリプトの名前を定義することも可能です:

```
##My box plot script=name
```

定義されていない場合、ファイル名がスクリプトの名前として使用されます。

スクリプト本体

今、機能を追加できるスクリプトの **見出し** を設定し終わりました:

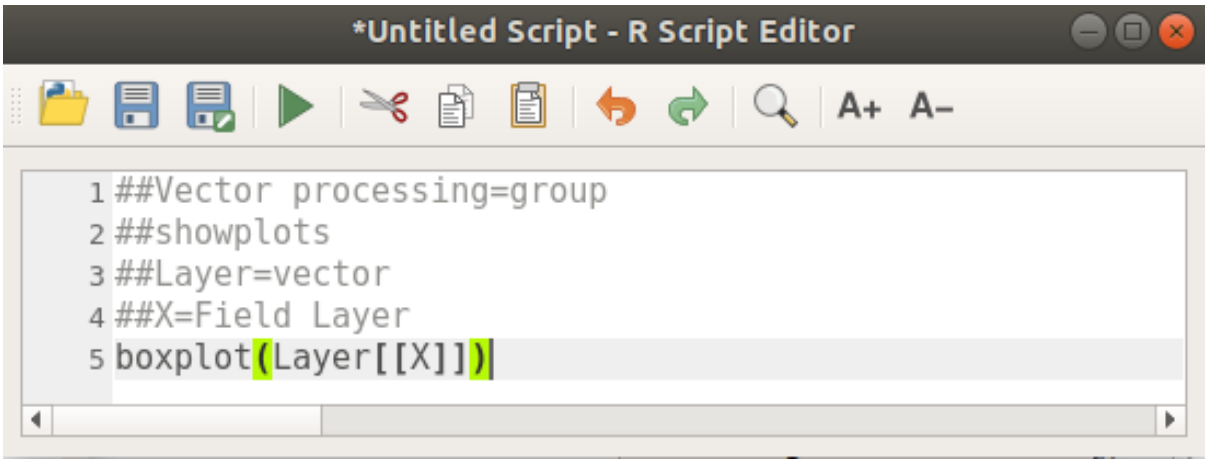
```
boxplot(Layer[[X]])
```

boxplot は R 関数の名前で、パラメータ **Layer** は入力データセットに定義した名前、**X** はそのデータセットのフィールドに定義した名前です。

> **警告**
>
> パラメータ **X** は二重の角括弧 ([[]]) の中に入れる必要があります。

最後のスクリプトは次のようになります:

```
##Vector processing=group
##showplots
##Layer=vector
##X=Field Layer
boxplot(Layer[[X]])
```

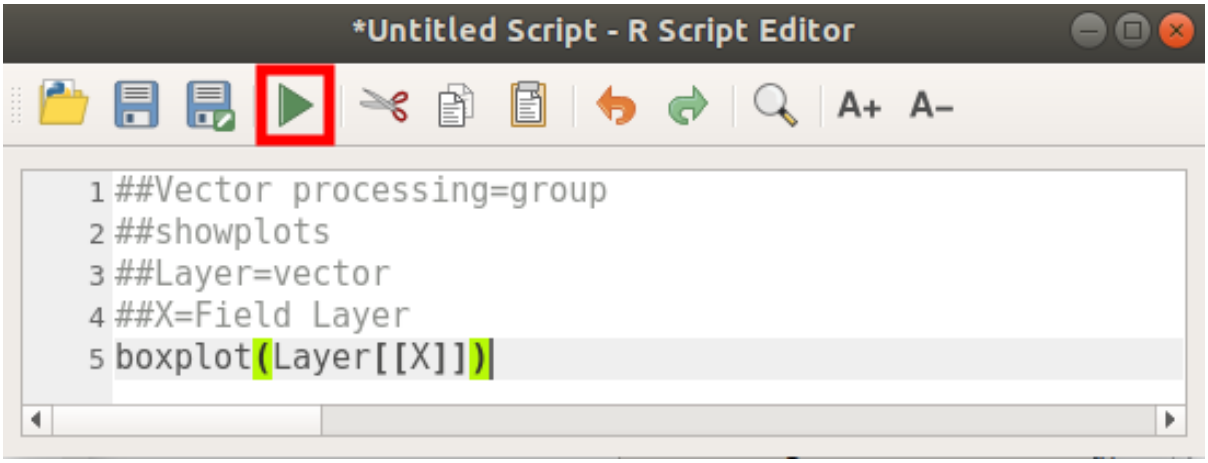


プロセシングが提案するデフォルトのパス (processing/rscripts) にスクリプトを保存します。スクリプトの見出しに name を定義していない場合、選択したファイル名がプロセシングツールボックスのスクリプト

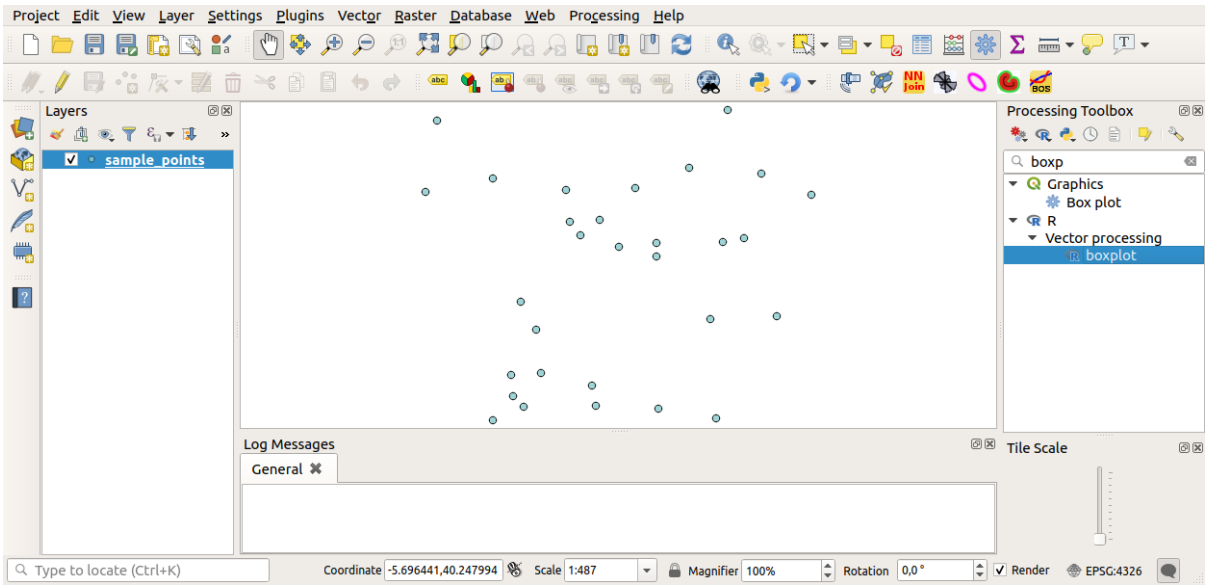
の名前となります。

> **注釈**
>
> スクリプトは好きな場所に保存できますが、その場合プロセシングでは自動的にプロセシングツールボックスに含めることができないので、手動でアップロードする必要があります。

今、エディタ・ウィンドウの上部にあるボタンを使用して、それを実行します。



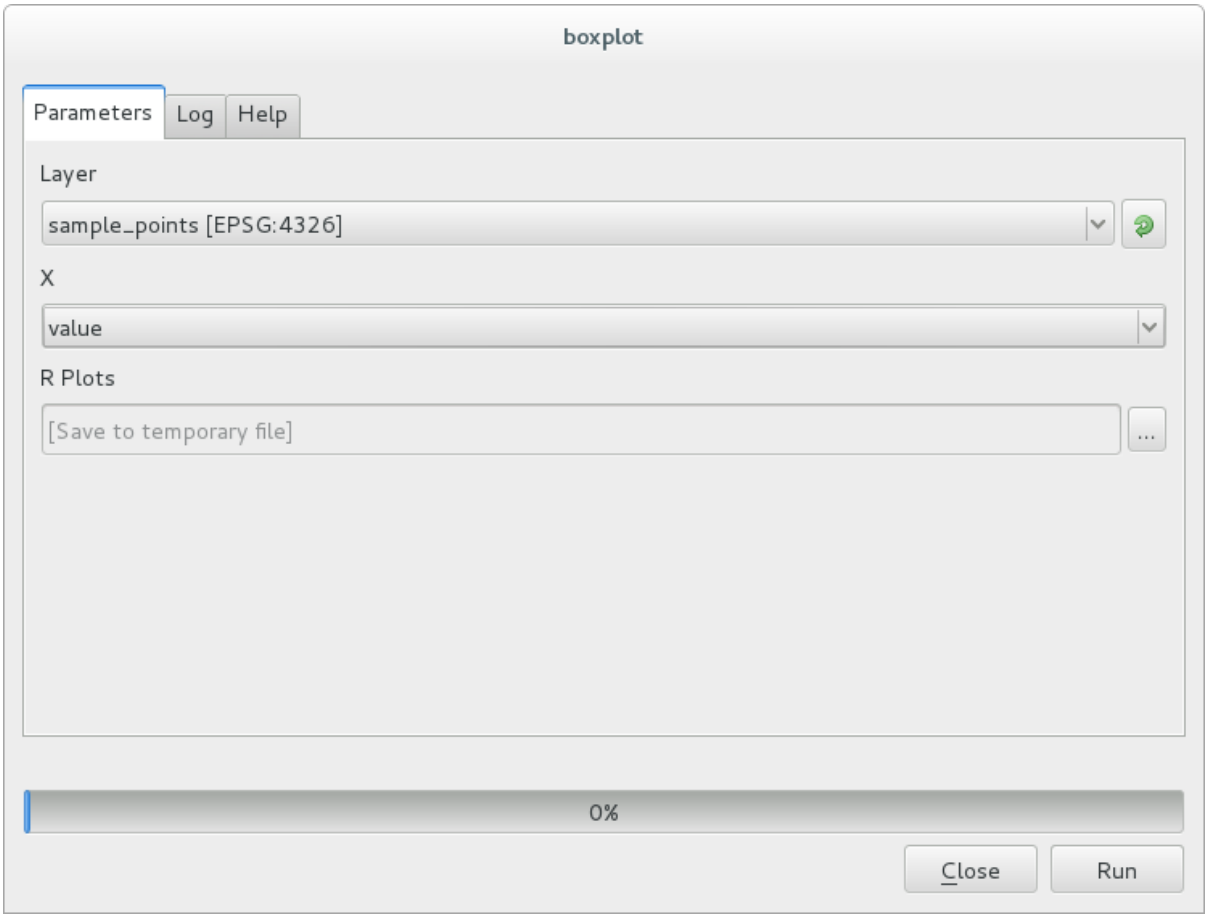
エディタウィンドウを閉じたら、プロセシングのテキストボックスを使って、スクリプトを探します:



これで、プロセシングアルゴリズムウィンドウに必要なパラメータを入力することができます:

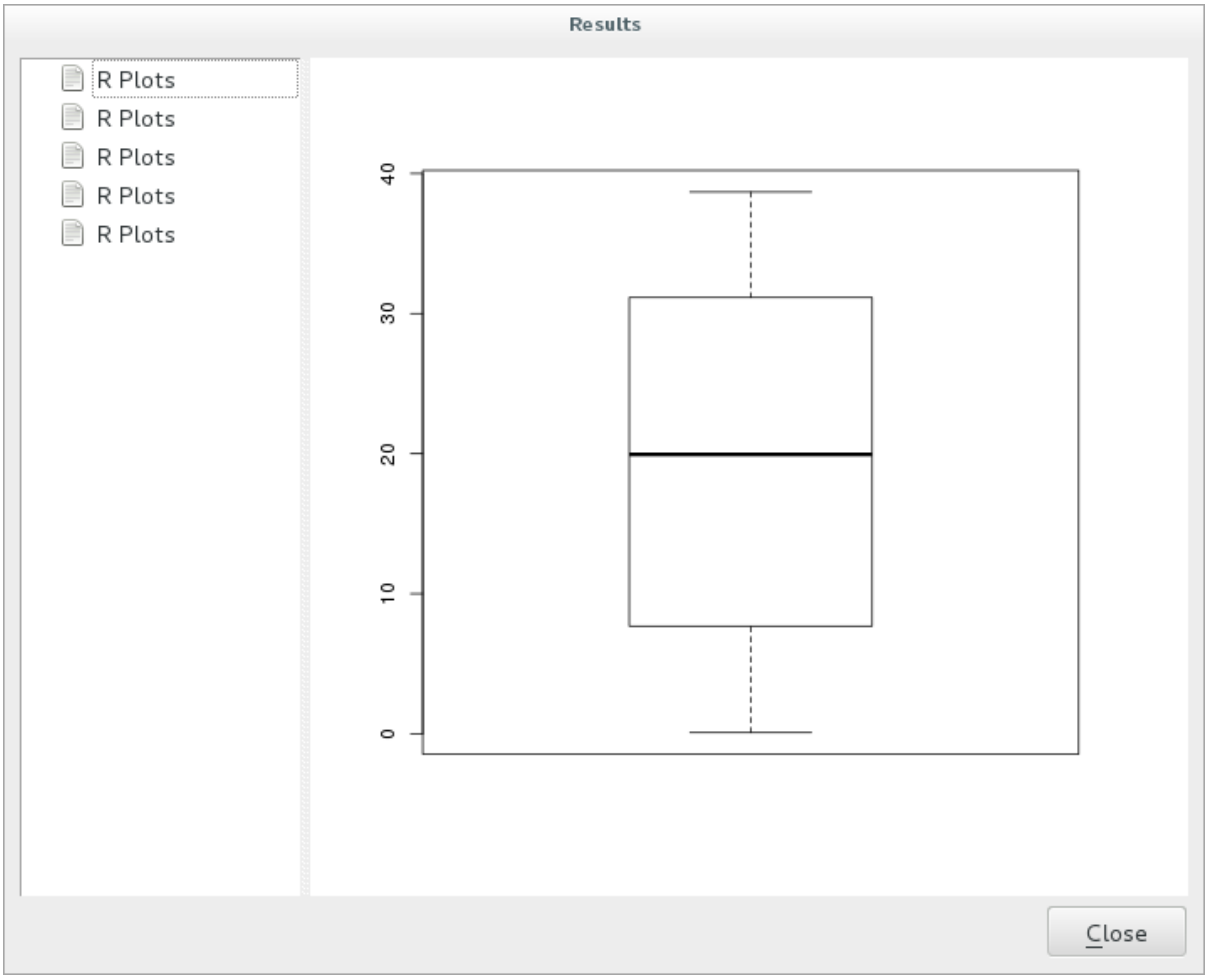
- for **Layer** choose `sample_points`
- for the **X** field choose `value`

Click on *Run*.



結果ウィンドウ が自動的に開くはずですが、そうでない場合は **プロセシング** **結果ビューア**... をクリックするだけです。

ビューアにあるリンクをクリックすると次が表示されます:



注釈

プロットで右クリックすることでその画像を開いたりコピーしたり保存することができます。

17.33.3 ベクタを作成する

また、ベクタレイヤを作成し、QGIS に自動的に読み込ませることもできます。

以下の例は、R スクリプトのオンラインコレクションにある `Random sampling grid` スクリプトから引用したものです（このオンラインコレクションのスクリプトは https://github.com/qgis/QGIS-Processing/tree/master/rscripts で見ることができます）。

この演習の目的は、`sp` パッケージの `spsample` 関数を使用して、範囲を制限する入力ベクタレイヤを使ってランダムポイントベクタレイヤを作成することです。

スクリプトのパラメーター

前と同じように、スクリプト本体の前にいくつかのパラメーターを設定する必要があります。

1. スクリプトを入れるグループ名（ここでは **ポイントパターン解析**）を指定します:

```
##Point pattern analysis=group
```

2. ランダムポイントの配置を制約する入力パラメータ（ベクタレイヤ）を定義します:

```
##Layer=vector
```

3. 作成するポイントの数を入力パラメータで設定します（Size、デフォルト値は ``10``）:

```
##Size=number 10
```

> **注釈**
>
> デフォルト値（10）が定義されているため、ユーザーはこの数値を変更するか数値のないパラメータのままにすることもできます。

4. （Output と呼ばれる）出力ベクタレイヤがあることを指定します:

```
##Output=output vector
```

スクリプト本体

今、関数の本体を追加できます。

1. spsample 関数を使います:

```
pts=spsample(Layer, Size, type="random")
```

この関数は、*Layer* を使用して、点の配置を制約します（線レイヤの場合、点はレイヤ内のいずれかの線上にある必要があり、ポリゴンレイヤの場合、点はポリゴン内にある必要があります）。点の数は *Size* パラメータで指定します。サンプリング方法は、*random* です。

2. 出力（``Output``パラメータ）を生成します:

```
Output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```

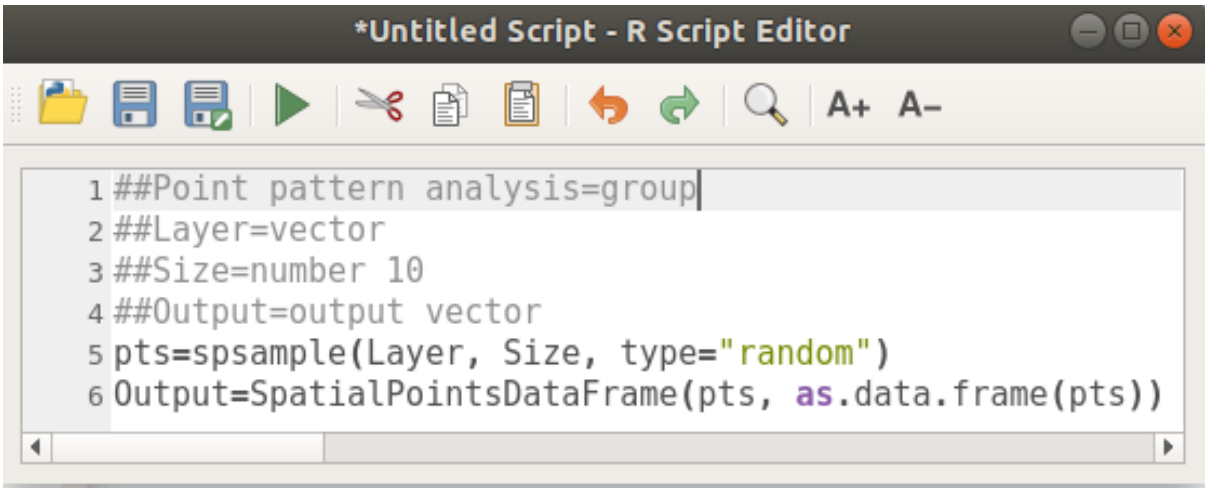
最後のスクリプトは次のようになります:

```
##Point pattern analysis=group
##Layer=vector
##Size=number 10
##Output=output vector
```

(次のページに続く)

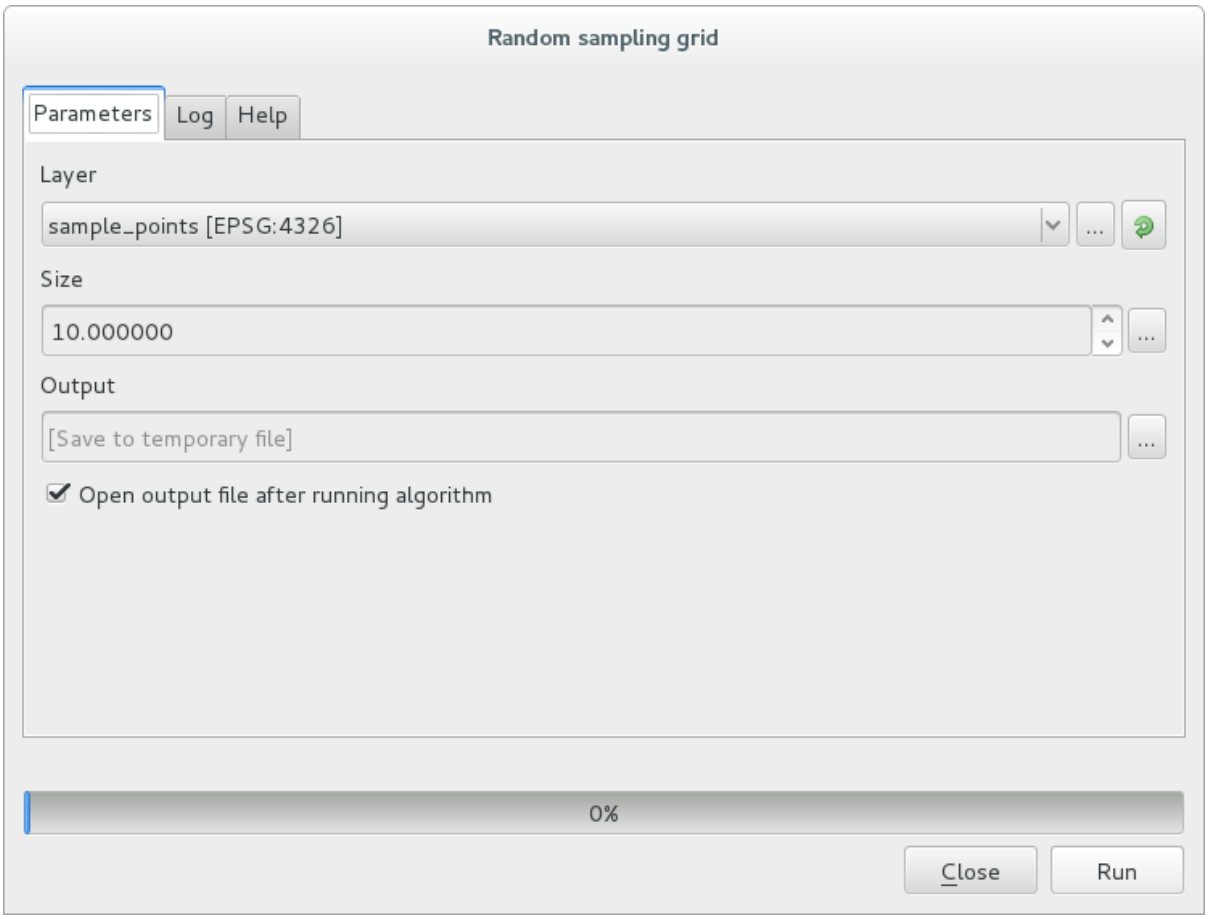
(前のページからの続き)

```
pts=spsample(Layer, Size, type="random")
Output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```



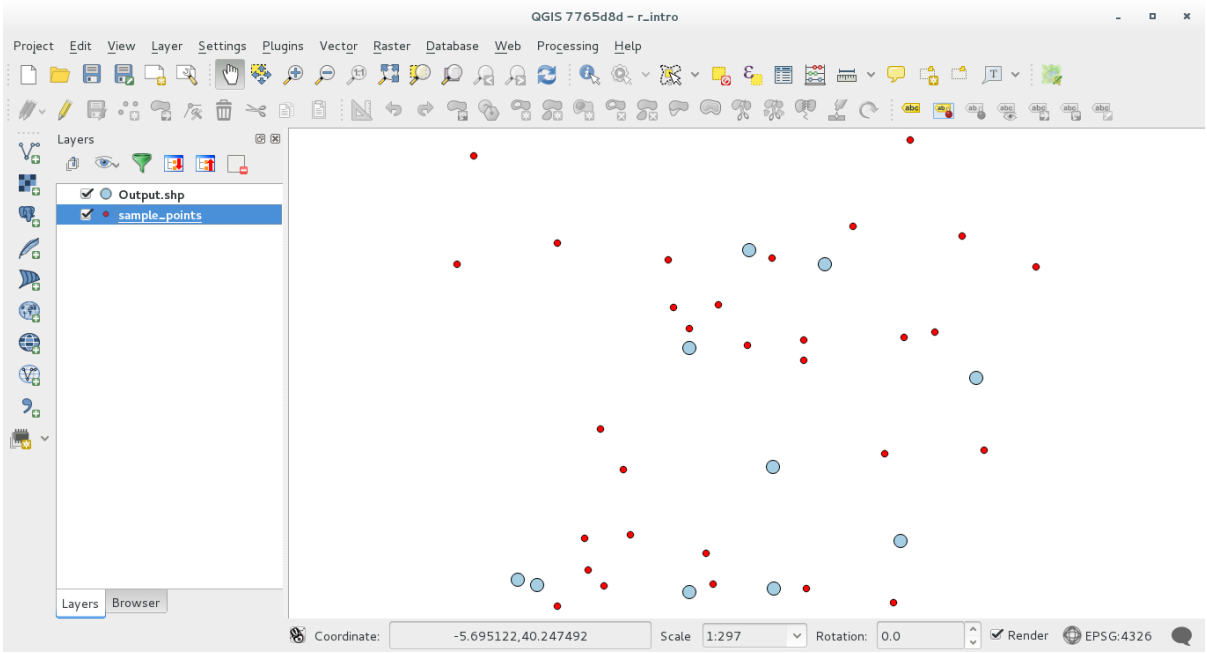
保存し、実行ボタンをクリックして実行します。

新しいウィンドウで正しいパラメーターを入力し：



そして実行をクリックしてください。

結果レイヤが目次に追加され、そのポイントがマップキャンバスに表示されます:



17.33.4 R からのテキスト・グラフ出力 - 文法

（``Processing R Provider plugin``を使った）プロセシング は、R から結果を取り出すために特別な文法を使用します:

- コマンドの前に > は、`>lillie.test(Layer[[Field]])` 中でのように、結果が R 出力（結果ビューア）に送られるべきことを意味します
- プロットの後に + を付けると、オーバーレイプロットを有効にします。例えば、`plot(Layer[[X]], Layer[[Y]]) + abline(h=mean(Layer[[X]]))` です。

17.34 地滑りを予測する

Module contributed by Paolo Cavallini - Faunalia

> **注釈**
>
> この章では、地滑りの可能性を予測するために過度に単純化したモデルを作成する方法を示します。

まず、傾斜を計算します（様々なバックエンドの中から選択します；興味のある読者は出力の差分を計算できます)。

- *GRASS* *r.slope*
- *SAGA* **傾斜、向き、曲率**

- *GDAL* **傾斜**

それから、気象衛星局での降雨値の補間に基づいて、予測雨量のモデルを作成します。

- *GRASS* *v.surf.rst*（解像度：500 メートル）

地すべりの確率は非常に大ざっぱには降雨と傾斜の両方に関連します（もちろん、現実のモデルは、より多くのレイヤー、および適切なパラメーターを使用します)、`(降雨*傾斜)/100` としましょう。

- *SAGA* **ラスター計算機** 降雨、傾斜: `(a*b)/100` (または: *GRASS* *r.mapcalc*)
- その後、予測される降雨の危険性が最大である自治体がどこか計算してみましょう： *SAGA* **ポリゴンでのラスター統計**（関心のパラメーターは **最大** と **平均**）。

第18章　モジュール: QGISで空間データベースを使う

In this module you will learn about how to use Spatial Databases with QGIS to manage, display and manipulate data in the database as well as performing analysis by querying. We will primarily use PostgreSQL (which was covered in previous sections), but the same concepts are applicable to other spatial database implementations including SpatiaLite.

18.1 レッスン: QGISブラウザでデータベースを操作する

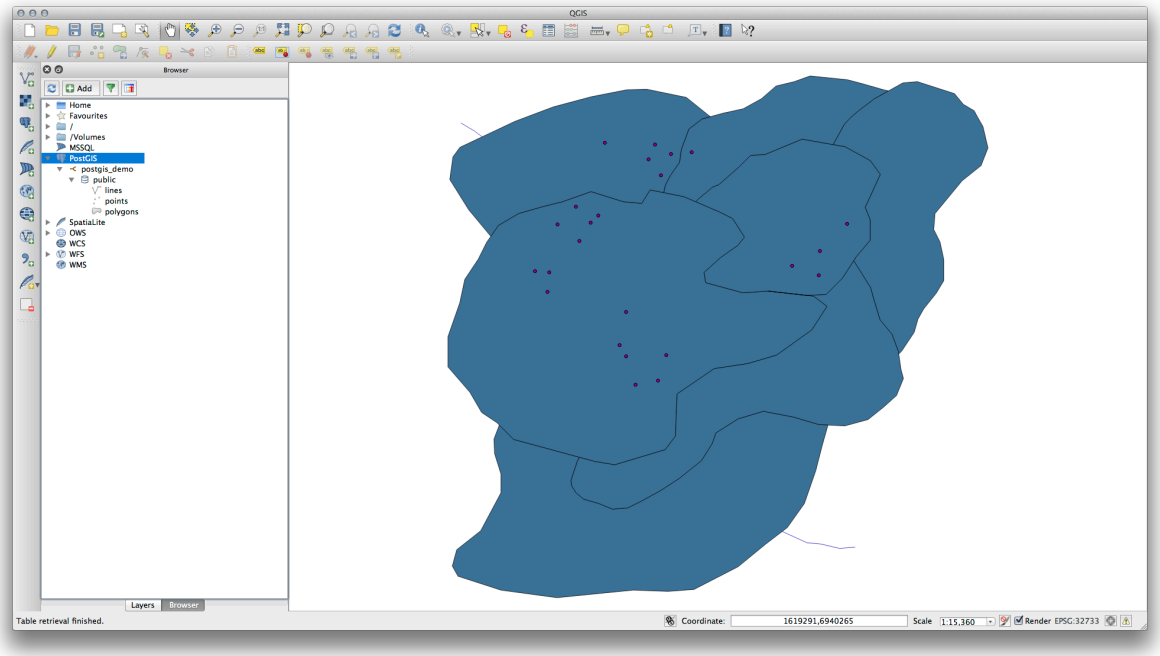
前の 2 つのモジュールでは、リレーショナルデータベースの基本的な概念、特徴および機能だけでなく、保存、管理、クエリとリレーショナルデータベース内の空間データを操作できる機能拡張について見ました。このセクションでは、QGIS で空間データベースを効果的に使用する方法について深掘りします。

このレッスンの目的： QGIS のブラウザのインタフェースを使って空間的なデータベースをどのように作用させるのかを学びます。

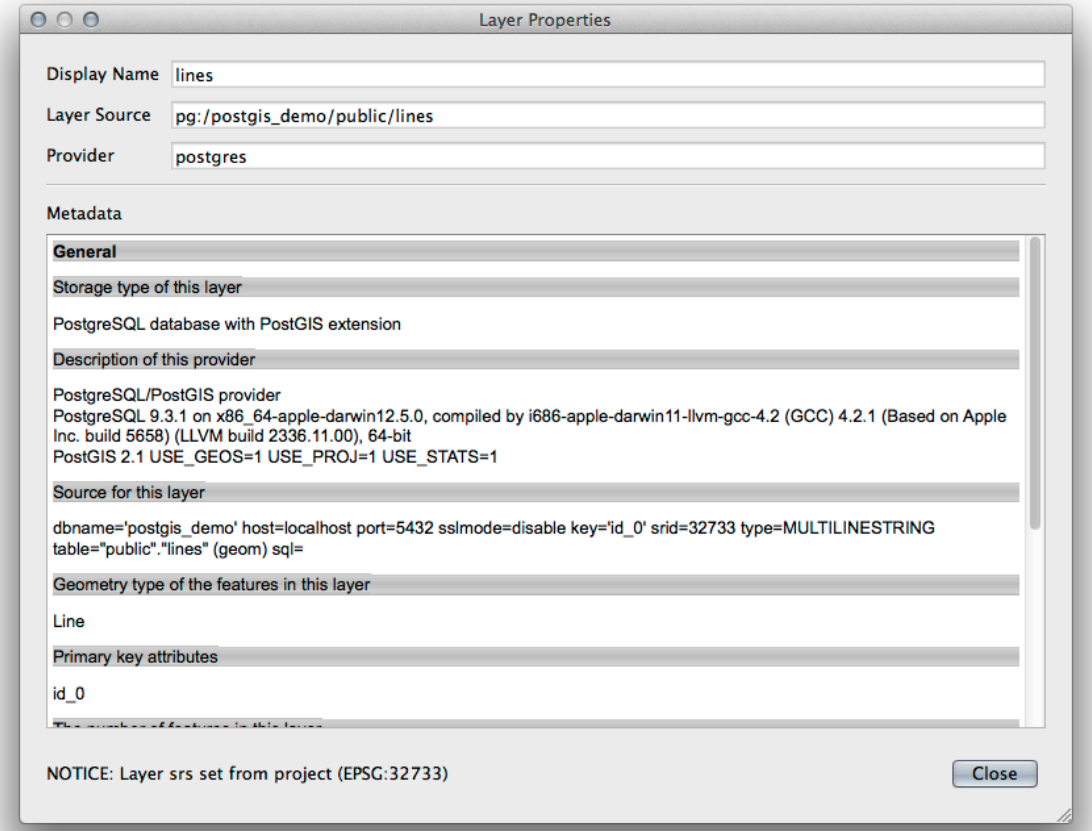
18.1.1 ★☆☆（初級レベル）理解しよう: ブラウザを使ってQGISにデータベーステーブルを追加する

すでに QGIS レイヤとしてデータベースからテーブルを追加する方法を簡単に見てきました。これをもう少し詳細に見て、これが QGIS で行うことができるさまざまな方法を見てみましょう。新しいブラウザ・インタフェースを見ることで開始できます。

- QGIS で新しい空の地図を開きます。
- Open the Browser by clicking the *Browser* tab. If you do not see this tab, you can enable it by going to *View* *Panels* *Browser*.
- Open the PostgreSQL portion of the tree and you should find your previously configured connection available (you may need to click the *Refresh* button at the top of the browser window).



- ここに記載されているテーブル/レイヤのいずれかをダブルクリックすると、それがマップキャンバスに追加されます。
- このビューでテーブル/レイヤを右クリックすると、いくつかのオプションを提供します。**プロパティ**項目をクリックしてレイヤのプロパティを見てください。



注釈

Of course you can also use this interface to connect to PostgreSQL databases hosted on a server external to your workstation. Right clicking on the PostgreSQL entry in the tree will allow you to specify connection parameters for a new connection.

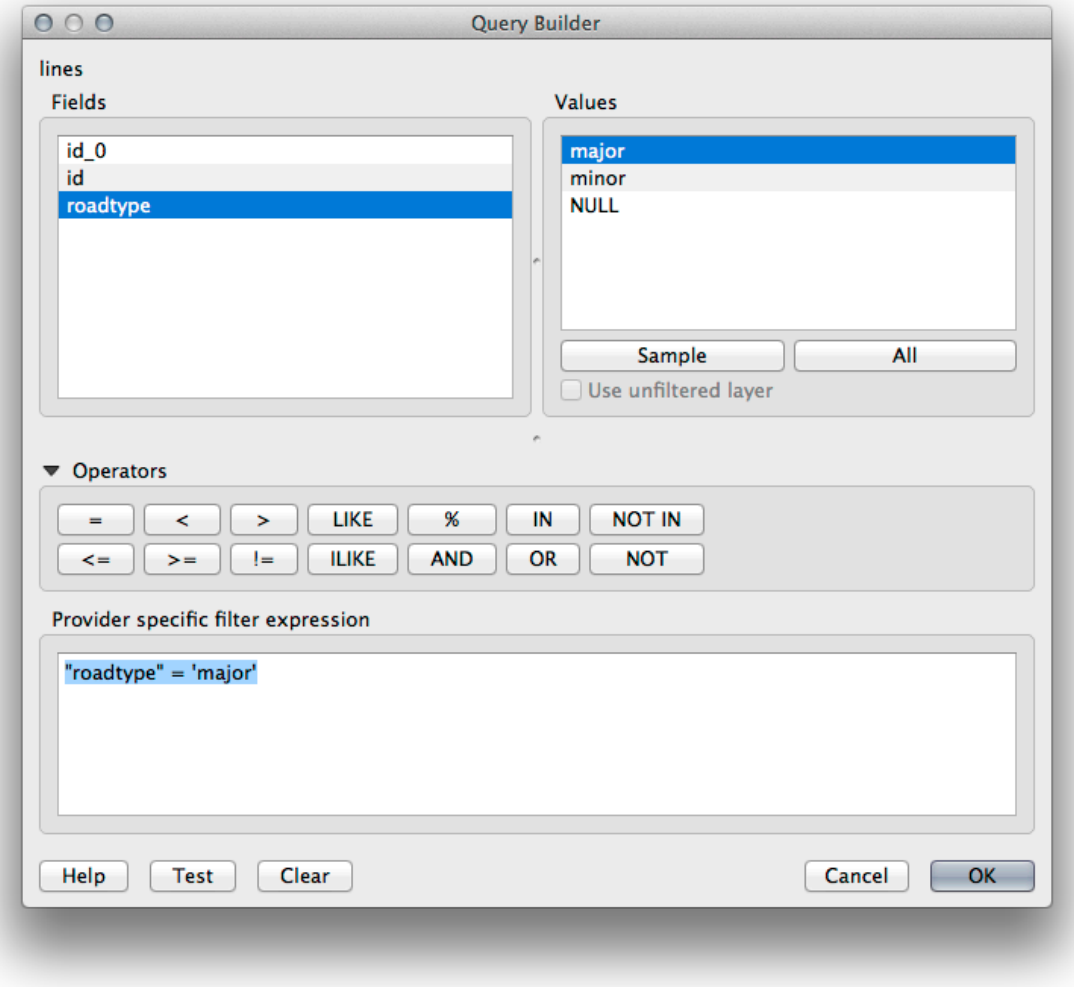
18.1.2 ★☆☆ （初級レベル） 理解しよう: レコードのフィルタセットをレイヤとして追加する

今、QGIS レイヤーとしてテーブル全体を追加する方法を見てきたので、前のセクションで学んだクエリを使用して、レイヤーとしてテーブルからレコードのフィルタセットを追加する方法を学ぶのはいいかもしれません。

- レイヤーなしで新しい空の地図を開始
- Click the *Add PostgreSQL Layers* button or select *Layer* *Add PostgreSQL Layers* from the menu.
- In the *Add PostgreSQL Table(s)* dialog that comes up, connect to the *postgis_demo* connection.
- Expand the *public schema* and you should find the three tables we were working with previously.

- Click the `lines` layer to select it, but instead of adding it, click the *Set Filter* button to bring up the *Query Builder* dialog.
- それを直接入力してボタンを使用するか、次の式を構築する:

```
"roadtype" = 'major'
```



- *OK* をクリックしてフィルタの編集を完了し、 **追加** をクリックして地図にフィルタレイヤーを追加します。
- Rename the `lines` layer in the tree to `roads_primary`.

地図にはレイヤー全体ではなく主要道路のみが追加されていることに気づくでしょう。

18.1.3 結論

QGIS ブラウザを使用して空間データベースと対話する方法、およびクエリフィルタに基づいて地図にレイヤーを追加する方法を見てきました。

18.1.4 次は?

次はデータベース管理作業のより完全なセットのために QGIS の DB マネージャインターフェイスで作業する方法について説明します。

18.2 レッスン: DB マネージャを使用して QGIS で空間データベースと連携する

すでに QGIS や他のツールで多くのデータベース操作を実行する方法を見てきましたが、今度は、この同じ機能の多くに加えてさらに管理指向のツールを提供する DB 管理者ツールを検討します。

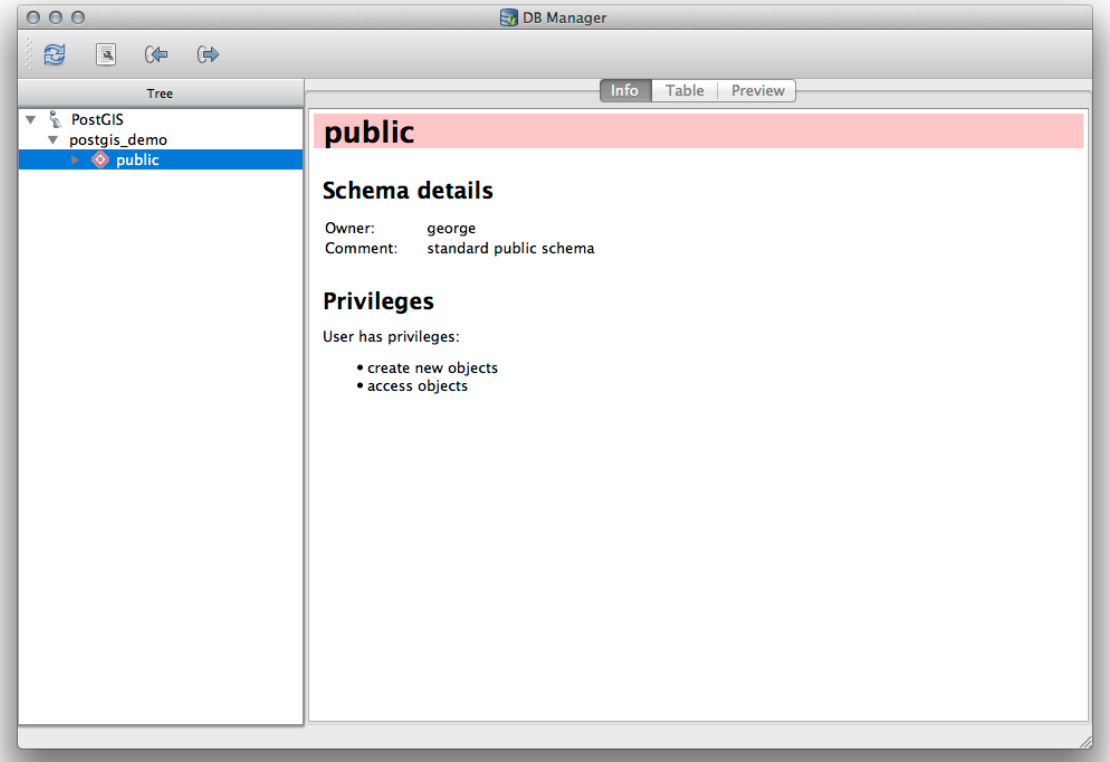
このレッスンの目標： QGIS の DB マネージャを使用して、空間データベースと対話する方法を学びます。

18.2.1 ★☆☆ (Basic level) Follow Along: Managing PostgreSQL Databases with DB Manager

最初に DB マネージャのインタフェースを開く必要があります。それには **データベース** *DB* **マネージャ** *DB* **マネージャ** メニューを選択するか、ツールバー上の  DB マネージャ アイコンを選択します。

すでに設定した接続が表示され、`myPG` セクションとその `public` スキーマを展開すると、前のセクションで扱ったテーブルを見ることができるはずです。

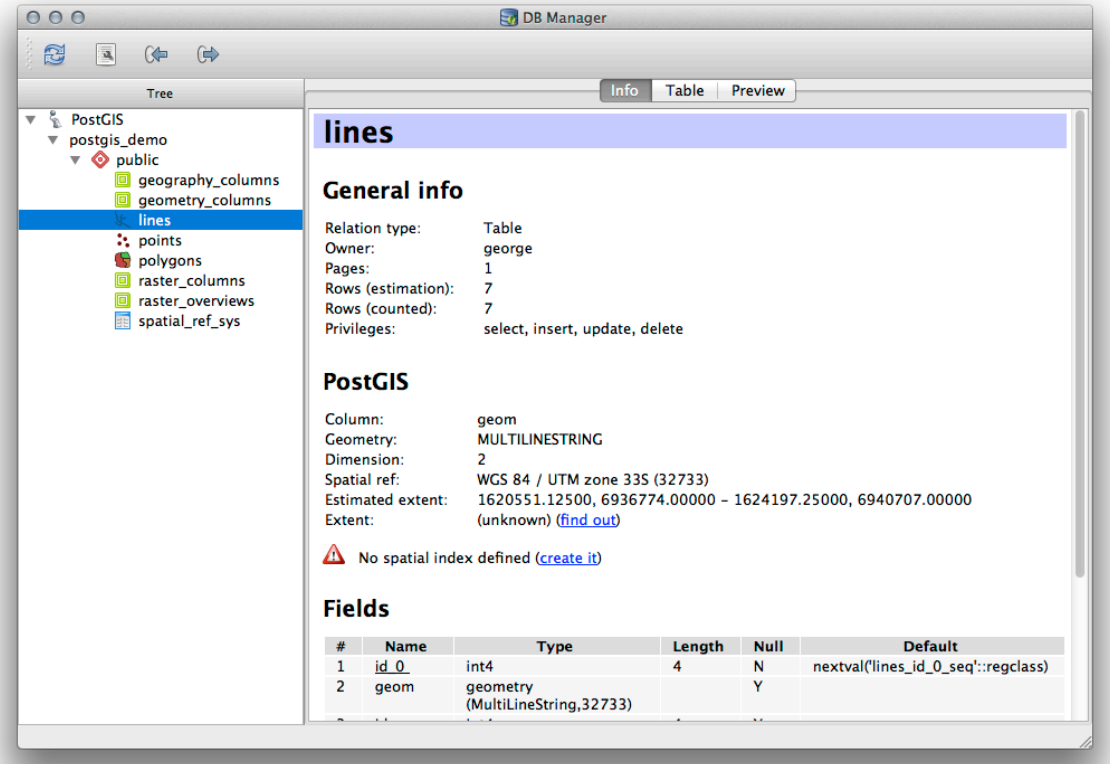
まず最初に気づくのは、データベースに含まれるスキーマに関するメタデータが表示されるようになったことです。



スキーマは PostgreSQL データベースのデータテーブルやその他のオブジェクトをグループ化する方法であり、権限やその他の制約のためのコンテナでもあります。PostgreSQL のスキーマの管理はこのマニュアルの範囲外ですが、PostgreSQL documentation on Schemas にスキーマに関するより多くの情報があります。DB マネージャを使用して新しいスキーマを作成することができますが、効果的に管理するためには pgAdmin III やコマンドラインインタフェースのようなツールを使用する必要があります。

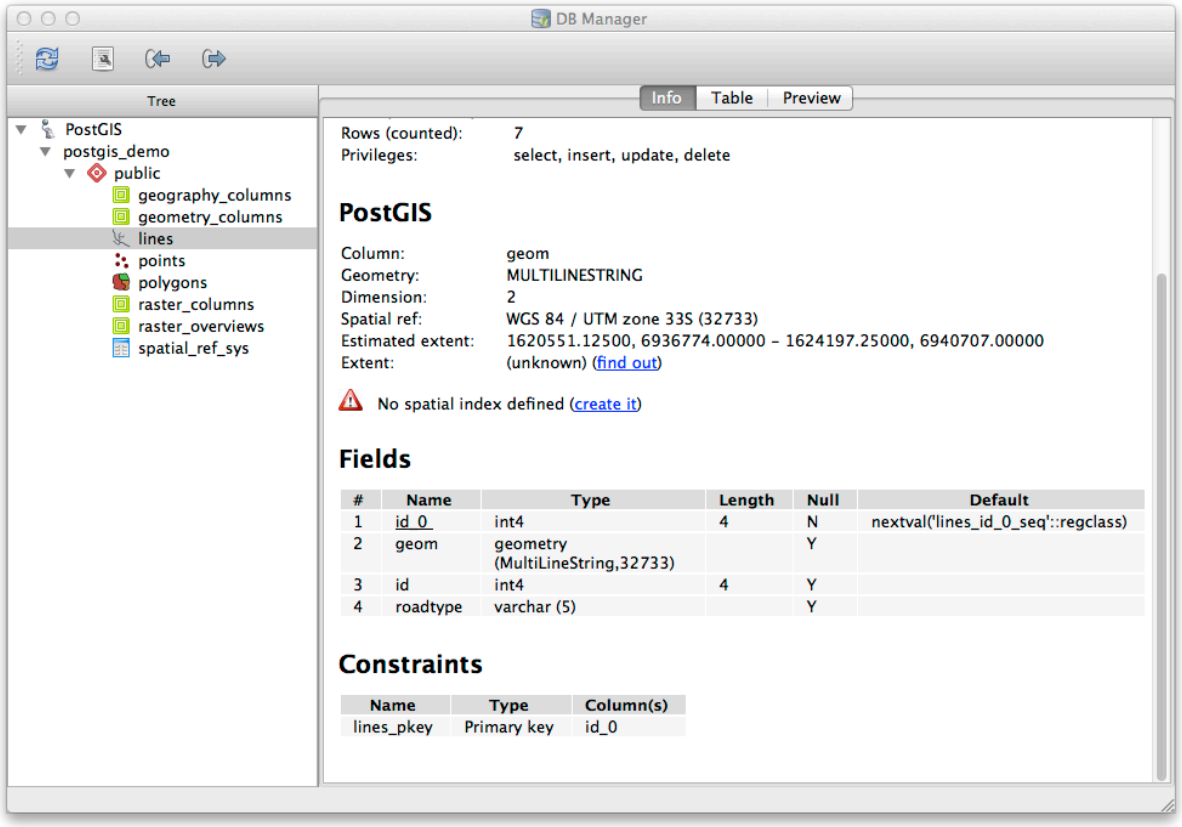
DB マネージャは、データベース内のテーブルを管理するために使用できます。すでに、コマンドライン上でテーブルを作成し管理するためのさまざまな方法を見てきていますが、ここで DB マネージャでこれを行う方法を見てみましょう。

まず、ツリー内の名前をクリックして **情報** タブの中を見ることで、テーブルのメタデータを見てみることは有用です。

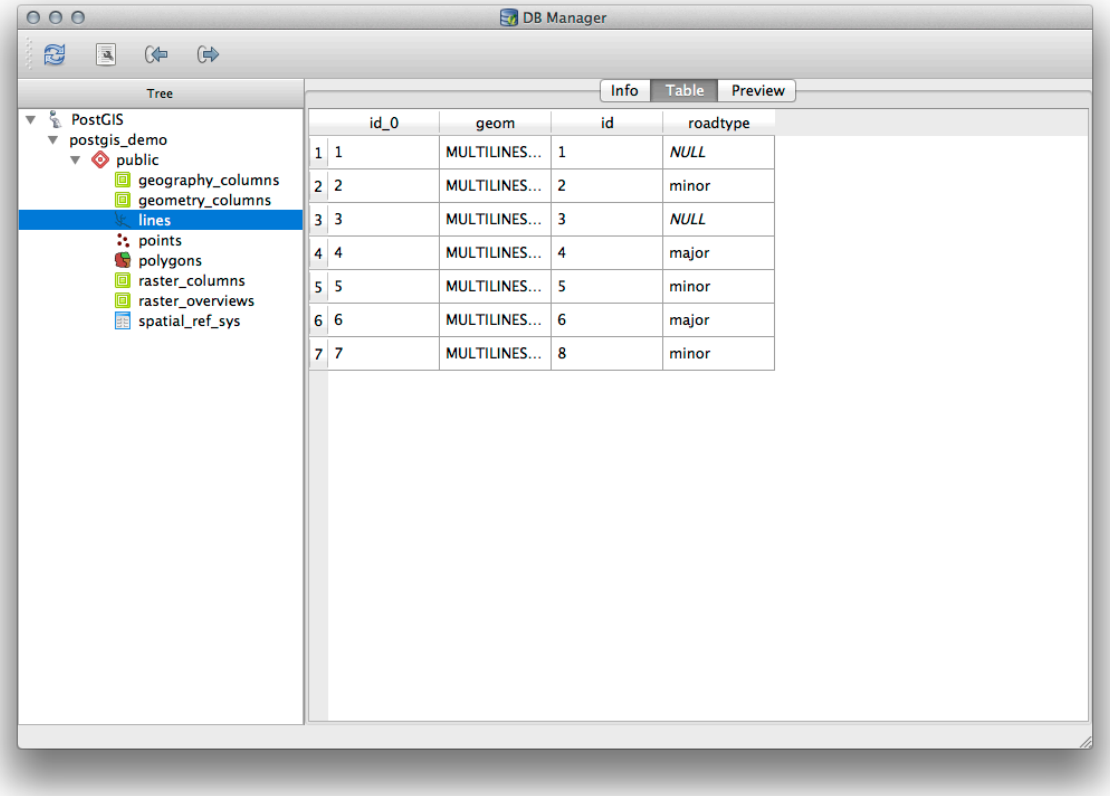


このパネルでは、テーブルについての **一般情報** も、ジオメトリと空間参照系に関して PostGIS 拡張が維持している情報を見ることができます。

情報 タブで下にスクロールした場合、表示しているテーブルに対して、 **属性** 、 **制約** 、 **インデックス** についての詳細な情報を見ることができます。



単にレイヤツリーにレイヤの属性テーブルを表示することによって、これを行う可能性がありますほとんど同じ方法でデータベース内のレコードを見て DB マネージャを使用することも非常に便利。 **テーブル** タブを選択して、データを閲覧できます。

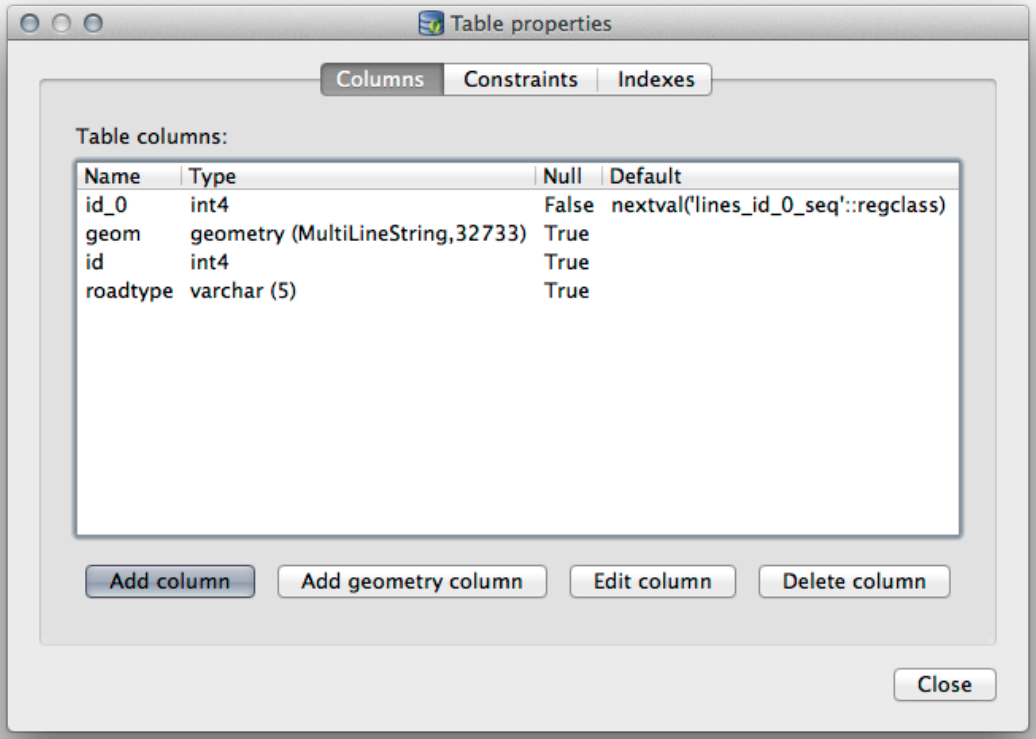


地図プレビューでレイヤのデータが表示される **プレビュー** タブもあります。

ツリー内のレイヤを右クリックし、**キャンバスに追加** をクリックすると地図にこのレイヤが追加されます。

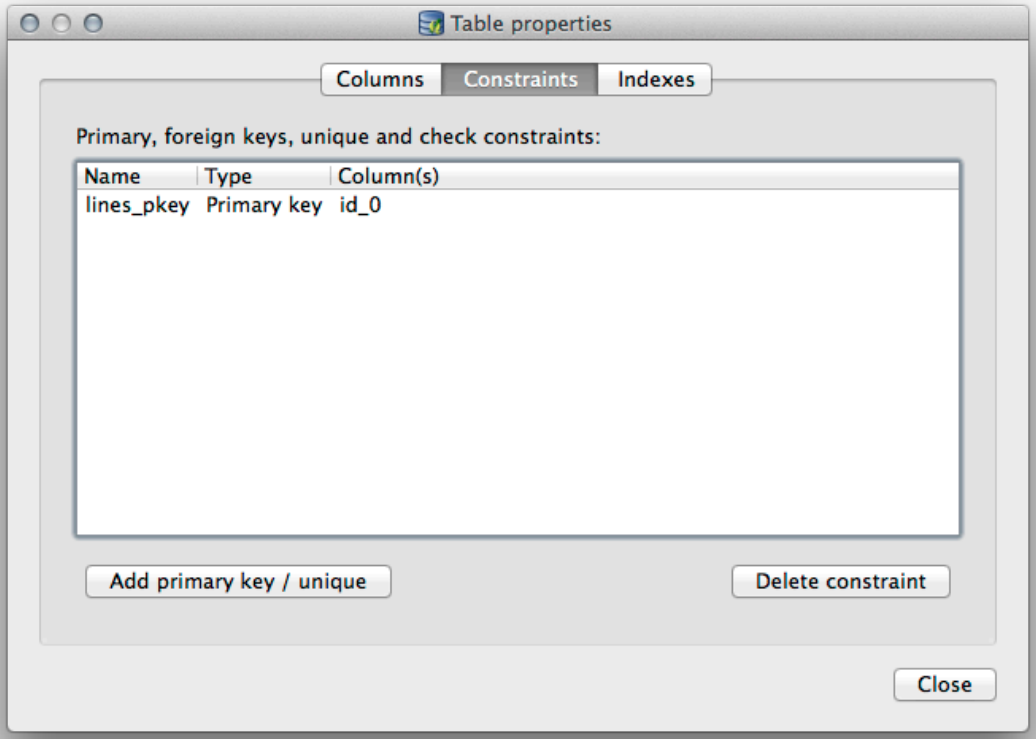
これまでのところはデータベース、そのスキーマとテーブル、それらのメタデータを閲覧しているだけですが、もしテーブルを変更して列を追加したいとしたらどうでしょう。DB マネージャを使えば、これを直接行うことができます。

1. ツリーで、編集するテーブルを選択します
2. メニューから **テーブル** **テーブルを編集** を選択し、**テーブルプロパティ** ダイアログを表示します。

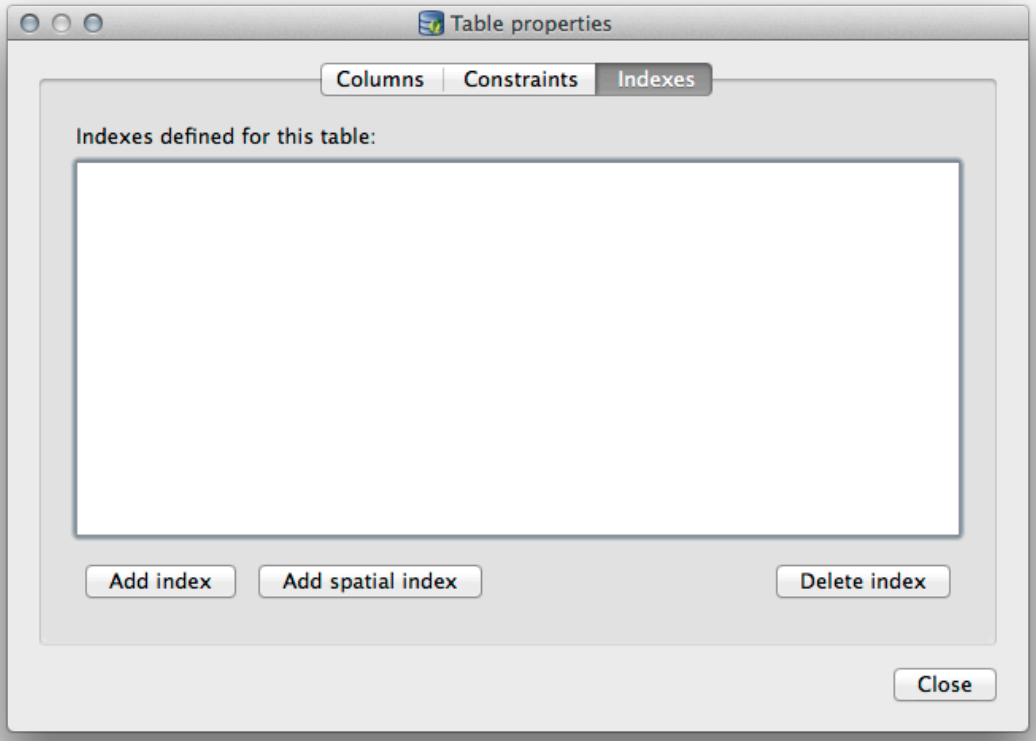


列を追加するには、このダイアログを使用してジオメトリ列を追加し、既存の列を編集したり、列を完全に削除できます。

制約 タブを使用、どのフィールドが主キーとして使用されるか、既存の制約を削除するために管理できます。



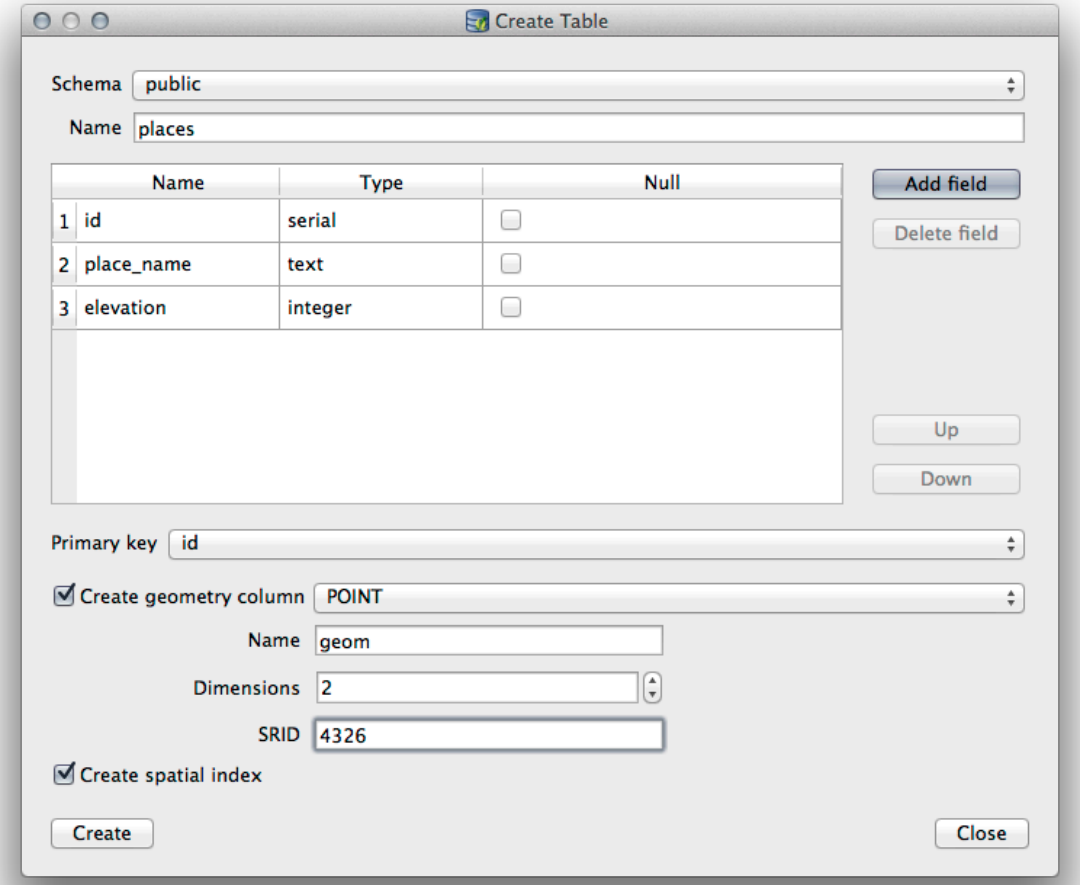
インデックス タブは、空間索引と通常のインデックスの両方を追加および削除するために使用できます。



18.2.2 ★☆☆ （初級レベル） 理解しよう: 新しいテーブルを作る

データベース内の既存のテーブルでの作業のプロセスを一通り終えましたので、ここで DB マネージャを使用して新しいテーブルを作成してみましょう。

1. まだ開いていない場合は、DB マネージャウィンドウを開き、データベースに既にあるテーブルのリストが表示されるまでツリーを展開します。
2. メニューから **テーブル** --> **テーブルを作成** を選択し、 [テーブルの作成] ダイアログを開きます。
3. デフォルトの `Public` スキーマを使用し、テーブル名を `places` とします。
4. 以下のように `id`, `place_name`, `elevation` フィールドを追加します
5. `id` フィールドが主キーとして設定されていることを確認します。
6. **ジオメトリカラムを作成** のチェックボックスをクリックし、`POINT` 型に設定されていることを確認し、名前を `geom` にして、`4326` を *SRID* として指定します。
7. **空間インデックスを作成** の横にあるチェックボックスをクリックし、**作成** をクリックしてテーブルを作成します。



8. テーブルが作成されたことを知らせるダイアログを閉じ、 **閉じる** をクリックして「テーブル作成」ダイアログを閉じます。

これで、DB マネージャで、テーブルを検査できますし、もちろんその中にデータがないことがわかります。ここからレイヤのメニューで **編集を切替** できます、テーブルに場所の追加を開始します。

18.2.3 ★☆☆ （初級レベル） 理解しよう: 基本的なデータベース管理

DB マネージャを使えば、基本的なデータベース管理のタスクを行うこともできます。より完全なデータベース管理ツールの代わりにはなりませんが、データベースを維持するために使えるいくつかの機能を提供します。

データベーステーブルは、多くの場合、非常に大きくなることがあり得ますし、頻繁に変更されているテーブルは、もはや PostgreSQL で必要とされてないレコードの残渣を残したままにすることがあります。*VACUUM* コマンドは一種のガベージコレクションを行い、パフォーマンス向上のためにテーブルを圧縮したりオプションで分析します。

DB マネージャから *VACUUM ANALYZE* コマンドを実行する方法について見てみましょう。

1. DB マネージャツリーにあるテーブルのひとつを選択します

2. メニューから **テーブル** **バキューム解析の実行** を選びます

PostgreSQL はこれで操作を実行します。テーブルの大きさにもよりますが、完了までに時間がかかるかもしれません。

VACUUM ANALYZE 処理については、PostgreSQL Documentation on VACUUM ANALYZE に詳細が記載されています。

18.2.4 ★☆☆ （初級レベル） 理解しよう: DB マネージャで SQL クエリを実行する

DB マネージャはまた、データベーステーブルに対してクエリを記述し、結果を表示するための方法を提供します。すでに **ブラウザ** パネルの中のこの種の機能を見てきましたが、再びここで DB マネージャでそれを見てみましょう。

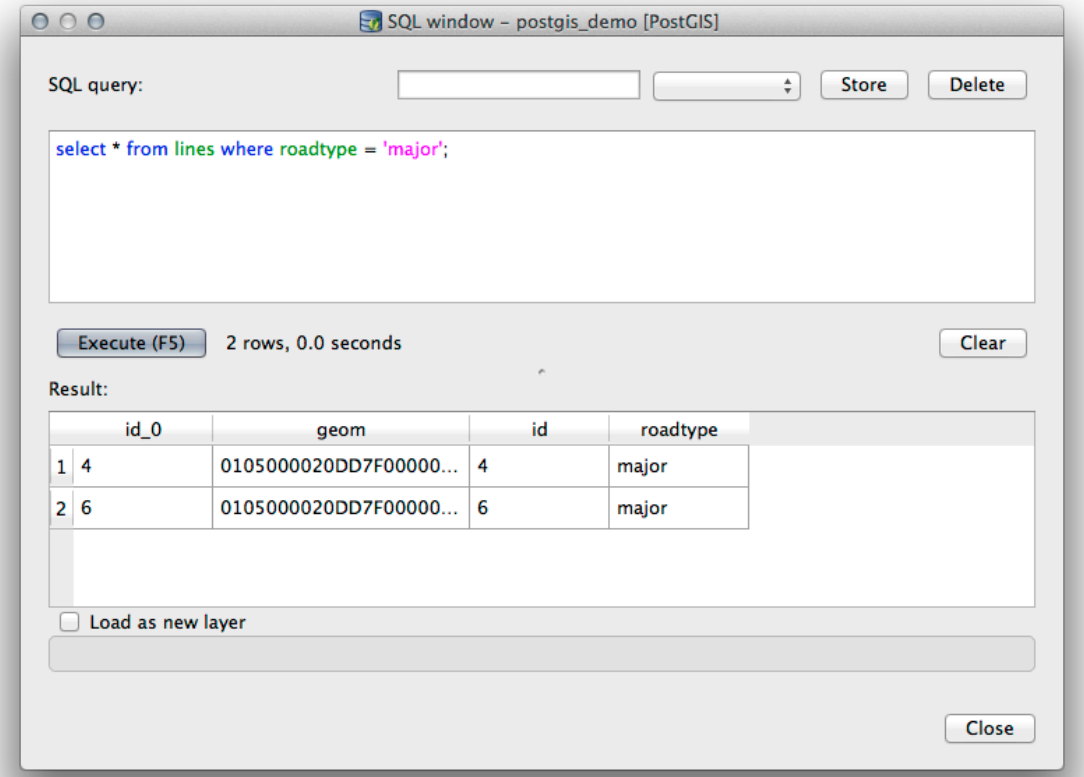
1. ツリーにある `lines` テーブルを選択します。
2. [DB マネージャ] ツールバーで *SQL* **ウィンドウ** ボタンを選択します。



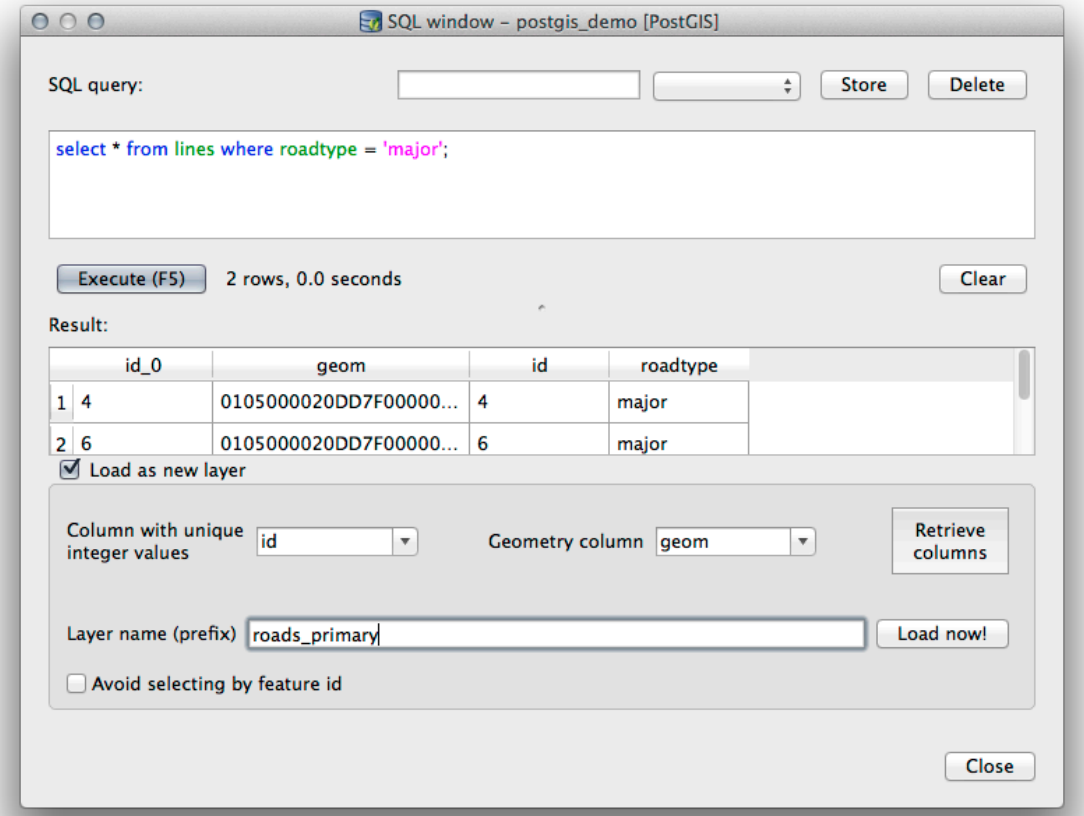
3. 次の *SQL* **クエリ** を与えられたスペースに構成します

```
select * from lines where roadtype = 'major';
```

4. クエリを実行するには、 **実行（*F5*）** ボタンをクリックします。
5. *Result* パネルに一致するレコードが表示されているはずです。



6. **新規レイヤとして読み込む** のチェックボックスをクリックして地図に結果を追加します。
7. id 列を **ユニーク値のカラム** として、geom 列を **ジオメトリのカラム** として選択します。
8. **レイヤ名（接頭辞）** として roads_primary を入力します。
9. **読み込み** をクリックして地図に新しいレイヤとして結果を読み込みます。



クエリと一致したレイヤーは今、地図上に表示されます。もちろんこのクエリツールは、以前のモジュールとセクション中で見たものなど、任意の SQL コマンドを実行するために使用できます。

18.2.5 DB マネージャを使用したデータベースへのデータのインポート

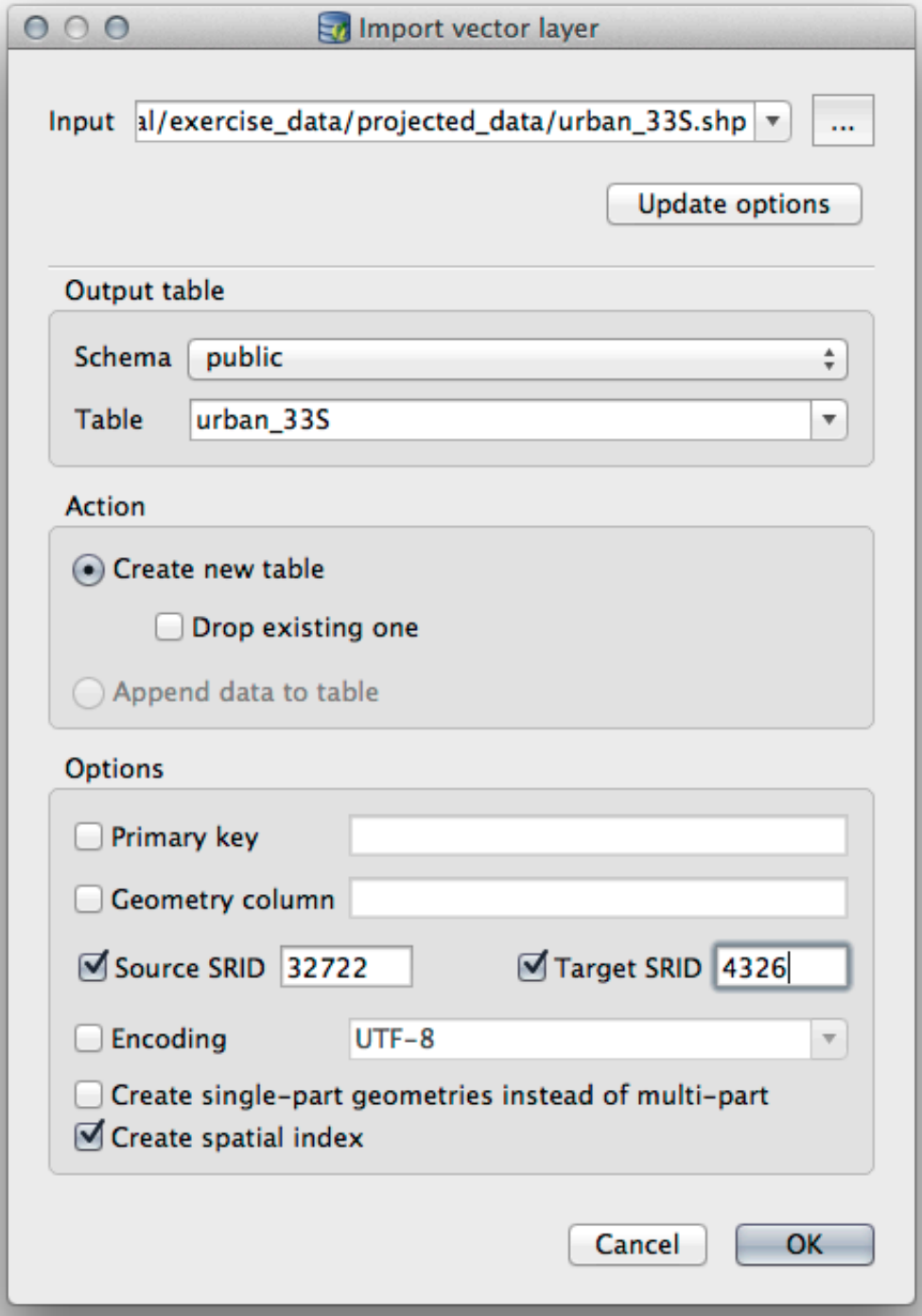
コマンドラインツールを使用して空間データベースにデータをインポートする方法をすでに見てきました。次に、DB マネージャを使用してインポートを行う方法を学習しましょう。

1. [DB マネージャ] ダイアログボックスのツールバー上の **レイヤー/ファイルを読み込み** ボタンをクリックしてください。



2. 入力データセットとして `exercise_data/projected_data` から `urban_33S.shp` ファイルを選択します。
3. フォームの値の一部を事前に埋めるために、 **オプションを更新** ボタンをクリックしてください。
4. **新しいテーブルを作成** オプションが選択されていることを確認してください

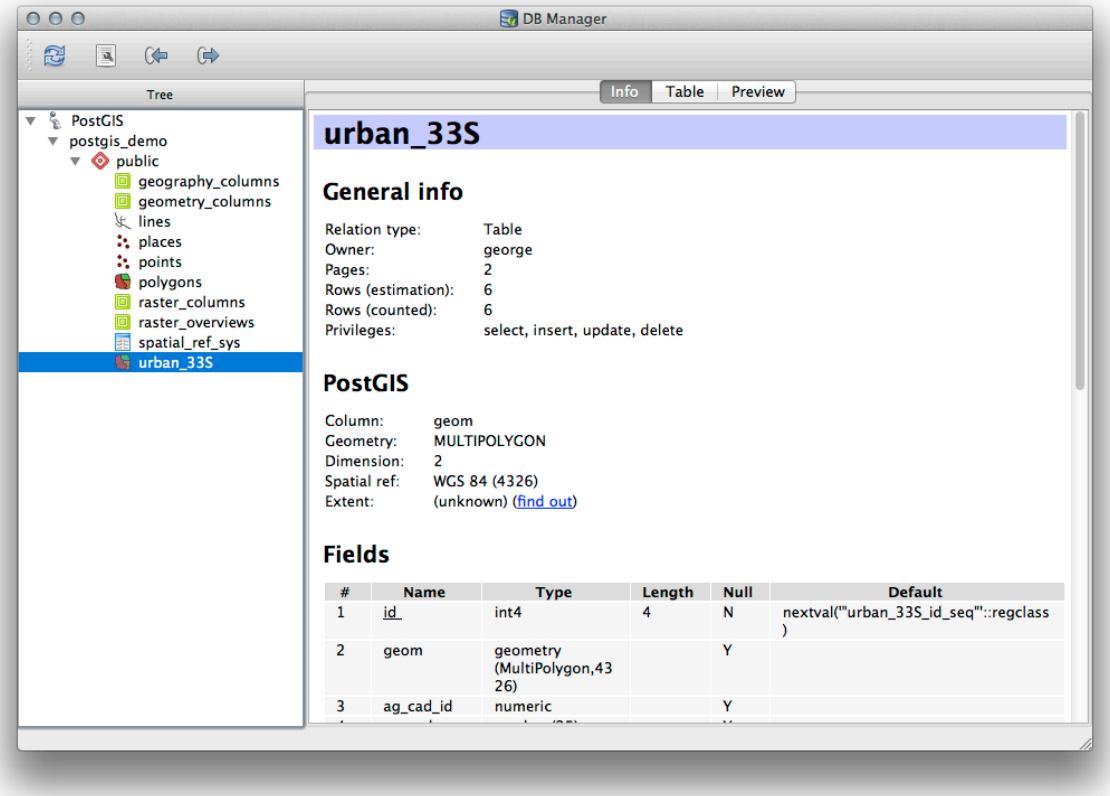
5. **変換前** *SRID* を ``32722`` に、 **変換後** *SRID* を ``4326`` に指定します
6. **空間索引を作成** の横にあるチェックボックスをオンにします。
7. インポートを実行するために *OK* をクリックします



8. 読み込みが成功したことを知らせるダイアログを閉じます

9. DB マネージャツールバーにある **リフレッシュ** ボタンをクリックします

これで、ツリーでテーブルをクリックすることで、データベース内のテーブルを検査することができます。*Spatial ref:* が `WGS 84 (4326)` と表示されていることを確認し、データが再投影されていることを確認します。



ツリー中のテーブルを右クリックして **キャンバスに追加** を選択すると地図にレイヤとしてテーブルが追加されます。

18.2.6 DB マネージャを使用したデータベースからのデータの書き出し

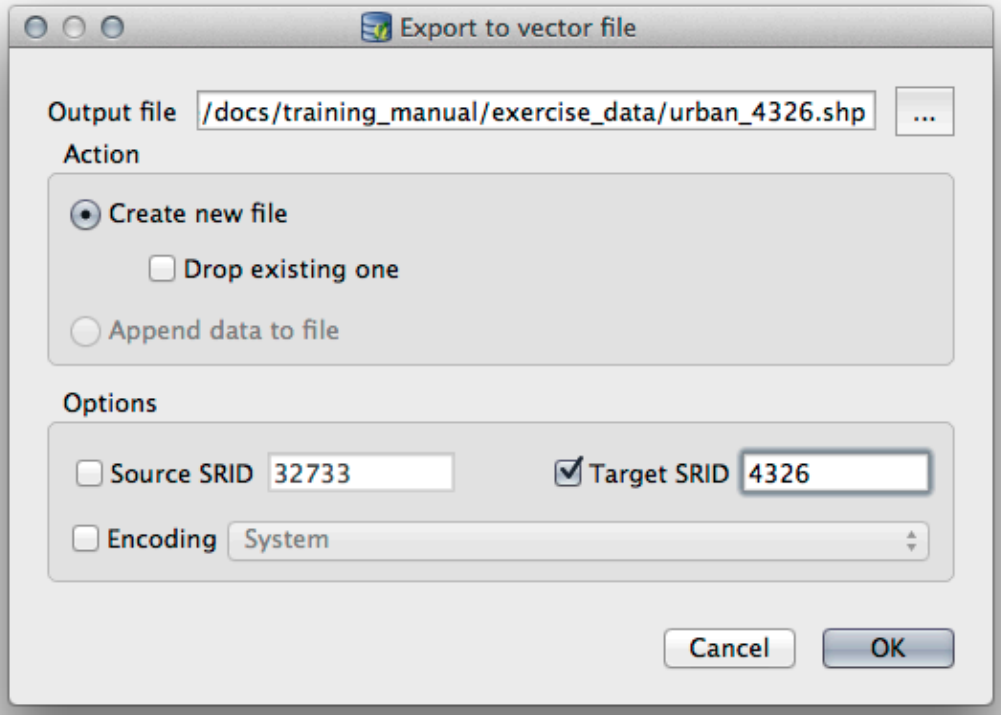
DB マネージャは空間データベースからデータを書き出すためにも当然使用できますので、それがどのように行われるかを見てみましょう。

1. ツリーで `lines` レイヤを選択し、ツールバーの **ファイルにエクスポート** ボタンをクリックして **ベクタファイルに出力する** ダイアログを表示します。

2. ... ボタンをクリックして **出力ファイル** を選択し、 `exercise_data` ディレクトリに `urban_4326` という名前でデータを保存します。

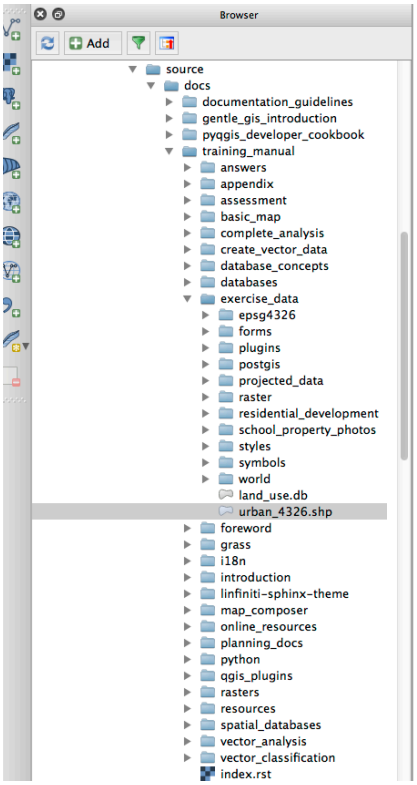
3. **変換後** *SRID* に `4326` を設定します。

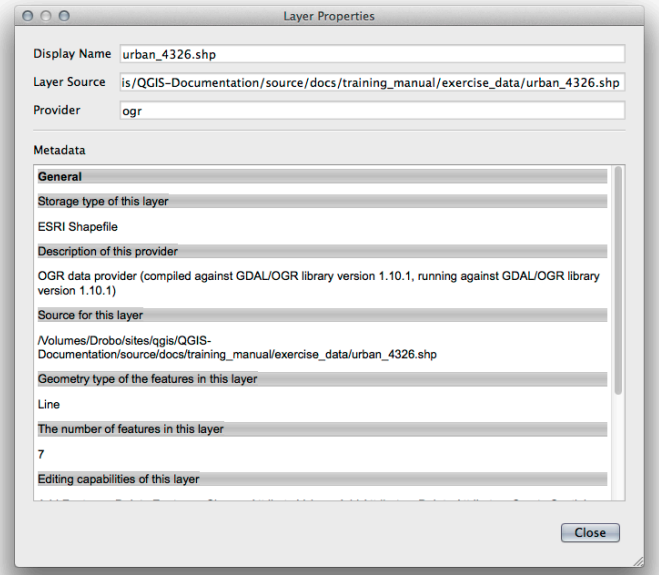
4. *OK* をクリックしてエクスポートを開始します。



5. 書き出しが成功したことを知らせるダイアログを閉じ、DB マネージャを閉じます。

ブラウザパネルで作成したシェープファイルを検査できます。





18.2.7 結論

ここまで、QGIS の DB マネージャインターフェイスを使用して空間データベースを管理する方法、データに対して SQL クエリを実行する方法、データのインポートとエクスポートの方法について説明しました。

18.2.8 次は?

次に、これらの同じテクニックを *SpatiaLite* データベースで使用する方法について説明します。

18.3 レッスン：QGIS で SpatiaLite データベースを操作する

PostGIS は一般的にサーバー上で同時に複数のユーザーに空間データベース機能を提供するために使用されますが、QGIS では *SpatiaLite* と呼ばれるファイル形式の使用もサポートします。これは、空間データベース全体を単一ファイルに保存するための軽量でポータブルな方法です。明らかに、これら 2 種類の空間データベースは異なる目的に使用する必要がありますが、同じ基本原則と手法が両方に適用されます。新しい SpatiaLite データベースを作成し、QGIS でこれらのデータベースを操作するために提供される機能を調べてみましょう。

このレッスンの目標： QGIS ブラウザインターフェースを使用して SpatiaLite データベースと対話する方法を学ぶ。

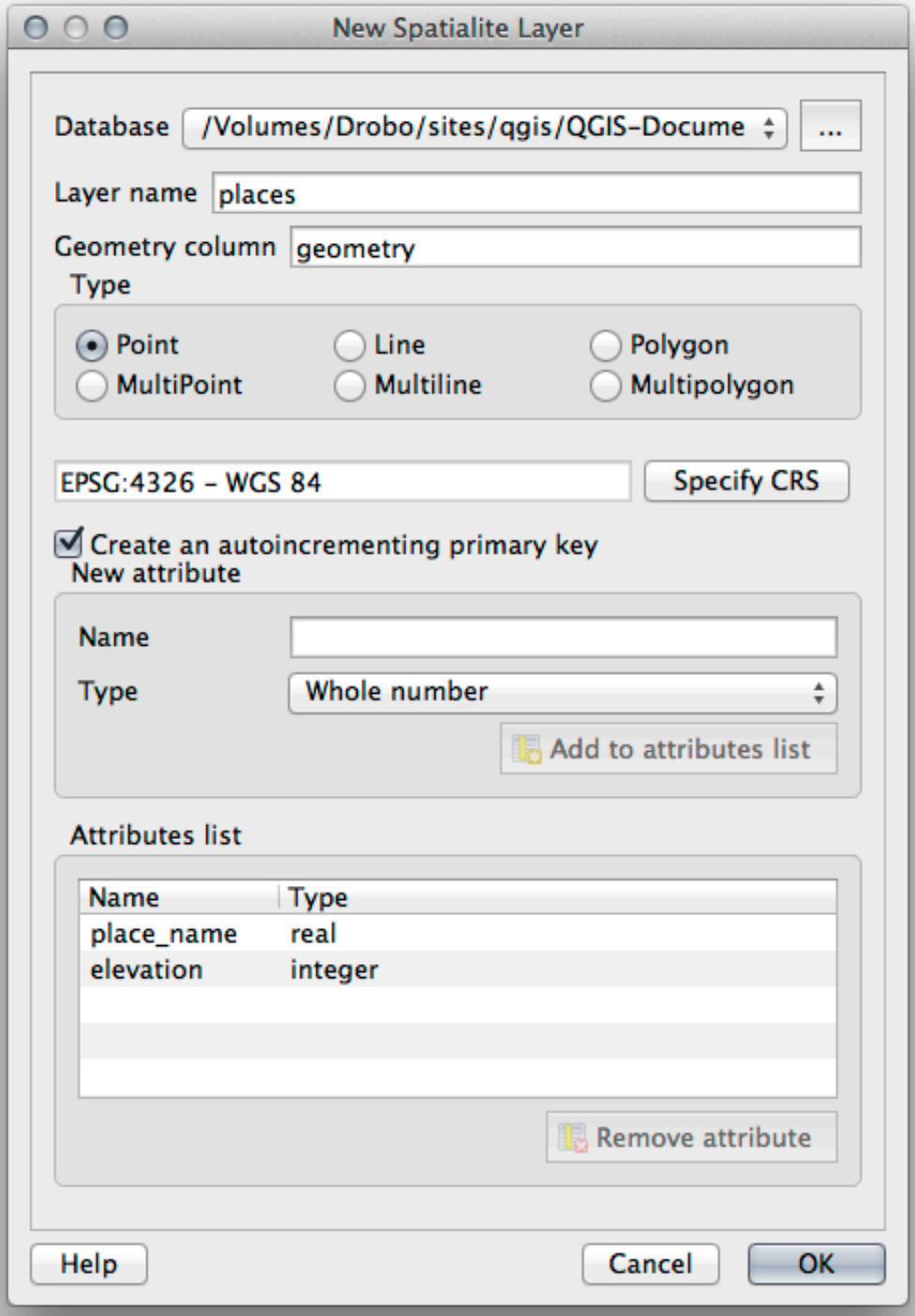
18.3.1 ★☆☆ （初級レベル） 理解しよう: ブラウザで SpatiaLite データベースを作る

ブラウザパネルを使用して、新しい SpatiaLite データベースを作成し、QGIS で使用するためにセットアップすることができます。

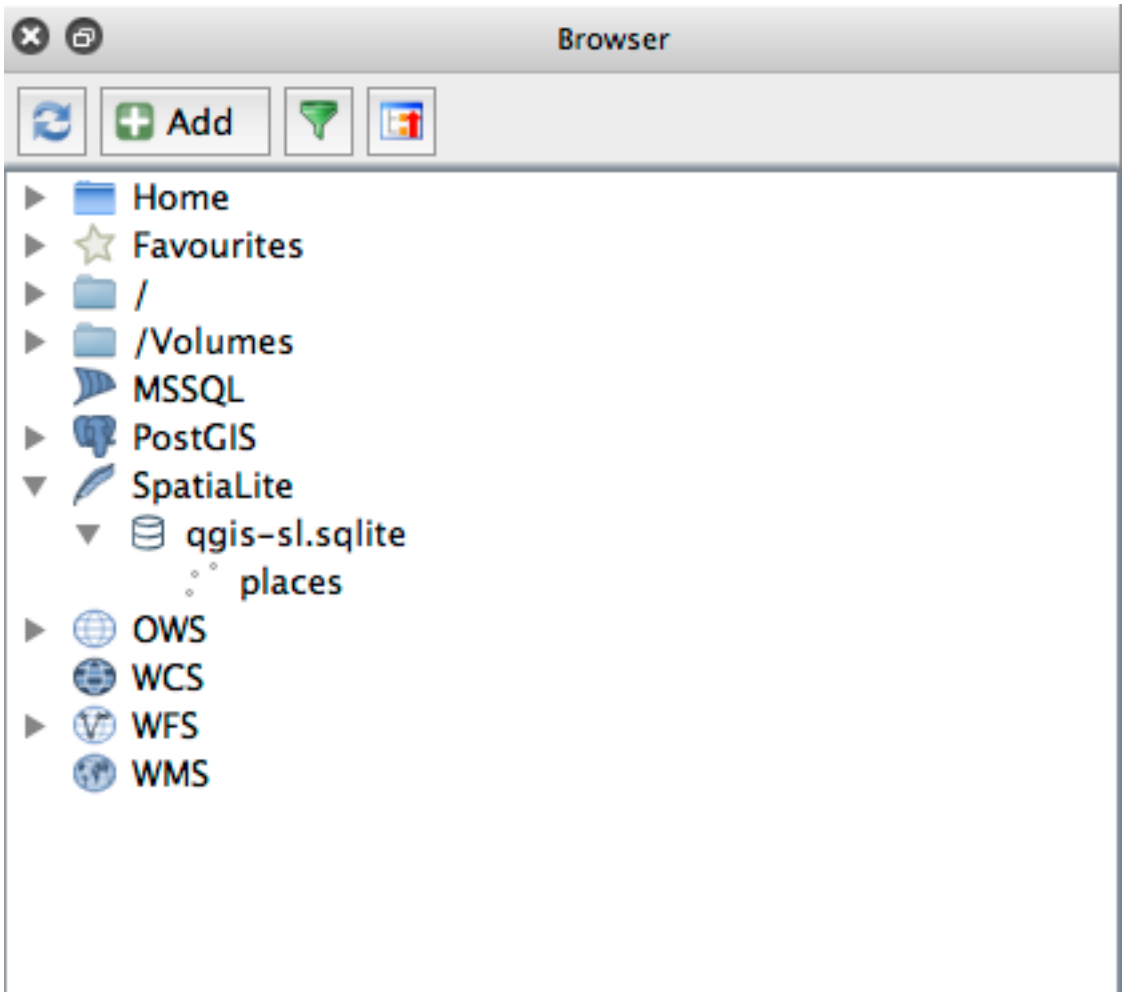
1. ブラウザツリーの *SpatiaLite* エントリを右クリックし、 **データベースの作成** を選択します。
2. ファイルシステムのどこにファイルを保存するかを指定し、「qgis-sl.db」という名前を付けます。
3. ブラウザツリーの *SpatiaLite* エントリをもう一度右クリックし、 *NewConnection* アイテムを選択します。最後の手順で作成したファイルを見つけて開きます。

これでブラウザツリーに新しいデータベースが構成され, その下には何も持たないことから, 行える操作は削除できることだけだということがわかります. このデータベースには何のテーブルも追加していないのでこれは当然です. それでは先に進んでみましょう.

1. 新しいレイヤーを作成するためのボタンを見つけ、ドロップダウンを使用して新しい SpatiaLite レイヤーを作成するか、 **レイヤ->新規->**  **新しい** *SpatiaLite* **レイヤー** を選択します。
2. 前の手順で作成したデータベースをドロップダウン内から選択します.
3. そのレイヤに名前 `places` を付けます。
4. 次に **自動増分の主キーを作成する** のチェックボックスを選択します.
5. 以下に示すように、2 つの属性を追加します
6. *OK* をクリックしてテーブルを作成します.



7. ブラウザの上部にある更新ボタンをクリックすると、 ``場所`` テーブルが一覧表示されます。



テーブルを右クリックして, 前のレッスンで行ったようにプロパティを表示することができます.

ここから編集のセッションを開始して直接新しいデータベースにデータを追加することができます.

また、DB Manager を使用してデータベースにデータをインポートする方法についても学びました。これと同じ手法を使用して、新しい SpatiaLiteDB にデータをインポートできます。

18.3.2 結論

SpatiaLite データベースを作成し、それらにテーブルを追加し、これらのテーブルを QGIS のレイヤーとして使用する方法を見てきました。

第19章　付録：このマニュアルに貢献する

このコースに教材を追加するには、この付録のガイドラインに従わなければなりません。明確にする場合を除き、この付録の条件を変更してはいけません。これは、このマニュアルの品質と一貫性を維持するためです。

19.1 リソースのダウンロード

この文書のソースは GitHub で提供されています。Git のバージョン管理システムを使用する方法については、 GitHub.com を参照してください。

19.2 マニュアルの形式

このマニュアルは、reStructuredText マーク付け言語を使った Python ドキュメントジェネレータ、Sphinx を使って書かれました。これらのツールの使い方はそれぞれのサイトで入手できます。

19.3 モジュールを追加する

新しいモジュールを追加するには:

1. 最初に新しいモジュールの名前を持った新しいディレクトリを（ `qgis-training-manual` ディレクトリのトップレベルのすぐ下に）作成します。
2. この新しいディレクトリの下に、 `index.rst` というファイルを作成します。今のところ、このファイルは空のままにします。
3. 最上位ディレクトリの下の `index.rst` ファイル開きます。その最初の数行は:

```
.. toctree::
   :maxdepth: 2

   foreword/index
   introduction/index
```

これは、名前 `index` が続く、ディレクトリ名のリストであることに注意します。これは、トップレベルのインデックスファイルに各ディレクトリ内のインデックスファイルを指示します。リストされている順序で文書中の順序が決定されます。

1. このリストのモジュールを表示したい場所に、新しいモジュールの名前（すなわち、新しいディレクトリに付けた名前）に `/index` を続けたものを追加します。
2. 後のモジュールは前のモジュールで提示される知識の上に構築するように、モジュールの順序を論理的に維持することを忘れないでください。
3. 新しいモジュールの独自のインデックスファイル（ *module_name*`/index.rst` ）を開きます。
4. ページの先頭に沿ってモジュールの見出しを作ります:
    1. アスタリスク (*) の最初の行を書きます。
    2. `Module:` というマークアップ語句の後に、モジュールの名前を付けた 1 行を続けます。
    3. 同じ数のアスタリスクをもう 1 行書いて終えます。

    注釈

    アンダーラインとオーバーラインは、モジュールの表題を含む行よりも短くすべきではありません。

5. この下に空白行を置きます。
6. モジュールの目的と内容を説明する短い段落を書きます。
7. 1 行を空白のままに残し、次のテキストを追加します:

```
.. toctree::
   :maxdepth: 2

   lesson1
   lesson2
```

... ここで、`lesson1` 、 `lesson2` などは、計画したレッスンの名前です。

モジュールレベルのインデックスファイルは次のようになります。

```
********************************************************************************
Module: Module Name
********************************************************************************

Short paragraph describing the module.

.. toctree::
   :maxdepth: 2

   lesson1
   lesson2
```

19.4 レッスンを追加する

新規または既存のモジュールにレッスンを追加するには:

1. モジュールディレクトリを開きます。
2. `index.rst` ファイル（新しいモジュールの場合に上で作成）を開きます。
3. 計画したレッスンの名前が、上で示されるように、 `toctree` ディレクティブ下に表示されていることを確認します。
4. モジュールディレクトリの下に新しいファイルを作成します。
5. このファイルにモジュールの `index.rst` ファイルの中に与えた名前とまったく同じ名前を付け、拡張子 `.rst` を追加します。

> **注釈**
>
> 編集する目的では、 `.rst` ファイルは通常のテキストファイル（ `.txt` ）とまったく同じように動作します。

1. レッスンを書き始めは、マークアップ語句 `Lesson` にレッスン名を続けて書きます。
2. 次の行には、等号 (='') の行をレッスンの表題より短くならないように書きます。
3. この後に空行を置きます。
4. レッスンの意図された目的について短い説明を書きます。
5. 主題への一般的な紹介を含めます。例として、このマニュアル中の既存のレッスンを参照してください。
6. この下には、この語句から始まる、新しい段落を開始します:

```
**The goal for this lesson:**
```

7. このレッスンを完了することによる意図した成果を簡単に説明します。
8. 1 つのまたは 2 つの文にレッスンの目標を記述できない場合は、主題を複数のレッスンに分けることを検討してください。

次に説明するように、各レッスンは複数のセクションに細分化されます。

19.5 セクションを追加する

セクションには、「理解しよう」と「自分でやってみよう」の 2 種類があります。

- 「理解しよう」セクションは指示の詳細なセットで、QGIS の所定の態様を使用する方法を読者に教示することを意図しています。これは通常、スクリーンショットを散りばめた状態で、クリックごとの指示をできる限り明確に示すことで行われます。

- 「やってみよう」セクションは、読者に自分でやらせる短い課題を与えます。通常、その下には結び付いた解答欄があり、その課題を完了する方法を示したり、説明したり、可能であれば期待される結果を示したりします。

すべてのセクションには難易度が付けられています。簡単なセクションは ★☆☆ 、中級は ★★☆ 、そして上級は ★★★ で表されます。

19.5.1 「理解しよう」セクションを追加する

1. （上記のように）このセクションを開始するには、意図した難易度のマークアップ語句を書き込みます。
2. スペースを置き、次に「理解しよう:」を書きます。
3. もうひとつスペースを置いて、セクションの名前を書きます（初回のみ大文字だけでなく、固有名詞のための大文字を使用）。
4. 次の行には、負号／ダッシュ (-') の行をセクションの表題より短くならないように書きます。
5. その目的を説明し、セクションへの簡単な紹介を書きます。そして、例証される手続きについての詳細な（クリック毎の）指示を与えます。
6. 必要に応じて各セクションには、内部リンク、外部リンク、およびスクリーンショットが含まれます。
7. 可能ならば、それを終了し、次のセクションに自然につながる短い段落で、各セクションを終了してみてください。

19.5.2 「自分でやってみよう」セクションを追加する

1. （上記のように）このセクションを開始するには、意図した難易度のマークアップ語句を書き込みます。
2. スペースを置き、次に「自分でやってみよう:」を書きます。
3. 次の行には、負号／ダッシュ (-') の行をセクションの表題より短くならないように書きます。
4. 読者に完成させたい練習を説明します。必要に応じて、前のセクション、レッスンやモジュールを参照します。
5. 単なる文章での説明でははっきりしない場合、要件を明確にするためにスクリーンショットを含めます。

ほとんどの場合、このセクションで与えられた課題を完成する方法についての解答を提供したいと思うでしょう。そのためには、指示の下に解答ブロックを作って与える必要があります。

1. まず、解答を格納するカスタム折りたたみウィジェットを作ります:

```
.. admonition:: Answer
   :class: dropdown
```

2. 上のブロックに対してインデントを保ちながら、必要に応じてリンクや画像を使い、課題を完了するための指示を書きます。

19.6 結論を追加

レッスンを終えるには:

1. 語句「結論」を書き、マイナス/ハイフン (-) の新しい行を続けます。
2. どの概念がレッスンで取り扱われたかを説明しながら、レッスンの結論を書きます。

19.7 参考文献セクションを追加

このセクションは任意です。

- 語句「参考文献」を書き、マイナス/ハイフン（ - ）の新しい行を続けます。
- 適切な外部のウェブサイトへのリンクを含めます。

19.8 [次は] セクションを追加

1. 語句「次は?」を書き、マイナス/ハイフン (-) の新しい行を続けます。
2. このレッスンがどのように学生にとって次のレッスンまたはモジュールの準備になったかを説明します。
3. 必要であれば前のレッスンの「次は」のセクションを、新しいレッスンを参照するように変更することを忘れないでください。これは既存のレッスンの間に、または既存のレッスンの後に新しいレッスンを挿入した場合に必要となります。

19.9 マークアップを使用する

このドキュメントの基準を遵守するため、テキストに標準的なマークアップを追加する必要があります。

19.9.1 新しい概念

新しい概念を説明している場合は、その新しい概念の名前をアスタリスク (*) で囲んで斜体で記述する必要があります。

```
This sample text shows how to introduce a *new concept*.
```

19.9.2 強調

- 新しい概念ではない重要な用語を強調するには、その用語を二重のアスタリスク（**）で囲んで太字で記述します。
- これは控えめに使いましょう！多用すると、読者には怒鳴っていたり見下しているように見えることがあります。

```
This sample text shows how to use **emphasis** in a sentence. Include the
punctuation mark if it is followed by a **comma,** or at the **end of the
sentence.**
```

19.9.3 画像

- 画像を追加する場合は、レッスンファイルの隣にある `img` フォルダに保存します。
- 文書にそれを入れるにはこのようにします:

```
.. figure:: img/image_file.extension
   :align: center
```

- 画像マークアップの上方および下方の空行を残すことを忘れないでください。

19.9.4 内部リンク

- リンクのアンカーを作成するには、リンクが指すようにしたい場所の上に次の行を書きます:

```
.. _link-name:
```

- この行の上および下に空行を残すことを忘れないでください。
- リンクを作成するには、それを次のように参照します:

```
:ref:`Descriptive link text <link-name>`
```

19.9.5 外部リンク

- 外部リンクを作成するには、このように書き出します:

```
`Descriptive link text <link-url>`_
```

19.9.6 等幅テキストを使う

- ユーザーが入力する必要のあるテキスト、パス名、またはテーブルや列の名前などのデータベース要素の名前を書いているときは、それを 等幅テキスト で記述する必要があります。例えば:

```
Enter the following path in the text box: ``path/to/file``.
```

19.9.7 ラベルする GUI 項目

- GUI の項目、ボタンなど、を参照している場合は、 *GUI* **ラベルフォーマット** の中にその名前を書く必要があります。例えば:

```
To access this tool, click on the :guilabel:`Tool Name` button.
```

- これは、ユーザーがボタンをクリックする必要なしに、ツールの名前を言及している場合にも適用されます。

19.9.8 メニューの選択

- メニューを通じてユーザーを導く場合は、 **メニュー　選択　フォーマット** を使用する必要があります。例えば:

```
To use the :guilabel:`Tool Name` tool, go to :menuselection:`Plugins -->
Tool Type --> Tool Name`.
```

19.9.9 注を追加する

- レッスンの流れの一部とすることが容易でない余分な詳細を説明するために、テキストに注を加えることが必要になる場合があります。これがそのマークアップです:

```
[Normal paragraph.]

.. note:: Note text.
  New line within note.

  New paragraph within note.

[Unindented text resumes normal paragraph.]
```

19.9.10 後援/原作者注を追加する

スポンサーに代わって新しいモジュール、レッスンまたはセクションを記述する場合は、スポンサーが希望する短いスポンサーメッセージを含める必要があります。これはスポンサーの名前を読者に通知しなければならず、そのスポンサーが主催するモジュール、レッスンやセクションの見出しの下に表示されなければなりません。しかし、それはスポンサーの会社の広告であってはいけません。

スポンサーに代わってではなく、自身の能力で自発的にモジュール、レッスン、またはセクションを書いた場合は、執筆したモジュール、レッスンやセクションの見出しの下に原作者注を入れることができます。これは、 この [モジュール/レッスン/セクション] は [著者名] による寄稿です という形を取る必要があります。それ以上のテキストや連絡先などは追加しないでください。そのような詳細は、追加した部分の名前と一緒に、前文の「寄稿者」セクションに追加されるべきです。機能強化、修正および/または追加を行っただけの場合は、編集者として自分自身をリストします。

19.10 ありがとうございました！

このプロジェクトに貢献していただきありがとうございます！そうすることで、QGIS はユーザーからより利用しやすくなり、全体として QGIS プロジェクトに価値を付加しています。

第20章　練習データを準備する

重要

このプロセスは講座の召集者、または、その講座のためにローカライズしたサンプル・データセットを作りたい経験豊富な QGIS ユーザ向けにしました。デフォルトのデータセットはトレーニングマニュアルとともに提供されていますが、この手順に従ってそれを置き換えることができます。

提供されているサンプルデータ はトレーニングマニュアルと一緒に Swellendam とその周辺の町を指します。 Swellendam は南アフリカの西ケープ州ケープタウンの約 2 時間の東に位置しています。データセットは英語とアフリカーンス語の両方で地物名を含みます。

このデータセットは誰でも問題なく使うことができますが、自分の国や故郷のデータを使うことを好む方もいるかもしれません。それを選択した場合は、ローカライズされたデータが、モジュール 3 からモジュール 7.2 のすべてのレッスンで使用されます。それ以降のモジュールはより複雑なデータソースを使用しているので、あなたの地域で利用できる場合もできない場合もあります。

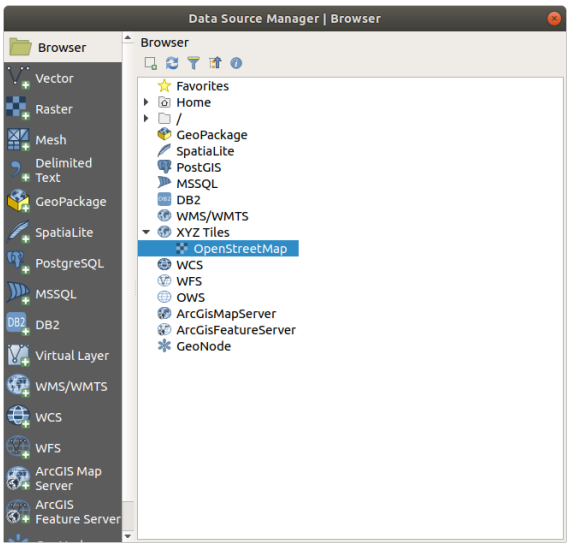
注釈

以下の手順では、QGIS の十分な知識を持っており、教材として使用されることを意図していないと仮定します。

20.1 OSM を基にしたベクタファイルを作る

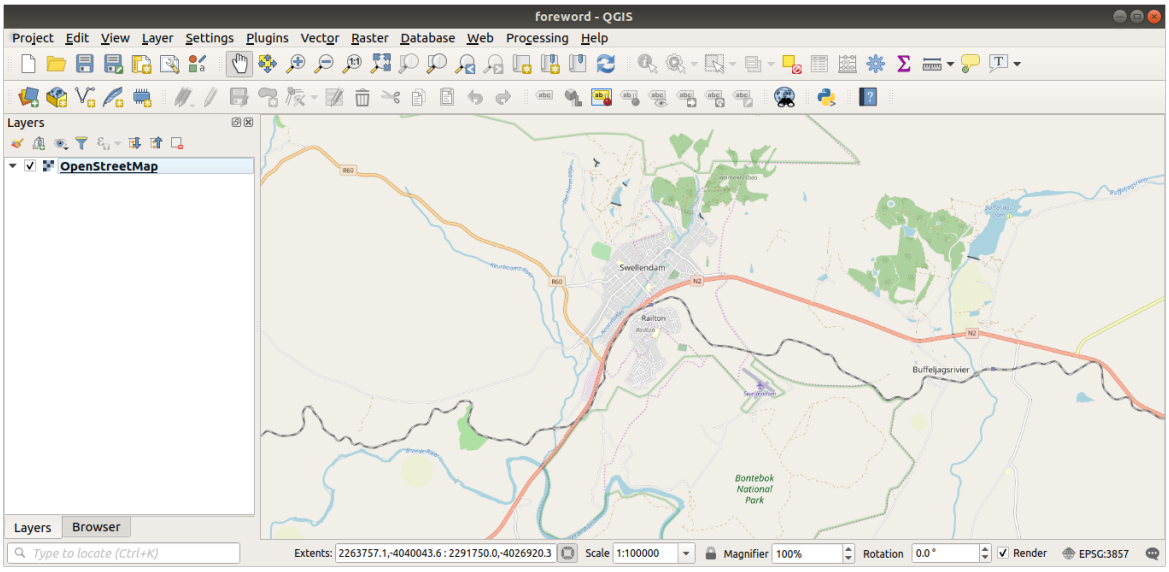
デフォルトのデータセットをコースのためのローカライズされたデータに置き換えたい場合、QGIS に組み込まれたツールを使って簡単に行うことができます。あなたが使おうとする領域は、都市部と農村部の適当な混合地であり、河川や水面、道路、地域の境界 (自然保護区や農場など) などが分かれてる必要があります。

1. QGIS プロジェクトを開く
2. **レイヤ　データソースマネージャ** を選択し、 **データソースマネージャ** ダイアログを開きます。
3. **ブラウザ** タブで、 *XYZ Tiles* ドロップダウンメニューを展開し、 *OpenStreetMap* アイテムをダブルクリックします。



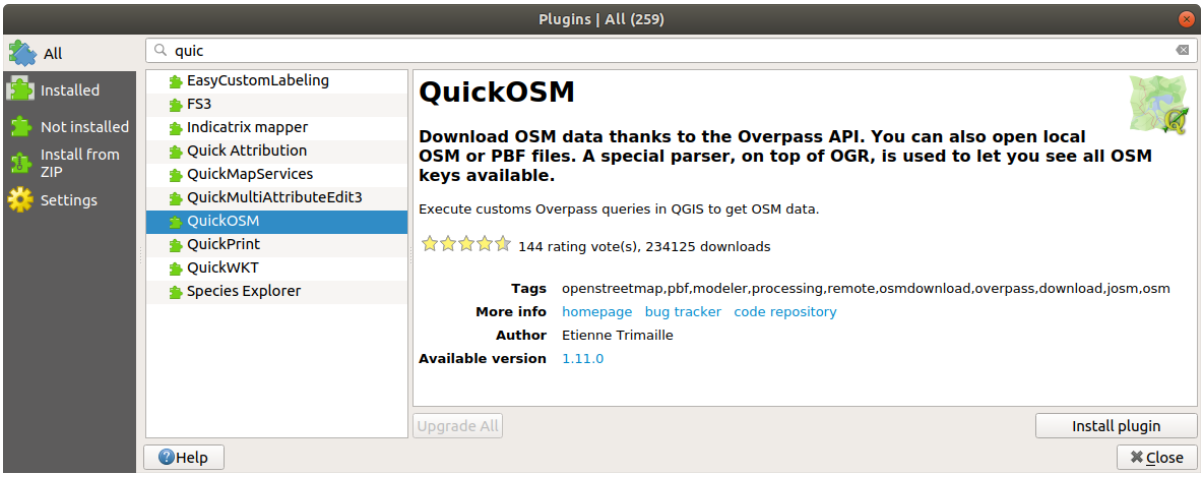
マップキャンバスに世界の地図が表示されるようになります。

4. **データソースマネージャ** ダイアログを閉じます
5. 学習エリアとして使用したいエリアに移動します

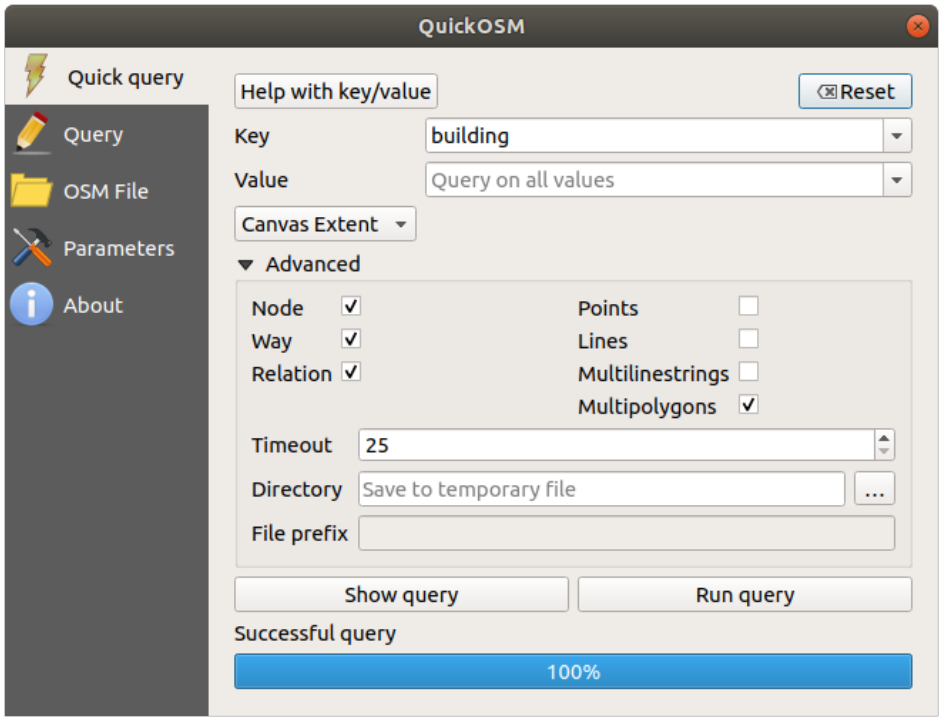


データを抽出する領域ができたので、抽出ツールを有効にしましょう。

1. **プラグイン　プラグインの管理とインストール...** を選びます
2. **すべて** タブで、検索ボックスに `QuickOSM` と入力します
3. QuickOSM プラグインを選択し、 **インストール** を押し、ダイアログを **閉じる** します。



4. **ベクタ** *QuickOSM* *QuickOSM...* メニューから、新しいプラグインを実行します
5. *Quick query* タブで、 *Key* ドロップダウンメニューから `building` を選択します
6. *Value* フィールドを空にして、すべてのビルに問い合わせることを意味します。
7. 次のドロップダウンメニューで *Canvas Extent* を選択します
8. 下の *Advanced* グループを展開し、 *Multipolygons* を除く右側のすべてのジオメトリタイプのチェックを外します。
9. *Run query* を押します

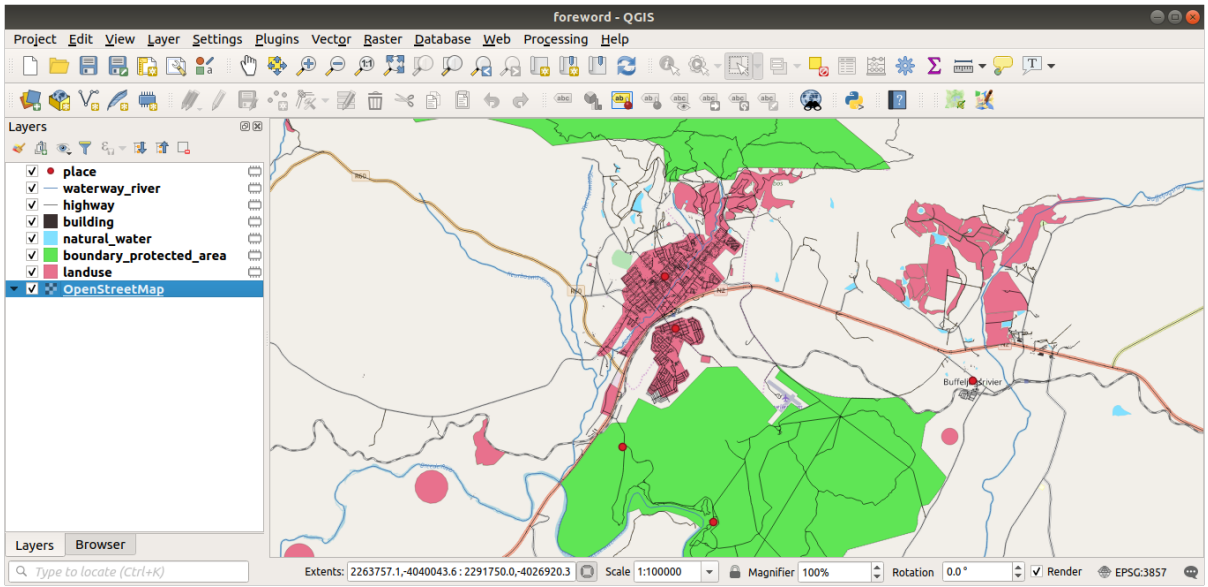


新しい `building` レイヤが **レイヤ** パネルに追加され、選択した範囲の建物が表示されます。

10. 他のデータを抽出するため、上記と同様に行います：
    1. `Key = landuse` とジオメトリ型 `Multipolygons`。

2. Key = boundary、 Value = protected_area とジオメトリ型 Multipolygons。
3. Key = natural, Value = water とジオメトリ型 Multipolygons 。
4. Key = highway とジオメトリ型 Lines と Multilines をチェックする。
5. Key = waterway、 Value = river とジオメトリ型 Lines と Multilines をチェックする。
6. Key = place とジオメトリ型 Points 。

このプロセスにより、レイヤーが一時ファイルとして追加されます（名前の横にある アイコンで示されます)。



あなたの地域に含まれるデータをサンプリングして、あなたの地域でどのような結果が得られるかを確認できます。

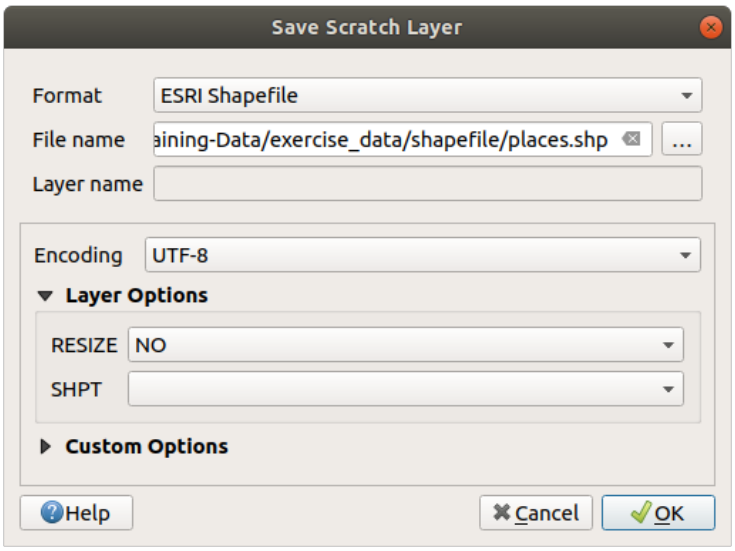
講座で使用するために、結果のデータを保存する必要があります。データに応じて、ESRI シェープファイル、GeoPackage、SpatiaLite 形式を使用します。

To convert the place temporary layer to another format:

1. Click the icon next to the place layer to open the *Save Scratch Layer* dialog.

> **注釈**
>
> 一時レイヤーのプロパティ（CRS、範囲、フィールドなど）のいずれかを変更する必要がある場合は、代わりに **書出->地物に名前を付けて保存...** コンテキストメニューを使用し、を確認してください。 **保存したファイルを地図に追加する** オプションがチェックされています。これにより、新しいレイヤーが追加されます。

2. *ESRI* **シェープファイル** 形式を選択します。
3. ... ボタンを使用して exercise_data/shapefile/ フォルダを参照し、ファイルを places.shp として保存します。



4. *OK* を押します

   In the *Layers* panel, the temporary `place` layer is replaced with the saved `places` shapefile layer and the temporary icon next to it removed.

5. レイヤーをダブルクリックして **レイヤのプロパティ->ソース** タブを開き、 **レイヤ名** プロパティをファイル名と一致するように更新します。

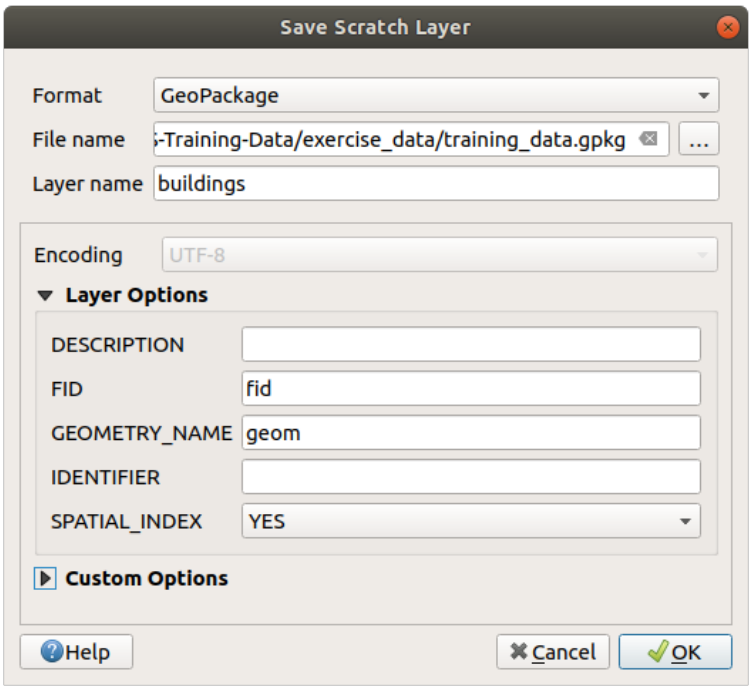
6. 他のレイヤーに対してこのプロセスを繰り返し、次のように名前を変更します：

   - `natural_water` を `water`
   - `waterway_river` を `rivers`
   - `boundary_protected_area` を `protected_areas`

   各結果データセットは、 `exercise_data/shapefile/` ディレクトリに保存する必要があります。

The next step is to create a GeoPackage file from the `building` layer to use during the course:

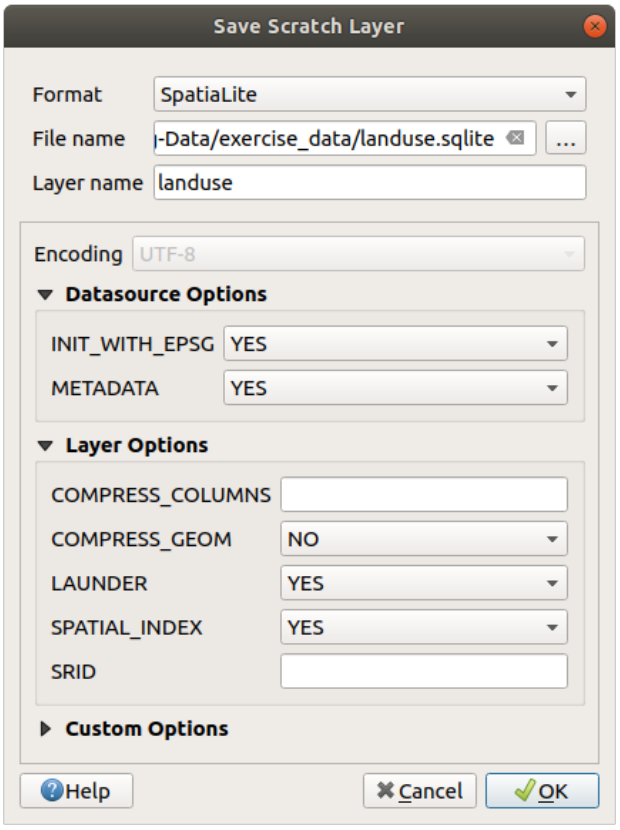
1. Click the  icon next to the `building` layer
2. *GeoPackage* 形式を選びます
3. ファイルを `training_data.gpkg` として `exercise_data/` フォルダの下に保存します
4. **レイヤ名** がデフォルトのファイル名として記入されます。それを `buildings` に置き換えます。



5. *OK* を押します
6. レイヤのプロパティダイアログで名前を変更します
7. Repeat the process with the `highway` layer, saving it as `roads` in the same GeoPackage database.

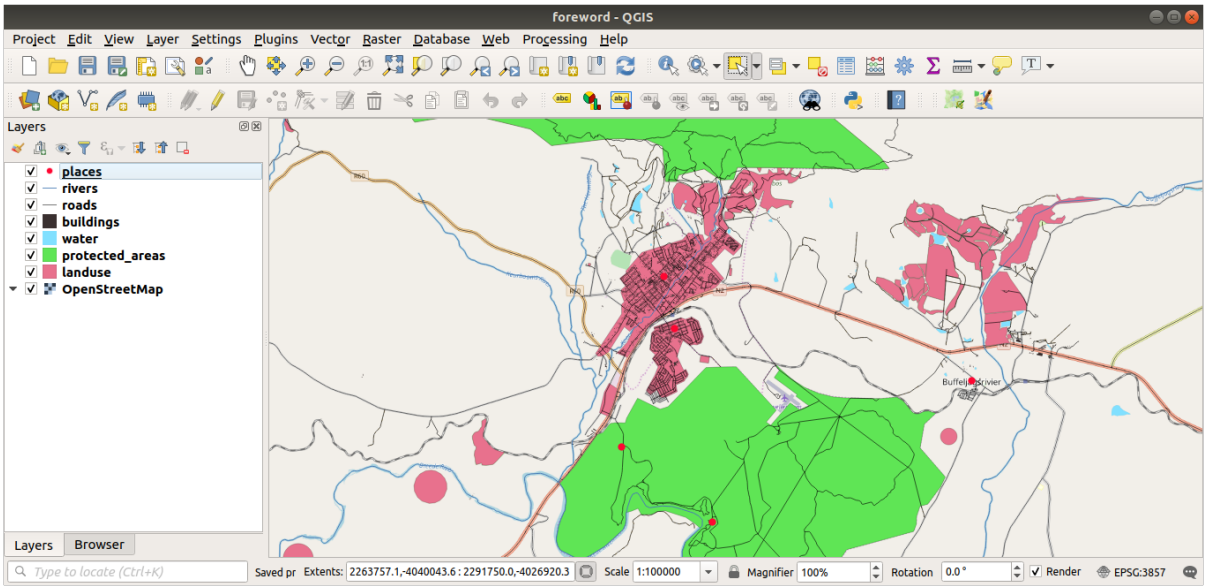
最後のステップは、残りの一時ファイルを SpatiaLite ファイルとして保存することです。

1. Click the icon next to the `landuse` layer
2. *SpatiaLite* 形式を選択します
3. そのファイルを `exercise_data/` フォルダの下に `landuse.sqlite` として保存します。デフォルトでは、 **レイヤー名** がファイル名として入力されます。変更しないでください。



4. *OK* を押します

これで、次のような地図が作成されます（レイヤーが地図に追加されると、QGIS がランダムに色を割り当てるため、シンボルは確かに大きく異なります）：



重要なことは、上に示したものと一致する 7 つのベクターレイヤーがあり、それらすべてのレイヤーにいくつかのデータがあることです。

20.2 SRTM DEM tiff ファイルを作る

モジュール **モジュール: ベクタデータを作る** および **モジュール: ラスタ** の場合、講座用に選択した地域をカバーするラスターイメージ（SRTM DEM）も必要です。

CGIAR-CGI は、 https://srtm.csi.cgiar.org/srtmdata/ からダウンロードできる SRTMDEM を提供します。

使用することを選択した地域全体をカバーする画像が必要になります。範囲座標を見つけるには、QGIS で  最大のレイヤーの範囲にズームし、ステータスバーの  **範囲** ボックスの値を選択します。 `GeoTiff` 形式のままにします。フォームに入力したら、 **ここをクリックして検索を開始**>> ボタンをクリックし、ファイルをダウンロードします。

必要なファイルをダウンロードしたら、それらを `raster/SRTM` サブフォルダの下の `exercise_data` ディレクトリに保存する必要があります。

20.3 画像の tiff ファイルを作る

モジュール **モジュール: ベクタデータを作る** で、 **★☆☆（初級レベル）理解しよう: データソース** レッスンは、生徒がデジタイズするように求められる 3 つの学校の運動場のクローズアップ画像を示します。したがって、新しい SRTMDEMtiff ファイルを使用してこれらの画像を再現する必要があります。学校の運動場を使用する義務はありません。3 つの学校の土地利用タイプを使用できます（たとえば、異なる学校の建物、遊び場、駐車場）。

参考までに、その例題データの画像はこれです:



20.4 トークンを置き換える

ローカライズされたデータセットを作成したら、最後に substitutions.txt ファイル内のトークンを置き換えて、ローカライズ版トレーニングマニュアルに適切な名前が表示されるようにします。

置き換える必要のあるトークンは次のとおりです:

- majorUrbanName ：これはデフォルトで「スウェレンダム」になります。お住まいの地域の主要な町の名前に置き換えてください。
- schoolAreaType1 ：これはデフォルトで「陸上競技場」になります。お住まいの地域で最大の学校エリアタイプの名前に置き換えてください。
- largeLandUseArea ：これはデフォルトで「ボンテボック国立公園」になります。お住まいの地域の大きな土地利用ポリゴンの名前に置き換えてください。
- srtmFileName ：これはデフォルトで srtm_41_19.tif になります。これを SRTMDEM ファイルのファイル名に置き換えます。

- localCRS：これはデフォルトで WGS 84 / UTM34S になります。これをお住まいの地域の正しい CRS に置き換える必要があります。