
QGIS Training Manual

リリース 2.14

QGIS Project

2017年08月08日

Contents

1	コースのイントロダクション	1
1.1	序文	1
1.2	練習データの準備	3
2	Module: インタフェース	11
2.1	Lesson: はじめに	11
2.2	Lesson: 第一レイヤの追加	12
2.3	Lesson: インタフェースのあらまし	14
3	Module: 基礎的な地図作成	17
3.1	Lesson: ベクタデータの操作	17
3.2	Lesson: シンボロジー	21
4	Module: ベクタデータの分類	51
4.1	Lesson: 属性データ	51
4.2	Lesson: ラベルツール	52
4.3	Lesson: 分類	71
5	Module: マップの作成	91
5.1	Lesson: 地図組版を使用する	91
5.2	課題 1	100
6	Module: ベクタデータの作成	103
6.1	Lesson: Creating a New Vector Dataset	103
6.2	Lesson: 地物のトポロジ	113
6.3	Lesson: フォーム	125
6.4	Lesson: アクション	136
7	Module: ベクタ分析	149
7.1	Lesson: データの再投影と変形	149
7.2	Lesson: ベクタ分析	158
7.3	Lesson: ネットワーク分析	176
7.4	Lesson: 空間統計	187
8	Module: ラスタ	207
8.1	Lesson: ラスタデータの操作	207
8.2	Lesson: ラスタシンボロジーの変更	213
8.3	Lesson: 地形解析	222
9	Module: 分析の完了	235
9.1	Lesson: ラスタのベクタへの変換	235
9.2	Lesson: 分析を組み合わせる	238
9.3	課題	239
9.4	Lesson: 補足の実習	239
10	Module: プラグイン	253
10.1	Lesson: プラグインのインストールと管理	253
10.2	Lesson: 役に立つ QGIS プラグイン	257

11 Module: オンラインリソース	267
11.1 Lesson: Web Mapping Services	267
11.2 Lesson: Web Feature Services	276
12 Module: GRASS	285
12.1 Lesson: GRASS Setup	285
12.2 Lesson: GRASS ツール	296
13 Module: 評価	305
13.1 ベースマップの作成	305
13.2 データを分析	307
13.3 最終的な地図	308
14 Module: 森林アプリケーション	309
14.1 Lesson: 林業モジュール発表	309
14.2 Lesson: 地図を地理参照する	310
14.3 Lesson: 林分をデジタル化する	315
14.4 Lesson: 林分を更新する	329
14.5 Lesson: 体系的なサンプリングの設計	340
14.6 Lesson: アトラスツールで詳細な地図を作成します	346
14.7 Lesson: 森林パラメータの計算	361
14.8 Lesson: レーザー測量データからの DEM	367
14.9 Lesson: 地図プレゼンテーション	376
15 Module: PostgreSQL でのデータベース概念	383
15.1 Lesson: データベースの概要	383
15.2 Lesson: データモデルの実装	388
15.3 Lesson: モデルへのデータの追加	393
15.4 Lesson: 検索	396
15.5 Lesson: ビュー	400
15.6 Lesson: ルール	401
16 Module: 空間データベースの概念と PostGIS	403
16.1 Lesson: PostGIS の設定	403
16.2 Lesson: 単純地物モデル	406
16.3 Lesson: インポートとエクスポート	411
16.4 Lesson: 空間検索	413
16.5 Lesson: ジオメトリの構成	421
17 QGIS プロセッシングガイド	429
17.1 はじめに	429
17.2 開始前の重要な警告	429
17.3 プロセッシングフレームワークを構成する	431
17.4 最初のアルゴリズムを実行します。ツールボックス	433
17.5 より多くのアルゴリズムとデータタイプ	436
17.6 空間参照系。再投影	443
17.7 選択	446
17.8 外部のアルゴリズムを実行します	448
17.9 処理ログ	453
17.10 ラスタ電卓。無データ値	454
17.11 ベクトル計算機	458
17.12 範囲の定義	462
17.13 HTML 出力	466
17.14 最初の分析例	468
17.15 ラスターレイヤーをクリッピングし、マージ	477
17.16 水文解析	486
17.17 グラフィカルなモデラーで始まります	497
17.18 より複雑なモデル	508
17.19 モデラーでの数値計算	513

17.20	モデル内のモデル	518
17.21	データ補間	519
17.22	その他の補間	527
17.23	アルゴリズムの反復実行	533
17.24	アルゴリズムのより多くの反復実行	538
17.25	バッチ処理インターフェース	540
17.26	バッチ プロセッシング インタフェースのモデル	544
17.27	その他のプログラム	545
17.28	補間と等高線作成	546
17.29	ベクトルの簡素化とスムージング	547
17.30	太陽光ファームを計画	548
17.31	処理で R スクリプトを使用する	548
17.32	処理スクリプトで R の構文	557
17.33	処理のための R 構文の概要表	560
17.34	地滑りを予測	561
18	Module: QGIS での空間データベースの利用	563
18.1	Lesson: QGIS ブラウザーにおけるデータベースについて	563
18.2	Lesson: Using DB Manager to work with Spatial Databases in QGIS	567
18.3	Lesson: QGIS における SpatiaLite データベースの操作	579
19	Module: インタフェース	583
19.1	概要	583
19.2	Lesson: Python の基礎	583
20	付録：このマニュアルに貢献する	585
20.1	リソースのダウンロード	585
20.2	マニュアルの形式	585
20.3	モジュールの追加	585
20.4	レッスンの追加	586
20.5	セクションの追加	587
20.6	結論を追加	588
20.7	また、読取部を追加します。	588
20.8	何の次のセクションを追加します。	588
20.9	マークアップを使用する	589
20.10	ありがとうございました！	591
21	回答シート	593
21.1	Results For 最初のレイヤを追加	593
21.2	Results For インタフェースのあらまし	593
21.3	Results For ベクターデータで作業	594
21.4	Results For シンボル	594
21.5	Results For 属性データ	600
21.6	Results For ラベルツール	601
21.7	Results For 分類	605
21.8	Results For 新しいベクターデータセットの作成	606
21.9	Results For ベクター分析	610
21.10	Results For ラスタ分析	621
21.11	Results For 分析を完了させる	626
21.12	Results For WMS	632
21.13	Results For データベースの概念	635
21.14	Results For 空間問合せ	638
21.15	Results For ジオメトリ建設	639
21.16	Results For 簡易機能モデル	640
22	索引とテーブル	641

Chapter 1

コースのイントロダクション

1.1 序文

1.1.1 背景

GIS にジェントルはじめは : *ref*: 2008 年に、私たちは打ち上げ 1、専門用語と新しい用語で過負荷にすることなく、GIS について学びたい人のための完全に自由、オープンコンテンツリソース。これは、南アフリカ政府が主催して、世界中の人々は、彼らがそうで、大学のトレーニングコースを実行して自分自身に GIS を教えるとする材料を使用しているか、私たちに伝えるために、私たちへの書き込みで、驚異的な成功を収めています。ジェントルはじめに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テキスト（私たちはすべての例で QGIS を使用している）ことを目指しています。QGIS アプリケーションの詳細な機能の概要を提供して QGIS マニュアルもあります。しかし、チュートリアルのように構成されるのではなく、リファレンスガイドとして、Linfinity コンサルティング CC で。私たちは頻繁にトレーニングコースを実行し、第三のリソースが必要であると認識している - この作品を作るために私たちが促し - トレーナー・研修形式で QGIS の重要な側面を学ぶことを通じて、読者の順番をリード 1。

QGIS、PostgreSQL と PostGIS の上で 5 日のコースをたどるために必要なすべての材料を提供することを、このトレーニング・マニュアルは目的とします。コースは同様に初心者向きで、中間体と上級者のための内容も構築されており、テキストを通して多くのエクササイズを注釈がついている答えを完備したようにします。

1.1.2 ライセンス



Linfinity コンサルティング CC によるフリー QuantumGIS トレーニングマニュアル。Linfinity から以前のバージョンに基づいており、クリエイティブコモンズの表示 4.0 国際 <<http://creativecommons.org/licenses/by/4.0/>> 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たアクセス権は以下で入手可能です。

自由にこの作品をコピーして、修正して、分担し直すことができる自由なライセンス下で、我々はこの QGIS トレーニング・マニュアルを出版しました。ライセンスの完全なコピーは、この文書の終わりに利用できます。単純な条件として、使用ガイドラインは以下の通りです：

- あなた自身の作品として、この作品を表現したり、この作品から任意の著作者のテキストやクレジットを削除することはできません。
- それが提供されたときのものより制限された権限下で、この作品を再配布することはできません。
- 仕事に実質的な部分を追加し、プロジェクト（少なくとも 1 つの完全なモジュール）に戻すことが貢献している場合、あなたはこの文書の著者リストの最後に自分の名前を追加できます（フロントページに表示されます）

- マイナーチェンジと訂正に貢献したい場合は、下のコントリビュータリストに、あなたを追加すればいいかもしれません。
- あなたがその全体において、この文書を翻訳する場合は、”Translated by Joe Bloggs” 形式で制作者リストにあなたの名前を追加することがあります。
- あなたはモジュールまたはレッスンをスポンサーした場合、あなたは著者にレッスンコントリビューットの始めに謝辞を含めるよう要求できます。

ノート: このレッスンは MegaCorp によって主催されています。

- あなたがこのライセンスの下ですることを確信できないならば、office@linfiniti.com を通じて我々に連絡してください。そして、我々はあなたがするつもりであることが受け入れられるかどうか、あなたにアドバイスします。
- このような <http://lulu.com> として自費出版サイトの下に、この作品を公開する場合、我々はあなたが QGIS プロジェクトに収益を寄付することを要求します。
- 作者の表現された許可があるときを除いて、この作品を商業化してはいけません。明確にするため、商業化によって我々は (例えば雑誌で記事として使用するためのコンテンツを販売)、市販の派生作品を作成し、あなたが利益のために販売しないかもしれないことを意味します。すべての利益が QGIS プロジェクトに与えられている場合は例外です。トレーニングコースを実施するときには (そして我々はそうすることをお勧めします)、コース自体が本質的に商業用であっても、この作品は教科書として使用できます。言い換えれば、教科書としてこの作品を使用するトレーニングコースを実行してお金を稼ぐのは歓迎されますが、本自体の販売をオフに利益ないかもしれない - このようなすべての利益は QGIS に戻って貢献しなければなりません。

1.1.3 主催のチャプター

この作品は、決してあなたは QGIS で行うことができますし、我々はすべてのギャップを埋めるために新しい材料を追加するために他の人を励ますすべてのものについての完全な論文です。Linfiniti コンサルティング CC。また、生成されたすべてのような作品は、コアコンテンツの一部になるべきと同じライセンスの下で公開されることは理解した上で、商用サービスとしてあなたのための追加の材料を作成することができます。

1.1.4 作者

- Rüdiger Thiede (rudi@linfiniti.com) - Rudi は QGIS 教材や PostGIS マテリアルの一部を書いています。
- Tim Sutton (tim@linfiniti.com) - Tim は監督やガイドプロジェクトおよび PostgreSQL と PostGIS の部分を共同執筆しています。Tim はまた、このマニュアルで使用するカスタム sphinx テーマを執筆しています。
- Horst Dster (horst.duester@kappasys.ch) - Horst は、PostgreSQL と PostGIS の部分を共同執筆
- Marcelle Sutton (marcelle@linfiniti.com) - Marcelle は、この作品の作成中にプルーフリーディングおよび編集上の助言を提供しました。

1.1.5 個人のコントリビュータ

あなたの名前をここに！

1.1.6 スポンサー

- Cape Peninsula University of Technology

1.1.7 データ

ノート: マニュアル全体で使用されるサンプルデータはこちらからダウンロードできます: <https://github.com/qgis/QGIS-Training-Data/archive/QGIS-Training-Data-v1.0.zip>。 ** exercise_data ** という名前のフォルダにファイルを保存できます。

このリソースに付属のサンプルデータは、自由に利用可能で、次のソースに由来しています:

- OpenStreetMap (<http://www.openstreetmap.org/>) 由来の道路と地名データセット
- Property boundaries (urban and rural), water bodies from NGI (<http://www.ngi.gov.za/>)
- SRTM DEM from the CGIAR-CGI (<http://srtm.csi.cgiar.org/>)

1.1.8 最新バージョン

QGIS ドキュメントの Web サイト (<http://docs.qgis.org>) の一部であるオンライン版を訪問すれば、このドキュメントの最新バージョンをいつでも入手できます。

ノート: ドキュメントやトレーニングマニュアルのオンラインおよび PDF バージョンへのリンクがあります。

Tim Sutton, May 2012

1.2 練習データの準備

トレーニングマニュアルと一緒に提供されているサンプルデータは 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。

誰もが無理なく、このデータセットを使用することができますが、自分の国や故郷からのデータを使用することを好むかもしれません。あなたがそうすることを選択した場合は、ローカライズされたデータは、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、お住まいの地域ではご使用になれない場合があり、より複雑なデータソースを使用しています。

ノート: このプロセスはコース召集者、または、それらのコースのために局所的なサンプル・データセットをつくりたいより多くの QGIS ユーザーを経験しました。デフォルト・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。しかし、デフォルトデータセットを置き換えたいなら、あなたはこれらの指示に従うかもしれません。

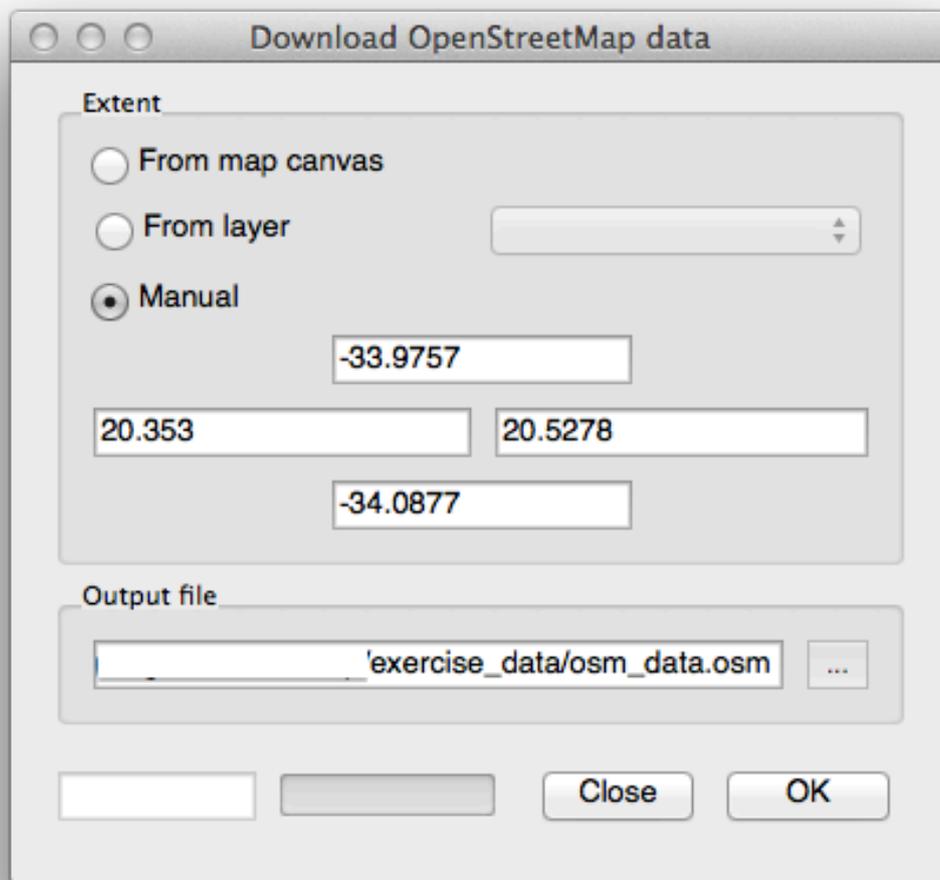
ノート: マニュアル全体で使用されるサンプルデータはこちらからダウンロードできます: <https://github.com/qgis/QGIS-Training-Data/archive/QGIS-Training-Data-v1.0.zip>。 ** exercise_data ** という名前のフォルダにファイルを保存できます。

1.2.1 Try Yourself

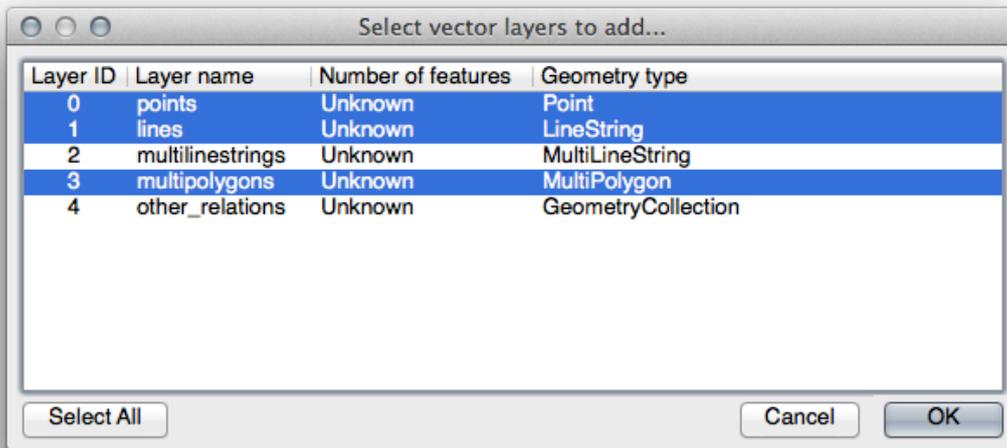
ノート: 以下の手順では、QGIS の十分な知識を持っており、教材として使用されることを意図していないと仮定します。

あなたのコースにローカライズされたデータに設定されたデフォルトのデータを交換したい場合、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度な混合地であり、河川や水面、道路、地域の境界 (自然保護区や農場など) などが分かれています。

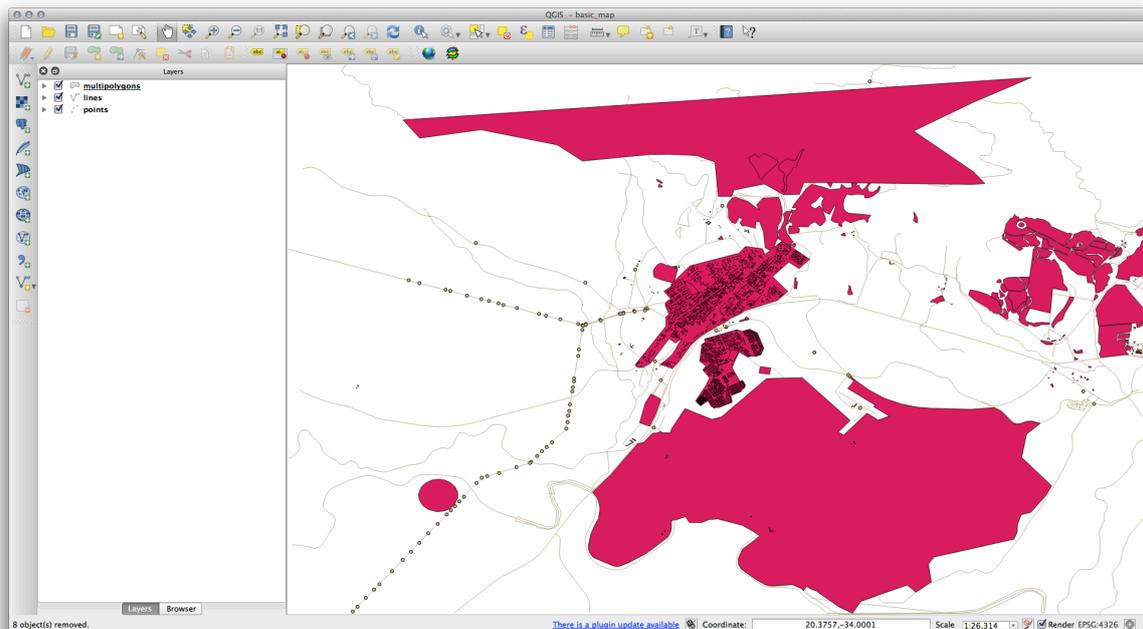
- QGIS プロジェクトを開く
- *Vector* メニューのドロップダウンで *OpenStreetMap->データをダウンロード* を選択してください。その後、使用したい地域の座標を手動で入力することもできますし、既存のレイヤを使用して座標を設定することもできます。
- 結果の保存場所となる.osm ファイルを選択し、 *Ok* をクリックします。



- *Add Vector Layer* ボタンを使って .osm ファイルを開くことができます。ブラウザウィンドウ内で *All files* を選択するか、QGIS のウィンドウにドラッグ&ドロップすることができます。
- 表示されたダイアログで、 *other_relations* と *multilinestrings* レイヤー を除くすべてのレイヤを選択します。



これは、OSM の命名規則に関連した地図に 3 つのレイヤーをロードします（ベクトルデータを見るためにズームイン/アウトする必要があるかもしれません）。



これらのレイヤから利用するデータを抽出し、それらの名前を変更し、対応するシェープファイルを作成する必要があります。

- まず、multipolygons 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。
- *General* タブの中で *Query Builder* をクリックし、*Query builder* ウィンドウを開きます。

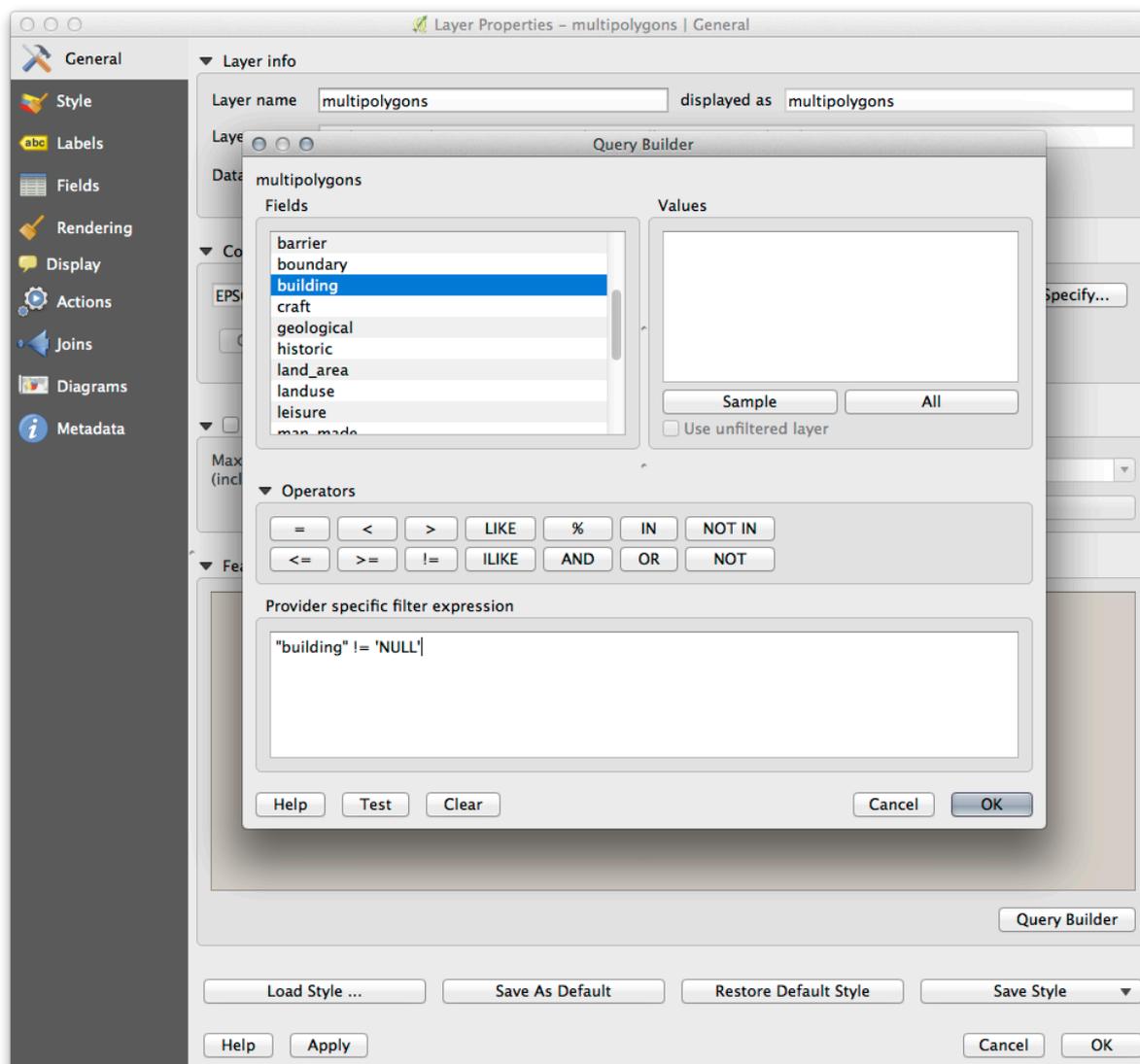
このレイヤは、トレーニングマニュアルで使用するために抽出する必要がある 3 つのフィールドを含みます：

- building
- :kbd:`natural` (特に、水部)
- landuse

あなたの地域は、お住まいの地域が得られます結果の種類を確認するために含まれているデータをサンプリングできます。「土地利用」が結果を返さないことを発見した場合は、それは自由に除外して感じます。

あなたは、必要となるデータを抽出するために、各フィールドのフィルタ式を作成する必要があります。ここでは例として、「建物」フィールドを使用します。

- テキストエリアに次の式を入力します：`building != "NULL"`。そして *Test* をクリックして、クエリが返されます。どのように多くの結果を参照してください。お住まいの地域のために戻ってきたようなデータ OSM 見るために 属性 Table 結果の数が少ない場合は、レイヤーのを見てしたいことがあります。



- *Ok* をクリックし、地図から建物でない属性のレイヤーが削除されていることを確認します。
- コース中に使用するため、結果データはシェープファイルとして保存する必要があります。
- *multipolygons* レイヤを右クリックし 名前を付けて保存... を選択します
 - ファイルの種類が ESRI Shapefile であることを確認し、”epsg4326” ディレクトリの下に、新しい *exercise_data* ディレクトリにファイルを保存します。
 - 必ず *No Symbolology* が選択されていることを確認します (後にコースの一部でシンボルを追加します)。
 - You can also select *Add saved file to map*.

buildings がマップに追加されると。次の式を利用して *natural* と *landuse* フィールドのための手順を繰り返すことができます。

ノート: 次のフィルタ式を実行する前に *multipolygons* レイヤから (*Layer properties* ダイアログを使って) 前のフィルタをクリアしてください!

- natural: “natural = ‘water’”
- landuse: “landuse != ‘NULL’”

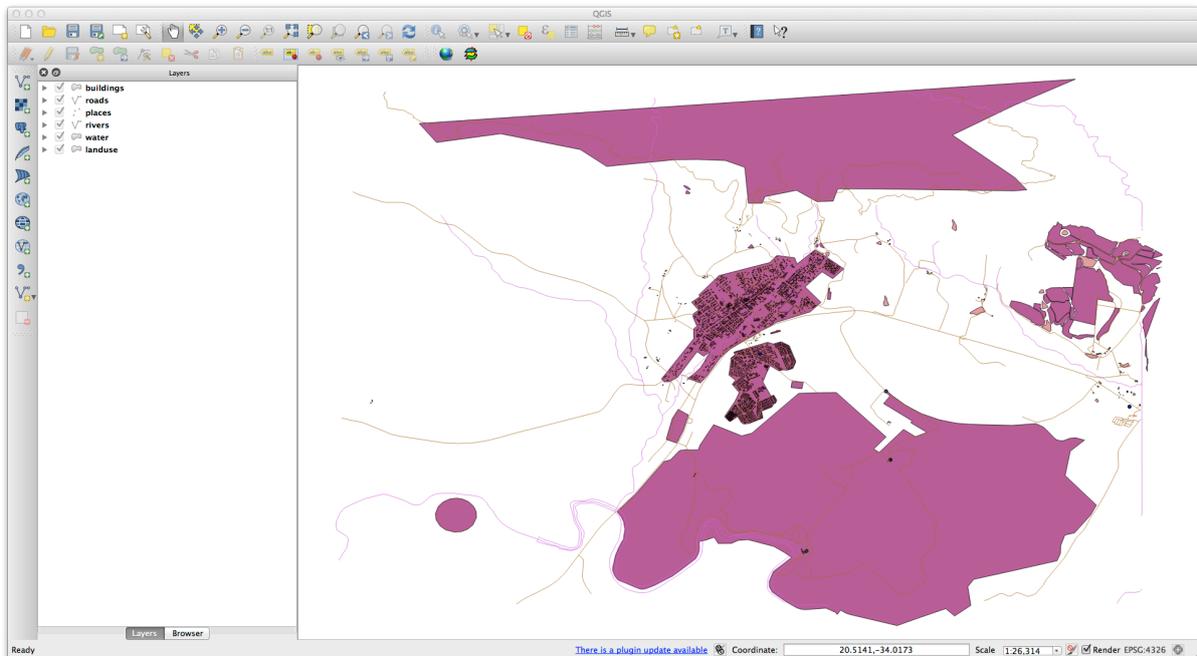
得られた各データセットは、”epsg4326” ディレクトリの下の新しい *exercise_data* ディレクトリ内に保存する必要があります (つまり、”water”、”landuse”)。

その後、*lines* と *points* のフィールドを抽出しレイヤーに対応するディレクトリへ保存する必要があります。

- *lines*: “highway != ‘NULL’” to roads, and “waterway != ‘NULL’” to rivers
- *points*: “place != ‘NULL’” to places

上記のデータを抽出することができたら *multipolygons*、*lines*、*points* の各レイヤーを削除することができます。

これで、次のようなマップになっている必要があります (シンボルは異なるものになりますが、それは問題ありません)。



重要なことは、上に示したものと同じ6つのレイヤーがあり、すべてのレイヤーはいくつかのデータを持っていることです。

最後のステップは、もちろん使用するために *landuse* レイヤから *spatialite* ファイルを作成することです。

- *landuse* レイヤを右クリックし、名前を付けて保存... を選択します。
- フォーマットとして *SpatialLite* を選択し、”epsg4326” ディレクトリの下に *landuse* というファイルを保存します。
- Click *Ok*.
- Delete the *landuse.shp* and its related files (if created).

1.2.2 Try Yourself SRTM DEM tiff ファイルの作成

モジュール 6 (ベクトルデータを作成) し、モジュール 8 (ラスタ) のために、コースのために選択した領域を覆うラスタ画像 (SRTM の DEM を) も必要になります。

The CGIAR-CGI (<http://srtm.csi.cgiar.org/>) provides some SRTM DEM you can download from <http://srtm.csi.cgiar.org/SELECTION/inputCoord.asp>.

使用することを選択した領域全体をカバーする画像が必要になります。トレーニングマニュアルと同じデータを保持する場合は、そうでない場合は、範囲を適応させる、上記 `set_osm_region_` の図の範囲を使用できます。GeoTiff フォーマットのままにしてください。フォームが満たされたら、検索を開始するにはここをクリック>> ボタンをクリックしてファイル (複数可) をダウンロードしてください。

必要なファイルをダウンロードしたら、それらを ラスター / SRTM サブフォルダの下 `:file:'exercise_data'` ディレクトリに保存する必要があります。

1.2.3 Try Yourself 画像の TIFF ファイルを作成する

モジュール 6、レッスンでは 1.2 は、学生がデジタル化するように要求されている 3 つの学校のスポーツフィールドのクローズアップ画像を示します。そのため、あなたの新しい SRTM の DEM の TIFF ファイル (複数可) を使用してこれらのイメージを再現する必要があります。学校のスポーツフィールドを使用する義務はありません: 任意の 3 校土地利用タイプは、(例えば、異なる校舎、運動場や駐車場) を使用することができます。

参考までに、サンプルデータの画像があります:





1.2.4 Try Yourself トークンの置き換え

`conf.py` ファイルを適切な名前は、トレーニングマニュアルのローカライズ版で表示されるように、ローカライズされたデータセットを作成したので、最後のステップはトークンを交換することです。

置き換える必要のあるトークンは次のとおりです:

- `majorUrbanName`: これはデフォルトは "Swellendam" ですが、選んだ範囲の主要な町の名前に置き換えます。
- `schoolAreaType1`: これはデフォルトで "athletics field" ですが、選んだ範囲で最大の学校エリアの名前に置き換えます。
- `largeLandUseArea`: これはデフォルトで "Bontebok National Park" ですが、選んだ範囲で広範囲な土地利用ポリゴン名に置き換えます。
- `srtmFileName`: これはデフォルトで `srtm_41_19.tif` ですが、SRTM DEM ファイルのファイル名に置き換えてください。
- `localCRS`: このデフォルトは WGS 84 / UTM 34S です。これをあなたの地域に合わせて、正しい CRS に置き換える必要があります。

Chapter 2

Module: インタフェース

2.1 Lesson: はじめに

私達のコースへようこそ! これより QGIS の簡単で効率的な使い方について、数日をかけて教えていきます。もし QGIS を使うのが初めての場合は、はじめかたについても説明します。もし既に使用している場合であっても、あなたが使いたい全ての機能を QGIS で実現する方法についてより知ることができるはずです。

ここでは QGIS のユーザーインターフェースだけでなく、プロジェクトについても説明していきます。

このセクションを終えれば、QGIS 画面の要素を性格に把握することができ、それぞれどのように機能するのか、そして shapefile の読み込み方を知ることができるようになります。

警告: このコースでは GIS データセットの追加、消去、そして変更のしかたについて紹介します。練習用にデータセットを用意してあります。ここで紹介する機能を実際に自分のデータに使用する前には必ずデータのバックアップを取っておいて下さい。

2.1.1 チュートリアルを使いかた

こんな感じの文字は、画面上でクリックできることを示します。

looks → *like* → *this* のテキストはメニューの指示です。

この種のテキストは、あなたがタイプできるコマンド、パス、ファイル名のようなものを指します。

2.1.2 階層型コースの目的

このコースでは、異なるユーザの経験レベルの内容を用意する。どのカテゴリがあなたに合うと考えるかによりませんが、当然結果の異なるコースのセットを期待することができます。各カテゴリには、次のために必須である情報が含まれているので、経験レベルの、または下にあるすべての演習を行うことが重要です。



初級編

このカテゴリにおいて、コースは、理論的な GIS の知識や GIS プログラムの動作にほとんど、またはまったく経験を持っていないことを前提としています。

限られた理論的背景は、プログラムで実行されるアクションの目的を説明するために提供されますが、重点は、実行して学習することにあります。

あなたはコースを完了すると、GIS の可能性と、どのように QGIS を経由してその力を活用していくかのよりよいコンセプトを持つこととなります。



中級編

このカテゴリでは、あなたが業務知識と GIS の日常用途の経験を持つと想定します。

初心者レベルの指示に従うことでおなじみのグラウンドを提供するだけでなく、QGIS は他の GIS ソフトとは少し異なることができることに気づくでしょう。また、QGIS で分析機能を使用する方法を学習します。

このコースを完了すると、日常利用する GIS で必要とする機能のすべてのために QGIS を使うことに対して、不安を感じるはずがありません。



上級編

このカテゴリでは、あなたが GIS の利用経験があり、空間データベース、リモートサーバー上のデータの使用、あるいは分析目的でスクリプトを書くなどの知識と経験があることを想定します。

前述の2つのレベルのカテゴリによって、あなたは QGIS のインターフェイスに慣れることができ、使用したい基本的な機能にどのようにアクセスすればいいのかという知識が確実につくでしょう。またあなたは、QGIS のプラグインシステムやデータベースアクセスシステム等を利用する方法などを学ぶことができます。

コースを完成するとき、そのより先進の機能だけでなく QGIS の日常的な操作をよく知っていなければなりません。

2.1.3 なぜ QGIS?

情報がますます空間的に理解できるようになるにつれて、ツールが行える GIS の機能に関して、不足に感じることはありません。QGIS を使わずに、ほかのソフトウェアパッケージの GIS を使わなければいけない理由があるのでしょうか？

理由は以下のとおりです：

- ***ランチのように自由です***QGIS をインストールし、使うことに必要なお金はゼロです。初期費用も更新料もありません。
- ***自由であり、無料です***もしあなたがさらに機能的なものを QGIS に求めるのであれば、次のリリースにその機能が入っていることをただ祈る以上にできることがあります。その機能が発展するのを後援するか、もしくは、あなたがプログラミングに詳しくれば、あなた自身でその機能を追加することが可能です。
- ***常に成長しつづけています***誰でも機能を追加・改良できるので、QGIS は停滞することは決してありません。新しい機能の成長は、あなたが求めているものをより早く現実のものにしてくれます。
- ***豊富なヘルプおよびマニュアルが利用可能です。***あなたが何かで立ち往生している場合は、大規模なドキュメントをあなたの仲間である QGIS ユーザー、あるいは開発者をオンにすることができます。
- ***クロスプラットフォーム***QGIS は MacOS, Windows と Linux にインストールできます。

今、あなたが QGIS を使いたい理由を知っていることを、私たちはお見せすることができます。最初のレッスンは、あなたの最初の QGIS マップを作成する際にご案内します。

2.2 Lesson: 第一レイヤの追加

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マップを作成します。

このレッスンの目標: 例のようなマップを始める

ノート: このエクササイズを始める前に、QGIS は、お使いのコンピュータにインストールする必要があります。また、*QGIS データのダウンロードエリア*から `training_manual_exercise_data.zip` ファイルをダウンロード<<https://github.com/qgis/QGIS-Training-Data/archive/QGIS-Training-Data-v1.0.zip>> `_`。 20

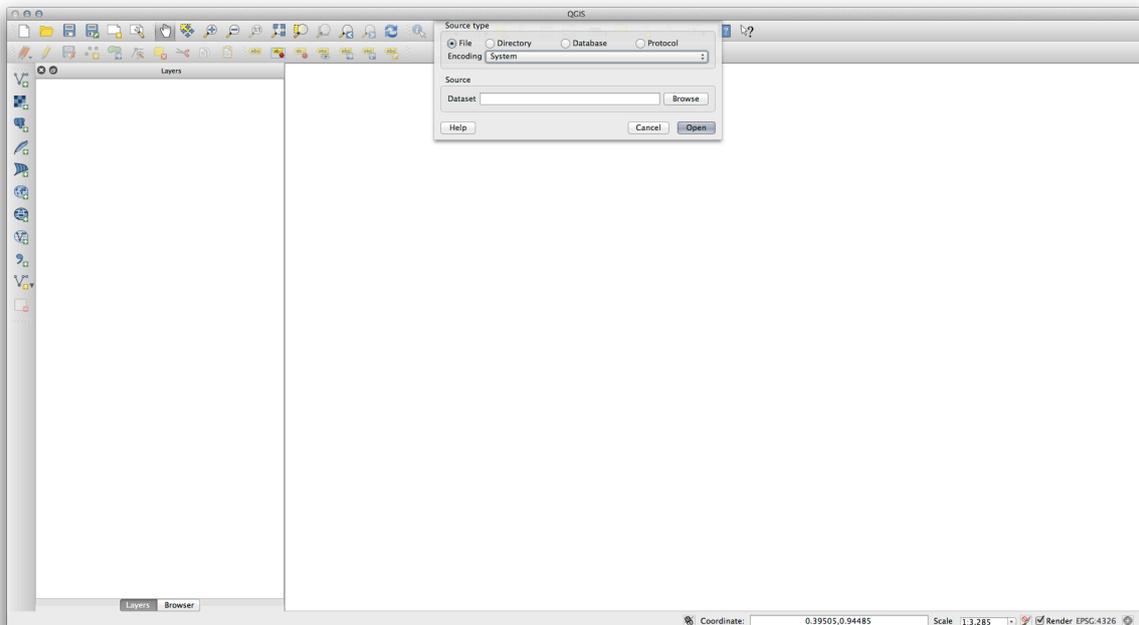
デスクトップのショートカット、メニュー、アイテムなどから QGIS を起動します。この設定はインストール時の設定に依存します。

ノート: このコースのスクリーンショットは、MacOS の上で実行されている QGIS2.0 で撮影されました。あなたの設定に応じて、あなたが遭遇した画面と仕様が異なる場合があります。しかし、利用可能なボタンはすべて同じであり、コマンドはいかなる OS 上でも動作します。このコースを使用するには QGIS2.0 (執筆時点での最新版) が必要です。

すぐにはじめましょう!

2.2.1 Follow Along: マップの準備

- QGIS を開きます。新規に空白のマップがあります。
- 探し: guilabel: 'ベクトル Layer' ボタンを追加します。|newVectorLayer|
- それをクリックすると次のダイアログが開きます:



- Click on the *Browse* button and navigate to the file `exercise_data/epsg4326/roads.shp` (in your course directory). With this file selected, click *Open*. You will see the original dialog, but with the file path filled in. Click *Open* here as well. The data you specified will now load.

おめでとうございます! ベースマップがあります。作業を保存するにはよいタイミングでしょう。

- guilabel: '保存 As' ボタン: クリックしてください |fileSaveAs|
- Save the map under `exercise_data/` and call it `basic_map.qgs`.

結果の確認

2.2.2 In Conclusion

レイヤの追加方法とベースマップの作成について学びました!

2.2.3 What's Next?

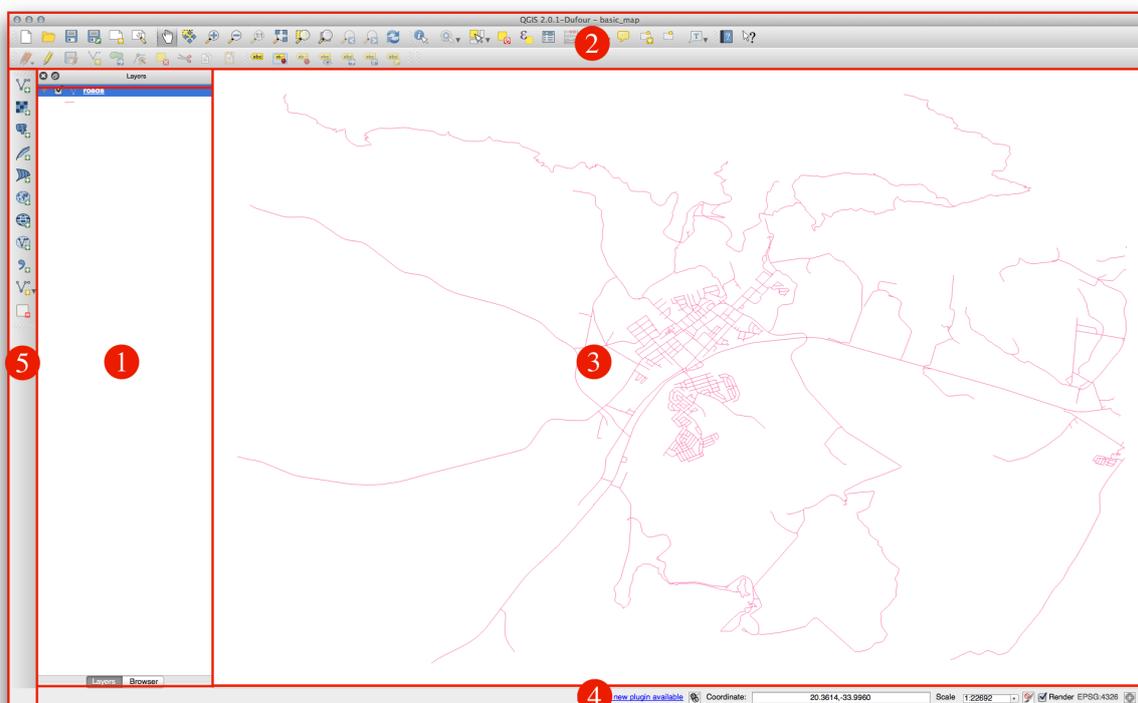
Now you're familiar with the function of the *Add Vector Layer* button, but what about all the others? How does this interface work? Before we go on with the more involved stuff, let's first take a good look at the general layout of the QGIS interface. This is the topic of the next lesson.

2.3 Lesson: インタフェースのあらまし

インターフェイスの基本的な構造を形成するメニュー、ツールバー、マップキャンバスとレイヤーリストに慣れてもらえるよう、私たちは、QGIS のユーザーインターフェイスを探る。

このレッスンの目標: QGIS のユーザインターフェイスの基礎を理解する。

2.3.1 Try Yourself:基礎



上の図で特定される要素は以下のとおりです：

1. レイヤーリスト/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



レイヤーリスト

レイヤーリストでは、いつでも、あなたが利用可能な時に、すべてのレイヤー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) 折りたたまれたアイテムを展開すると、レイヤの現在の見た目について多くの情報を提供します。

レイヤーを右クリックすると、多くの追加オプションとメニューを提供します。あなたはやがて、それらのいくつかを使用するでしょうから、その周辺を見てみます！

QGIS のバージョンによっては表示順を調整するセパレートとレイヤリストの下のチェックボックスがあります。それが見つからなくても心配いりません。それが存在するなら、現在チェックが入っています。

ノート: 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。



ブラウザパネル

QGIS ブラウザは、あなたのデータセットに簡単にナビゲートさせるための QGIS のパネルです。一般的なベクタファイル (e.g. ESRI シェープファイルまたは MapInfo ファイル)、データベース (e.g. PostGIS, Oracle, Spatialite または MSSQL Spatial) そして WMS/WFS 接続にアクセスできます。また、GRASS データの表示が可能です。



ツールバー

ツールの最もしばしば使われるセットは、基本的なアクセスのためのツールバーに変換できます。例えば、ファイルのツールバーには、保存、読み込み、印刷、および新しいプロジェクトを開始できます。設定 -> *Toolbars* メニュー を介して、必要に応じてツールバーを追加または削除、最も頻繁に使用するツールだけを参照するためのインタフェースを簡単にカスタマイズできます。

それらは、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、メニューを介してアクセス可能なままになります。削除する場合たとえば、:guilabel: (含まれています *Save* ボタン) 'File' ツールバーを、まだ *Project* メニュー、次にクリック *Save* をクリックして地図を保存できます。



マップキャンバス

マップ自体が表示されている場所です。



ステータスバー

現在の地図に関する情報を記しています。また、地図の縮尺を調整し、地図上にマウスカーソルの座標を表示することができます。

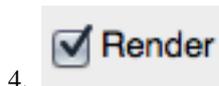
2.3.2 Try Yourself 1

上の図を参照することなく、自分自身の画面に上記の 4 つの要素を識別するようにしてください。自分の名前と機能を特定できます参照してください。数日中にそれらを使用するように、これらの要素に慣れてくるだろう。

Check your results

2.3.3 Try Yourself 2

画面にこれらの各ツールを探してみてください。彼らの目的は何ですか？



ノート: 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によって短縮できることに注意してください。どのように崩壊し、ツールバーで二重の右矢印ボタンをクリックして非表示のツールを見ることができます。しばらくの間、ツールの上にマウスを保持することにより、任意のツールの名前をツールチップを表示できます。

Check your results

2.3.4 What's Next?

今、QGIS インターフェースは利用可能なツールを使用して、地図上の改善を開始でき、どのように機能するかを見てきました！これは、次のレッスンのトピックです。

Chapter 3

Module: 基礎的な地図作成

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マップを作成します。

3.1 Lesson: ベクトルデータの操作

おそらくベクトルデータは、あなたが毎日 GIS を使うデータの中で最も一般的な種類でしょう。ここでは地理データに関するラインとポリゴンについて説明しています。ベクトルデータセット内のすべてのオブジェクトは**feature**と呼ばれ、その特徴を記述したデータに関連付けられています。

このレッスンの目標: ベクトルデータの構造と、どのようにマップへベクトルデータをロードするかを学びます。

3.1.1 Follow Along: レイヤー属性の表示

これは、オブジェクトが**どの場所**にあるのか、使用されているデータだけで表現されていない事を知っておくのが重要です。またこれらのオブジェクトが**どのようなもの**であるかがわかります。

前のレッスンで使用した *roads* レイヤーがマップにロードされている必要があります。今あなたが見ることができるのは単なる道路の位置だけです。

選択された *roads* を見えるように表示するには、レイヤーパネル:

-  ボタンをクリックします。

roads に関してより多くのデータが記録されたテーブルが表示されます。この追加データは 属性データ と呼ばれています。行はマップ上で道路をどのように表現するかを表しています。

これらの定義は一般的に GIS で使用されているので、定義を覚えておくことは不可欠です!

- ここで属性テーブルを閉じてください。

ベクトルデータは、座標平面上で点、線、多角形の面での地物を表しています。これは通常、道路や街区のような個別の地物を格納するために使用されています。

3.1.2 Follow Along: シェープファイルによるベクトルデータのロード

シェープファイルは、関連するグループのファイルに GIS データを保存することができる特定のファイル形式です。各レイヤは、同じ名前で拡張子の異なる複数のファイルで構成されます。シェープファイルはほとんどの GIS ソフトウェアで読むことができるので、あちこちに送ることが容易です。

ベクトルレイヤを追加する方法については、前のセクションでの練習を参照してください。

同じ方法で、次のマップへの ‘epsg4326’ フォルダ : ファイル : データ・セットをロードします。

- “場所”
- “水域”
- “河川”
- “建物”

結果をチェック

3.1.3 Follow Along: データベースによるベクトルデータのロード

データベースを使用すると、関連付けられた大量のデータを 1 つのファイルに保存できます。読者は既に Microsoft Access などのデータベース管理システム (DBMS) に精通しているかもしれません。GIS アプリケーションでもデータベースを利用できます。GIS 固有の DBMS (たとえば、PostGIS など) は、空間データを処理する必要があるため、余分な機能があります。

- このアイコンをクリックしてください : | addSpatialLiteLayer |

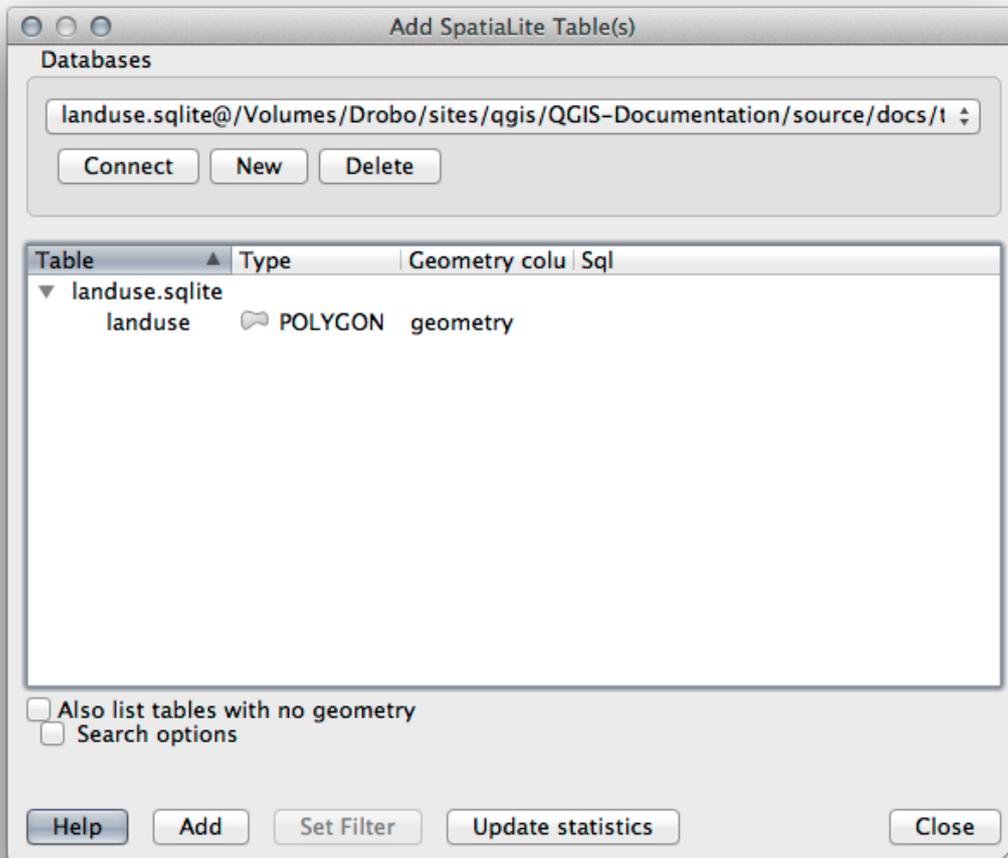
(もしそれが全く表示されない場合は、レイヤ管理 ツールバーのチェックが有効になっているか確認します。)

新しいダイアログが表示されます。このダイアログで:

- Click the *New* button.
- 同じ:file: ‘ epsg4326’ フォルダ中で:file: ‘ landuse.sqlite’ ファイルを見つける必要があります。それを選択し、:guilabel: ‘開く ‘をクリックしてください。

今、再び最初のダイアログが表示されます。ドロップダウンが三つのボタンの上を選択していることに注意してください。今お使いのコンピュータ上のデータベースファイルのパスに続いて、「landuse.sqlite @ ...」読み込みます。

- 接続 ボタンをクリックします。空のボックスでこのように見えるはず:



- Click on the `landuse` layer to select it, then click `Add`

ノート: 地図を保存することをよく忘れるのでご注意ください! 地図ファイルは、どのデータも直接は含みませんが、地図にロードされたレイヤーを記憶しています。

結果の確認

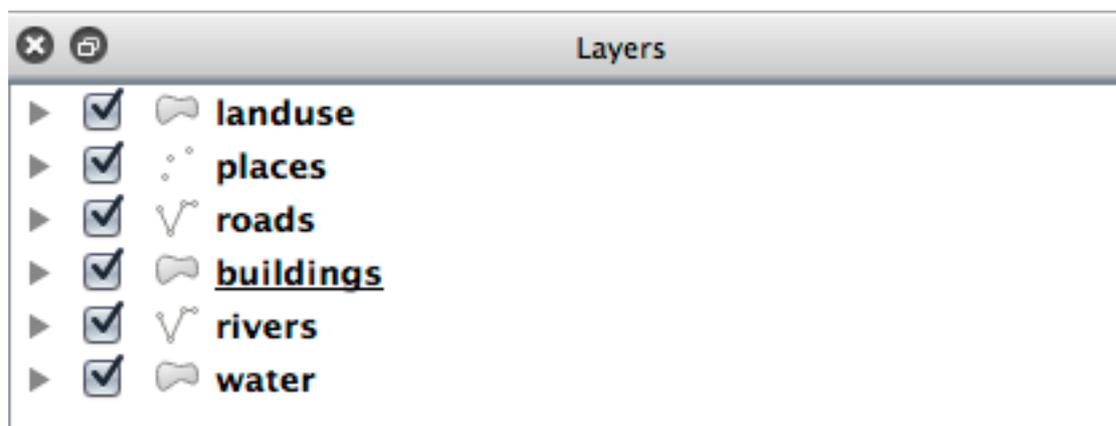
3.1.4 Follow Along: レイヤーの順序入れ替え

レイヤーリストにあるレイヤーは、特定の順序でマップ上に描かれています。リストの一番下にあるレイヤーが最初に描かれ、そして一番上のレイヤー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。

ノート: 使用している QGIS のバージョンによっては、レイヤーリストの `guiLabel`: `レンダリング順序` を制御する `読み` の下にチェックボックスがあるかもしれません。レイヤーをレイヤーリスト内で上下移動することで地図内での表示が前面や背面になるよう、これをチェック (オン) する必要があります。このオプションがない QGIS のバージョンの場合は、デフォルトでオンになっていますので、心配する必要はありません。

レイヤーがマップ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。他のレイヤーがその上にあることで道路レイヤーが完全に隠れている可能性があります。

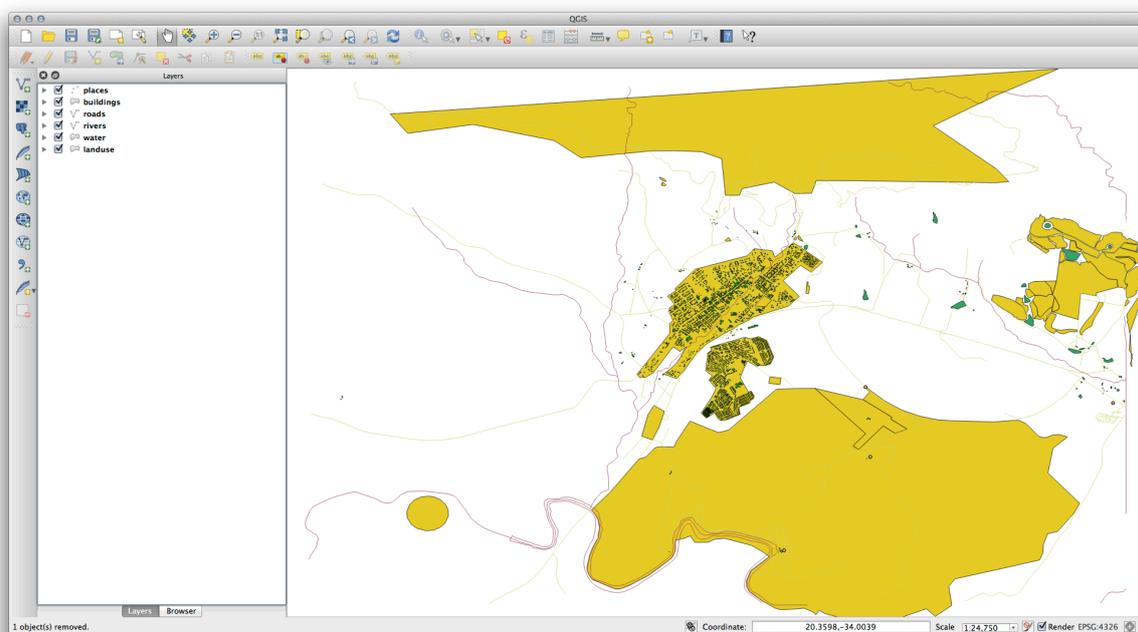
たとえば、このレイヤー順では …



... 道路や場所は市街地の*下*に実行されるので隠される結果になります。

問題を解決するために:

- レイヤーリスト中でレイヤーをクリックしてドラッグします。
- 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。

3.1.5 In Conclusion

今、いくつかの異なるソースから必要なすべてのレイヤーを追加しました。

3.1.6 What's Next?

レイヤーをロードするときに自動的に割り当てられるランダムなパレットを使用すると、現在のマップはおそらく読みやすくはありません。色や記号を自分で選択して割り当てることが望ましいでしょう。これは、次のレッスンで方法を学ぶことです。

3.2 Lesson: シンボロジー

レイヤのシンボルは、地図上の外観です。空間的な側面を持つデータを表現する他の方法以上の GIS の基本的な強みは、GIS を使用すると、作業データの動的な視覚表現を持っているということです。

したがって、(個々のレイヤのシンボルに依存する) 地図の外観は非常に重要です。地図が何を表現するかを作成した地図のエンドユーザーが簡単にわかるようにする必要があります。同様に重要なこととして、それで作業しているように、データを探索できるようにする必要があり、かつ優れたシンボルは、多くのことができます。

つまり、適切なシンボルを持つことは、高級でもなくまたは単に素晴らしいことではありません。実際には、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それが不可欠です。

****このレッスンの目標:****ベクタレイヤに、あなたが望むシンボロジーを作成できるようになる

3.2.1 Follow Along: 色の変更

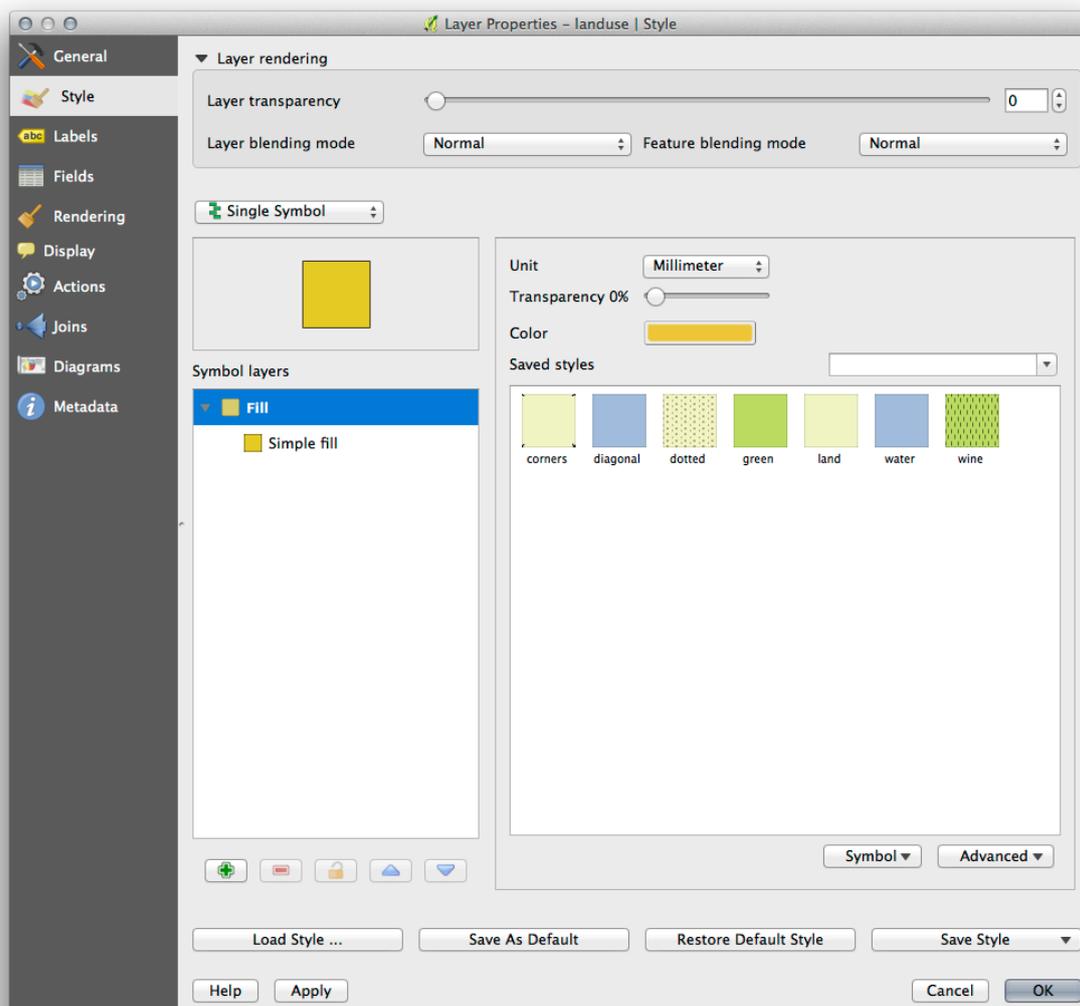
レイヤのシンボロジーを変更するには、そのレイヤのレイヤのプロパティを表示します。 *landuse* レイヤの色を変更するのを始めましょう。

- レイヤリストで *landuse* レイヤを右クリックします。
- Select the menu item *Properties* in the menu that appears.

ノート: デフォルトでは、レイヤリストでリスト上でダブルクリックすることで、レイヤプロパティにアクセスできます。

プロパティ ウィンドウにて:

- 左のスタイル タブを選択します。



- カラー ラベルの隣にある色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。

- グレーを選択して、OK をクリックします。
- レイヤプロパティ ウィンドウで再度 OK をクリックし、色の変更がレイヤに適用されたことを確認するでしょう。

3.2.2 Try Yourself

water レイヤを薄い青色に変更しましょう。

結果の確認

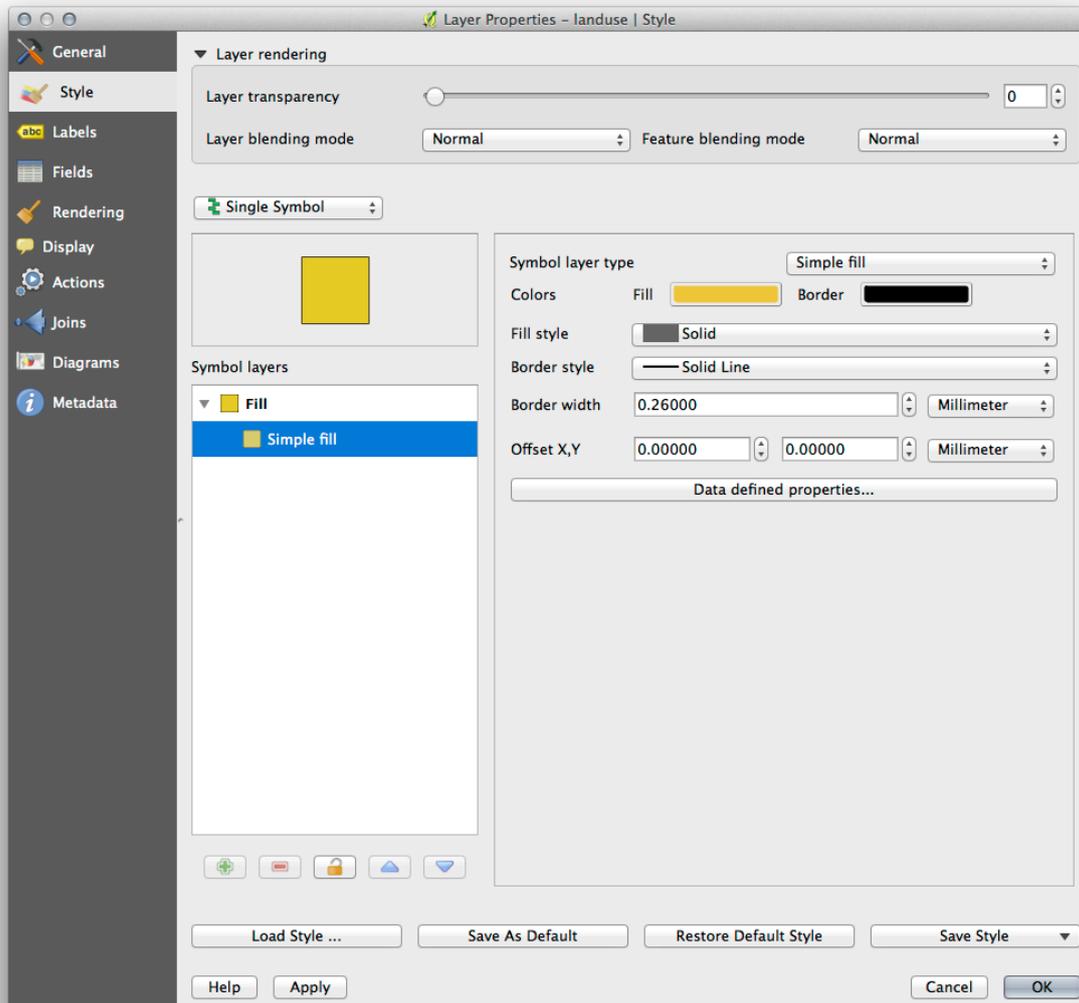
3.2.3 Follow Along: シンボル構造の変更

これはこれまでのところ良いものですが、ちょうどその色よりも、レイヤのシンボルに適したものがあります。次に、視覚的に雑然となるように様々な土地利用エリア間のラインを削除したいと思います。

- Open the *Layer Properties* window for the *landuse* layer.

スタイル タブで、以前と同じ種類のダイアログを確認します。しかし、今回はただすぐに色を変更する以上のことを行っています。

- シンボルレイヤー パネルの中で、(必要な場合は):`guiabel:塗り` ‘ドロップダウンを展開し、:`guiabel:単`純塗りつづし オプションを選択します :



- ボーダー *style* ドロップダウンをクリックします。現時点では、それは短い線と言葉を示すべきである:`guiabel:ソリッド Line`。
- これを ペンなし に変更します。
- Click *OK*.

今、*landuse* レイヤでは、エリア間にいかなるラインを持ちません。

3.2.4 Try Yourself

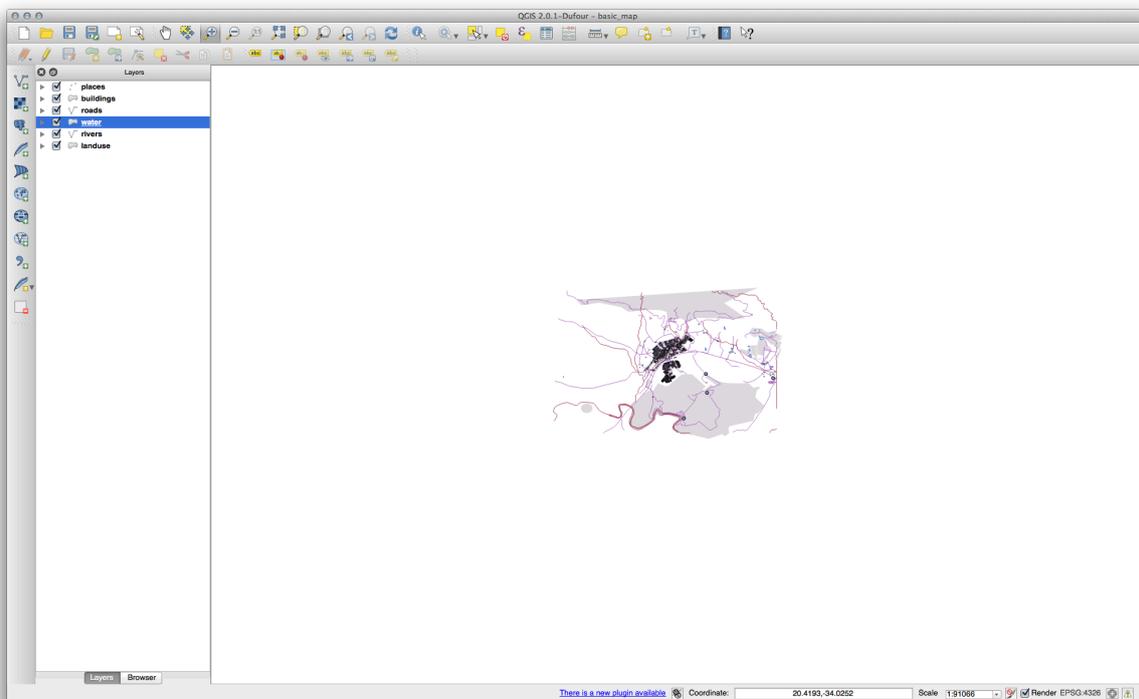
- 濃い青の外形線を持つため、*water* レイヤのシンボロジーを再度変えます。
- 水路の表示を明確に表現するため *rivers* レイヤのシンボロジーを変更します。

結果の確認

3.2.5 Follow Along: スケールにもとづく表示

時には、レイヤーが与えられた縮尺に適していないとわかりることがあります。例えば、すべての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、不適切な縮尺ではデータセットを非表示にすることができるようになりたいです。

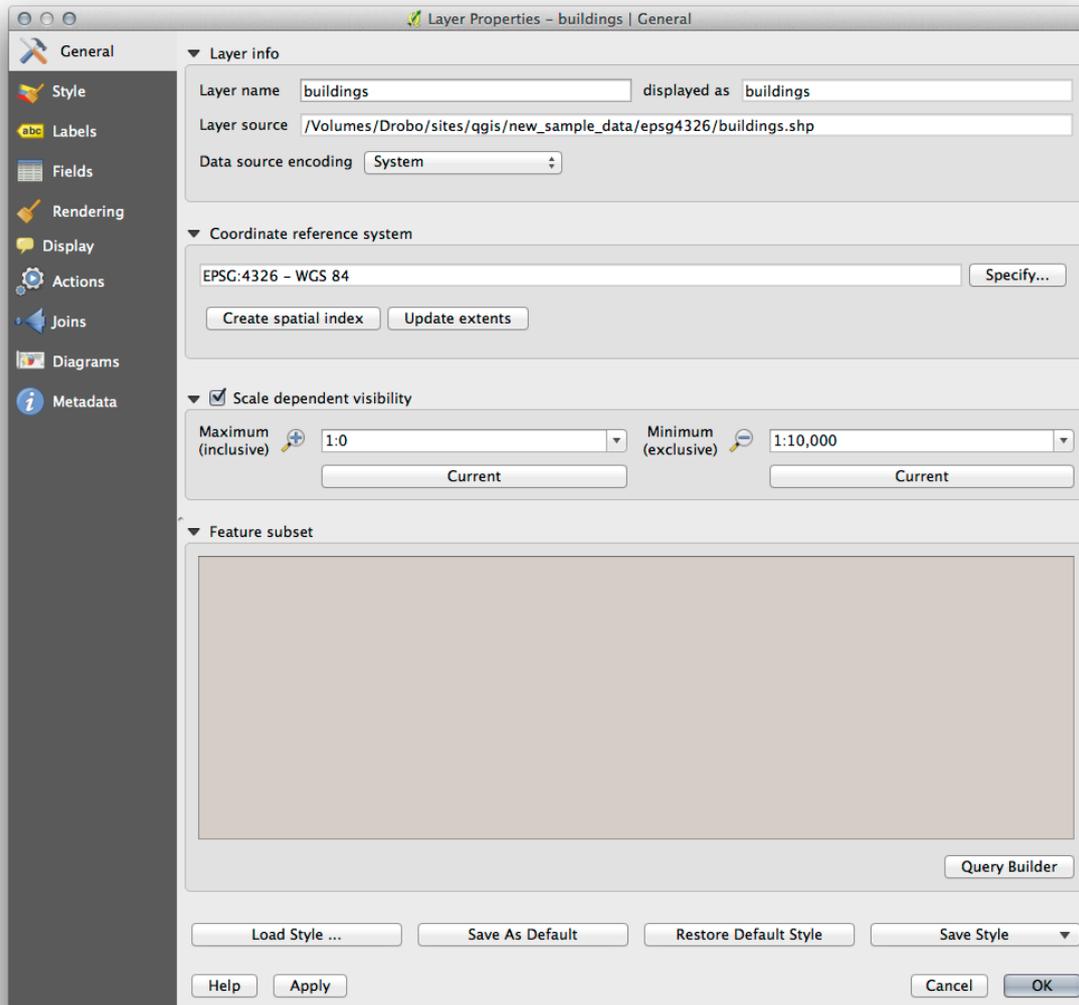
現在の場合、小縮尺で視界から建物を非表示にすると決定できます。このマップは、たとえば...



...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。

スケールに基づいたレンダリングを可能にする

- Open the *Layer Properties* dialog for the *buildings* layer.
- 一般情報 タブをアクティブにします。
- チェックボックスをクリックすることで、スケールベースのレンダリングを有効にする : guilabel : スケール依存 *visibility* :



- *Minimum* '値 : KBD : guilabel : 変更し 1 : 10,000'.
- Click *OK*.

地図で拡大・縮小し *buildings* レイヤが現れたり消えるのに注目することをテストします。

ノート: インクリメントにズームインするために、マウスホイールを使うことができます。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:



3.2.6 Follow Along: シンボルレイヤの追加

今、レイヤーのための単純なシンボルを変更する方法を知っているので、次のステップは、より複雑なシンボル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。

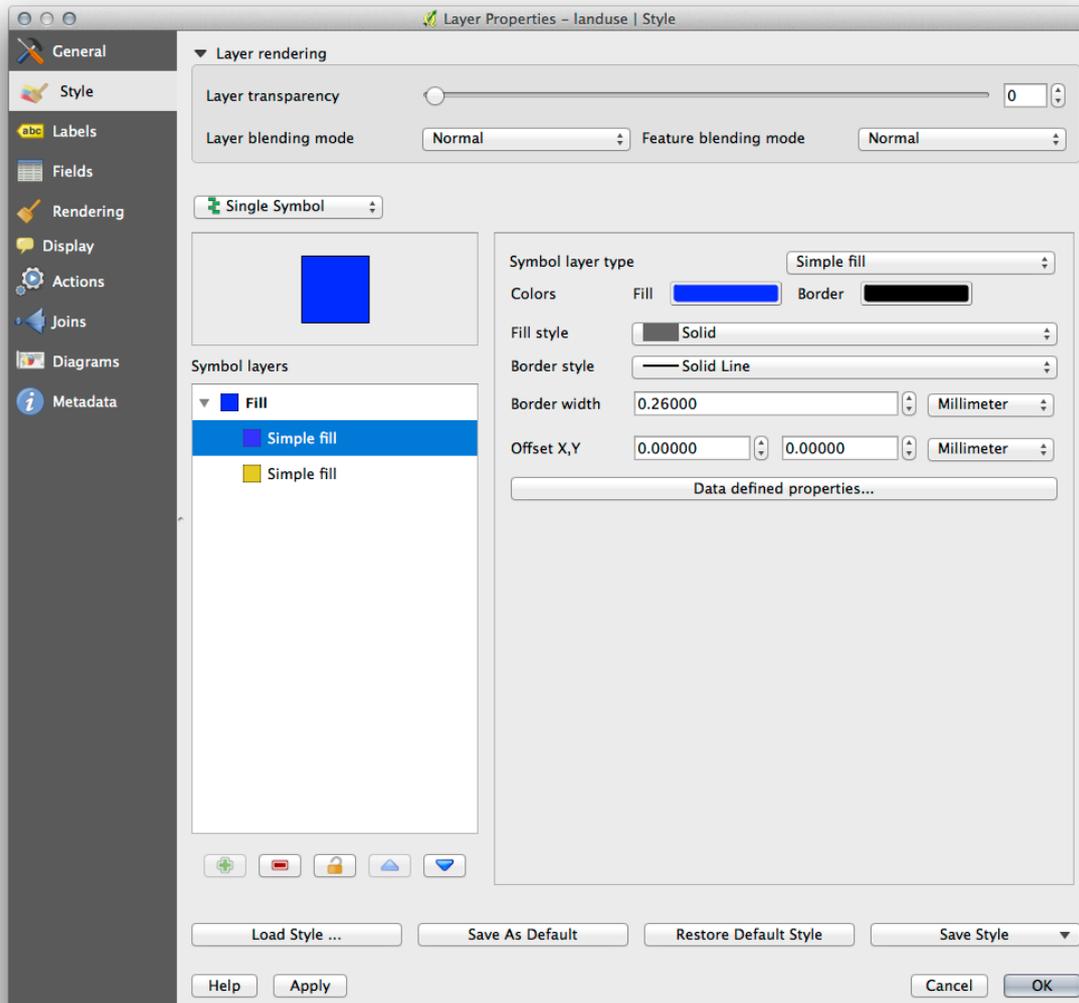
- *guilabel* : に戻る '*landuse*' 層のシンボルプロパティパネルを (クリックして : *guilabel* : '*単純 fill*' 中 : *guilabel* : '*シンボル layers*' パネル)。

この例では、現在のシンボルは、('*NO Pen*' の境界線のスタイル :: *guilabel* すなわち、それは使用しています) 何のアウトラインを持っていません。

guiLabel : 'Fill'中 : guiLabel : 'シンボル layers'パネル] を選択します。guiLabel : 'シンボル layer'の追加] ボタンをクリック :



- それをクリックし、ダイアログがこのような見え方になります:



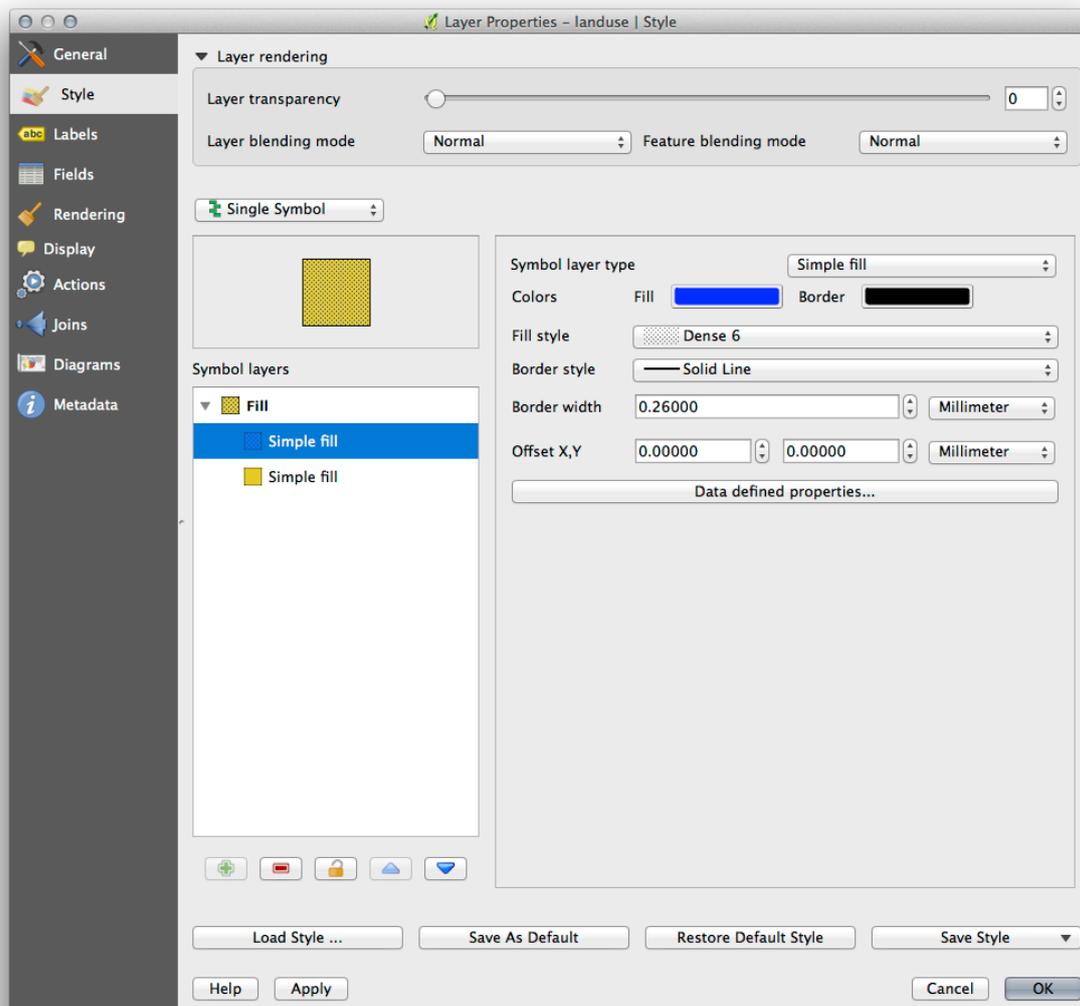
(それは、例えば、色が多少異なって見えるかもしれませんが、とにかくそれを変更するつもりです。)

今、第2のシンボル層があります。ソリッドカラーなので、それは当然のことながら、完全にシンボルの前のようなものを非表示になります。guilabel: 'ソリッド Line'の境界線スタイル、我々はしたくないプラス、それは持っています。明らかに、この記号は、変更する必要があります。

ノート: これは、マップレイヤとシンボル層との間に混乱しないことが重要です。マップ層がマップにロードされているベクター(またはラスター)です。シンボルレイヤは、マップ層を表すために使用されるシンボルの一部です。このコースは、通常は層が、常に混乱を防ぐために、シンボルレイヤと呼ばれるシンボルレイヤとしてマップレイヤを参照します。

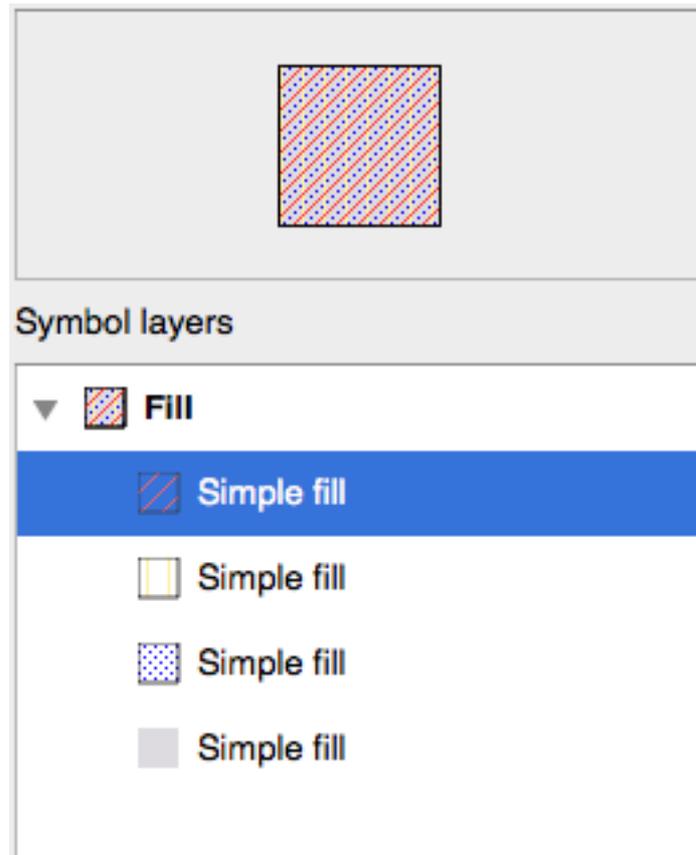
guilabel: 'シンプル Fill'層選択: 新しいと

- 以前のように、NO Pen: guilabel: への境界線スタイルを設定します。
- guilabel: 以外に塗りつぶしスタイルを変更する Solid'か: guilabel: NO brush'。例えば:



- Click *OK*. Now you can see your results and tweak them as needed.

複数シンボルレイヤを追加し、そのようにレイヤのテクスチャの種類を作成することができます。



それは楽しみです！しかし、それはおそらく、実際のマップに使用するためにあまりにも多くの色を持っていて...

3.2.7 Try Yourself

- シンプルに、必要に応じて、ズームイン作成するために覚えて、しかしためのテクスチャをそらさない : `guiLabel` : ‘上記の方法を使用して buildings’層が。

結果の確認

3.2.8 Follow Along: シンボルレベルの順序

シンボルレイヤがレンダリングされる時、それらは、異なるマップレイヤがレンダリングされる方法と同様のシーケンス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。

- `roads` レイヤに余分なシンボルレイヤ与えます (上記実証シンボルレイヤを追加するためのメソッドを使用して)。
- ベースラインに 0.3 のペン幅、白色を与え、ペン `Style` ‘ドロップダウンから `guiLabel`: ‘Line 破線を選択します。
- 新しい、最上レイヤに厚 1.3 を与え、それがソリッド `Line` であることを確認してください。

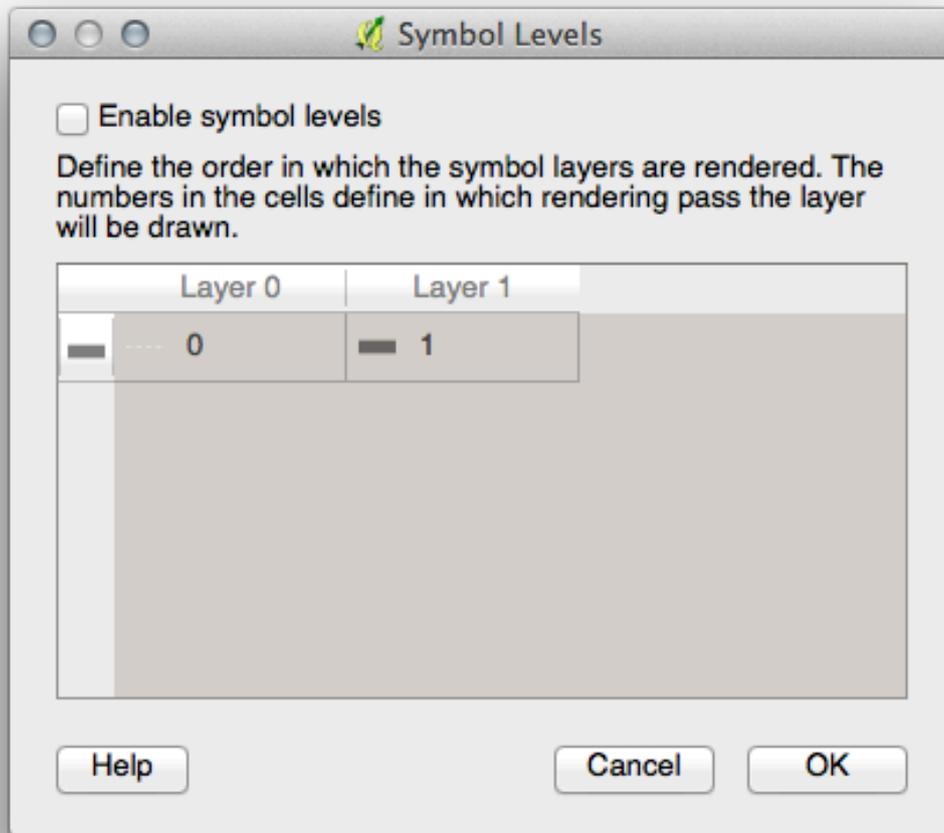
あなたはこれが発生したことがわかります。



まあそれは、我々が望むすべてのものではありません！

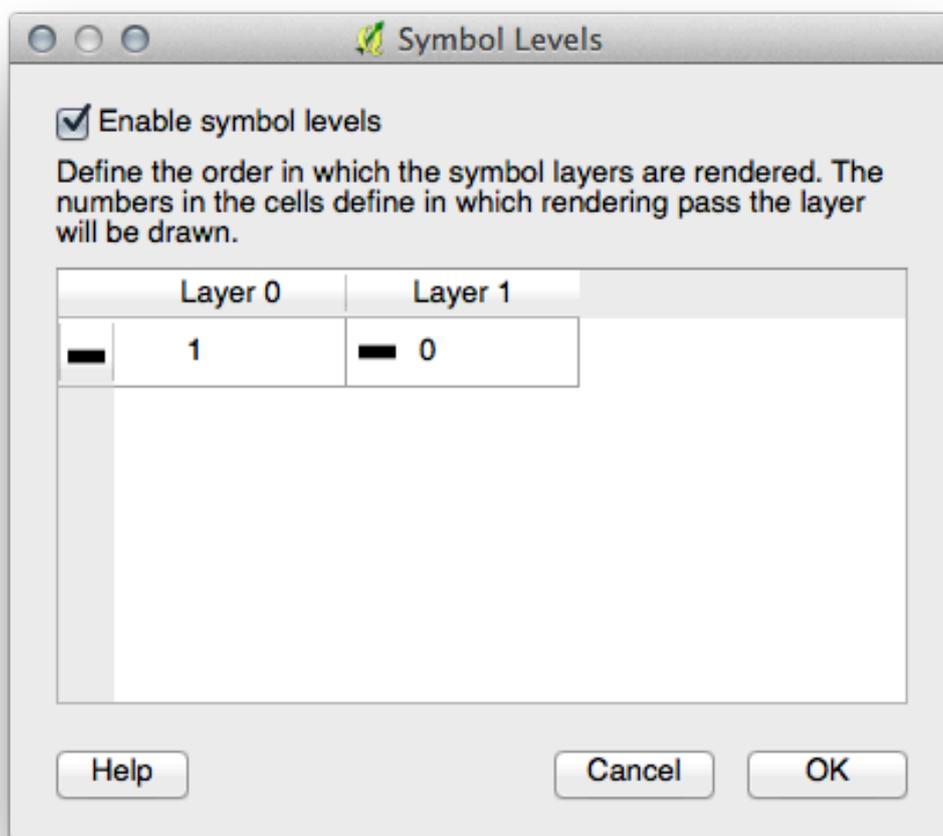
これを防ぐには、シンボルレベルを並べ替えることにより、異なるシンボルレイヤがレンダリングされる順序を制御することができます。

guiabel : 'で Line'層 : guiabel : シンボルレイヤの順序を変更するには、選択し 'シンボル layers' パネルを、[Next] をクリックします guiabel : '詳細設定 ->シンボルレベル...' 下の右にウィンドウの隅。これは、このようなダイアログが開きます。



選択 : guilabel : ‘シンボル levels’を有効にします。その後、対応するレベルの番号を入力することにより、各シンボルの層の順序を設定することができます。0は最下層です。

我々のケースでは、このように順序を入れ替えたいと思います:



これは太い黒線の上に破線、白線をレンダリングします。

- *OK* を2回クリックし、マップに戻ります。

マップはこのように見えるでしょう:



また、1本の道路が他の上にレンダリングされないように、道路の合流点は今、「マージ」されていることにも注意してください。

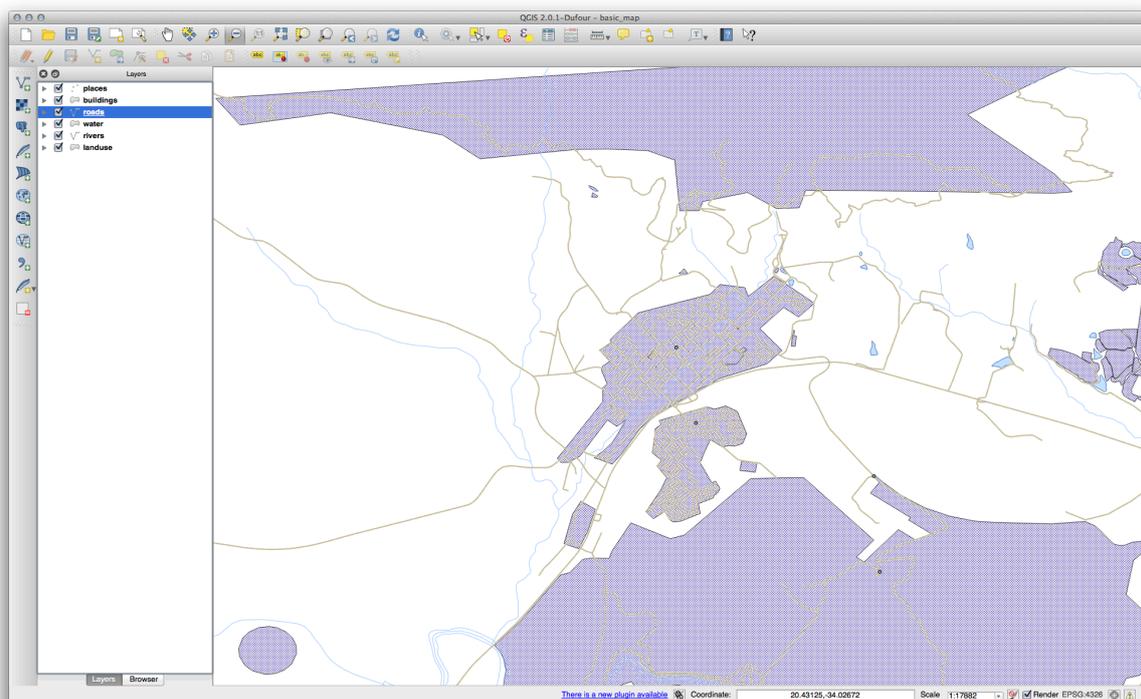
設定が完了したら、将来的に再び記号を変更する場合は、作業を失うことにならないように、記号自体を保存してください。:guilabel:‘レイヤのプロパティ’ダイアログの:guilabel:‘スタイル’タブの下の:guilabel:‘スタイルを保存...’ボタンをクリックすることで、現在のシンボルのスタイルを保存できます。一般的に、:guilabel:‘QGISのレイヤースタイルファイル’として保存する必要があります。

Save your style under `exercise_data/styles`. You can load a previously saved style at any time by clicking the *Load Style ...* button. Before you change a style, keep in mind that any unsaved style you are replacing will be lost.

3.2.9 Try Yourself

- もう一度 :guilabel:‘roads’レイヤーの外観を変更します。

道路は薄い、淡い黄色のアウトラインで、狭い半ば灰色でなければなりません。menuselection: ->シンボルレベル... “ダイアログアドバンスドあなたが経由レイヤーのレンダリング順序を変更する必要があることを覚えておいてください。

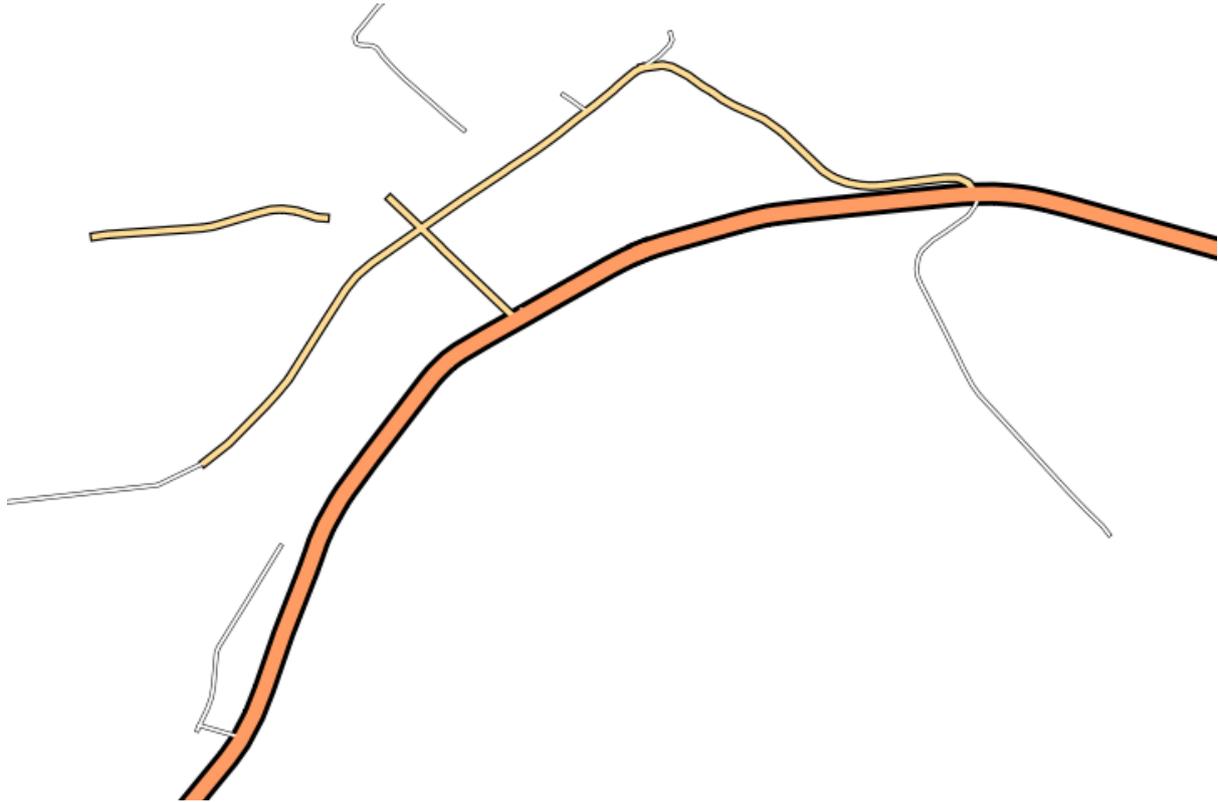


結果の確認

3.2.10 Try Yourself

シンボルレベルも分類された層（すなわち、層が複数のシンボルを持つ）のために働きます。我々はまだ分類をカバーしていないので、あなたには、いくつかの基本的な事前に分類されたデータで動作します。

- 新しいマップを作成し、唯一の追加：gui-label：‘roads’データセット。
- KBD :: *exercise_data / styles* KBD :: で提供 ‘advanced_levels_demo.qml’スタイルを適用します。
- Swellendam のエリアにズームします。
- シンボルレイヤを使用して、層の輪郭は、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。



結果の確認

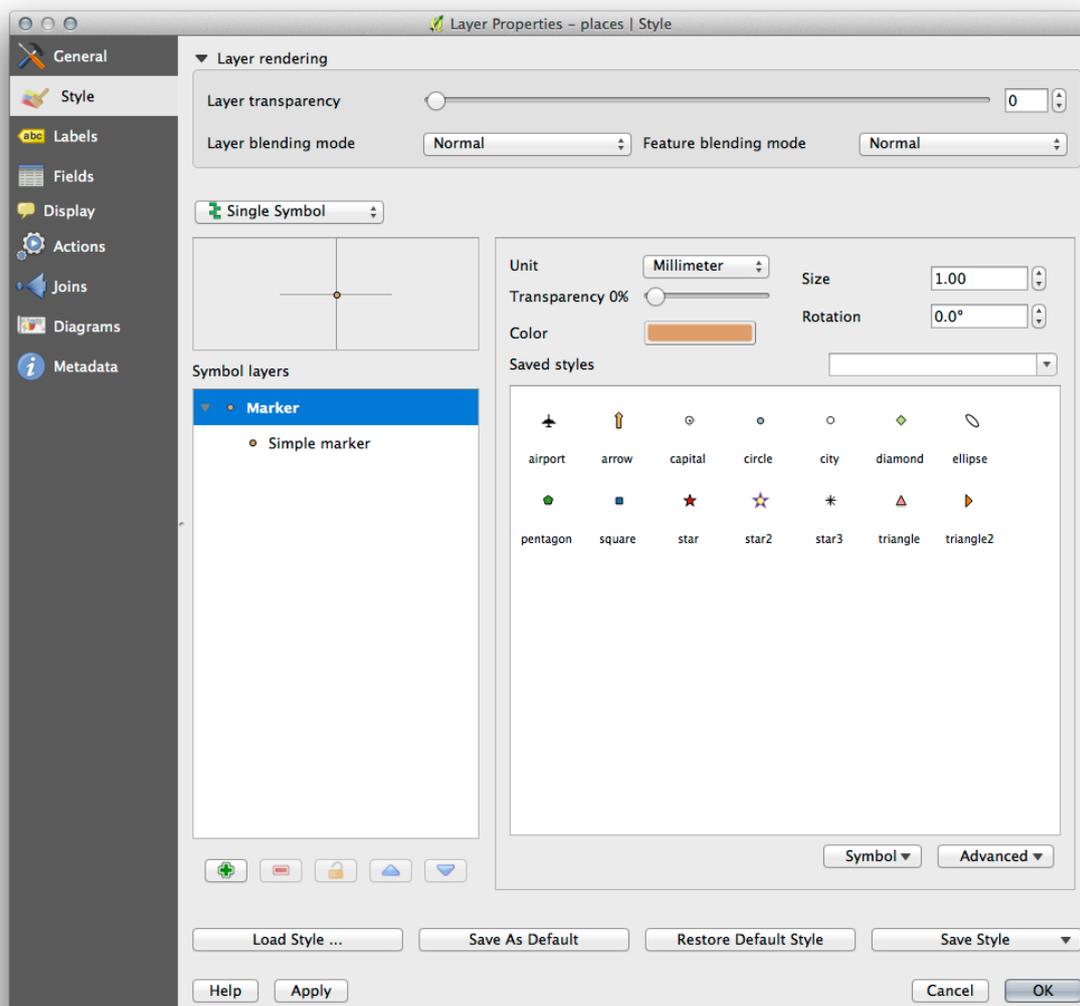
3.2.11 Follow Along: シンボルレイヤタイプ

色を塗りつぶし設定し、事前に定義されたパターンを使用することに加えて、完全に異なるシンボルレイヤタイプを使用できます。今まで使用してきた唯一のタイプは*シンプル塗りつぶし*型でした。より高度なシンボルレイヤの種類は、さらにシンボルをカスタマイズできます。

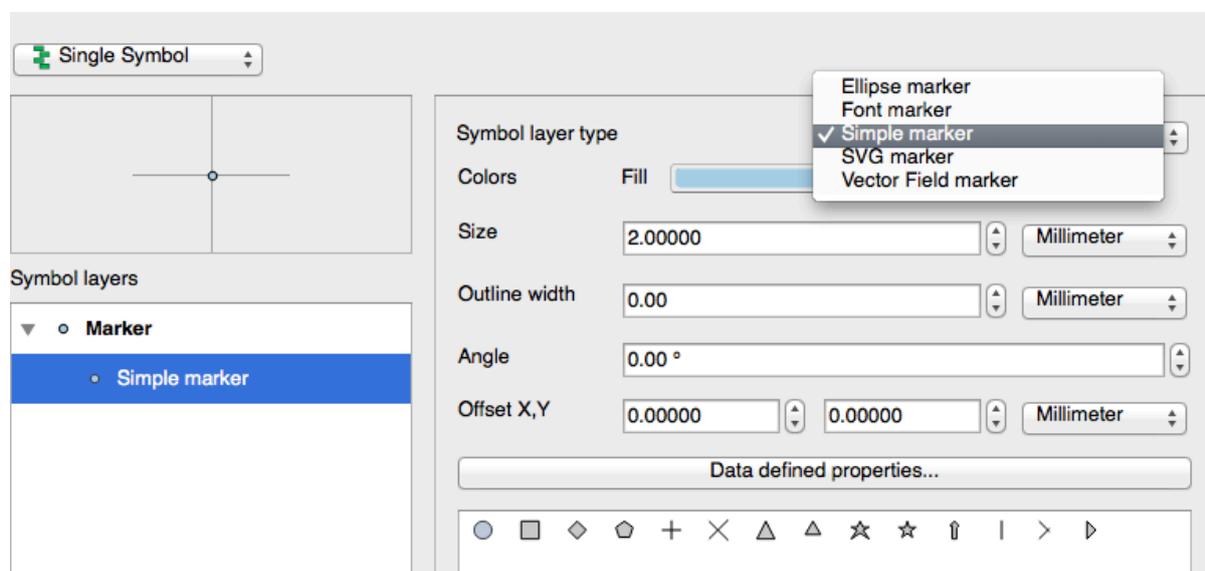
ベクターの各種類（点、線、多角形）は、シンボルレイヤタイプの独自のセットを有しています。まず、ポイントのために利用可能なタイプを見ていきます。

ポイントシンボルレイヤタイプ

- ベーシックマップ プロジェクトを開きます。
- guilabel : ‘places’層のためのシンボルプロパティを変更します。



- guilabel: ‘でシンプル marker’層: guilabel: あなたが選択することで、様々なシンボルレイヤタイプにアクセスすることができ guilabel: ‘シンボルレイヤ type’がドロップダウン ‘シンボル layers’パネルを、[Next] をクリックします

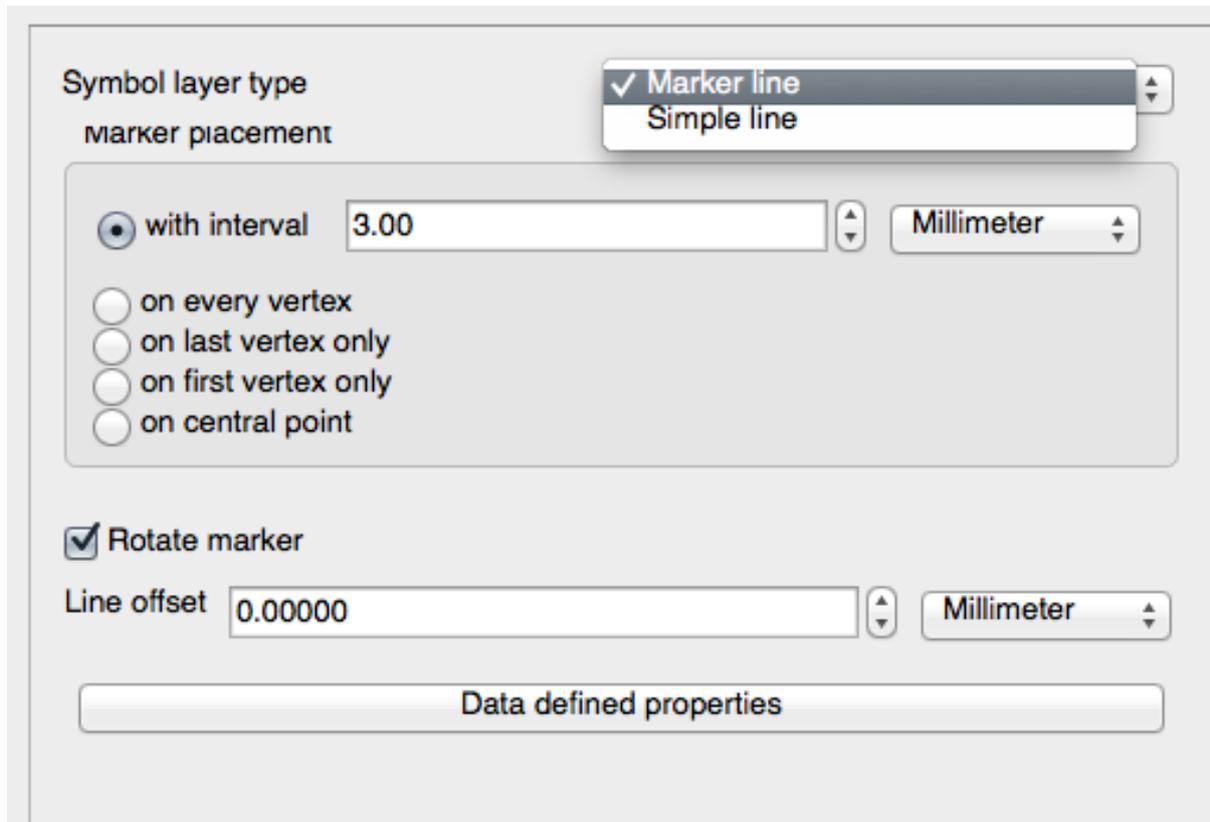


- 利用できるさまざまなオプションを調査し、あなたが適切だと思うスタイルでシンボルを選択してください。
- guilabel: と白の境界線と淡い緑色の塗りつぶしを使用した単純な marker、: guilabel: 'size'の: KBD: 3,00'と: の width'概要: guilabel 疑いで、ラウンドを使用している場合: の kbd: '0.5。

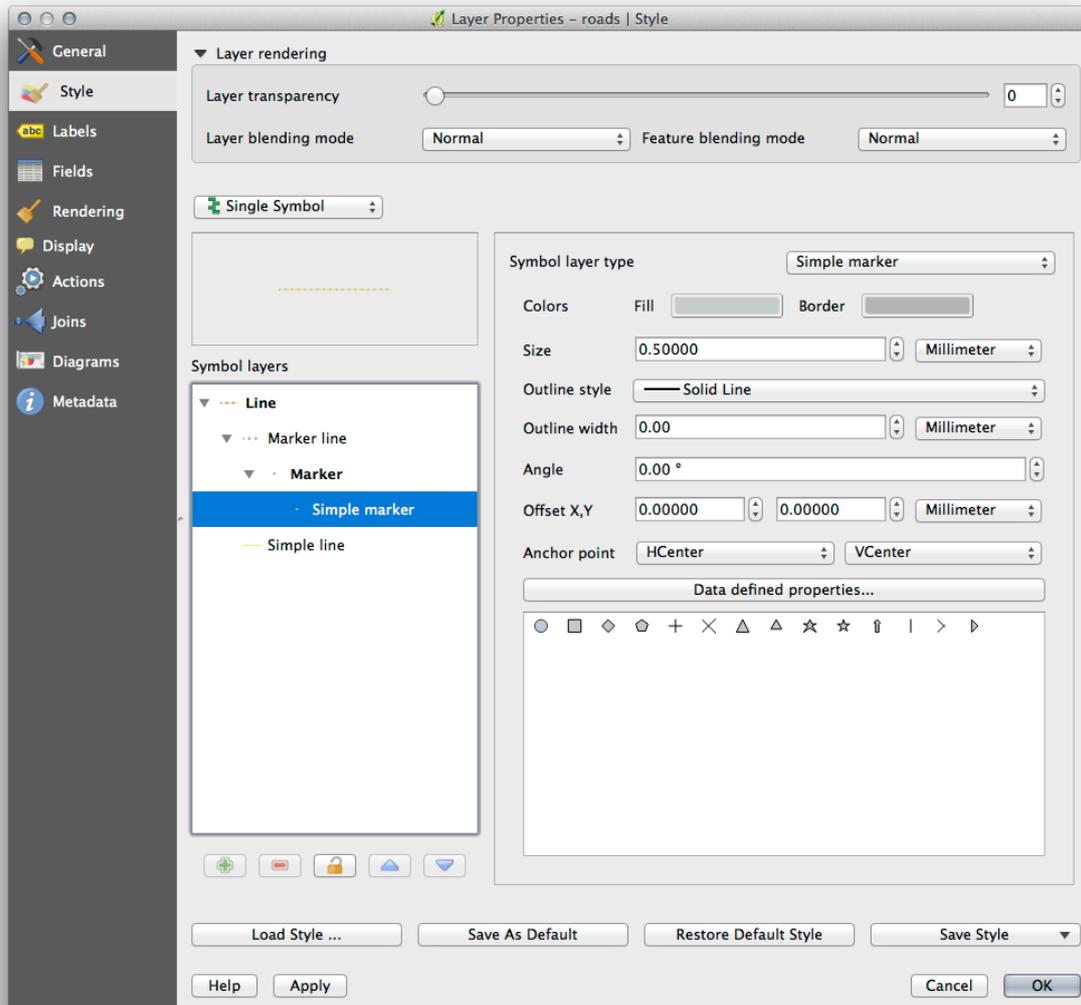
ラインシンボルレイヤタイプ

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:

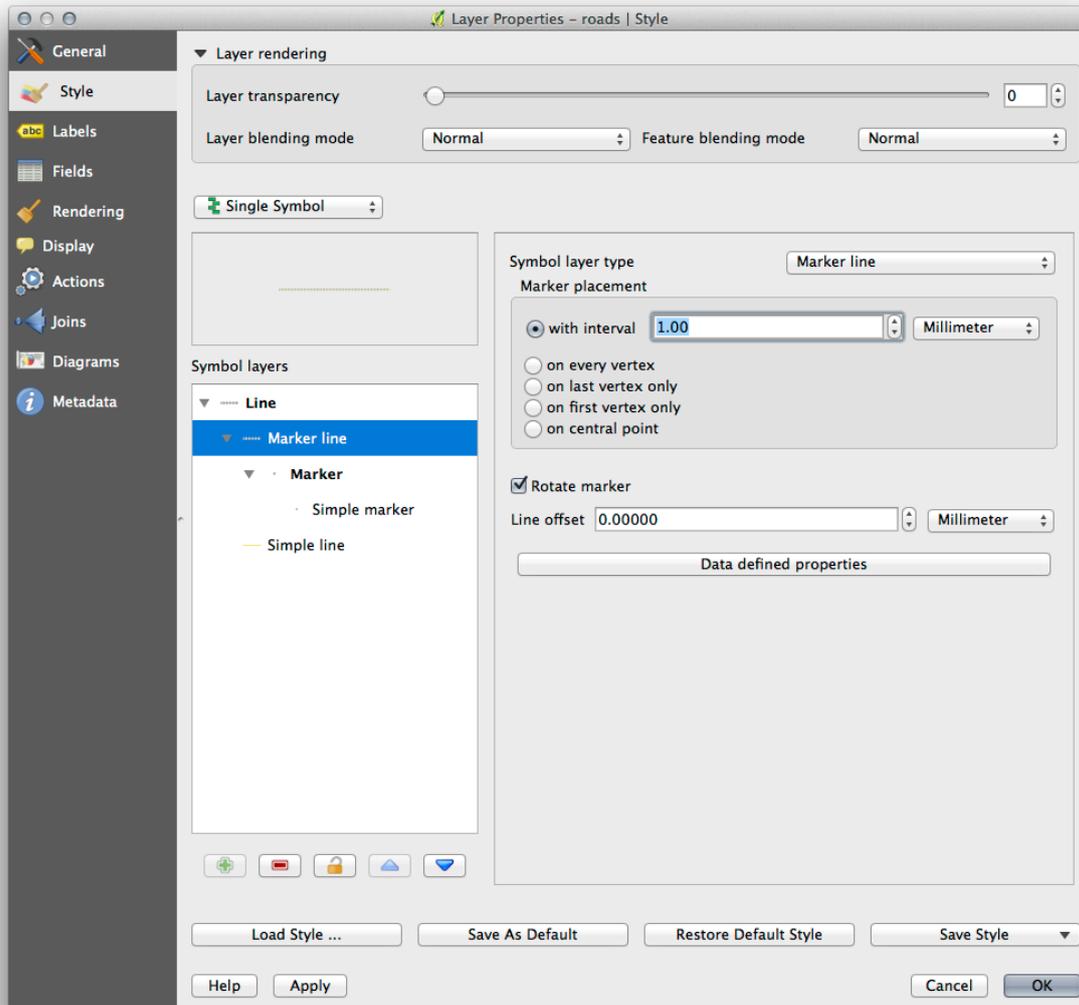
- guilabel: 'へ roads'層の一番上のシンボルレイヤ: guilabel: 'マーカー line'のシンボル層の種類を変更します。



- guilabel: 'でシンプル marker'層: guilabel: 'シンボル layers'パネル] を選択します。このダイアログを一致させるために、シンボルのプロパティを変更します。



- 間隔を 1,00 に変更します:



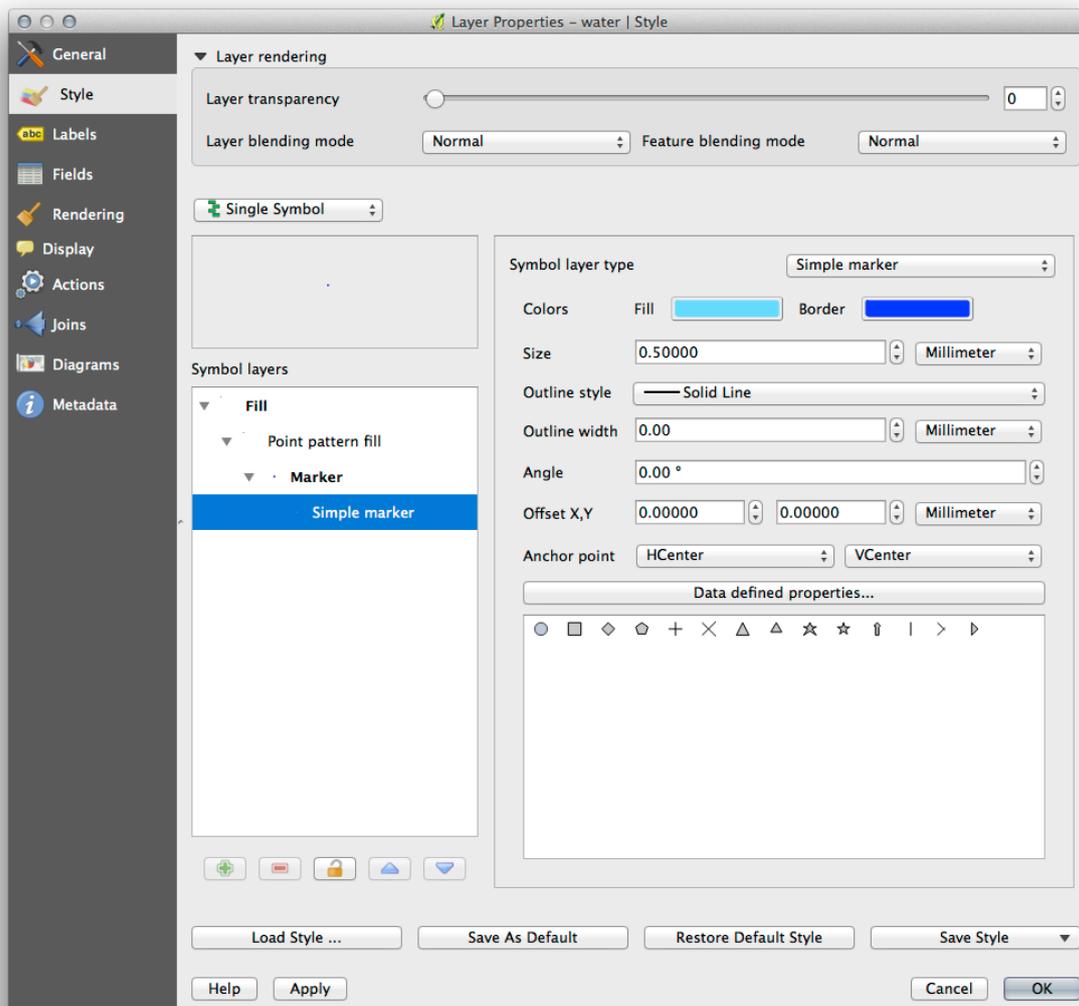
- スタイルを適用する前に - シンボルレベルは (我々が以前に使用>シンボル levels‘ダイアログ‘高度 :: guilabel 経由で) 正しいことを確認してください。

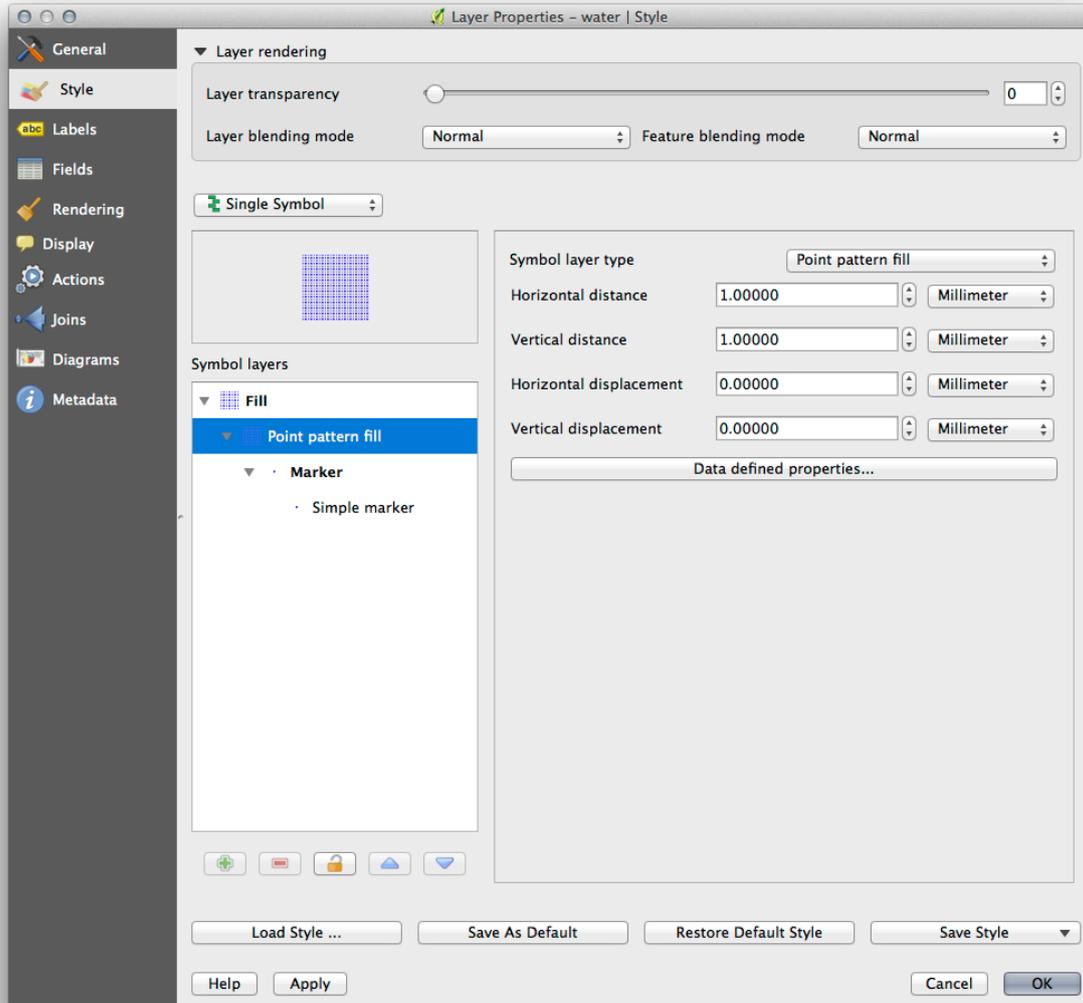
スタイルを適用したら、地図上でその結果を見てみましょう。見ることができるように、これらのシンボルは、道路と一緒に向きを変えるが、常にそれに沿って曲げないでください。これは、いくつかの目的のためではなく、他人のために有用です。ご希望の場合は、戻って、それは前にした方法に問題のシンボルレイヤを変更できます。

ポリゴンシンボルレイヤタイプ

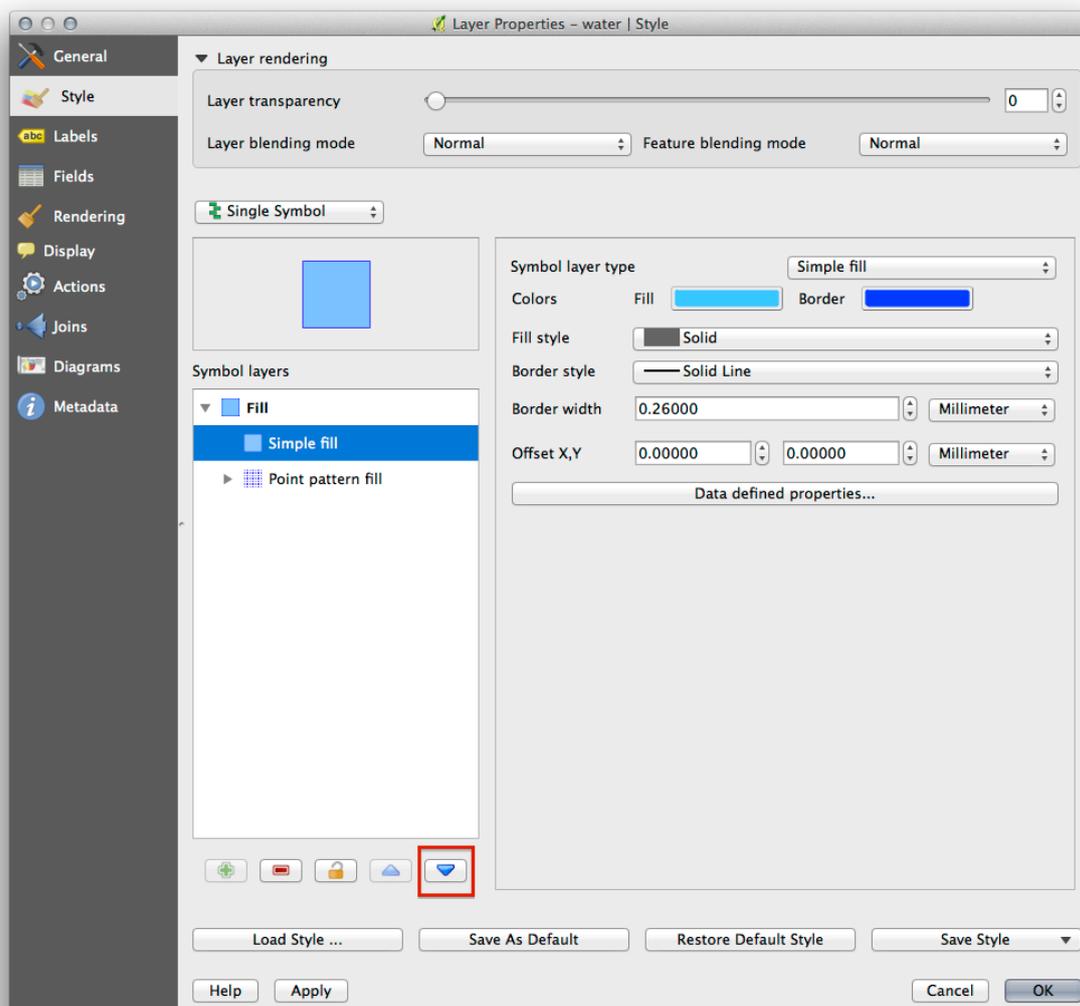
ポリゴンデータのために利用できるさまざまなオプションを表示するには：

- 他の層のための前と同じように、‘water‘層： guilabel： のシンボル層の種類を変更します。
- 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。
- これらのうち、適した1つを選択します。
- 次のオプションを使用して、ポイントパターン *fill*： guilabel： 疑問がある場合は、使用します。





- guilabel : ‘シンプル fill’通常で新しいシンボルレイヤを追加します。
- それ暗い青い枠と同じ水色ください。
- guilabel : と点パターンシンボル層の下に移動 ‘down’ ボタンを移動させます。



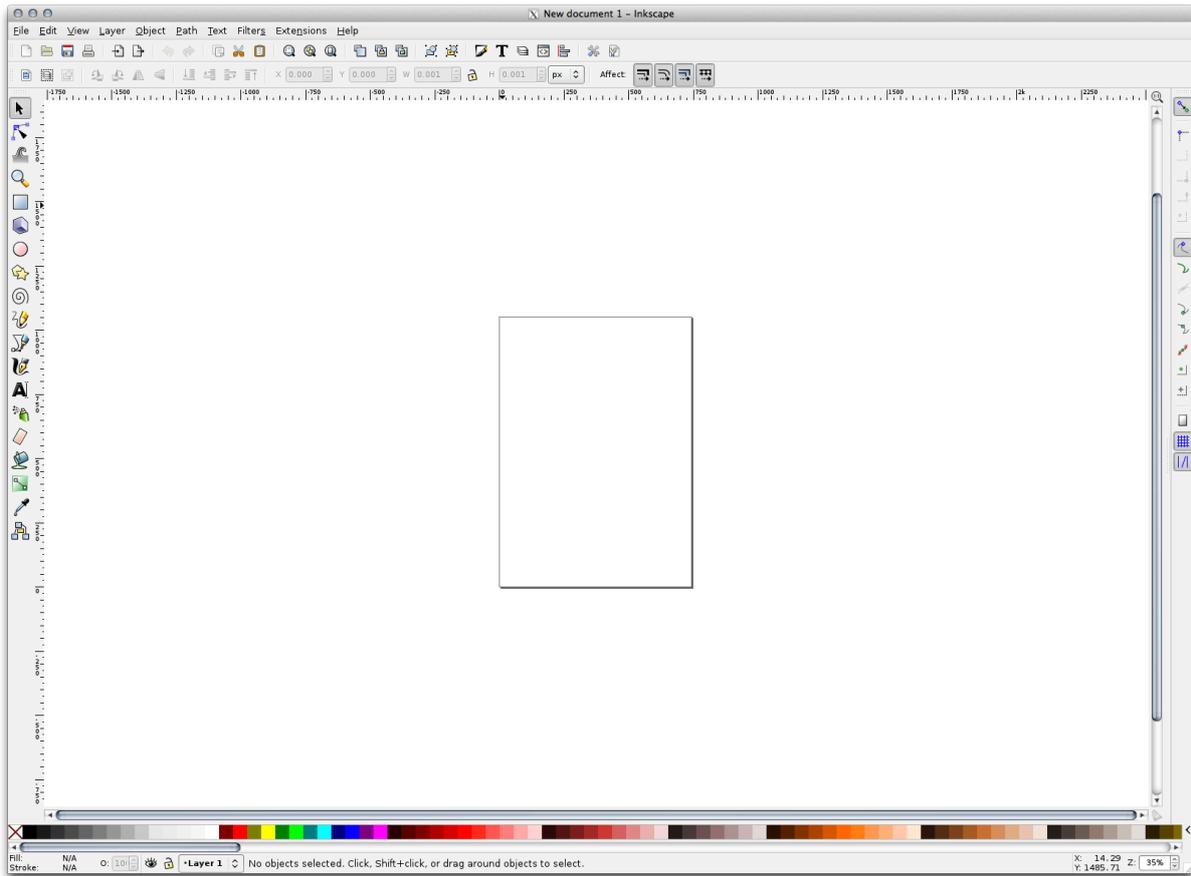
その結果、あなたはテクスチャを構成する個々のドットの大きさ、形状や距離を変えることができるという利点があり、水の層のためのテクスチャシンボルを持っています。

3.2.12 Follow Along: カスタム SVG 塗りつぶしの作成

ノート: この演習を行うには、無料のベクター編集ソフトウェア *Inkscape* の <https://inkscape.org/> _インストールされている必要があります。

- Inkscape プログラムを起動します。

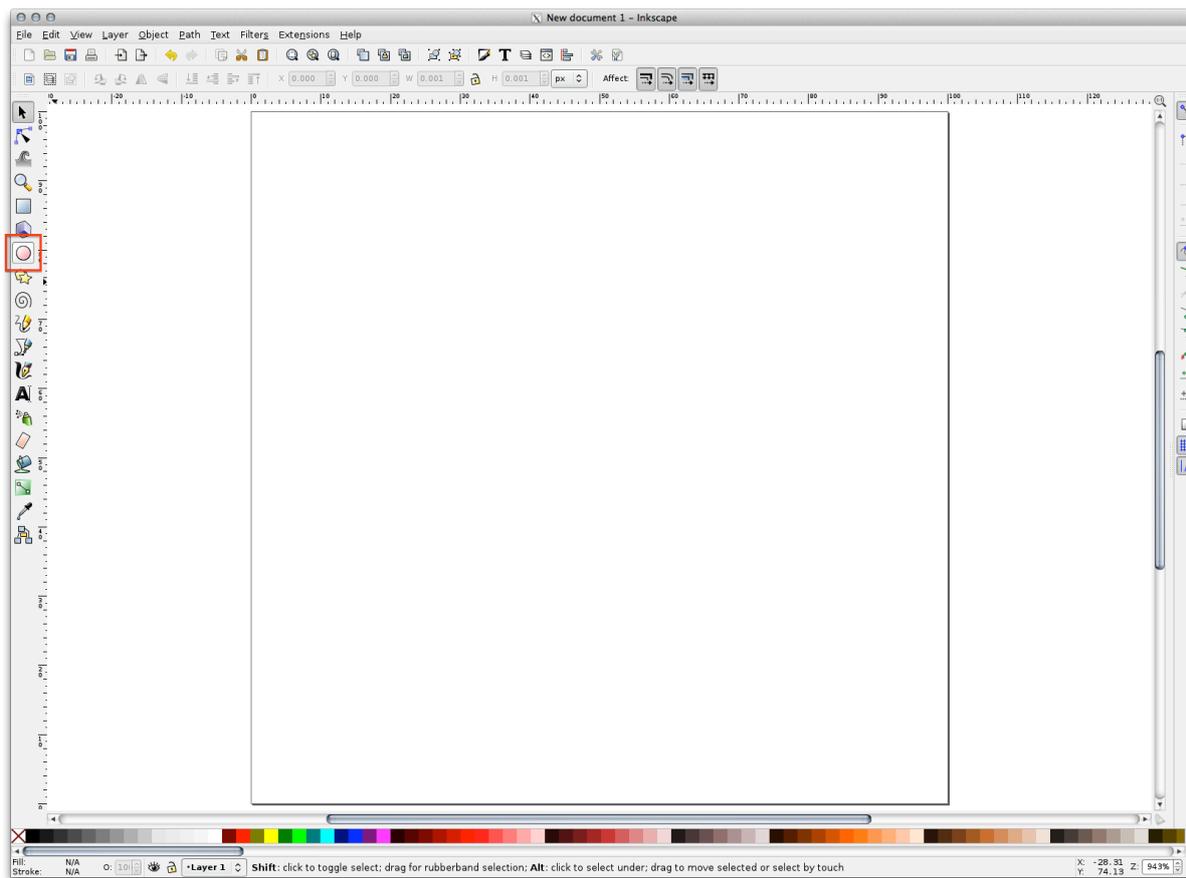
以下のインターフェースを確認できるでしょう。



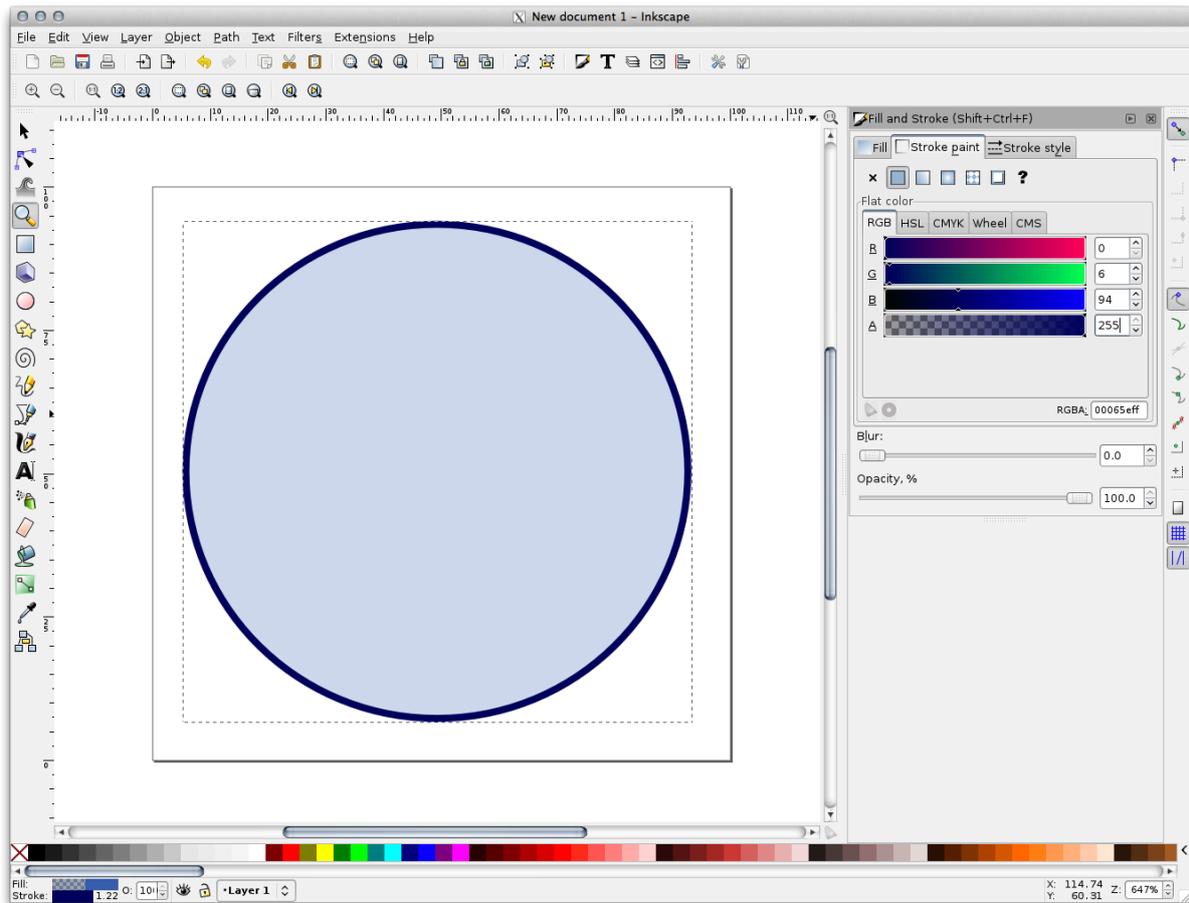
コーレルのように、他のベクター画像編集プログラムを使用している場合は、このおなじみを見つける必要があります。

まず、小さなテキストチャに適したサイズにキャンバスを変更します。

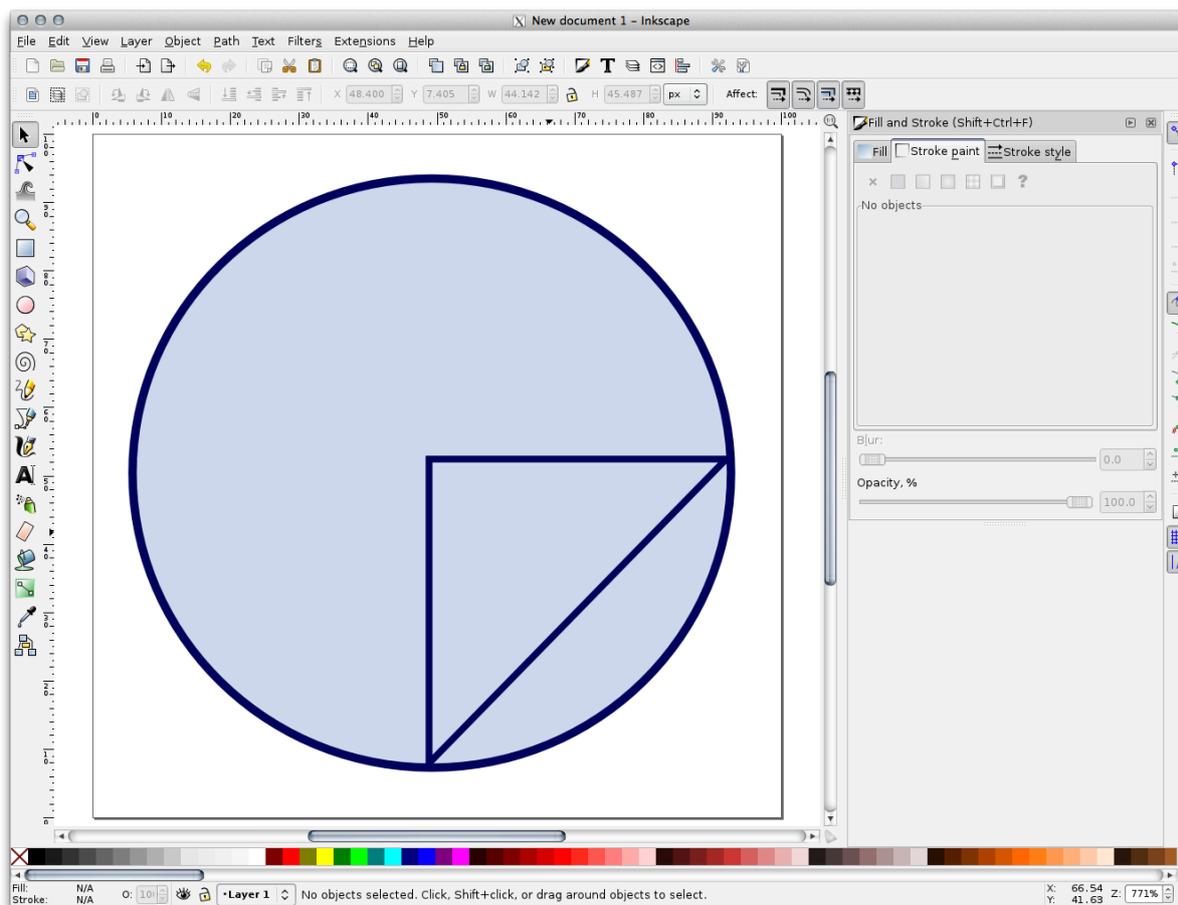
- メニュー項目をクリックして：menuselection：ファイル ->ドキュメント *Properties*。guilabel：‘ドキュメント *Properties*’ダイアログこれはあなたを与えるだろう。
- Change the *Units* to *px*.
- Change the *Width* and *Height* to 100.
- 実行後はダイアログを閉じます。
- 作業しているページを参照してください> Page‘ ->ズーム - ‘ビュー：menuselection：メニュー項目をクリックしてください。
- 円 ツールを選択します。



- 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。あなたはそれを描いている間 ‘Ctrl’ ボタン : KBD : 楕円が円に変わるようにするには、ホールド。
- 先ほど作成した円を右クリックし、その : guilabel : ‘塗りと Stroke’ 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 :
 - 、何とか薄いグレー青 ‘Fill’ 色 : guilabel : 変化
 - guilabel : 国境に暗い色の割り当て ストローク *paint*] タブを、
 - guilabel : ‘ストローク style’ タブと下の境界線の太さを減らします。



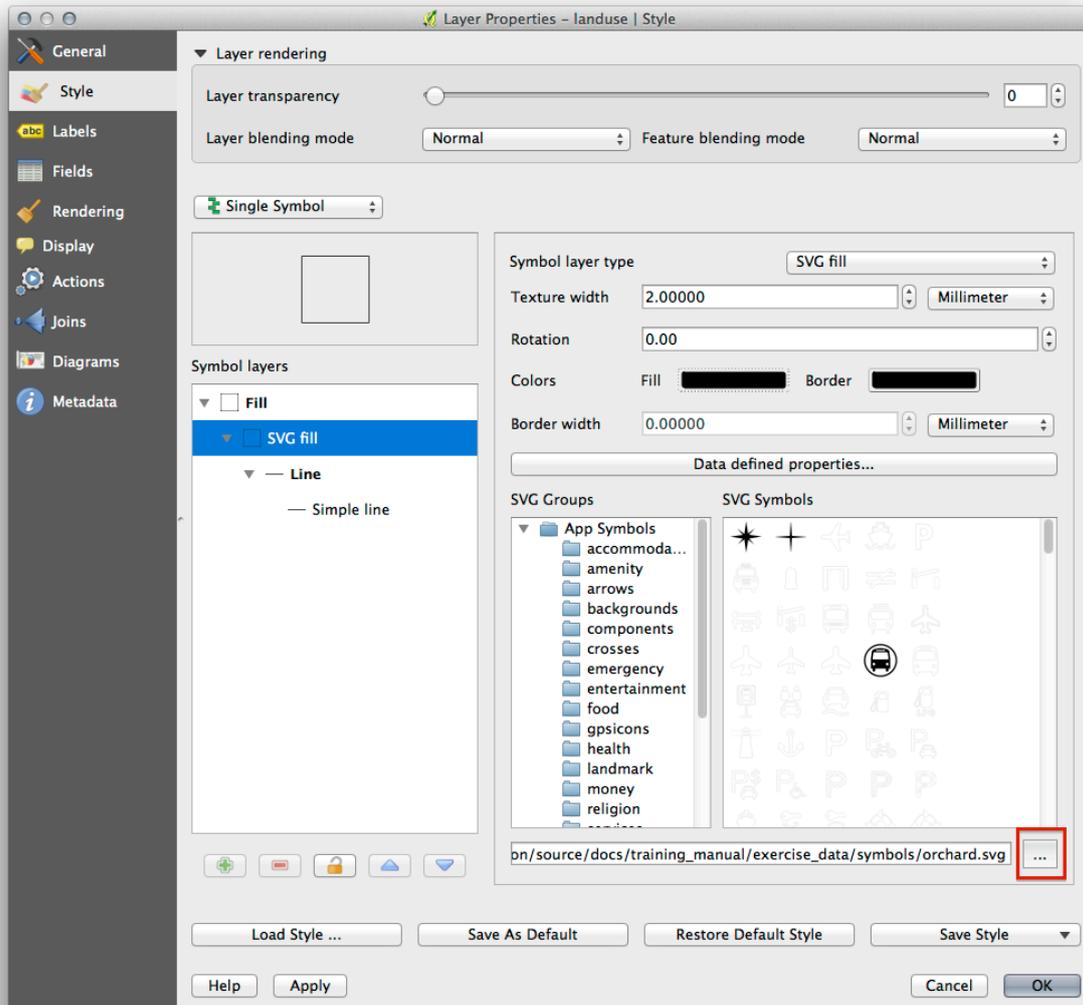
- guilabel : ‘Pencil’ツール使って線を引きます :
 - 行を開始するには、一度クリックしてください。ホールド : KBD : ‘ctrl’は、それが 15 度刻みにスナップ作ります。
 - 水平方向にポインタを移動し、クリックするだけでポイントを置きます。
 - クリックして、ラインの頂点にスナップし、クリックするだけで終わる垂直線を、トレースします。
 - 今、二つの端の頂点に参加。
 - このようなシンボルで終わるように、円のストロークに合わせ、必要に応じて、それを周りに移動するには三角形のシンボルの色と幅を変更します。



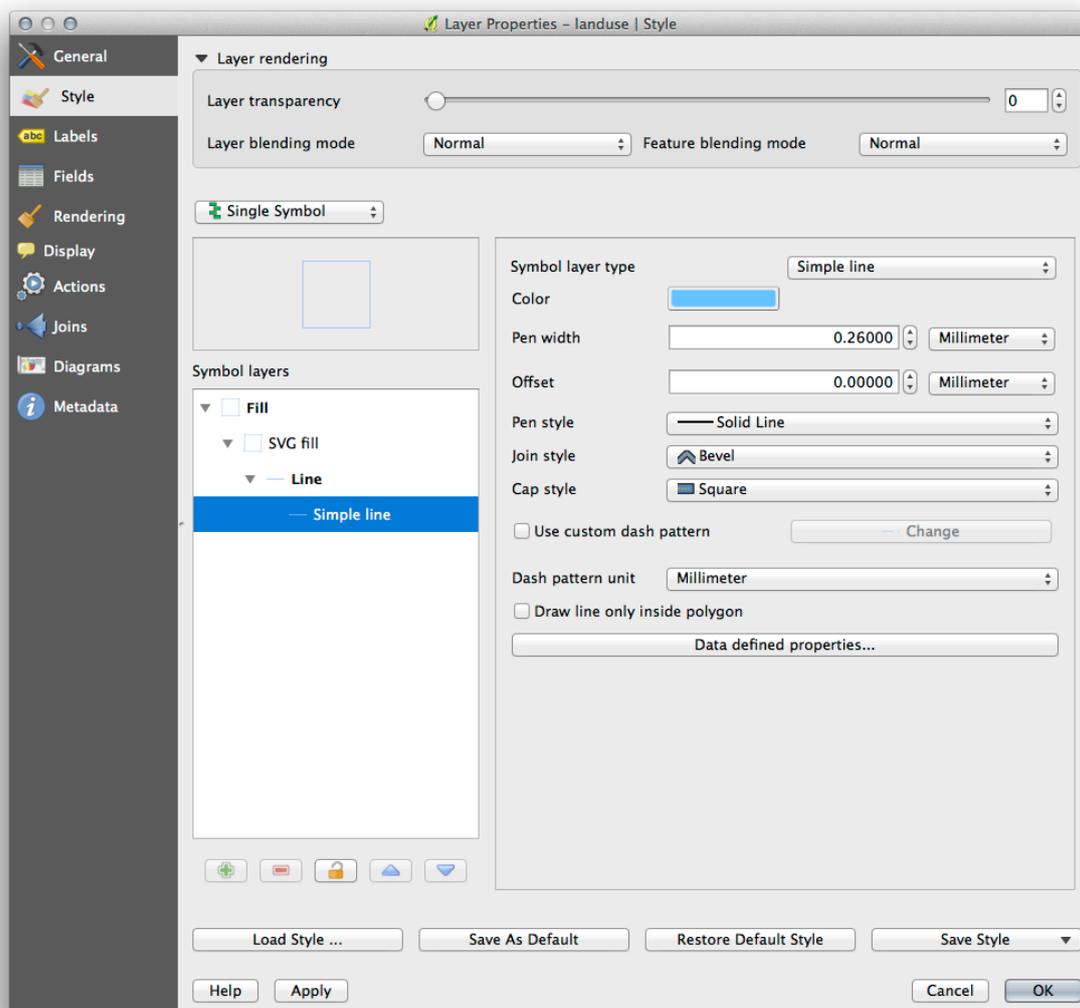
- guilabel : コースは下であるディレクトリの下 `landuse_symbol` : 記号は、あなたが満たしを取得する場合は、その後、として保存の kbd : ‘ exercise_data / symbols ‘、SVG ファイルとして。

QGIS では:

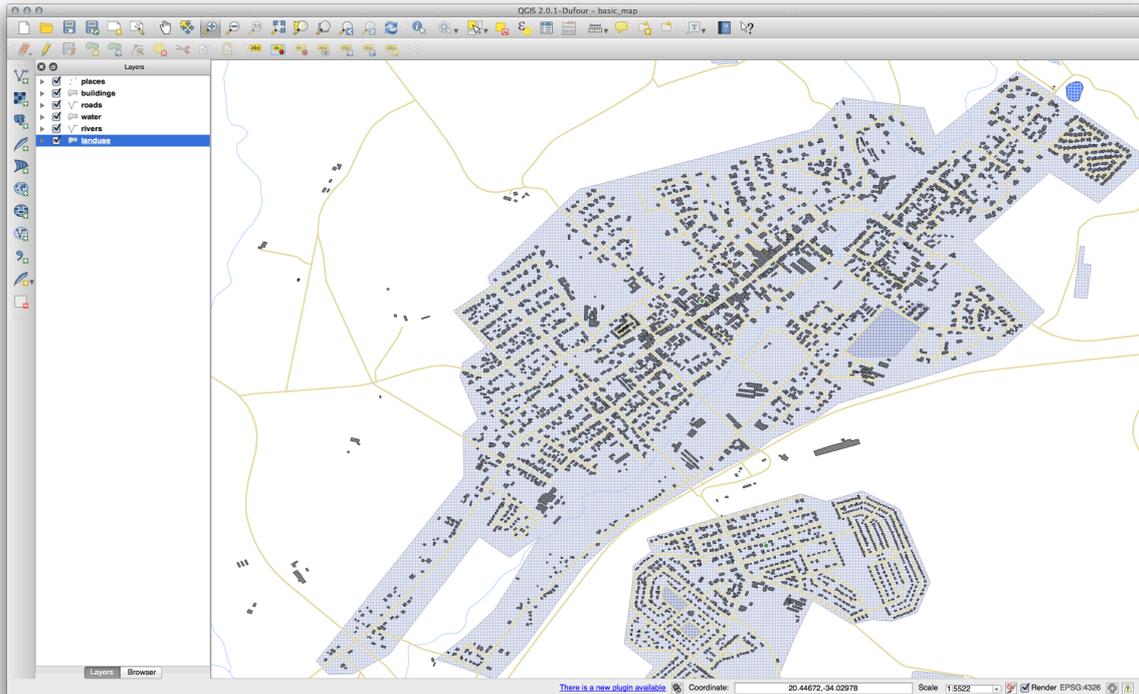
- guilabel : レイヤー *Properties* ‘用 : ファイル : landuse ‘層開きます。
- 以下に示すように、シンボルレイヤ *type* ‘がオプション : `guilabel ::` で *guilabel* : *Style* ‘タブ、として “ SVG Fill “ を選択することにより、シンボル構造を変更します。
- browseButton **|クリック|** : guilabel : ‘Browse ‘ボタンは、あなたの SVG 画像を選択します。これは、シンボルツリーに追加されますと、あなたは現在、異なる特性 (色、角度、効果、単位を...) をカスタマイズすることができます。



また、(下記参照) SVG 層の境界を更新したいことがあります。



ダイアログを検証したら、の機能：ファイル：‘landuse’層は今、次のマップ上のような質感を示し、記号のセットによってカバーされるべきです。guilabel：‘テクスチャwidth’テクスチャが表示されていない場合は、マップキャンバスにズームインまたはより大きなレイヤのプロパティで設定する必要があるかもしれません。



3.2.13 In Conclusion

異なる層のためのシンボルを変更すると、読みやすいマップにベクトルファイルのコレクションを変えてきました。だけでなく、あなたは何が起きているか見ることができる、それが見ても、うれしいです！

3.2.14 Further Reading

[Examples of Beautiful Maps](#)

3.2.15 What's Next?

全体のレイヤーのためのシンボルを変更すると便利ですが、それぞれのレイヤーの中に含まれている情報は、まだこれらの地図を読んで誰かに利用できません。呼ばれる街は何ですか？特定の領域は、行政にどの地域属しているのですか？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。

ノート：最近のマップを保存するのを覚えてますか？

Chapter 4

Module: ベクタデータの分類

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、マップを使う人が、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。

4.1 Lesson: 属性データ

今まで、地図に行った変更の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、すべての土地利用エリアが似ている、そしてすべての道路が似ています。地図を見るとき、視聴者は、彼らが見ている道路については何も知りません。何らかの地域に何らかの形状の道があるとだけ。

しかし、GISの全体強みは、地図上に表示されているすべてのオブジェクトも属性を持っていることです。GISでの地図は単なる絵ではありません。彼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ます。

このレッスンの目的：オブジェクトの属性データを探索し、様々なデータがのために役立つことができるかを理解するために。

4.1.1 Follow Along: 属性データ

places レイヤーの属性テーブルを開きます（必要に応じて前のセクション「ベクトルデータを使った作業」を参照してください）。どのフィールドがラベル形式で表現するために最も有用でしょうか？それはなぜでしょうか？

結果をチェック

4.1.2 In Conclusion

今、使っているデータに実際にあるものを見るために属性テーブルを使用する方法を知っています。気になる属性を持っていればどんなデータセットでも有用でしょう。必要な属性を知っている場合は、与えられたデータセットを使用できるかどうか、また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。

4.1.3 What's Next?

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。

4.2 Lesson: ラベルツール

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。任意の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。

ノート: レイヤプロパティ ダイアログはラベル タブを持ち同じ機能を提供していますが、この例ではツールバーのボタンからアクセスできる ラベルツール を使用します。

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。

4.2.1 Follow Along: ラベルの使用

ラベルツールにアクセスする前に、それがアクティブになっていることを確認する必要があります。

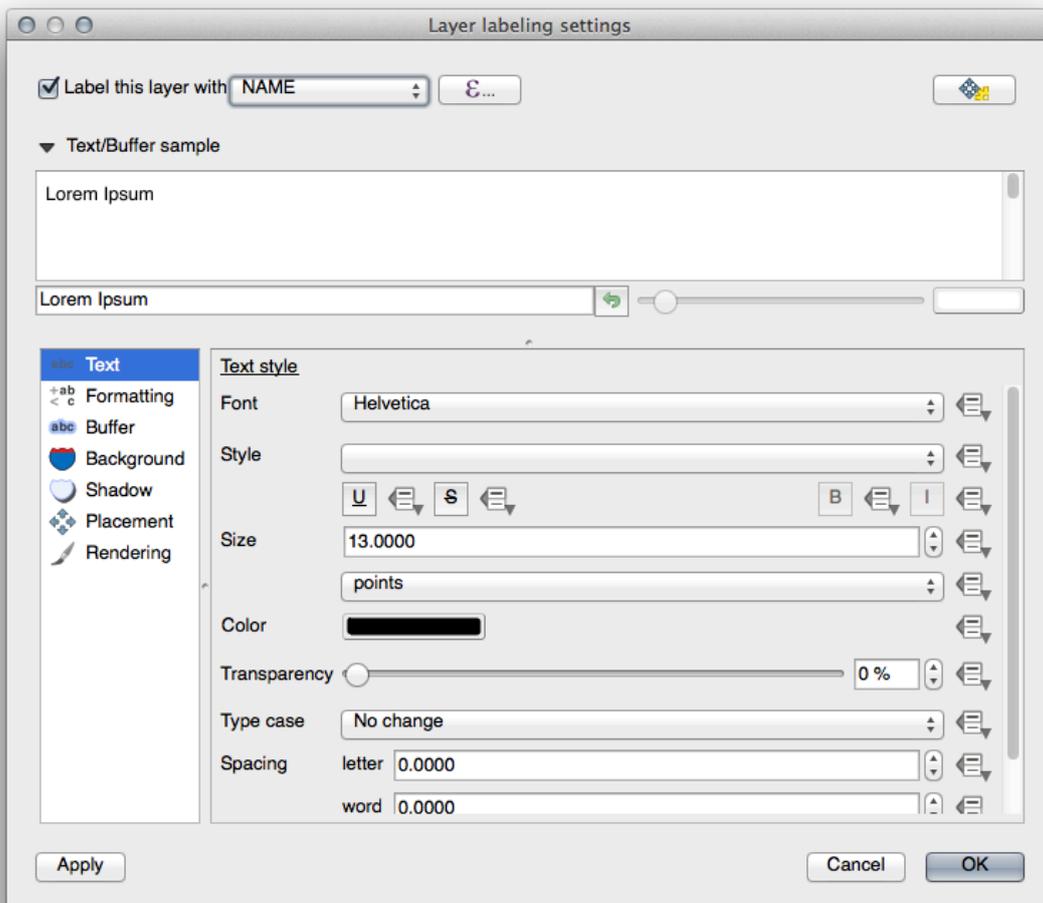
- メニュー項目 ビュー → ツールバー に移動します。
- ラベル アイテムの横にチェックマークがあることを確認して下さい。もしなければ ラベル アイテム をクリックするとアクティブになります。
- レイヤリストで *places* レイヤをクリックしてハイライト表示させます。
- 以下のツールバーボタンをクリックします: 

レイヤラベリング設定 ダイアログが表示されます。

- このレイヤのラベル の隣のボックスをチェックします。

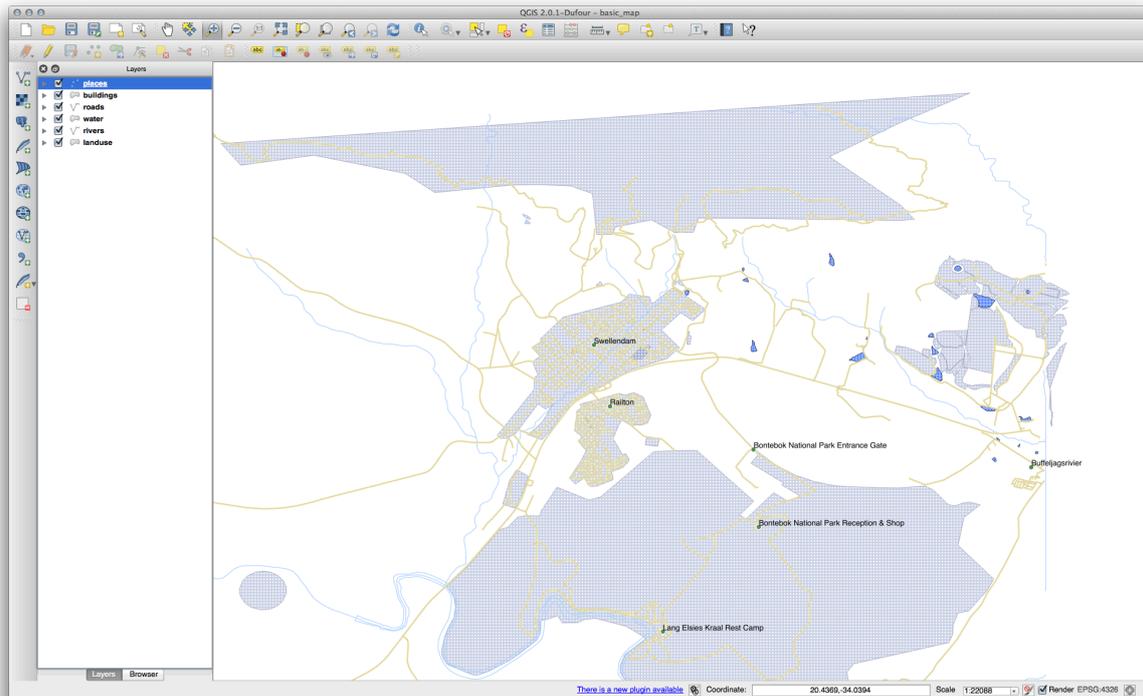
ラベルに使用する属性フィールドを選択します。前のレッスンであなたはこの目的には `NAME` フィールドが最も適していると決めました。

- リストから *name* を選択します:



- *OK* をクリックします。

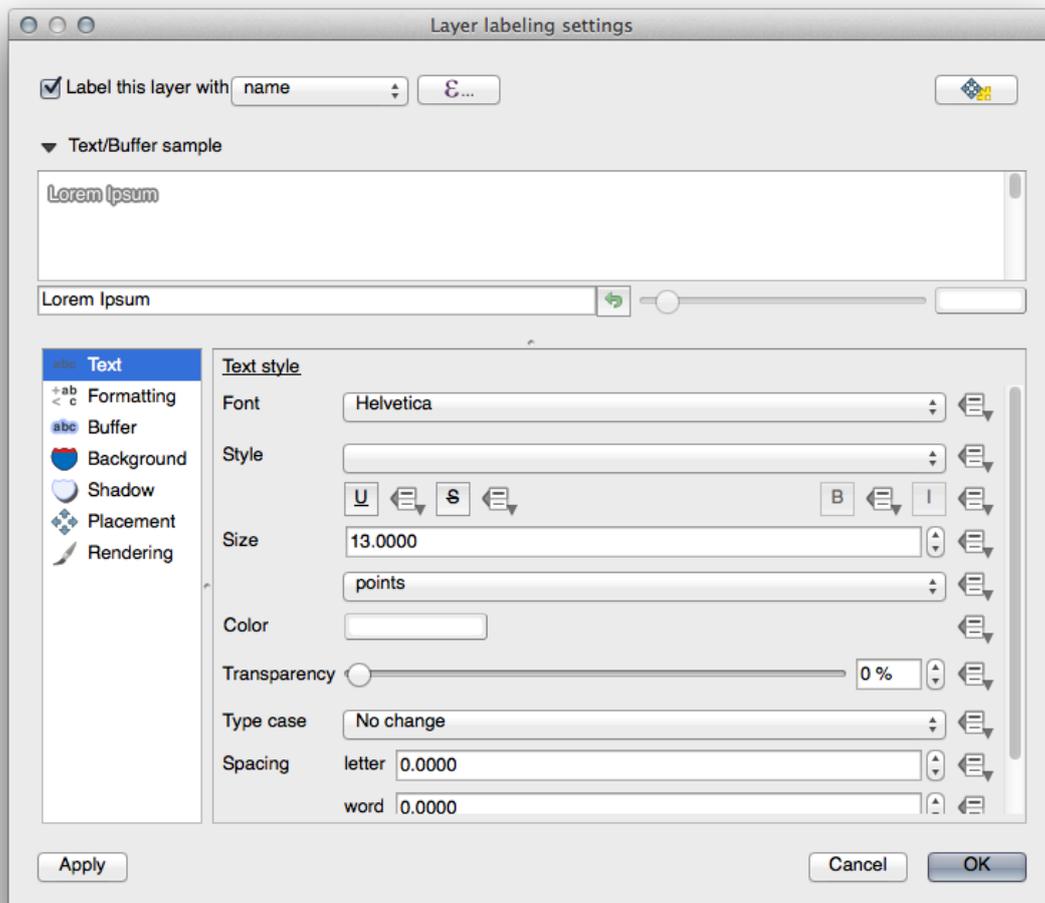
今、地図にはこのようなラベルが表示されるはずですが:



4.2.2 Follow Along: ラベルオプションの変更

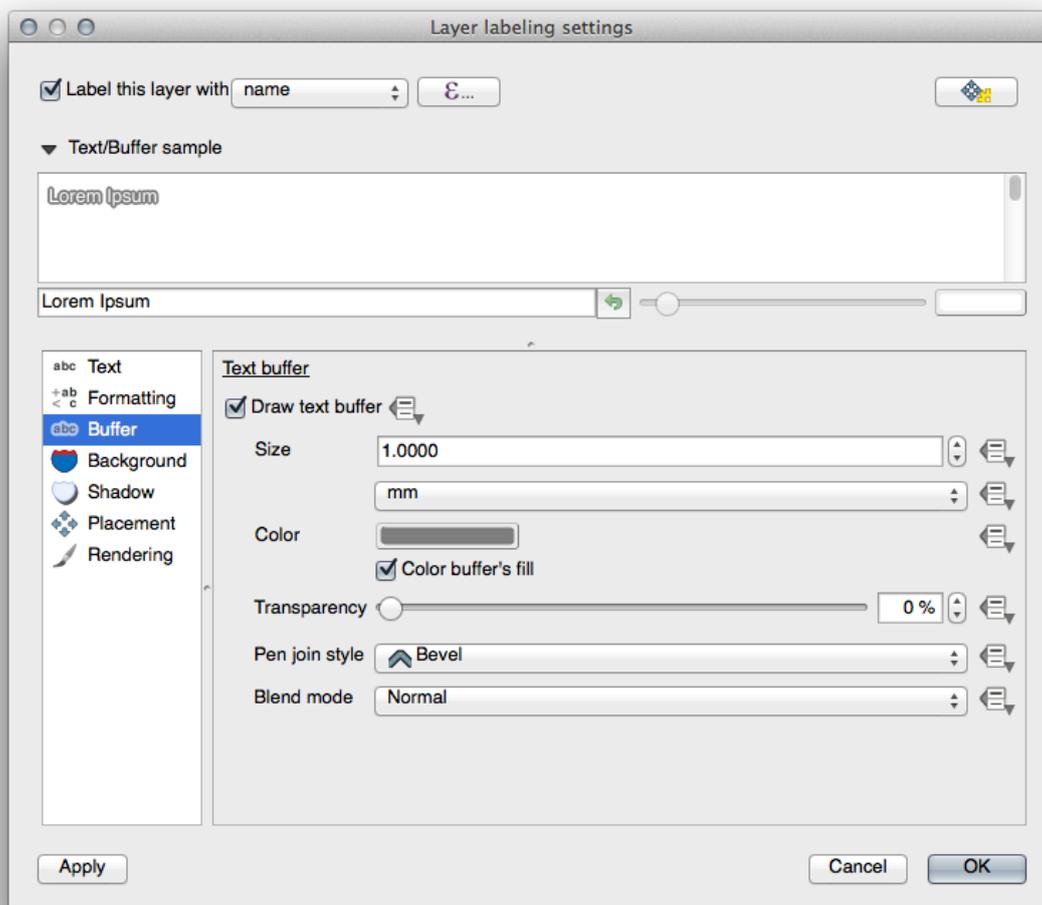
以前のレッスンであなたが選択したスタイルによってはラベルが適切にフォーマットされず、重複していたりポイントマーカーから遠く離れていたりしているかもしれません。

- 前のようにツールバーのボタンを押してラベルツールを再度開きます。
- 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:



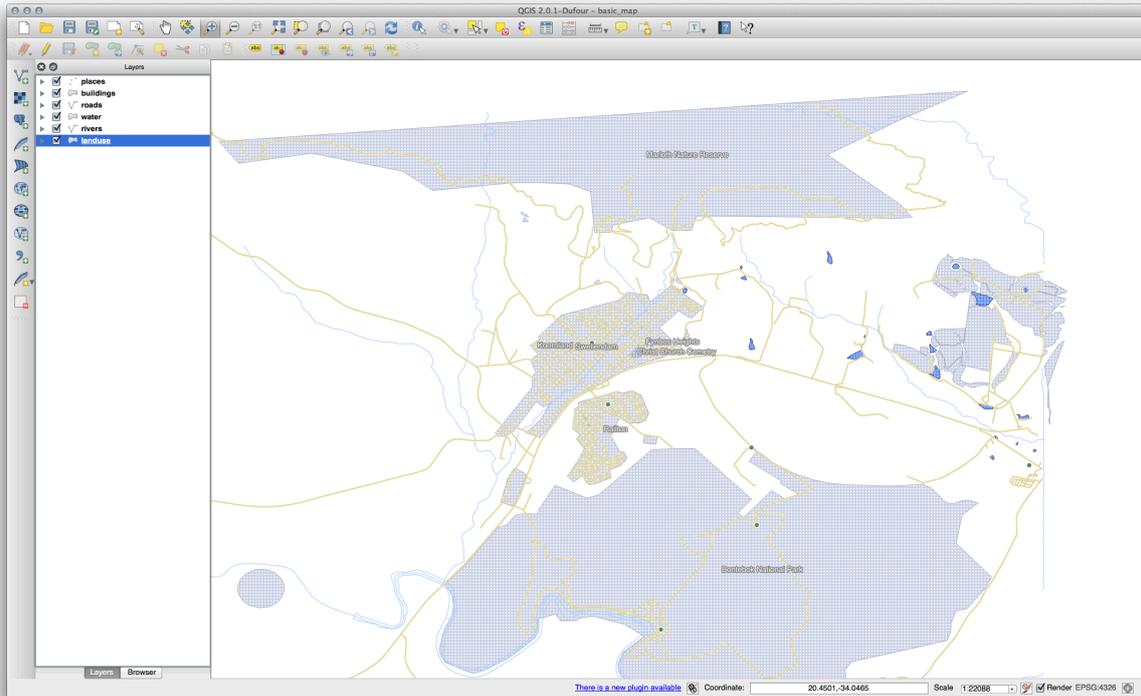
フォントの問題が解決されました! では、ポイントにラベルが重なる問題を見てみましょう。しかし、その前にバッファ オプションを見てみます。

- ラベルツール ダイアログを開きます。
- 左側のオプションリストから バッファ を選択します。
- テキスト縁取りの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テキスト書式設定を更新します:



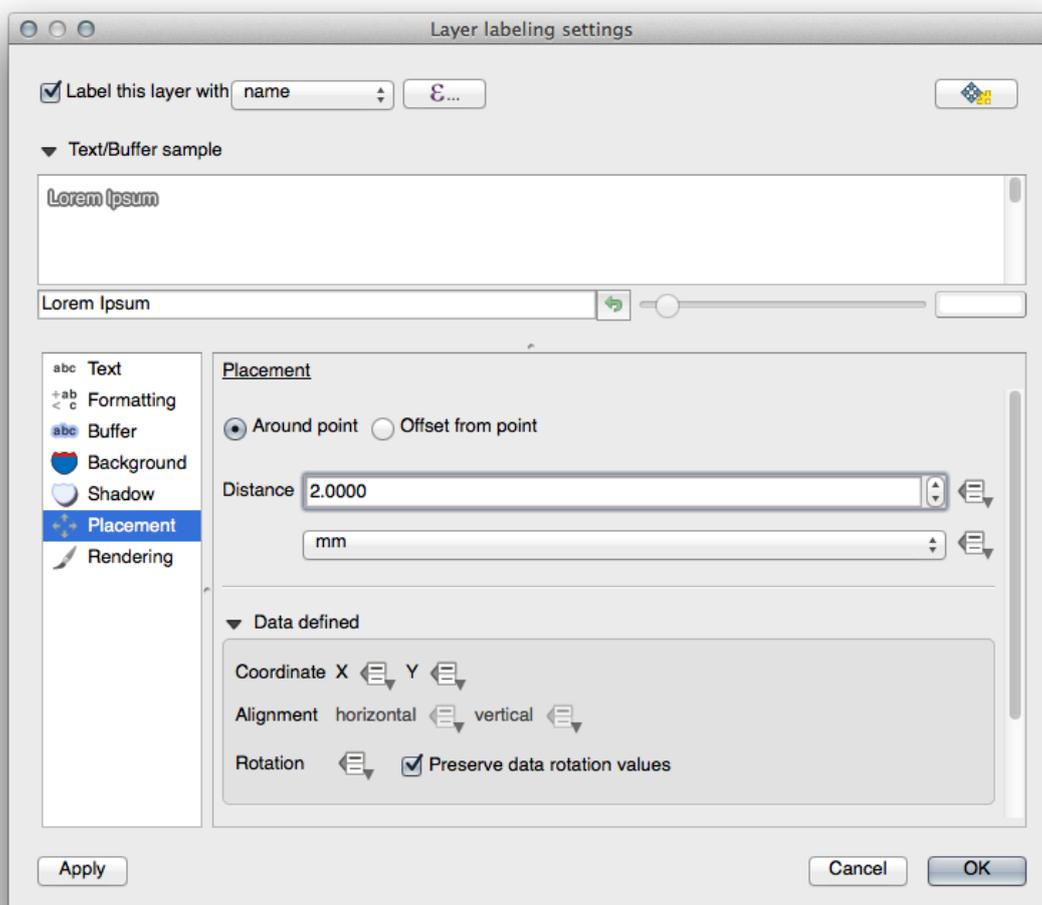
- 適用 をクリックします。

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。

- ラベルツール ダイアログで 配置 タブに移ります。
- 距離 の値を 2mm に変更して、ポイント周辺 が選択されていることを確認します:



- 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。

4.2.3 Follow Along: レイヤシンボロジの代わりにラベルを使用する

多くの場合、ポイントの位置はあまり的確である必要はありません。たとえば、*places* レイヤのポイントのほとんどは町全体や郊外を指し示し、そのような地物に関連するポイントは縮尺ではそれほど的確ではありません。実際には、詳細すぎるポイントを与えるとしばしば地図を読む人に混乱を招くことがあります。

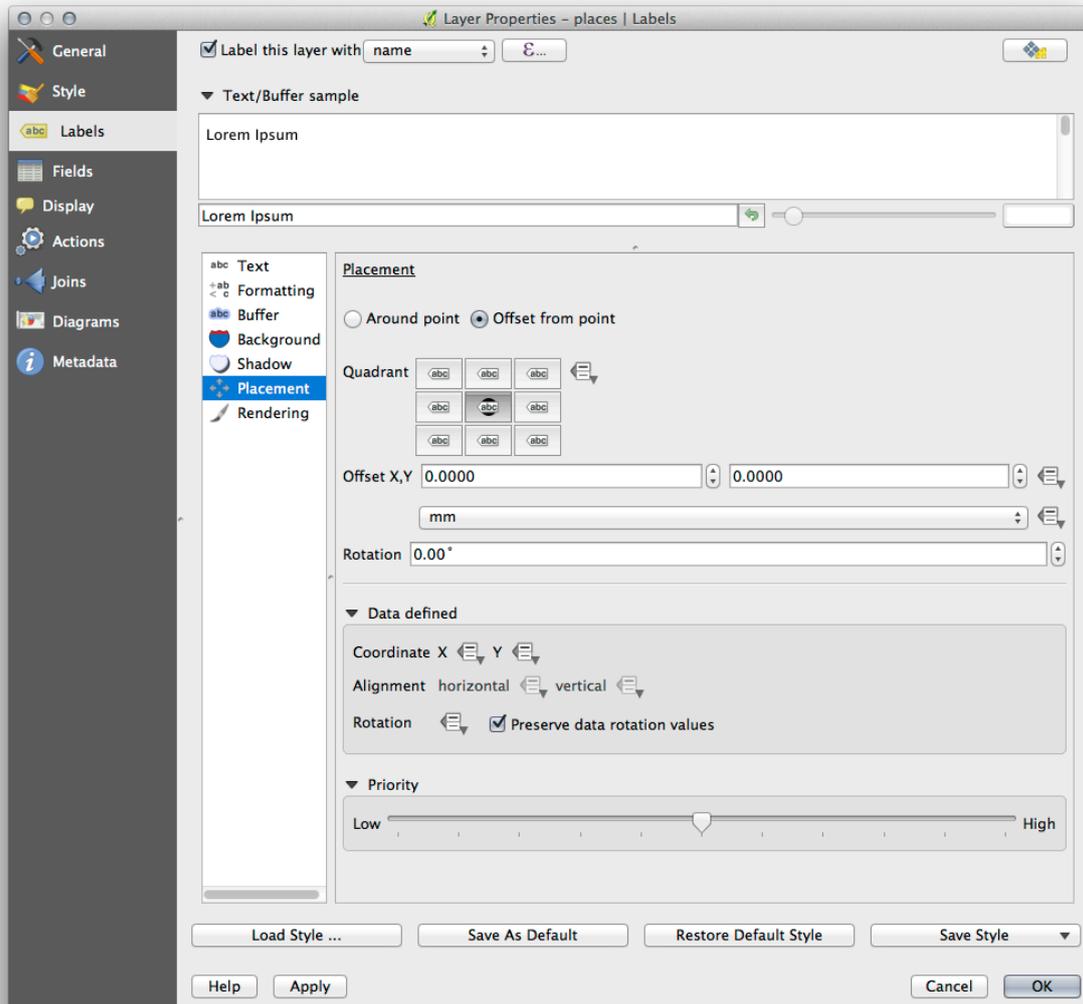
例を挙げると: 例えば、世界地図上では欧州連合のために与えられた点はポーランドのどこかにあります。*European Union* のラベルの付いたポイントがポーランドにあるので、地図を読んでいる人には欧州連合の首都はポーランドにあるように見える場合があります。

だから、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。

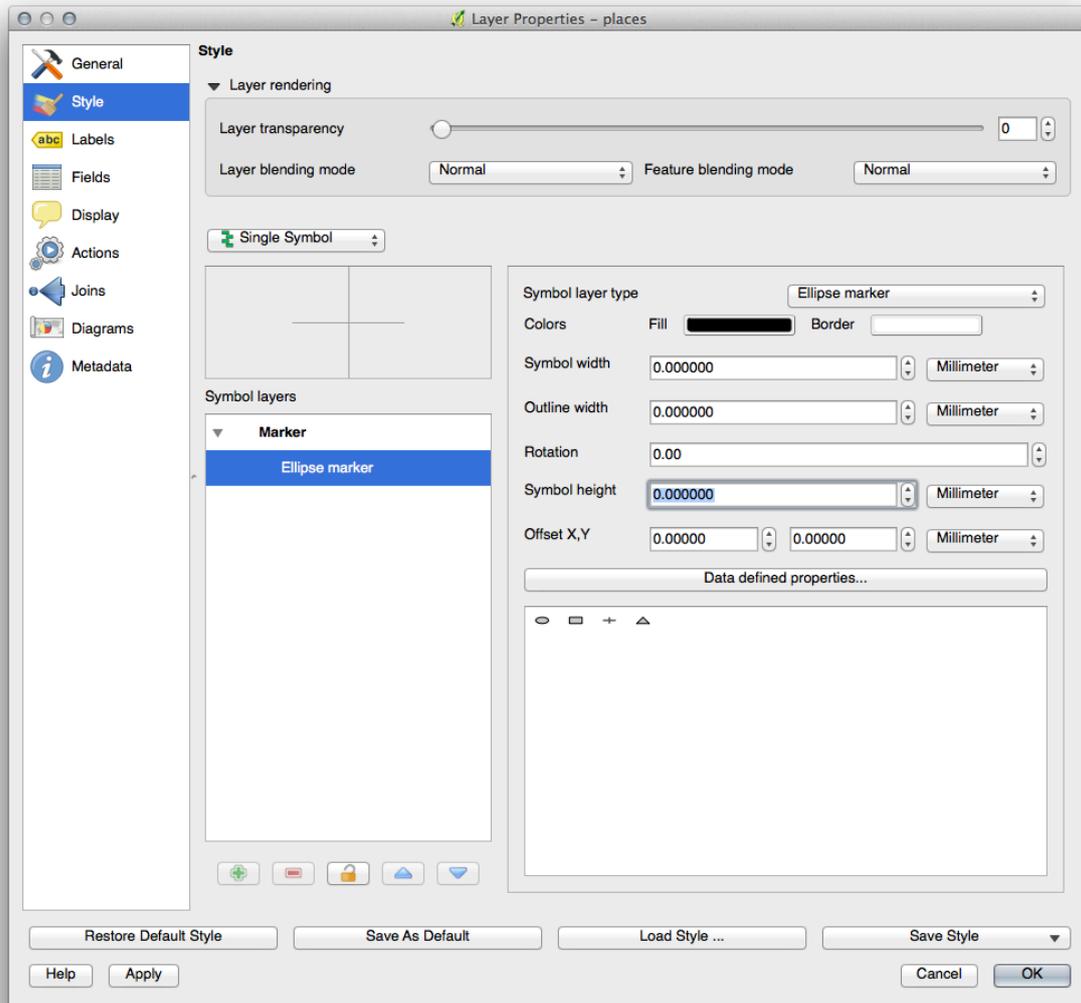
QGIS ではラベルが参照するポイントの直上にラベルの位置を変更することによってこれを行うことができます。

- *places* レイヤの レイヤラベリング設定 ダイアログを開きます。
- オプションリストの中から 配置 オプションを選択します。
- ポイントからのオフセット ボタンをクリックします。

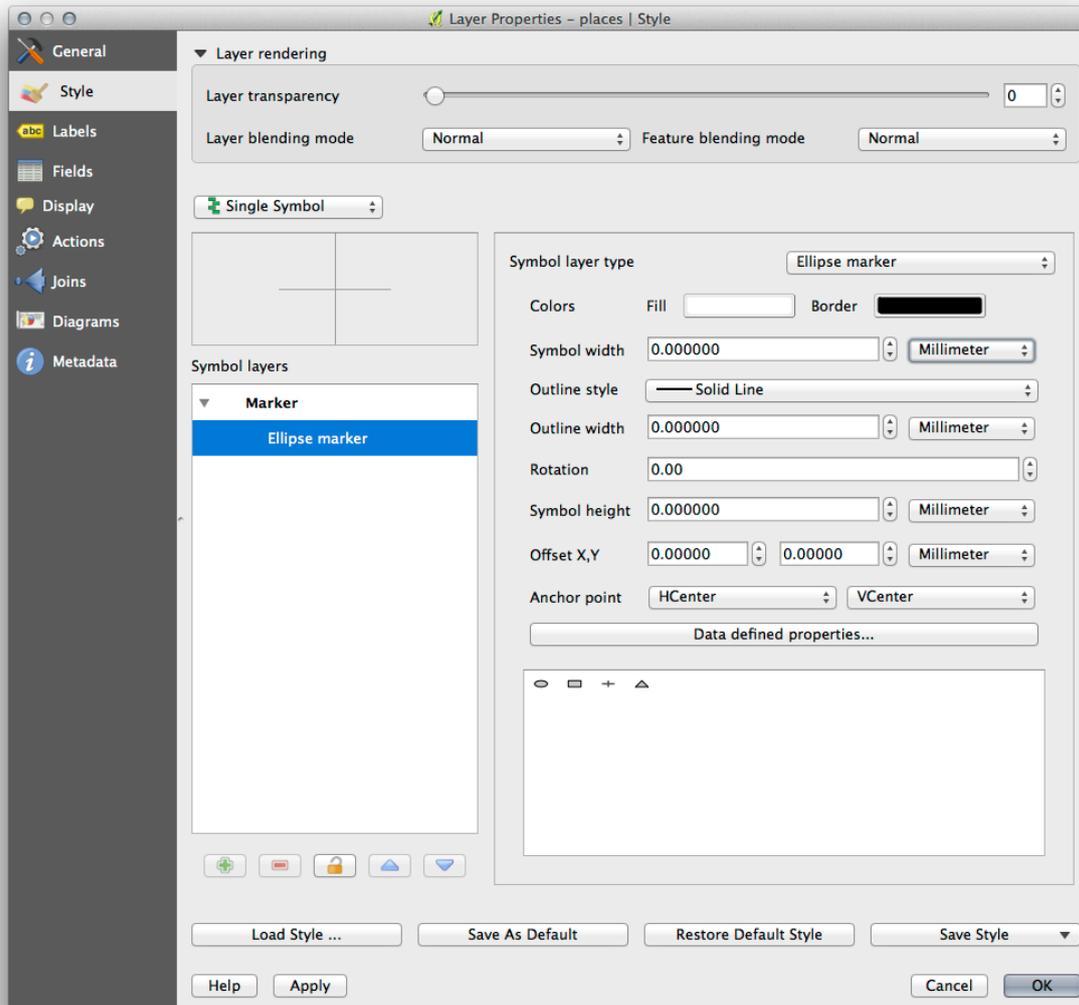
ポイントマーカースとの相対位置でラベルの位置を設定することのできる象限 オプションが現れます。この場合、私たちはラベルは点を中心に配置したいので中央の象限を選択します:



- ポイントシンボルを隠すためにいつものようにレイヤスタイルを編集して、楕円マーカースの幅と高さのサイズを 0 に設定します:



- OK をクリックすると、この結果が表示されます:



マップ上でズームアウトした場合にはラベルの一部がより小さな縮尺では重なりを避けるために消えていることがわかりました。これは多くのポイントを持つデータセットを扱う場合に都合がよい場合がありますが、有用な情報を失ってしまう場合もあります。このレッスンの後ほどのエクササイズでは、このような場合で処理するための別の手段を取り上げます。

4.2.4 Try Yourself ラベルのカスタマイズ

- シンボルとラベルの設定をポイントマーカと 2.00mm のラベルオフセットに戻します。この段階でポイントマーカやラベルのスタイリングを調整したいと思うことがあります。

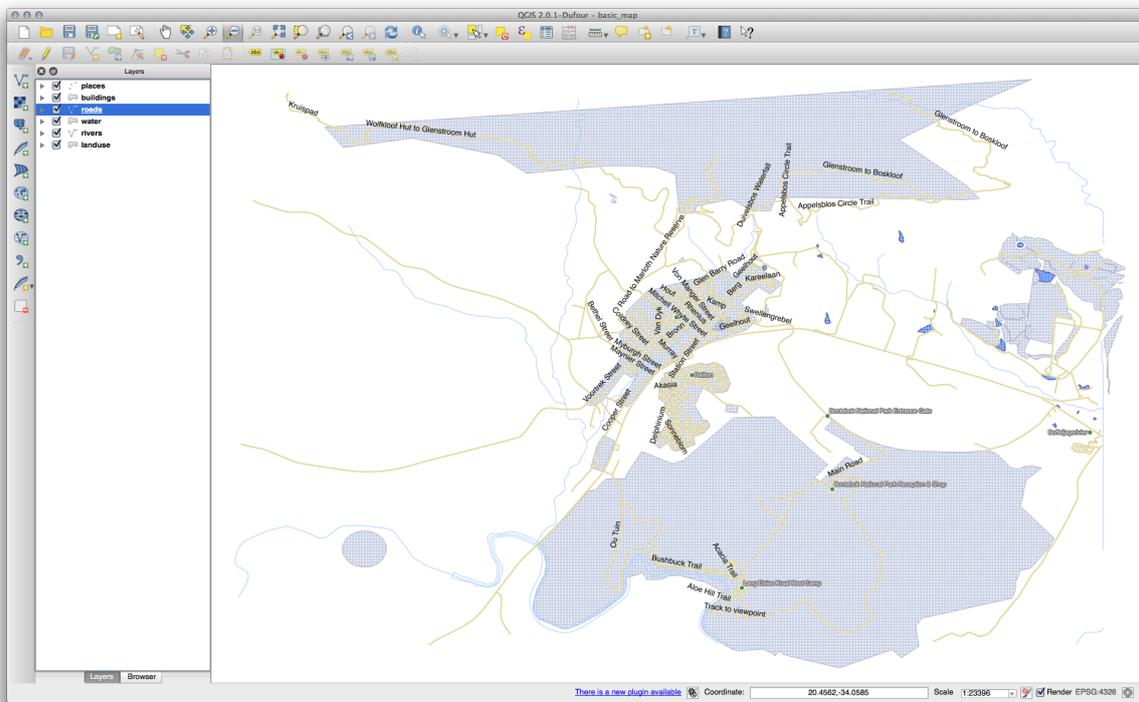
結果をチェックする

- 地図の縮尺を 1:100000 に設定します。ステータスバーのスケールボックスに入力することでこれを行うことができます。
- この縮尺での表示に合うようにラベルを変更します。

結果をチェックする

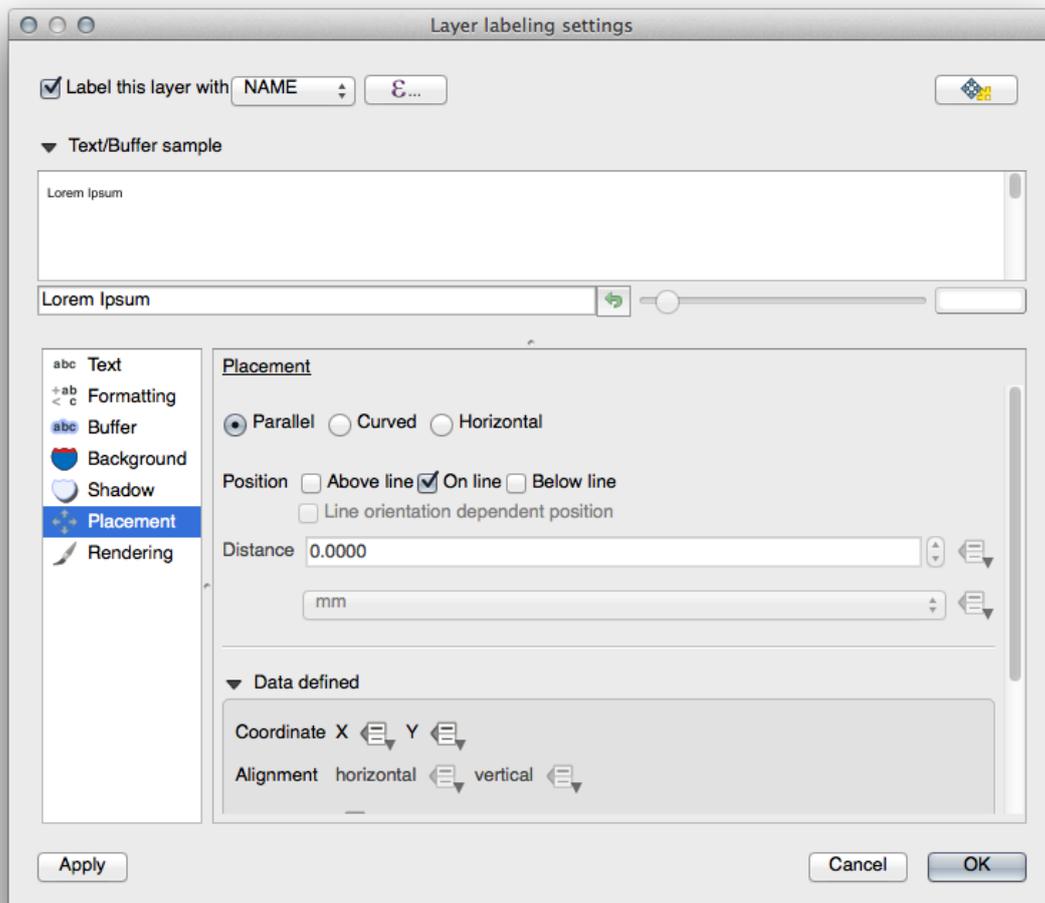
4.2.5 Follow Along: ラインのラベリング

今、あなたはラベリングがどのように動作するか知っていますが、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：



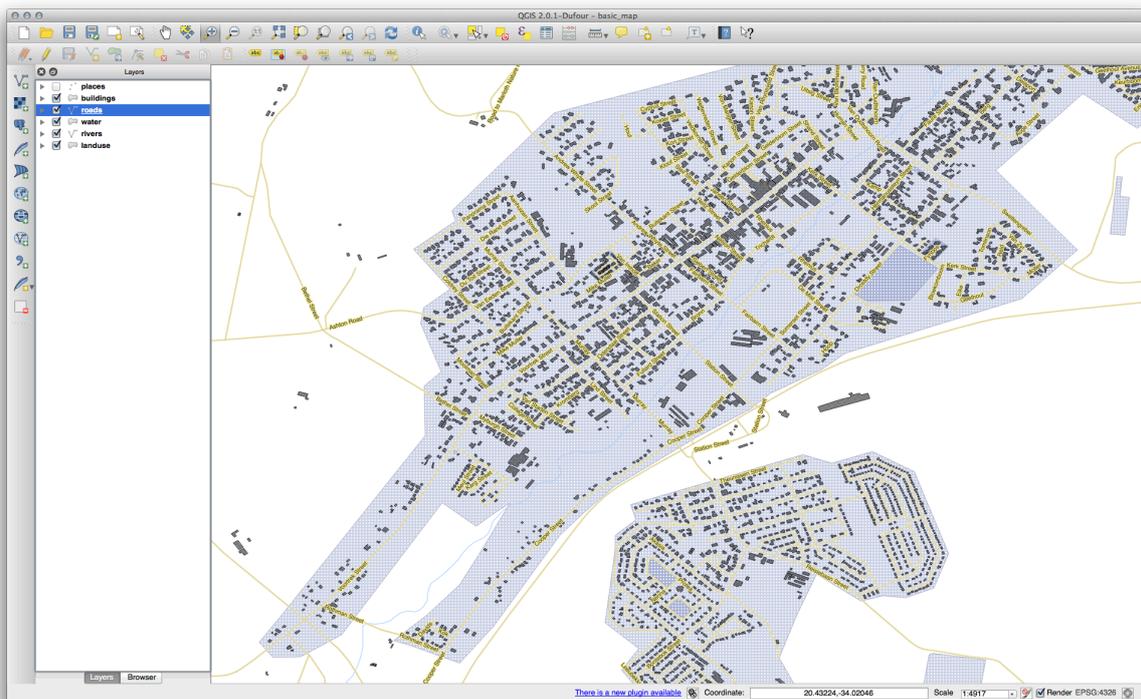
では、*roads* レイヤのラベルの体裁を理解しやすいように再調整します。

- 気が散らないように *Places* レイヤを非表示にします。
- *streets* レイヤのラベルを前のように有効化します。
- より多くのラベルが表示されるように **フォント サイズ** を 10 に設定します。
- Swellendam の市街地にズームインします。
- ラベルツール ダイアログの **配置 オプション**で次の設定をします：



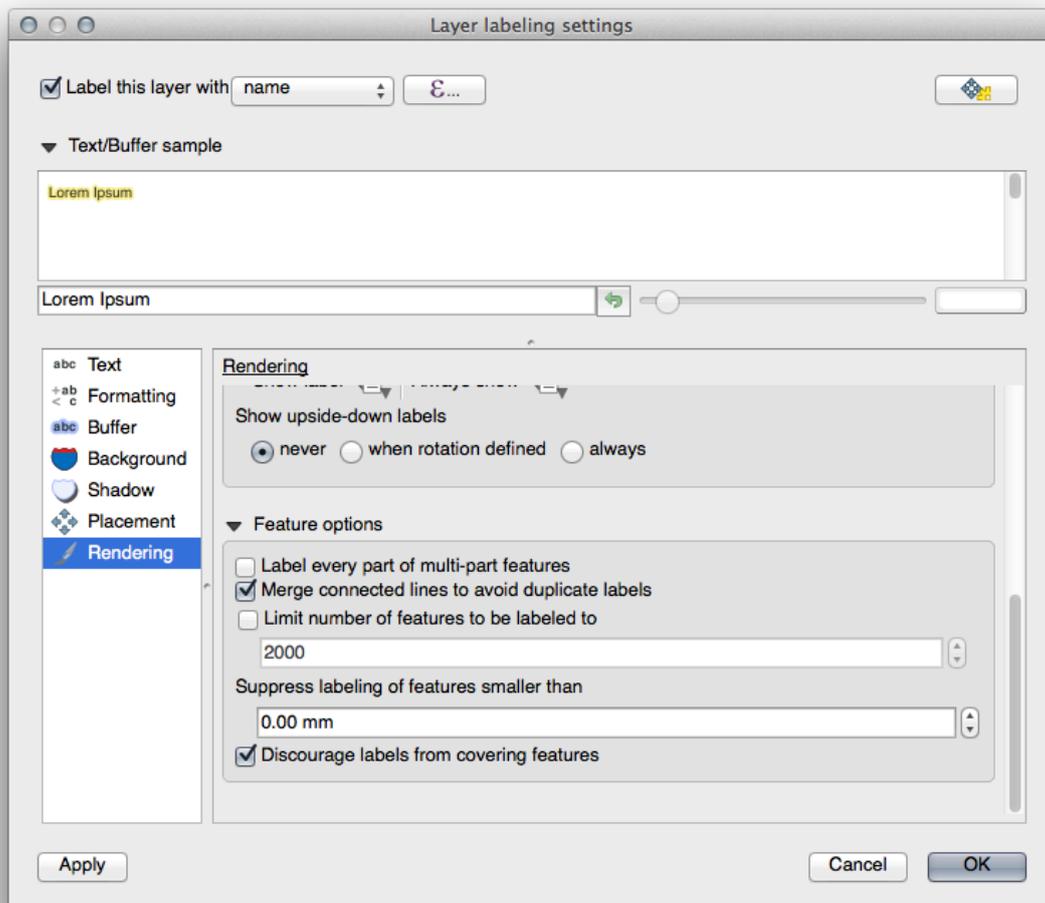
おそらく、テキストスタイルが既定値を使用していてその結果としてラベルは非常に読みづらいことがわかります。ラベルテキストの書式に暗い灰または黒の色と淡黄色のバッファを設定します。

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:

- レイヤラベリング設定 ダイアログで レンダリング オプションを選択し、重複ラベルを除去するために接続するラインを結合する を選択します:



- *OK* をクリックします

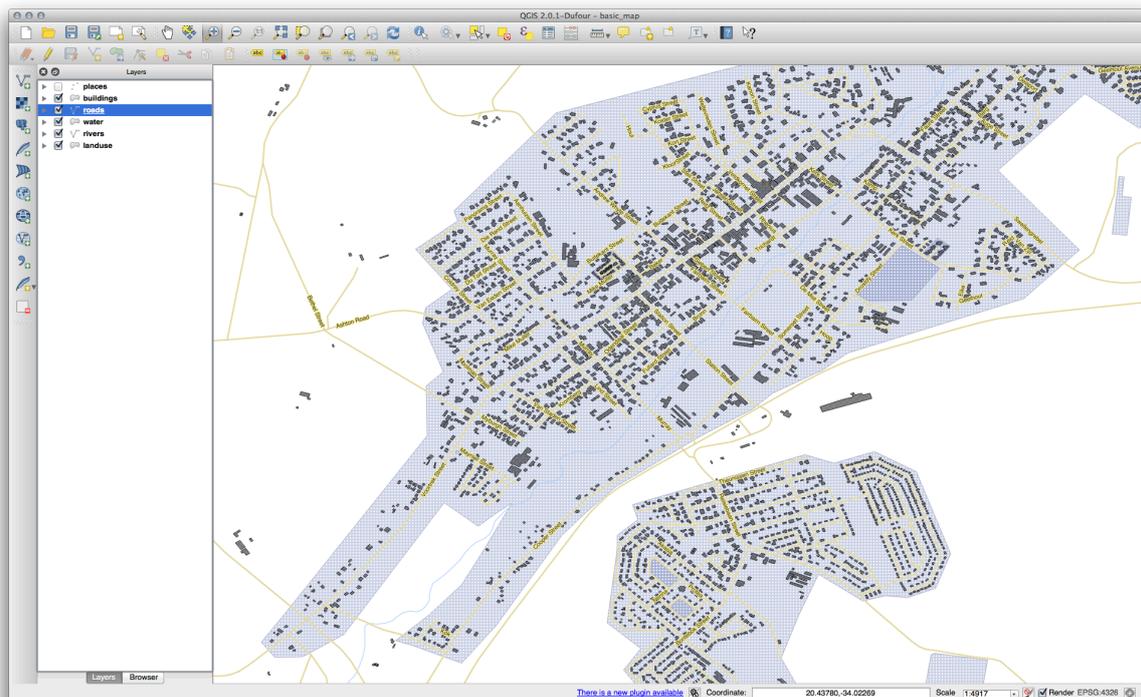
もう1つの有用な機能はラベルを付けるには短すぎる地物にラベルが描画されないようにするものです。

- 同じレンダリングパネルでこれより地物が小さい場合ラベリングを行わないの値を 5mm に設定します。適用 ボタンをクリックした時に表示される結果に注目して下さい。

同様にして異なる 配置 設定を試してみてください。前に見てきたように 水平 オプションはこの場合にはお勧めできません。その代わりに曲がって オプションを試してみましょう。

- レイヤラベリング設定 ダイアログの 配置 パネルで曲がって オプションを選択します。

この結果です:



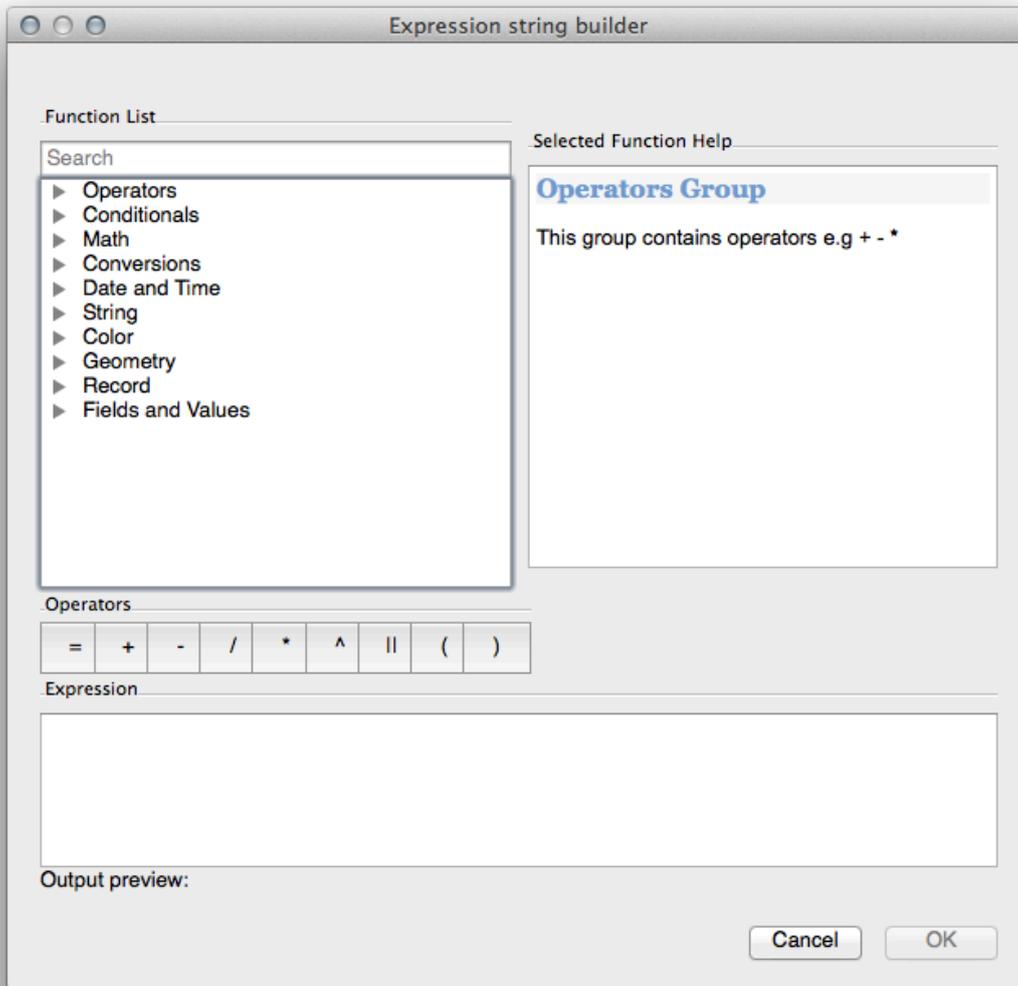
ご覧のように、それらのいくつかは曲がりくねった街路に沿わせると判読可能な状態ではなくなってしまふせいで以前表示されていたラベルの多くが非表示になります。より有用であるようにと考えるか、またはより格好がよくなるようにと考えるかによって、これらのオプションのどれを使うかを決めることができます。

4.2.6 Follow Along: データ定義による設定

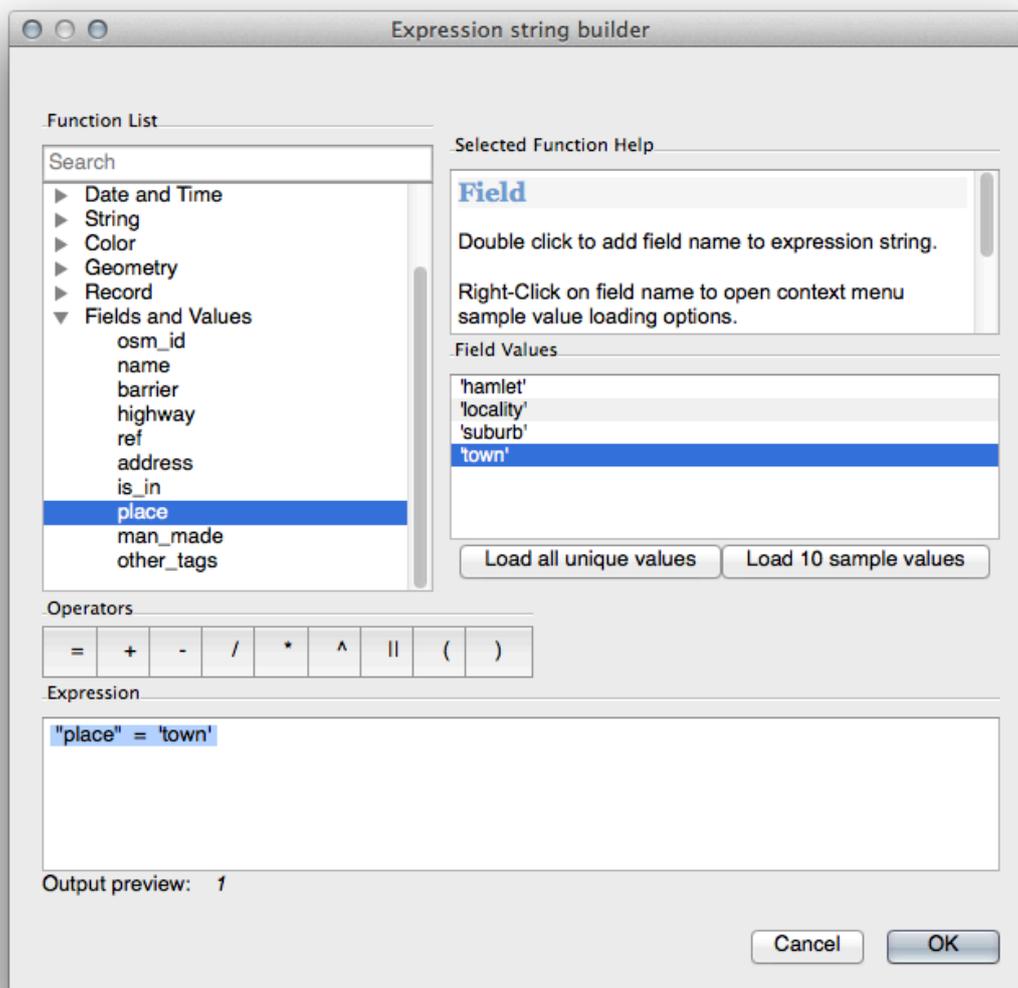
- *Streets* レイヤのラベリングを無効にします。
- *Places* レイヤのラベリングを再度有効にします。
-  ボタンをクリックして *Places* の属性テーブルを開きます。

今私たちに関心のあるフィールドが1つあります。place は各オブジェクトの都市エリアの種類を定義します。このデータを用いてラベルスタイルに影響を与えることができます。

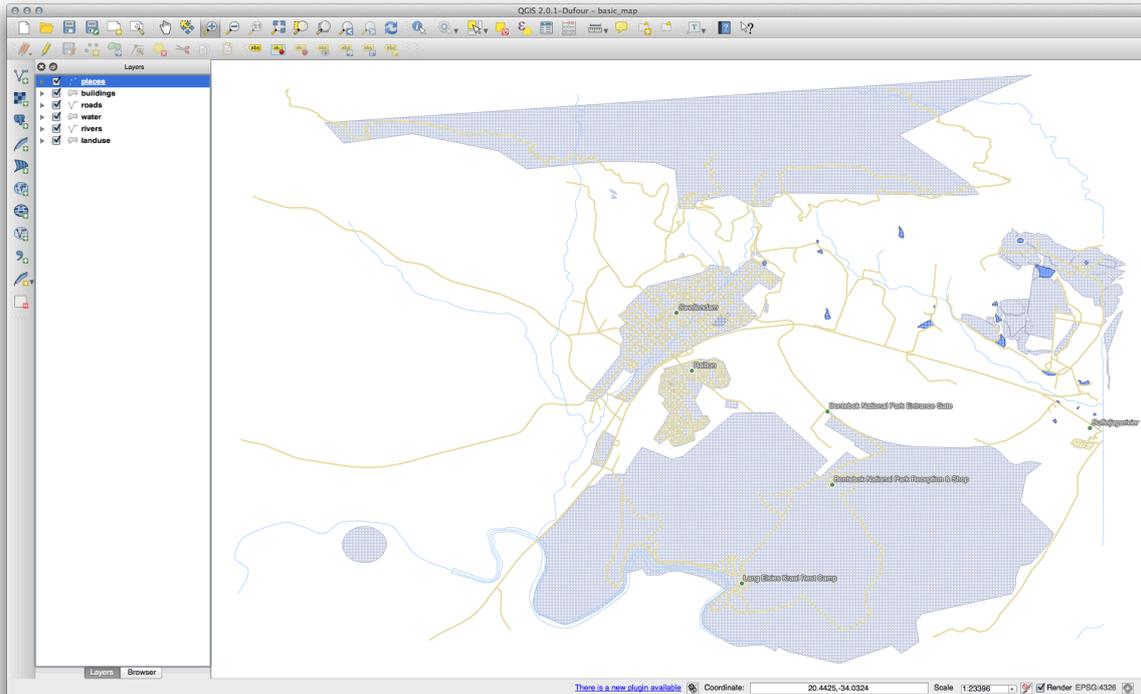
- *places* のラベルダイアログのテキストパネルに移動します。
- イタリアックテキストの右側のボタンをクリックし、編集... を選択して式文字列ビルダーを開きます:



テキスト入力で "place" = 'town' と入力し、Ok を 2 回クリックします:



その効果に注目してください:



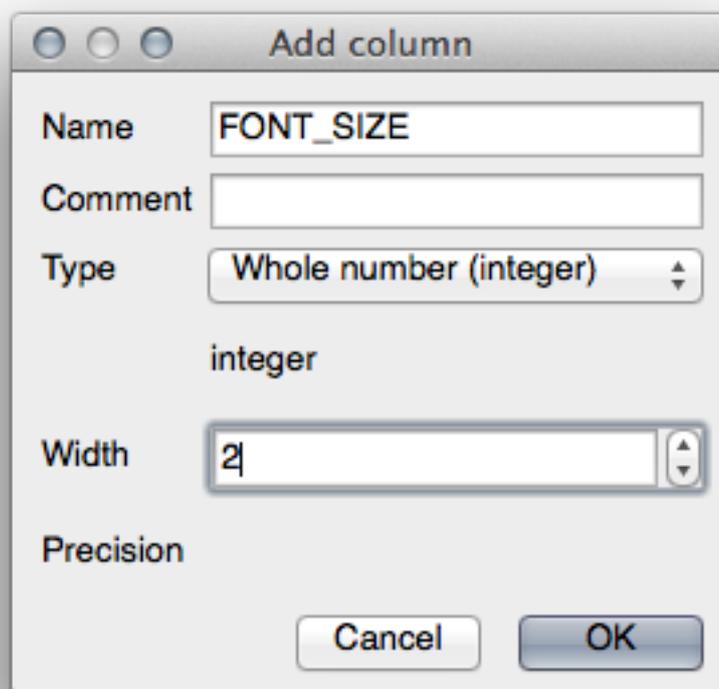
4.2.7 Try Yourself データ定義による設定の使用

ノート: 私たちはいくつかの高度なラベリング設定を示すために、ここで少し先にジャンプしています。高度な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スキップして必要なマテリアルをカバーした時に戻って来て下さい。

- *places* の属性テーブルを開きます。
- このボタンをクリックして編集モードにします: 
- 新しい列を追加します:



- このように構成します:



- これを使用して、それぞれの場所の種類に応じてカスタムフォントサイズを設定します (すなわち PLACE フィールドの各キー)。

結果をチェックする

4.2.8 ラベリングのさらなる可能性

このコースではすべてのオプションをカバーすることはできませんが、ラベルツールには他にも多くの有用な機能があることを覚えておいて下さい。縮尺ベースのレンダリングを設定したり、ラベルのレンダリングの優先順位を変更したり、すべてのラベルオプションをレイヤの属性を用いて設定したりすることができます。ラベルの回転や XY 位置、その他のプロパティを設定することもできます (この目的のために割り当てられた属性フィールドを持っている場合)。そして、これらのプロパティをメインのラベルツールの隣にあるツールを使って編集することができます:



(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。)

ラベリングシステムの可能性をもっと探してみてください。

4.2.9 In Conclusion

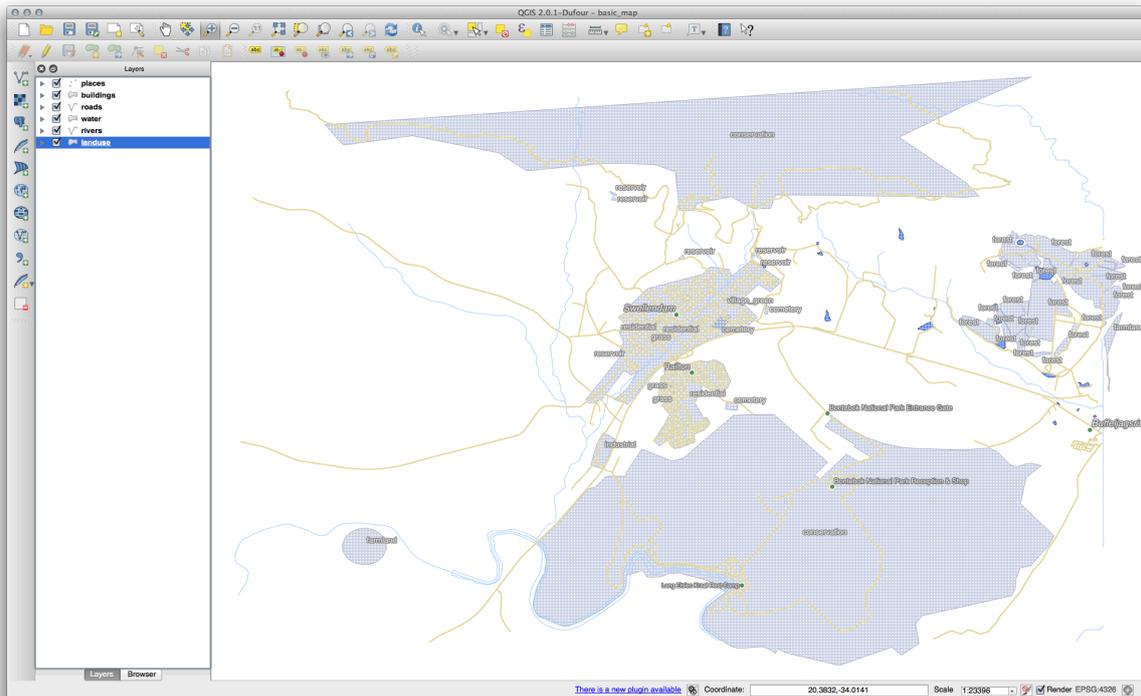
あなたは動的なラベル作成のためにレイヤの属性を使用する方法を学びました。これによりあなたの地図の情報量を増やし、地図をスタイリッシュにすることができます!

4.2.10 What's Next?

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!

4.3 Lesson: 分類

ラベルは個々の場所の名前などの情報を通信するためには良い方法ですが、すべてについては使用できません。例えば、誰かがそれぞれの *landuse* 地域が何に使用されているかを知りたいとしましょう。ラベルを使用すると、このようなものが得られます：

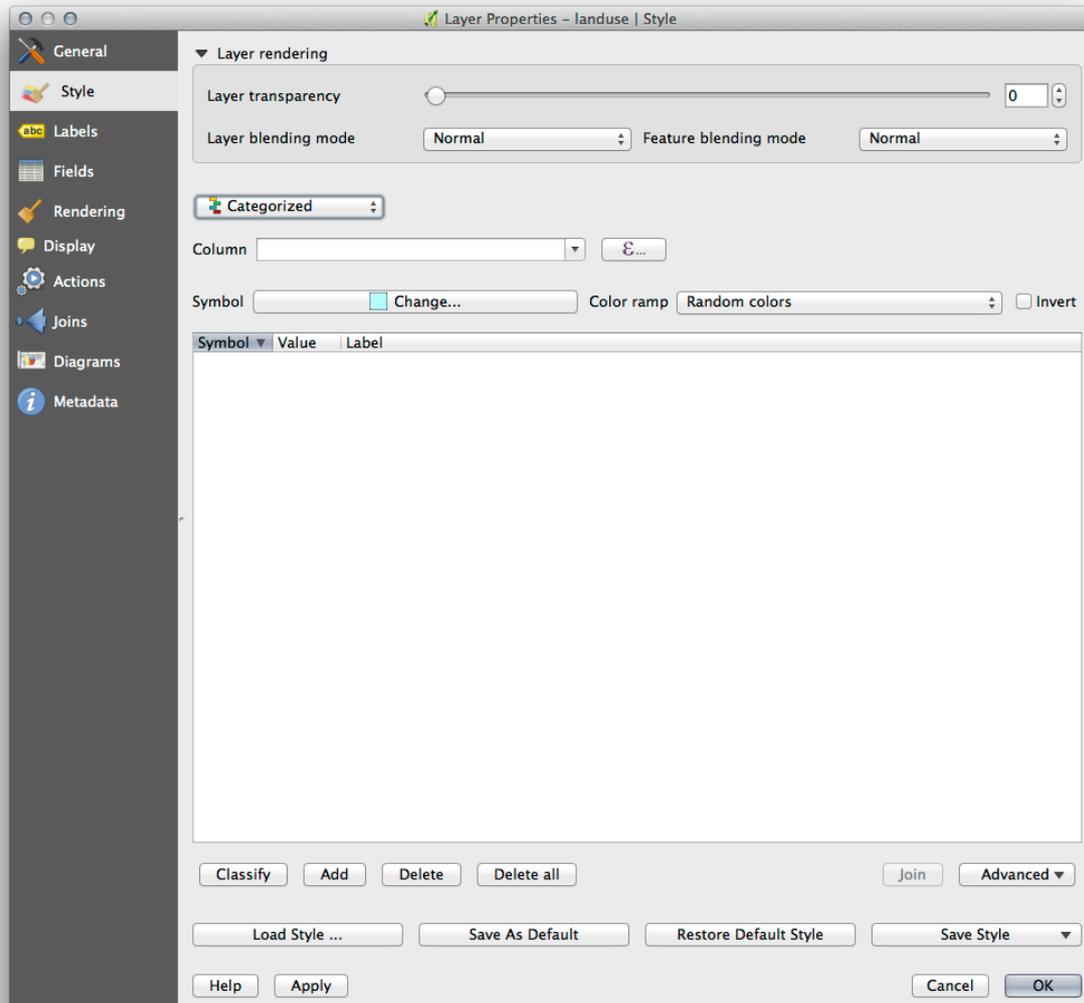


これでは地図の標識が読みにくくなり、地図上に多数の異なる土地利用の領域がある場合は圧倒的にさえなります。

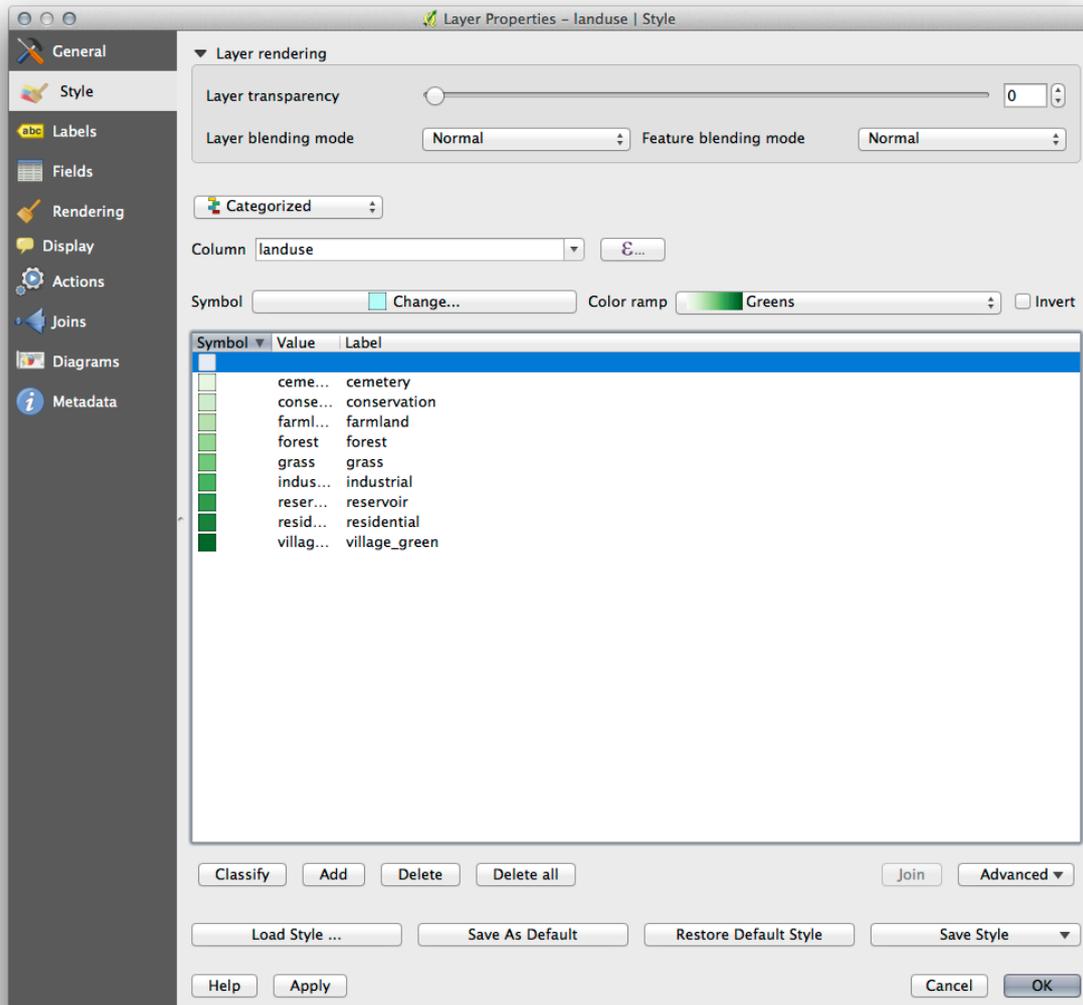
****このレッスンの目標：****効果的にベクトルデータを分類する方法を学習します。

4.3.1 Follow Along: 公称データを分類

- *landuse* 層のための レイヤ *Properties* ダイアログを開きます。
- *Style* タブに移動します。
- シングル *Symbol* 言うドロップダウンをクリックし、それを *Categorized* に変更します：

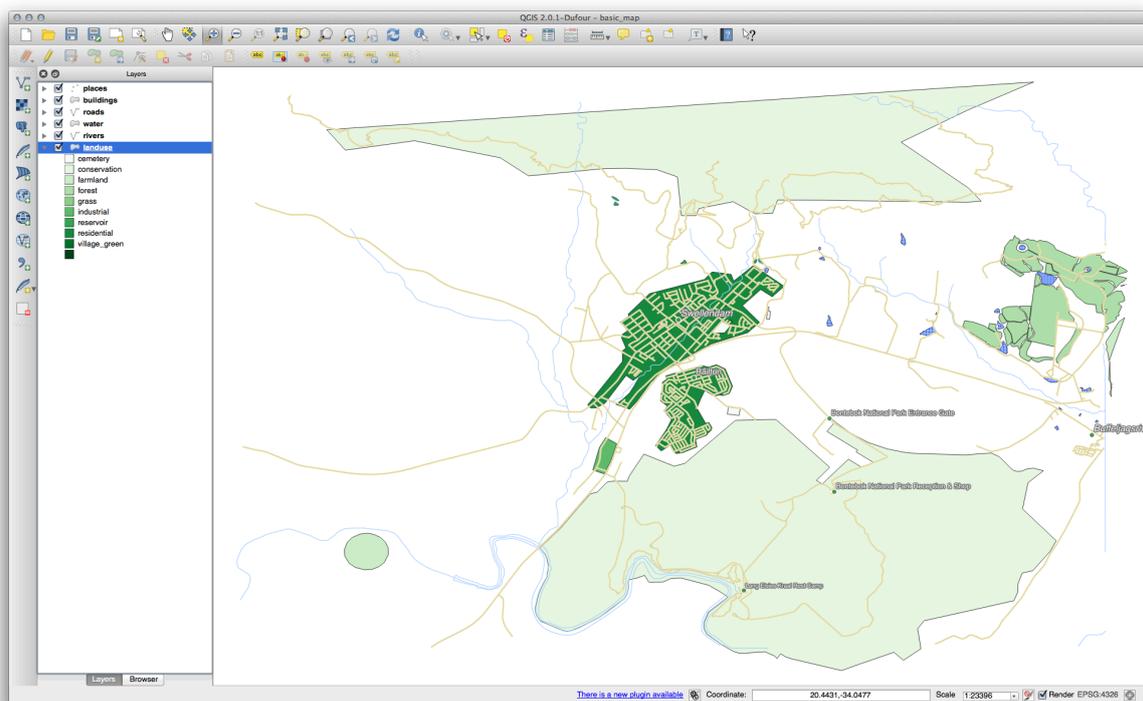


- COLUMN へ *landuse* と カラー ramp へ *Greens* 新しいパネルで、変更します。
- guilabel : ‘Classify’ というボタンをクリックしてください :

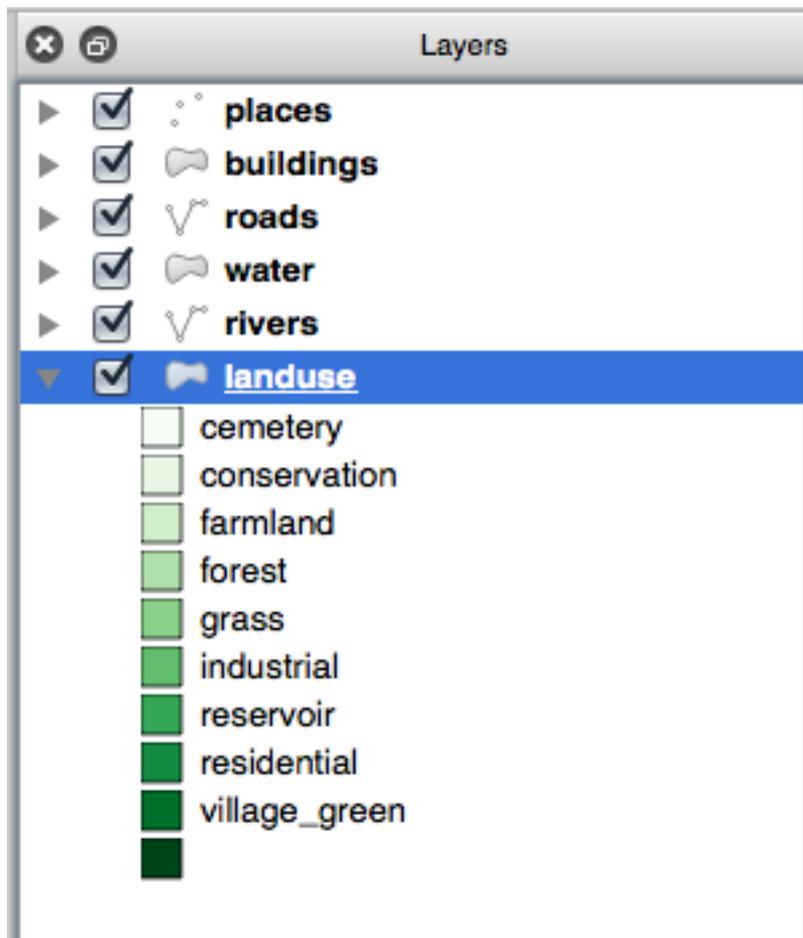


- Click *OK*.

このように表示されるはずですが:



- レイヤー list 中の *landuse* の隣の矢印 (またはプラス記号) をクリックすると、カテゴリが説明表示されます :



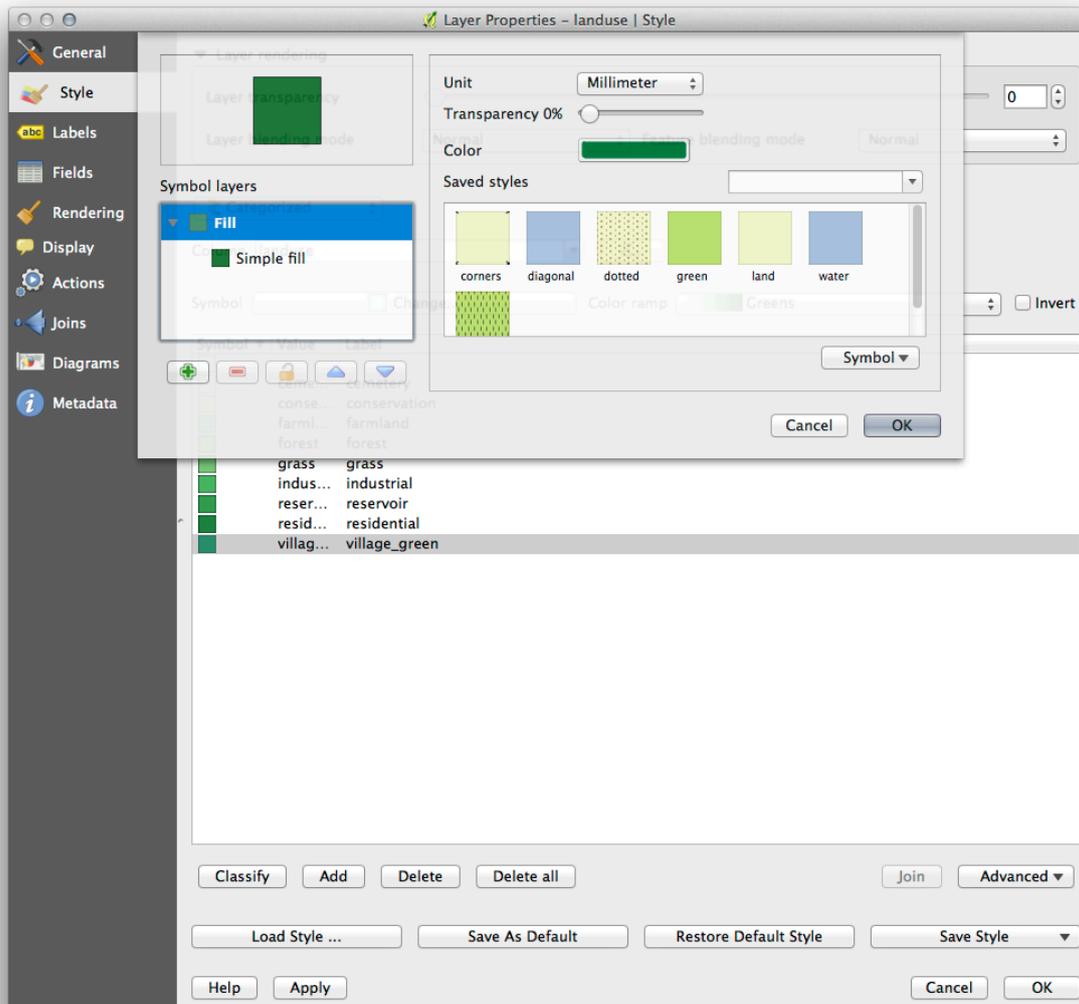
今、土地利用ポリゴンが適切に着色されていると、同じ土地利用との領域が同じ色になるように分類され

ています。 *landuse* レイヤーから黒枠を削除したいかもしれません。

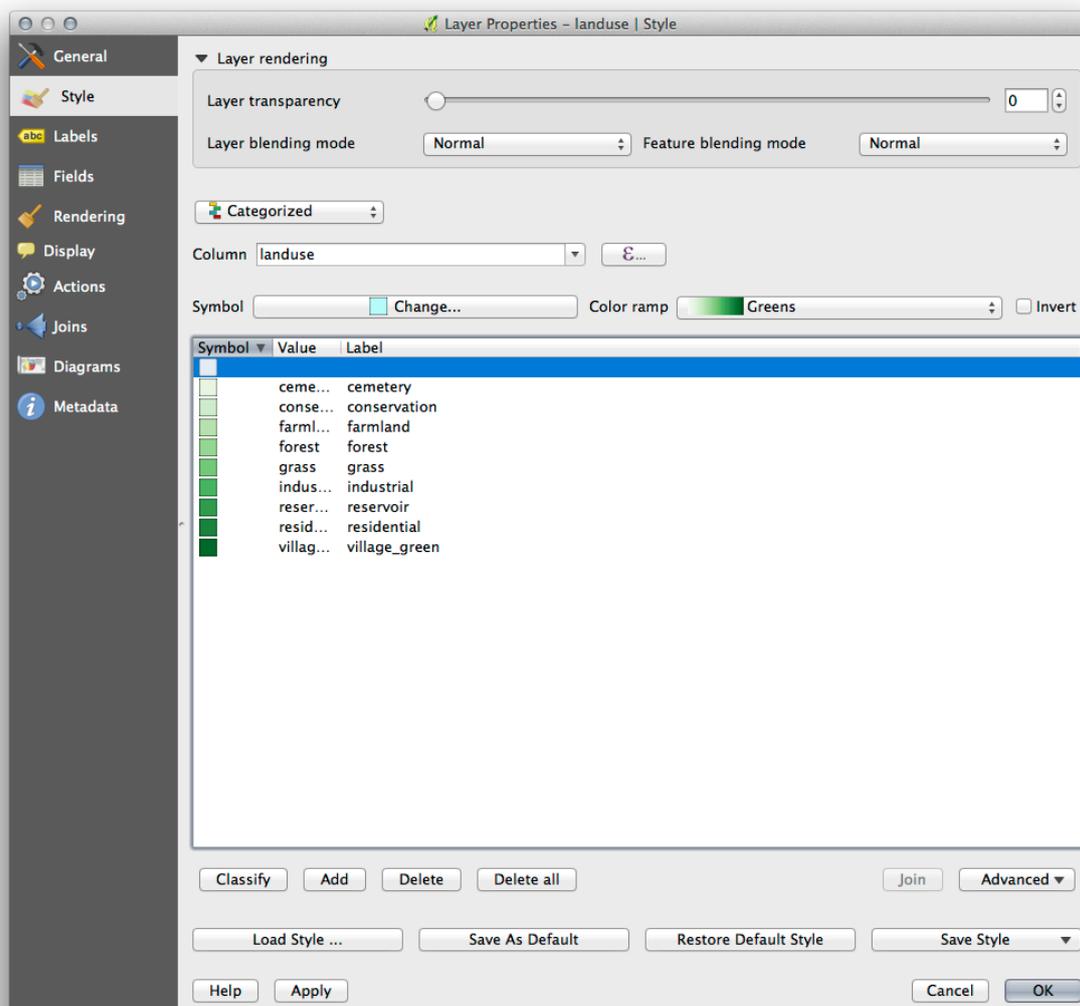
- レイヤー *Properties* を開き、 *Style*] タブに行き、 *Symbol* を選択。
- シンプル *Fill* レイヤーから境界を除去することによりシンボルを変更し、 *OK* をクリックしてください。

各分類のために私たちの新しい塗りつぶしの色を残し、土地利用ポリゴンのアウトラインが削除されていることがわかります。

- したい場合は、該当する色のブロックをダブルクリックすると、各土地利用の領域の塗りつぶし色を変更することができます。



空の一つのカテゴリーがあることに注意してください：



この空のカテゴリには、土地利用の値が定義されているか、*NULL* 値を持たない任意のオブジェクトを色付けするために使用されます。*NULL* 値を持つ領域がまだマップ上に表示されるように、この空のカテゴリを維持することが重要です。空白または *NULL* 値をより明らかに表現するように色を変更したいかもしれませんが。

すべての苦勞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！

4.3.2 Try Yourself その他の分類

基本的なレベルのコンテンツを以下しているだけの場合は、*buildings* レイヤーを分類するために、上記得られた知識を使用しています。*building* 列に対して分類を設定し *Spectral* のカラーランプを使用。

ノート：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。

4.3.3 Follow Along: 比分類

名義、順序、間隔 と 比率 : 分類の 4 つのタイプがあります。

名義分類では、オブジェクトがに分類されているカテゴリは名前ベースです。彼らは秩序を持っていません。たとえば、次のように町の名前、地区コード、など

順序分類では、カテゴリは、特定の順序で配置されています。例えば、世界の都市は、世界貿易、旅行、文化、などのためにその重要度に応じてランクを与られています

間隔分類では、数字は、正、負、ゼロの値を有するスケールです。例えば：凍結下/上海面下に上記高さ/温度(0) 等

比率分類では、数字のみ正のゼロ値を有するスケールです。例えば：絶対零度より高い温度(0 度ケルビン)、点からの距離、月当たりの所与の街路上のトラフィックの平均量、等

上記の例では、我々はそれをするによって投与される町に各ファームを割り当てる公称分類を用います。今、私たちは地域で農場を分類する比率の分類を使用します。

- (維持したい場合は) 保存スタイル... でボタン *Style* ダイアログをクリックして自分の土地利用のシンボルを保存します。

レイヤを再分類しているので、保存されていない場合は、既存のクラスが失われます。

- *Style* ダイアログ閉じます。
- *landuse* レイヤーの属性テーブルを開きます。

サイズによって土地利用区域を分類したいのですが、問題があります：彼らは、サイズフィールドを持っていないので、我々はものを作る必要があります。

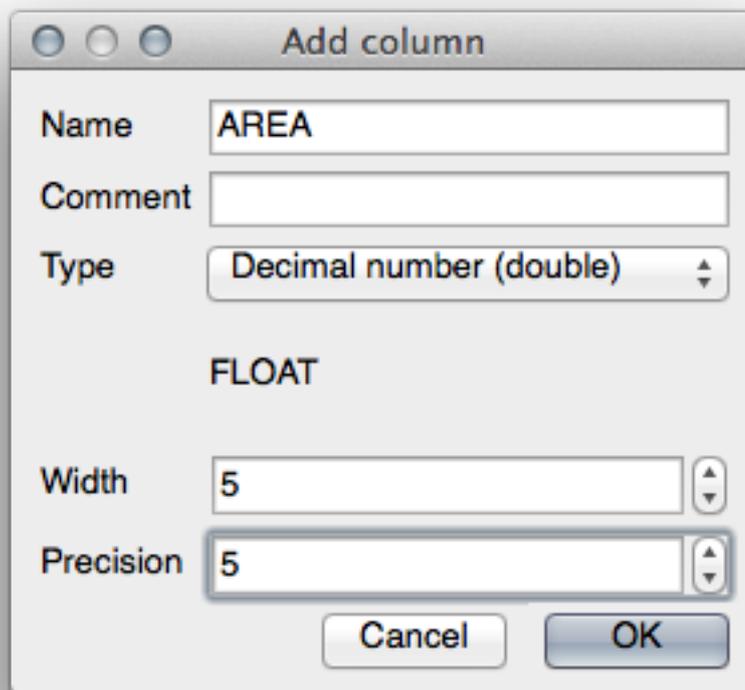
- このボタンをクリックして編集モードを入力します。



- このボタンで新しい列を追加します。



- このように、表示されたダイアログを設定します。



- Click *OK*.

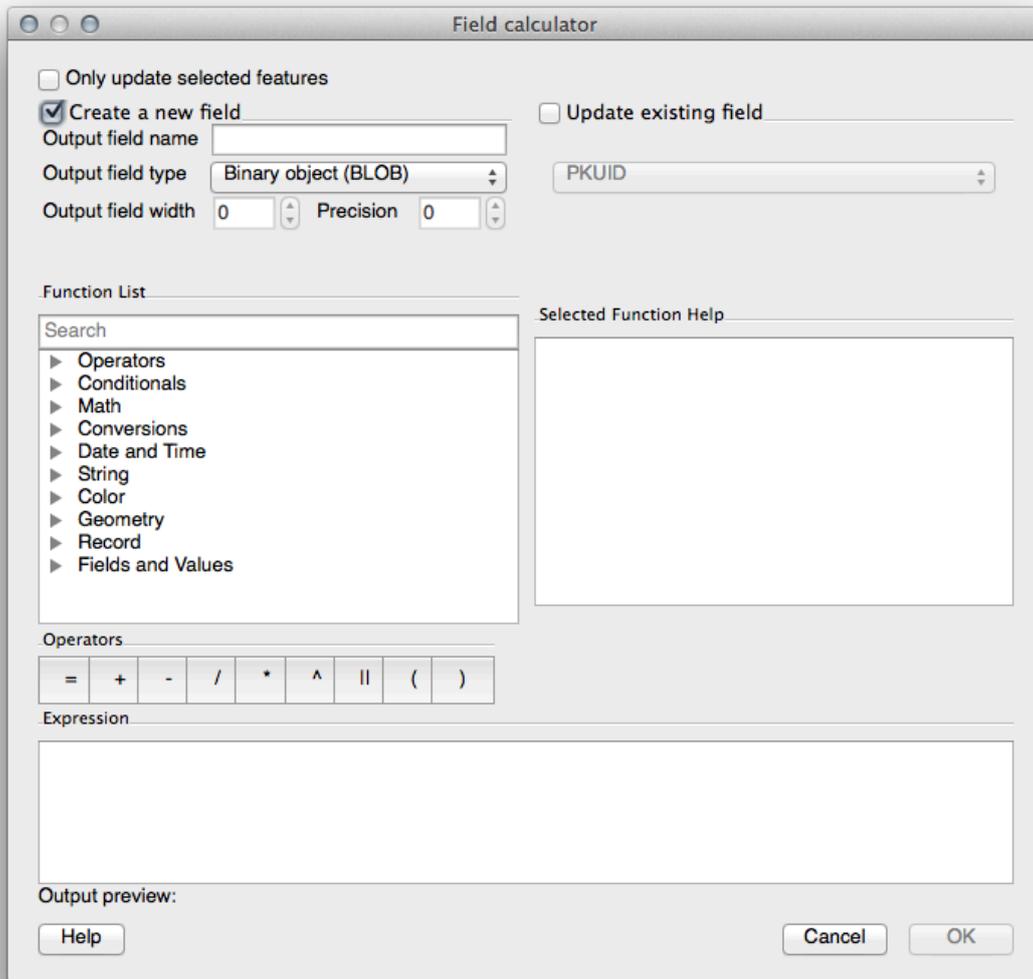
新しいフィールドが (表の右端に、それを見るために水平方向にスクロールする必要がある場合があります) 追加されます。NULL 値、現時点ではそれが読み込まれていないが、それだけをたくさん持っています。

この問題を解決するために、我々は領域を計算する必要があります。

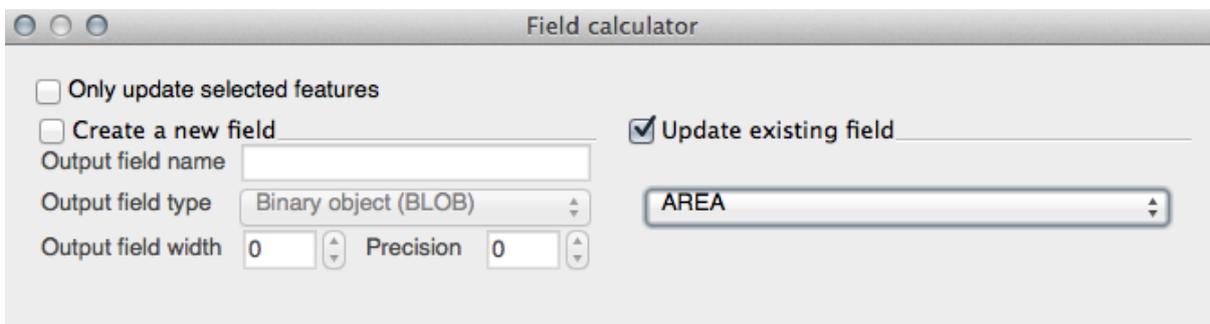
- フィールドの電卓を開きます。



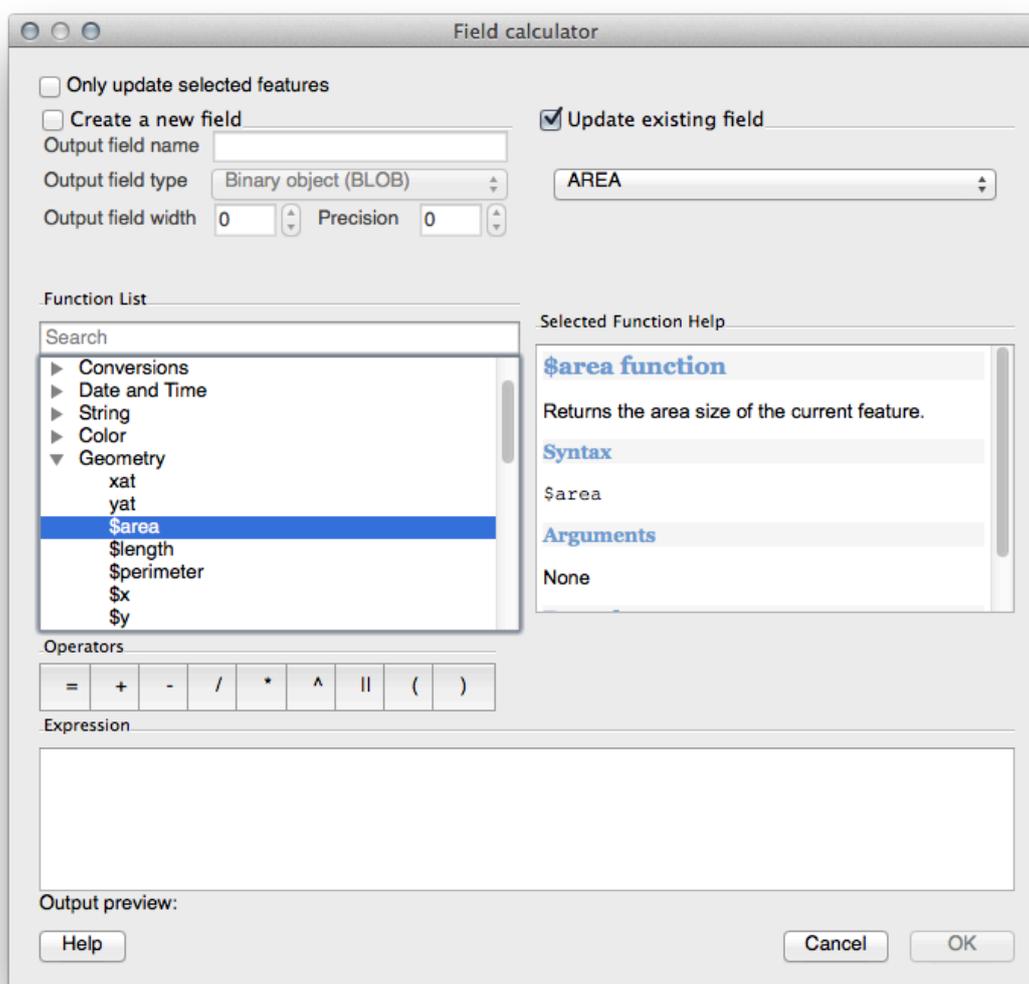
このダイアログを取得します：



- このように見えるように、ダイアログの上部にある値を変更します。



- 関数 List 中で、ジオメトリを -> \$は area を選択：

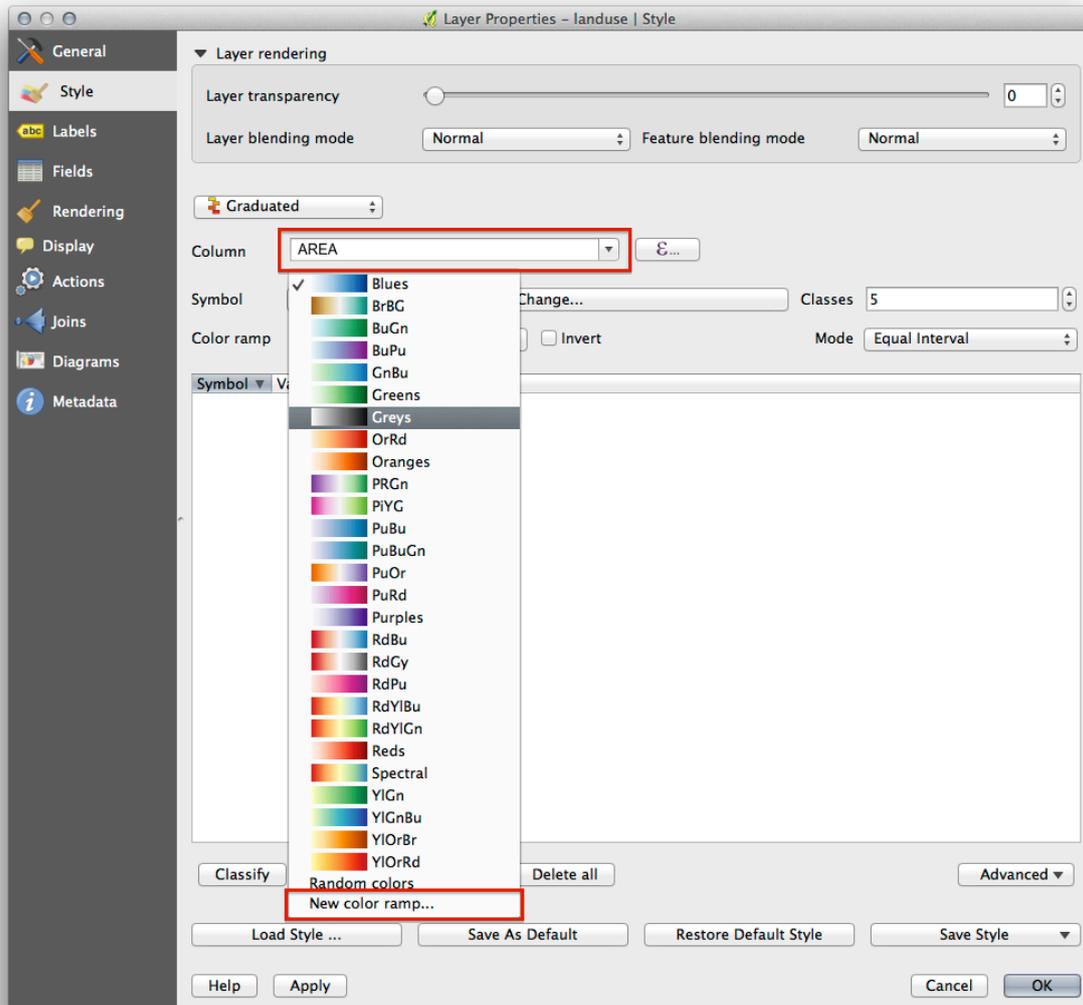


- *Expression* フィールドに表示されるように、それをダブルクリックします。
- Click *OK*.

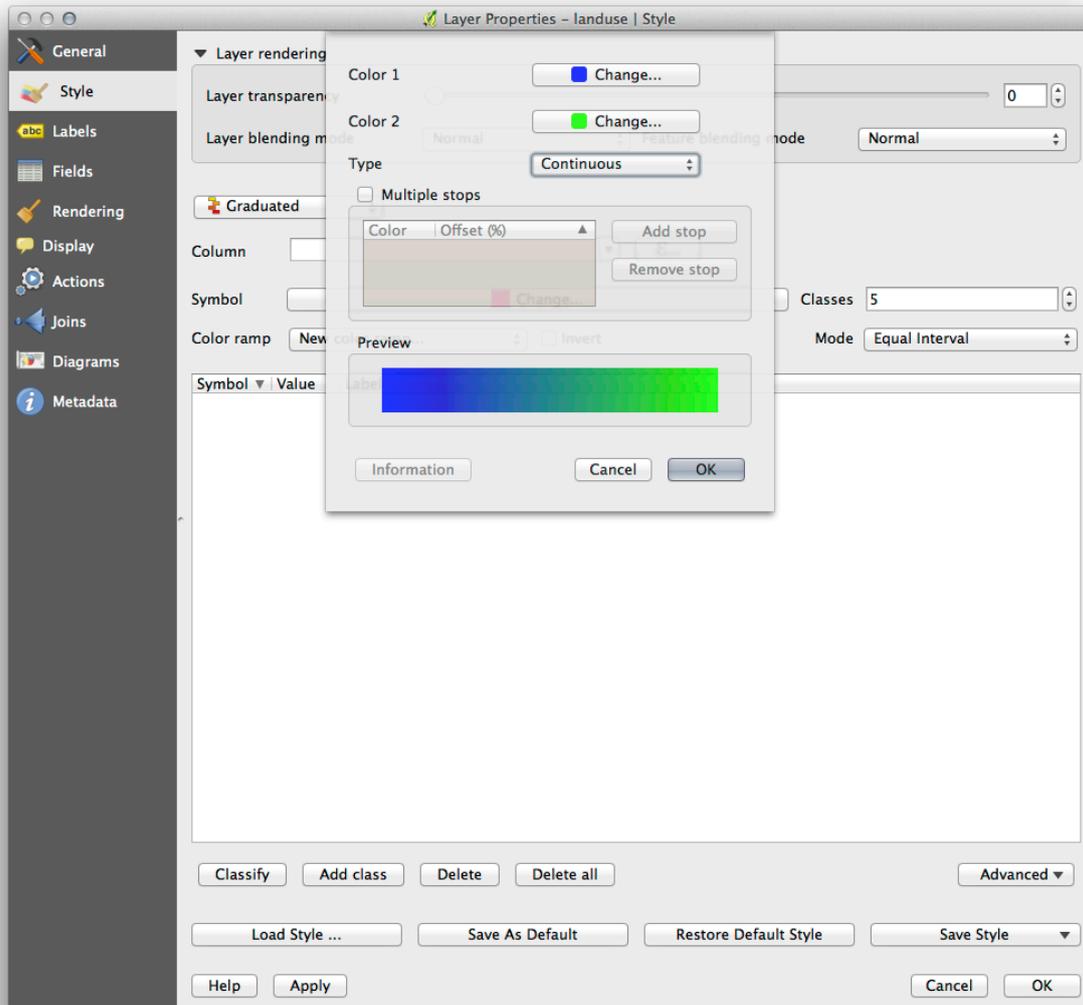
今すぐ AREA フィールドは、(データをリフレッシュするために列ヘッダをクリックする必要があるかもしれませんが)の値が取り込まれます。編集内容を保存して *Ok* をクリックしてください。

ノート: これらの領域は度です。その後、我々は、平方メートルでそれらを計算します。

- `guiLabel` : レイヤー *properties* ダイアログの `guiLabel` : `Style` タブを開きます。
- Change the classification style from *Categorized* to *Graduated*.
- *COLUMN* を *AREA* に変更します :
- 以下の下 : `guiLabel` : カラー *ramp*、オプションを選択します : `guiLabel` : ‘新しいカラーランプを...’ このダイアログを取得します :



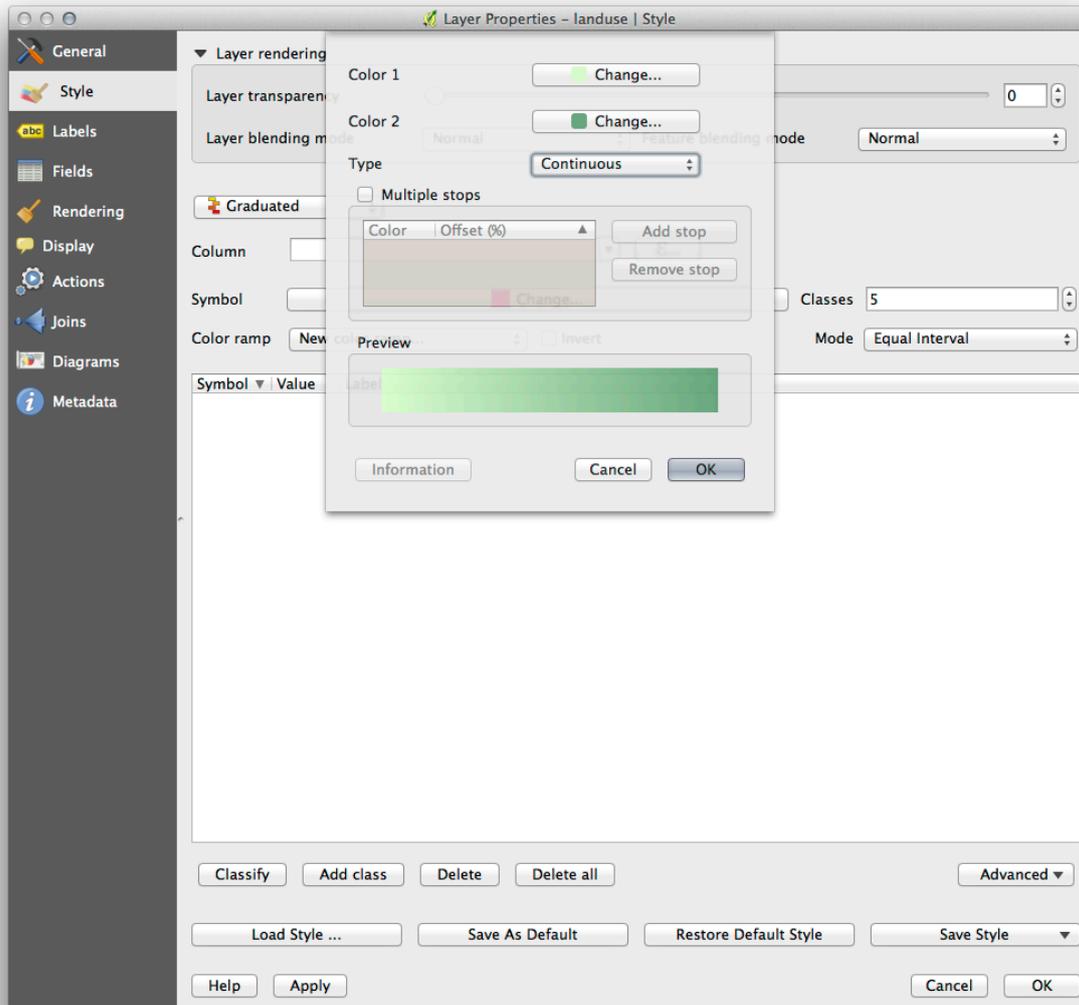
- Gradient 選択して（それが選択されていない場合）OK をクリックします。この表示されます：



これを使用して、小さな面積の地域は:guilabel:1 の色、大きな面積 2 色で表示します。

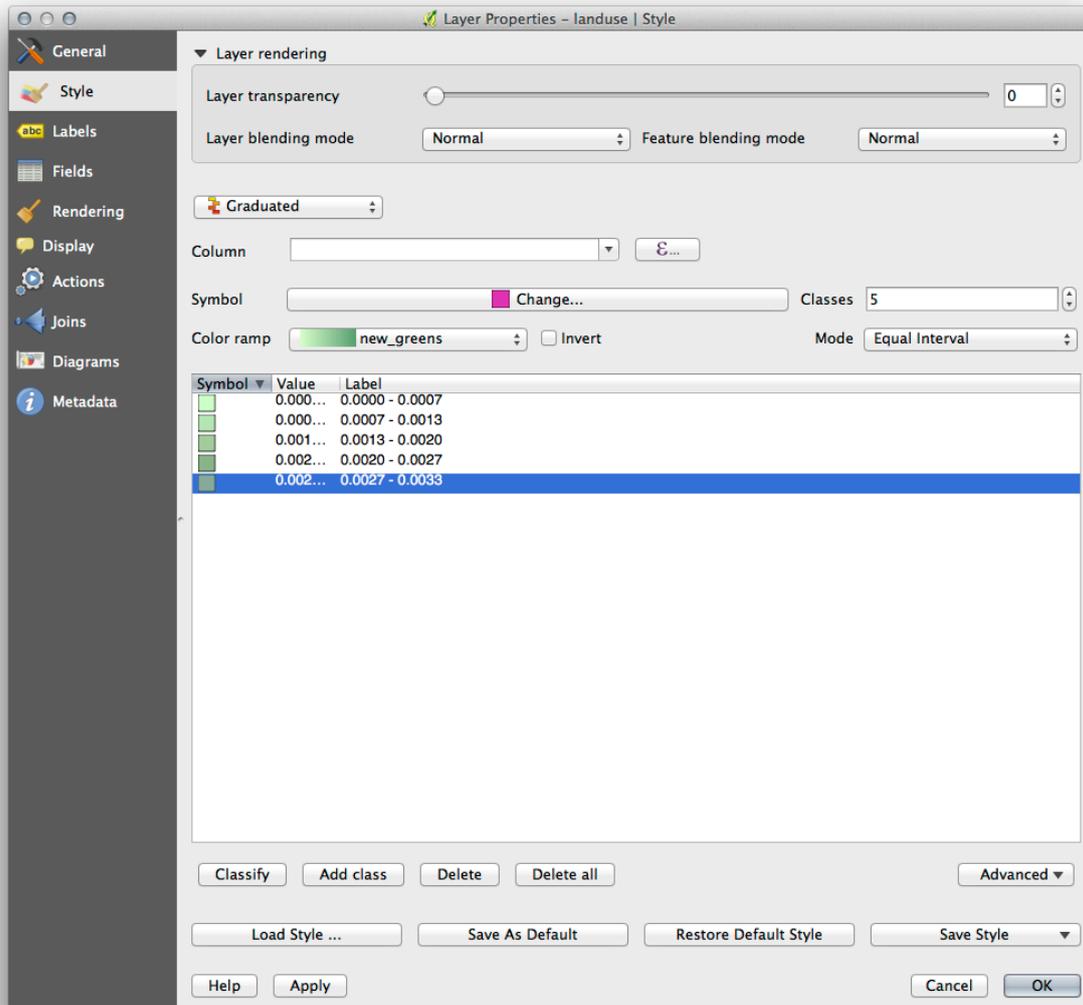
- 適切な色を選択してください。

例では、結果は次のようになります。



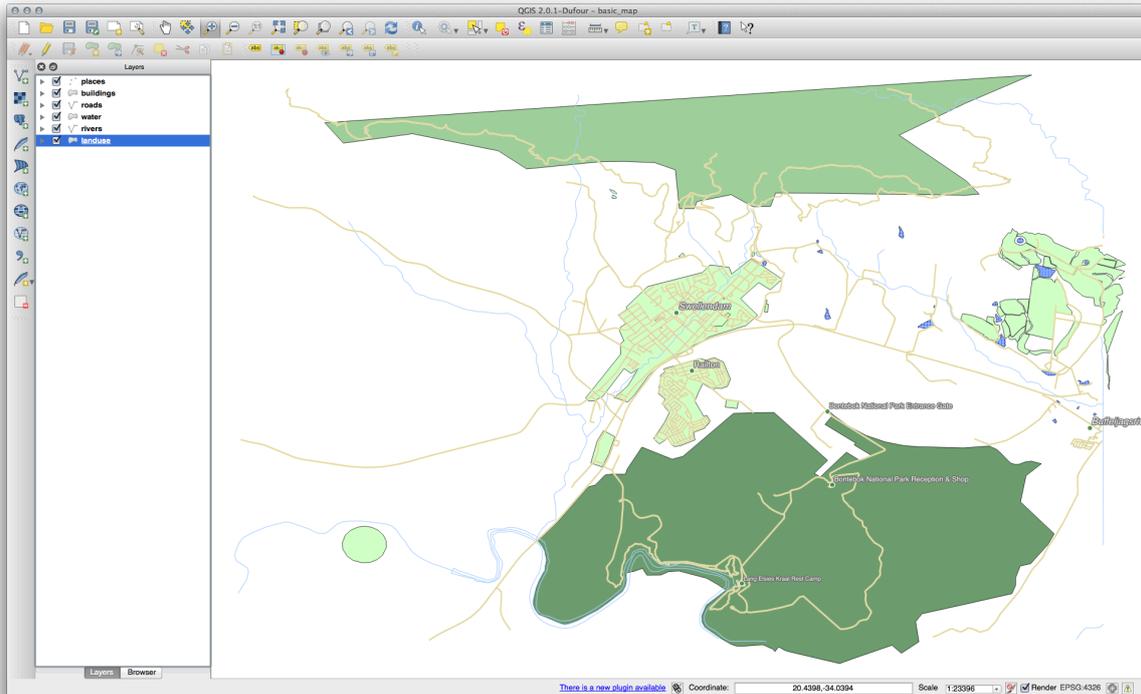
- Click *OK*.
- 新しいカラーランプに適した名前を選択します。
- 名前に充填した後 *OK* をクリック。

今、あなたはこのようなものがあるでしょう：



そのまま他のすべてを残します。

- *Ok* をクリック :



4.3.4 Try Yourself 分類の絞り込み

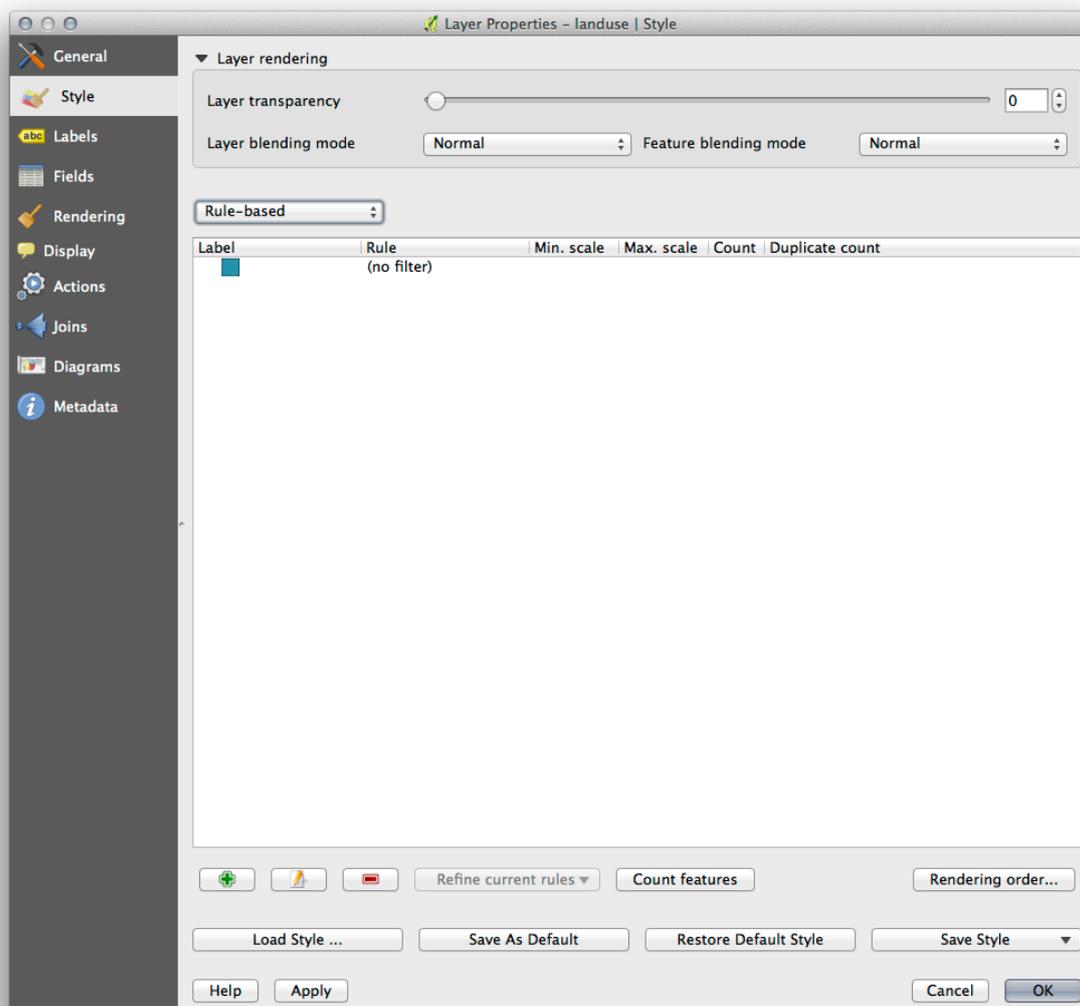
- クラス間の線を取り除きます。
- 理にかなった分類を得るまでモードと *Classes* の値を変更します。

Check your results

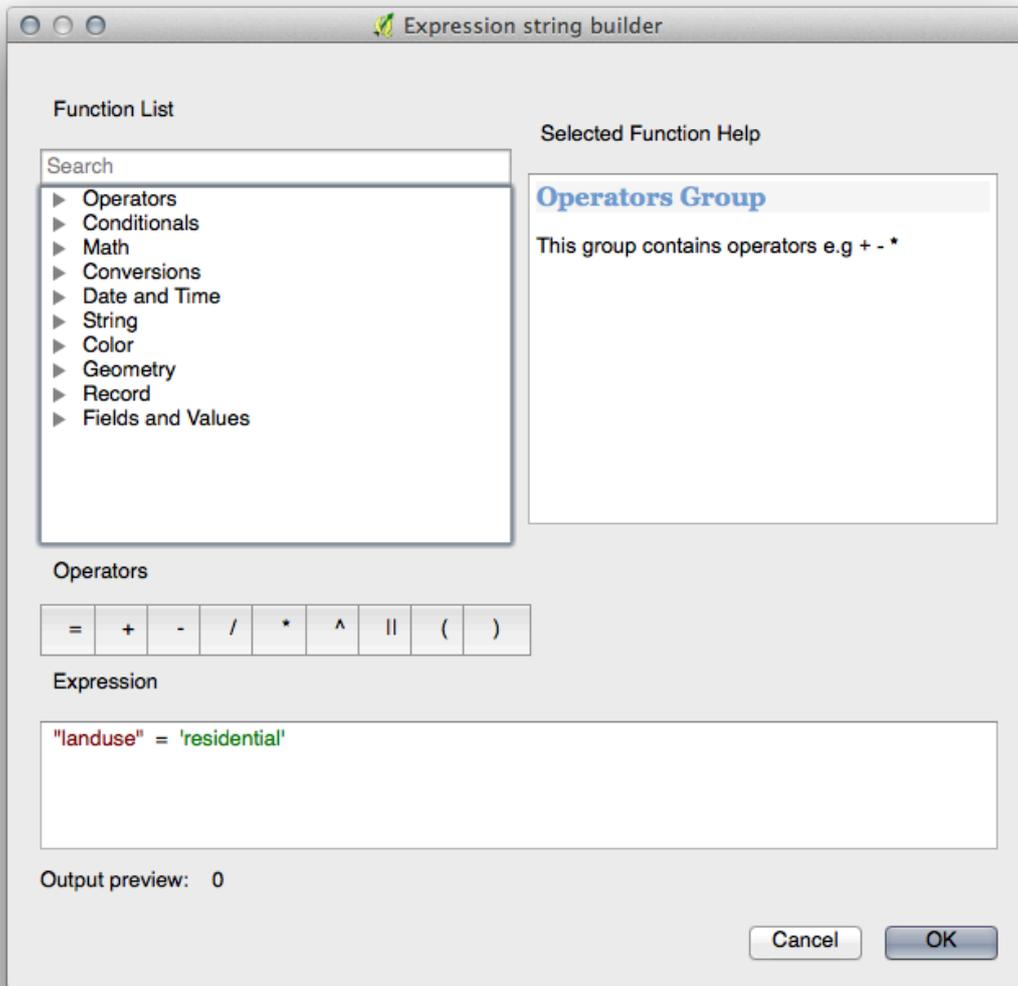
4.3.5 Follow Along: Rule-based Classification

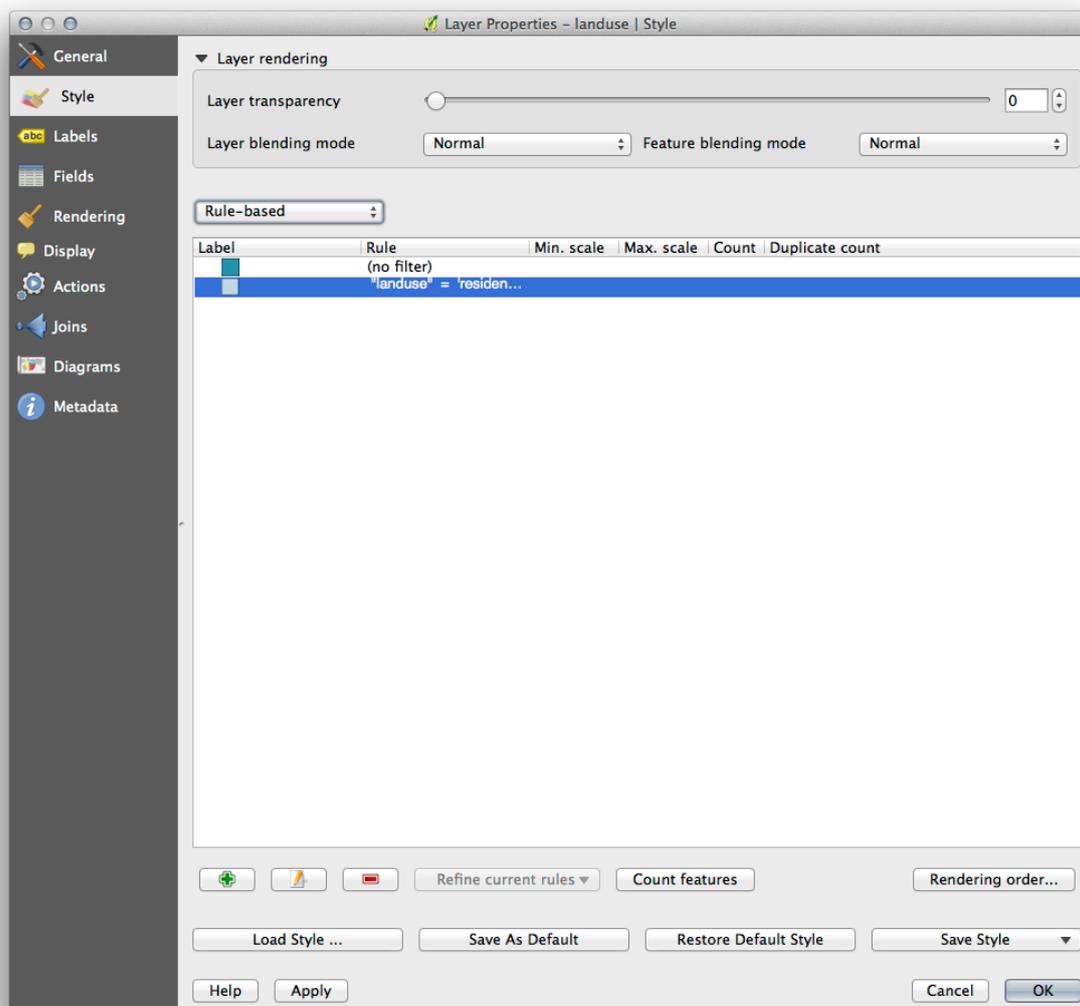
これは、分類のための複数の条件を組み合わせることで便利ですが、残念ながら通常の分類は唯一のアカウントに属性を 1 つ取ります。ルールベースの分類が便利になるところです。

- *landuse* 層のためのレイヤ *Properties* ダイアログを開きます。
- *Style* タブに切り替えます。
- ルール *based* に分類スタイルを切り替えます。これを取得します：



-  rule 追加 ボタンをクリックします。
- 新しいダイアログが表示されます。
- 省略記号をクリックします 次へ... ボタン Filter テキスト領域。
- 表示されたクエリビルダを使用して、条件を入力します KBD :! ‘「土地利用」=「住宅」AND「名」=「majorUrbanName | guilabel : ‘、‘クリック Ok’をし、そのために薄いブルーグレーを選択し、境界線を削除します。



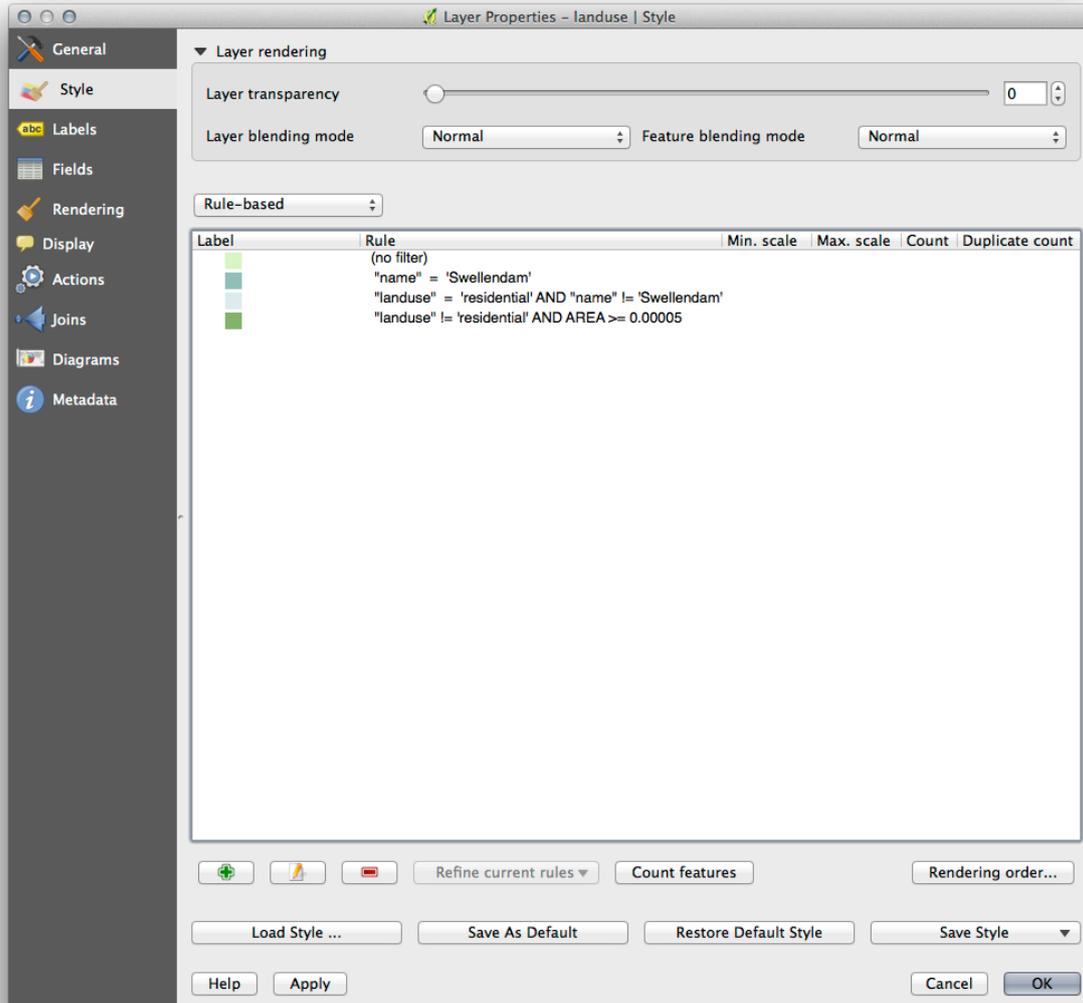


- 0.00005'> = '住宅' AND "AREA" ' "土地利用" と半ば緑の色を選択します！KBD：新しい基準を追加します。
- 別の新しい基準 :kbd:|majorUrbanName ' "名前" = ' | ' を追加、地域の町の重要性を示すためにそれを濃い灰色、青色を割り当てます。
- クリックして、リストの先頭にこの基準をドラッグします。

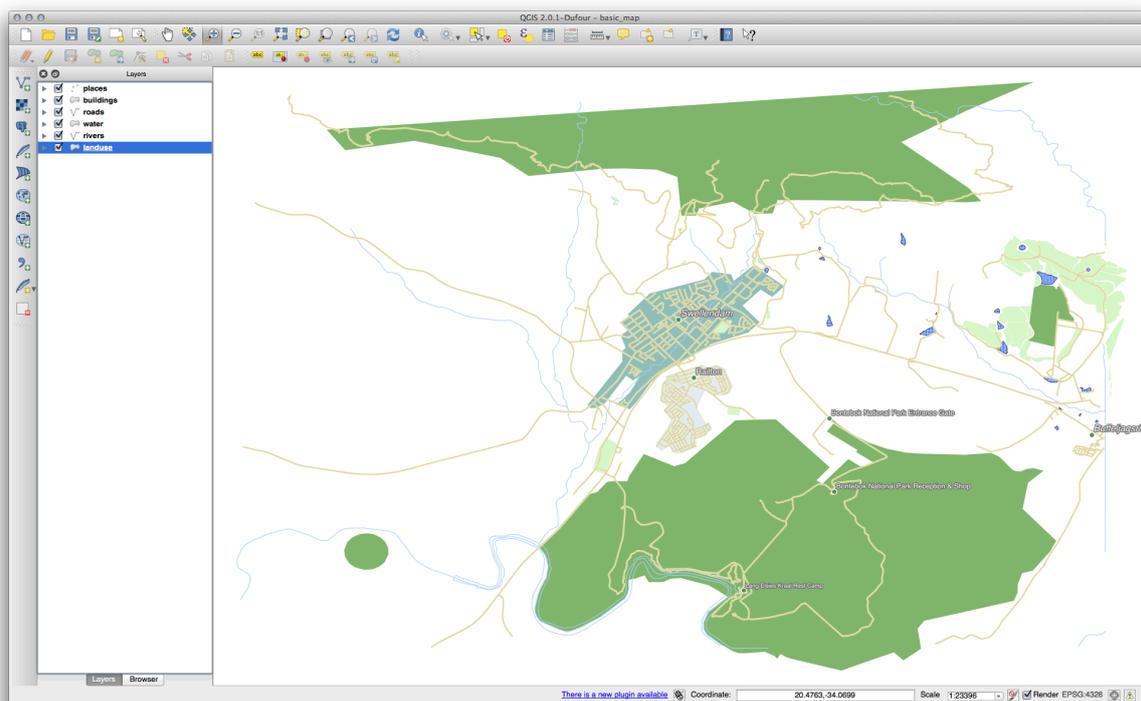
（「Swellendam」すなわち 0.00005 は、住宅ではなく、ではないこと小さいもの）これらのフィルタは、彼らは総称して地図上にいくつかの領域を除外することで、排他的です。（フィルタなし）カテゴリこれは除外ポリゴンは、デフォルトのスタイルを取ることを意味します。

マップ上の除外ポリゴンは住宅地も、そのデフォルトのカテゴリに適した淡い緑色を与えることができないことを知っています。

ダイアログは次のようになります。



- このシンボルを適用します。
マップは次のようになります。



今、最も顕著な住宅地とその大きさに応じて色の他の非住宅地 Swellendam で地図を持っています。

4.3.6 In Conclusion

シンボルは、読みやすいように、レイヤーの属性を表すことができます。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、地図読者が地物の重要性を理解することができます。直面する問題に応じて、それらを解決するために、異なる分類技術を適用します。

4.3.7 What's Next?

今、見栄えの良い地図を持っているが、どのように QGIS の外に、プリントアウト、または画像や PDF に作ることができる形式にそれを得ようとしていますか？それは、次のレッスンの話題です！

Chapter 5

Module: マップの作成

このモジュールでは、品質のよい地図を作成するため、必要なマップコンポジットとともに、QGIS のマップコンポーザの使い方について学習します。

5.1 Lesson: 地図組版を使用する

地図が得られたので、今度はそれを印刷したり、ドキュメントにエクスポートできるようにする必要があります。その理由は、GIS のマップファイルは画像ではありません。むしろ、それは地図ファイルの意志、データまたは（例えば QGIS など）同じ GIS プログラムを持っていない人のためにそう、すべてのレイヤー、そのラベル、色などを参照して、GIS プログラムの状態を保存し、役に立ちません。幸いにも、プリンタが接続されている場合は、QGIS は誰のコンピュータが読み取ることができる形式にその地図ファイルをエクスポートするだけでなく、地図をプリントアウトできます。エクスポートと印刷はどちらも地図組版を経由して処理されます。

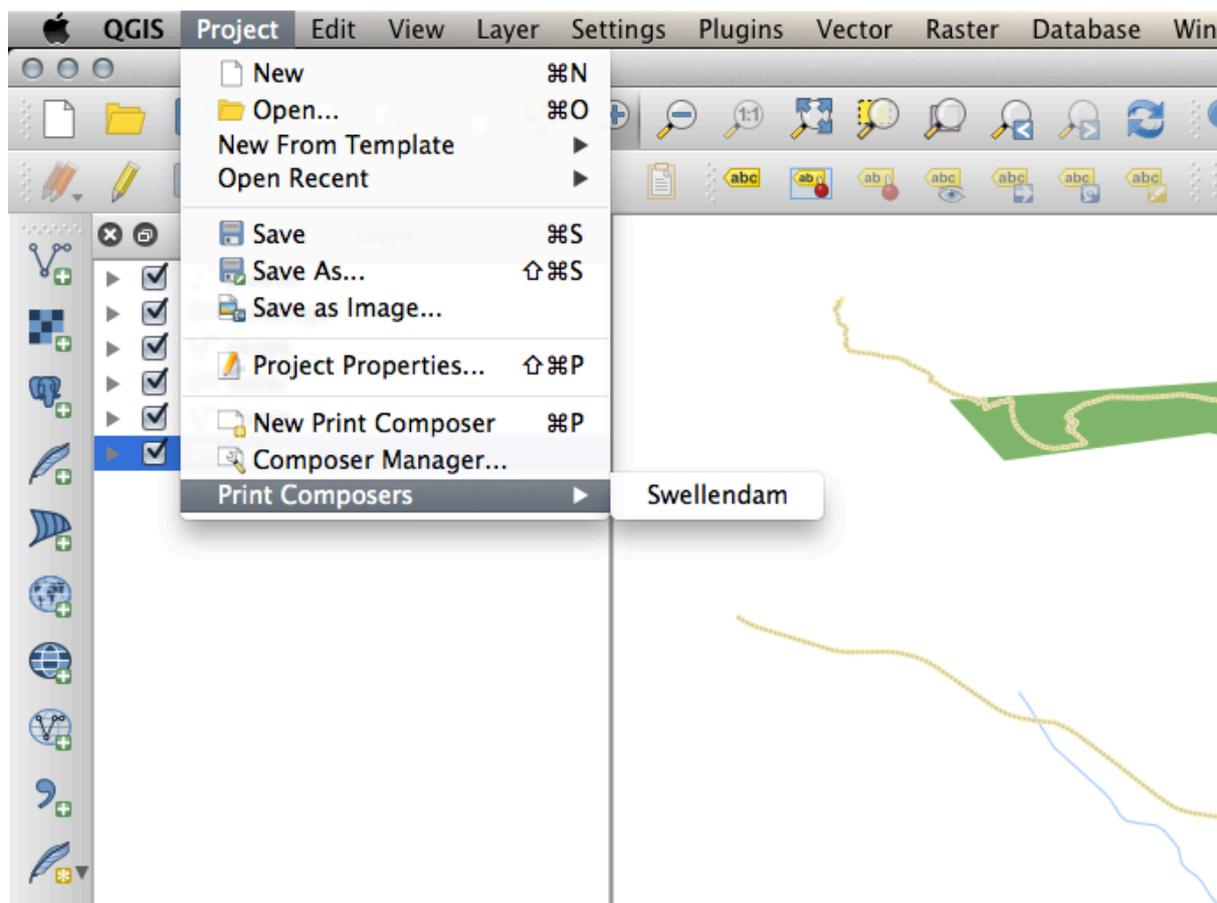
このレッスンの目標：QGIS の地図コンポーザを使用して、すべて要求されるように設定された基本的地図を作成する。

5.1.1 Follow Along: 組版マネージャー

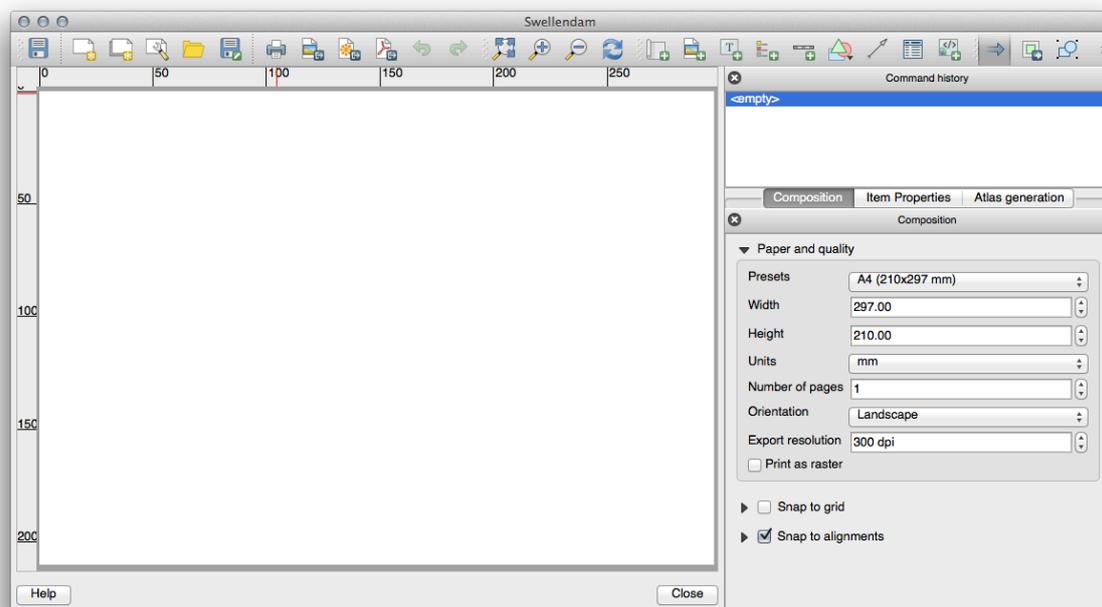
QGIS では同じ地図ファイルを使用して複数の地図を作成できます。このような理由から、組版マネージャーと呼ばれるツールがあります。

- このツールを開くにはプロジェクト->組版マネージャメニューエントリをクリックします。空の組版マネージャダイアログが表示されます。
- 追加 ボタンをクリックして、新しいコンポーザに Swellendam という名前を与えてください。
- OK をクリックします。
- 表示 ボタンをクリックします。

（ダイアログを閉じ、ファイル->印刷コンポーザメニュー経由でもコンポーザに移動できたでしょう、下の画像のように）



そこに着くためにどの道をたどっても、今印刷コンポーザ ウィンドウが表示されます：



5.1.2 Follow Along: 基本マップ組版

この例では、組版物はすでに望んだ形になっていました。同様であることを確認してください。

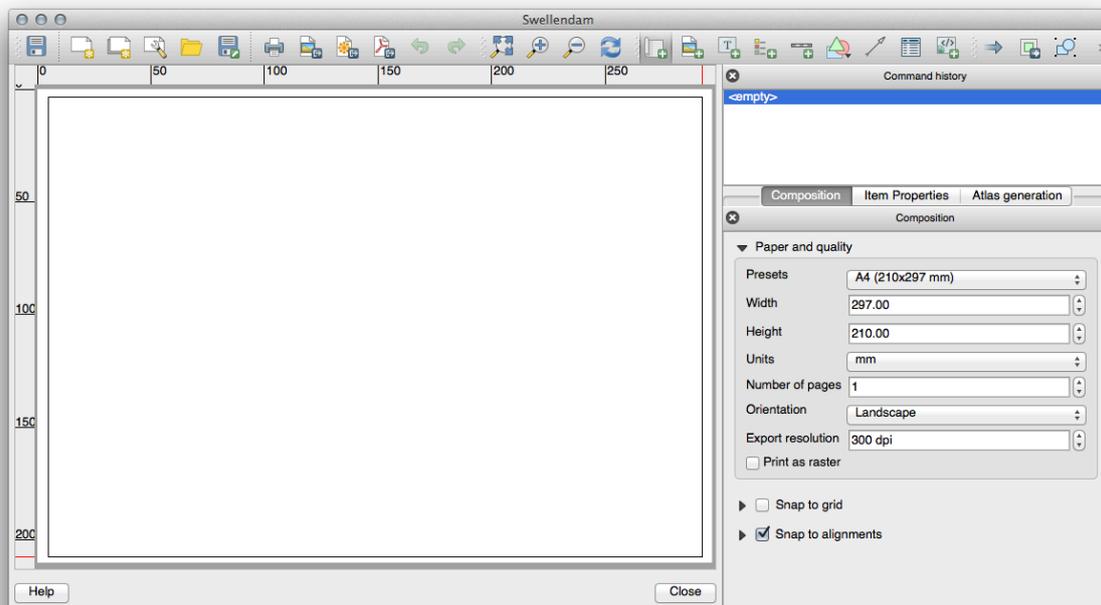
- 印刷コンポーザ ウィンドウで、組版 → 紙と品質 の下の値が次のように設定されていることをチェックしてください：
- サイズ：A4 (210×297mm)
- 向き：風景
- 品質：300dpi

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！

- 新しい地図を追加 ボタンをクリックします：

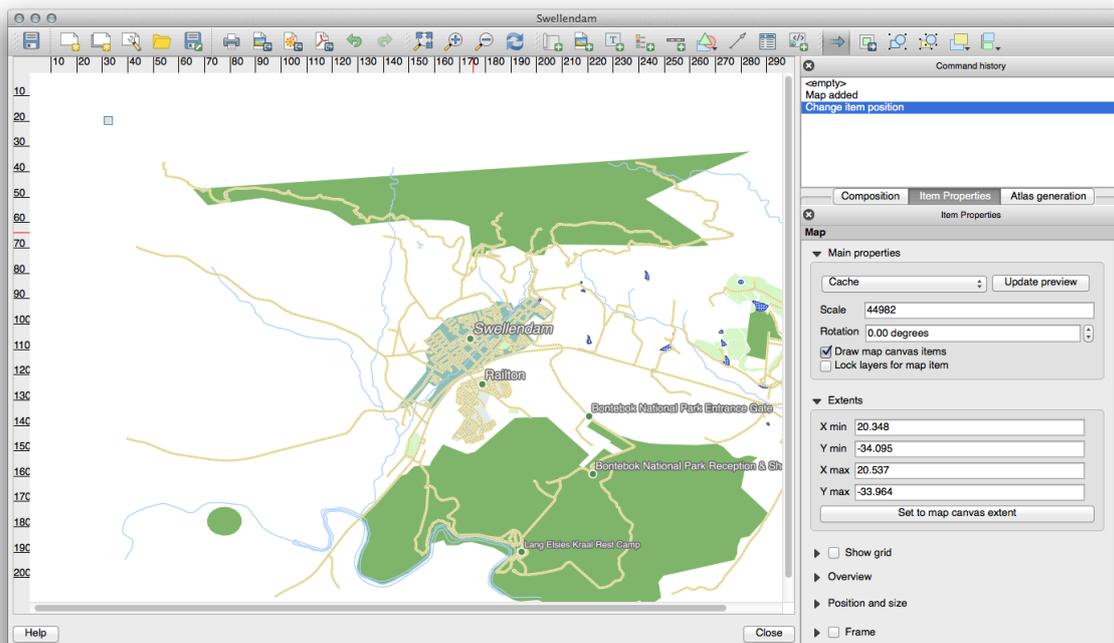
このツールをアクティブにすると、ページ上に地図を配置できるようになります。

- 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。

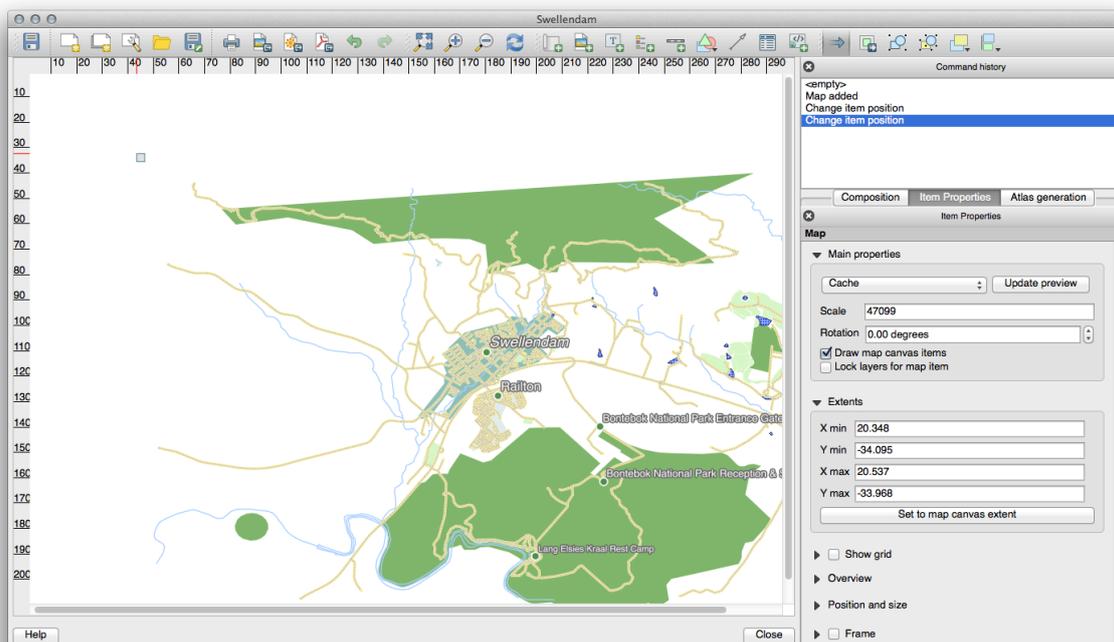


地図がページに表示されます。

- クリックしてドラッグすることで地図を移動します。



- コーナーでボックスをクリックし、ドラッグしてサイズを変更：



ノート：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もので、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。

- タイトルの上部に沿って縁に沿って余白、およびスペースを残すようにしてください。
- これらのボタンを使用してページ（マップでなく！）上でズームイン、ズームアウト：



- メイン QGIS ウィンドウでマップをズームとパン。移動項目 *content* ツール使用しても地図をパンできます：

ズームインするとき、地図ビューは、それ自体で更新されません。それはそれを配置したい場所にページをズームしている間に地図を再描画して時間を無駄にしないようにするためですが、それはまた、ズームインまたはズームアウトすると、地図が間違っただけの解像度になり醜かったり読めない、ことを意味します。

- このボタンをクリックすることで強制的に地図を再描画します：



地図を与えてくれたサイズと位置が最終である必要はないことに注意してください。満足していない場合はいつでも、後で戻って変更できます。今のところ、この地図上の仕事を保存したことを確認する必要があります。QGIS で *Composer* がメイン地図ファイルの一部なので、メインのプロジェクトを保存する必要があります。メイン QGIS ウィンドウ (レイヤー *list* と 1、前に使用していた他のすべてのおなじみの要素) に移動し、そしていつものようにそこからプロジェクトを保存します。

5.1.3 Follow Along: タイトルを追加する

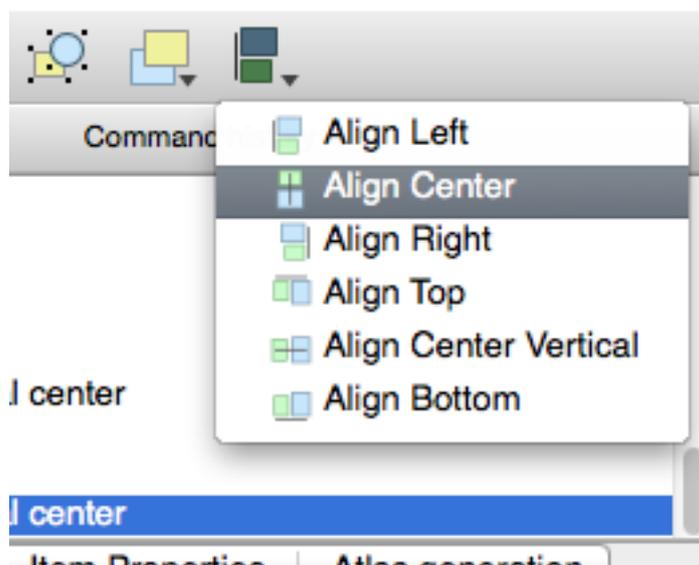
今、地図はページ上で格好良いですが、読者は/ユーザはまだ何が起きているのか言われていません。彼らは地図要素を追加することによって、彼らのために提供よ何であるいくつかのコンテキストを必要とします。まずは、タイトルを追加してみましょう。

- このボタンをクリックしてください：
- 地図上で、ページをクリックすると、ラベルが地図の上部に表示されます。
- そのサイズを変更し、ページの上中央に配置します。それは地図をサイズ変更したり移動するのと同じ方法でサイズ変更や移動できます。

タイトルを移動すると、ページの中央にタイトルを配置助けるガイドラインが現れることに気づくでしょう。しかし、地図 (ページでなく) にタイトルを位置決めするのに役立つツールもあります。



- 選択するために、地図をクリックしてください。
- 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。
- 整列 ボタン  を探し、横にあるドロップダウン矢印をクリックして位置決めオプションを表示し、中央揃え をクリックしてください：



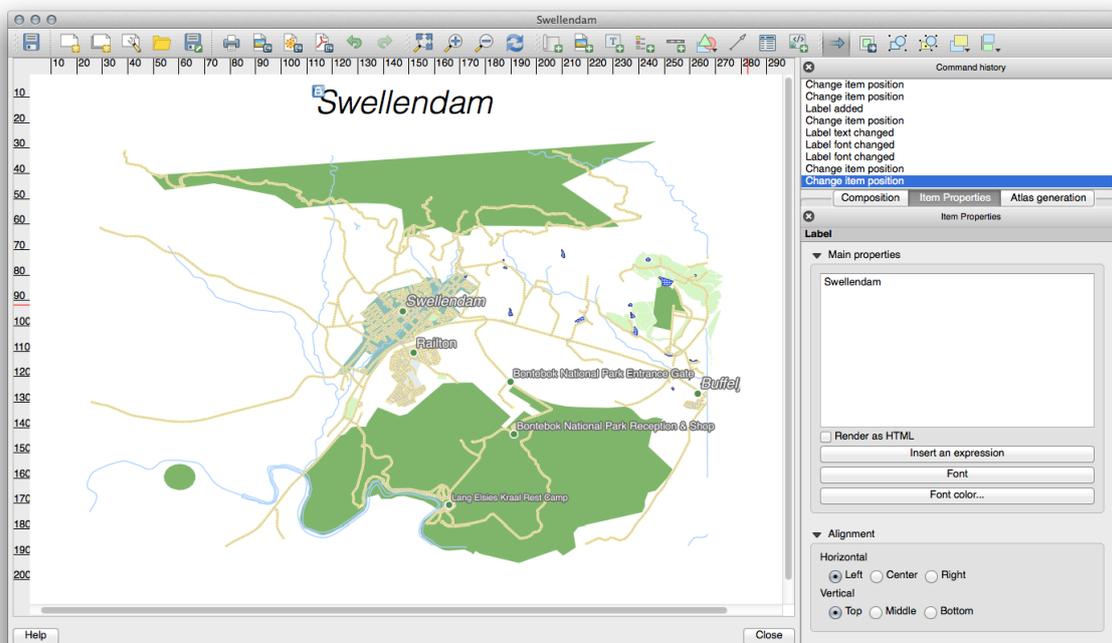
今、それらを揃えたことを中心に、これらの要素を誤って移動しないことを確認します：

- 地図とラベルの両方を右クリックします。

小さなロックアイコンが、要素が今ドラッグできないことを伝えるために隅に表示されます。けれども、いつでも要素を右クリックして再びロックを解除できます。

今、ラベルはマップではなく、内容に集中しています。ラベルの内容を中央に：

- それをクリックして、ラベルを選択します。
- *Composer* ウィンドウのサイドパネルの項目 *Properties* タブをクリックします。
- レベルの文字を”Swellendam“に変更します：
- フォントと整列オプションを設定するには、このインターフェイスを使用します。



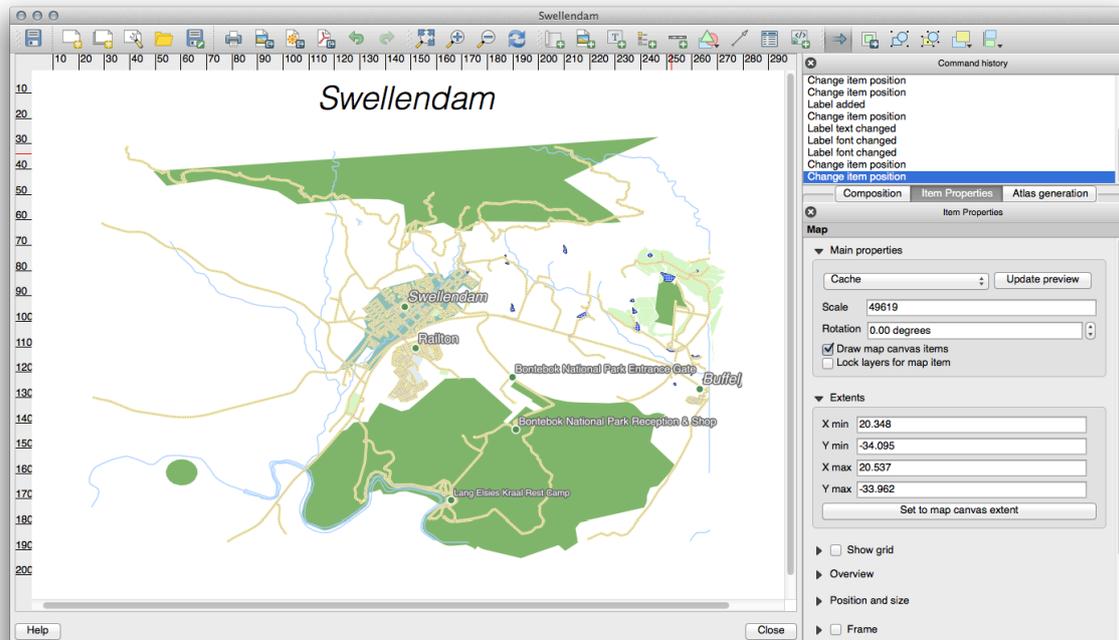
- 大きい実用的なフォントを選択し（例では大きさ 36 のデフォルトのフォントを使用します）、水平位置揃えを中央に設定します。

フォントの色も変更できますが、デフォルトごとにそれを黒のままにしておくのがおそらく最善です。

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。

- 項目 *Properties* タブで、*Frame* オプションが表示されるまで、スクロールダウンします。
- *Frame* 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。

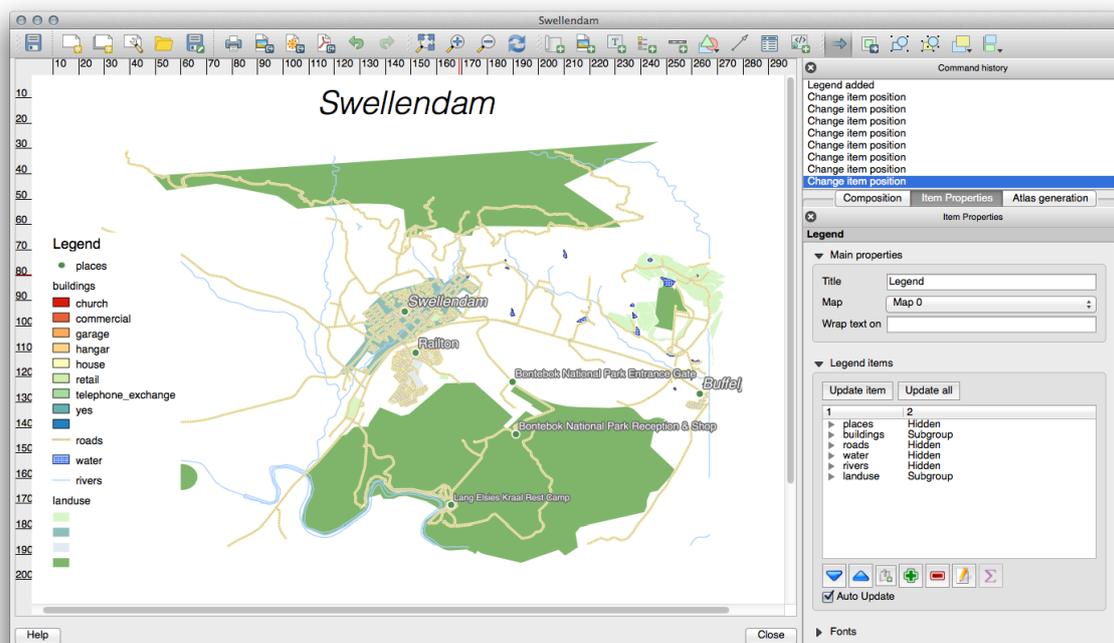
この例では、フレームを有効にしていますので、ここで私たちのページは、これまでのところです。



5.1.4 Follow Along: 凡例の追加

地図の読者には、実際に意味の地図上のさまざまな何の事見ることができるようする必要があります。いくつかのケースでは、地名のように、これは非常に明白です。他の例では、農場の色のように、推測することはより困難です。新しい凡例を追加してみましょう。

- このボタンをクリックしてください: 
- 凡例を配置するページをクリックし、希望の場所に移動します:



5.1.5 Follow Along: 凡例項目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。

- 項目 *Properties* タブで、凡例項目 パネルを見つけることができます。
- *buildings* エントリを選択します。

• *minus* ボタン:  をクリックすると、凡例からそれを削除します

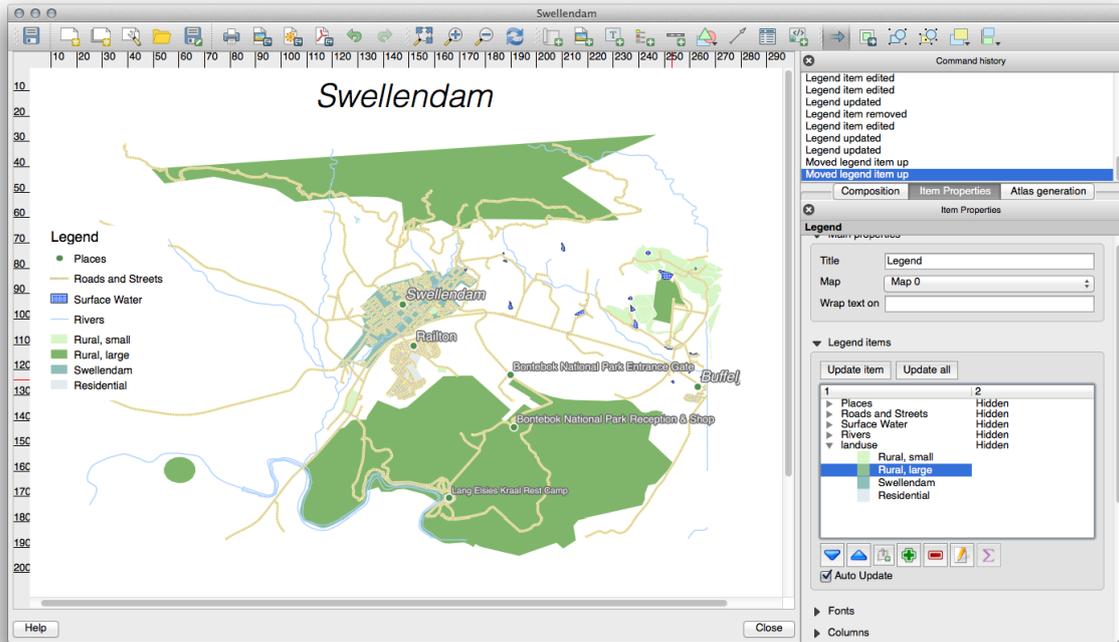
また、アイテムの名前を変更できます。

- 同じリストからレイヤーを選択します。

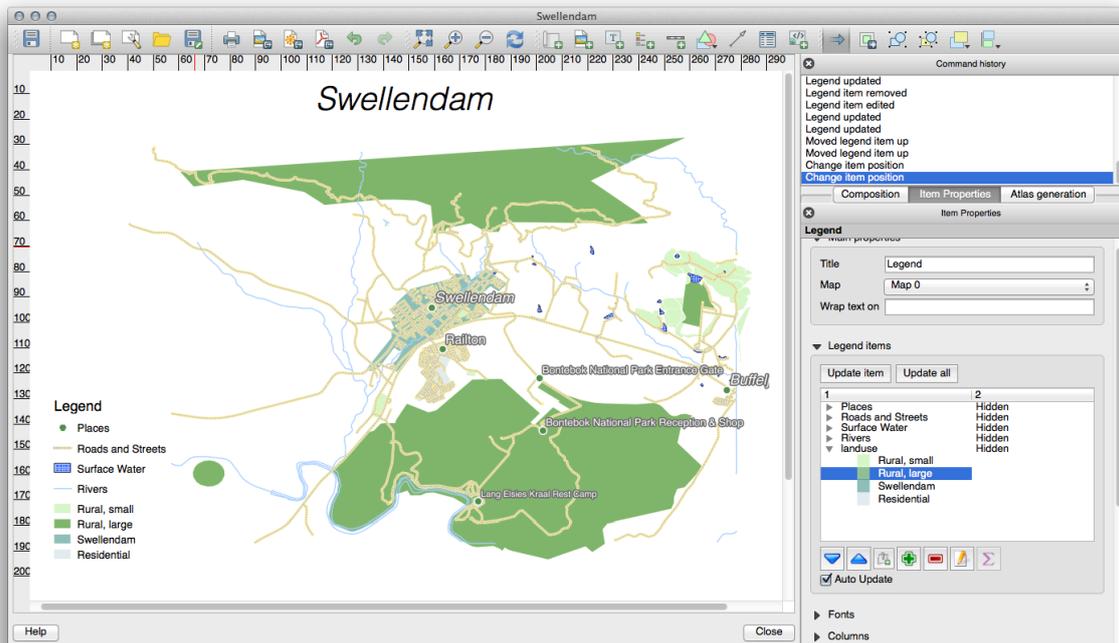
• *Edit* ボタンをクリックする: 

- レイヤーの名前を変更します *Places*、*道路および街路*、*Surface Water*、および *河川*。

- *landuse* を *Hidden* に設定し、その後、下向きの矢印をクリックして、凡例で各カテゴリを編集してそれらに名前を付けます。また、アイテムを並べ替えできます。



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サイズを変更したい場合があります。これが結果です。



5.1.6 Follow Along: 地図を書き出す

ノート: 多くの場合、作業を保存するために覚えていましたか？

最後に、地図は書き出す準備ができています！コンポーザウィンドウの左上隅の近くには書き出しボタンが

表示されます：



Print ボタン、プリンタとインターフェイス左側のボタンです。プリンタオプションは、作業しているプリンタの機種によって異なりますので、プリンタのマニュアルまたはこのトピックの詳細については、おそらく印刷の一般的なガイドに相談するのが良いでしょう。

他の 3 つのボタンがファイルに地図のページを書き出しできます。選択できる書出形式は 3 つあります：

- *Image* としてエクスポート
- *SVG* としてエクスポート
- *PDF* としてエクスポート

画像としてエクスポートすると、その中から選択できる様々な一般的な画像フォーマットの選択を提供します。これはおそらく最も簡単なオプションですが、それが作成した画像は「死んで」おり、編集することは困難です。

他の 2 つのオプションは、より一般的です。

地図製作者（公開のための地図を編集したいかもしれません）に地図を送信する場合、それは SVG としてエクスポートすることが最善です。SVG は、「スケーラブルベクターグラフィック」の略で、**Inkscape** の <https://inkscape.org/> または他のベクター画像編集ソフトなどのプログラムにインポートできます。

クライアントにマップを送信する必要がある場合、それは PDF の印刷オプションを設定する方が簡単ですので、それは、PDF を使用するのが最も一般的です。彼らは彼らがこのフォーマットをインポートして編集することができますプログラムを持っている場合、一部の地図製作者は、同様に PDF を好むかもしれません。

我々の目的のために、私たちは PDF を使用するつもりです。

- *PDF* としてエクスポート ボタンをクリックしてください： 
- いつものように、保存場所とファイル名を選択します。
- Click *Save*.

5.1.7 In Conclusion

- 組版 ウィンドウ閉じます。
- 地図を保存します。
- お使いのオペレーティングシステムのファイルマネージャを使用して、書き出した PDF ファイルを検索します。
- 開いてください。
- その栄光に浴します。

初めての QGIS 地図プロジェクトの完成おめでとうございます！

5.1.8 What's Next?

次のページでは、完成する課題を与られます。これによって、これまでに学んだテクニックを練習できます。

5.2 課題 1

あなたの既存のマップ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な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上で、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  基本または  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、自分の地図で行いたいものがあったら、実装しようとしてみてください。

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は、評価のために、地図を PDF にエクスポート、の最終版を提出する必要があります。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、独自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！

5.2.1 In Conclusion

最初の 4 つのモジュールでは、ベクトルマップを作成し、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、ベクトルデータを作成および編集する；ベクトルデータを分析する；ラスターデータを使用および分析する；ラスターとベクターの両方のデータ・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。

Chapter 6

Module: ベクタデータの作成

既存のデータを使用してマップ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。

6.1 Lesson: Creating a New Vector Dataset

使用するデータはどこ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：新しいベクトルデータセットを作成します。

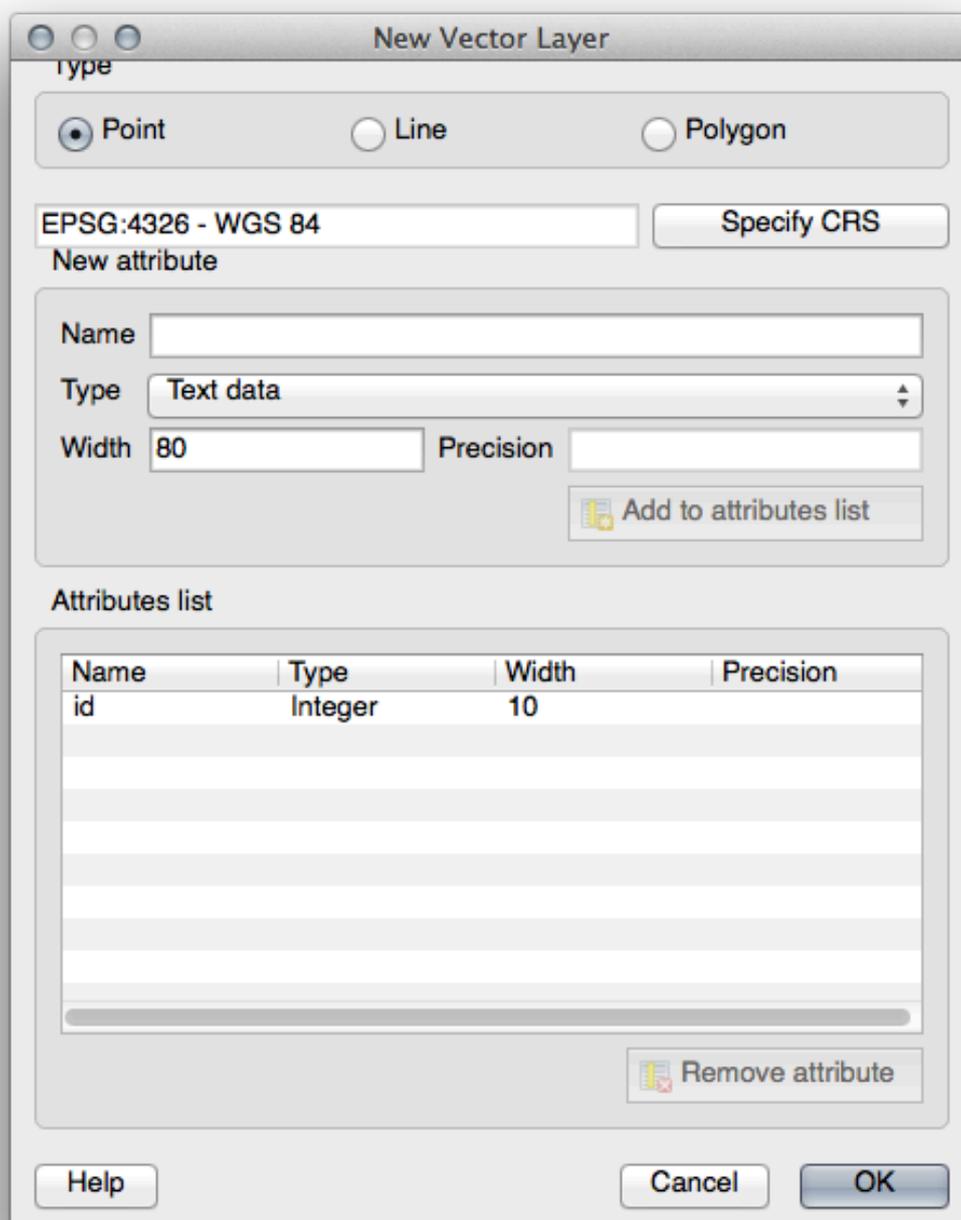
6.1.1 Follow Along: The Layer Creation Dialog

新しいベクトル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ー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。

新しいレイヤーを定義できる 新しいベクターレイヤー ダイアログを開く必要があります。

- 移動し、メニュー項目 レイヤー ->新規 ->新規シェープファイル *Layer* をクリックしてください。

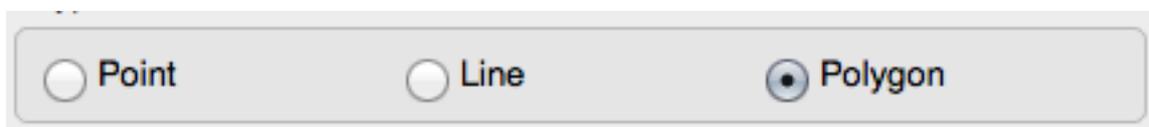
次のダイアログが表示されます：



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トルレイヤータイプは、バックグラウンドで「別々に構築」されているので、一度レイヤーを作成したらそのタイプは変更できません。

次の演習では、領域を記述する新しい地物を作成する予定です。このような地物のためには、ポリゴンデータセットを作成する必要があるでしょう。

- *Polygon* ラジオボタンをクリックしてください：



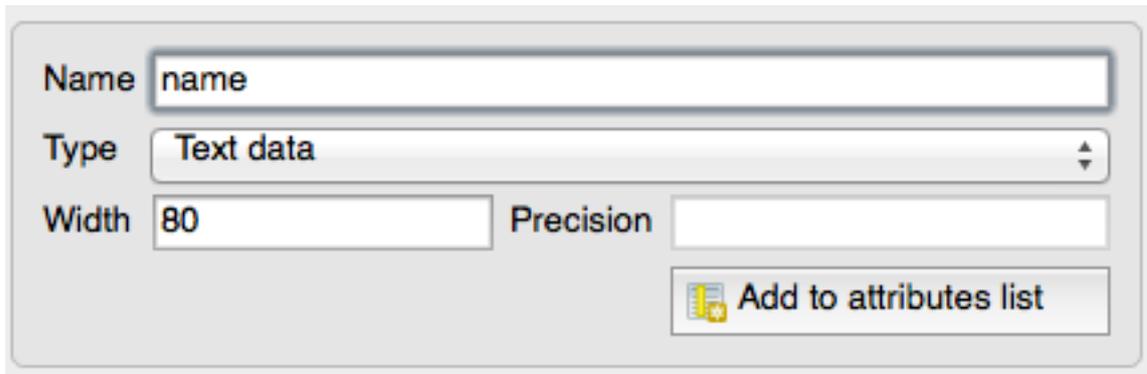
これは、ダイアログの残りの部分には影響しませんが、それは、ベクターデータセットが作成されたときにジオメトリの正しいタイプが使用されるようになります。

次のフィールドを使用すると、座標参照系、または CRS を指定できます。CRS は座標によって地球上の点を記述する方法を指定します。これを行うにはさまざまな方法があるので、多くの異なる CRS があります。このプロジェクトの CRS は WGS84 であるので、デフォルトですでに正しいです。

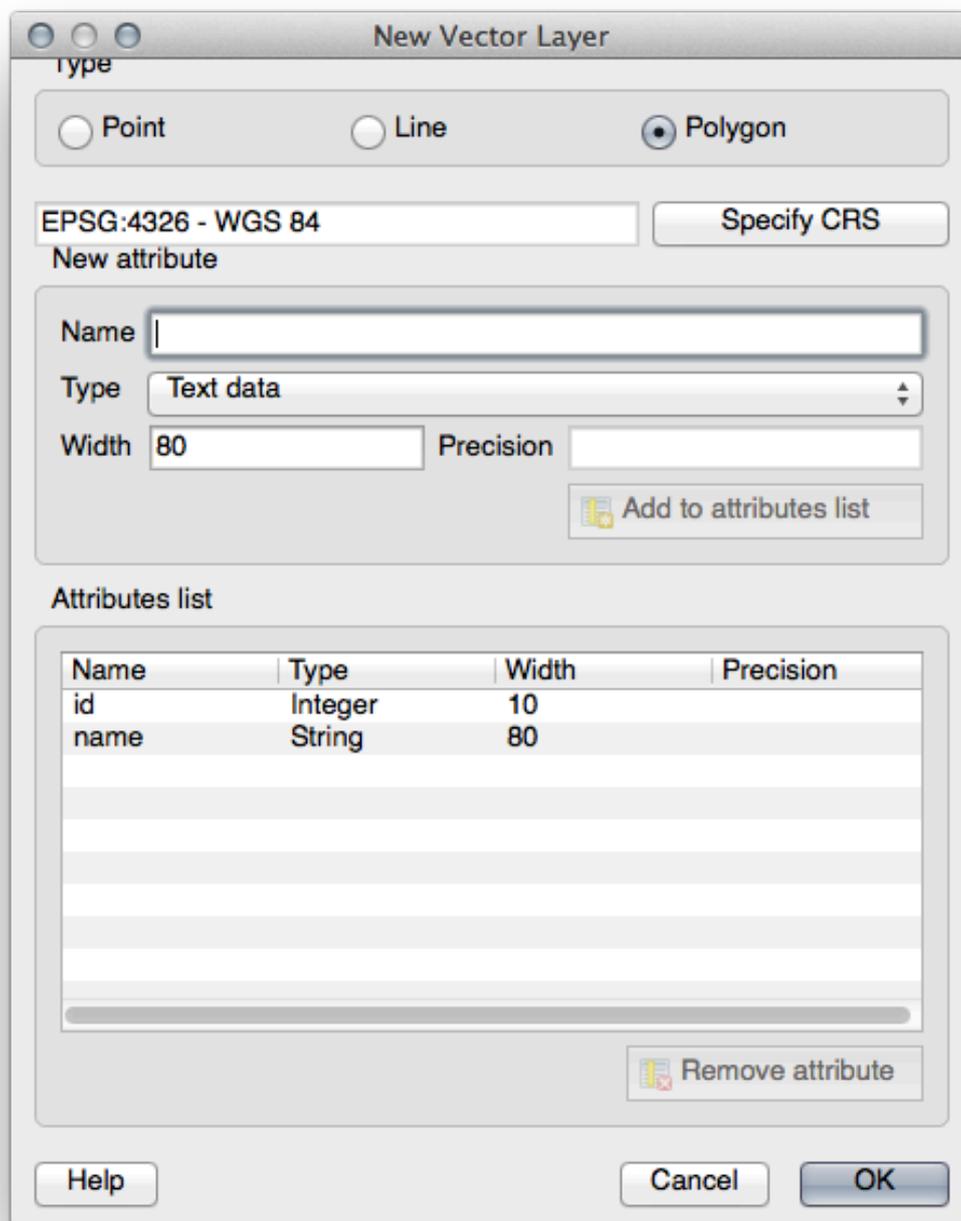


次に 新規属性 の下にグループ化されたフィールドのコレクションがあります。デフォルトでは、新規レイヤーに属性は1つ、以下の `id` フィールドだけです（属性リストが表示されるはずですが）。しかし、作成したデータが有用であるためには、この新しいレイヤーに作成している地物について実際に何かしらの言う必要があります。現在の目的のためには、`name` という名前のフィールドを1つ追加すれば十分でしょう。

- 以下の設定を複製し、属性リストに追加 ボタンをクリックします。



- ダイアログが次のようになることを確認します。



- *OK* をクリックします。保存ダイアログが表示されます。
- `exercise_data` ディレクトリに移動します。
- `school_property.shp` として、新しいレイヤーを保存します。

新しいレイヤーがレイヤーリストの中に表示されるようになります。

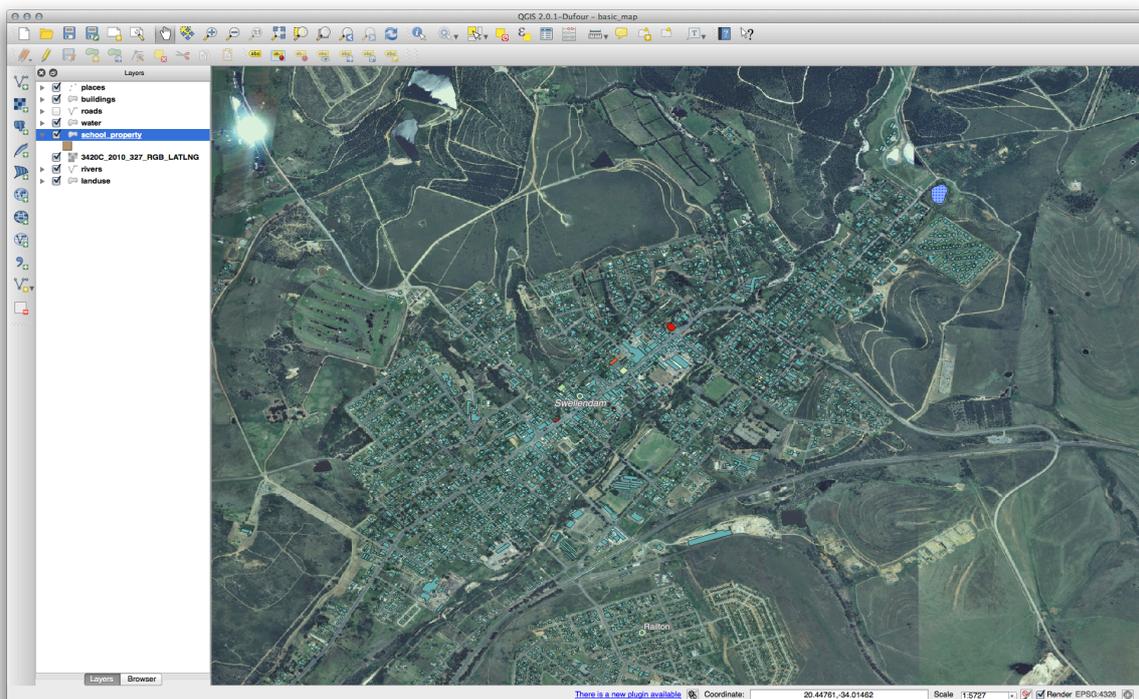
6.1.2 Follow Along: Data Sources

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、デジタル化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。

現在の例では、デジタル化のアプローチを使用しているでしょう。必要に応じてそれらをインポートする必要がありますので、サンプルラスタデータセットが提供されます。

- **Add Raster Layer** ボタンをクリックします: 
- `exercise_data/raster/` に移動します。
- ファイル `3420C_2010_327_RGB_LATLNG.tif` を選択します。
- **開く** をクリックします。画像が地図にロードされます。
- *gui*label: ‘レイヤー list’ で新しいイメージを探します。
- まだ他のレイヤーが見えるように、それをクリックしてリストの一番下にドラッグします。
- 検索して、この領域にズーム :



ノート: 建物 レイヤシンボルは、一部またはラスタレイヤーのすべてをカバーしている場合、レイヤーパネルの中で、一時的にそれを選択解除してレイヤーを無効にできます。道路 シンボルも邪魔だと思えば非表示にしておかまいません。

これらの 3 つのフィールドがデジタル化されます :



デジタル化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、一般的に誤って重要なデータを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 レイヤーに対する編集モードに入ります。

- レイヤー *list* 中のレイヤーをクリックして選択します。(そうしないと、間違ったレイヤーを編集します、正しいレイヤが選択されていることを非常に確認してください！)
- 編集を切り替え ボタンをクリックしてください： 

このボタンを見つけることができない場合は、*Digitizing* 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 → デジタル化 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。

編集モードに入ったらすぐ、デジタル化ツールが有効になっているのがわかるでしょう：



他の 4 つの関連するボタンはまだ有効でないですが、新しいデータとの対話を開始すると有効になります。



ツールバーの左から右に、彼らは以下のとおりです。

- 編集を保存 : レイヤーに加えられた変更を保存します。

- 地物を追加 : 新しい地物のデジタル化を開始します。
- 地物を移動 (複数可) : 地物全体をあたりに移動します。
- ノードツール : 地物の一部のみを移動します。
- 選択地物を削除 : 選択された地物を削除します。
- 地物切り取り : 選択した地物を切りとります。
- 地物をコピー : 選択した地物をコピーします。
- 地物を貼り付け : カットまたはコピーされた地物を再び地図に貼り付けます。

新しい機能を追加したい。

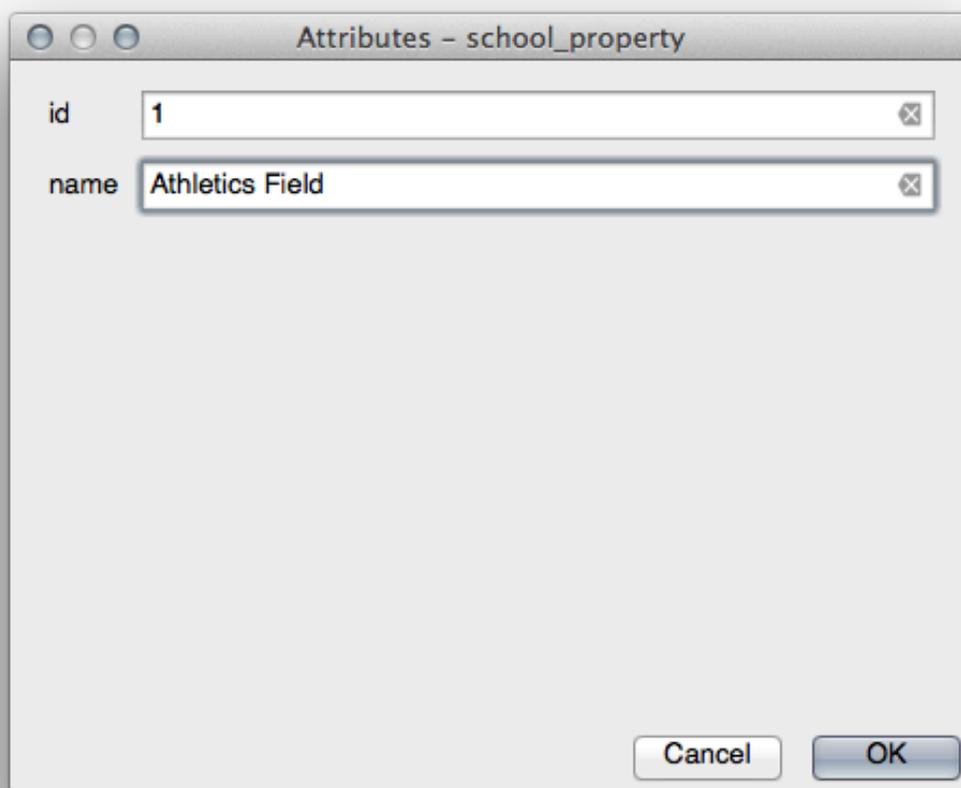
- 今 地物を追加 ボタンをクリックし、学校のフィールドのデジタル化を開始します。

マウスカーソルが十字になってきていることがわかります。これにより、デジタル化しているポイントをより正確に配置できます。デジタル化ツールを使用しているも、マウスホイールを回転させることにより、地図上でズームイン、ズームアウトでき、マウスホイールを押しながらマップの中をドラッグして周りにパンできることを覚えておいてください。

デジタル化している第 1 の地物は athletics field です :



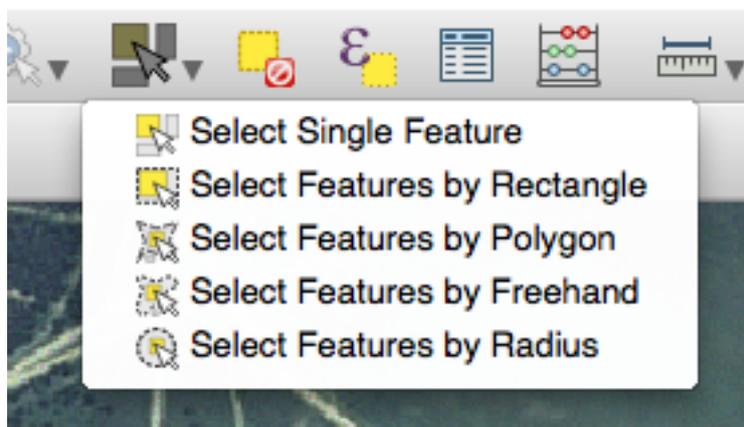
- どこかのフィールドの縁に沿ってポイントをクリックすることで、デジタル化を開始します。
- 描画しているシェイプがフィールドを完全にカバーするまで、縁に沿ってさらにクリックすることにより、より多くのポイントを置きます。
- 最後のポイントを配置した後、右クリックしてポリゴンの描画を終了します。これにより地物が確定し、属性 ダイアログが表示されます。
- 値を以下のように埋めます:



- *OK* をクリックすると新しい地物が作成されました！

地物をデジタル化しているときにミスをしてしまった場合、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。ミスをしてしまった場合は、上記のような地物を作成し終わるまで、デジタル化を継続します。その後：

- 単一の地物を選択 ツールで地物を選択します。



使用することができます。

- 地物を移動（複数可）ツールは、地物全体を移動します
- ノード *Tool* でミスクリックをした点だけを移動し

- *Selected* を削除 でもう一度試すことができるよう完全に地物を取り除き、
- 編集->Undo メニュー項目や CTRL+z キーボードショートカットは、ミスをやりに戻します。

6.1.3 Try Yourself

- 学校自体と上のフィールドをデジタル化します。デジタル化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がユニークな id 値を持っている必要があることを忘れないでください！

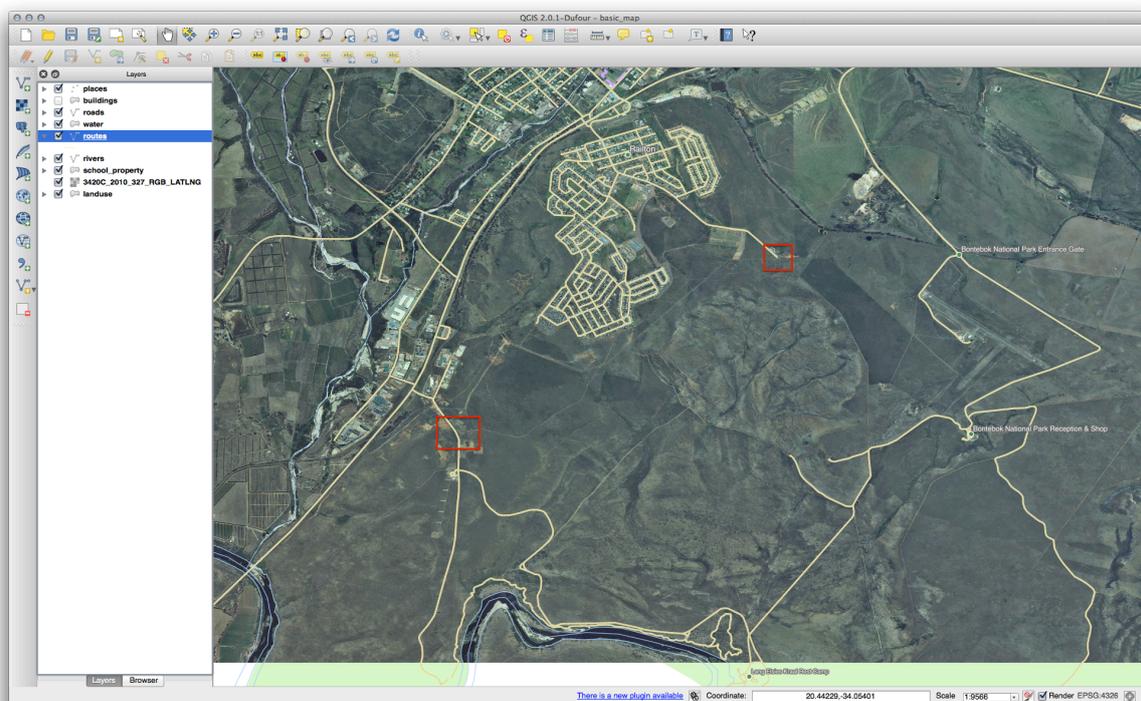
ノート：レイヤーに地物を追加し終わったら、編集内容を保存して、編集モードを終了することを忘れないでください。

ノート：以前のレッスンで学んだ技術を使用して、*school_property* の塗りつぶし、外形線とラベル配置や書式のスタイルを設定できます。この例では、塗りなしで薄紫色の点線の外形線を使用します。

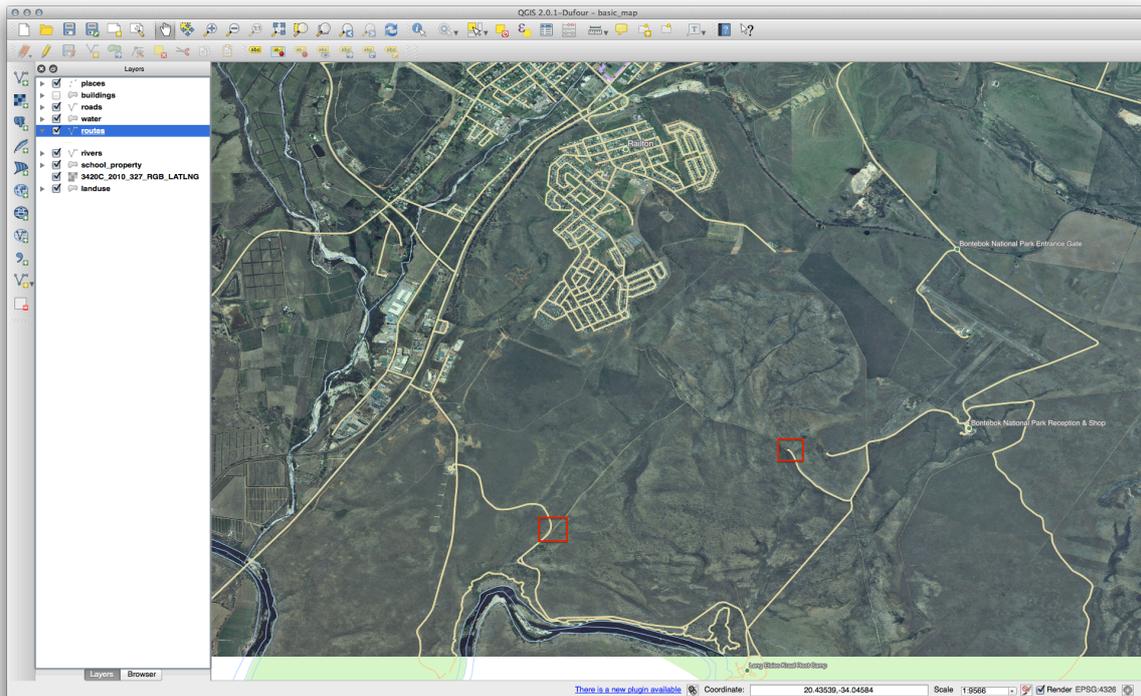
6.1.4 Try Yourself

- 属性 `id` および `type` を持つ `:kbd: 'routes.shp'` と呼ばれる新しいライン地物を作成します。(手がかりに上記のアプローチを使用してください。)
- すでに道路のレイヤーの上にマークされていない2つのルートを実行していきま。一つはパスで、もう一つはトラックです。

パスがマークされた道路から開始と終了、Railton の郊外の南の端に沿って実行されます。



トラックはさらにもう少し南へ:



一度に一つ、上のパスとトラックをデジタル化 *routes* レイヤー。任意のコーナーまたはターンでの点（左クリック）を使用して、できるだけ正確に経路に従うよう試みます。

各ルートを作成するとき、それらに *PATH* か *track* の属性値 *type* を与えます。

おそらくポイントだけがマークされていることがわかります。レイヤプロパティ ダイアログを使用して、ルートにスタイルを追加します。パスとトラックに異なるスタイルを与えること自由に感じなさい。

編集内容を保存し、*Edit* モードを切り替えます。

結果を確認します

6.1.5 In Conclusion

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。

デジタル化は 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。

6.1.6 What's Next?

GIS レイヤーの地物は、ただの絵ではなく、空間中のオブジェクトです。例えば、隣接するポリゴンは、彼らがお互いに関連している場所を知っています。これは、トポロジーと呼ばれます。次のレッスンでは、これが役立つことができる理由の例が表示されます。

6.2 Lesson: 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ーの有用な側面です。

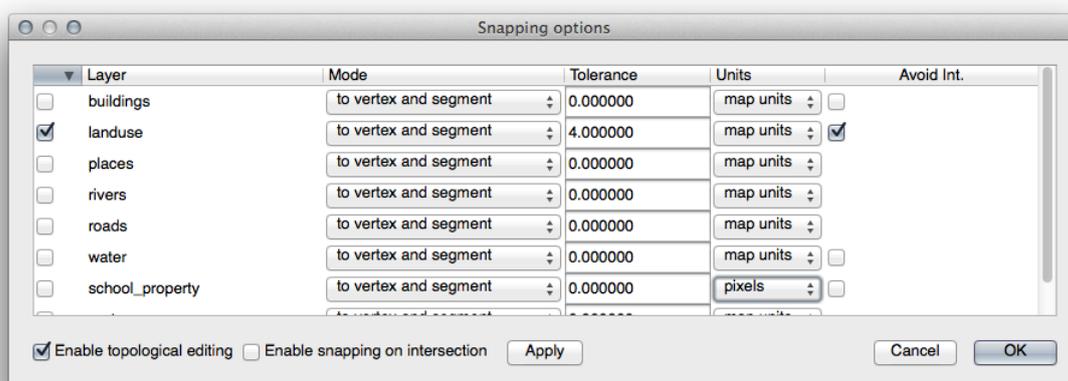
たとえば、2つの地物が境界線を共有し、あなたがトポロジを使用して境界線を編集する場合、はじめの地物を編集し、そしてもう1つの地物を編集して、それらが合うように慎重に境界線を描く必要はありません。あなたはそれらの共有された境界線を編集することができ、両方の地物は同時に変化します。

このレッスンの目標: 例を用いてトポロジを理解します。

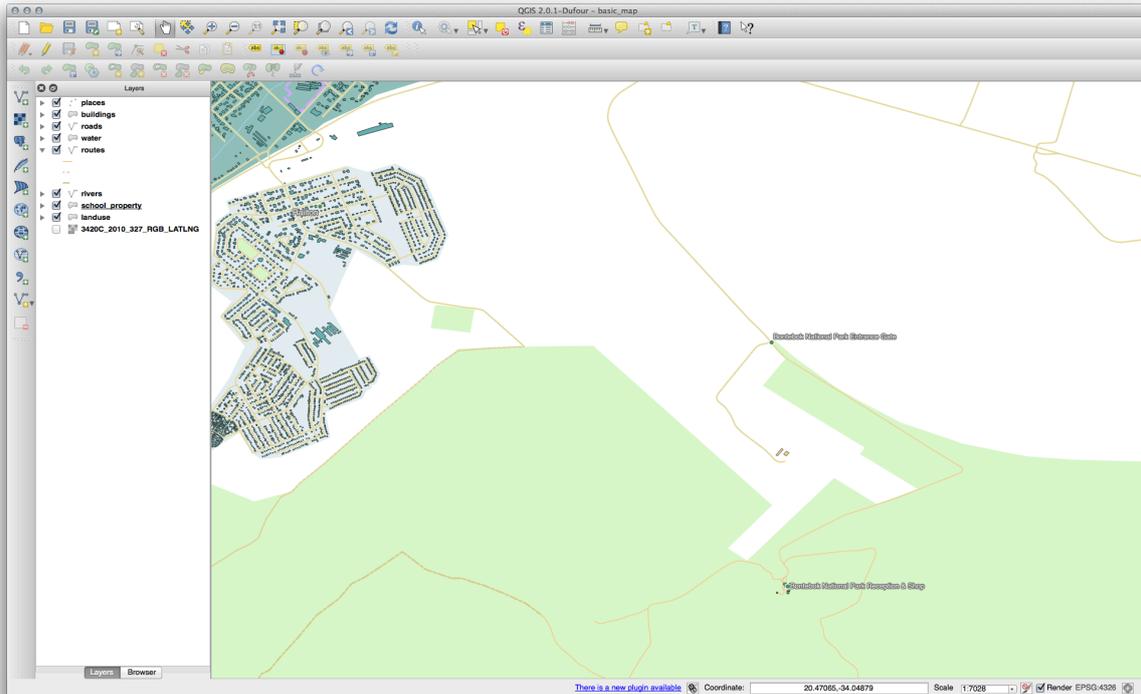
6.2.1 Follow Along: スナップ

トポロジ編集を簡単に行うにはスナップを有効にするのがベストです。これによってデジタイズ中にマウスカーソルが他のオブジェクトにスナップします。スナップオプションを設定するには:

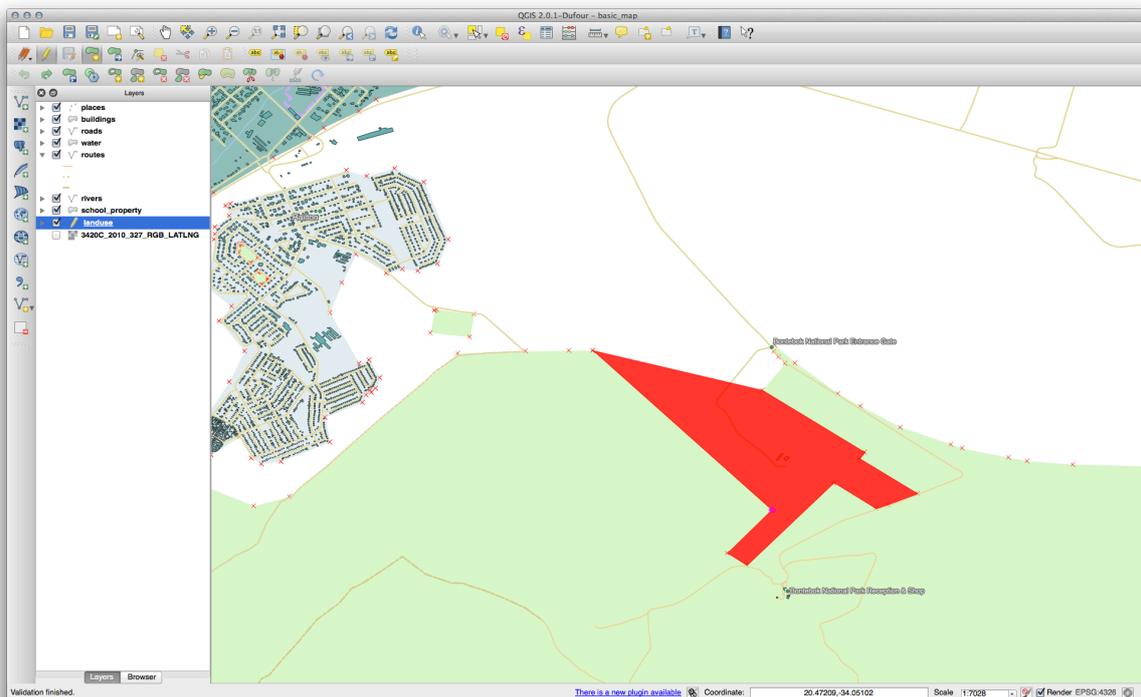
- メニューエントリ 設定 → スナップオプション... に移動します。
- 次のように スナップオプション ダイアログを設定します:



- 交差禁止 列のボックスがチェックされていることを確認します。
- OK をクリックして変更を保存し、ダイアログボックスを離れます。
- landuse レイヤを選択して編集モードにします。
- ビュー → ツールバー で 先進的なデジタイズ ツールバー が有効になっていることを確認します。
- 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):



- Bontebok National Park のこの新しい (架空の) エリアをデジタイズします:



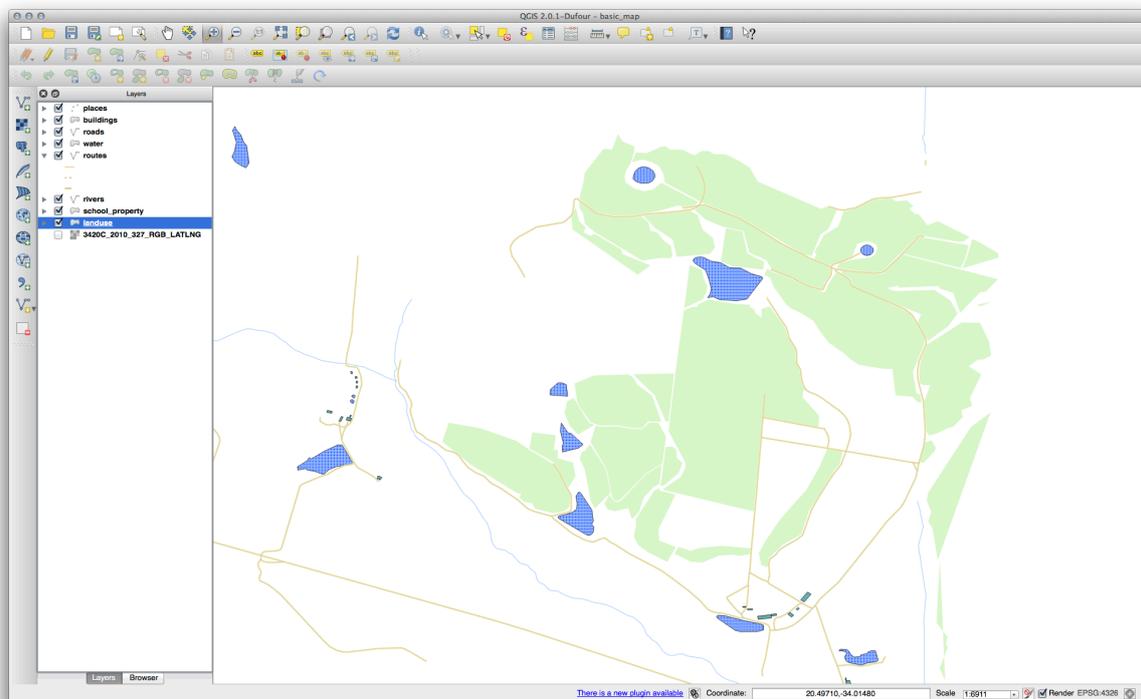
• プロンプトが表示されたら *OGC_FID* に 999 を入力します。他の値はそのままにしておいていいです。慎重にデジタイズして、カーソルを隣接する農場の頂点にスナップさせていれば、新しい農場と隣接する既存の農場の間にギャップがないことに気付くでしょう。

- 先進的なデジタイズ ツールバーの元に戻す/やり直しツールに注目して下さい:

やり直し | 元に戻します

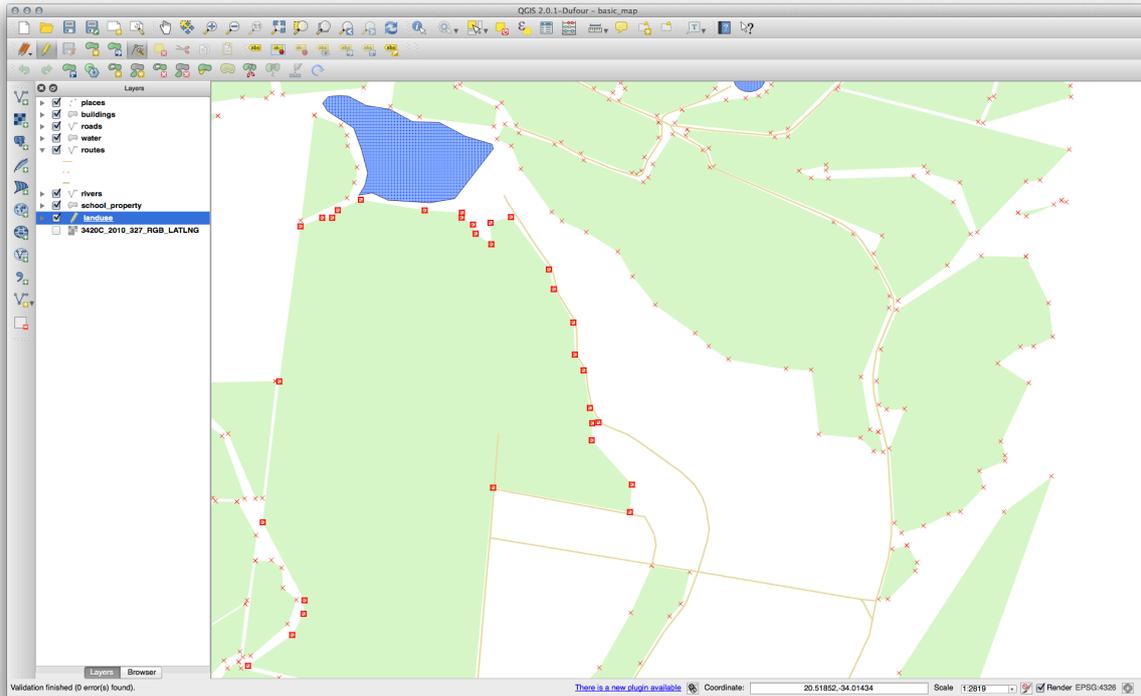
6.2.2 Follow Along: トポロジカル地物の修正

トポロジ地物は時々更新しなければならないことがあります。この例では、*landuse* レイヤは最近 1 つのエリアに結合された複雑な森林エリアをいくつか含みます:

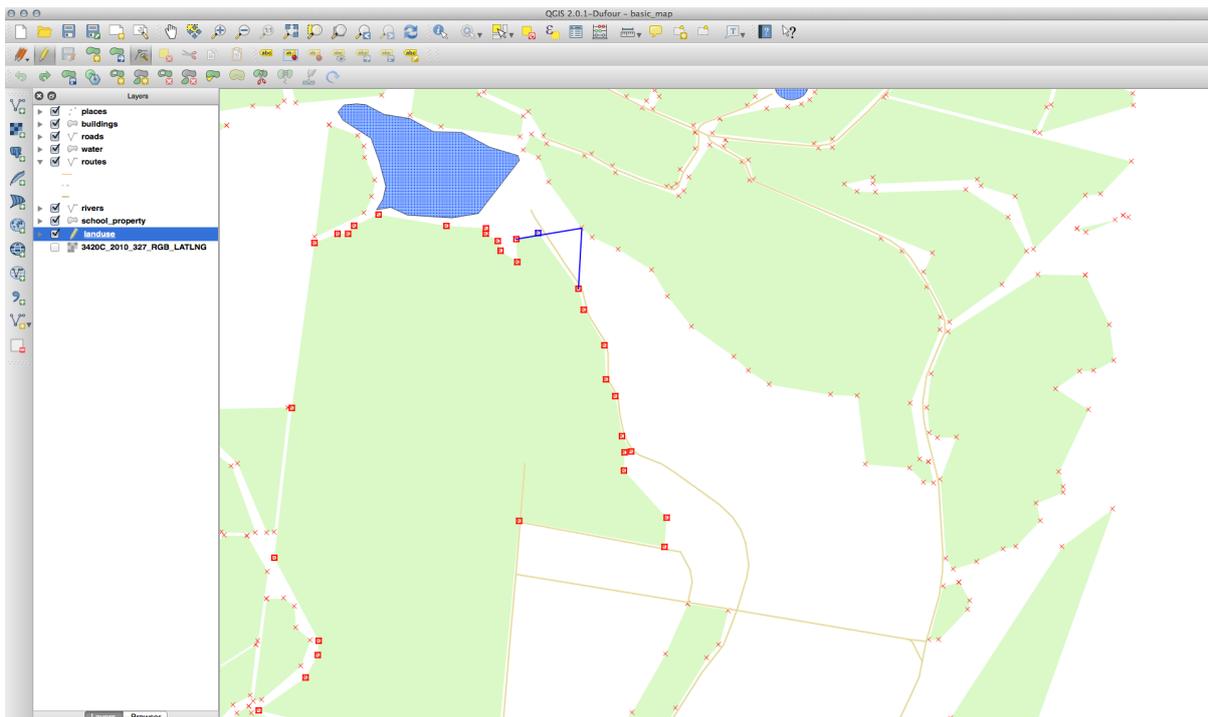


新しいポリゴンを作って森林エリアを結合するのではなく、ノードツールを使って既存のポリゴンを編集し、それらを結合しましょう。

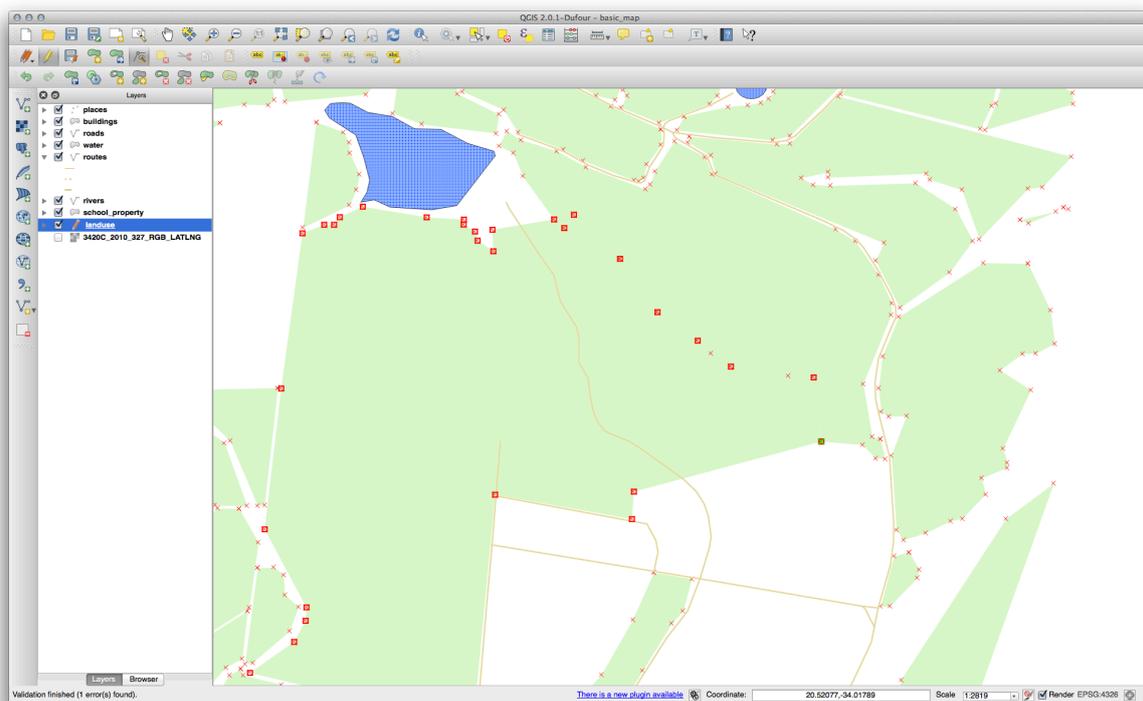
- まだアクティブになっていないなら編集モードにします。
- ノードツールを選択します。
- 森林の 1 つのエリアを選んで、2 つの森林区域が触れ合うように頂点を選択し隣接する頂点へ移動させます:



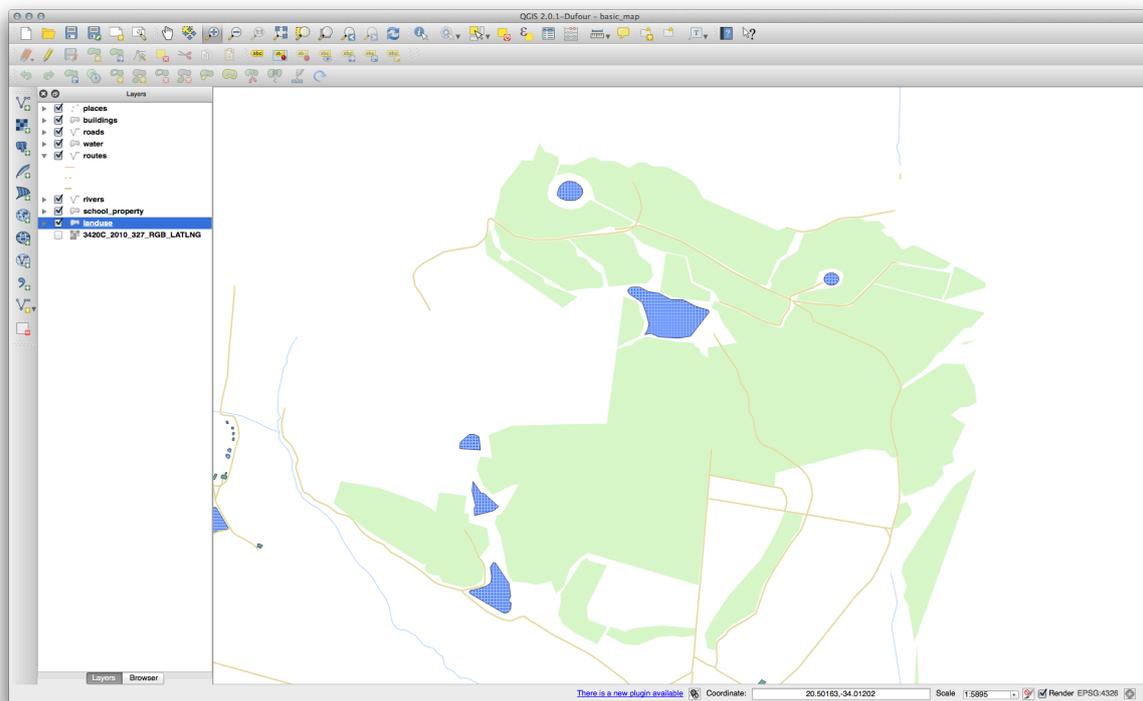
- 所定の位置にスナップされるまでノードをドラッグします。



トポロジとして正しい境界線はこのようになります:



ノードツールを使ってさらにいくつかのエリアを結合させます。適切な場合には 地物の追加 ツールを使用できます。私たちの例のデータを使っているならば、森林エリアはこのように見えます:



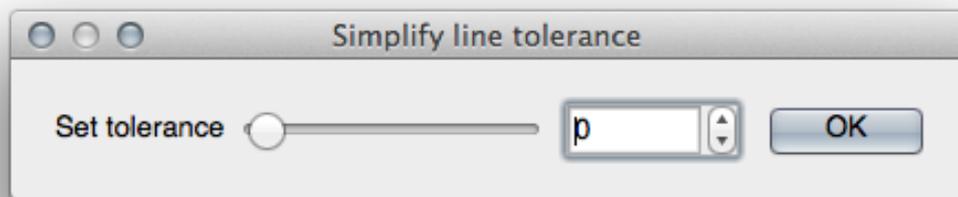
あなたが結合した森林エリアがより多くても、より少なくても、違うエリアであったとしても気にしないで下さい。

6.2.3 Follow Along: ツール: 地物の簡素化

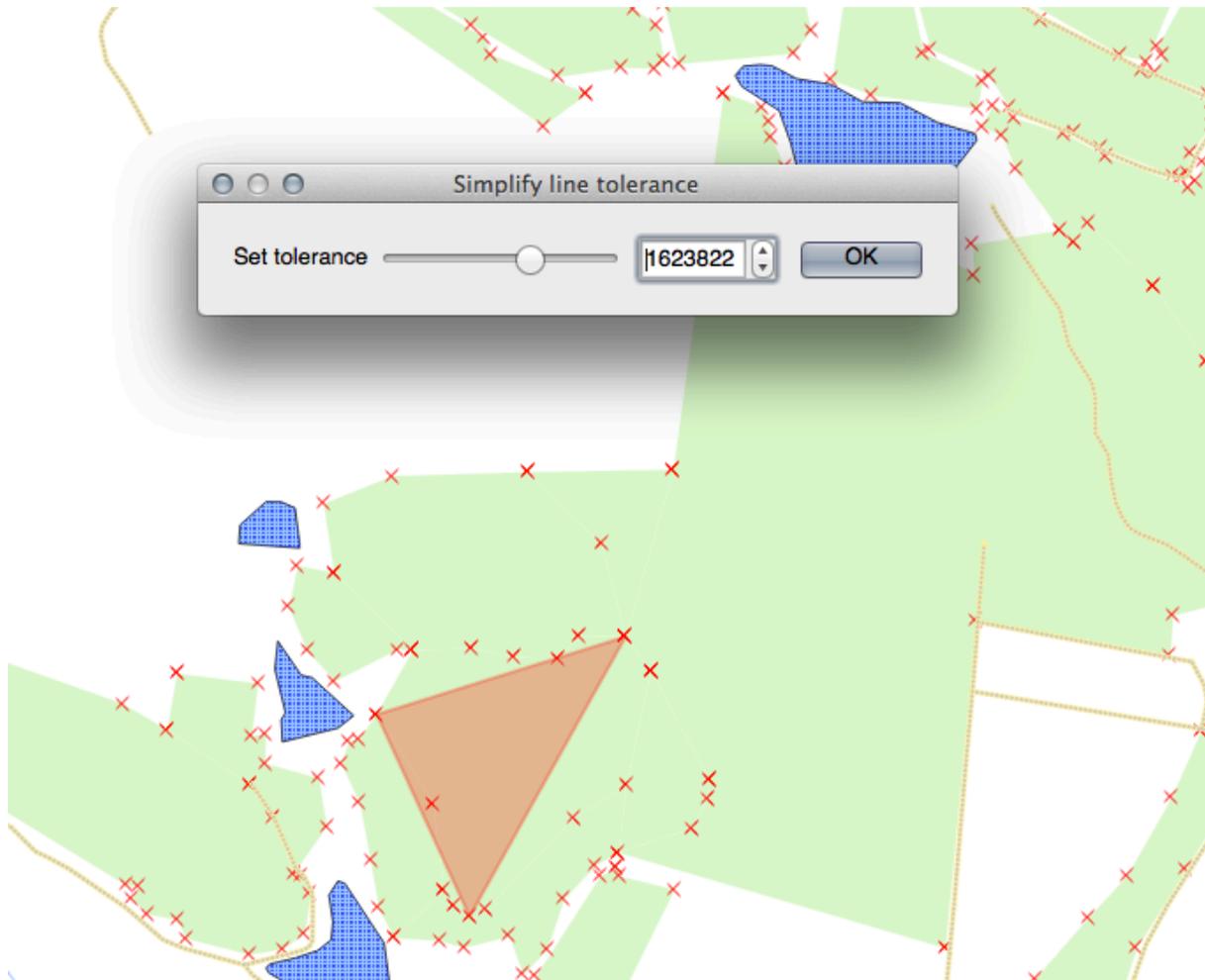
これは 地物の簡素化 ツールです:



- それをクリックしてアクティブにします。
- ノードツール または 地物の追加 ツールを使用して結合したエリアの1つをクリックします。このダイアログボックスが表示されます:



- スライダーを左右に動かして何が起こるか見て下さい:



これで複雑な地物のノードの量を減らすことができます。

- *Ok* をクリックします

このツールがトポロジに何をやるかに注目して下さい。簡素化されたポリゴンは、もはや隣接するポリゴンには触れていません。このツールは独立した地物の一般化に適していることがわかります。その利点は、それが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することです。

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。

6.2.4 Try Yourself ツール: リングの追加

これは リングの追加 ツールです:



これを使うと地物に穴を抜くことができます。穴のすべての辺は地物の内側にある必要があります。たとえば、南アフリカ共和国の外側の境界線をデジタイズし、レソトの穴を追加する必要がある場合はこのツールを使用します。

このツールを試してみると現在のスナップオプションによりポリゴンの真ん中にリングを作成できないことがわかります。もしあなたが除こうとするエリアがポリゴンの境界線につながっている場合にはよいでしょう。

- 以前に使用したダイアログで土地利用レイヤのスナップを無効にします。
- リングの追加 ツールを使用して Bontebok National Park の真ん中にギャップを作成してみます。
- リングの削除 ツールを使用して新しい地物を削除します:



ノート: リングを削除するにはその隅を選択する必要があります。

結果をチェックする

6.2.5 Try Yourself ツール: 部分の追加

これは 部分の追加 ツールです:



これを使うとメインの地物に直接つながっていない余分な部分を作成することができます。たとえば、南アフリカ共和国の本土の境界線をデジタイズしたが、プリンス・エドワード諸島をまだ追加していない場合は、このツールを使用して作成します。

- このツールを使用するには、まず 1 個の地物を選択する ツールを使用して、部分を追加したいポリゴンを選択する必要があります:

|選択|

- 部分の追加 ツールを使って Bontebok National Park に外部のエリアを追加してみます。
- 部分の削除 ツールを使って新しい地物を削除します:



ノート: それを削除するには部分の頂点を選択する必要があります。

結果をチェックする

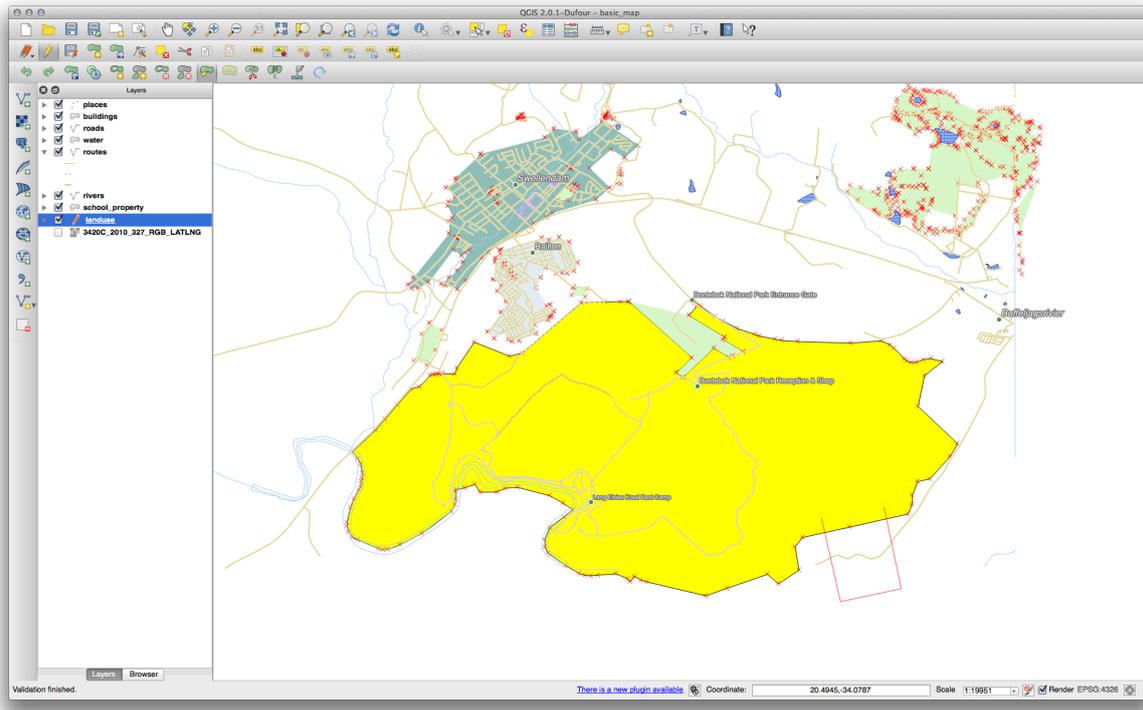
6.2.6 Follow Along: ツール: 地物の変形

これは 地物の変形 ツールです:

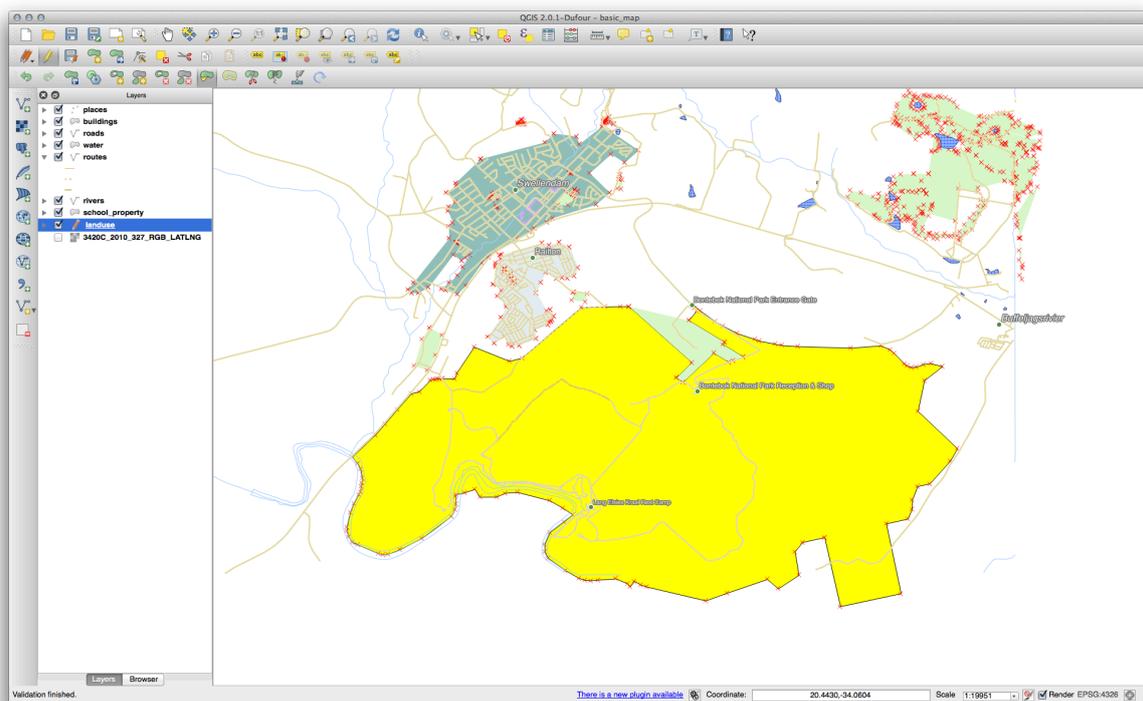
再構築

これを使うと既存の地物にこぶを追加できます。このツールを選択して:

- Bontebok National Park の内部を左クリックしてポリゴンの描画を開始します。
- 3 つの点を追加します。最後は元のポリゴンの内側に戻り、1 辺が開いた長方形を描きます。
- 右クリックしてポイントのマーキングを終了します:

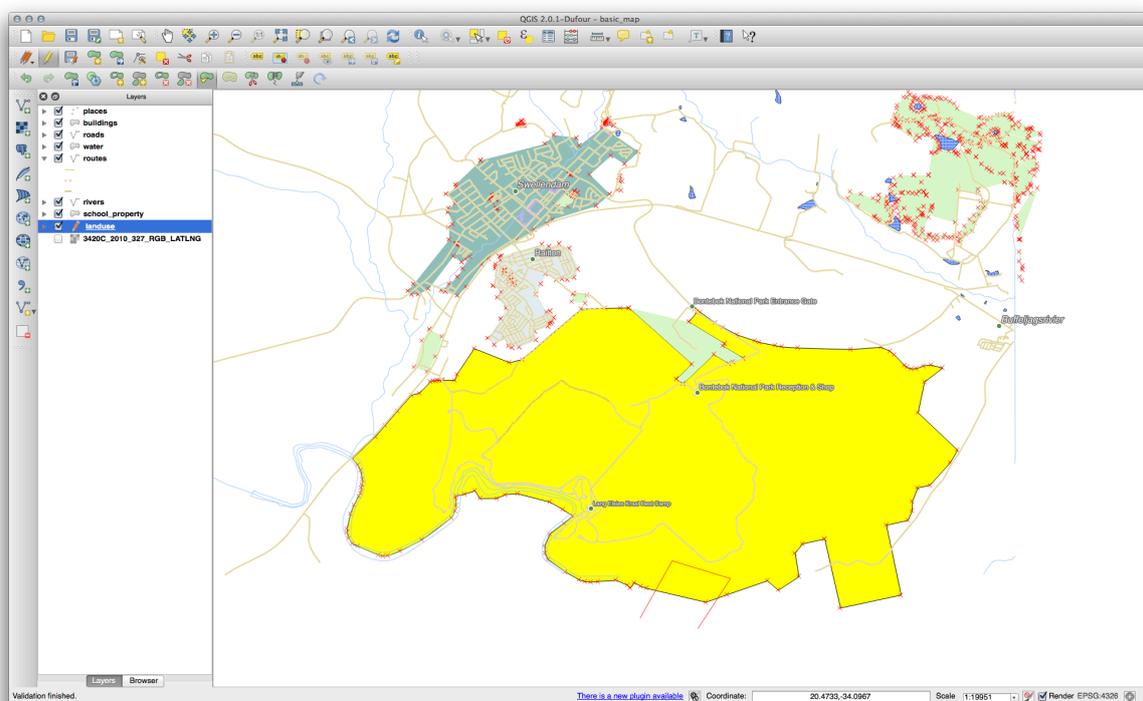


これは次のような結果をもたらします:

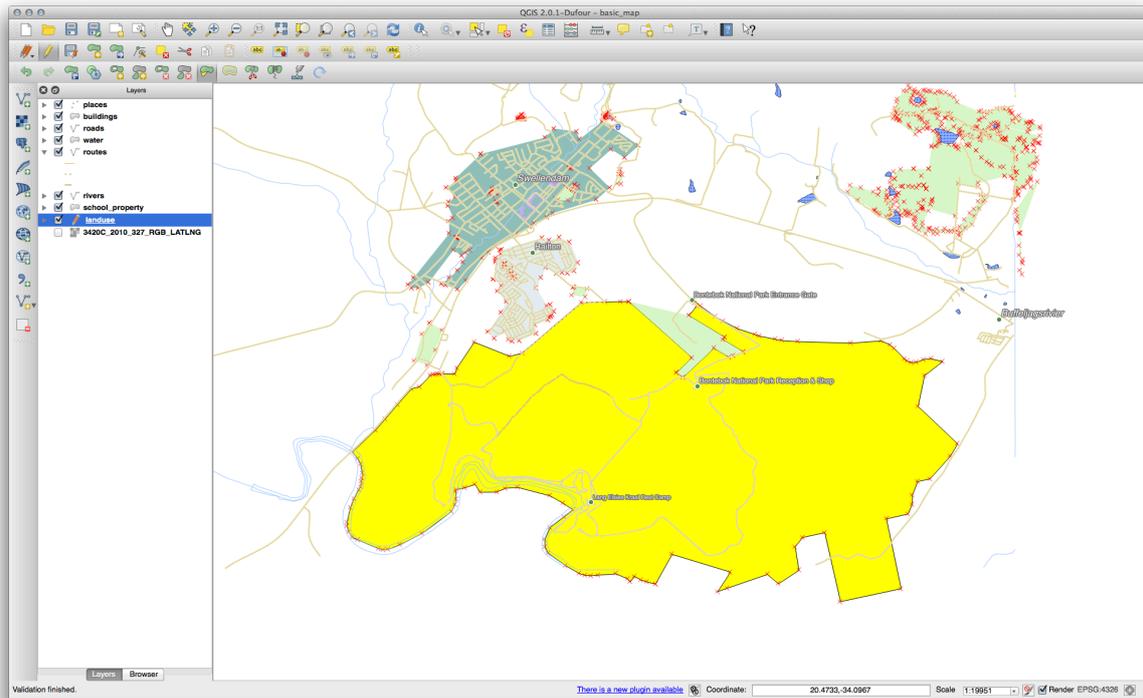


逆の操作を行うこともできます:

- ポリゴンの外側をクリックします。
- ポリゴンに長方形を描画します。
- ポリゴンの外側を再度右クリックします:



以上の結果:



6.2.7 Try Yourself ツール: 地物の分割

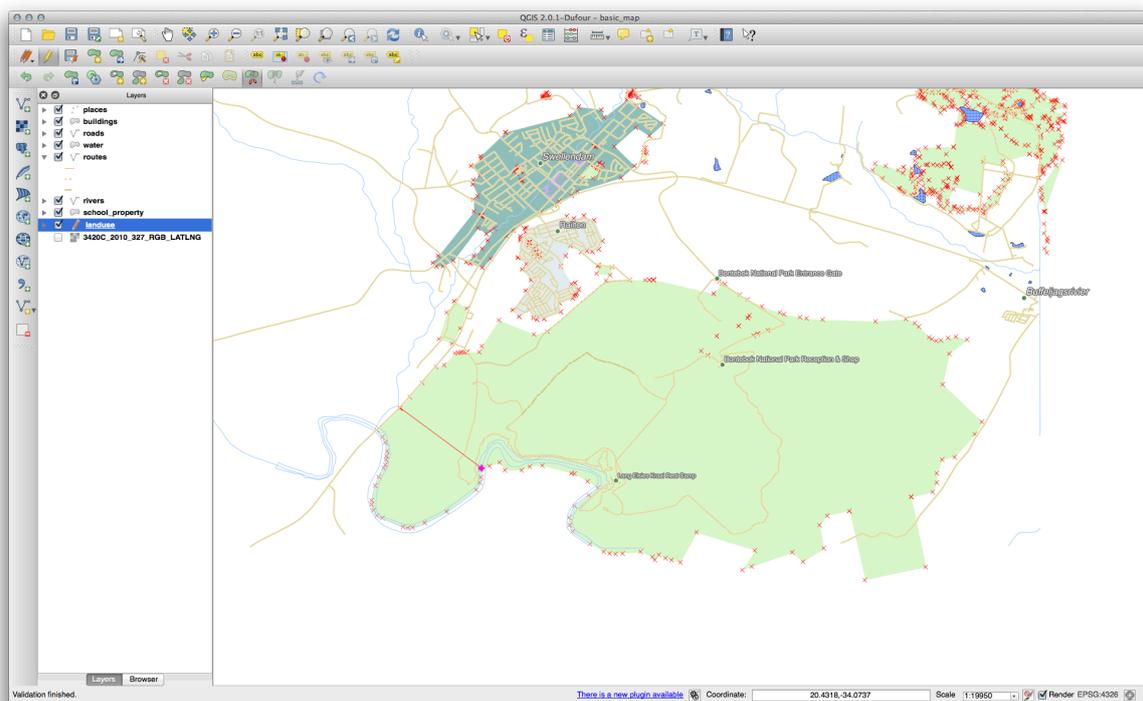
地物の分割 ツールは2つの部分の一方を削除しない点を除いて農場の一部を取り除いた方法と似ています。このツールではそれらの両方を保持します。



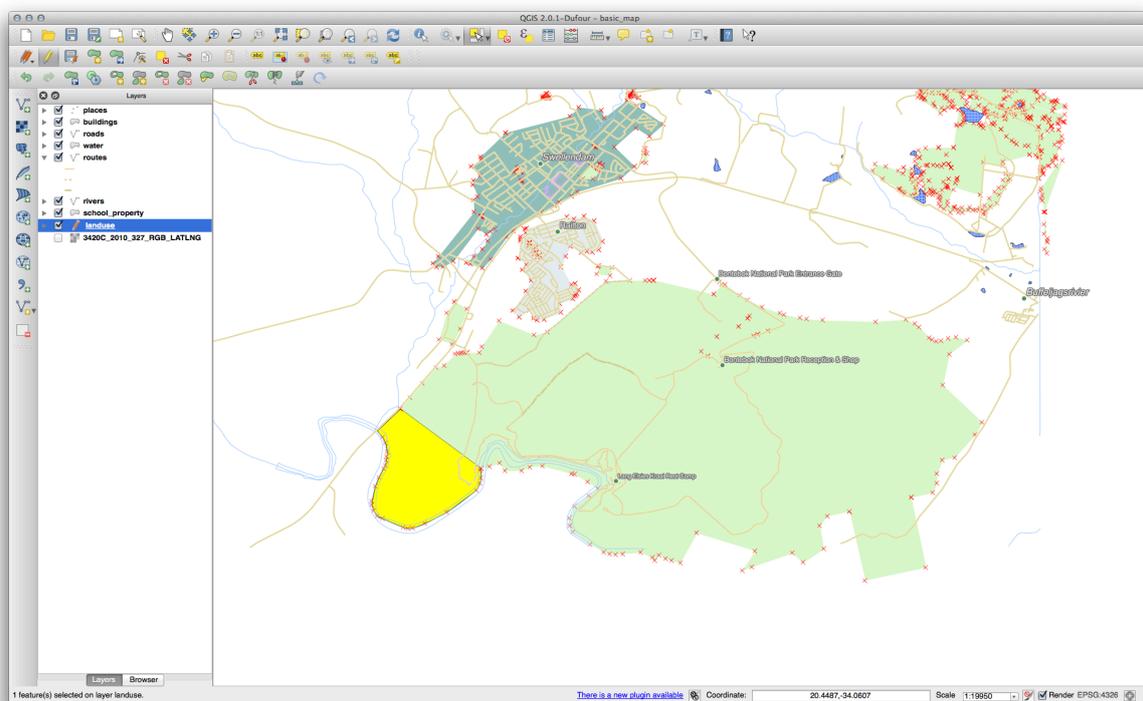
- まず *landuse* レイヤのスナップを再度有効にします。

ツールを使って Bontebok National Park からコーナーを分割します。

- 地物の分離 ツールを選択し、線の描画を開始する点をクリックします。分割するコーナーの反対側の点をクリックし、右クリックして線を完成させます:



- 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が。しかし landuse レイヤのシンボロジにはボーダーがないことを思い出して下さい。したがって新しい区分線は表示されません。
- / 個の地物を選択する を使用して今しがた分割したコーナーを選択します。新しい地物がハイライト表示されます:



6.2.8 Try Yourself ツール: 地物の結合

作成した地物を元のポリゴンに結合します:

- 選択地物の結合 と 選択地物の属性結合 ツールを試します。
- 相違点に注目してください。

結果をチェックする

6.2.9 In Conclusion

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。

6.2.10 What's Next?

オブジェクトの形状を簡単にデジタイズする方法がわかりましたが、属性を追加することにはまだ少し不安があります! 次は属性の編集がより簡単でより効果的であるようにフォームの使い方を学びます。

6.3 Lesson: 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、ユーザビリティ上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することができます。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。

6.3.1 Follow Along: QGIS のフォームデザイン機能の使用

- レイヤリストで *roads* レイヤを選択します。
- 前のように 編集モード にします。
- 属性テーブルを開きます。
- テーブル内の任意のセルを右クリックします。フォームのオープン エントリのある短いメニューが表示されます。
- それをクリックすると QGIS がこのレイヤのために生成したフォームが表示されます。

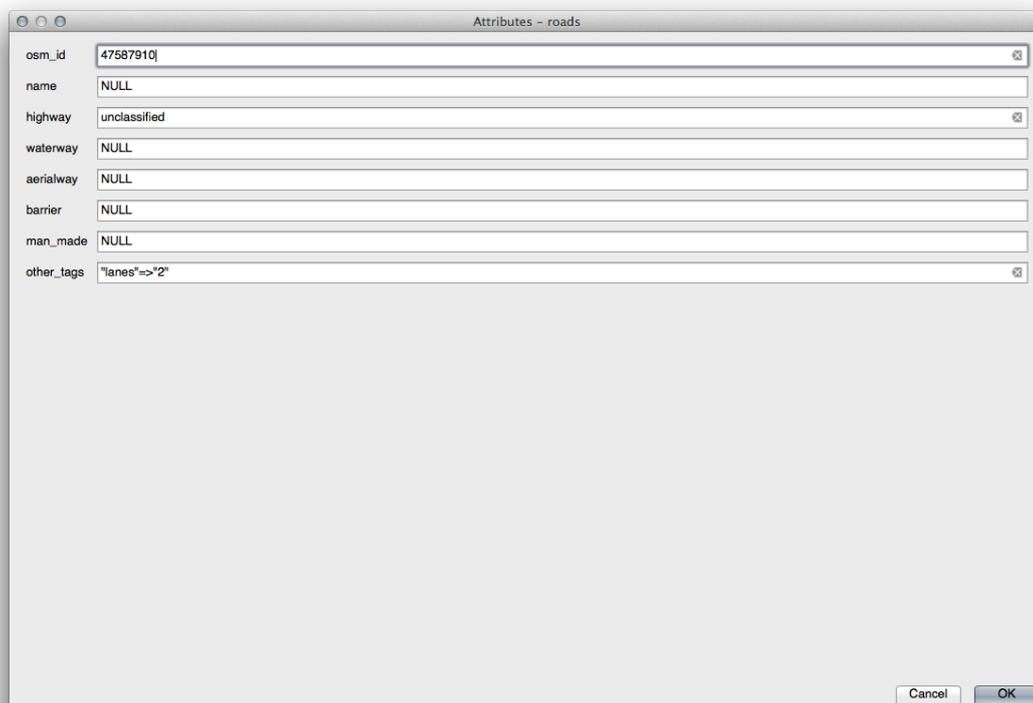
明らかに、毎回 属性テーブル で特定の街路を検索するのではなく、地図を見ながらこれを行うことができるといいですね。

- レイヤリストで *roads* レイヤを選択します。
- 地物情報表示 ツールを使って地図上の任意の街路をクリックします。



- *Identify Results* パネルが開き、ツリービューにクリックされた地物についてのフィールド値および他の一般的な情報が表示されます。
- パネル下部で *Auto open form* チェックボックスをチェックしてください。

- 今、マップ内の任意の路上でもう一度クリックします。前回に沿って：guilabel：‘あなたは今、おなじみのフォームが表示されます、Results’ダイアログを識別します。

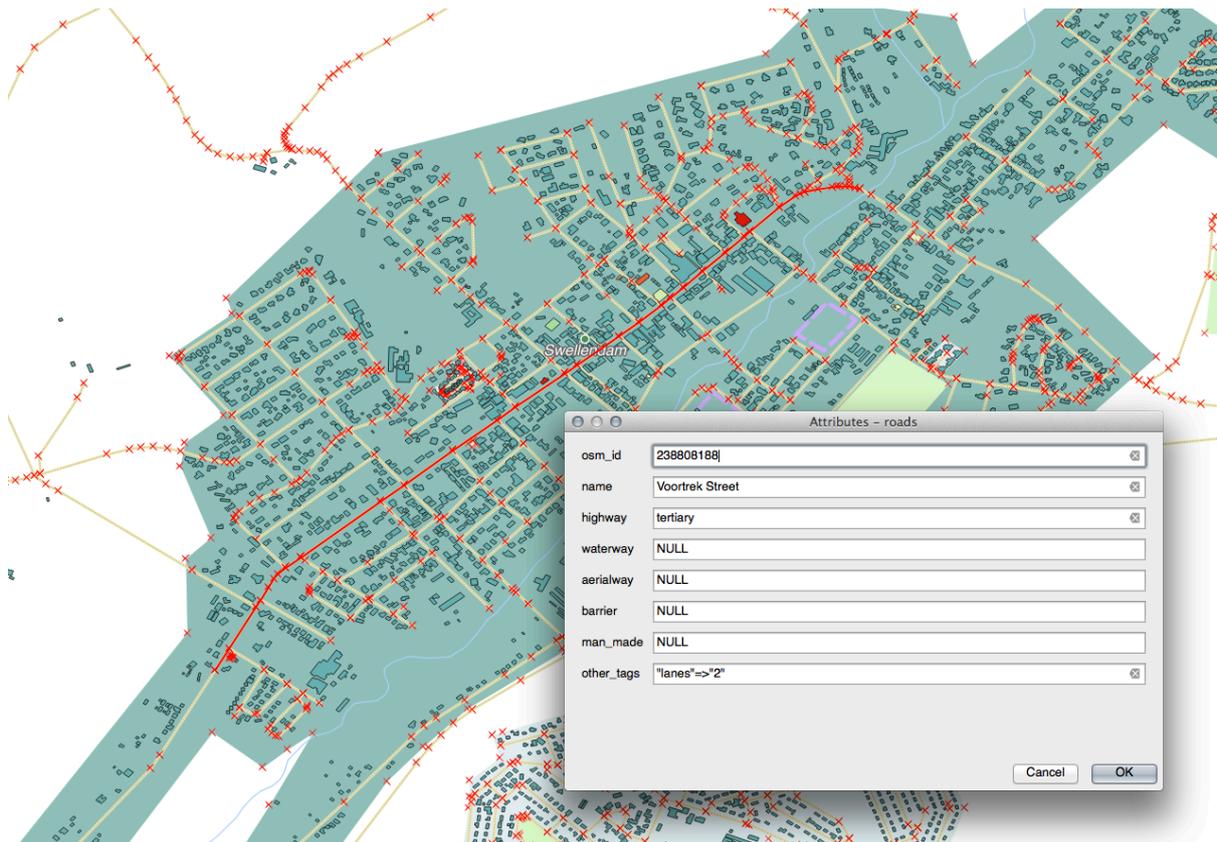


- guilabel：‘Identify’ツール、しない限り、そのフォームのポップ・アップ：guilabel：あなたが持つ単一の機能をクリックするたびに、‘オートオープン form’は未チェックです。

6.3.2 Try Yourself フォームを使った値の編集

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することができます。

- 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)。
- 地物情報表示 ツールを使用して Swellendam を通り抜けるメインストリートをクリックします：



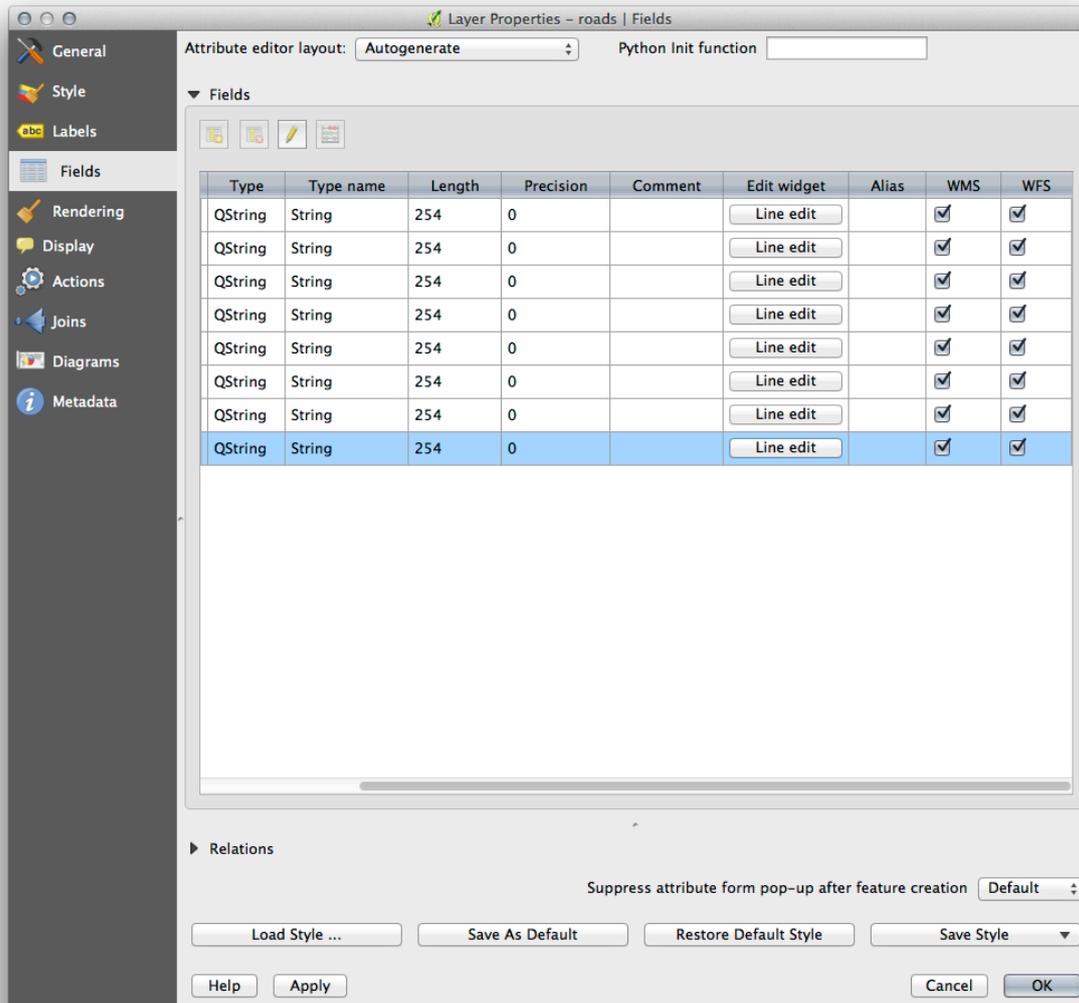
- *highway* の値を変更して *secondary* にします。
- 編集内容を保存します。
- 編集モードを終了します。
- 属性テーブルを開いて、属性テーブルの値が更新されていること、すなわちソースデータが更新されていることに注意して下さい。

ノート: デフォルトのデータセットを使用している場合は、地図には Voortrek Street と呼ばれる道が複数あることがわかります。

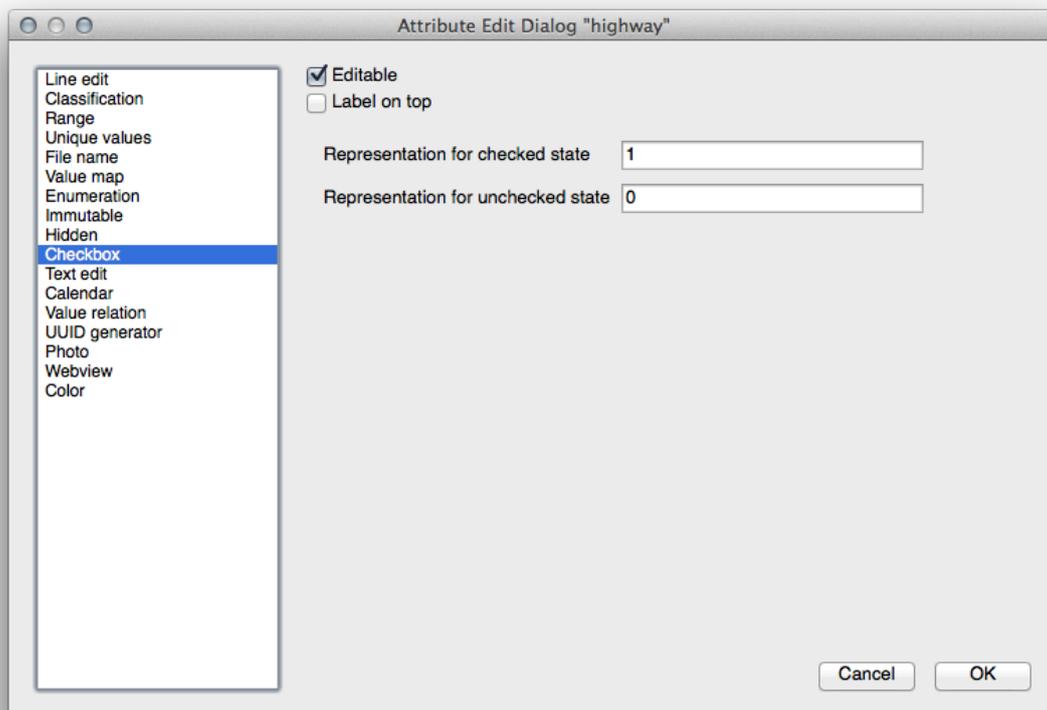
6.3.3 Follow Along: フォームのフィールドタイプの設定

フォームを使用して編集するのはよいのですが、まだ何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。

- *roads* レイヤの レイヤプロパティ を開きます。
- フィールド タブに切り替えます。次が表示されます:



- *man_made* と同じ行の 行編集 ボタンをクリックしてダイアログを開きます。
- オプションのリストから チェックボックス を選択します:



- *OK* をクリックします。
- 編集モードにします (*roads* レイヤが編集モードではない場合)。
- 地物情報表示 ツールをクリックします。
- 前に選択した同じ道路をクリックします。

man_made 属性の隣にチェックボックスが表示されます。チェックされている場合は `True`、チェックされていない場合は `False` を意味します。

6.3.4 Try Yourself

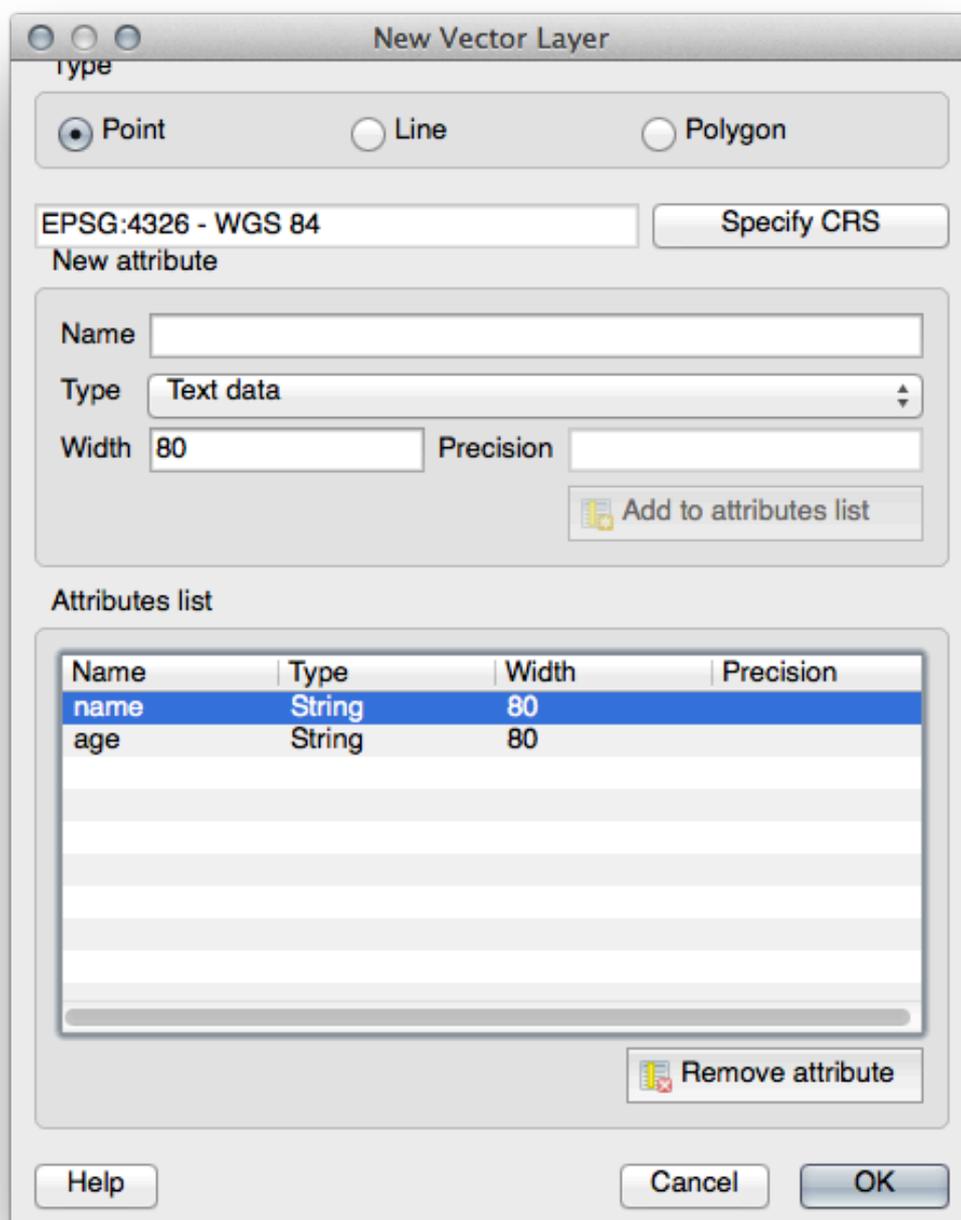
highway フィールドに、より適切なフォームウィジェットを設定します。

結果をチェックする

6.3.5 Try Yourself テストデータの作成

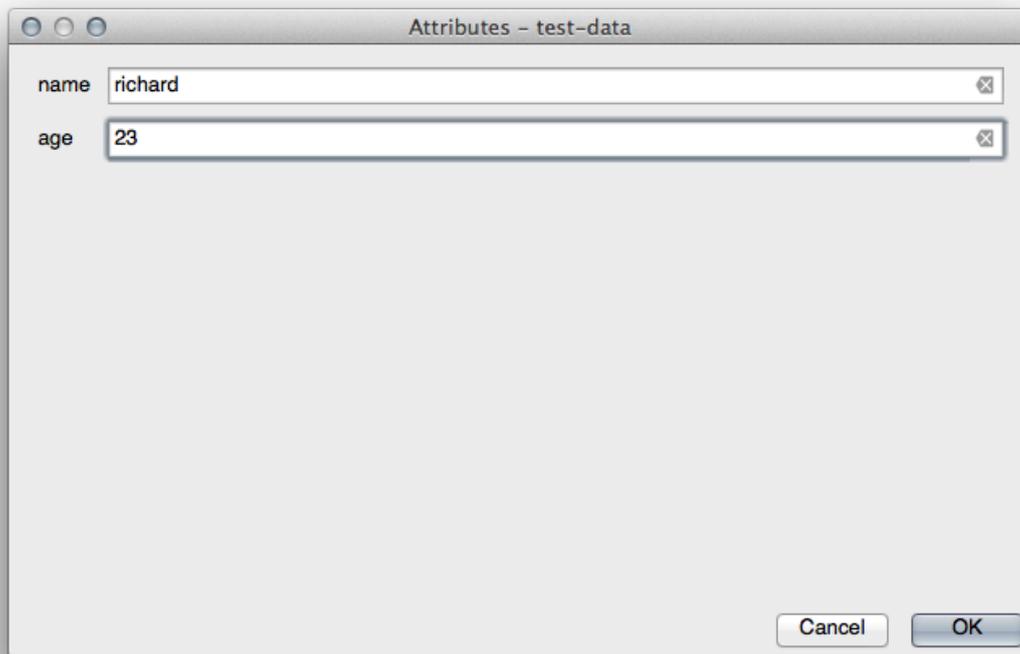
まったくのゼロから独自のカスタム フォームを設計することもできます。

- `test-data` という名前で 2 つの属性を持つ単純なポイントレイヤを作成します:
 - 名前 (テキスト)
 - 年齢 (テキスト)



- 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。

ノート: 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。



6.3.6 Follow Along: 新しいフォームの作成

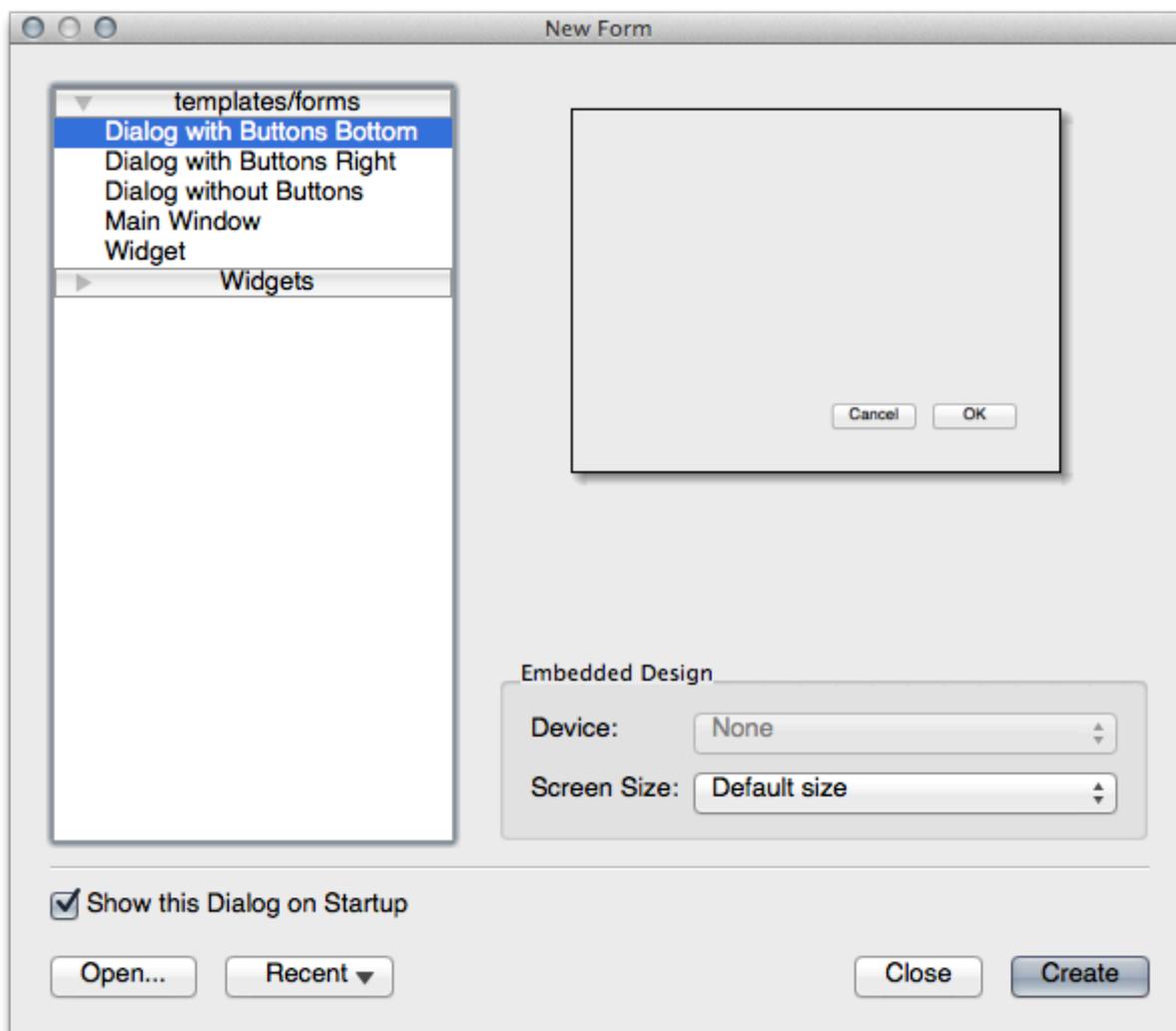
では、属性データのキャプチャ段階のための独自のカスタムフォームを作成したいと思います。これには *Qt4 Designer* がインストールされている必要があります (フォームを作成する人だけに必要です)。Windows を使用している場合はコース教材の一部として提供されているはずですが、他の OS を使用している場合はそれを求める必要があります。Ubuntu ではターミナルで次の操作を行います:

ノート: 執筆時点では Qt5 は利用可能な最新バージョンです。しかし、このプロセスは具体的に Qt4 を必要とし、必ずしも Qt5 と互換性がありません。

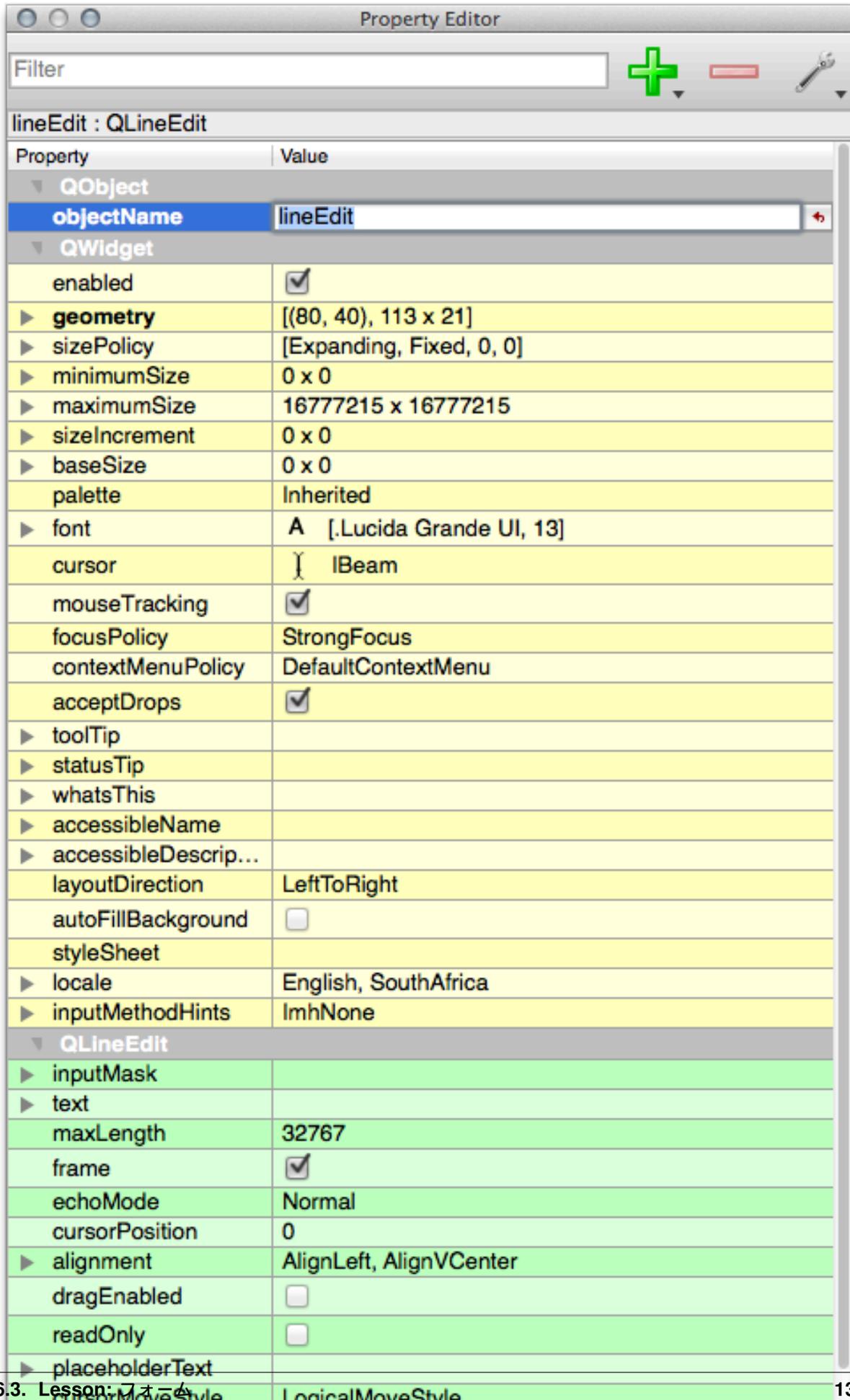
```
sudo apt-get install qt4-designer
```

... とすれば自動的にインストールされるはずですが、そうでなければソフトウェアセンターで探します。

- Windows のスタートメニューを開き (またはお使いの OS に適切なアプローチで)、*デザイナ* を開始します。
- 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:



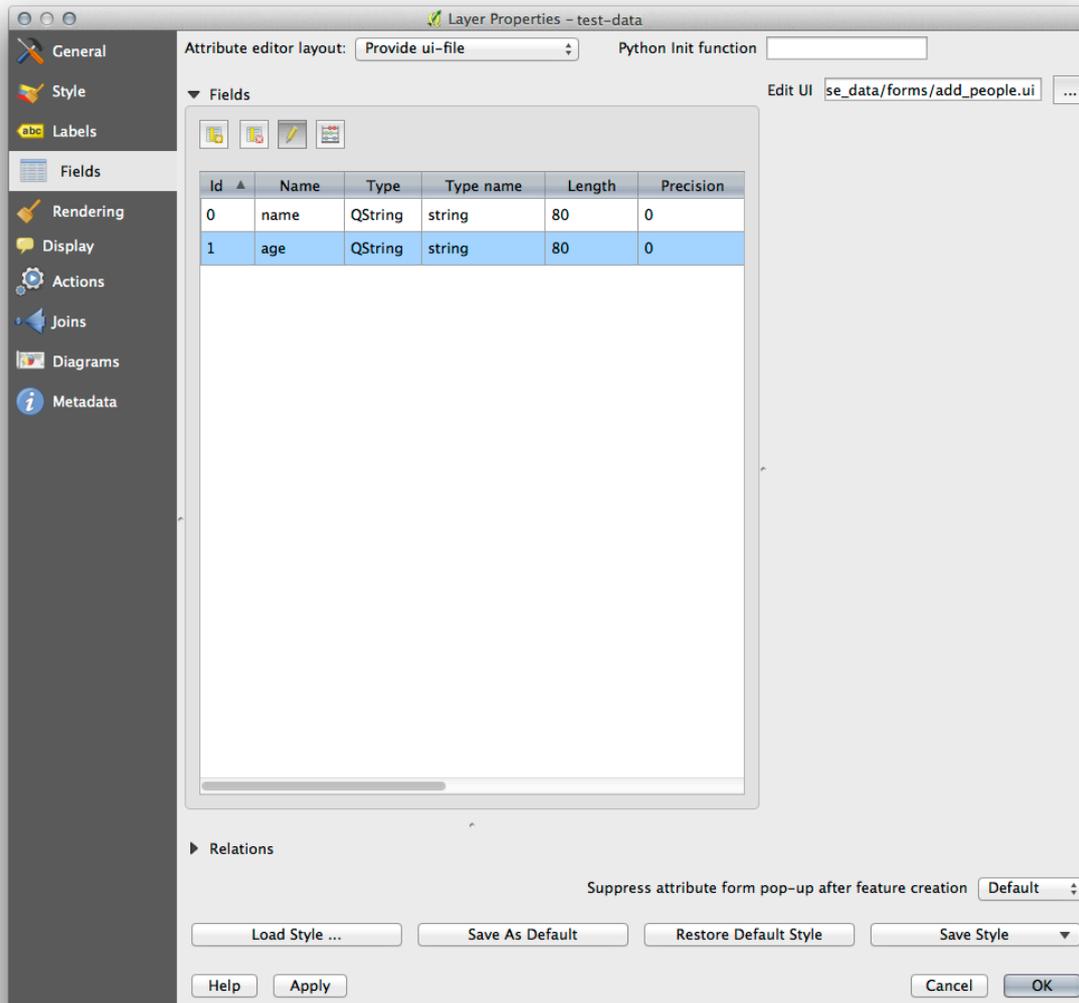
- 画面の左側 (デフォルト) にある ウィジェットボックス で *Line Edit* アイテムを探します。
- このアイテムをクリックしてフォームにドラッグします。フォーム上に新しい *Line Edit* が作成されます。
- *Line Edit* 要素を選択すると、その プロパティ が画面の片側に沿って表示されます (デフォルトで右側):



- 名前を `Name` に設定します。
- 同様に新しい Spin Box を作成し、名前を `Age` に設定します。
- 新しい人の追加 というテキストで `Label` を追加します。フォントは太字にします (設定項目はオブジェクト プロパティ で見つけます)。あるいは (ラベルを追加する代わりに) ダイアログ自体のタイトルを設定することもできます。
- ダイアログの任意の場所をクリックします。
- 垂直に並べる ボタンを見つめます (デフォルトでは画面の上端に沿うツールバーにあります)。これはダイアログを自動的にレイアウトします。
- プロパティでダイアログの最大サイズを 200 (width) x 100 (height) に設定します。
- `exercise_data/forms/add_people.ui` として新しいフォームを保存します。
- 保存が完了したら *Qt4* デザイナ プログラムを閉じます。

6.3.7 Follow Along: レイヤをフォームに関連付ける

- QGIS に戻ります。
- 凡例で `test-data` レイヤをダブルクリックしてプロパティにアクセスします。
- レイヤプロパティ ダイアログの フィールド タブをクリックします。
- 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。
- 省略記号ボタンをクリックして、作成した `add_people.ui` を選択します:



- レイヤプロパティ ダイアログの OK をクリックします。
- 編集モードに切り替えて、新しいポイントを追加します。
- 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)。
- 地物情報表示 ツールを使ってポイントの 1 つをクリックした場合、地物情報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 地物フォームを見る を選択するとフォームを起動することができます。
- このレイヤが編集モードの場合はコンテキストメニューには代わりに 地物編集フォーム が表示されます。キャプチャ後でもフォームで属性の調整ができます。

6.3.8 In Conclusion

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ル化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。

6.3.9 Further Reading

もし上の高度なセクションを完了して、Python の知識がある場合には [このブログエントリ](#) をチェックしてみてください。Python ロジックを用いたカスタム地物フォームの作成についての記事で、データ検証や自動補完などの高度な機能を可能にしています。

6.3.10 What's Next?

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、あなたが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。

6.4 Lesson: アクション

前のレッスンでは既定のアクションを見てきました。今度はあなた独自のアクションを定義します。アクションは地物をクリックしたときに起きるものです。それは地図に多くの特別な機能を追加できます。たとえば、オブジェクトに関する追加情報を取得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全く新しい次元を追加できます!

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。

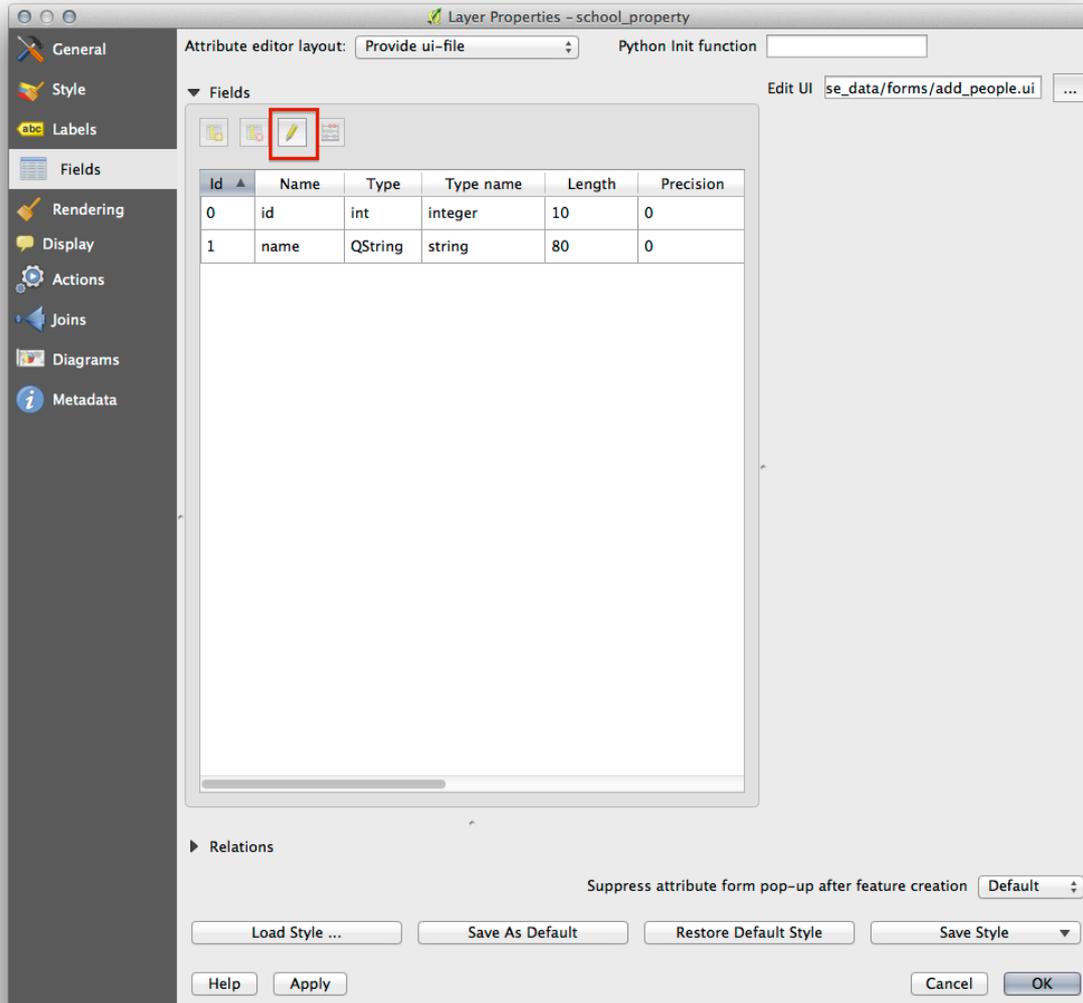
6.4.1 Follow Along: 画像を開く

以前作成した *school_property* レイヤを使用します。コース資料にはあなたがデジタイズした 3 つの地所のそれぞれの写真が含まれます。私たちが次にやろうとしていることはそれぞれの地所とその画像を関連付けることです。それでは、地所をクリックした時にその画像を開くアクションを作成しましょう。

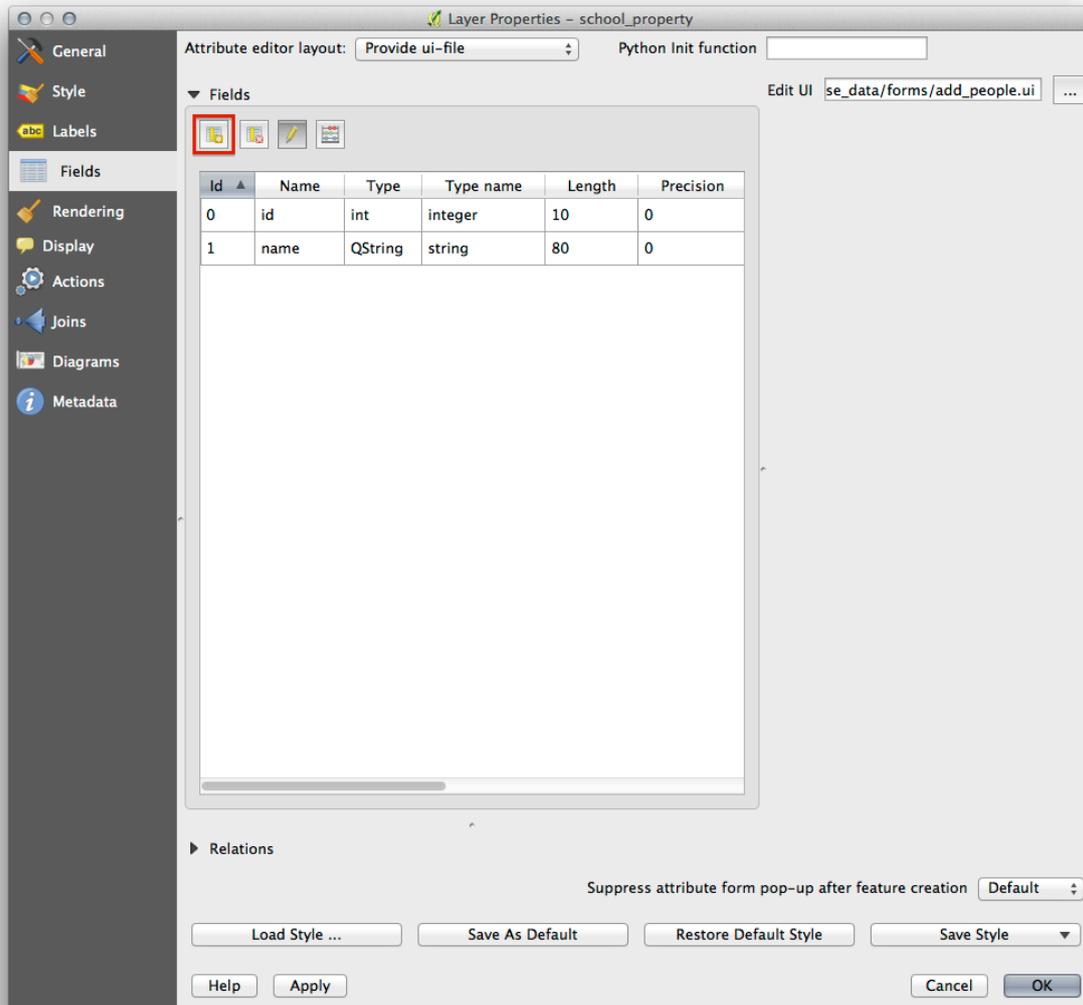
6.4.2 Follow Along: 画像のためのフィールドの追加

school_property レイヤにはまだ画像と地所を関連付ける方法がありません。まずこの目的のためのフィールドを作成します。

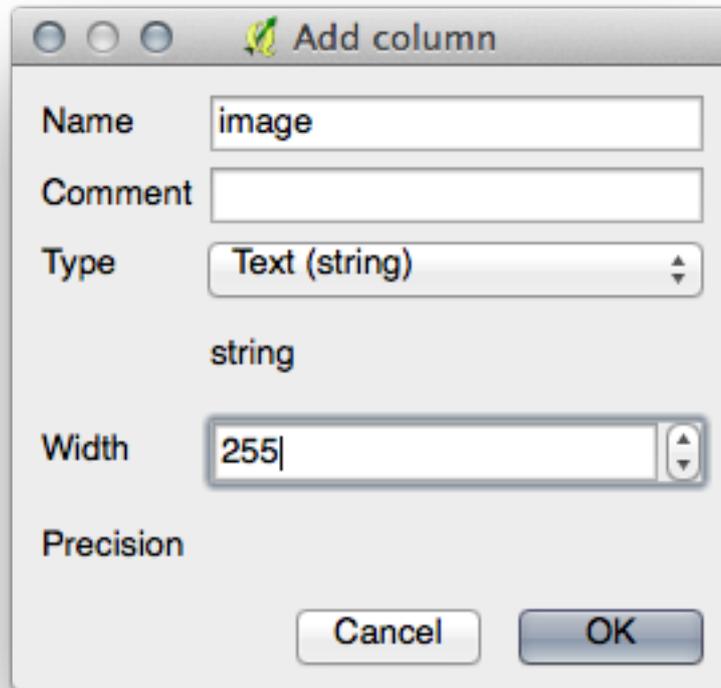
- レイヤプロパティ ダイアログを開きます。
- フィールド タブをクリックします。
- 編集モードに切り替えます:



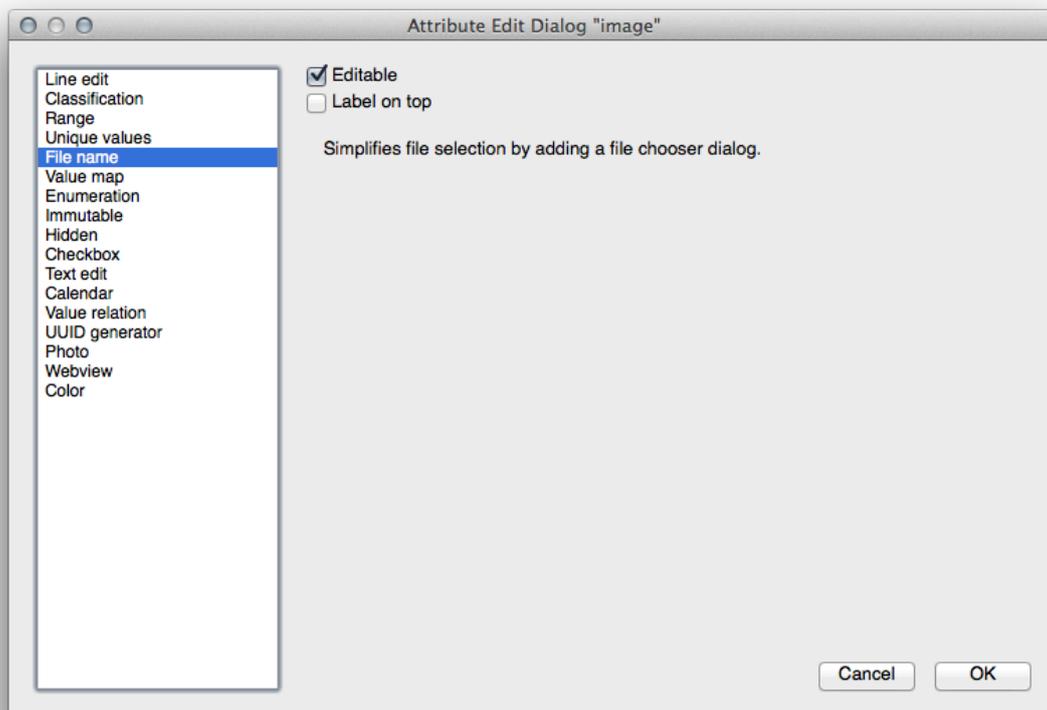
- 新しい列を追加します:



- 下記の値を入力します:

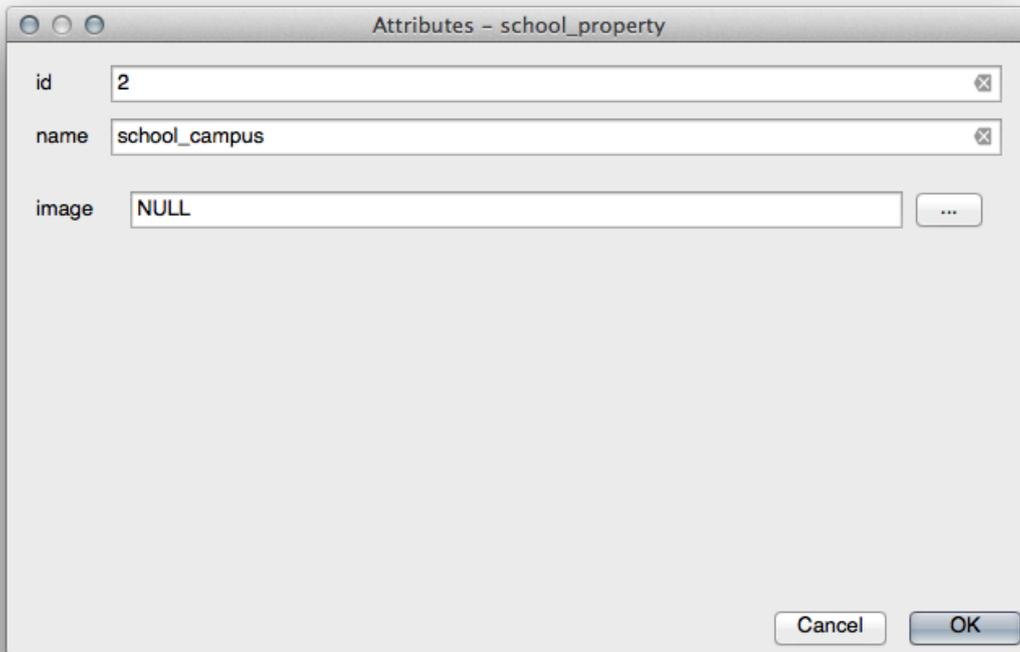


- フィールドを作成した後、新しいフィールドの横にある 行編集 ボタンをクリックします。
- そこで ファイル名 を設定します:



- レイヤプロパティ ダイアログの OK をクリックします。
- 地物情報表示 ツールを使用して *school_property* レイヤの 3 つの地物のいずれかをクリックします。

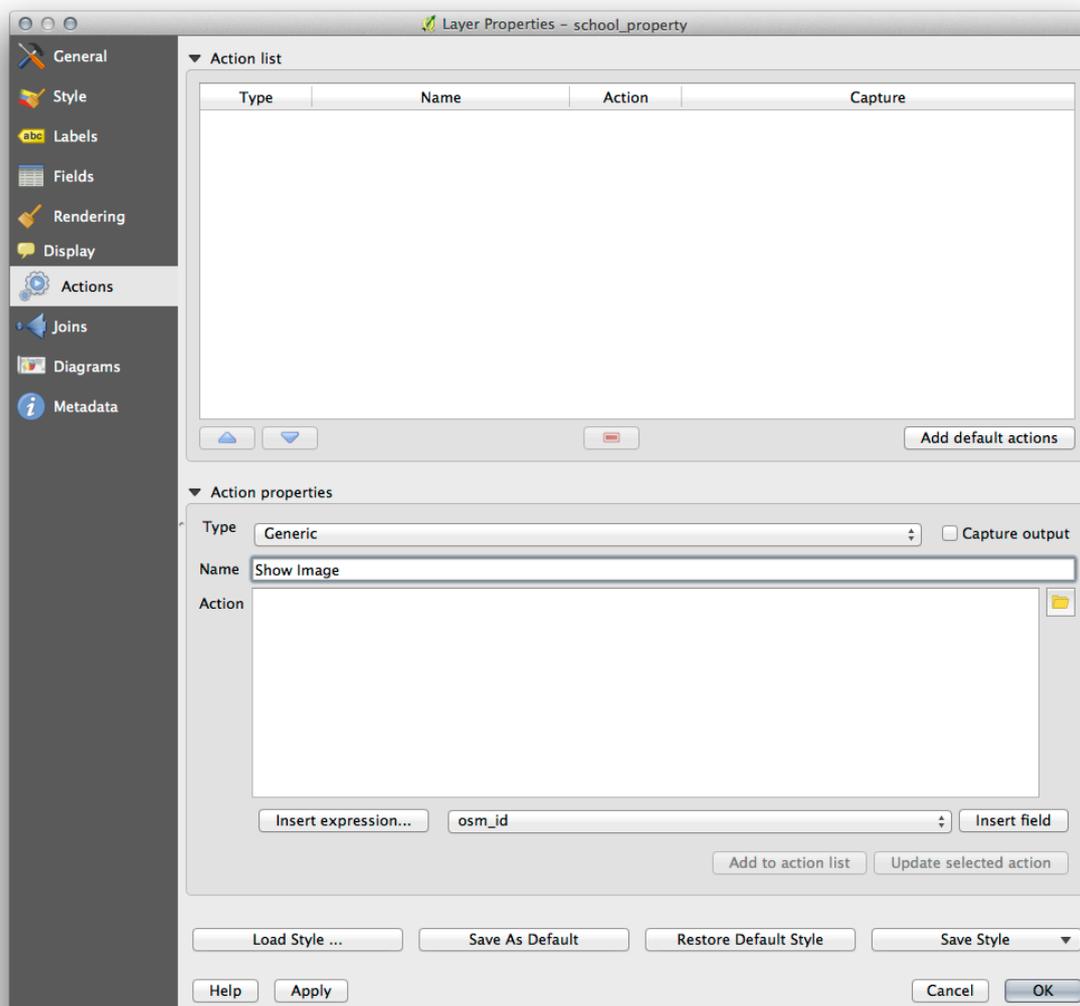
編集モードのままなので、ダイアログがアクティブになり、このようになるはずです:



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

6.4.3 Follow Along: アクションの作成

- *school_property* レイヤのアクション フォームを開きます。
- アクションプロパティ パネルで 名称 フィールドに 画像表示 と入力します:



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

- タイプ ドロップダウンリストをクリックし、開く を選択します。

Ubuntu Linux

- *Gnome Image Viewer* を使用する場合は *Action* に `eog` と入力します。 *ImageMagick* を使用する場合は `display` と入力します。コマンドの後に空白を 1 つ入れることを覚えておいて下さい!

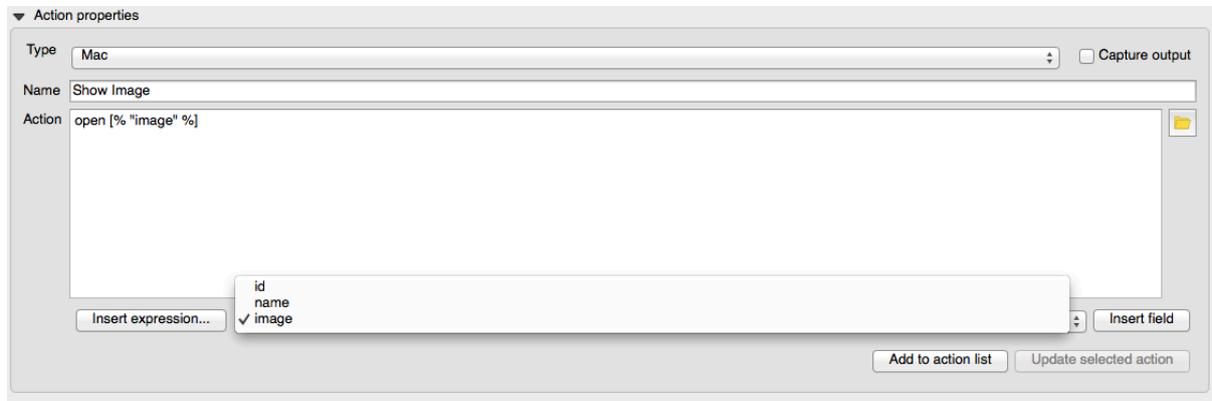
MacOS

- タイプ ドロップダウンリストをクリックし、 *Mac* を選択します。
- アクション に `open` と入力します。コマンドの後に空白を 1 つ入れることを覚えておいて下さい!!

コマンドの入力の続き

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る。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。

- リストから *image* を選択します:



- フィールドを挿入 ボタンをクリックします。QGIS はアクション フィールドに [% "image" %] の句を追加します。
- アクションリストへの追加 ボタンをクリックします。
- レイヤプロパティ ダイアログの OK をクリックします。

さて新しいアクションをテストしましょう:

- レイヤリストで *school_property* レイヤをクリックしてハイライトさせます。
- 地物アクションの実行 ボタンを見つけます (属性テーブルを開く ボタンと同じツールバーにあります):

- このボタンの右にある下向き矢印をクリックします。これまでのところ、このレイヤにはあなたが今定義した 1 つのアクションだけがあります。
- 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。
- 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。
- その地所の画像が表示されます。

6.4.4 Follow Along: インターネットの検索

私たちは地図を見ていて農場がある地域についてもっと知りたくなったとしましょう。その地域について何も知らず、それについての一般的な情報を見つけたいとします。あなたが今コンピュータを使用しているならば、あなたの最初の衝動はおそらく地域の名前を Google で検索することでしょう。では、QGIS にそれを自動的にさせてみましょう!

- *landuse* レイヤの属性テーブルを開きます。

name フィールドを使ってそれぞれの土地利用地域を Google で検索するようにします。

- 属性テーブルを閉じます。
- レイヤプロパティ の アクション に戻ります。
- アクションプロパティ → 名称 フィールドに Google 検索 と入力します。

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

- 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

- アクションに `xdg-open` と入力します。これは Ubuntu に Chrome や Firefox 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。

MacOS

- アクションに `open` と入力します。これは MacOS に Safari 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。

コマンドの入力の続き

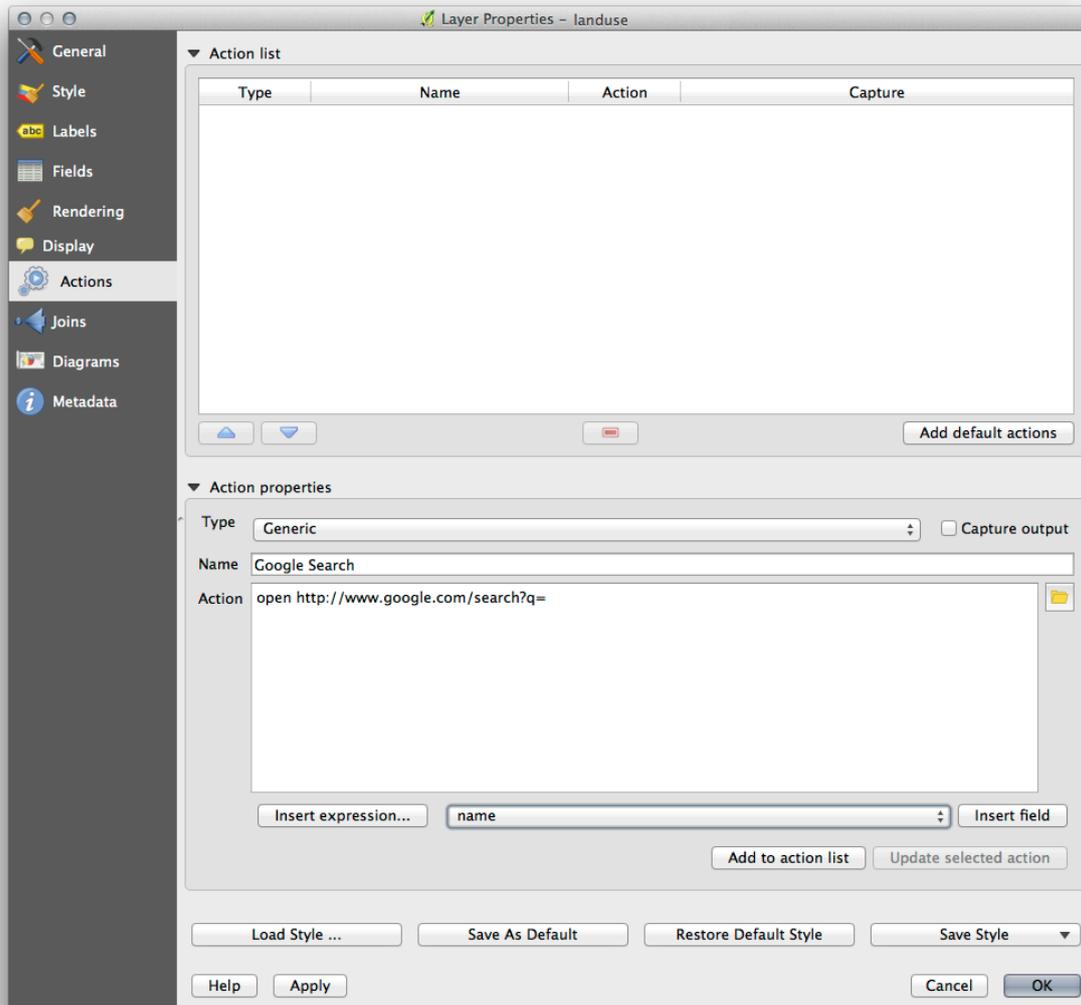
上でどのコマンドを使った場合でも次に、開くべきインターネットアドレスを知らせなければいけません。Google を訪問させて語句を自動的に検索させます。

通常 Google を使用するときには Google の検索バーに検索語句を入力します。しかしこの場合はコンピュータにこれをさせます。Google で何か検索をするには (検索バーを直接使わない場合)、インターネットブラウザにアドレス `http://www.google.com/search?q=SEARCH_PHRASE` を与えます。ここで `SEARCH_PHRASE` はあなたが検索したい何かです。私たちはまだ検索する語句を知らないのはじめの部分 (検索語句はない) だけ入力します。

- アクションフィールドに `http://www.google.com/search?q=` と入力します。コマンドが入力されている場合はコマンドの後にスペースを追加してください!

QGIS でクリックされた地物の `name` の値をブラウザで Google 検索するようにしましょう。

- 名称 フィールドを選択します。
- フィールドを挿入 をクリックします:



これは QGIS に次の句を追加させます:

Type	Mac
Name	Google Search
Action	open http://www.google.com/search?q=[% "name" %]

これが意味することは QGIS がブラウザを開き、それにアドレス `http://www.google.com/search?q=[% "name" %]` を送ることです。しかし、`[% "name" %]` は QGIS に検索語句として name フィールドの内容を使わせます。

たとえば、クリックした土地利用地域の名前が Marloth Nature Reserve である場合、QGIS はブラウザに `http://www.google.com/search?q=Marloth%20Nature%20Reserve` を送ります。すると、ブラウザは Google を訪問し、"Marloth Nature Reserve" を検索するでしょう。

- もしまだ出来ていないなら、上で説明したように設定して下さい。
- アクションリストへの追加 ボタンをクリックします。新しいアクションが上のリストに表示されます。
- レイヤプロパティ ダイアログの OK をクリックします。

では新しいアクションをためします。

- レイヤリスト で *landuse* レイヤをアクティブにして、地物アクションの実行 ボタンをクリックします。
- 地図上の任意の土地利用地域をクリックします。ブラウザーが起動し、自動的に、地域の *name* の値として記録されている町の Google 検索が開始されます。

ノート: 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!

6.4.5 Follow Along: QGIS で直接 Web ページを開く

これまで外部のブラウザで Web ページを開く方法を見てきました。このアプローチには不可知な依存関係を追加するという点で若干の欠点があります。エンドユーザが自分のシステムでアクションを実行するのに必要なソフトウェアを持っているでしょうか? これまで見てきたように、どの OS を使っているかわからなければ同じようなアクションのための同じような基本コマンドでさえあるとは限りません。OS のバージョンによってはブラウザを開く上記コマンドがまったく動作しないかもしれません。これは克服できない問題である可能性があります。

しかし、QGIS は信じられないほど強力で汎用性の高い Qt4 ライブラリの上にあります。さらに QGIS のアクションには任意でトークン化された (すなわち、フィールドの属性の内容にもとづいた変数情報を使用する) Python コマンドが使えます!

では 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。それは外部のブラウザでサイトを開くのと大体同じ考えですが、Qt4 の QWebView クラス (webkit ベースの HTML ウィジェット) を使用し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のシステムにはブラウザは必要ありません。

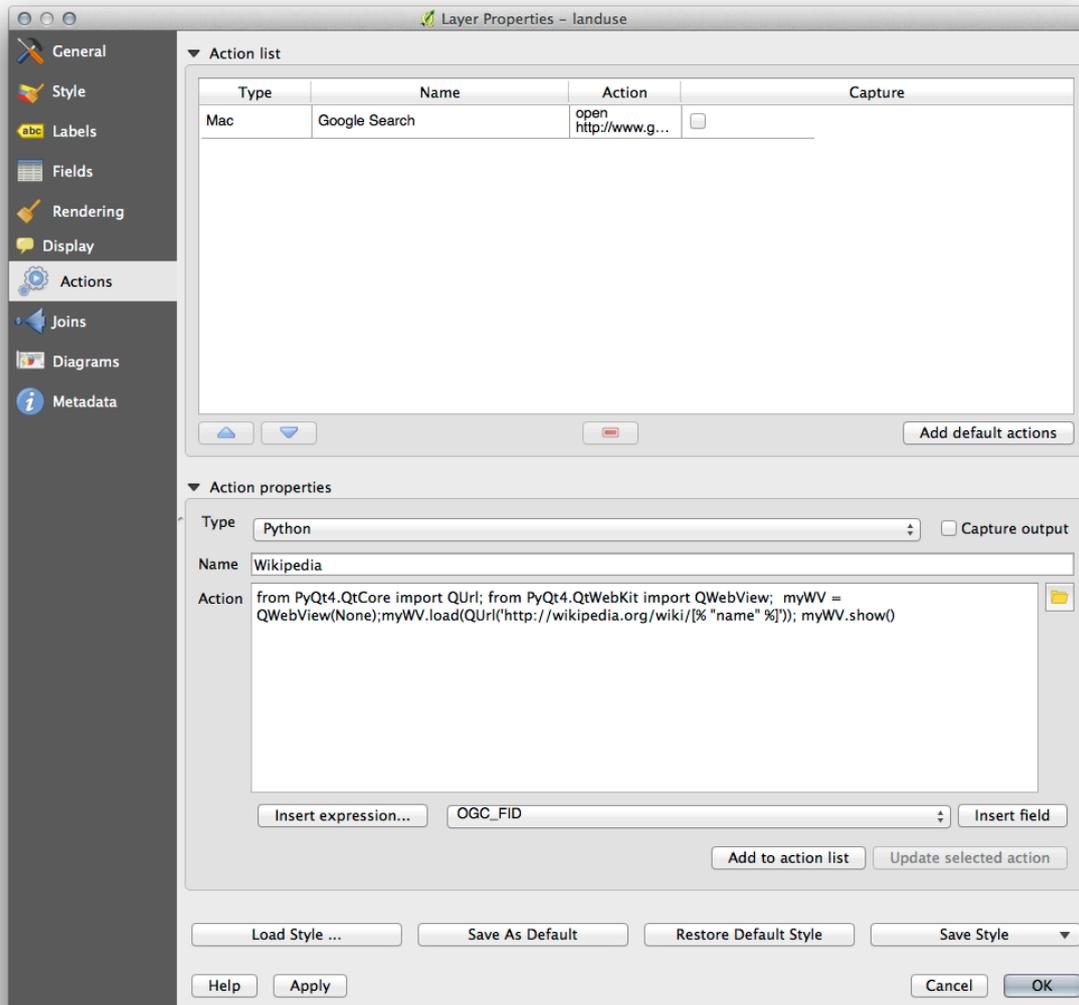
今回は Google の代わりに Wikipedia を使ってみましょう。そうするとリクエストの URL は次のようになります:

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

レイヤアクションを作成するには:

- レイヤプロパティ ダイアログを開いて アクション タブへ向かいます。
- 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。
 - *Type*: Python
 - *Name*: Wikipedia
 - *Action* (all on one line):


```
from PyQt4.QtCore import QUrl; from
PyQt4.QtWebKit import QWebView; myWV = QWebView(None);
myWV.load(QUrl('http://wikipedia.org/wiki/[% "name" %]'));
myWV.show()
```



ここでは:

- すべての Python のコードはコマンドがセミコロンで区切られて 1 行になっています (通常 Python のコマンドは改行で区切ります)。
- [% "name" %] はアクションが呼び出される時に実際の属性値と置き換えられます (これまでのように)。
- コードは単に新しい QWebView インスタンスを作成し、その URL を設定し、ユーザーのデスクトップ上のウィンドウとして表示させる show() を呼び出します。

これはいくぶん不自然な例であることに注意してください。Python では意味的に重要なインデントが使われますので、セミコロンで区切ることは記述の最良の方法ではありません。ですから現実の世界では、Python モジュールからロジックをインポートして、引数にフィールド属性をとる関数を呼び出す可能性が高いと思います。

同様にしてユーザのシステムに特定のイメージビューアがあることを要求せずに画像を表示するアプローチを使用できるでしょう。

- 今しがた作成した Wikipedia アクションを使用して Wikipedia のページを読み込む上記の方法を試して下さい。

6.4.6 In Conclusion

アクションを使えば地図に特別な機能を与えることができます。それは QGIS で同じ地図を見るエンドユーザに役立ちます。Python だけでなく任意のオペレーティングシステムのシェルコマンドも使用することができます。あなたが具体化できる機能に関して不可能はありません!

6.4.7 What's Next?

あなたはあらゆる種類のベクタデータ作成を行いました。問題を解決するためにこのデータを分析する方法を学びます。それは次のモジュールのトピックです。

Chapter 7

Module: ベクタ分析

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クライアントのために住宅を探しています：

1. Swellendam である必要がある。
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、我々はこの新しい住宅開発に適したファームのプロパティを見つけるために、GIS 解析ツールの力を利用します。

7.1 Lesson: データの再投影と変形

座標参照系 (CRSs) について再度話しましょう。前にも簡潔に触れましたが、それは実質的に何を意味するのか議論していませんでした。

このレッスンの目標：ベクタデータセットの再投影と変形をする。

7.1.1 Follow Along: 投影法

マップ自身だけでなくすべてのデータの CRS は WGS84 と呼ばれています。これはデータを表現するのに一般的な空間参照系 (GCS) です。しかし、我々が見るように、問題があります。

- 現在のマップを保存します。
- `exercise_data/world/world.qgs` で確認できる世界地図を開きます。
- *Zoom In* ツールを使って南アフリカにズームします。
- 縮尺 フィールドで縮尺を設定してみてください。このフィールドは画面の下部に沿って `guiLabel:ステータスバー` 中にあります。南アフリカ上している間、この値を `kbd:'1:5000000'` (1 対 5 百万) に設定します。
- スケール フィールドに着目したままマップ周辺をパンニングします。

縮尺が変化していることに気づきます?それはに:kbd:‘1:5000000’でズームインした一点、画面の中央にあった、から離れて移動しているためです。すべてのその点を中心に、縮尺が異なっています。

理由を理解するために、地球の世界を考えます。それは北から南に沿って引かれる線があります。これらの経線は赤道で遠く離れますが、極で出会います。

GCS では、この球に取り組んでいるが、画面は平坦です。平らな面に球を表現しようとすると、歪みが発生し、テニスボールを開いてカットし、それを平らにしようとした場合に何が起こるかに似ています。これは地図上の意味は、経線も（それらは会うことになっている）の極で、互いに均等に離れて滞在するという事です。これは、地図上の離れた赤道から旅行として、見る対象の規模が大きくなると大きくなる、ということの意味します。これが私たちのために何を意味するのか、事実上、私たちの地図には一定の縮尺はないことです!

この問題を解決するには、のではなく、システム (PCS) を投影座標を使用してみましょう。PCS「プロジェクト」や縮尺変更のための手当を作り、それを修正する方法でデータを変換します。そのため、一定の規模を維持するために、我々は、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。

7.1.2 Follow Along: “オンザフライ” 再投影

QGIS ではデータを「その場で」再投影できます。これが意味することは、データ自体は別の CRS である場合でも、選択された CRS のものであるかのように、QGIS ではそれを投影できるということです。

- 「その場で」投影を有効にするには、QGIS ウィンドウの下部に沿った ステータス Bar にある *CRS Status* ボタンをクリックしてください:



- 表示されたダイアログで「その場で」CRS 変換を有効にする の隣のボックスをチェックします。
- *Filter* フィールド に語 :kbd:‘global’ を入力します。1 つの CRS (:guilabel: NSIDC の EASE-グリッド Global) が下のリストに表示されます。
- *NSIDC EASE-Grid Global* をクリックしてそれを選択します。それから OK をクリックします。
- 南アフリカの形状が変化するの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。
- 前のように、再度 1:5000000 のスケールにズームします。
- マップをパンニングします。
- スケールは同じ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。

- “オンザフライ” 再投影を再び無効にする:
 - *CRS* ステータス ボタンを再度クリックします。
 - ‘オンザフライ’ *CRS* 変換を有効にする のチェックを解除します。
 - Clicking *OK*.
- 別の CRS を有するレイヤーが地図にロードされたときに QGIS 2.0 では、「その場で」再投影が自動的に起動されます。「その場で」再投影がすることは何かを理解するため、この自動設定を無効にします:
 - Go to *Settings* → *Options...*
 - On the left panel of the dialog, select *CRS*.
 - レイヤーが異なる *CRS* を持つ場合に自動的に「その場で」再投影を有効にする のチェックを外します。
 - Click *OK*.

- 南アフリカのデータだけを持っている地図に別のベクトルレイヤーを追加します。exercise_data/世界/RSA.shp としてそれを見つけることができます。

何に気づくのですか？

レイヤは表示されません！しかし、修正するのは簡単ですよ？

- Right-click on the *RSA* layer in the *Layers list*.
- レイヤの領域にズームを選択します。

そう、今は南アフリカが見えます … しかし世界の残りはどこですか？

これらの二つのレイヤーの間でズームできることが判明したが、これまでそれらを同時に見ることはできません。その座標参照系は非常に異なっているからです。contинents データセットは度であるが、RSA データセットはメートルです。それでは、ケープタウンの与えられた点としましょう RSA データセットは、およそ次のとおりです。4 100 000 メートル離れた赤道から。しかし、中に continents データセット、その同じポイントは約 33.9 度離れた赤道から。

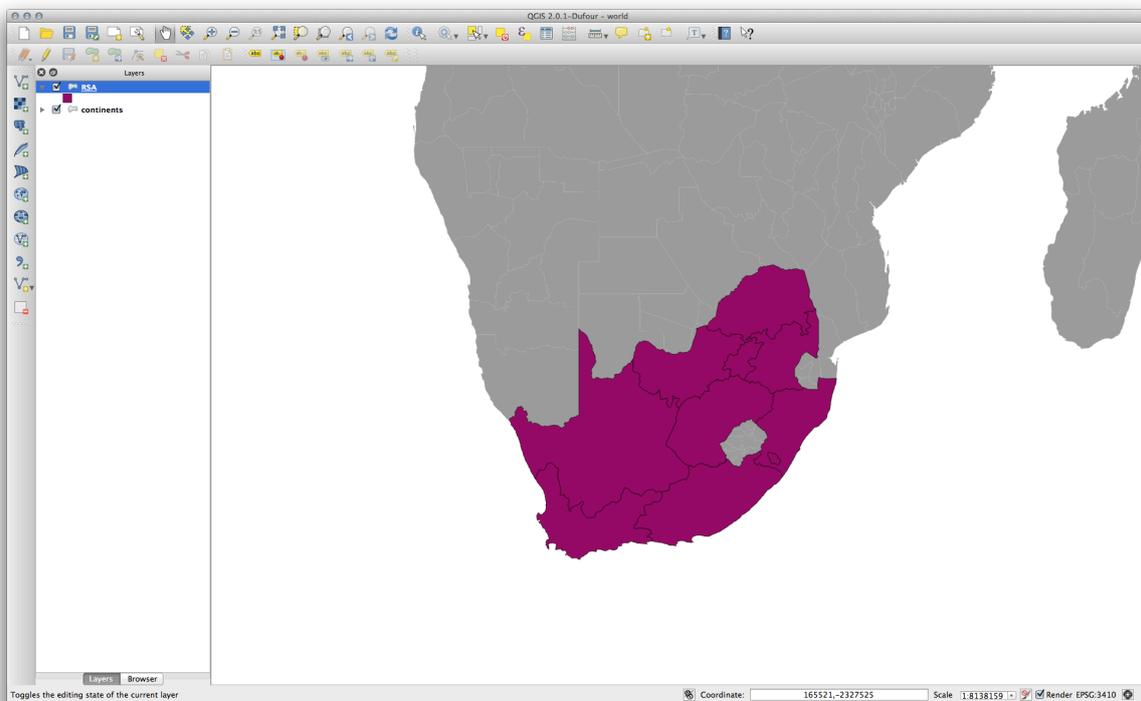
これは、同じ距離である - しかし、QGIS はそれを知りません。データを再投影することを告げていません。だから、限り、それが関係だとして、私たちが見南アフリカのバージョン RSA データセットは、正しい距離でケープタウンがあります赤道から 4 100 000 メートル。しかし、continents データセット中で、ケープタウンは赤道からわずか 33.9 メートル 離れたです！これが問題である理由を見ることができます

ケープタウンがあると 想定 されるところを QGIS は知りません -それはデータが QGIS に伝えるべきことだ。QGIS にケープタウンが赤道から 34 メートル離れていて、南アフリカが北から南へ約 12 メートルしかないデータが伝えた場合、それが QGIS が描画するだろうものです。

これを正すには:

- もう一度:guilabel:CRS Status ボタンをクリックして:guilabel:「その場で」CRS 変換を有効にする を再び以前のように切り替える。
- RSA データセットの範囲にズームします。

同じ CRS に投影しているので、今、2つのデータセットは、完璧に合致します：



異なるソースからのデータを結合すると、それは彼らが同じ CRS でないかもしれないことを覚えておくことが重要です。「オンザフライ」再投影は、あなたがそれらを一緒に表示することができます。

続ける前に、おそらく、オープンデータセットが異なる CRS を持つたび「オンザフライ」再投影が自動的にアクティブにする必要があります。必要は：

- 再度 設定 → オプション... を実行し、CRS を選択します。
- もしレイヤが異なる座標系を持つ場合、自動で「オンザフライ」リプロジェクションを有効にする をチェックします。

7.1.3 Follow Along: 他の CRS に設定したデータセットの保存

:guilabel:‘Classification‘ レッスンで建物のための領域を計算したときを覚えていますか？それは地域に応じて建物を分類できるようにやりました。

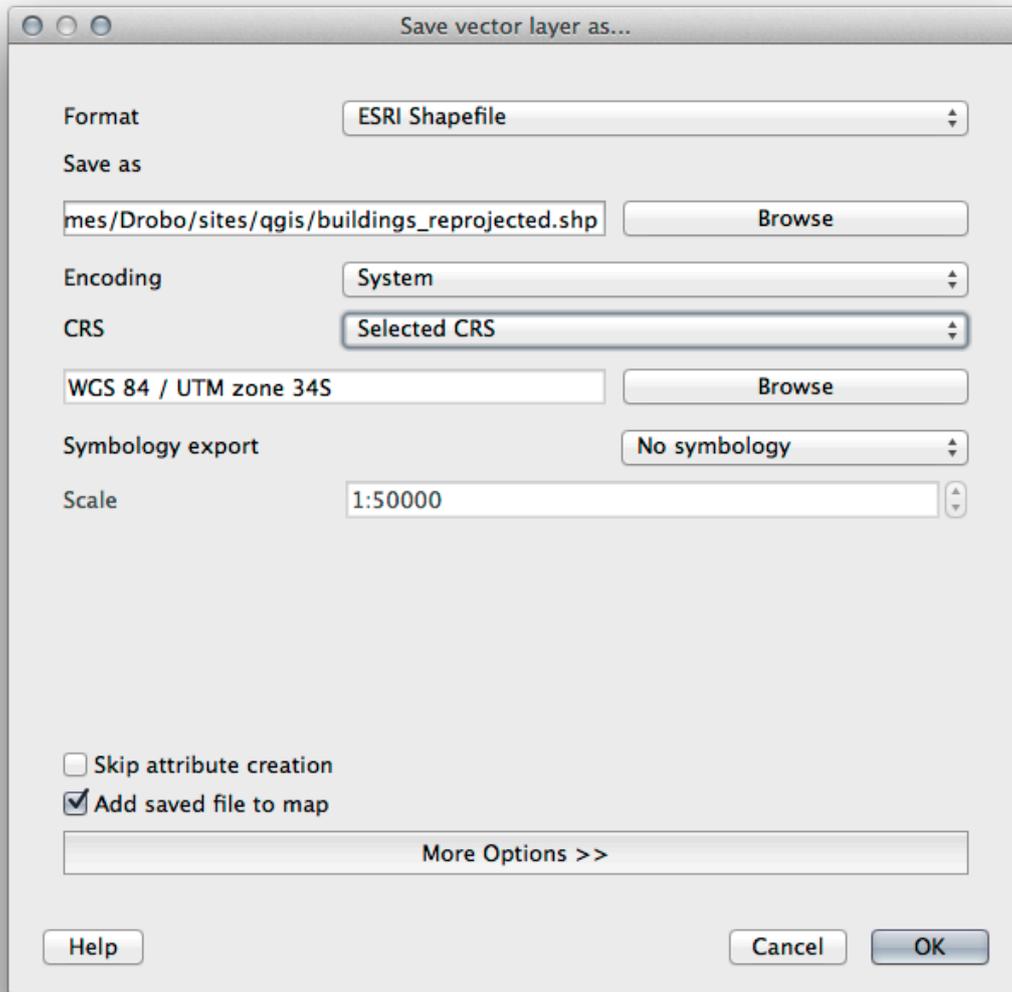
- あなたの通常のマップ（Swellendam データを含む）を開きます。
- Open the attribute table for the *buildings* layer.
- AREA フィールドが見えるまで右側にスクロールします。

エリアはすべて非常に小さく、注意してください。おそらくゼロ。これらの領域は、度で与えられているので、これはです - データが投影座標系ではありません。平方メートルでファームの面積を計算するために、データは、同様に平方メートルでなければなりません。だから、我々はそれを再投影する必要があります。

しかし、それだけで再投影「その場で」を使用する助けにはなりません。「オンザフライ」が言うん - それは、彼らが地図に表示されてそれだけでレイヤーを再投影、データは変更されません。真のデータ自体を再投影するには、新しい投影を使用して新しいファイルにエクスポートする必要があります。

- レイヤー *list* 中で:guilabel:*buildings* レイヤーを右クリックします。
- 表示されるメニューで:guilabel:名前を付けて保存... を選択します。ベクトルレイヤーを... として保存ダイアログが表示されます。
- Click on the *Browse* button next to the *Save as* field.
- *exercise_data/* に移動し、新しいレイヤーの名前を:kbd:*buildings_reprojected.shp* と指定します。
- エンコーディング はそのままにしておきます。
- レイヤー CRS‘ドロップダウンの値を:guilabel:‘選択された CRS に変更します。
- ドロップダウン下の *ブラウズ* ボタンをクリックします。
- CRS セレクタ ダイアログが表示されるでしょう。
- :guilabel:‘フィルター‘フィールドで、:kbd:‘34S‘を検索します。
- リストから :guilabel:‘WGS 84 / UTM zone 34S‘を選択します。
- Leave the *Symbology export* unchanged.

The *Save vector layer as...* dialog now looks like this:

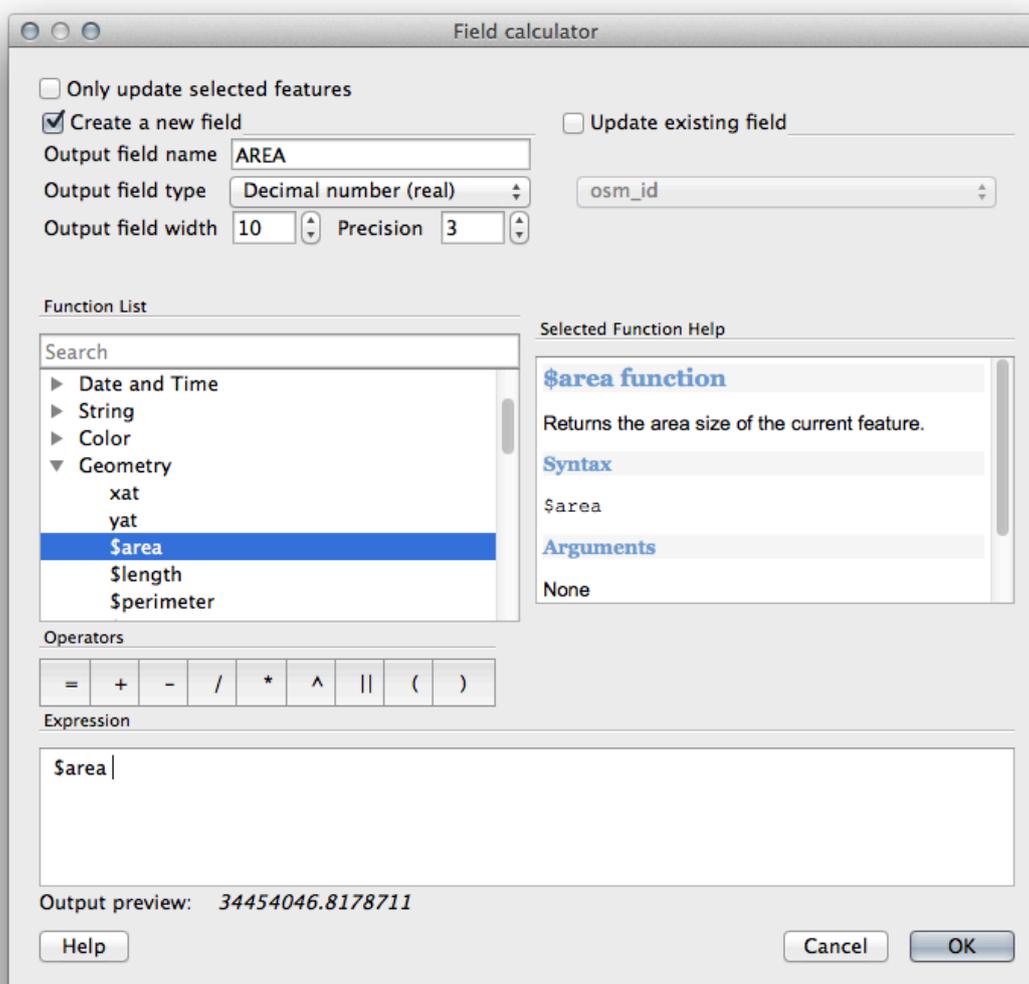


- Click *OK*.

- 新しい地図を起動し、作成したばかりの再投影レイヤーをロードします。

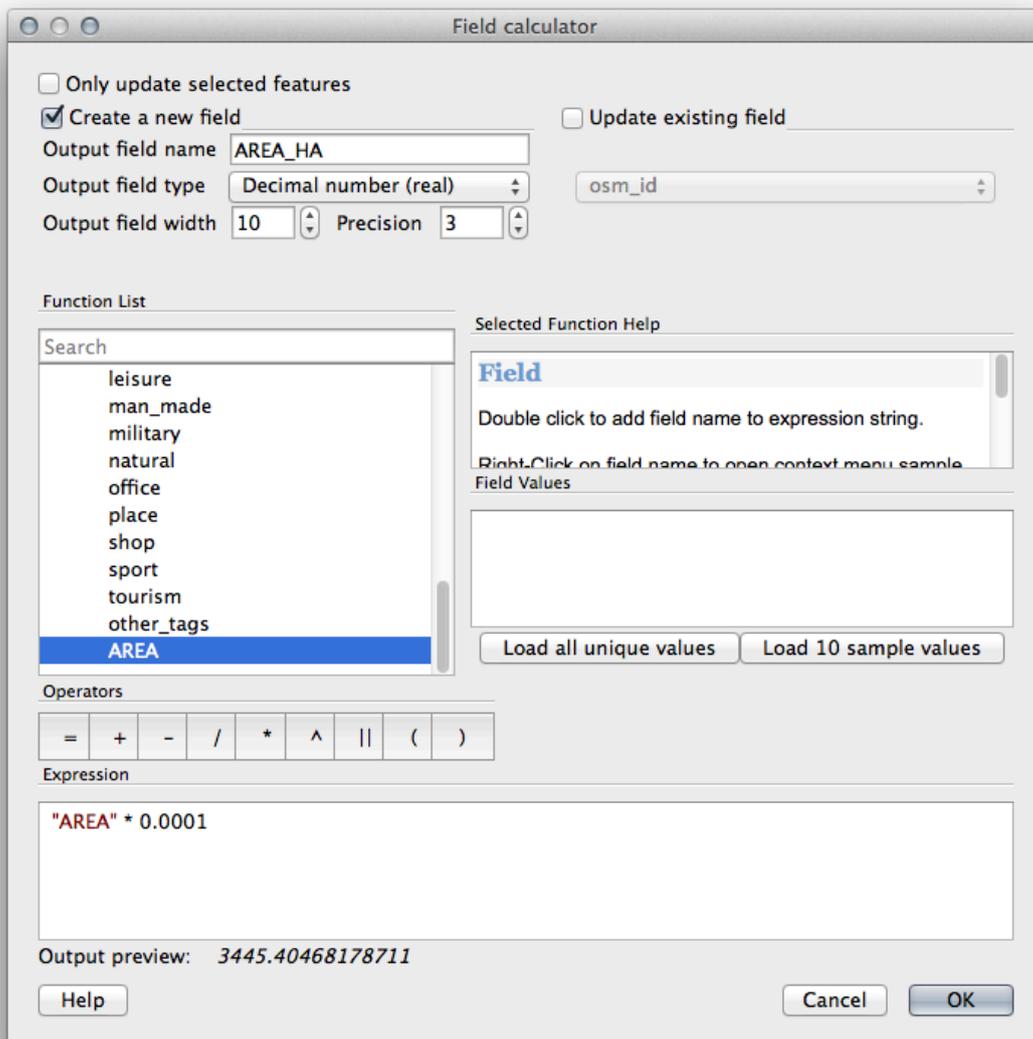
領域を計算する方法を思い出すには、以前の:guilabel:Classification についてのレッスンを参照してください。

- 前と同じ式を実行することによって面積 フィールドを更新（または追加）します：



平方メートルで各建物の大きさを持つ:kbd:面積 フィールドが追加されます

- 測定の他の単位で面積を計算するには、第二列を作成するために:kbd:‘AREA‘フィールド例えばヘクタールのために、使用：

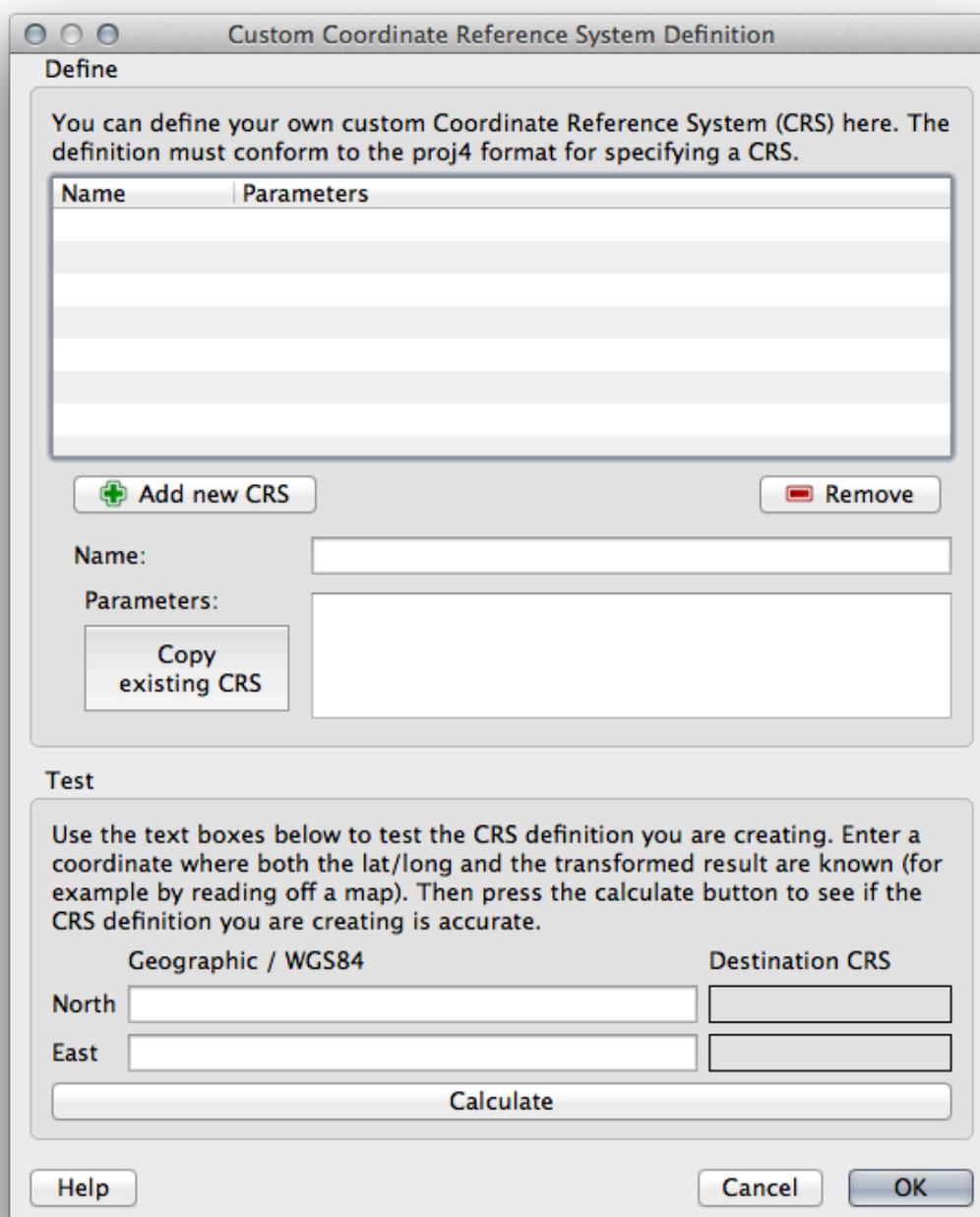


属性テーブルに新規の値を見てください。人々が実際にはない度に、メートルで、建物の大きさを引用し、これは、はるかに便利です。それは、必要に応じて層の空間特性に依存している地域、距離、および他の値を計算する前に、あなたのデータを再投影することをお勧めします理由です。

7.1.4 Follow Along: 独自の投影法の作成

デフォルトでは、QGIS に含まれたものだけよりも多くの突起があります。また、あなた自身の投影を作成することができます。

- マップを新規に開始します。
- Load the world/oceans.shp dataset.
- *Settings* → *Custom CRS...* に行くと、このダイアログが表示されるでしょう:



- *Add new CRS* ボタンをクリックして新しい投影法を作成します。

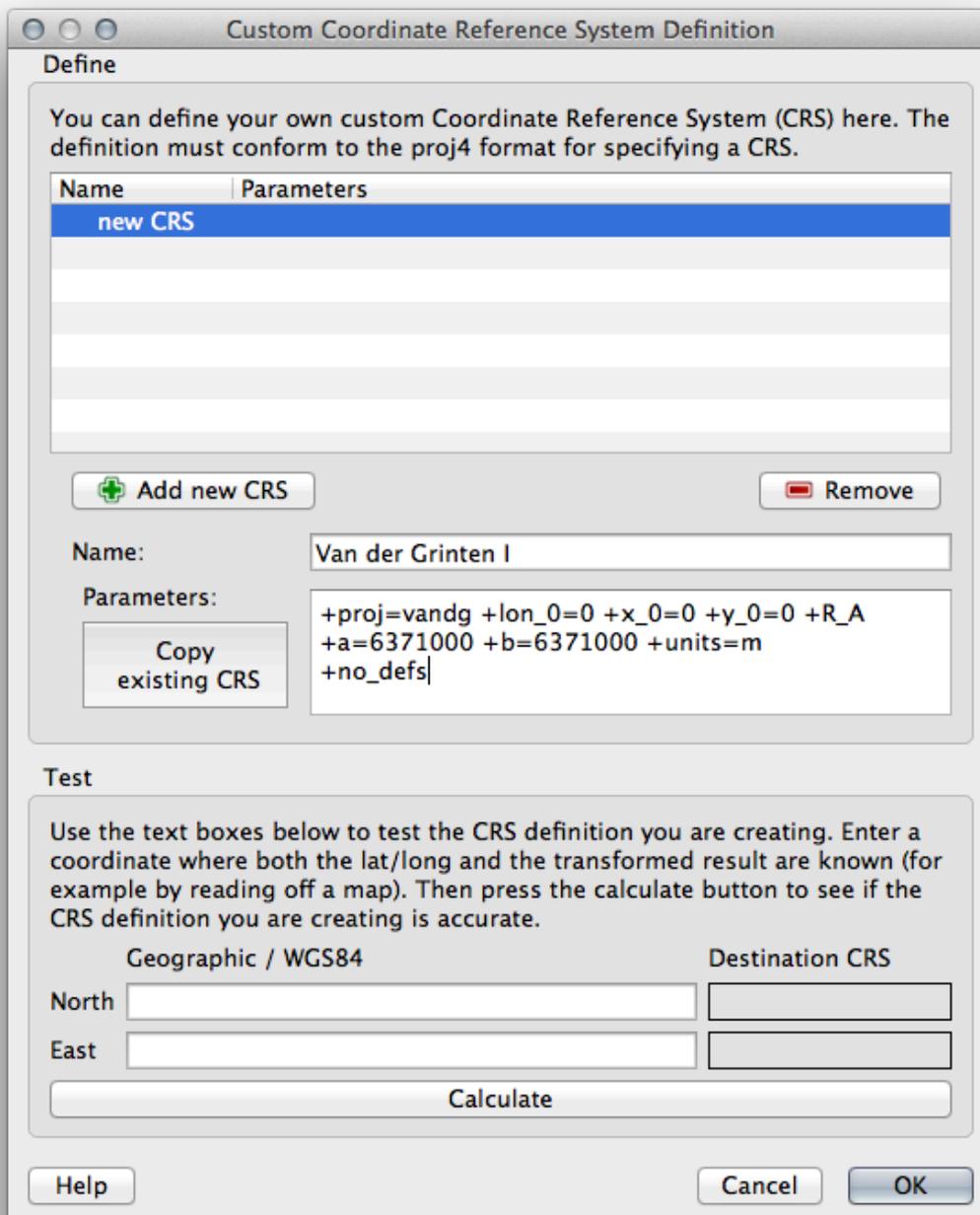
使用すると興味深い投影法は:kbd:Van der Grinten I と呼ばれるものです。

- 名称 フィールドでその名称を入力します。

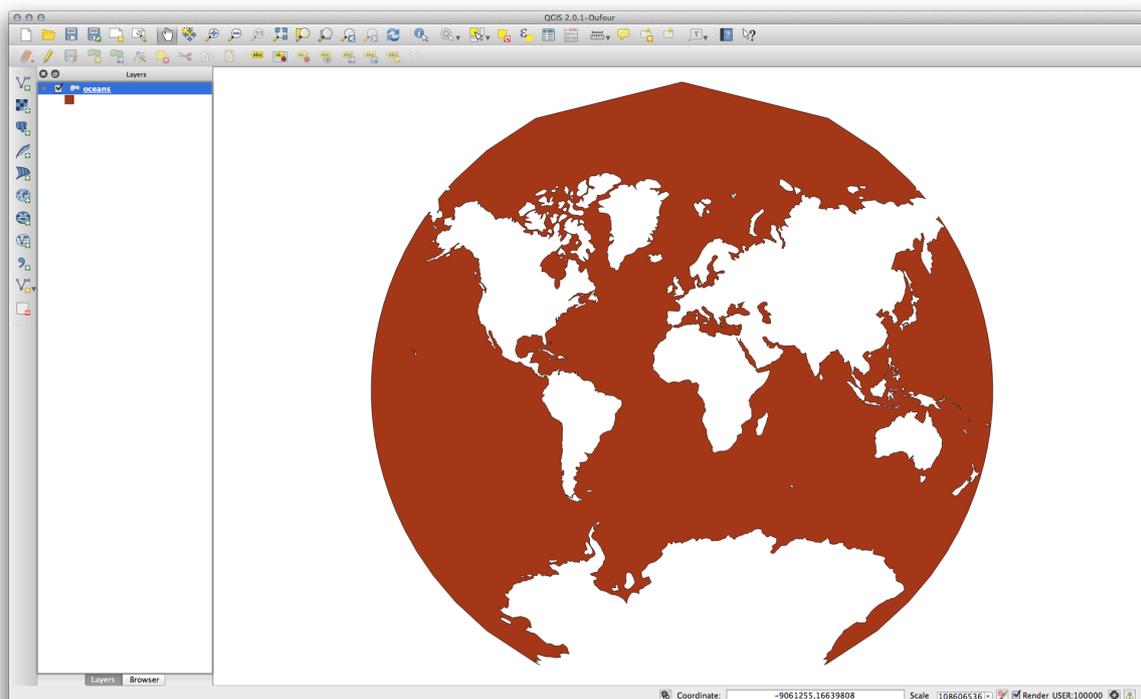
他のほとんどの突起がそうであるように、この突起は、代わりに長方形の一方の円形フィールドに地球を表します。

- そのパラメータとして、次の文字列を使います：

```
+proj=vandg +lon_0=0 +x_0=0 +y_0=0 +R_A +a=6371000 +b=6371000 +units=m
+no_defs
```



- Click *OK*.
- “オンザフライ” 再投影を有効にします。
- 新しく定義された投影を選択してください (: guilabel : ‘Filter’フィールドにその名前を検索)。
- この投影法を適用するため、したがってマップは再投影されるでしょう :



7.1.5 In Conclusion

異なる投影は、異なる目的のために有用です。正しい投影を選択することにより、あなたはマップ上の機能が正確に表現されていることを確認することができます。

7.1.6 Further Reading

このレッスンの*Advanced* セクションの材料は ‘この記事 <<http://tinyurl.com/75b92np>>’ からとりました。座標系についてさらに詳細な情報は ‘こちら<http://linfiniti.com/dla/worksheets/7_CRS.pdf>’ で入手できます。

7.1.7 What's Next?

次のレッスンでは、QGIS の様々なベクタ分析ツールを使ってベクタデータの分析をする方法について学習します。

7.2 Lesson: ベクタ分析

ベクトルデータは、さまざまな地物が空間内でどのように相互作用するかを明らかにするために分析できます。GIS には多くの異なる分析関連機能があるので、それらすべてを見ていくことはしません。むしろ、問題を提起し、QGIS が提供するツールを使用してそれを解決しようとするでしょう。

****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。

7.2.1 GIS プロセス

始める前に、任意の GIS の問題を解決するために使用できるプロセスの簡単な概要を与えることが有用でしょう。それを行う方法は次のとおりです。

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

7.2.2 問題

解決する問題を決定することからプロセスを開始してみましょう。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. Swellendam である必要がある。
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。

7.2.3 データ

これらの疑問に答えるため、次のデータが必要になるでしょう：

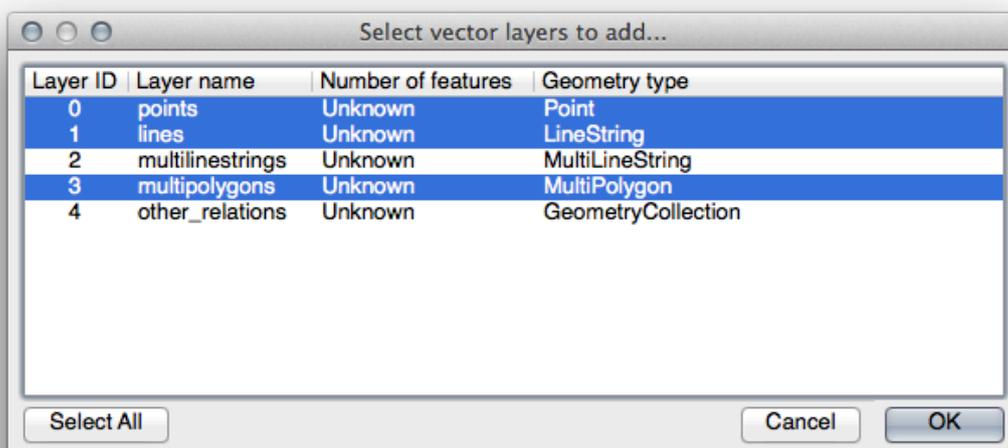
1. このエリアの住宅地属性 (建物)。
2. 街とその周辺の道路
3. 学校とレストランの位置
4. 建物のサイズ

このデータはすべて OSM を通して利用可能であり、このマニュアル全体で使用されているデータセットがこのレッスンのためにも使用できることがわかるはずです。しかし、完全なデータを持っていることを確認するために、QGIS 内蔵の OSM のダウンロードツールを使って OSM からデータを再ダウンロードします。

ノート：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。

7.2.4 Follow Along: プロジェクトの開始

- 新規の QGIS プロジェクトの開始
- 選択した地域のデータをダウンロードするには、ベクトル-> *OpenStreetMap* メニューで見つかる OpenStreetMap データダウンロードツールを使用します。
- Save the data as `osm_data.osm` in your `exercise_data` folder.
- *osm* 形式ベクトルデータのタイプであることに留意されたいです。通常通り、このデータをベクトルレイヤーとして追加します レイヤー ->ベクトルレイヤーを追加...、ちょうどダウンロードした新しい `osm_data.osm` ファイルを参照します。ファイル形式としてすべてのファイルを表示 を選択する必要があります。
- `osm_data.osm` を選択し 開く をクリックします。
- 表示されたダイアログで、`other_relations` と `multilinestrings` レイヤー を除く すべてのレイヤを選択します：



これで、あなたのマップに分断されたレイヤとして OSM データをインポートできます。

先ほど OSM からダウンロードされたデータは、前のレッスンから知っているように、地理座標系、WGS84、であり、緯度と経度の座標を使用します。メーターで距離を計算するためには投影座標系で作業する必要があります。プロジェクトの座標系をデータに適した CRS に設定することから開始してください。 Swellendam の場合には WGS 84 / UTM ゾーン 34S です：

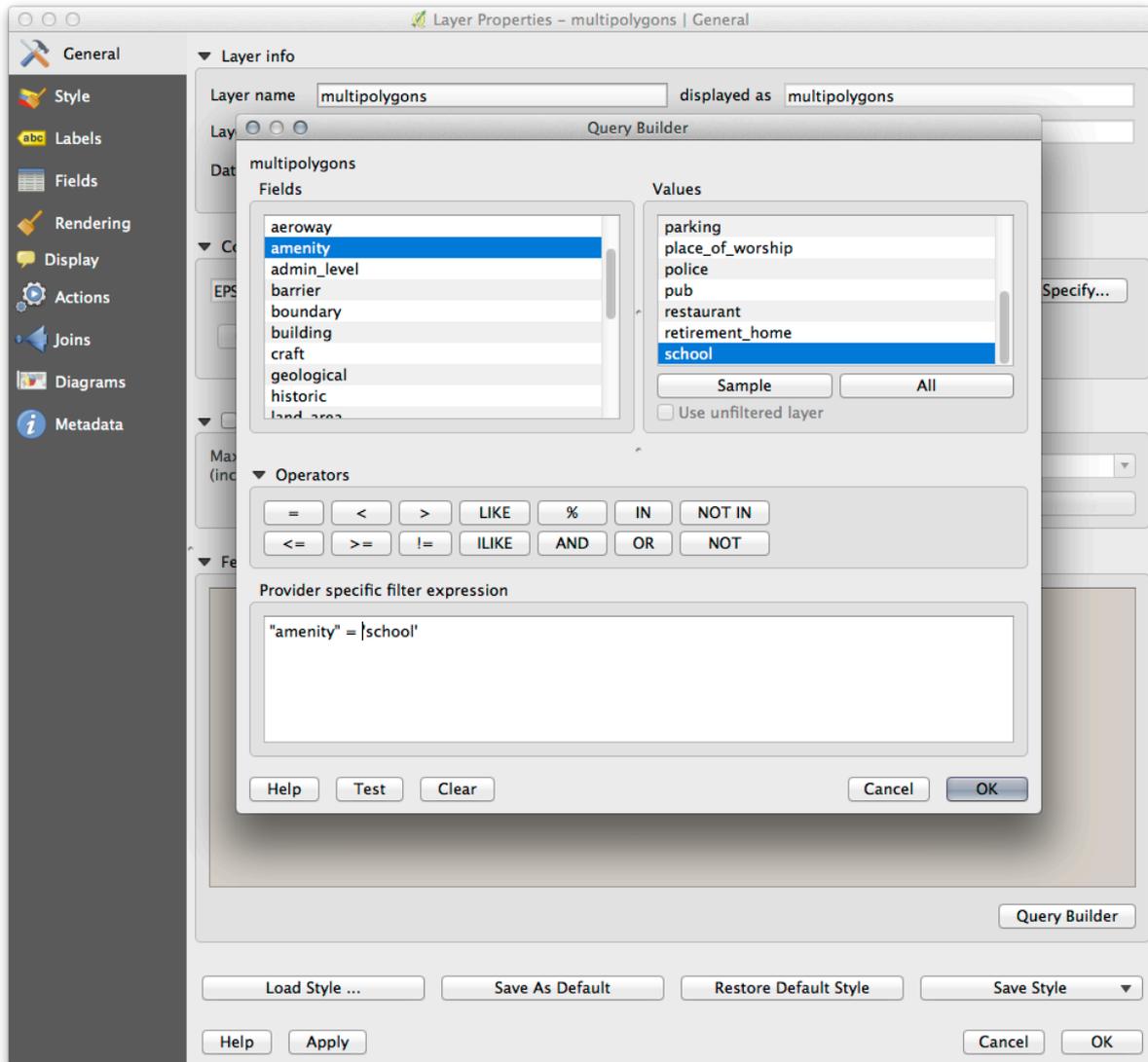
- プロジェクト Properties `ダイアログを開き、:guilabel: CRS`を選択して、:guilabel: `WGS 84 / UTM ゾーン 34S`を見つけるためにリストをフィルタをかけます。
- Click OK.

ここで、OSM データセットから必要な情報を抽出する必要があります。地域のすべての住宅、学校、レストラン、道路を表すレイヤーで終わる必要があります。その情報は *multipolygons* レイヤーの中にあり、その属性 *Table* 内にある情報を使用して抽出できます。 *schools* レイヤーで始めましょう

- レイヤーリスト 中の:guilabel:マルチポリゴン レイヤーを右クリックし、レイヤープロパティ を開きます。
- 一般情報 メニューに進みます。
- 地物 *subset* の下で [クエリビルダ] ボタンをクリックし クエリビルダ ダイアログを開きます。
- このダイアログの左側にある:guilabel:フィールド リストで、フィールド:kbd:amenity が表示されるまで。
- それを 1 回だけクリックします。
- 値 リストで:guilabel:すべて ボタンをクリックします:

さて我々は QGIS に:kbd:amenity が :kbd:`school`に等しいポリゴンであることのみ示す必要があります。

- フィールド リストの amenity をダブルクリックします。
- 以下の プロバイダ固有のフィルタ式 フィールドで何が起るか見ます：



語 "amenity" が表示されました。クエリの残りを作成するには:

- (演算子 'から') :*gui*label:'= ボタンをクリックします。
- 値 リストで school の値をダブルクリックします。
- OK を 2 回クリックします。

これは、唯一のお住まいの地域の学校を表示するために OSM の multipolygons レイヤーをフィルタリングします。今いずれかを行うことができます

- フィルターされた OSM レイヤーの名前を schools と変更し、 osm_data.osm から multipolygons レイヤーを再インポートする、または
- フィルタされたレイヤーを複製し、コピーの名前を変更し、クエリビルダをクリアし、クエリビルダで新しいクエリを作成します。

7.2.5 Try Yourself OSM からの必要なレイヤーの抽出

上記の技術を使用して、以下のレイヤーを作成するために、OSM からの残りのデータを抽出するクエリビルダツールを使用します。

- roads (OSM の lines レイヤー由来)

- restaurants (OSM の multipolygons レイヤ由来)
- houses (OSM の multipolygons レイヤ由来)

前のレッスンで作成した roads.shp を再利用するのが望ましいです。

結果の確認

- *exercise_data* 下に analysis.qgs として地図を保存します (この地図は、将来のモジュールで使用されます)。
- お使いのオペレーティングシステムのファイルマネージャで、*exercise_data* 下に新しいフォルダを作成してそれを residential_development と名付けます。ここが分析関数の結果になるデータセットを保存する場所です。

7.2.6 Try Yourself 主要道路の検索

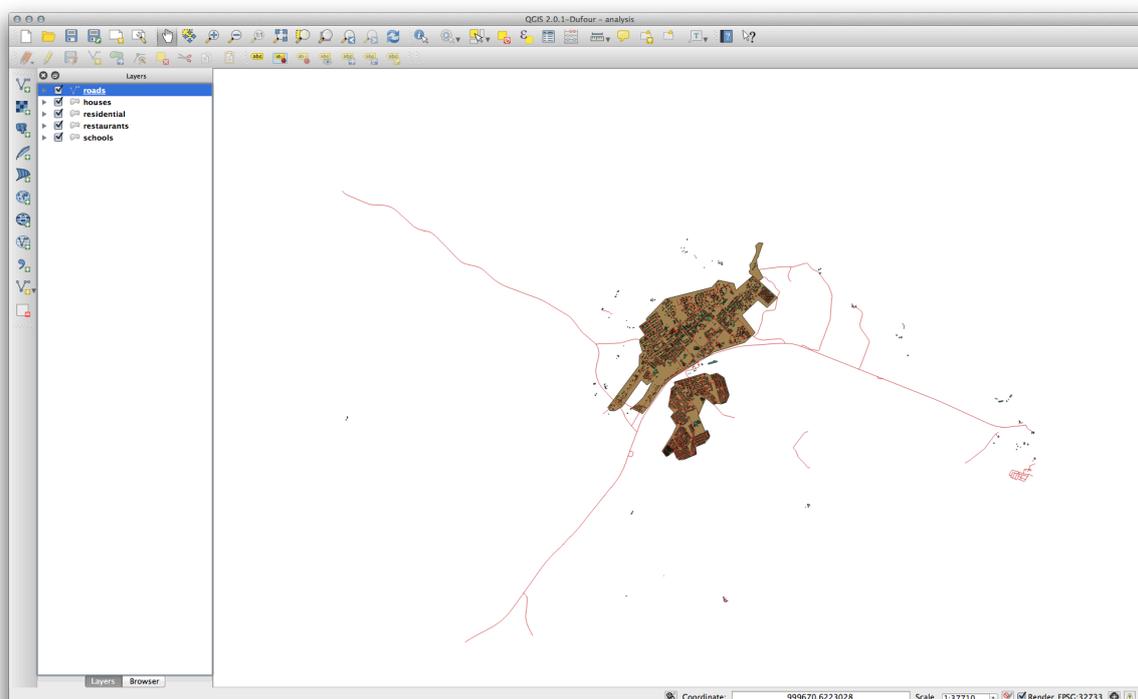
OSM のデータセット内の道路の一部は、未分類、tracks、PATH、footway とリストされています。道路データセットからこれらを除外したいです。

- roads レイヤー:kbd: のためのクエリ Builder 開き:guiabel: Clear クリックし、次のクエリを構築します:

```
"highway" != 'NULL' AND "highway" != 'unclassified' AND "highway" != 'track' AND "highway" != 'path' AND "highway" != 'footway'
```

上記の方法、値をダブルクリックしてボタンをクリックする、を使用するか、上記のコマンドをコピーして貼り付けできます。

直ちにマップから多くの道路を減らすべきです:

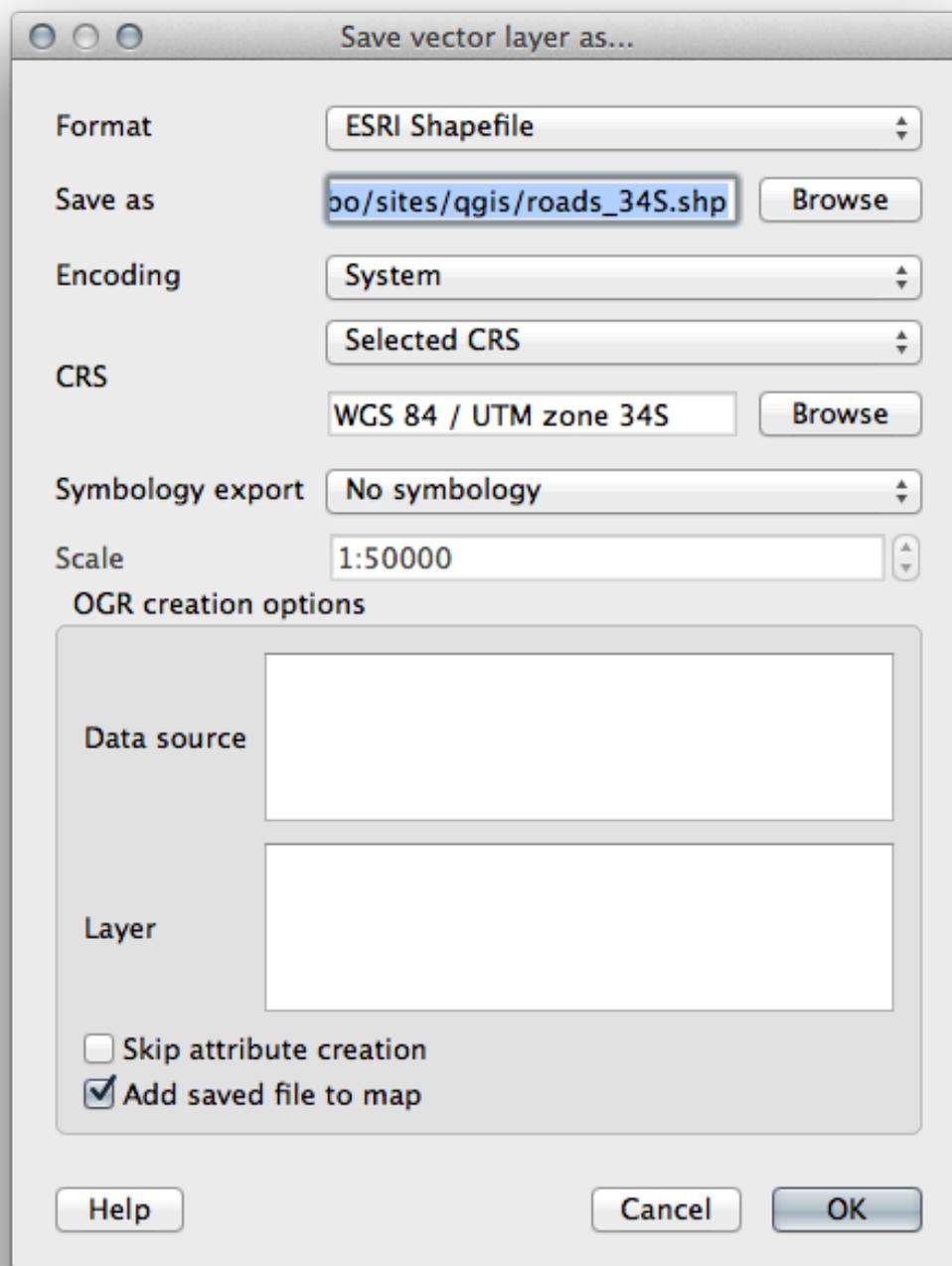


7.2.7 Try Yourself レイヤ CRS の変換

レイヤー内で距離を測定しようとしているので、レイヤの CRS を変更する必要があります。これを行うために、私たちは順番に各レイヤーを選択する必要があり、地図にその新しいレイヤーをインポートし、新しい投影で新しいシェープファイルにレイヤーを保存します。

ノート: :guilabel:この例では、使用している *WGS 84/UTM* ゾーン *34S* CRS をしていますが、お住まいの地域のためのより適切な *UTM* CRS を使用できます。

- Right click the `roads` layer in the `Layers` panel.
- `Save as...` をクリックします。
- 名前をつけてベクトルを保存 ダイアログで、以下の設定を選択し、`Ok` をクリックしてください (保存したファイルを地図に追加を選択したことを確かめて):



新しいシェープファイルが作成され、地図に結果のレイヤとして追加されます。

ノート: If you don't have activated *Enable 'on the fly' CRS transformation* or the *Automatically enable 'on the fly' reprojection if layers have different CRS settings* (see previous lesson), you might not be able to see the new layers you just added to the map. In this case, you can focus the map on any of the layers by right click on any layer and click *Zoom to layer extent*, or just enable any of the mentioned 'on the fly' options.

- 古い roads レイヤを削除します。

「_34S」元の名前に追加され、古い層のそれぞれを削除して新しいシェープファイルやレイヤーを作成し、レイヤごとにこのプロセスを繰り返します。

それぞれのレイヤーに対してプロセスが完了したら、いずれかのレイヤーの上で右クリックして、レイヤー範囲にズームをクリックして関心領域に地図を集中してください。

今、UTM 投影に OSM のデータを変換したので、計算を開始できます。

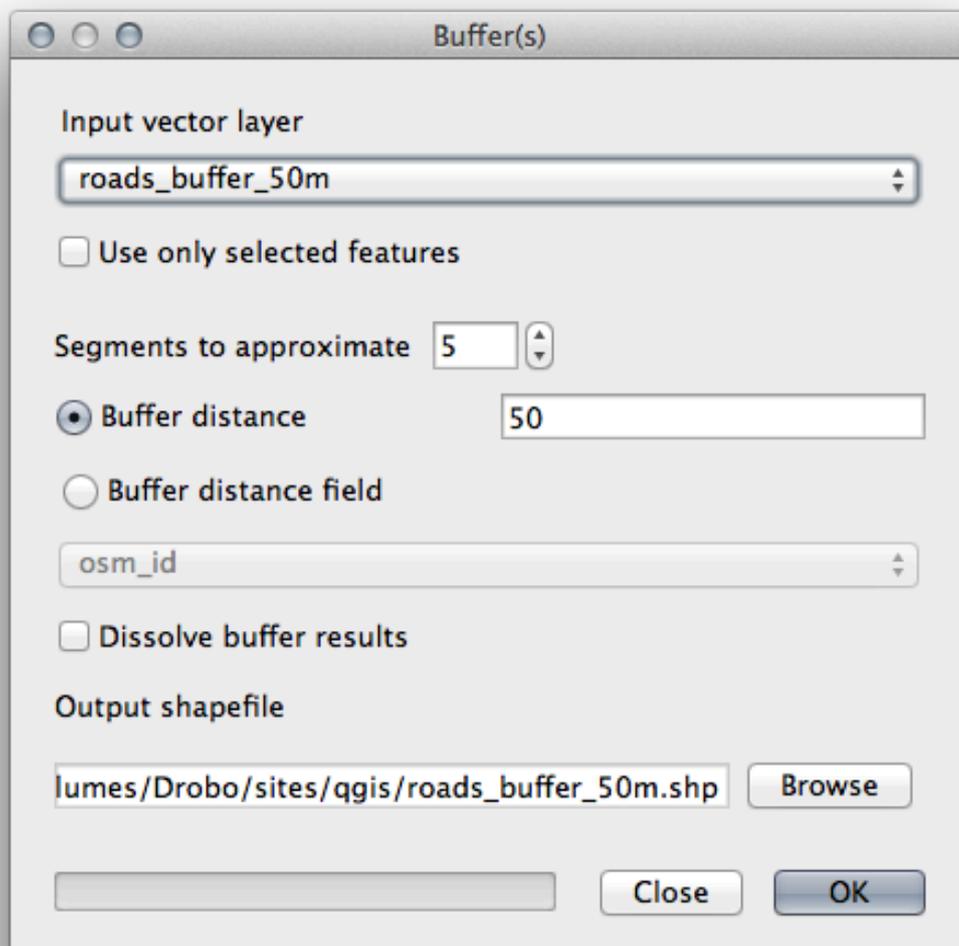
7.2.8 Follow Along: 問題の分析:学校と道路からの距離

QGIS はいかなるベクタからの距離を計算することができます。

- 作業中に地図を簡単にするために、`roads_34S` と `houses_34S` レイヤーだけ表示されていることを確認してください。
- Click on the *Vector* → *Geoprocessing Tools* → *Buffer(s)* tool:

新しいダイアログを表示します。

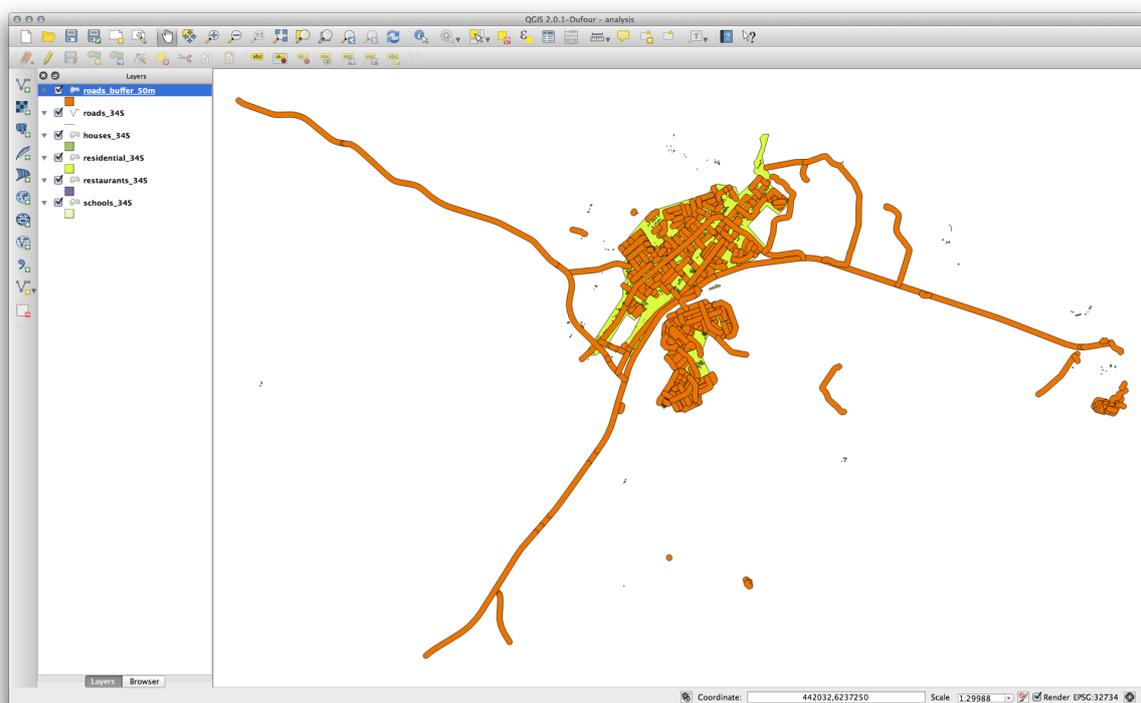
- このように設定します:



私達の入力データセットは、その基本的な計測単位としてメートルを使用する投影座標系であるため、:guilabel:‘バッファdistance‘はメートル単位です。投影されたデータを使用する必要があるのはこのためです。

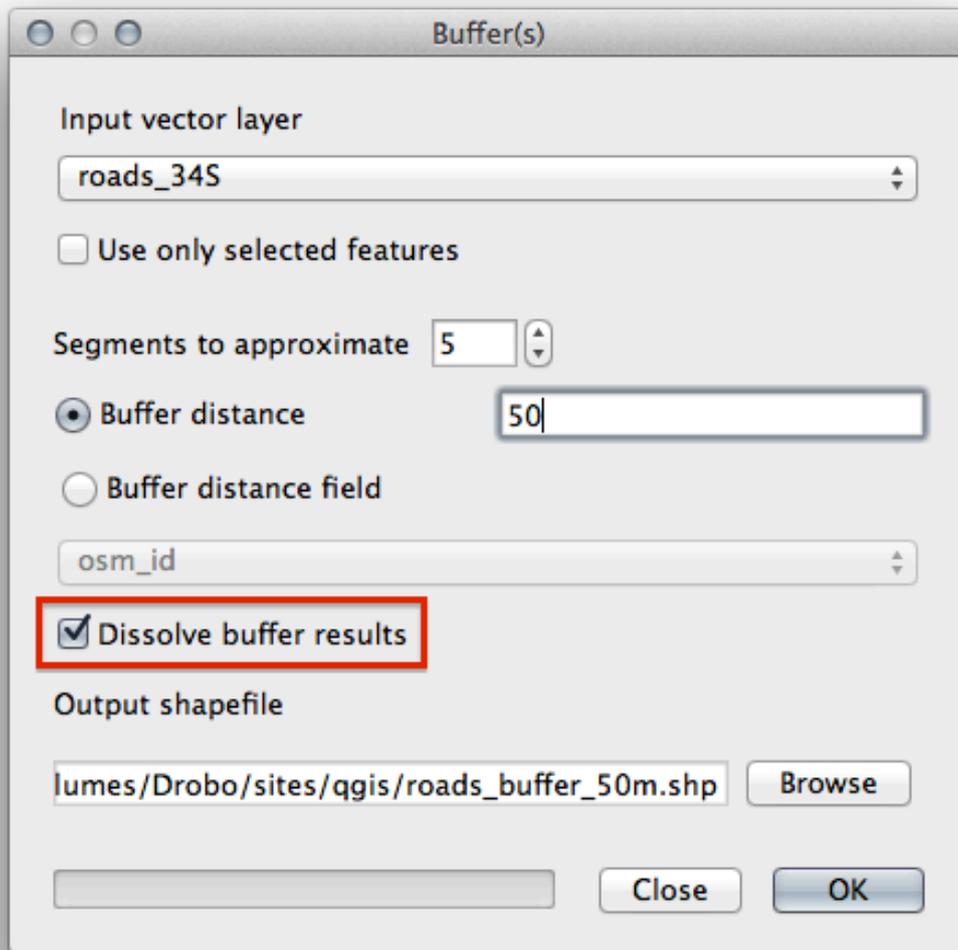
- exercise_data/residential_development/ roads_buffer_50m.shp 下に結果のレイヤーを roads_buffer_50m.shp として保存します。
- OK をクリックし、バッファを作成します。
- それは「TOC に新しいレイヤーを追加する」必要がある場合、それはあなたを要求すると、クリックしてください: guilabel: ‘Yes‘を。(: guilabel: ‘レイヤー list‘ 「TOC」は、それが意味していることで、「目次」の略)
- Close the *Buffer(s)* dialog.

今すぐあなたのマップは次のようになります:



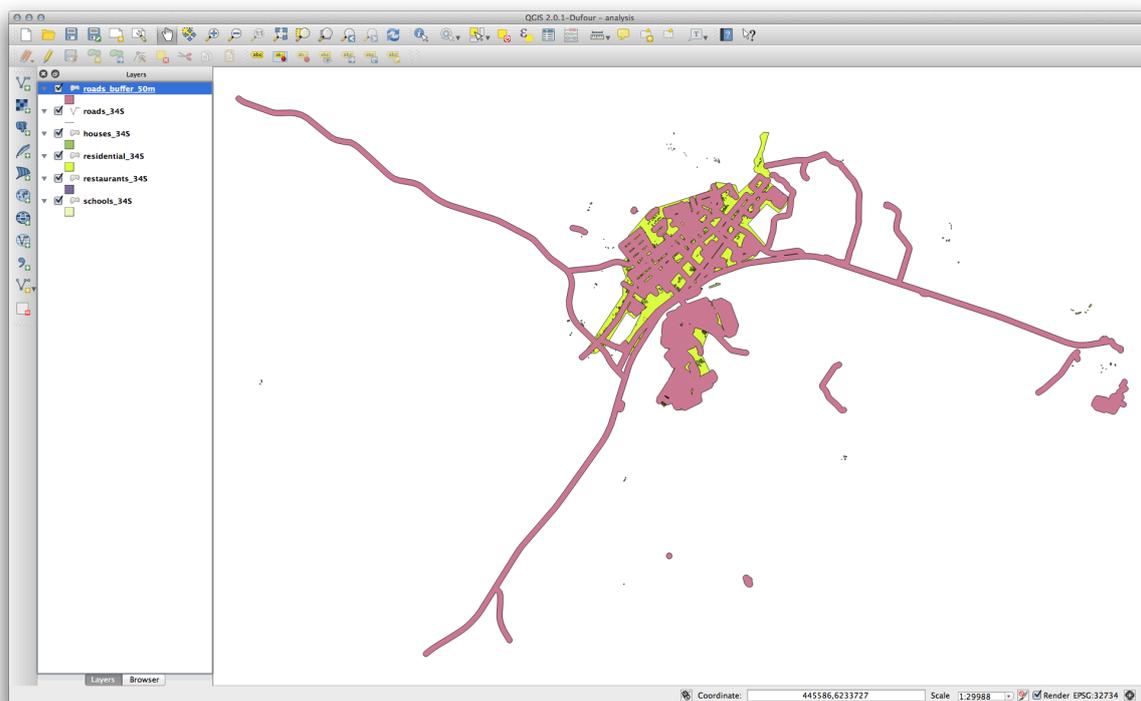
新しいレイヤーが Layers リストの最上にある場合は、おそらく地図の多くを不明瞭にしますが、これは道の 50 メートルの範囲内にある、地域内のすべての領域を提供しています。

ただし、すべての個々の道路に対応し、当社のバッファ内の別個の領域があることに気づくでしょう。この問題を取り除くために、層を除去し、ここに示した設定を使用してバッファを再作成します。



- バッファのディゾルブ結果 ボックスを今チェックしていることに注目ください。
- 前と同じ名前の下に出力を保存します (古いものを上書きする許可を求めてきたときは はい をクリック)。
- OK をクリックし バッファ ダイアログを再度閉じます。

レイヤリスト にいったんレイヤを追加すると、このように見えます:



今不要な下位区分はありません。

7.2.9 Try Yourself 学校からの距離

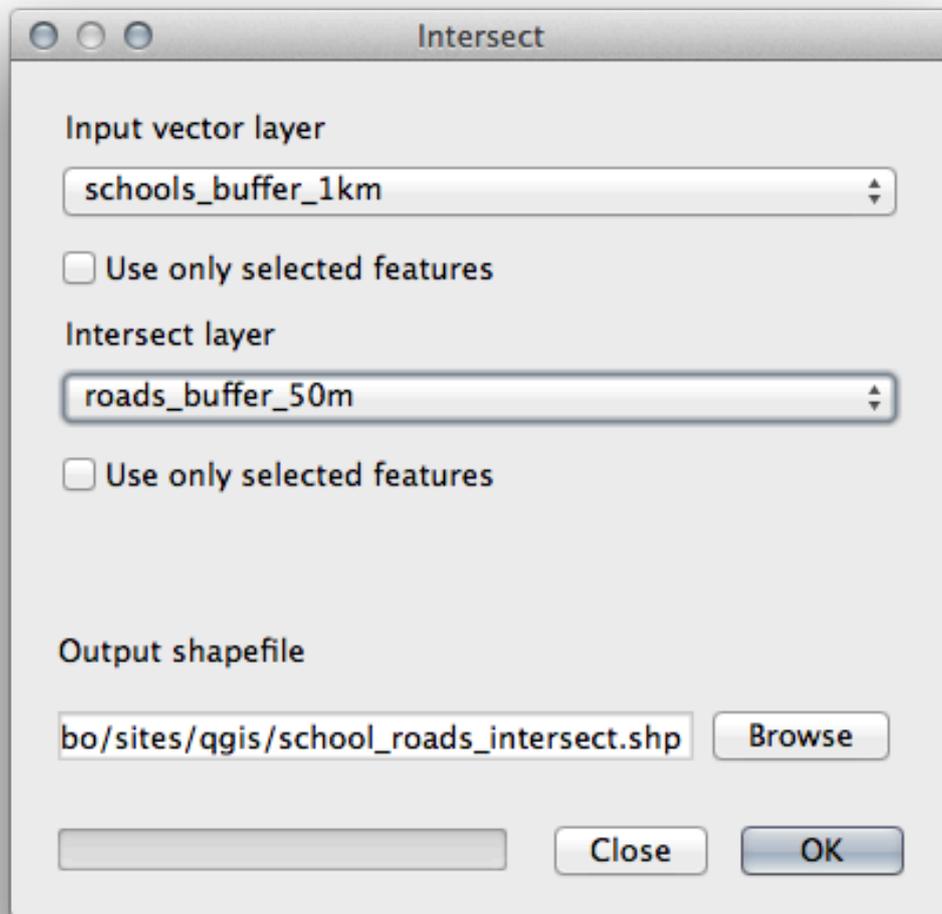
- 上記と同じアプローチを使用し、学校のためのバッファを作成します。

それは半径 1 km、そしていつものようにディレクトリの下に `schools_buffer_1km.shp` として保存する必要があります。

結果の確認

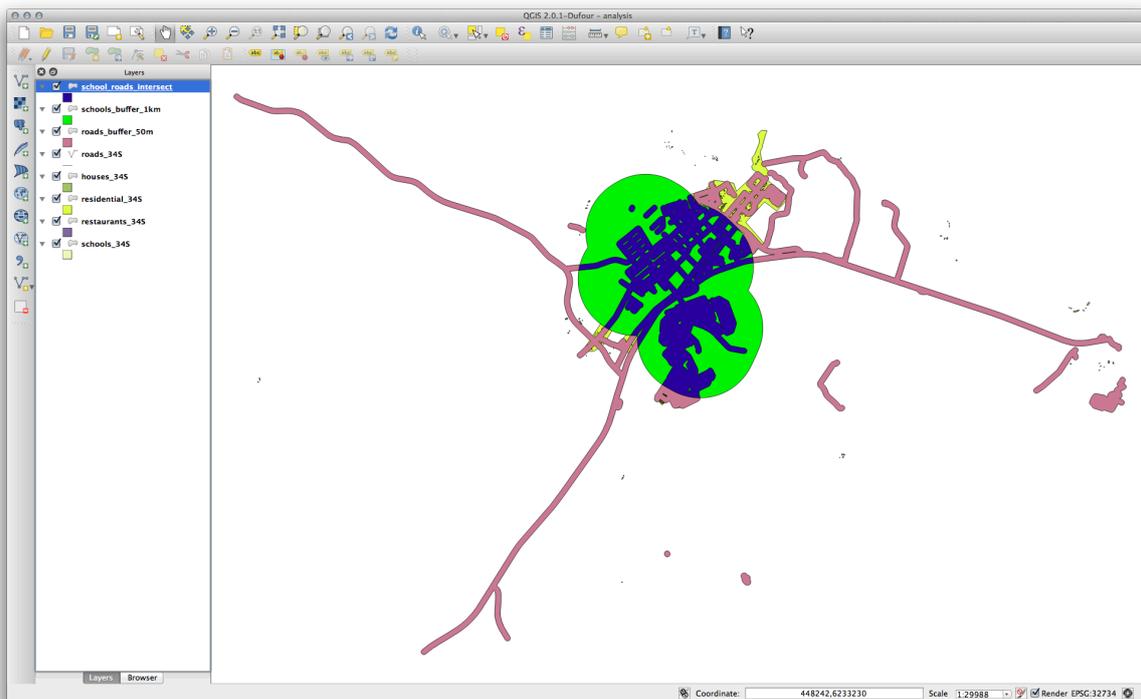
7.2.10 Follow Along: 重複エリア

今、道路が 50 メートル離れている領域を得ており、学校は 1 キロ以内にあります（直線距離で、道路でなく）。しかし明らかに、これらの基準の両方が満たされる領域を希望しているだけです。そのためには、*Intersect* ツールを使用する必要があります。ベクトル -> ジオプロセシングツール -> *Intersect* の下で見つける。このようにそれを設定します。

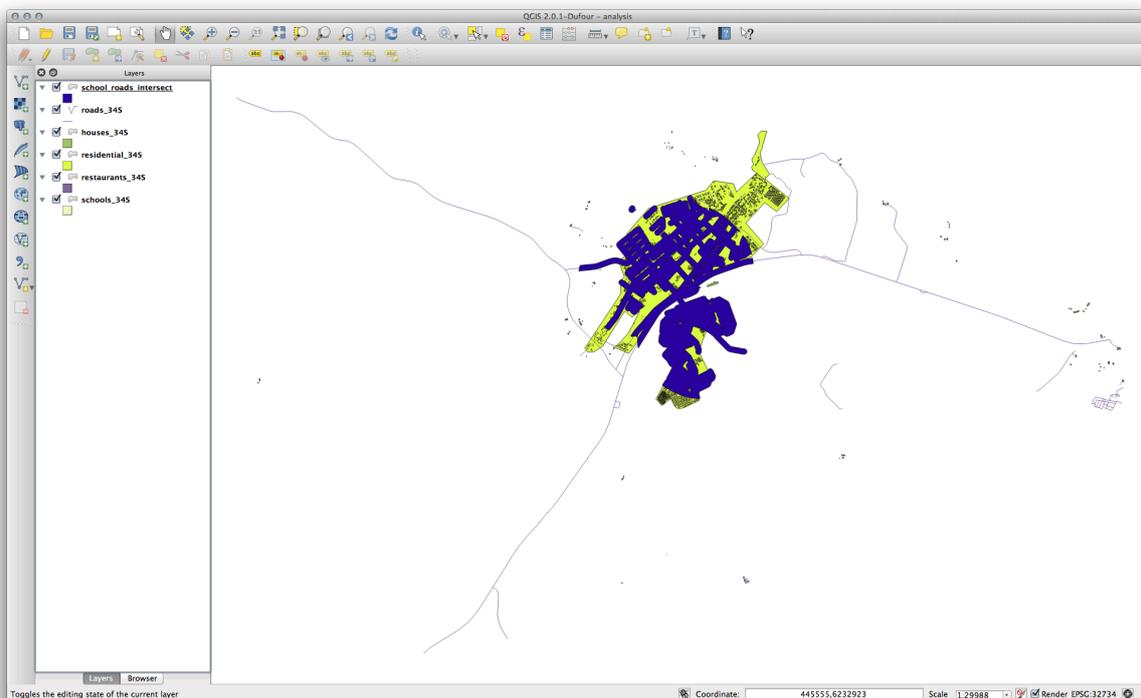


二つの入力層は、二つのバッファです。保存場所は、いつものようにあります。ファイル名は次のとおりです。 `road_school_buffers_intersect.shp`。それは次のようにセットアップしたら、クリックしてください `OK` とにレイヤを追加します プロンプトが表示されたら、レイヤー *list*。

両方の距離基準が一度に満たされている場所の下画像では、青色の領域が私たちを見ます！



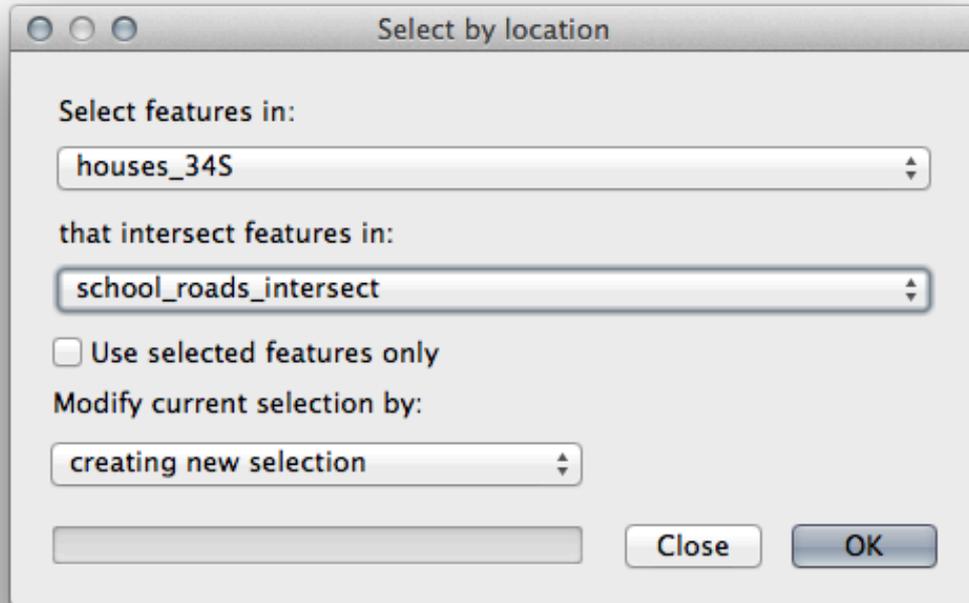
2つのバッファレイヤーを除去しそれらが重なる場所を示すものだけを維持しても良いです、それが最初の場所で本当に知りたいと思ったものですので：



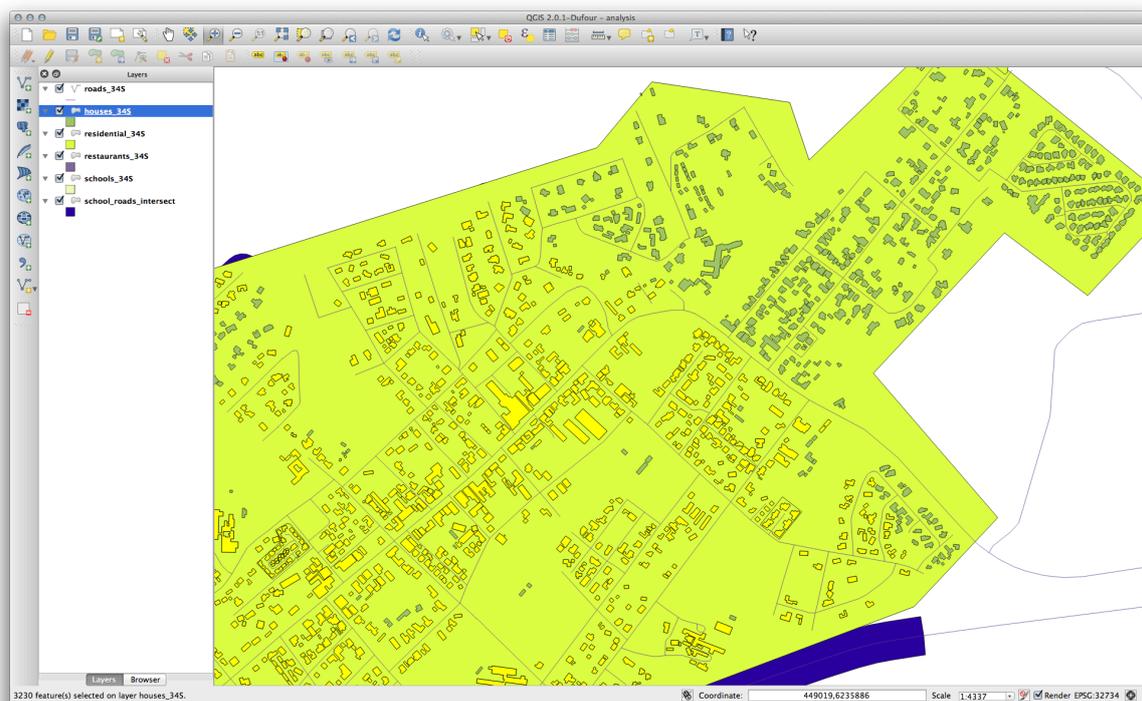
7.2.11 Follow Along: 建物の選択

これで建物がオーバーラップしなければならない領域が得られました。次は、その地域の建物を選択します。

- ベクトル -> 研究のツール -> *location* で選択 メニュー項目をクリックします。ダイアログが表示されます。
- このように設定します:

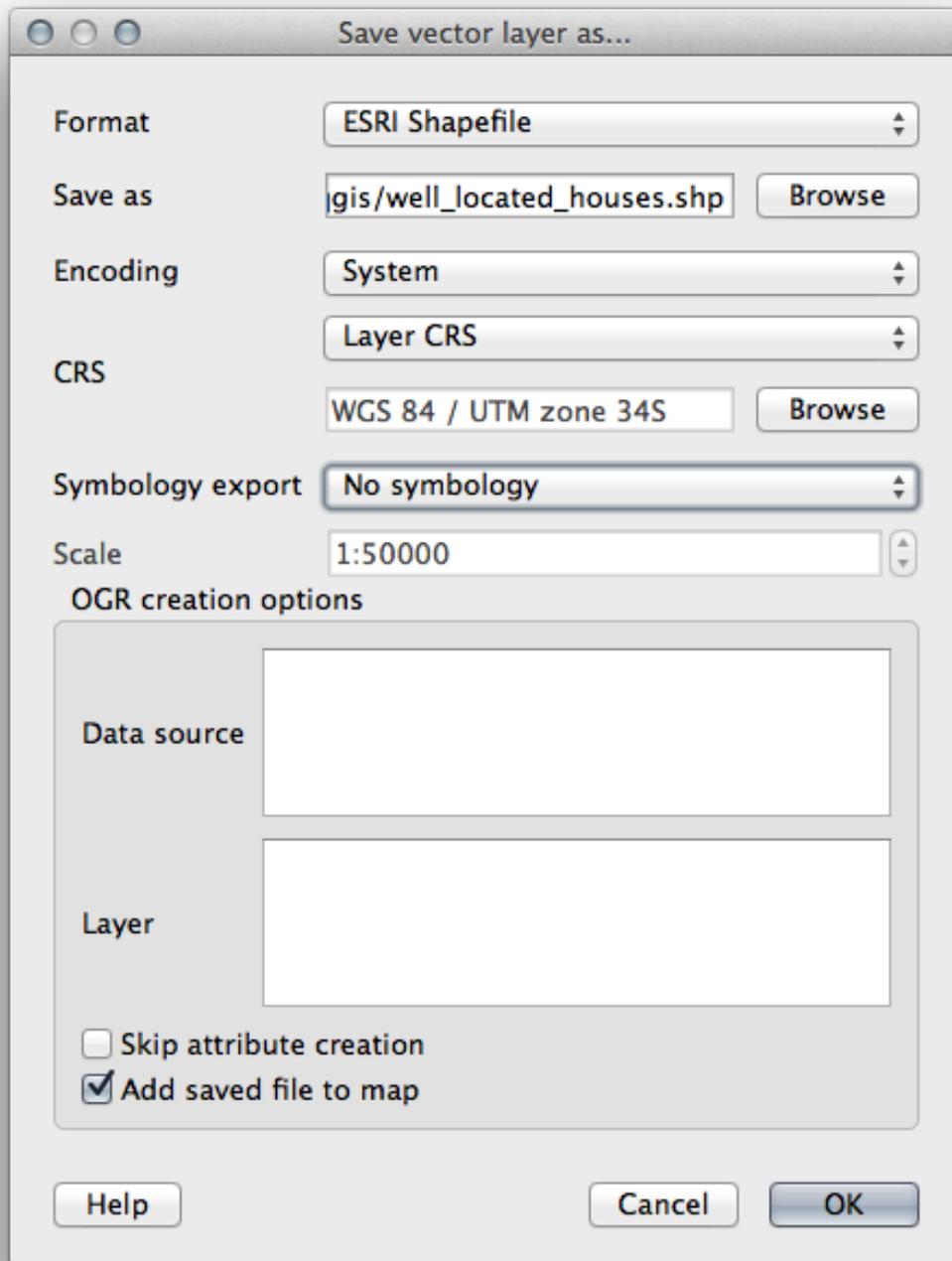


- *OK* をクリックし、閉じる をクリックします。
- おそらくあまり変更されていないように見えないと思われることでしょう。その場合は、レイヤーリストの一番下に *school_roads_intersect* レイヤーを移動し、次いでズームインします。



基準に一致して選択される建物は黄色でハイライトされる一方、緑色の建物は選択されないものです。今、選択された建物を新しいレイヤーとして保存できます。

- レイヤーリスト 中の *houses_34S* レイヤー上で右クリックします。
- 選択を別名保存... を選択します。
- ダイアログにてこのように設定します:



- ファイル名は `well_located_houses.shp` です。
- Click *OK*.

今、選択を別のレイヤーとしてしているし、`houses_34S` レイヤーを削除できます。

7.2.12 |TY|さらに、当社の建物をフィルター

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。

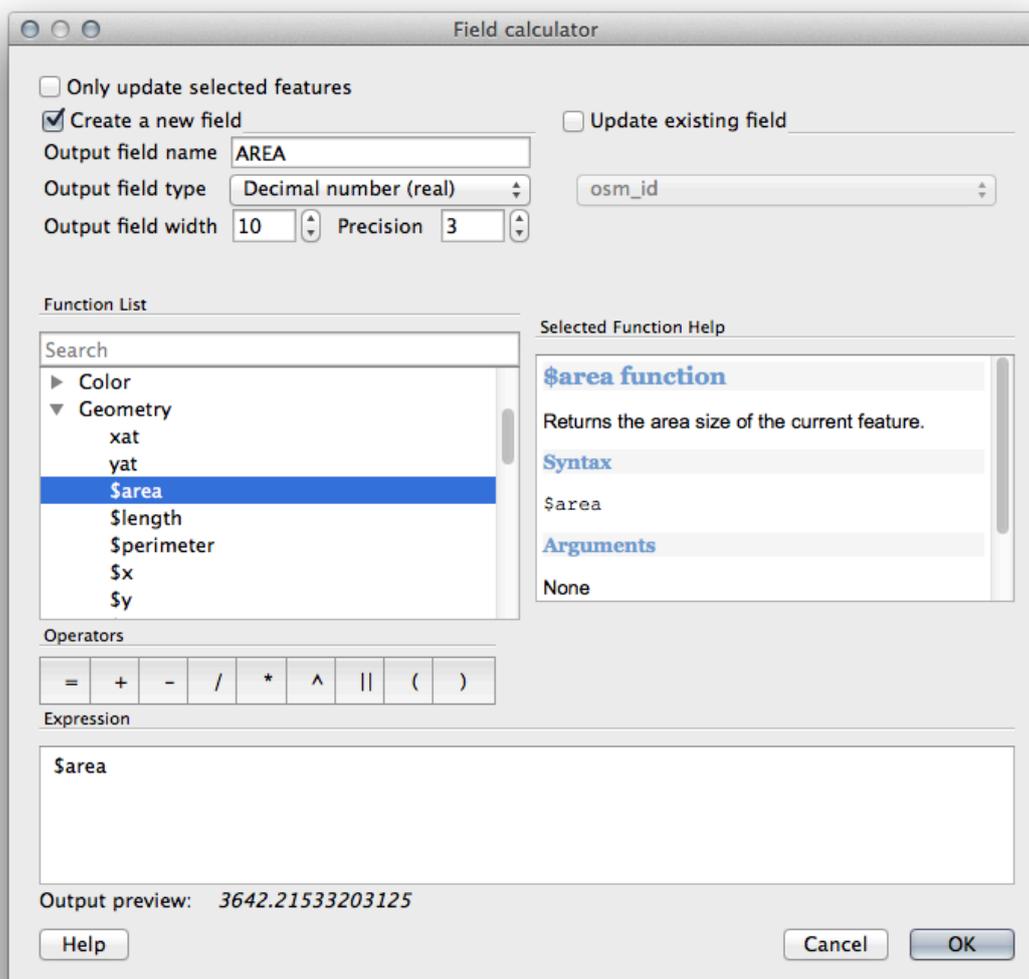
上記のプロセスを使用して `houses_restaurants_500m` という名前の新しいレイヤーを作成し、さらにこれで `well_located_houses` レイヤーをフィルタして、レストランの 500 メートル内にあるものだけ表示します。

結果の確認

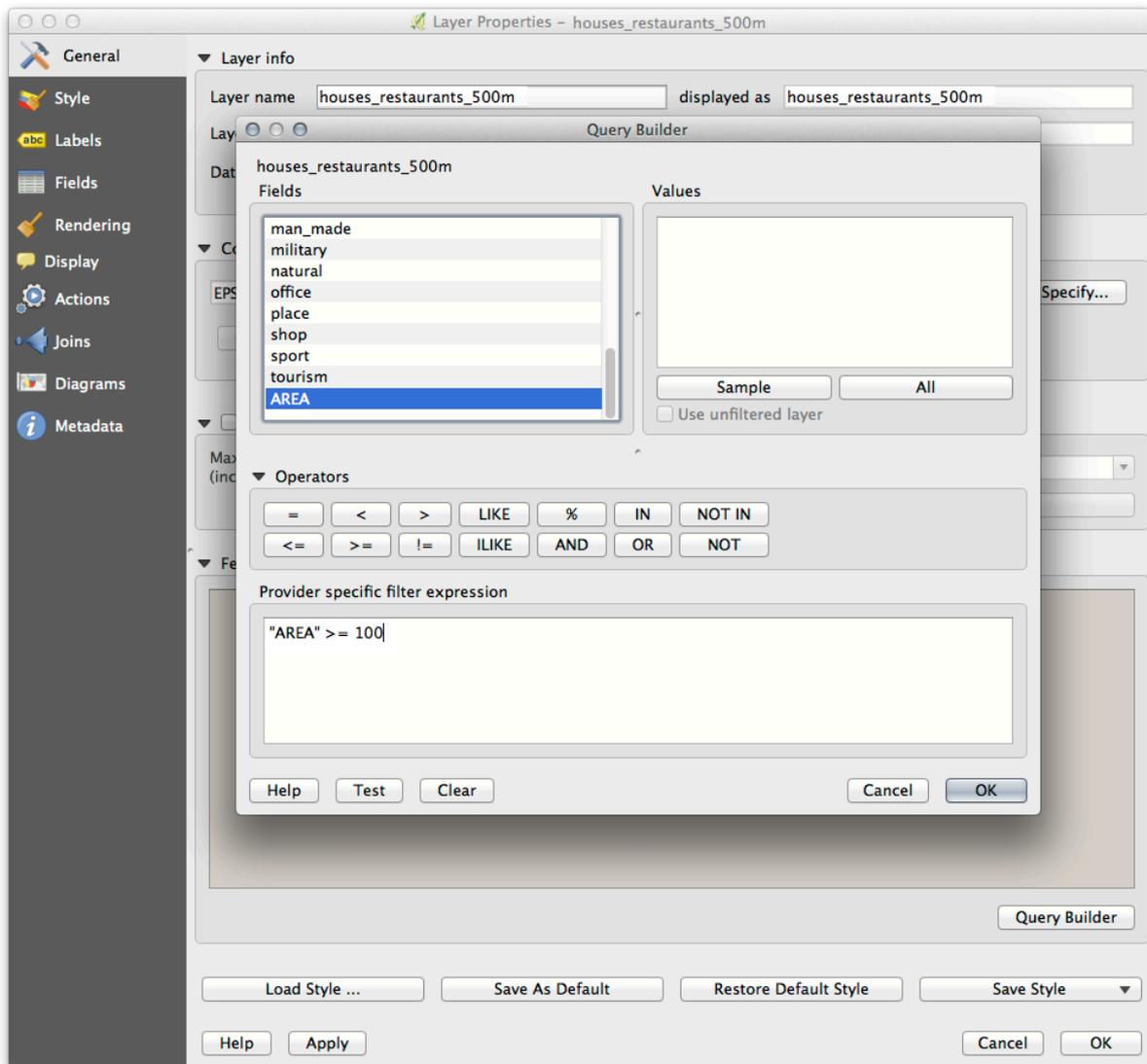
7.2.13 Follow Along: 正しいサイズの建物の選択

正しいサイズ（以上 100 平方メートル）である建物を参照するには、まず自分のサイズを計算する必要があります。

- `houses_restaurants_500m` レイヤーの属性テーブルを開きます。
- 編集モードにし、フィールド演算を開きます。
- このように設定します:



- リストで *AREA* を見つけることができない場合、このモジュールの前のレッスンで行ったように新しいフィールドを作成してみてください。
- Click *OK*.
- 属性テーブルの右側にスクロールします。今、*AREA* フィールドには *houses_restaurants_500m* レイヤーのすべての建物についてメートルでの面積があります。
- 編集を終了し、プロンプトが表示されたら、編集内容を保存するには、再度編集モードボタンをクリックします。
- このレッスンでは、以前のようにクエリを構築します。



- *OK* をクリックします。地図には今、私達の開始条件に一致し、100 メートル四方のサイズ以上である建物だけが表示されるはずです。

7.2.14 Try Yourself

- そうするために上記学んだ手法を用いて、新しいレイヤーとしてソリューションを保存します。ファイルは名前 *solution.shp* で、通常のディレクトリに保存する必要があります。

7.2.15 In Conclusion

QGIS ベクトル解析ツールと一緒に GIS の問題解決のアプローチを使用して、迅速かつ容易に複数の条件での問題を解決することができました。

7.2.16 What's Next?

次のレッスンでは、ある地点から別の地点への道に沿って最短距離を計算する方法を見ていきます。

7.3 Lesson: ネットワーク分析

2点間の最短距離を計算することは、よく例証される GIS の用法です。このツールは QGIS に入っていますが、デフォルトでは表示されていません。この短いレッスンでは、始めるために必要なことを紹介します。

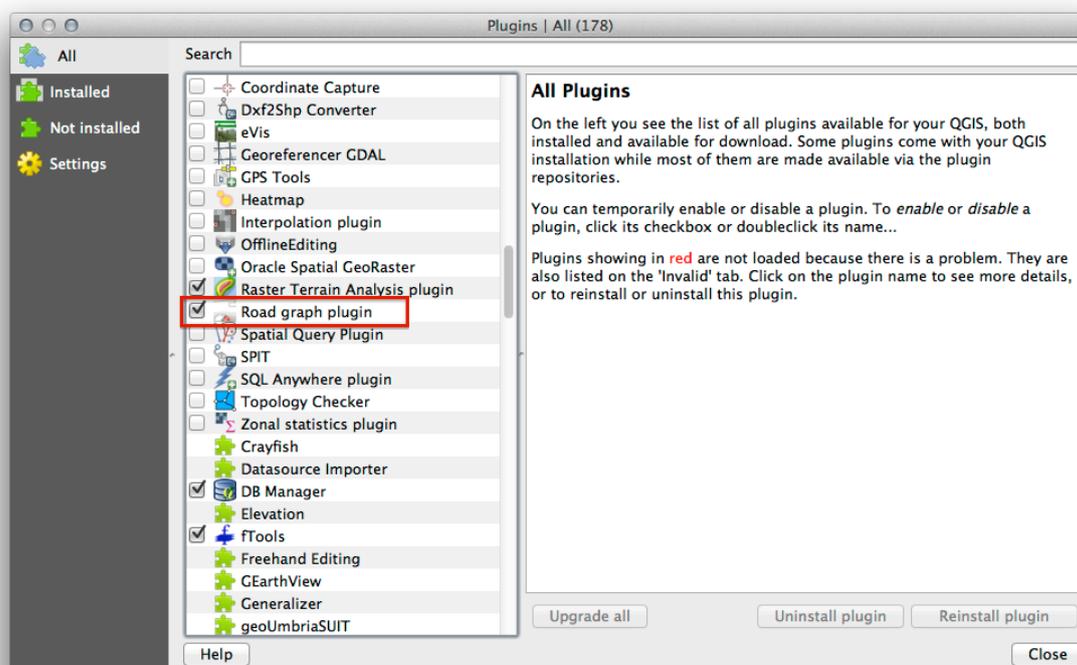
このレッスンの目標: 道路グラフ プラグインの有効化、設定と利用。

7.3.1 Follow Along: ツールの有効化

QGIS は、その基本的な機能に追加する多くのプラグインがあります。これらのプラグインの多くは、入手後すぐにプログラムと一緒に利用できるため便利です。しかし、それらはデフォルトではまだ非表示になっています。そのため、それらを使用するためには、まず有効にする必要があります。

道路グラフ プラグインの有効化:

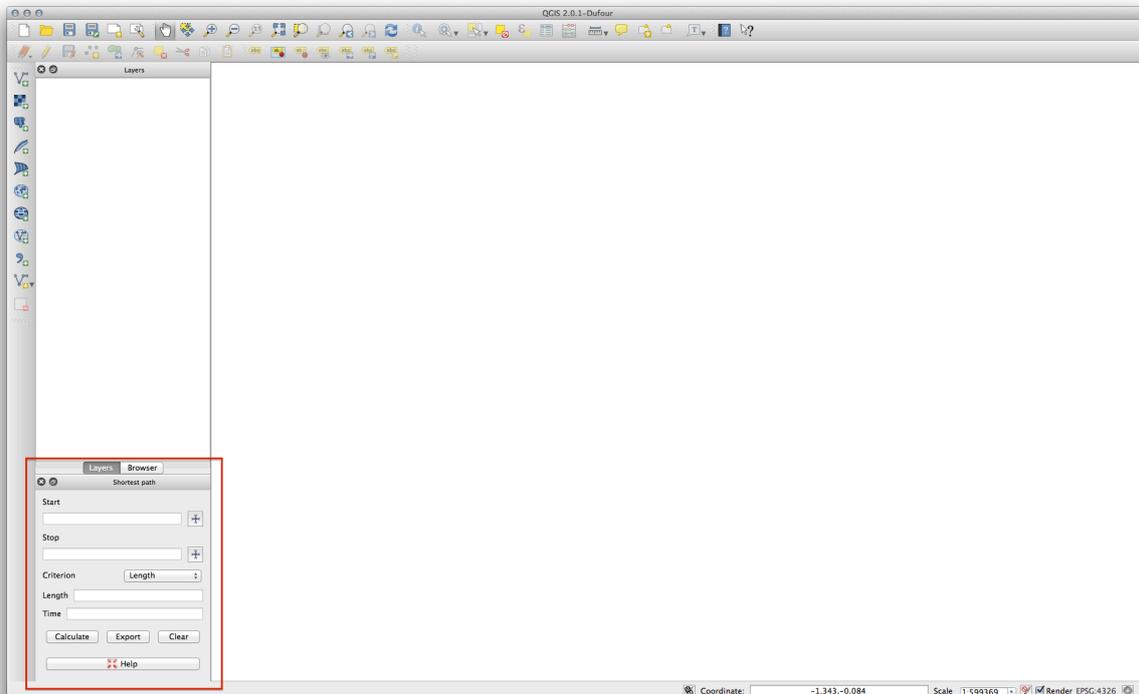
- QGIS のメイン画面にあるメニューの プラグイン → プラグインの管理とインストール... をクリックすることで、プラグインマネージャを起動します。ダイアログが表示されます。
- このようにプラグインを選択します:



- プラグインマネージャの 閉じる をクリックします。

ノート: インタフェース上でこのプラグインが表示されなければ、表示 → パネルに行き、最短パスのチェックマークが入っているかを確認します。

このパネルはインタフェースで表示されるでしょう:

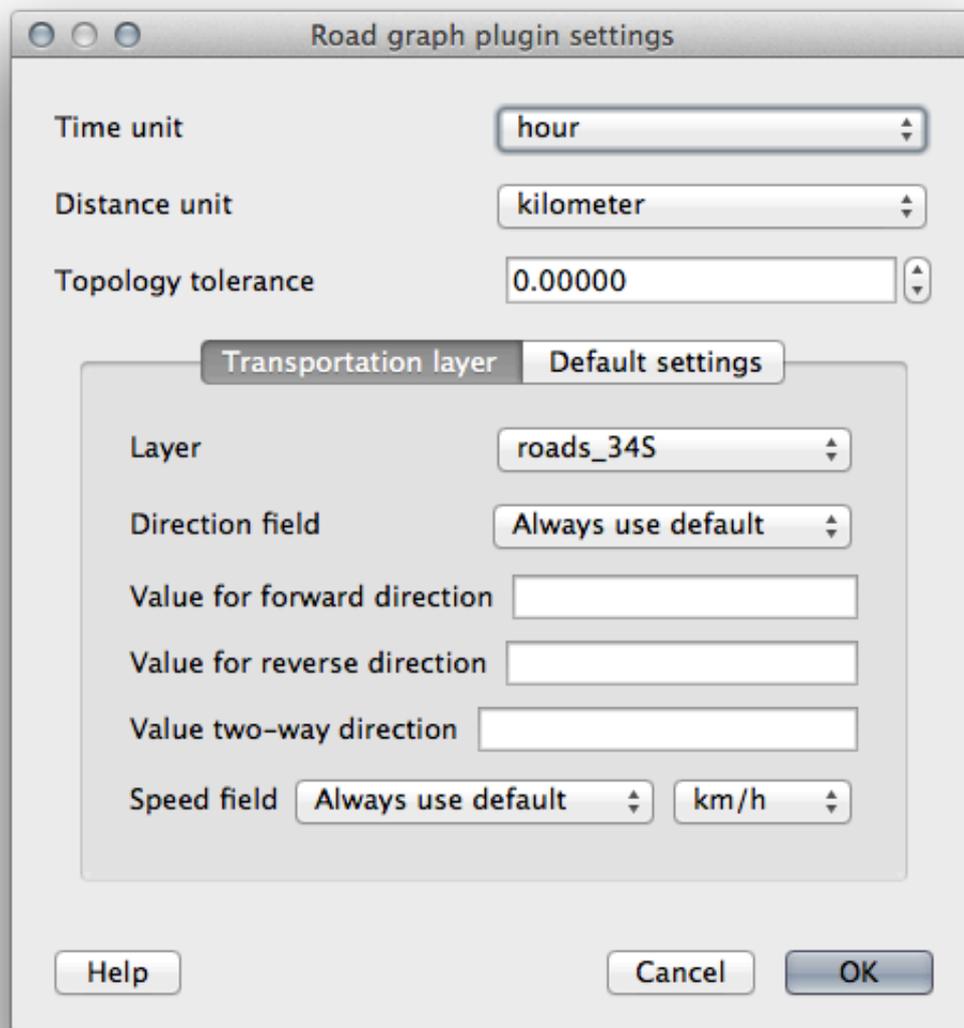


7.3.2 Follow Along: ツールの設定

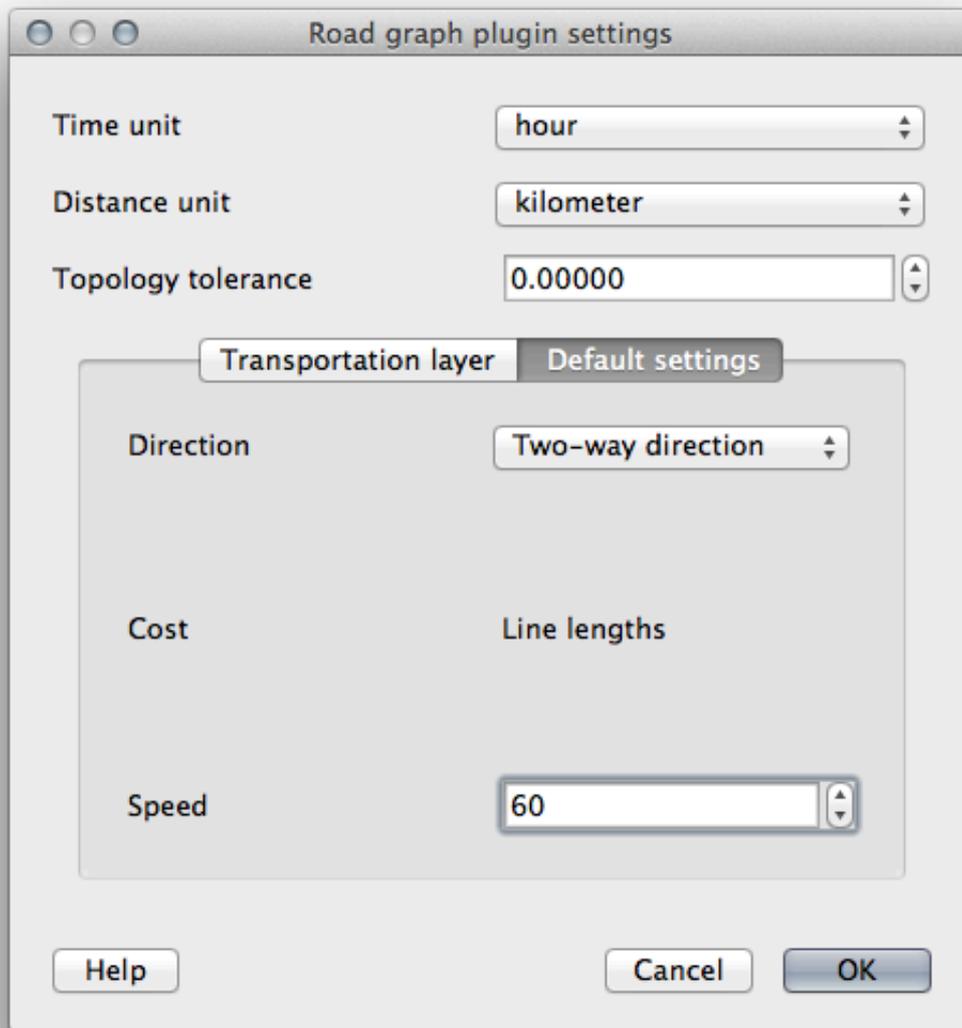
レイヤーでの計算をオンにするには、まず現在の地図を保存します。まだ行っていない場合は、roads_34S レイヤーを右クリックして 別名で保存... を選択することによってレイヤーをシェープファイルへ保存します。新しい地図を作成し、その中にこのレイヤーをロードします。

ネットワークを分析するには非常に多くの異なる構成が可能なので、プラグインでは設定する前に何も仮定しません。これは、プラグインは最初に設定しなければ全く何もしないことを意味します。

- ベクタ → 道路グラフ → 設定 をクリックします。ダイアログが表示されます。
- このように設定されていることを確認します (特に指定しない限りデフォルトを使用)。



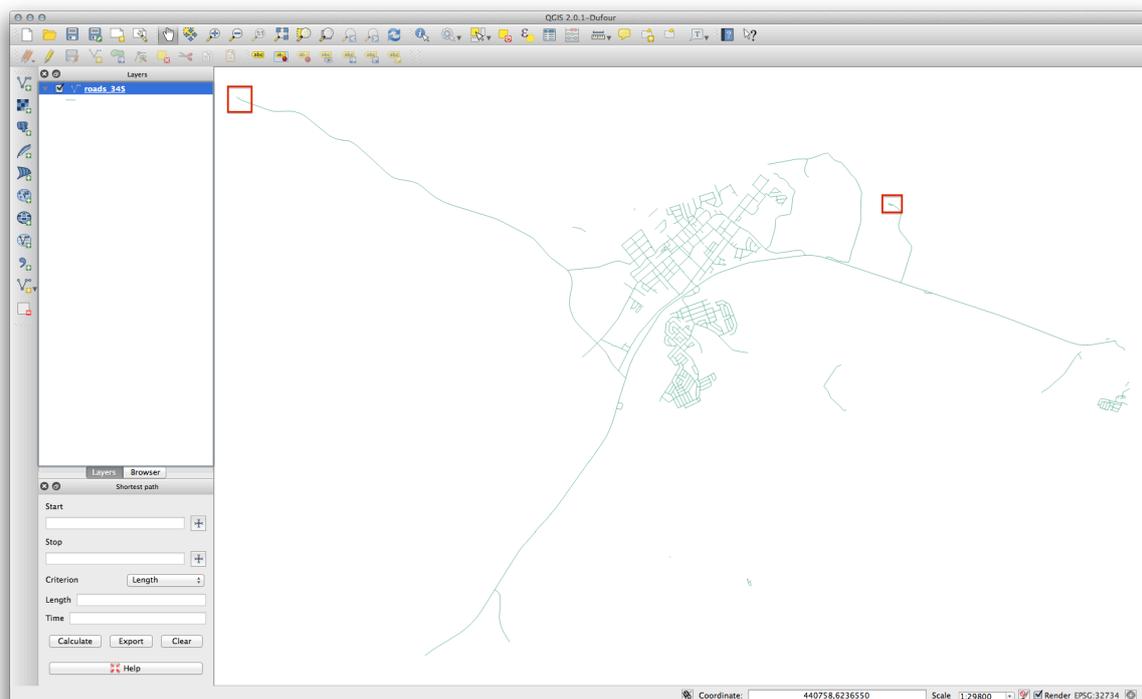
- 時間の単位 : 時間
- 距離の単位 : *km*
- レイヤ : *roads_34S*
- 速度フィールド : 常にデフォルトを使用 / *km/h*



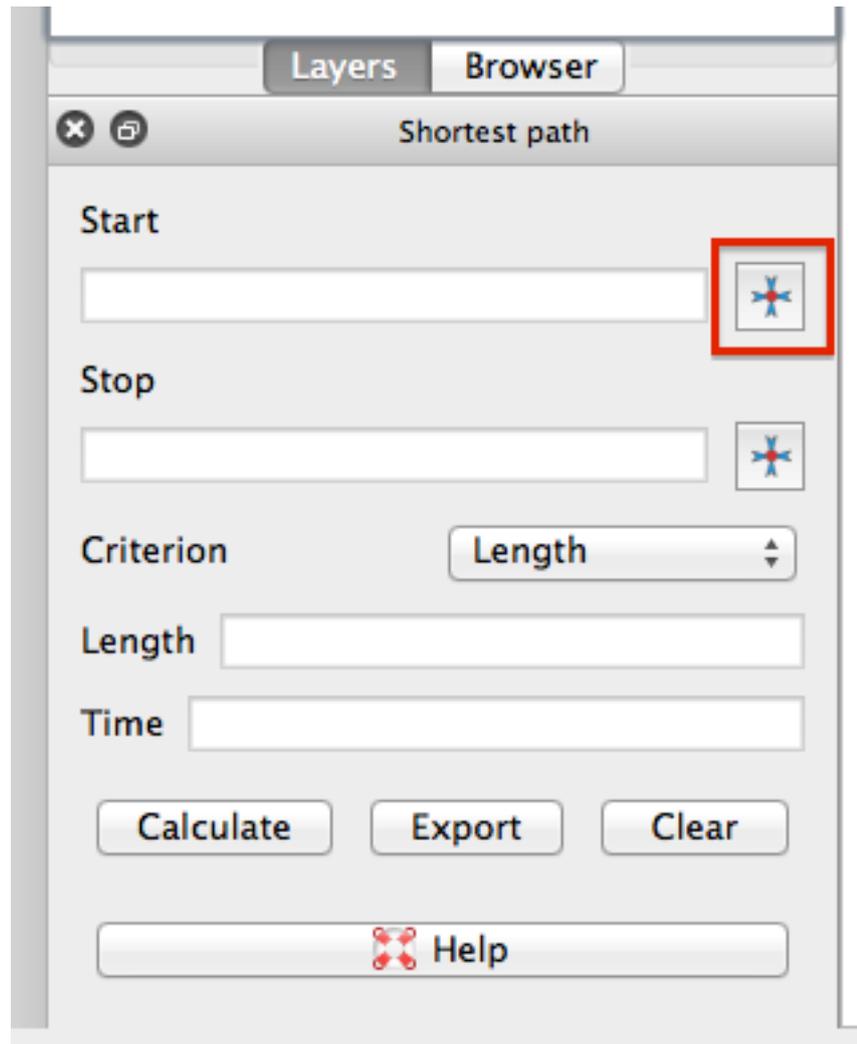
- 方向 : 両方向
- スピード : 60

7.3.3 Follow Along: ツールの使用

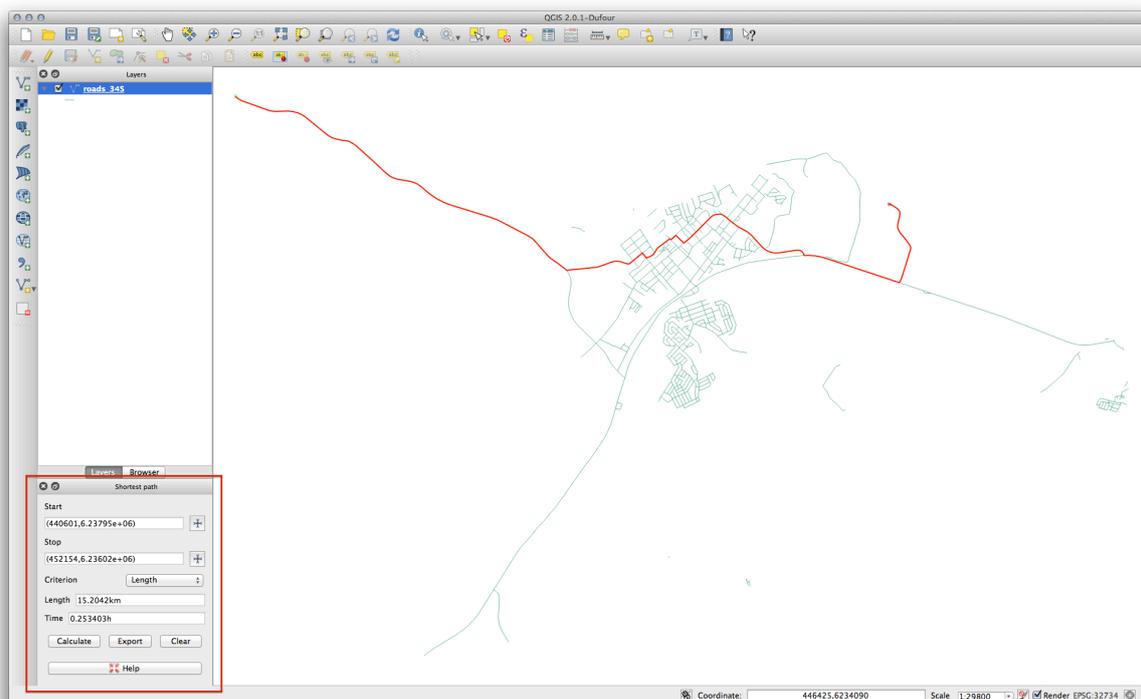
地図上で、道路上で、2つの点を見つけます。それらの点は何らかの意味を持っている必要はありませんが、道によって連結されており、それなりの距離で隔てられている必要があります。



- プラグインパネルで、開始 フィールドの隣にある ポイントのキャプチャ ボタンをクリックします。



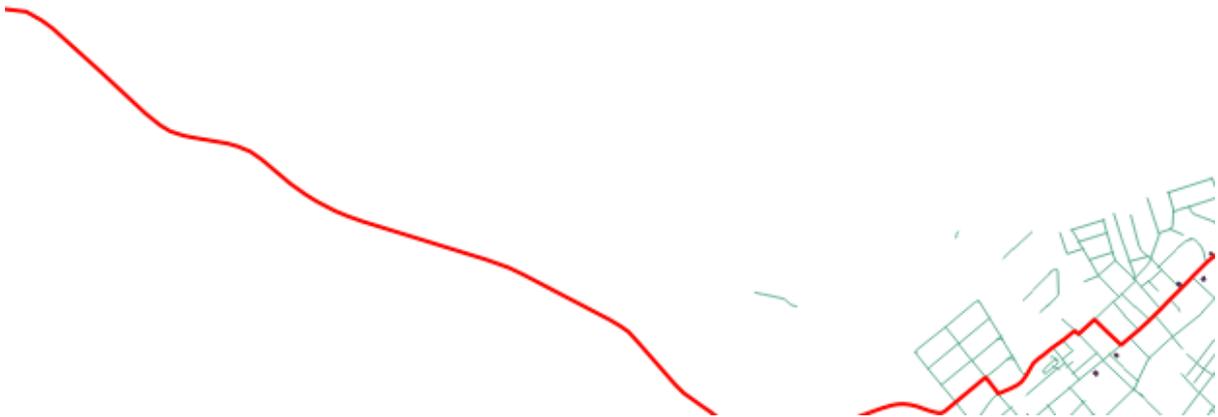
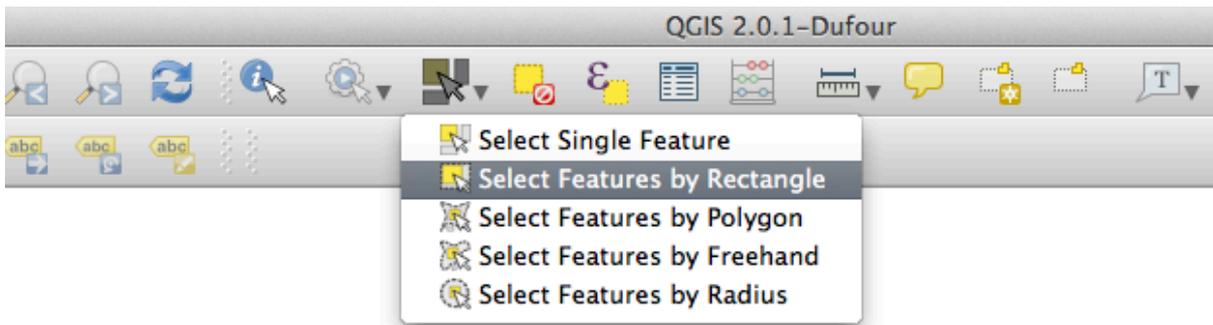
- 選択した始点でクリックします。
- *Stop* フィールドの隣 *Point* をキャプチャ ボタン を使用して選択した端点をキャプチャします。
- 計算 ボタンをクリックし、ソリューションを確認します。



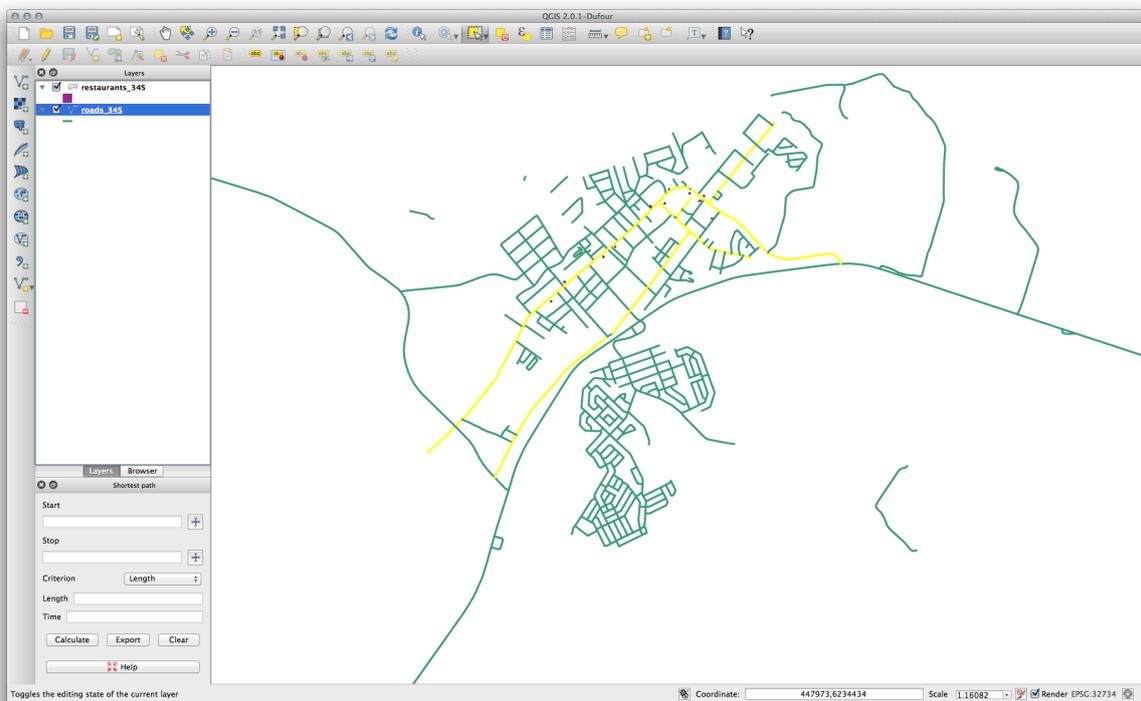
7.3.4 Follow Along: 階級の使用

ノート: Lesson developed by Linfiniti and S Motala (Cape Peninsula University of Technology)

- restaurants_34S レイヤ を地図に追加します (必要に応じて analysis 地図からそれを抽出します)。
- roads_34S レイヤー用の属性テーブルを開き、編集モードに入ります。
- 名前 SPEED の新しい列を追加し、それに幅 3 の `:guiabel:'全体数 (整数)'` タイプを与えます。
- メインウィンドウにて、四角でフィーチャを選択する ツールをアクティブにします。

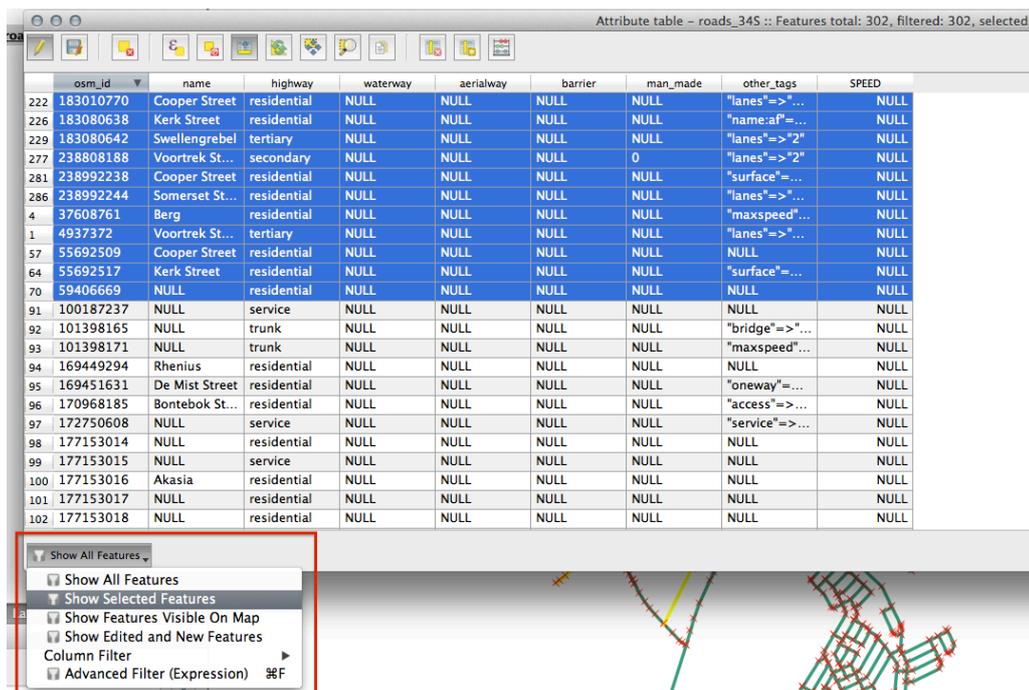


- 都市部の、ただし住宅地域ではない、主要道路を選択します :

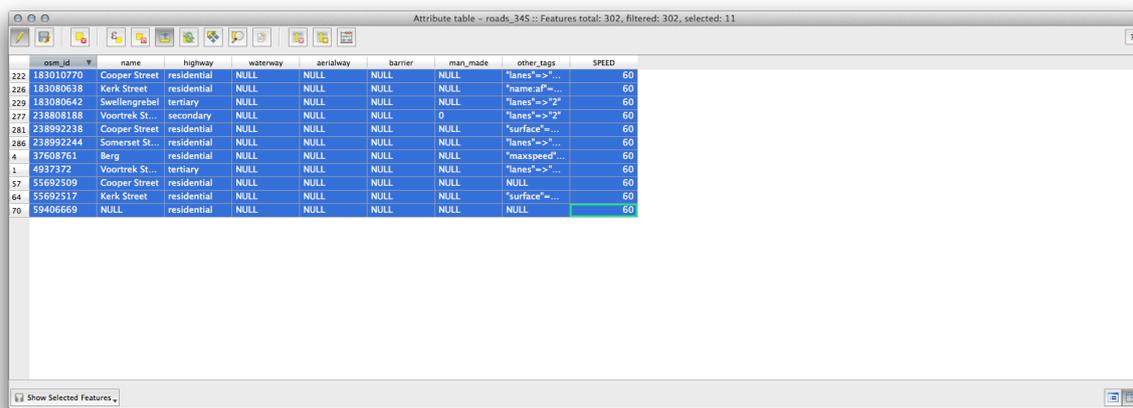


(複数の道を選択するには、ctrl ボタンを押しながら、選択に含めたいすべての道路がかかるようにボックスをドラッグします。)

- 属性テーブルで、選択地物を表示 を選択します。

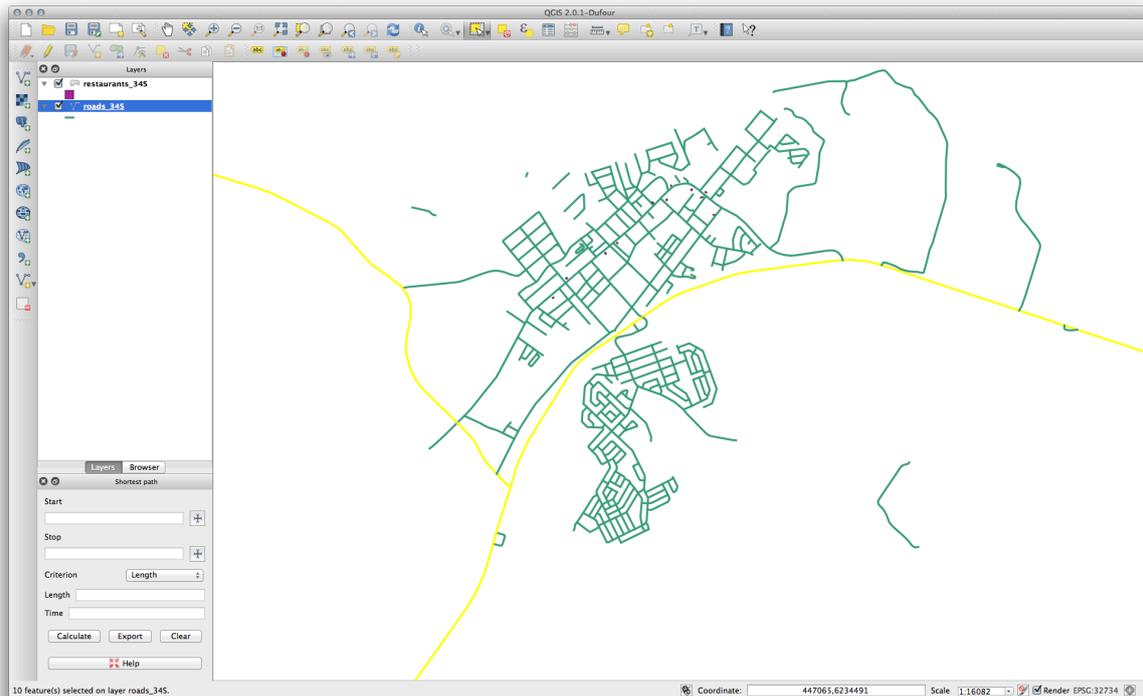


- 選択したすべての街路に対して 速度 値を 60 に設定します :

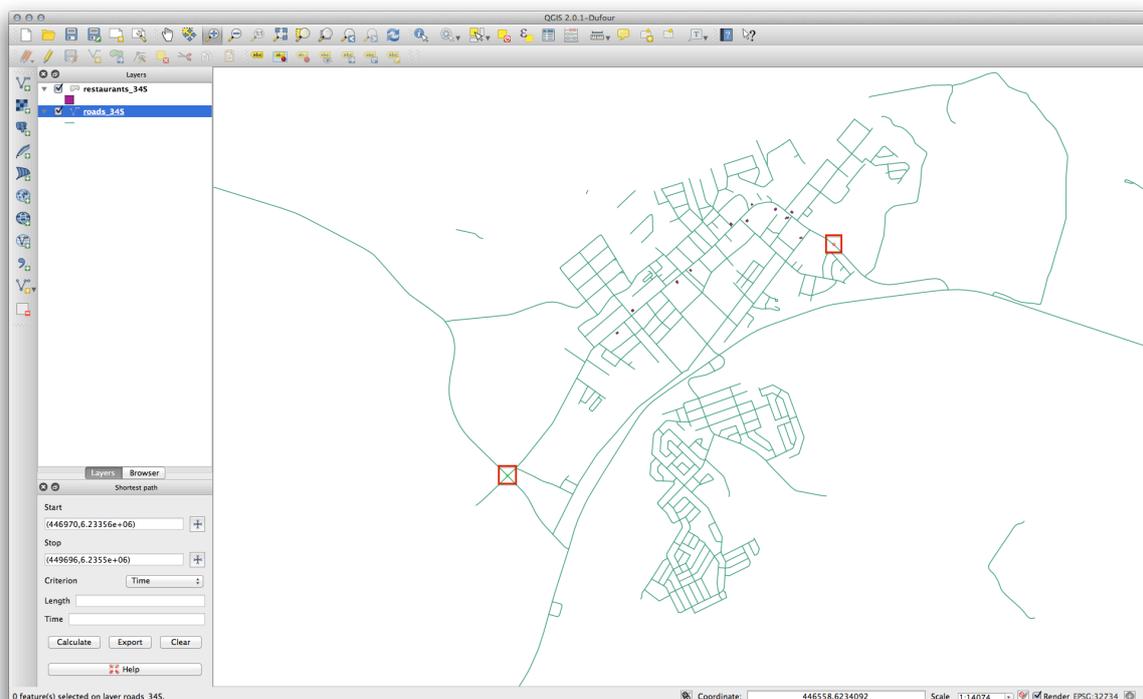


文脈上、これは、それらの道路の制限速度を 60km/h に設定していることを意味します。

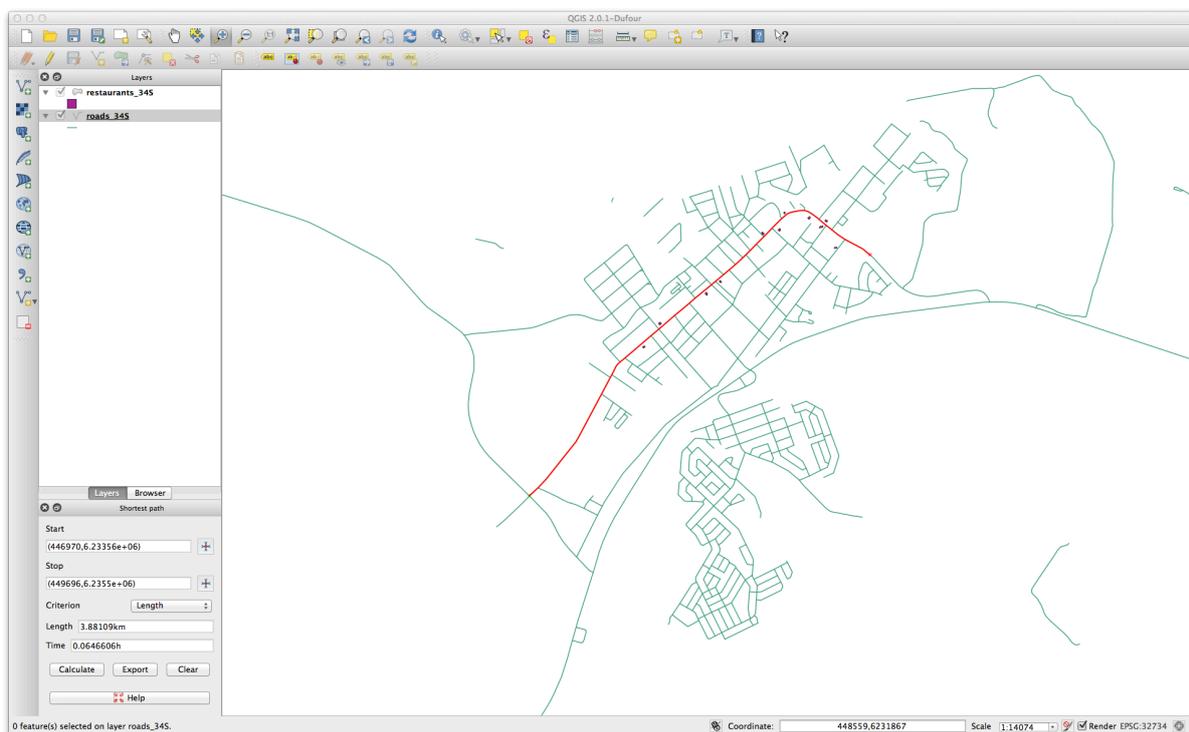
- 都市エリアの郊外にある高速道または主要道を選択します。



- 選択したすべての道路に対して、速度 値を 120 に設定します。
- 属性テーブルを閉じ、編集内容を保存し、編集モードを終了します。
- このレッスンで以前に説明したようにセットアップされたことを確実にするよう ベクタ → 道路グラフ → 道路グラフ設定 をチェックしますが、速度 値はたったいま作成した 速度 フィールドに設定してください。
- 最短経路 パネルで、始点 ボタンをクリックします。
- Swellendam の片側に非主要道路上に始点を、町の反対側の主要道路上に終点を設定します：

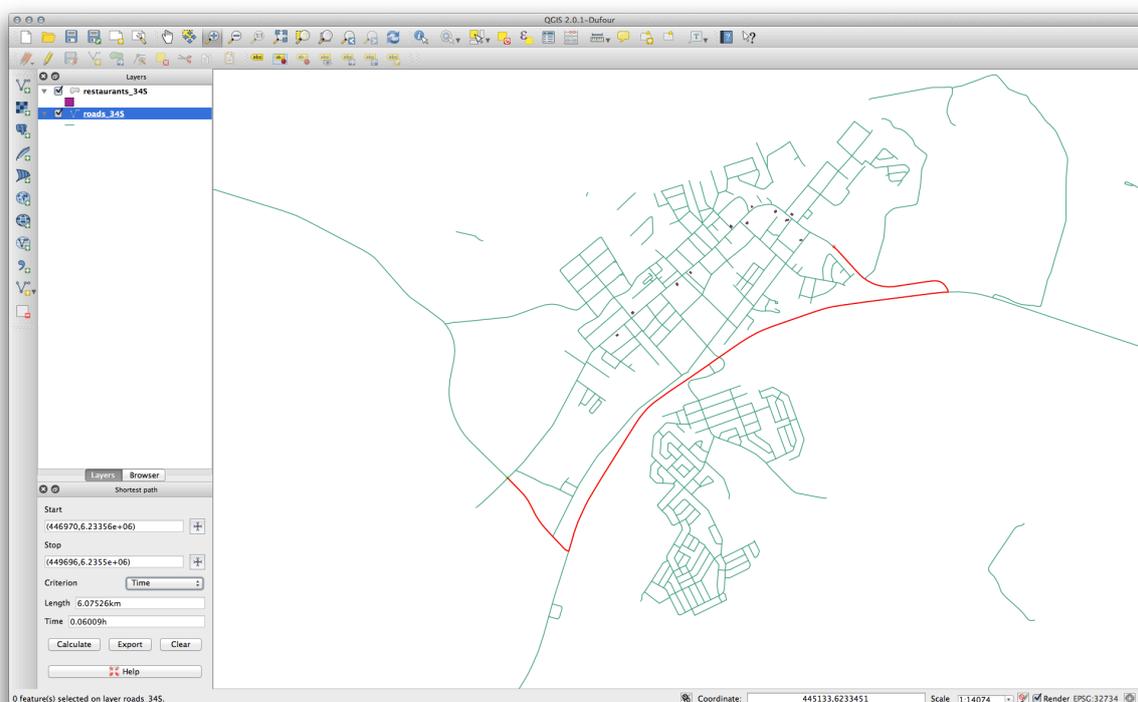


- 最短経路 パネルの 基準 ドロップダウンリストで 距離 を選択します。
- 計算 をクリックします。最短距離で経路が計算されます。



最短経路 パネル中の 距離 と 時間 の値に注目してください。

- 基準 を 時間 に設定します。
- *Calculate* をもう一度クリックします。最短時間で経路が計算されます。



毎回再計算しつつ、これらの基準の間で前後に切り替え、かかる 距離 と 時間 の変化に注意することができます。ルートを移動するのに要する時間に到着するためになされている前提は加速を考慮していないこ

とに注意してください、すべての回で制限速度で走行されることを前提としています。実際の状況では、制限速度よりは、より小さな区間に道路を分割して各区間の平均速度または期待速度に注意したいかもしれません。

もし、計算をクリックしたところで、経路が見つかりませんでしたというエラーが出る場合は、デジタル化した道路が実際にお互いに合致しているか確認してください。それらが全くタッチしていない場合は、地物を修正することで手直すか、プラグインの設定でトポロジ許容誤差を設定します。それらが交差することなく、互いの上を通過している場合、地物分割ツールを使用して、その交差点で道路を「分割」します：



ただし、地物を分割ツールは、編集モードで、選択した地物にのみ動作することを忘れないでください！エラーが返った場合、最短ルートはもっともはやいことがわかるかもしれません。

7.3.5 In Conclusion

これで道路グラフプラグインを使用して最短経路の問題を解決する方法がわかりましたね。

7.3.6 What's Next?

次は、ベクター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。

7.4 Lesson: 空間統計

ノート: Lesson developed by Linfiniti and S Motala (Cape Peninsula University of Technology)

空間統計では、分析し、与えられたベクターデータセットで何が起きているかを理解することができます。QGISは、この点で有用であることが分かる統計分析のためのいくつかの標準的なツールが含まれています。

****このレッスンの目標:****QGISの空間統計ツールの使い方を知ること。

7.4.1 Follow Along: テストデータセットの作成

ポイントデータセットの操作を知るために、ポイントのランダムセットを作成します。

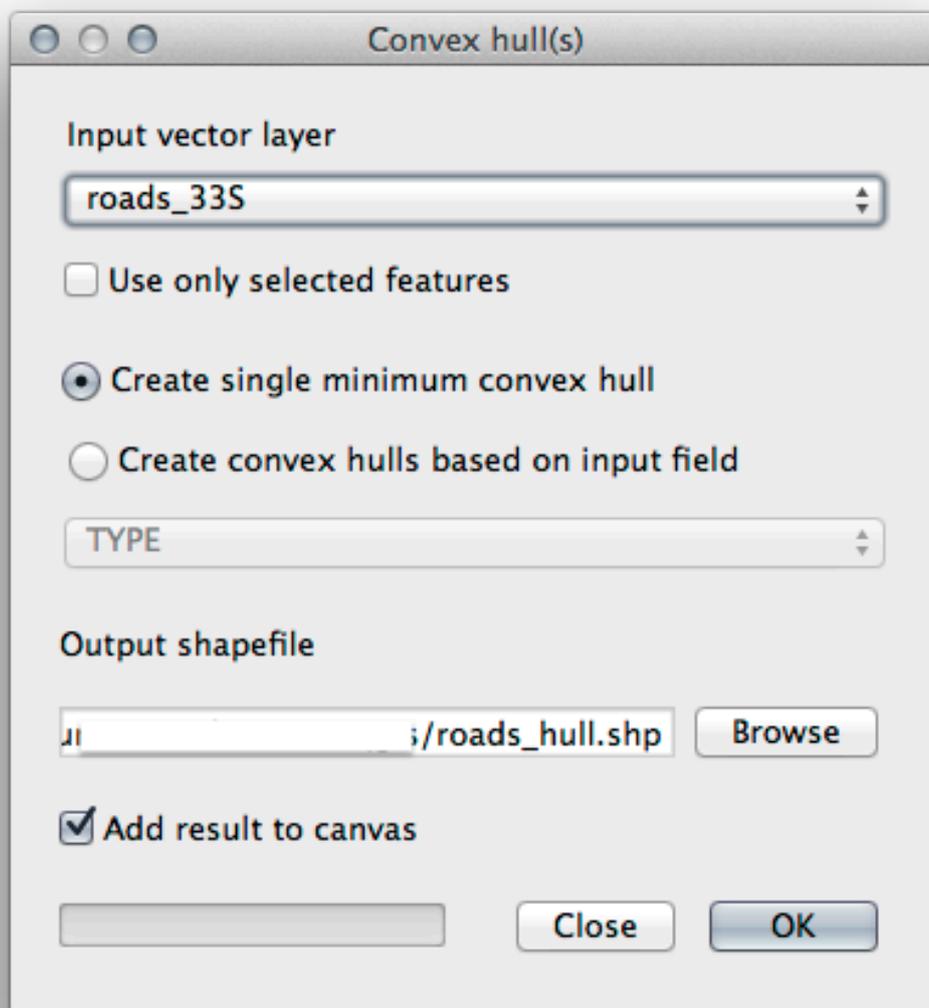
そのためには、ポイントを作成したいエリアの範囲を定義するポリゴンデータセットが必要です。

ストリートで覆われているエリアを使います。

- 空のマップを新規に開始します。
- roads_34S レイヤと exercise_data/raster/SRTM/ にある srtm_41_19.tif raster (elevation data) を追加します。

ノート: あなたの SRTM の DEM レイヤは道路レイヤとは異なる CRS を持っていることがわかるかもしれません。その場合は、それ以前のこのモジュールの学んだ技術を使用して、道路または DEM レイヤのいずれかを再投影することができます。

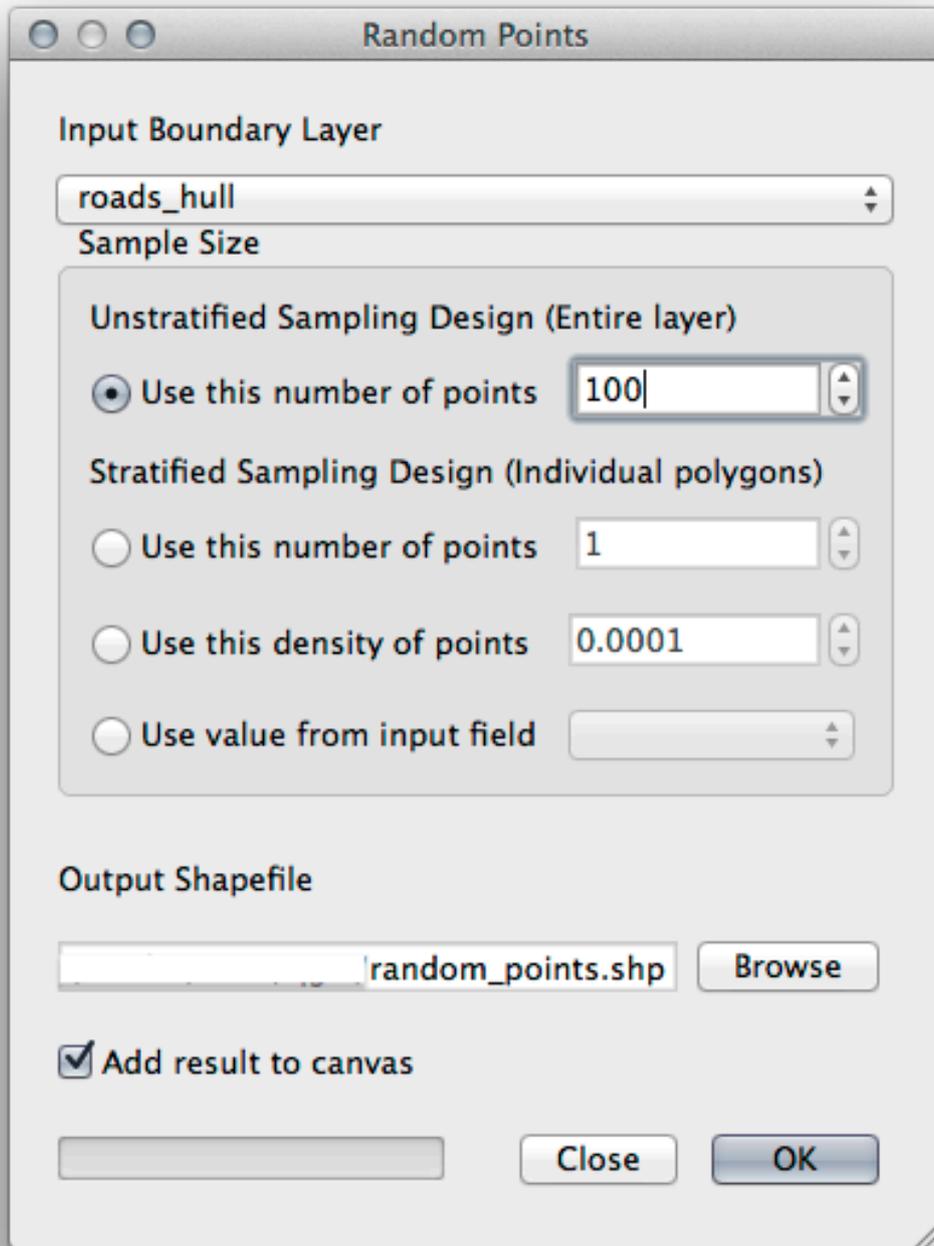
- 凸包ツール (ベクタ → ジオプロセッシングツール で利用可能) を使用し、道路をすべて覆ったエリアを生成します。



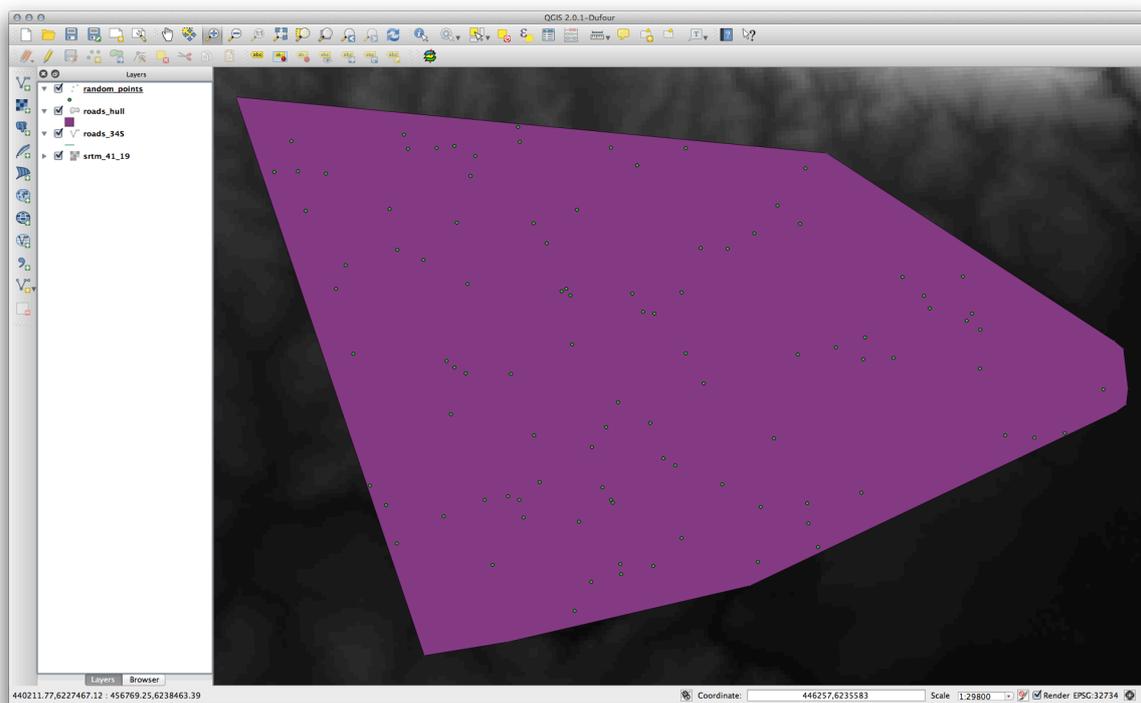
- Save the output under `exercise_data/spatial_statistics/` as `roads_hull.shp`.
- 結果をキャンバスに追加する オプションをチェックして、出力を TOC (レイヤー *list*) に追加します。

ランダム点群の作成

- ベクタ → 調査ツール → ランダム点群 ツールを使用して、このエリアにランダム点群を作成します。

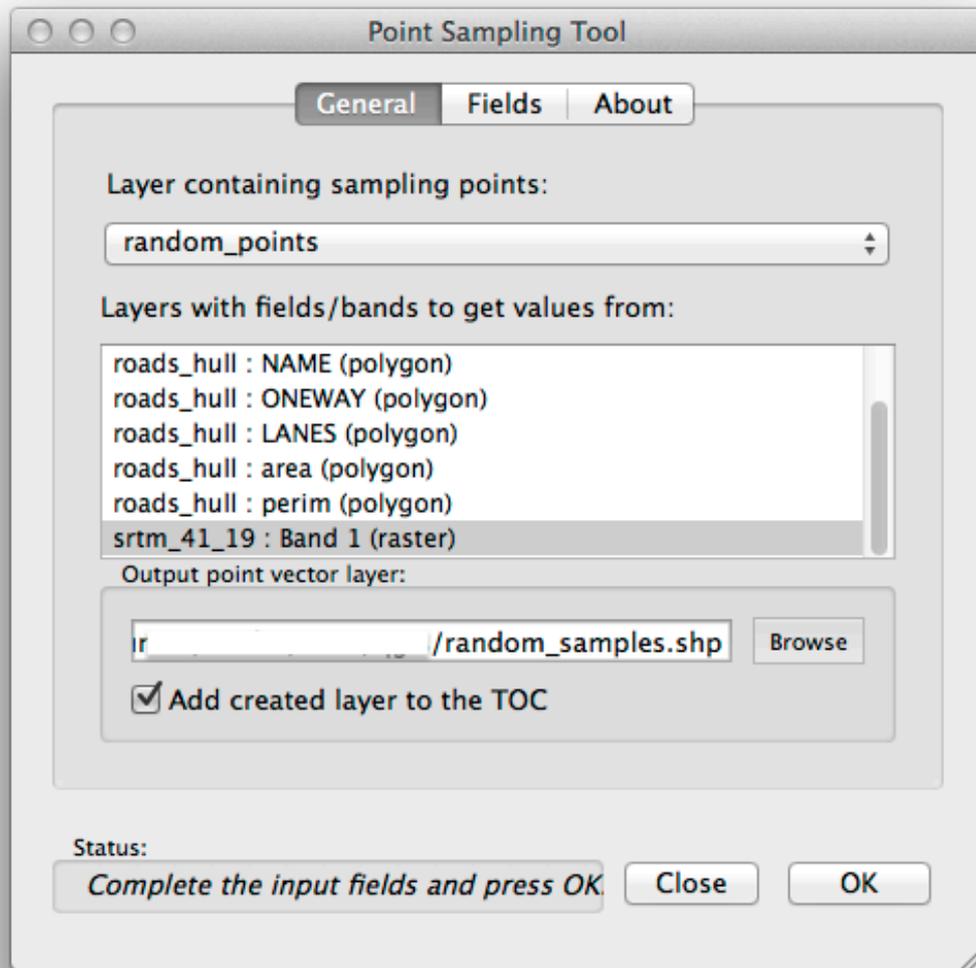


- exercise_data/spatial_statistics/ 以下に random_points.shp として保存します。
- 結果をキャンバスに追加する オプションをチェックして、出力を TOC (レイヤー list) に追加します。



データのサンプリング

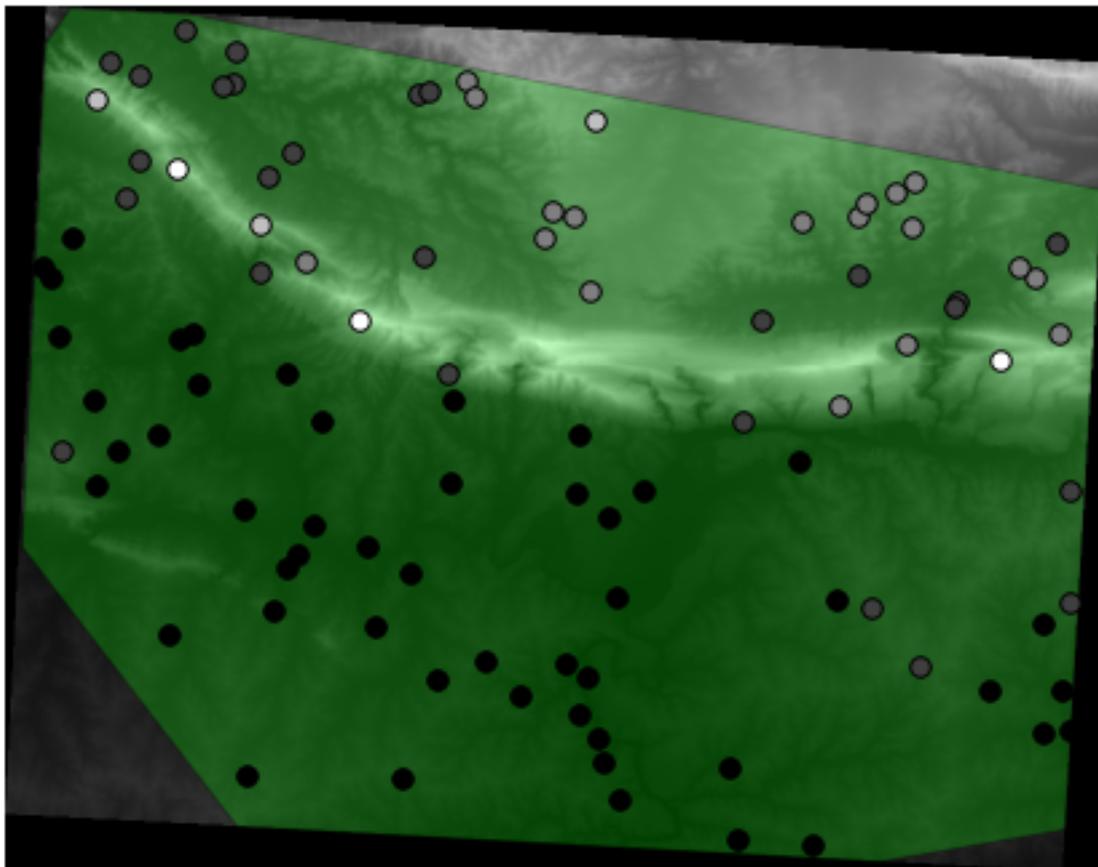
- ラスタからサンプルデータセットを作成するため、ポイントサンプルツール プラグインを使う必要があるでしょう。
- 必要に応じてプラグインで、モジュールを先に参照してください。
- *Plugin* → プラグインの管理とインストール... で point sampling というフレーズで検索し、このプラグインを見つけます。
- プラグインマネージャ で有効化されたら、プラグイン → 分析 → ポイントサンプリングツール を見つけることができます。



- サンプル点群を含むレイヤを ランダム点群 として、SRTM ラスタを値を取得するバンドとして選択します。
- “作成したレイヤを TOC に追加する” がチェックされていることを確認します。
- exercise_data/spatial_statistics/ 以下に random_samples.shp を出力して保存します。

今、random_samples レイヤーの属性テーブル内でラスタファイルからサンプリングしたデータを確認できます、それらは名前srtmFileName1の列にあります。

サンプルレイヤはここに示すとおりです:

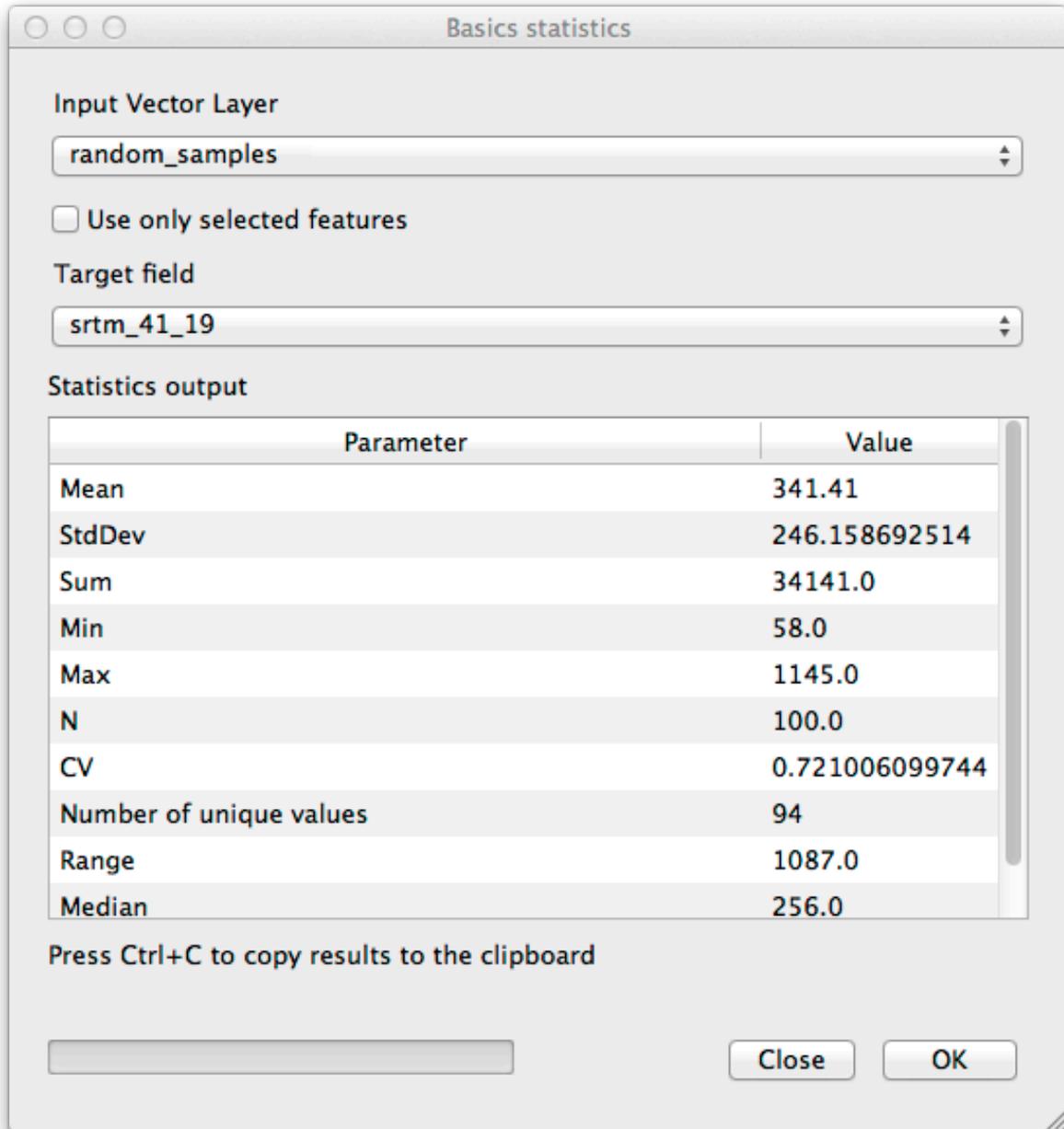


より暗い点が低い高度であるように、サンプル点はそれらの値によって分類されます。
あなたは、残りの統計エクササイズのためにこのサンプルレイヤを使っています。

7.4.2 Follow Along: 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。

- ベクトル->分析ツール->基本統計 メニューエントリをクリックします。
- 表示されたダイアログで、ソースとして *random_samples* レイヤーを指定します。
- ターゲットフィールドが *srtm_41_19.tif* に設定されていることを確認してください。これは統計を計算されるフィールドです。
- *OK* をクリックします。結果はこのようになります。



ノート: スプレッドシートに結果をコピーして貼り付けできます。データはセパレータ(コロン:)を使用します。

	A	B
1	Mean	343.9
2	StdDev	254.4824748
3	Sum	34390
4	Min	34
5	Max	1226
6	N	100
7	CV	0.739989749
8	Number of unique values	91
9	Range	1192
10	Median	269

- 実行後はプラグインダイアログを閉じます。

上記の統計情報を理解するには、この定義のリストを参照してください。

平均 平均（平均）値は、単純な値の量で割った値の合計です。

StdDev 標準偏差。値は平均値の周りにクラスタ化されているか、密接に指示を与えます。標準偏差が小さいほど、より近い値が平均値になる傾向があります。

合計 すべての値を加算します。

最小 値の最小値です。

最大 値の最大値です。

N サンプル/値の量です。

CV データセットの *spatial covariance*

ユニークな値の数 このデータセット全体でユニークな値の数。N = 100 のデータセットで 90 個のユニークな値がある場合、残りの 10 の値はどれかと同じです。

レンジ 最小および最大値間の差です。

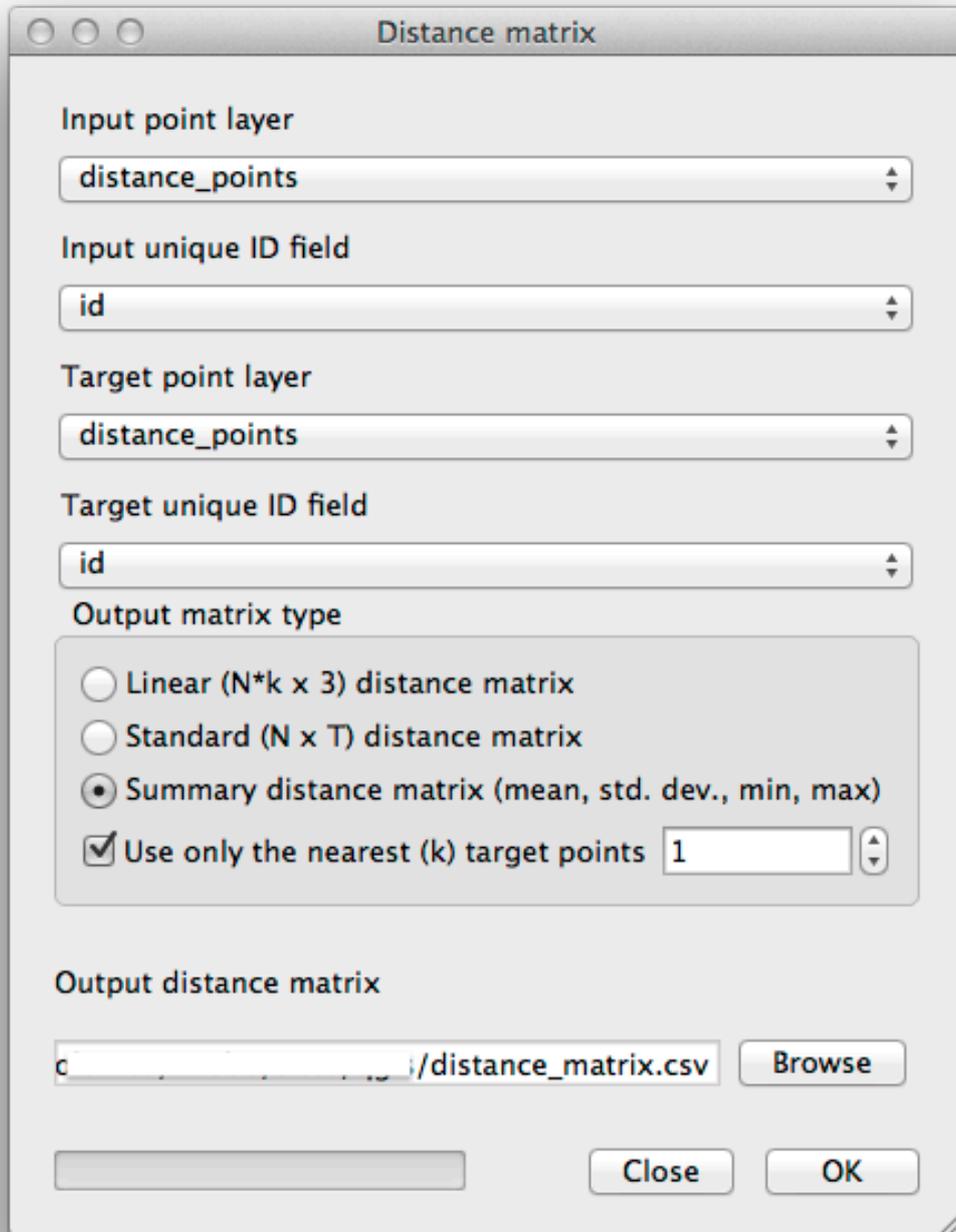
中間値 最小から最大までのすべての値を整理させた場合、真ん中の値（または N が偶数である場合は真ん中の 2 つの値の平均）は値の中央値です。

7.4.3 Follow Along: 距離マトリックスの算出

- 他のデータセットと同じ投影で新しい点レイヤーを作成します (WGS 84 / UTM 34S)。
- 編集モードに入り、どこか他の点のうち 3 点をデジタル化。
- あるいは、前と同じランダム点生成方法を使用するが、3 点だけを指定します。
- 新規レイヤを `distance_points.shp` として保存します。

これらのポイントを使って距離マトリックスを生成します。

- ベクトル->分析ツール->距離 *matrix* ツールを開きます。
- 入力レイヤーとして `distance_points` レイヤー、ターゲットレイヤーとして `random_samples` レイヤーを選択します。
- このように設定します:



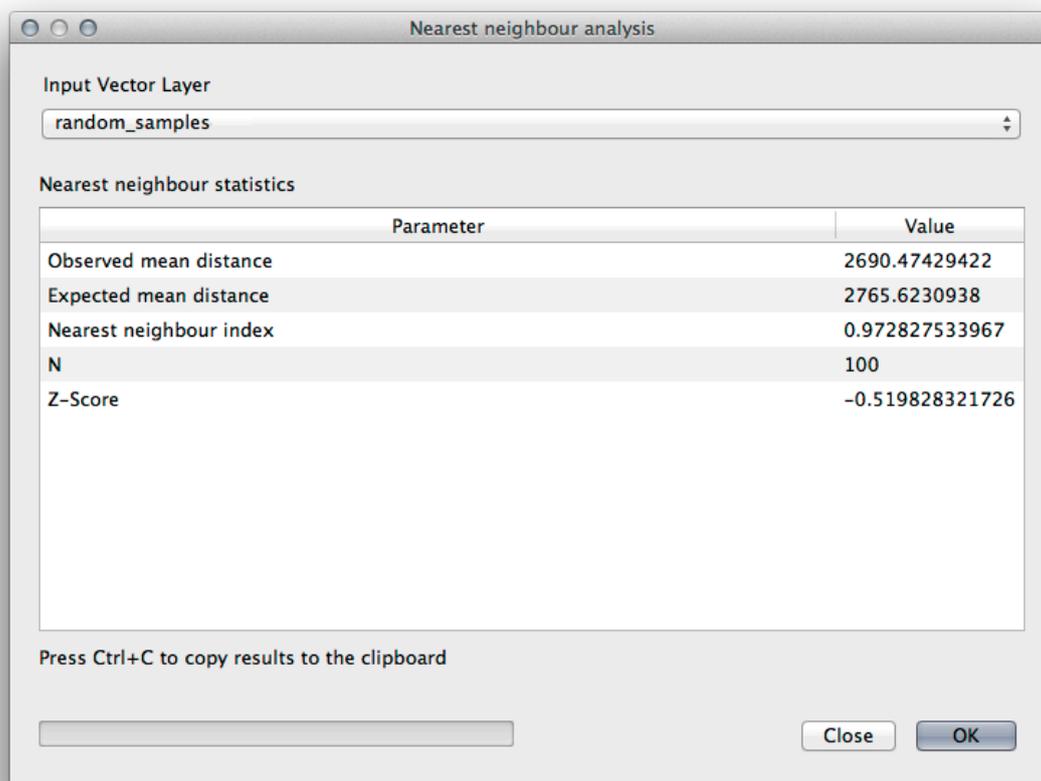
- Save the result as `distance_matrix.csv`.
- *OK* をクリックし、距離マトリックスを生成します。
- 結果を確認するには、スプレッドシートプログラムで開きます。次に例を示します。

InputID	MEAN	STDDEV	MIN	MAX
3	0.195448627921		0	0.195448627921
2	0.174928758638		0	0.174928758638
1	0.174928758638		0	0.174928758638

7.4.4 Follow Along: 最小近傍分析

最小近傍分析を行うために:

- ベクトル->分析ツール->最小近傍分析 メニュー項目をクリックします。
- 表示されるダイアログで、*random_samples* レイヤを選択し *OK* をクリックしてください。
- 結果は、たとえば、ダイアログのテキストウィンドウに表示されます。



ノート: スプレッドシートに結果をコピーして貼り付けできます。データはセパレータ (コロン :) を使用します。

7.4.5 Follow Along: 平均座標

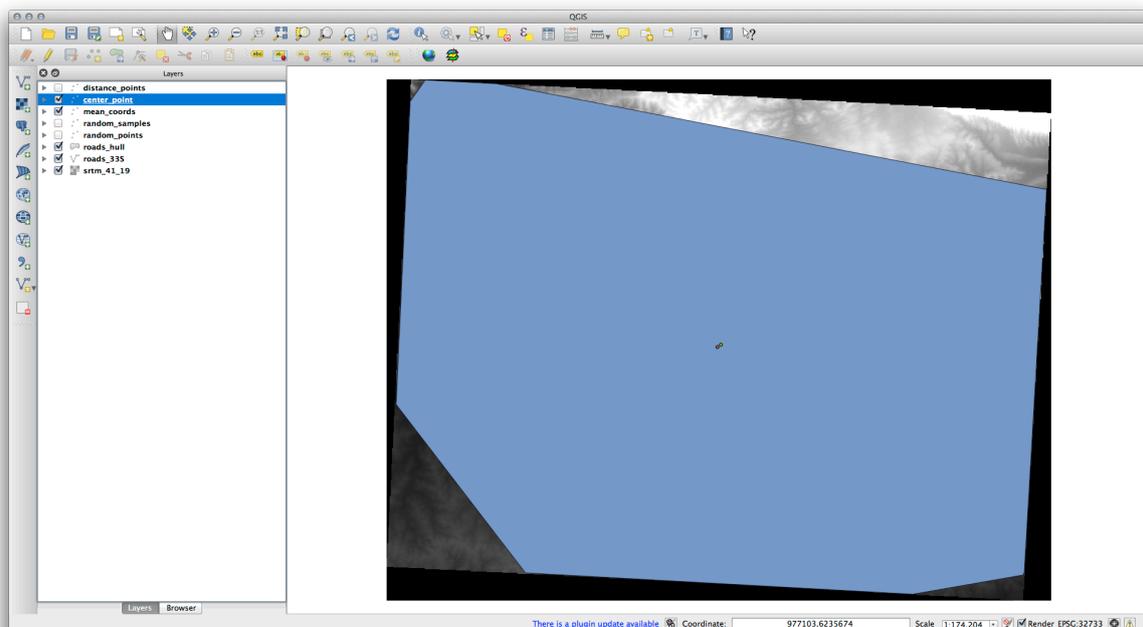
データセットの平均座標を取得するために:

- ベクター >分析ツール->平均座標 メニュー項目をクリックしてください。
- 表示されるダイアログで、入力レイヤーとして *random_samples* を指定しますが、オプションの選択肢はそのまま残します。
- Specify the output layer as *mean_coords.shp*.
- Click *OK*.
- メッセージが表示されたら レイヤリスト に追加されます。

ランダムなサンプルを作成するために使用されたポリゴンの座標の中央にこれを比較してみましょう。

- ベクトル->ジオメトリツール->ポリゴン重心 メニュー項目をクリックします。
- 表示されるダイアログで、入力レイヤーとして *roads_hull* を選択します。
- Save the result as *center_point*.
- メッセージが表示されたら (*Layers list*) に追加されます。

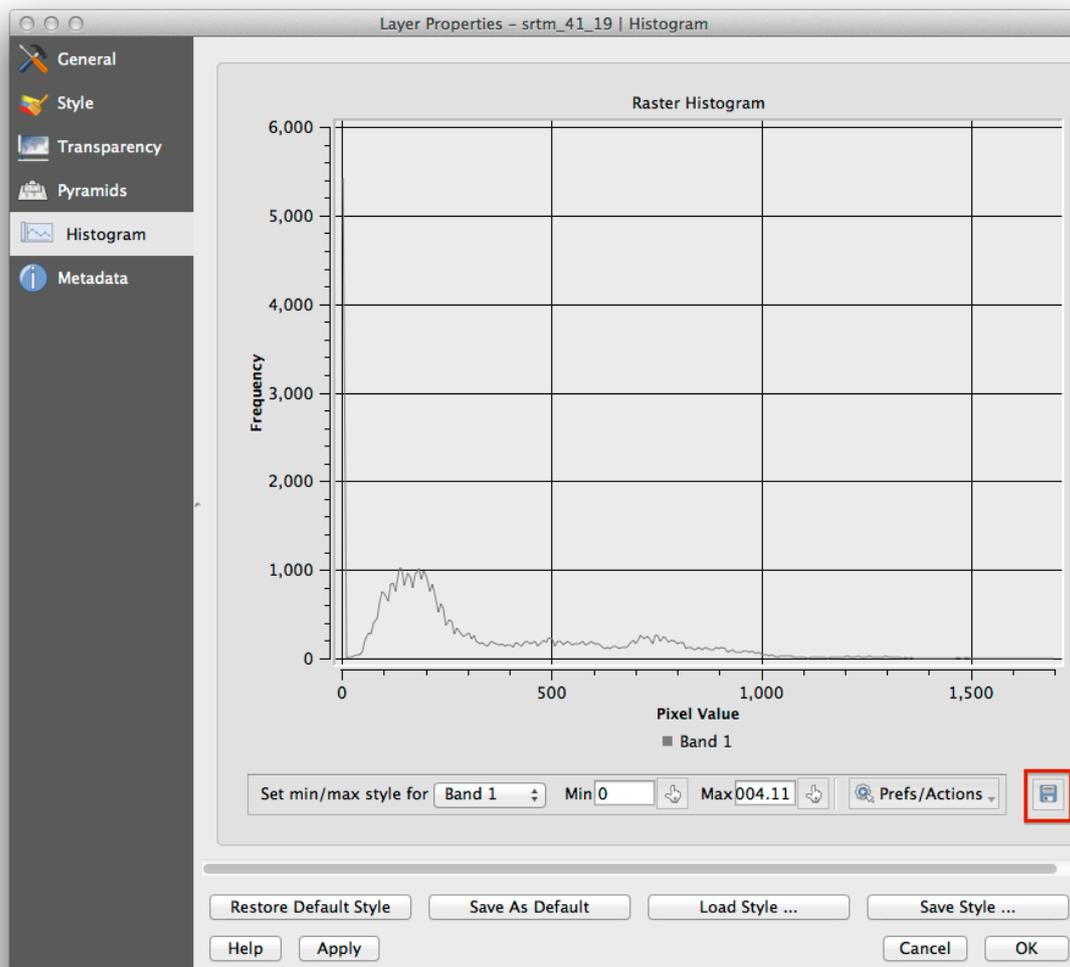
以下の例からもわかるように、平均座標と（オレンジ色）の研究領域の中心は必ずしも一致しません。



7.4.6 Follow Along: イメージヒストグラム

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。任意の画像層のレイヤ *Properties* ダイアログ QGIS でこれを実証するための最も簡単な方法はで利用可能な画像のヒストグラムを介してです。

- レイヤー *list* で SRTM の DEM レイヤーを右クリックします。
- プロパティ を選択します。
- 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 ヒストグラム計算 ボタンをクリックする必要があるかもしれません。画像内の値の度数を記述するグラフが表示されます。
- それをイメージとして出力できます:



- メタデータ タブを選択します。プロパティ ボックスに、内部のより詳細な情報を見ることができます。

平均値は 332.8 で、最大値は 1699 です！しかし、これらの値はヒストグラムに表示されません。なぜ？これは、平均以下の値を持つピクセルが豊富にあるのに比べて、それらが非常に少ないためです。これもまた、およそ 250 より高い値の度数をマークする目に見える赤い線が存在しないにもかかわらず、ヒストグラムがここまで右に拡張した理由です。

ですから、ヒストグラムは値の分布を示しており、すべての値がグラフに必ずしも表示されているではないことを覚えておいてください。

- (これで レイヤー *Properties* は閉じてかまいません。)

7.4.7 Follow Along: 空間的補間

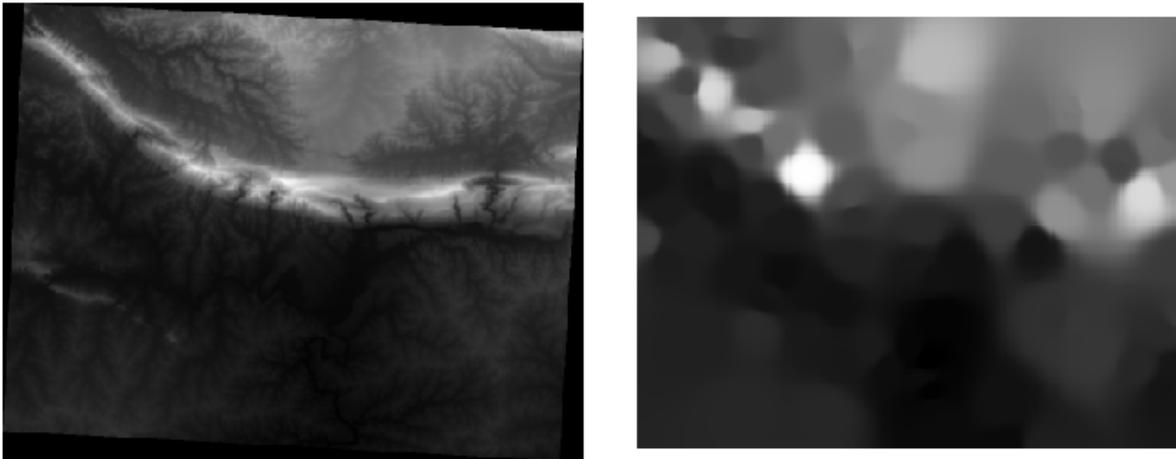
データを推定したいと思い、そこからサンプル点のコレクションを持っているとしましょう。たとえば、以前に作成した *random_samples* データセットへのアクセス権を持っている、そして地形がどのように見えるかのいくつかのアイデアを持ちたいかもしれません。

開始するには、ラスタ->分析->グリッド (補間) メニュー項目をクリックしてグリッド (補間) ツールを起動します。

- 入力ファイルフィールドで:`kbd:random_samples` を選択します。

- *Z Field* ボックスをチェックし、:kbd:‘srtm_41_19’ フィールドを選択します。
- 出力ファイル 場所を:kbd:exercise_data/spatial_statistics/interpolation.tif に設定します。
- アルゴリズム ボックスをチェックし、逆二乗距離 を選択します。
- Set the *Power* to 5.0 and the *Smoothing* to 2.0. Leave the other values as-is.
- Check the *Load into canvas when finished* box and click *OK*.
- 終わったら、処理完了 というダイアログで *OK* をクリックし、フィードバック情報を表示するダイアログで (現れた場合) *OK* をクリックし、:guilabel: グリッド (補間) ダイアログで 閉じる をクリックします。

ここにあるのは元のデータセット (左) と私たちのサンプルポイントから構築されたもの (右) との比較です。あなたのはサンプル点の位置のランダムな性質のため異なっている場合があります。

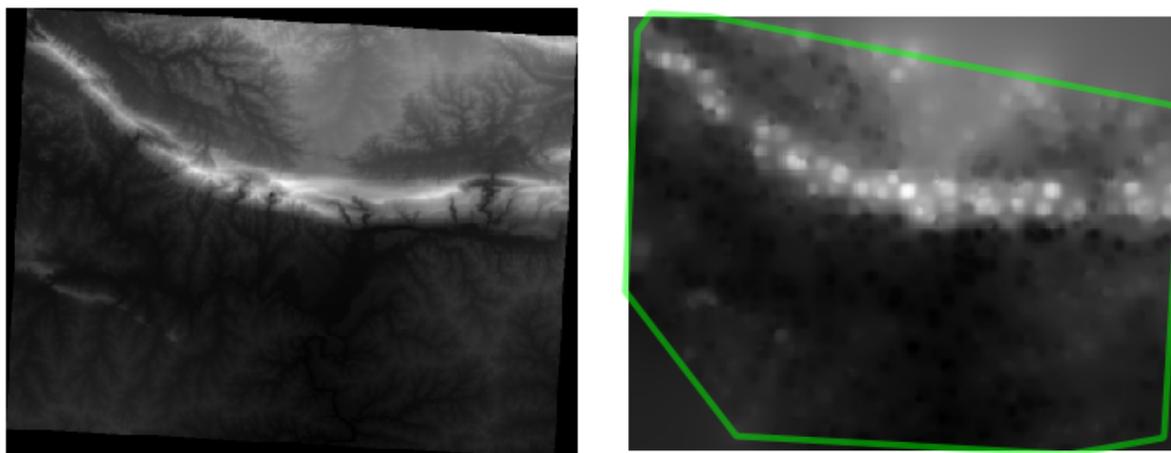


見ることができるように、サンプル 100 点では、地形の詳細な印象を受けるのは本当に十分ではありません。これは非常に一般的なアイデアを与えるが、それは同様に誤解を招くことができます。例えば、上記の画像では、東から西に実行高い、切れ目のない山があることは明らかではありません。むしろ、画像は西に高いピークと谷を、示しているようです。ただ目視検査を使用して、サンプルデータセットは、地形を表現していないことがわかります。

7.4.8 Try Yourself

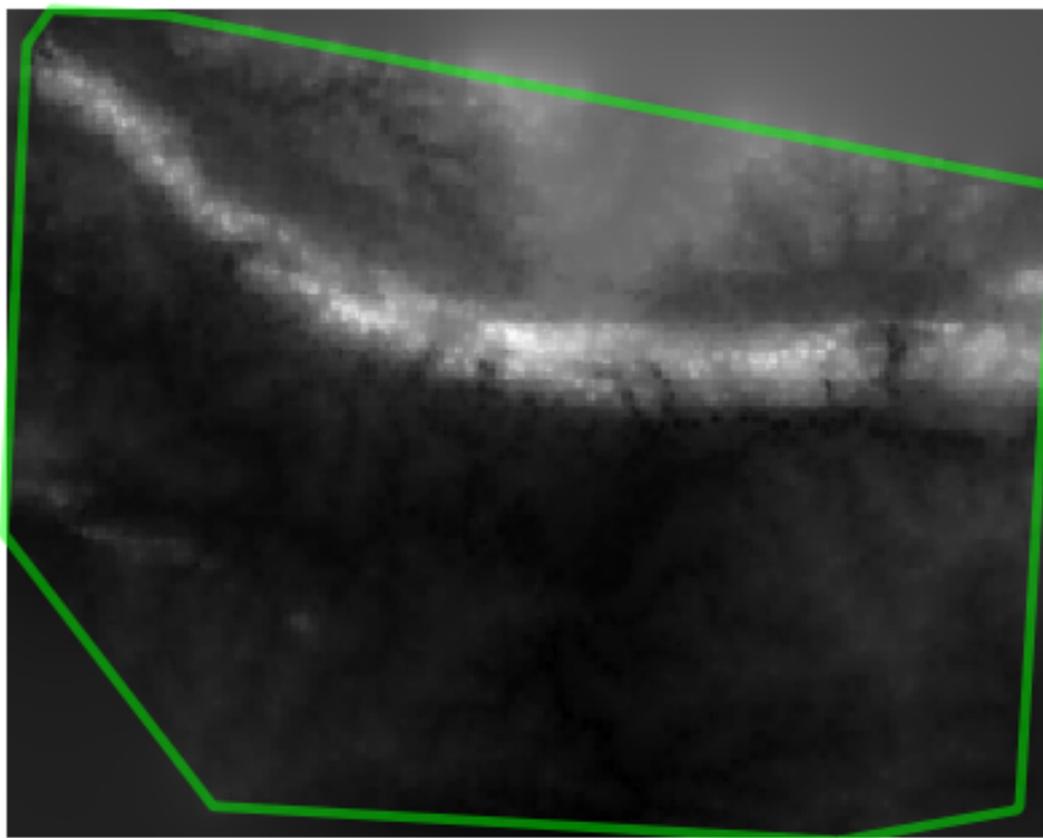
- 上に示したプロセスを使用して、ランダムな 1000 点の新しいセットを作成してください。
- オリジナルの DEM をサンプリングするためにこれらのポイントを使用してください。
- 上記のように、この新しいデータセットにグリッド (補間) ツールを使用します。
- *Power* と *Smoothing* をそれぞれ 5.0 、 2.0 と設定し、出力ファイル名を interpolation_1000.tif に設定します。

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。



境界は、縁部を越えて突然詳細がなくなることを説明する *roads_hull* レイヤー（ランダムサンプル点の境界を表す）を示しています。これは、サンプル点の密度がはるかに大きいため、地形をより良く表現しています。

サンプル 10 000 点だとどのように見えるかの例です：

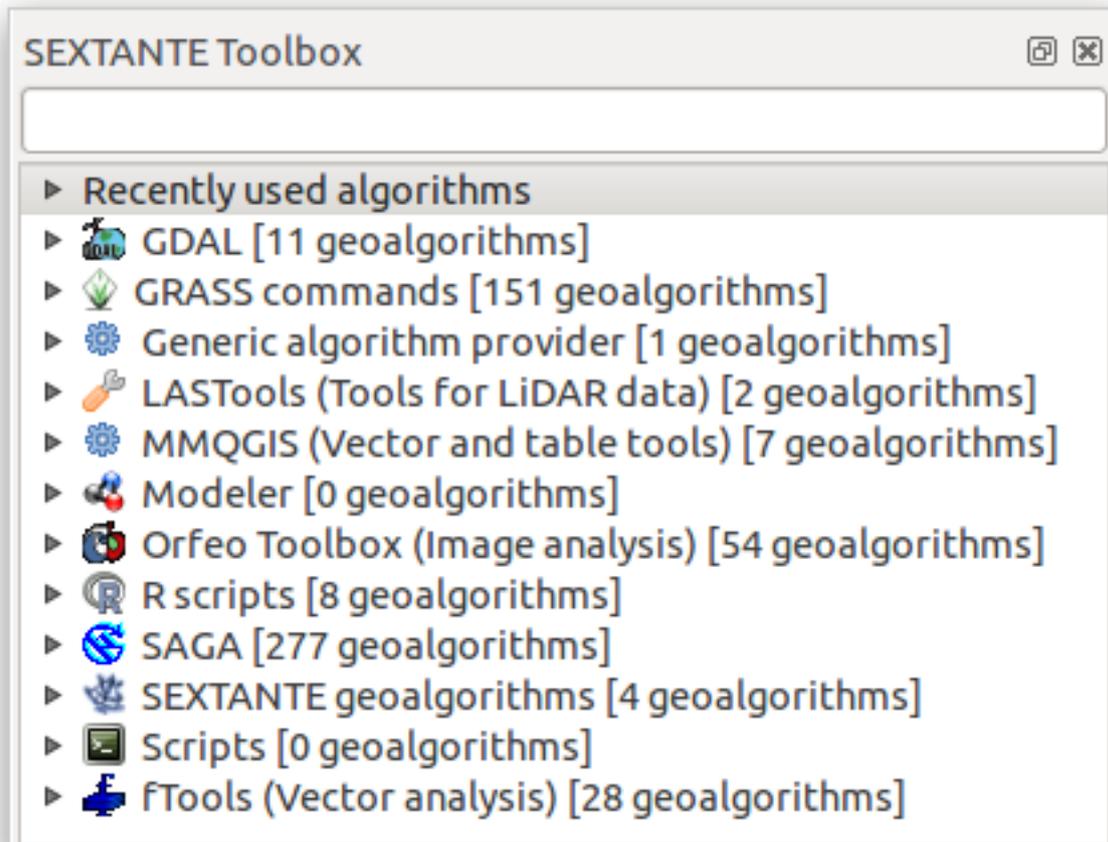


ノート： 速いコンピュータで作業されていない場合は、サンプルデータセットのサイズは、処理に多くの時間を必要としますので、10000 個のサンプル点でこれをやってみることはお勧めしません。

7.4.9 Follow Along: 追加の空間統計ツール

もともとは別のプロジェクト、その後はプラグインとしてアクセスできる、SEXTANTE ソフトウェアはバージョン 2.0 からのコア機能として QGIS に追加されました。その新しい名前 :menuselection: *Processing* で新しい QGIS メニューとして見つかります、そこからは空間分析ツールの豊富なツールボックスにアクセスでき、単一のインターフェース内からの様々なプラグインツールにアクセスできます。

- 処理->*Toolbox* メニューエントリを可能にすることにより、ツールのこのセットをアクティブにします。ツールボックスには、次のようになります。



おそらく、それはマップの右側に QGIS にドッキング表示されます。ここに記載されているツールは、実際のツールへのリンクであることに注意してください。それらのいくつかは SEXTANTE 独自のアルゴリズムであり、他には、GRASS、SAGA またはオルフェオツールボックスなどの外部アプリケーションからアクセスされているツールへのリンクです。既にそれらを利用できるように、この外部アプリケーションは、QGIS と一緒にインストールされています。処理ツールの設定を変更する必要があるか、例えば、外部アプリケーションの一つの新しいバージョンにアップデートする必要がある場合は、処理->オプションと *configurations* からその設定にアクセスできます。

7.4.10 Follow Along: 空間点パターン分析

random_samples データセット中のポイントの空間的な分布を簡単に表示するために、たった今開いた処理 *Toolbox* 経由で SAGA の空間ポイントパターン *Analysis* ツールを利用できます。

- 処理 *Toolbox* 中で、このツールを検索 空間ポイントパターン *Analysis*。
- ダブルクリックしそのダイアログを開きます。

SAGA のインストール

ノート: SAGA がシステムにインストールされていない場合は、プラグインのダイアログによって依存関係が欠落していると通知されます。そうでない場合は、次の手順はスキップできます。

Windows で

お使いの教材に含まれるものは、Windows 用 SAGA インストーラを見つけます。

- プログラムを起動し、Windows システム上で SAGA をインストールするには、その指示に従ってください。インストールされているフォルダのパスをメモしてください!

SAGA をインストールしたら、それが下にインストールされたパスが見つかるよう SEXTANTE を設定する必要があります。

- メニュー項目:menuselection:分析 -> SAGA オプションと *configuration* をクリックしてください。
- 表示されたダイアログで、拡張:guilabel:SAGA アイテムを展開し、SAGA の *folder* を探します。その値は空白になります。
- このスペースに、SAGA をインストールしたパスを挿入します。

Ubuntu で

- :guilabel:検索 で SAGA の GIS ソフトウェア *Center*、または語句を入力しますが端末で:kbd: *sudo apt-get* を使用している *SAGAgis* をインストール。(まず自分のソースに SAGA リポジトリを追加する必要があります。)
- それはすぐに動作しない場合は、QGIS を再起動する必要があるかもしれませんが、QGIS は、自動的に SAGA があります。

Mac で

自作ユーザーは、このコマンドで SAGA をインストールできます。

- *saga-core* の BREW インストール

自作を使用しない場合は、こちらの手順に従ってください。

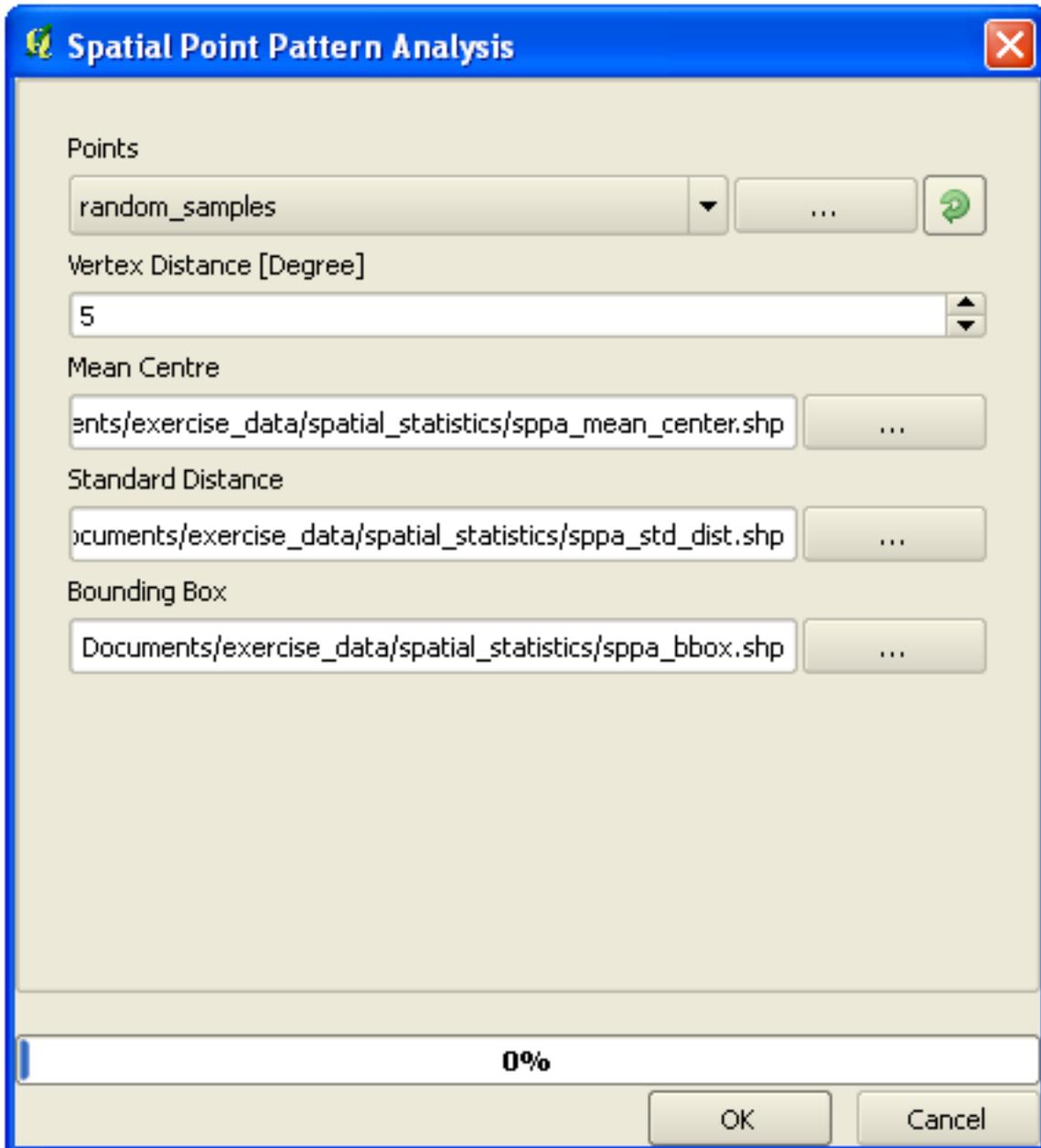
[http://sourceforge.net/apps/trac/saga-gis/wiki/Compiling%20SAGA200 % n%20Mac%20OS % 20X](http://sourceforge.net/apps/trac/saga-gis/wiki/Compiling%20SAGA200%20on%20Mac%20OS%20X)

インストール後

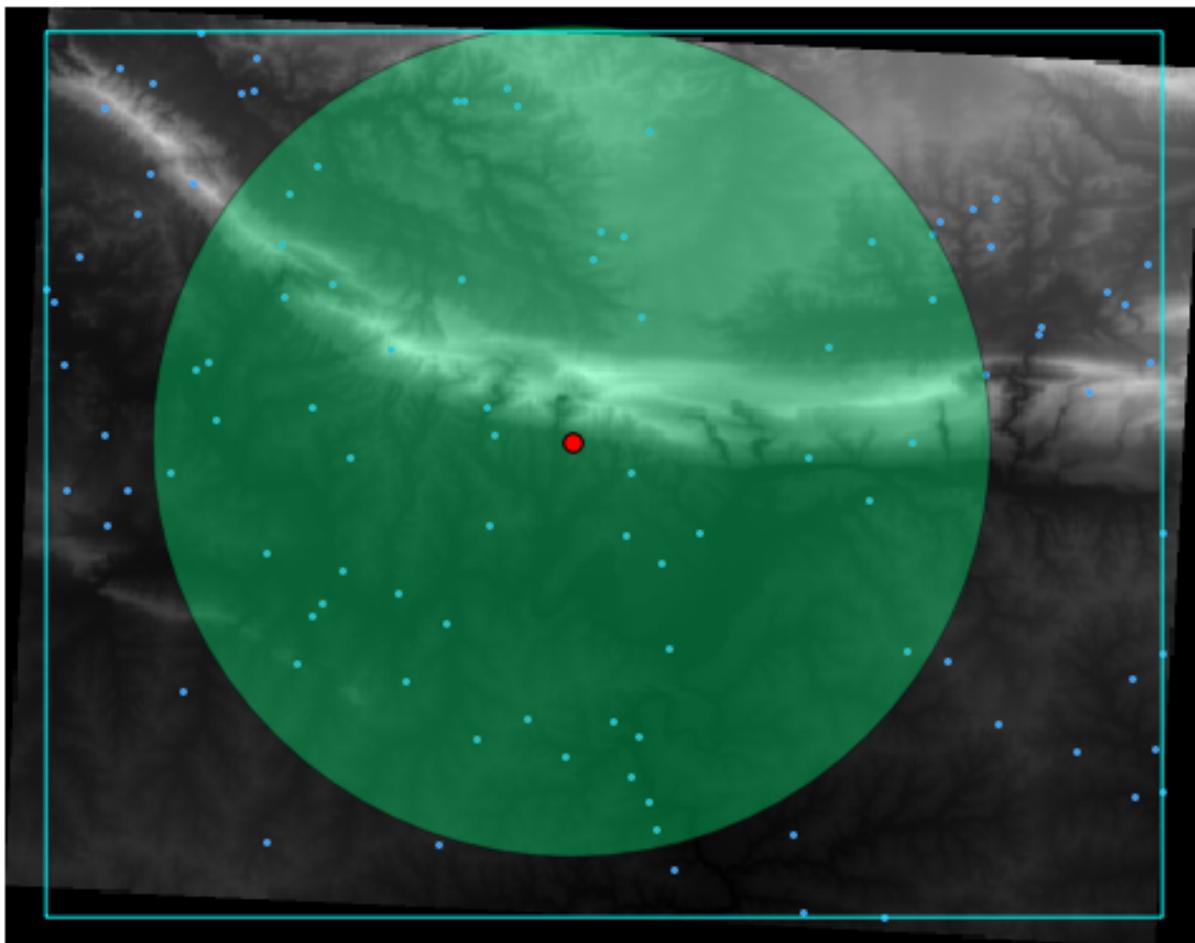
今、インストールされ SAGA が設定されていることを、その機能は、アクセスできるようになります。

SAGA の利用

- SAGA ダイアログを開きます。
- SAGA は、3 つの出力を生成し、したがって 3 つの出力経路を必要とします。
- 何でも適切と思うファイル名を使用して、*exercise_data / spatial_statistics/* 下にこれらの 3 つの出力を保存します。



出力は（記号は、この例のために変更された）次のようになります。



赤い点は平均中心です。大きな円は標準的距離で、点が平均中心の周りにどのくらい近くに分布しているかの指標を与えます。矩形はバウンディングボックスで、なおすべてのポイントを囲むであろう可能な最小の矩形を記述します。

7.4.11 Follow Along: 最小距離分析

多くの場合、このアルゴリズムの出力はシェープファイルではなく、データセットの統計的特性をまとめた表ではありません。これらの一つがある: `gui:label:最小距離分析` ツール。

- このツール: `gui:label:処理 Toolbox` 中で: `gui:label:最小距離 Analysis` として探します。

これは、分析されるベクタポイントデータセットを指定する以外に、他の入力を必要としません。

- `random_points` データセットを選択します。
- `OK` をクリックします。完了時に、DBF テーブルが: `gui:label:レイヤー list` に表示されます。
- それを選択し、その属性テーブルを開きます。数値はこれとは変わっている場合がありますが、結果はこの形式になります。

	NAME	VALUE
0	Mean Average	2823.45817848
1	Minimum	424.0860061
2	Maximum	9773.35250512
3	Standard Deviation	1662.40681133
4	Duplicates	0

7.4.12 In Conclusion

QGIS は、データセットの空間的な統計的性質を分析するための多くの可能性を可能にします。

7.4.13 What's Next?

これでベクトル解析はカバーしましたが、ラスタで何ができるかは見ないのでしょうか。それは次のモジュールでやります！

Chapter 8

Module: ラスタ

我々は以前にデジタイズするためのラスタを使用しましたが、ラスタデータは、直接使用することができます。このモジュールでは、どのようにして QGIS で行うのがわかります。

8.1 Lesson: 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。ベクタデータは頂点で構成され、そしてことによるとラインやエリアとつながる離散的な地物を有します。しかしラスタデータは画像のようなものです。それは多分現実世界のオブジェクトのさまざまな属性を表現しますが、これらのオブジェクトは個別のオブジェクトとして存在していません。むしろ、それらはさまざまに異なる色の値の画素を用いて表現されます。

この単位の間、あなたは既存の GIS 解析を補うためにラスタデータを使用します。

このレッスンの目標: QGIS でラスタデータを操作する方法を学習します。

8.1.1 Follow Along: ラスタデータの読み込み

- Open your `analysis.qgs` map (which you should have created and saved during the previous module).
- `solution` と `important_roads` レイヤを除く全てのレイヤを非アクティブにします。
- ラスタレイヤの追加 ボタンをクリックします。



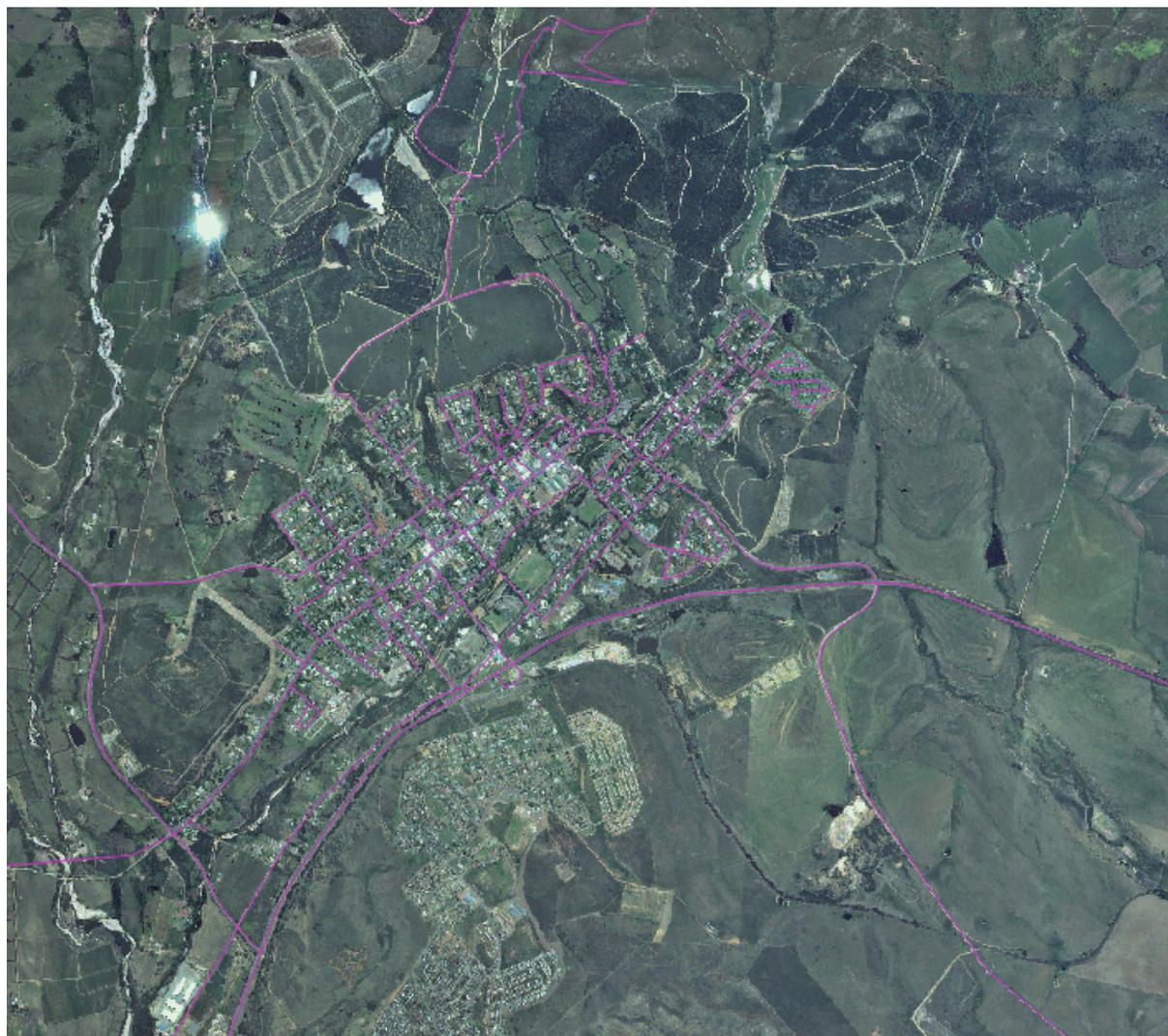
ラスタレイヤの追加 ダイアログが開きます。このプロジェクトのためのデータは `exercise_data/raster` にあります。

- それらをすべて別個に読み込むか、`ctrl` を押しながら 4 つ全てを順にクリックしてから一度に開きます。

はじめはあなたは地図上で何も起こっているように見えないと思うでしょう。ラスタは読み込まれていないのでしょうか? いいえ、それらはレイヤリストにはありますので明らかに読み込まれています。問題はそれらが同じ投影にないということです。幸いなことに、私たちは既にこのような状況で何をすべきかを見てきました。

- メニューの `プロジェクト -> プロジェクトのプロパティ` を選択します:
- メニューの `CRS` タブを選択します:
- “オンザフライ” 再投影を有効にします。
- 他のデータと同じ投影 (`WGS 84 / UTM zone 33S`) に設定します。
- `OK` をクリックします。

ラスタはうまくフィットするはずで:



私たちの全体の研究領域をカバーする 4 つの航空写真があります。

8.1.2 Follow Along: 仮想ラスタの作成

これからわかるように、あなたの solution レイヤは 4 つのすべての写真にわたっています。これが意味することはあなたは 4 つのラスタで作業する必要があるということです。それは理想的ではありません。1 つの (コンポジット) イメージに対して 1 つのファイルになっている方が良いと思うでしょう?

幸いなことに、QGIS ではこれを行うことができます。それも新しいラスタファイルを実際に作成して、多くのスペースを取ることなしに。あなたは 仮想ラスター を作成することができます。これはよく カタログ とも呼ばれることがあります、その名前はその機能を説明しています。それは本当に新しいラスタではありません。既存のラスタを 1 つのカタログ、つまりアクセスを容易にするための一つのファイルに構成する方法です。

カタログを作成するには:

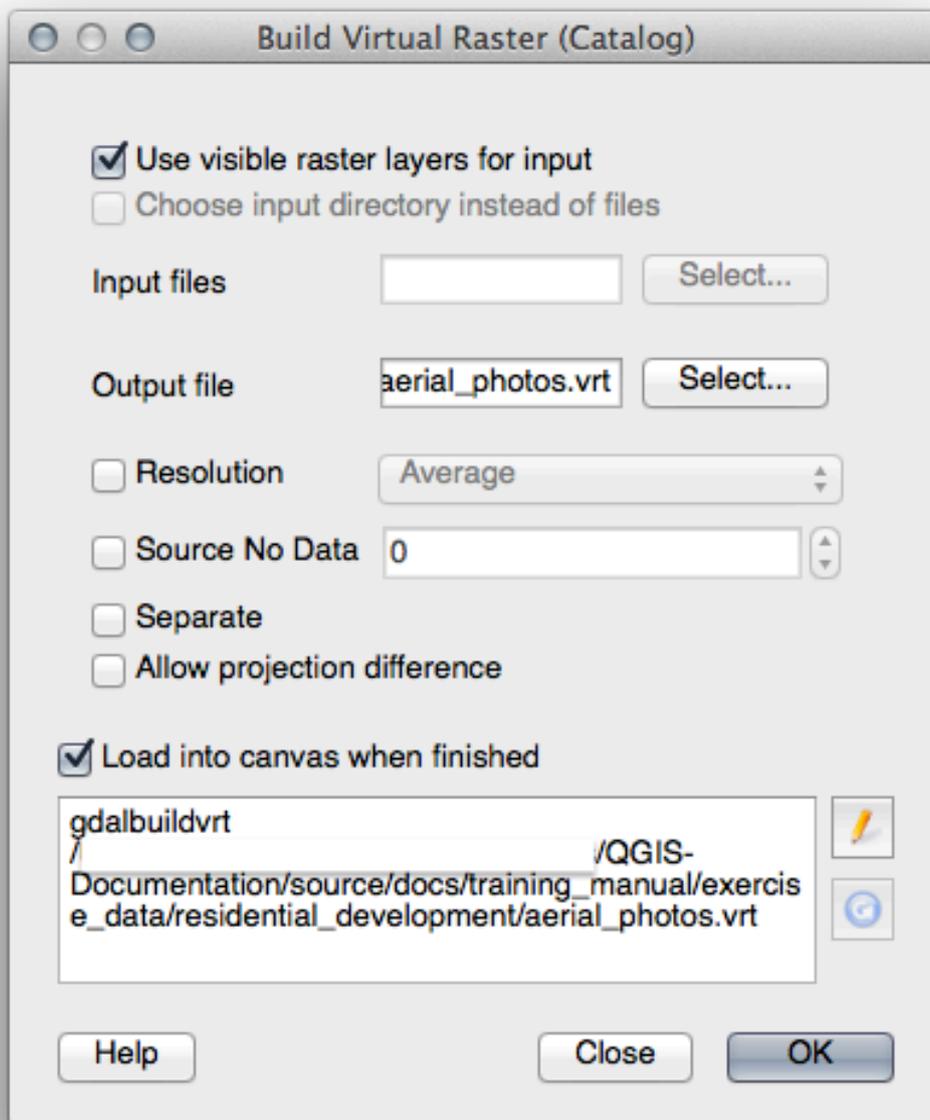
- メニュー項目 **ラスタ** → **その他** → **バーチャルラスタの構築 (カタログ)** をクリックします。
- 表示されるダイアログで、入力に可視のラスタレイヤを使うの隣のチェックボックスをオンにします。
- 出力場所に `exercise_data/residential_development` を入力します。
- ファイル名に `aerial_photos.vrt` を入力します。

- 終了時にキャンバスにロードします ボタンをチェックします。

下にあるテキストフィールドに注目してください。このダイアログが実際にやっているのは、あなたのためにそのテキストを書くことです。それは QGIS が実行しようとしている長いコマンドです。

ノート:  コマンドテキストが編集可能であることを留意して下さい。もし望むなら、さらにコマンドをカスタマイズすることができます。初期コマンド(この場合は、gdalbuildvrt)の構文についてのヘルプはオンラインで検索して下さい。

- OK をクリックしてコマンドを実行します。



それが完了するまでに時間がかかる場合があります。終わったらメッセージボックスでそう教えてくれます。

- OK をクリックしてメッセージボックスを閉じます。

- 仮想ラスタの構築 (カタログ) ダイアログの 閉じる をクリックします。(もう一度コマンドの実行が開始されてしまうので OK はクリックしないで下さいね。)
- これで レイヤーリスト から元の 4 つのラスタを削除することができます。
- 必要であれば、新しい *aerial_photos* ラスタカタログレイヤをクリックして、レイヤーリストの一番下にドラッグすると他のアクティブなレイヤが見えるようになります。

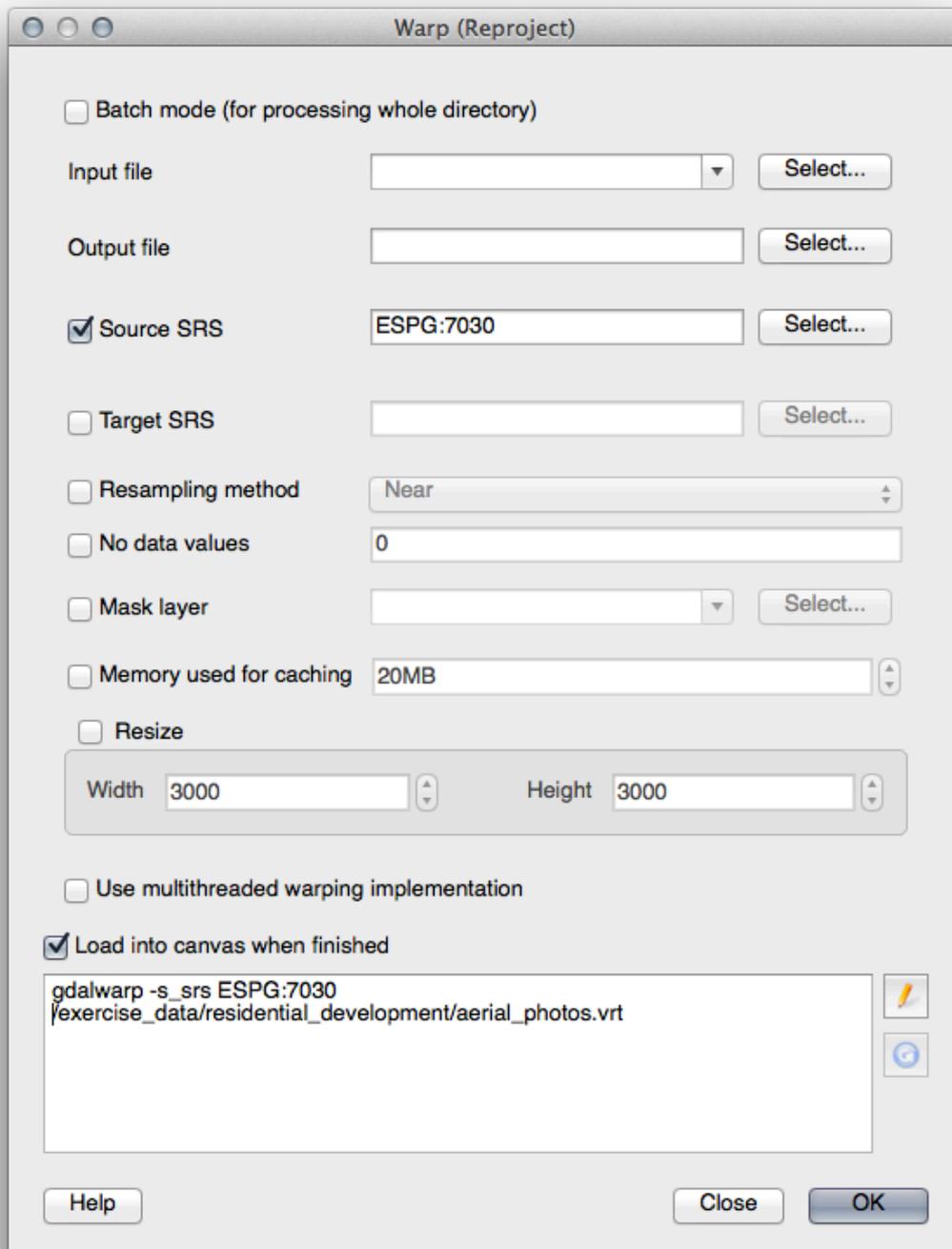
8.1.3 ラスタデータの変換

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”オンザフライ”に再投影することができます。しかし、もしあなたが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。

ラスタの再投影

- メニュー項目 ラスタ → 投影法 → ワープ (再投影) をクリックします。

このツールはディレクトリ全体の内容を再投影するための便利なバッチオプションを備えています。仮想ラスタ (カタログ) も再投影することができ、マルチスレッド処理モードを有効にすることもできます。

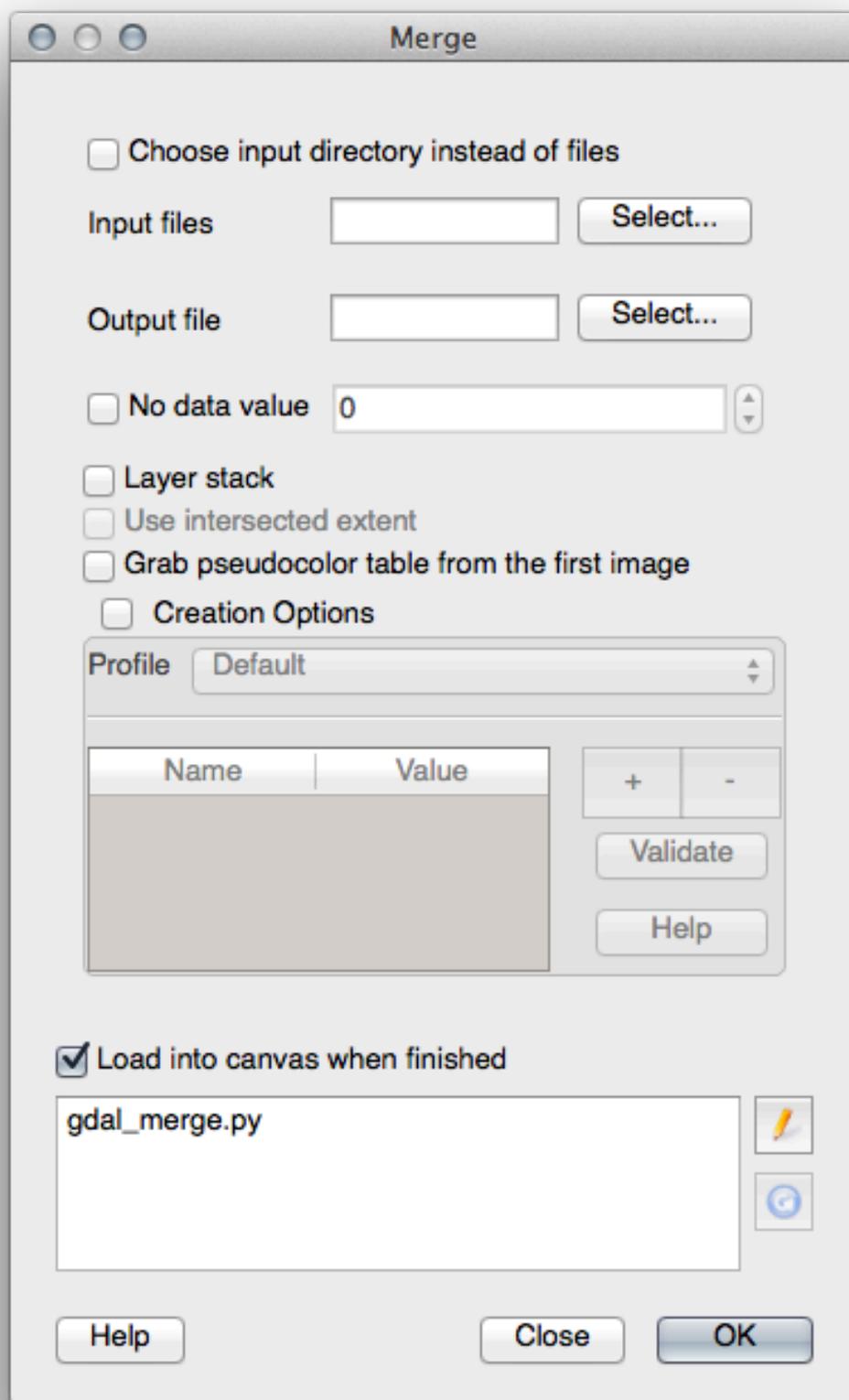


ラスタのマージ

- メニュー項目 ラスタ → その他 → 結合 をクリックします。

あなたは単一のファイルのかわりにディレクトリ全体を処理することもできます。とても便利なバッチ処理機能です。入力ファイルとして仮想ラスタを指定することもでき、全ての構成ラスタが処理されます。

また、生成オプション チェックボックスとリストを使用して独自のコマンドラインオプションを追加することができます。あなたがGDALライブラリの操作についての知識を持っている場合にのみ利用できます。



8.1.4 In Conclusion

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。

8.1.5 What's Next?

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ライズはどのように有用であるかを見ていきます。

8.2 Lesson: ラスタシンボロジーの変更

ラスターデータは航空写真ばかりではありません。ラスターデータには他にも多くの形態があり、それらの多くの場合ではデータが適切に表示されて役立つように適切にシンボライズすることが不可欠です。

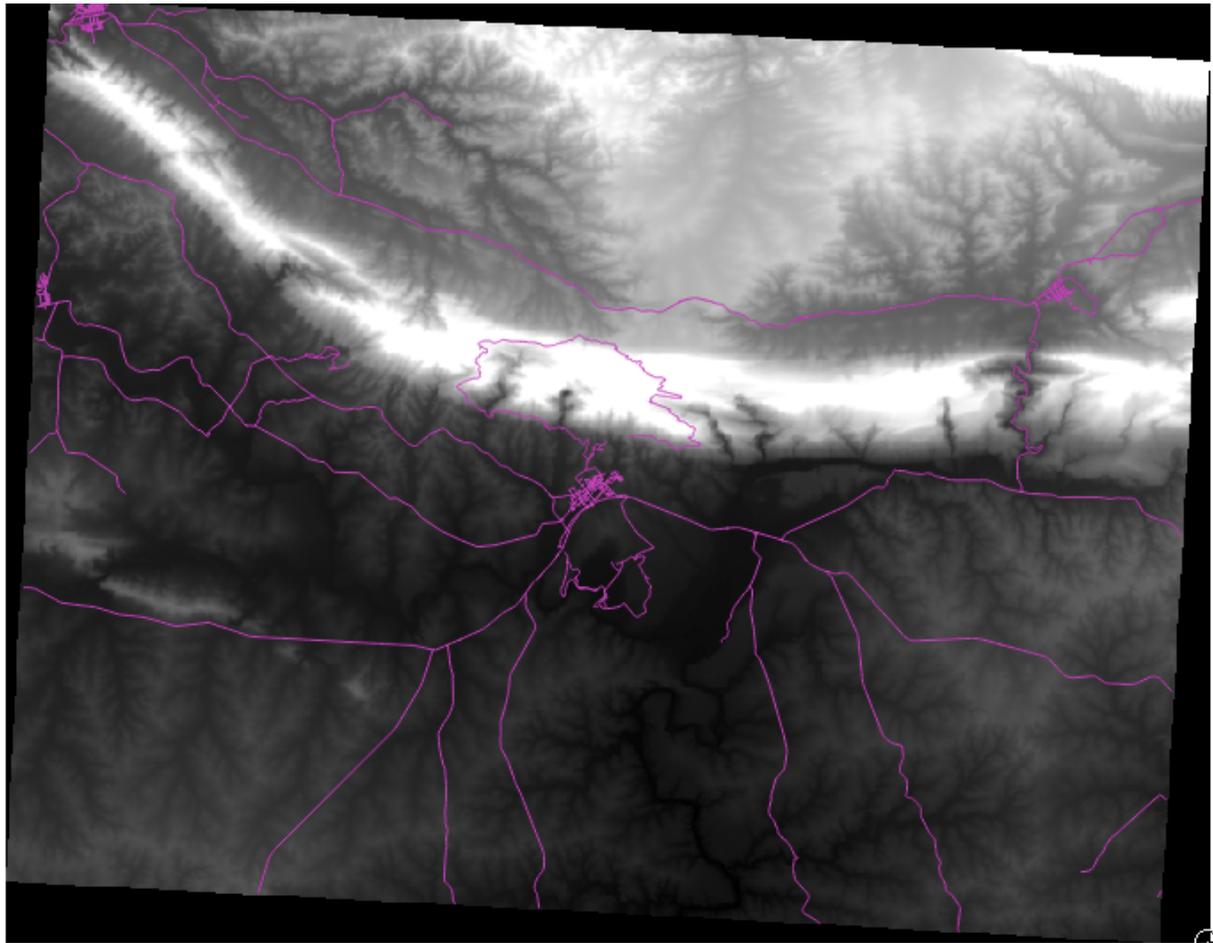
このレッスンの目標: ラスタレイヤのシンボロジーを変更します。

8.2.1 Try Yourself

- 前の実習で作成したカレントの地図 `analysis.qgs` から始めます。
- 新しいラスターデータセットをロードするには *Add Raster Layer* ボタンを使います。
- データセット `srtm_41_19.tif` を読み込みます。 `exercise_data/raster/SRTM/` ディレクトリにあります。
- *Layers list* に表示されたらその名前を `DEM` に変更します。
- このレイヤの領域にズームします。レイヤリストを右クリックしてレイヤの領域にズームを選択します。

このデータセットは 数値標高モデル (*DEM*) です。それは地形の標高を表す地図で、例えば山や谷はどこにあるか知ることができます。

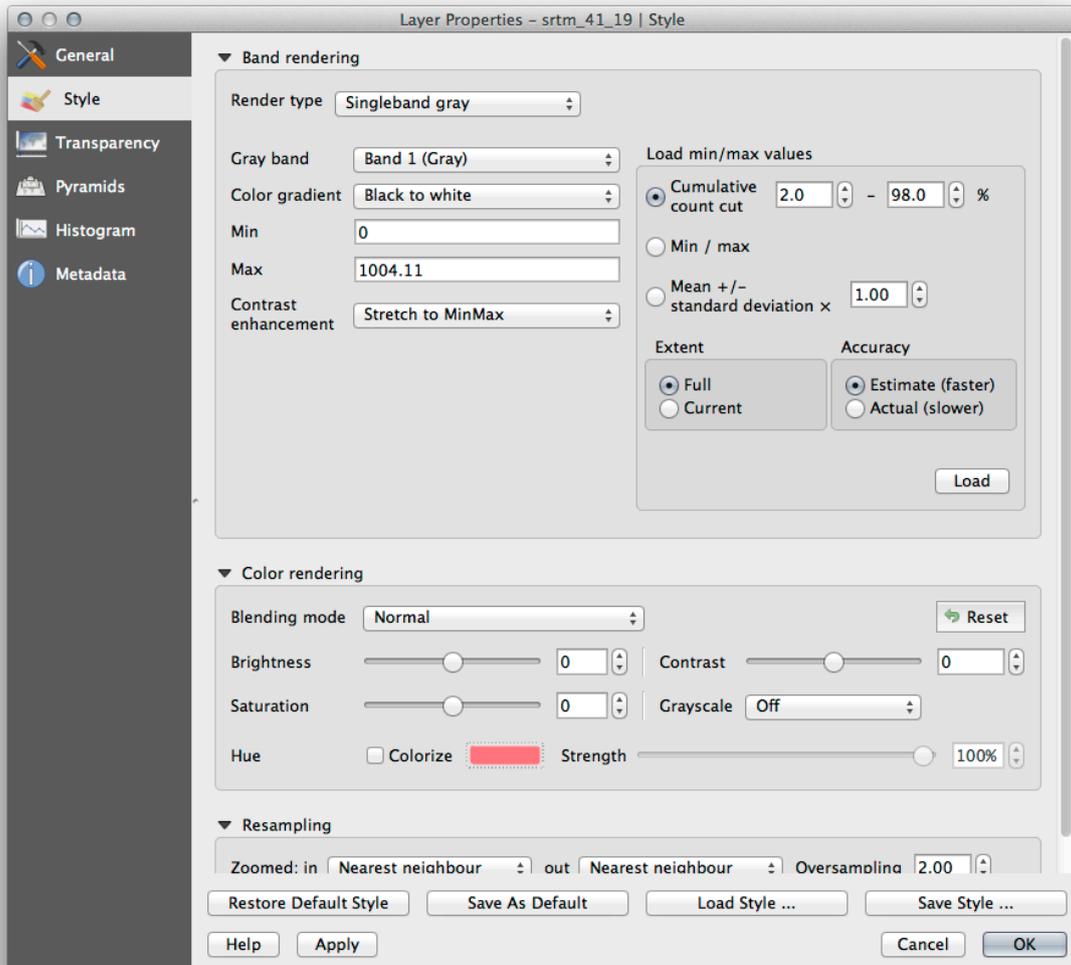
それがロードされると、*DEM* の基本的な引き伸ばされたグレースケール表現が表示されます。上にベクトルレイヤを重ねるとこのように表示されます:



QGIS は可視化のためにイメージに引きのばしを自動的に適用しました。引き続き私たちはこれがどのように機能するかについて詳しく学びます。

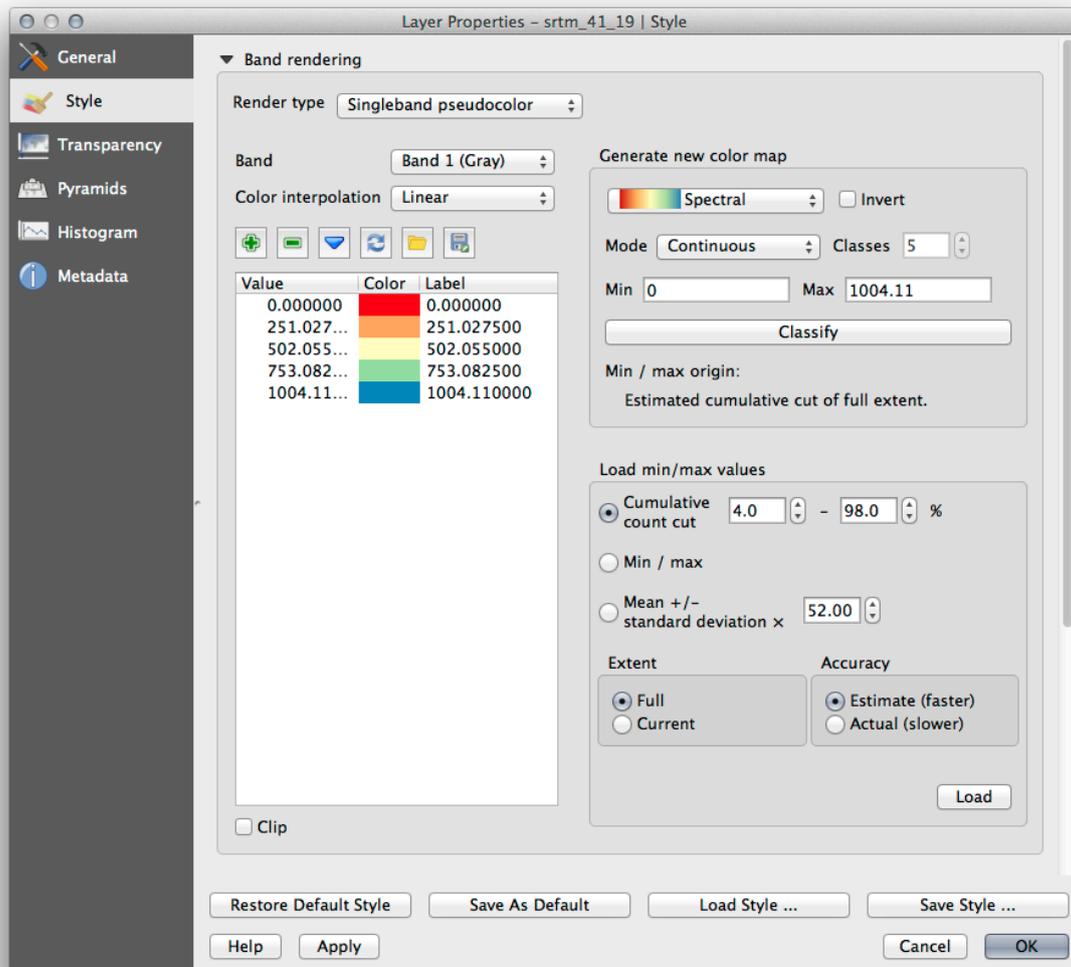
8.2.2 Follow Along: ラスタレイヤのシンボロジーの変更

- *SRTM* レイヤの レイヤプロパティ ダイアログを開きます。レイヤツリーのレイヤを右クリックしてプロパティ を選択します。
- スタイル タブに移ります。

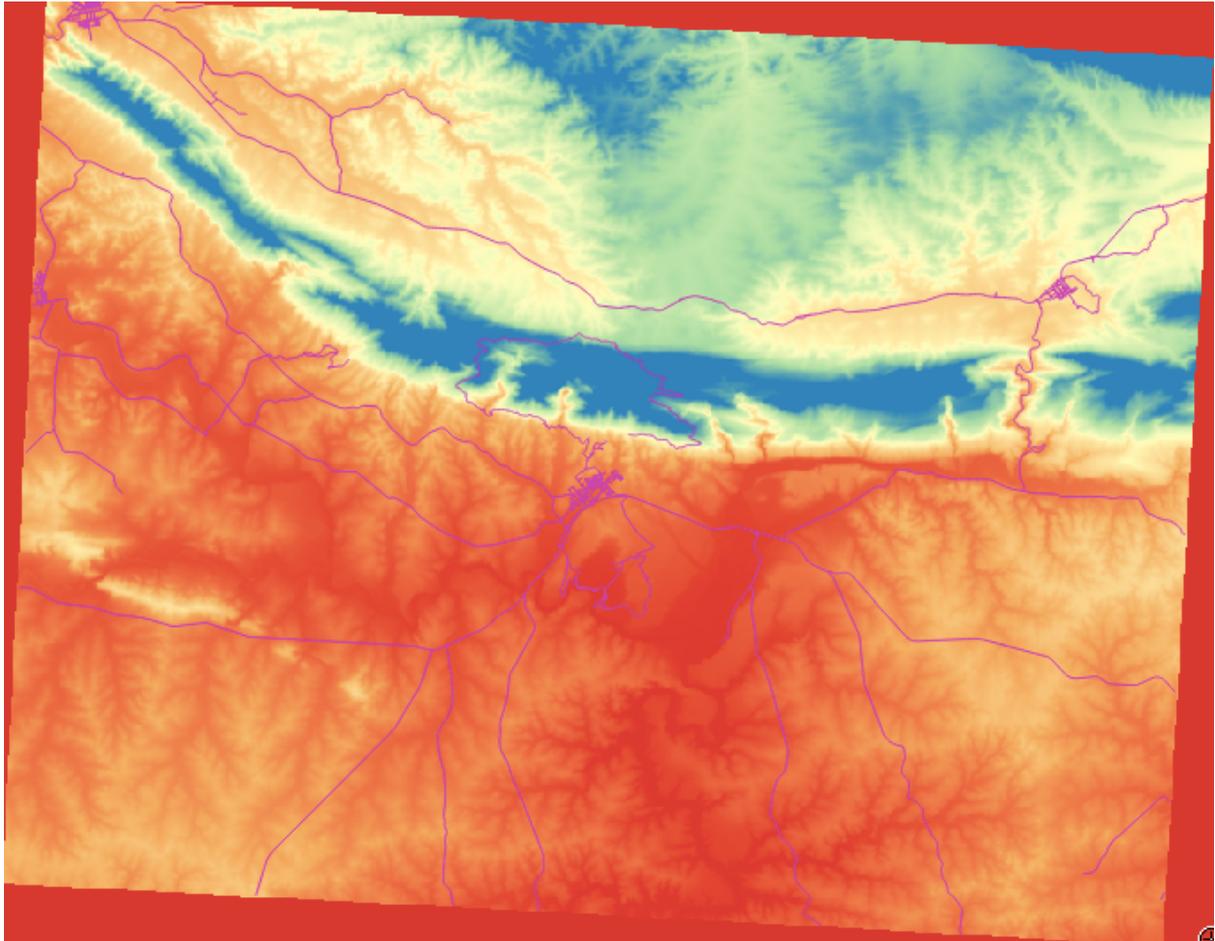


現在の設定は QGIS がデフォルトで適用したものです。それは DEM の表示の一つの方法にすぎません。では他の方法を探りましょう。

- レンダータイプを単バンド疑似カラーに変更します。与えられたデフォルトオプションを使います。
- 分類 ボタンをクリックして新しい色の分類を生成します。OK をクリックして DEM にこの分類を適用します。



このようにラスタが見えるでしょう。



これは DEM を表示する興味深い方法ですが、多分私たちはこれらの色を使ってシンボライズしたいとは思いません。

- もう一度 レイヤプロパティ ダイアログを開きます。
- レンダータイプ を 単バンドグレー に戻します。
- *OK* をクリックしてラスタにこの設定を適用します。

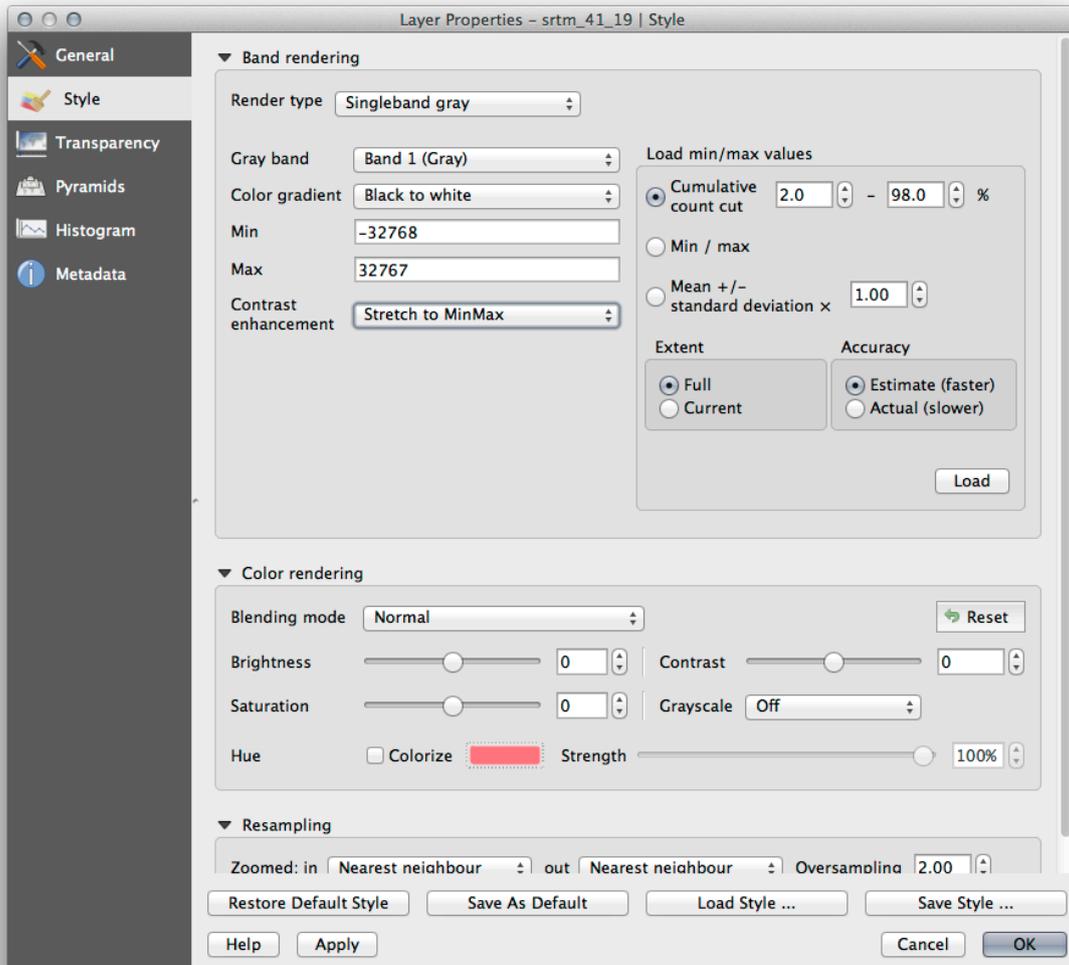
あまり有用ではないすべて灰色の四角形が表示されます。



これはコントラストをつけるために色の値を”引きのばす”デフォルト設定を失ったからです。

それではもう一度 DEM のデータの範囲に基づいて色の値を”引きのばす”ように QGIS に指示しましょう。QGIS に使用可能な全ての色を使わせませ (Grayscale の、黒と白とそれらの間の灰色のすべての色合い)。

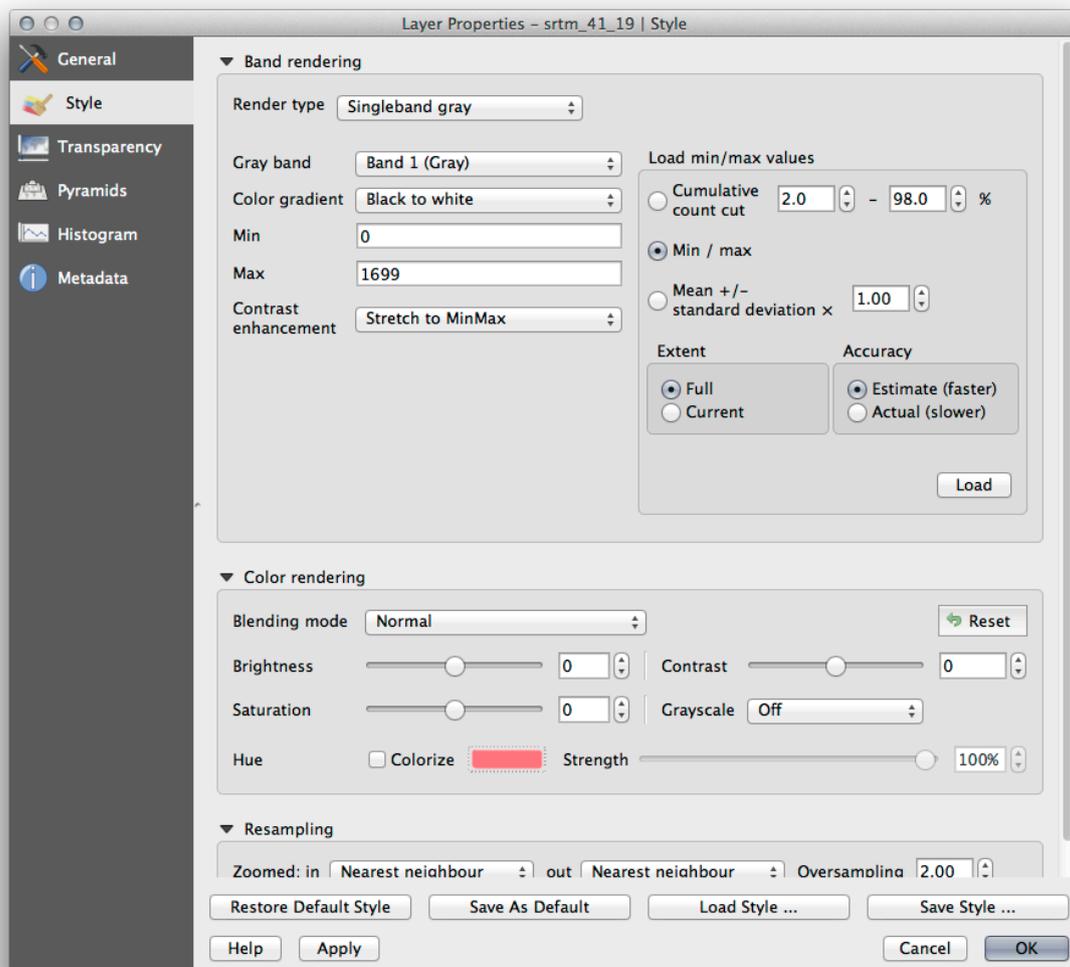
- 最小と最大の値を次に示す通りに指定します。
- コントラスト拡張を最小最大に引き延ばすに設定します:



しかし、引き延ばしのために使用されるべき最小値と最大値とは何でしょうか？ 最小 と 最大 の値の下に現在あるものは前に私たちに灰色の矩形を与えた値と同じです。そのかわりに私たちは画像の中にある実際の最小値と最大値を使用するべきでしょうか？ 幸いなことに、ラスタの最小値と最大値をロードすることによって、これらの値は簡単に決定することができます。

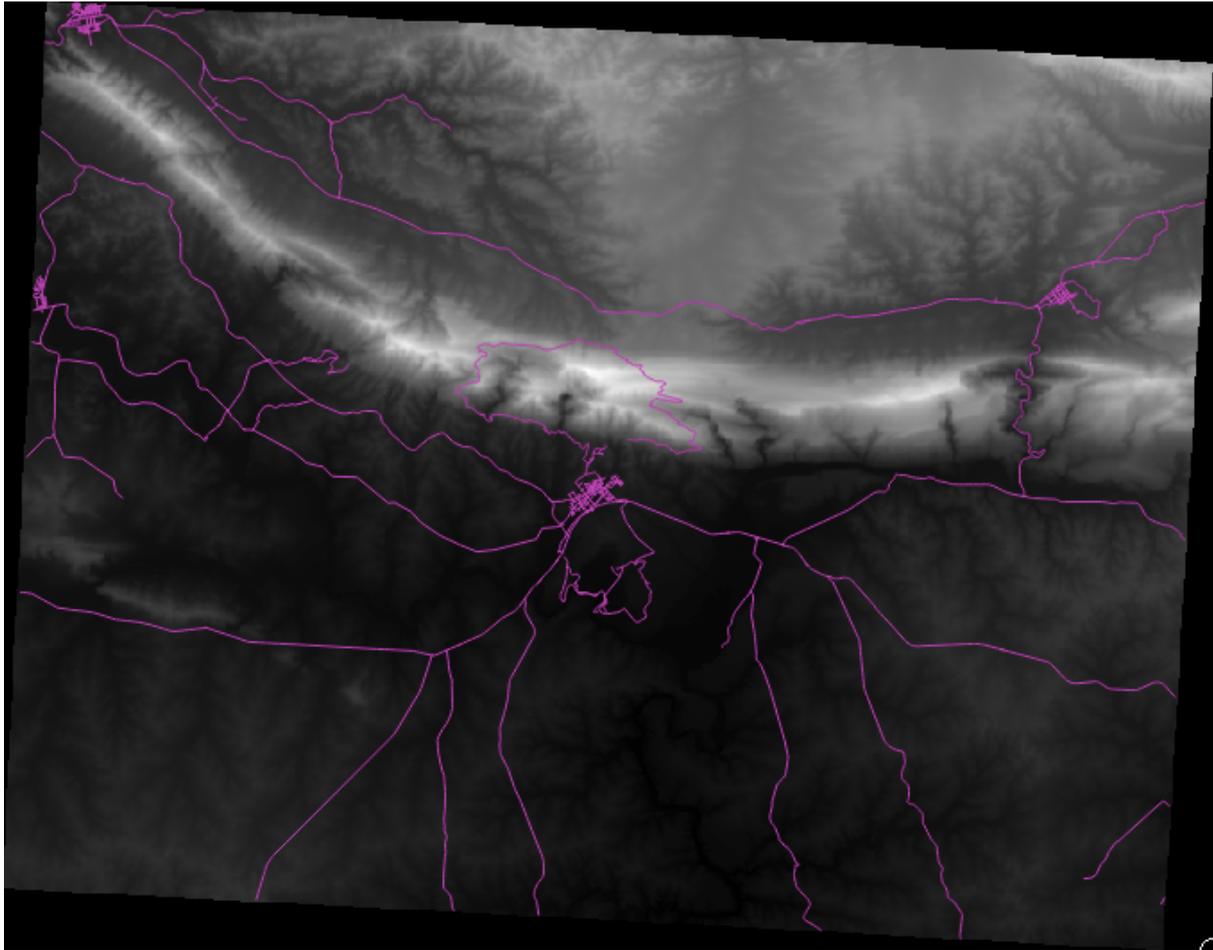
- 最小値/最大値のロード の下の 最小/最大 オプションを選択します。
- 読み込み ボタンをクリックします:

Custom min / max values が DEM の実際の値を反映するように変わったことに注目して下さい:



- *OK* をクリックしてこれらの設定をイメージに適用します。

これでラスタの値は再び適切に表示されます。より暗い色は谷を、より明るい色は山を表しています。



でも、もっといい、もっと簡単な方法はないでしょうか？

はい、あります。何がされる必要があるのかをあなたは理解しているので、このすべてを簡単に行うためのツールがあることを知って嬉しくなるでしょう。

- レイヤリスト から今ある DEM を削除します。
- ラスタをもう一度ロードして、前と同じように DEM に名前を変えます。再び灰色の矩形が現れました...
- ビュー → ツールバー → ラスタ を有効にして必要なツールを有効化します。これらのアイコンがインターフェイスに現れるでしょう。



左から 3 番目のボタン *Local Histogram Stretch* はあなたがズームしている地域に最もよいコントラストを与えるために自動的に最小値と最大値を引きのばします。これは大規模なデータセットの場合に便利です。左のボタン 表示範囲内でヒストグラムの累積的カットと引き伸ばし... は画像全体にわたる一定値に最小値と最大値を引きのばします。

- 左から 4 番目のボタン (ヒストグラムをフルデータセットに伸ばす) をクリックします。データが前のように正しく表示されます。

このツールバーの他のボタンを試してみて、ローカルエリアにズームインした時や完全にズームアウトした時に画像の引きのばしがどのように変わるかを見てみましょう。

8.2.3 In Conclusion

これらはあなたがラスタシンボロジーを使い始めるための基本的な機能にすぎません。QGIS は標準偏差を用いたレイヤのシンボロジーやマルチスペクトル画像の異なるバンドを異なる色で表現することもできます。

8.2.4 リファレンス

SRTM データセットは <http://srtm.csi.cgiar.org/> から入手しました。

8.2.5 What's Next?

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。

8.3 Lesson: 地形解析

ラスタの特定の種類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) はこの点で特に有用です。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。

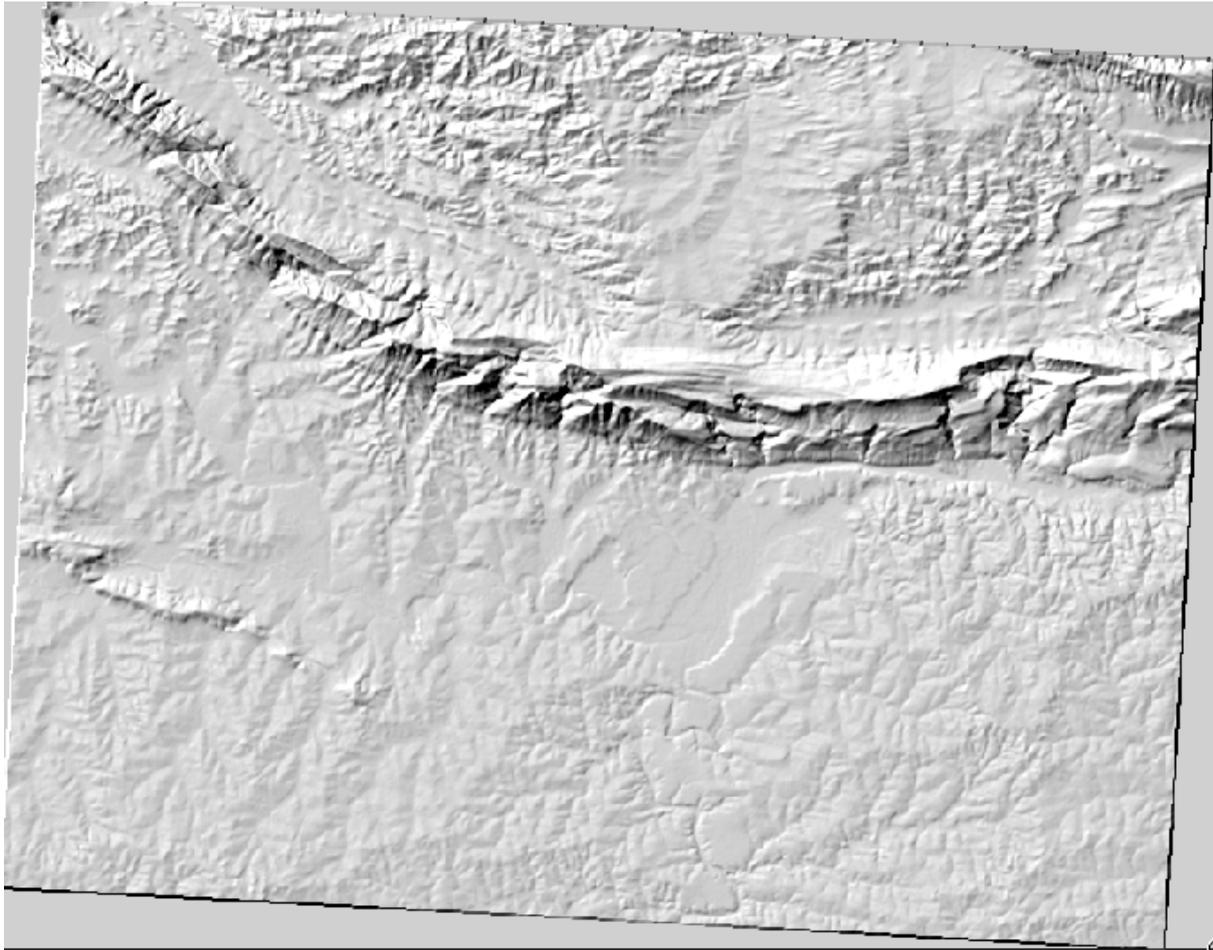
8.3.1 Follow Along: 陰影起伏の計算

マップ上に今ある DEM は地形の標高を示していますが、それは少し抽象的に見えることがあるかもしれません。それはあなたが必要とする地形に関するすべての 3D 情報を含んでいますが、3D オブジェクトのようには見えません。地形がより良く見えるようにするために、陰影起伏を計算することが可能です。それは 3D に見える画像を作成するために光と影を使って地形を表すラスタです。

DEM を扱うには QGIS のオールインワン DEM (地形モデル) 解析ツールを使用します。

- メニュー項目 ラスタ → 解析 → DEM (地形モデル) をクリックします。
- 表示されるダイアログで 入力ファイル が DEM であることを確認します。
- 出力ファイル を `exercise_data/residential_development` ディレクトリの `hillshade.tif` に設定します。
- モード オプションで 陰影図 が選択されていることも確認します。
- 終了時にキャンバスにロードします の隣にあるボックスをチェックします。
- 他のオプションは変更しないままにしておけばいいです。
- OK をクリックして陰影図を生成します。
- 処理が完了したら、OK をクリックしてメッセージを閉じます。
- DEM (地形モデル) のメインダイアログの 閉じる をクリックします。

`hillshade` と呼ばれる新しいレイヤが次のように表示されます:

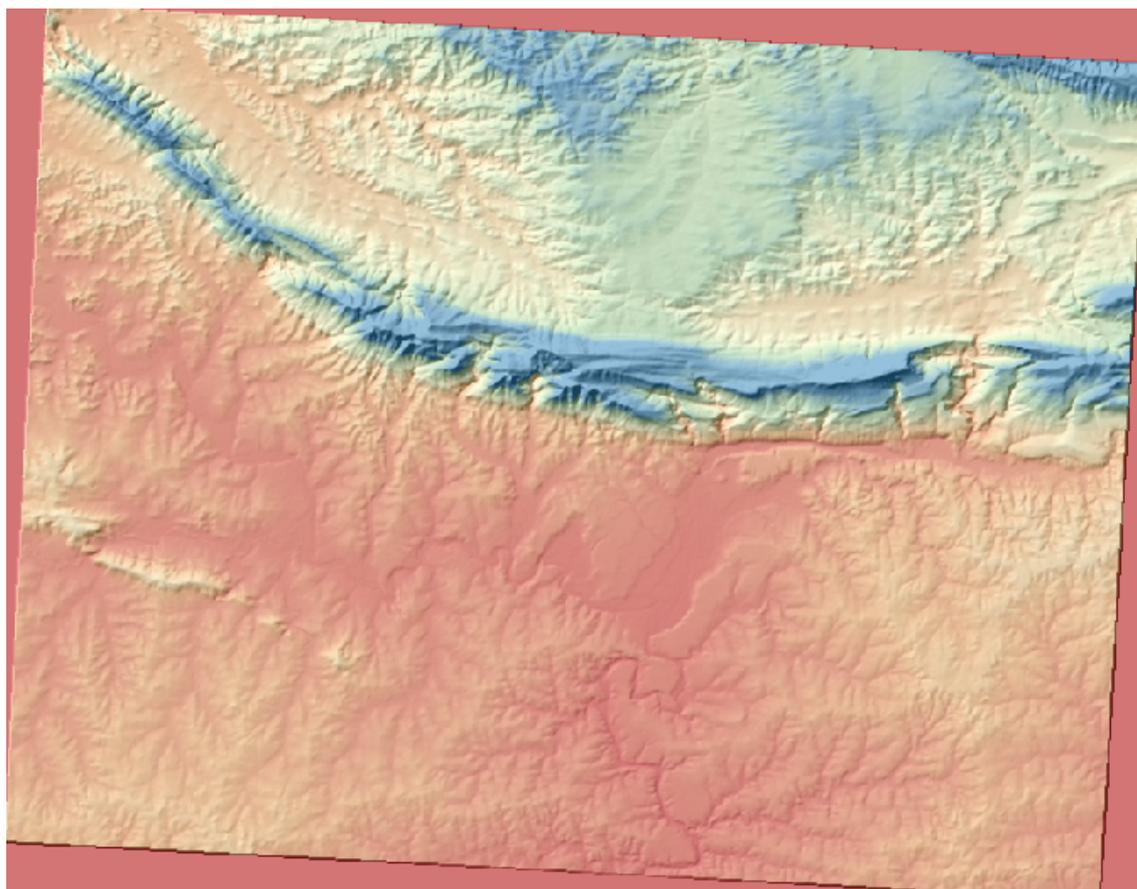


それは素晴らしく 3D に見えますが、私たちはこれを改善することができますか? 陰影図はそれだけで石膏模型のように見えます。どうにかしてそれを他のよりカラフルなラスタと一緒に使用することができないでしょうか? もちろんできます。オーバーレイとして陰影図を使用します。

8.3.2 Follow Along: 陰影図のオーバーレイとしての使用

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。

- 前の演習と同様に、オリジナルの *DEM* のシンボロジを 疑似カラー スキーマを使うように変更します。
- *DEM* と *hillshade* レイヤを除くすべてのレイヤを非表示にします。
- レイヤリストで *DEM* をクリックして *hillshade* レイヤの下になるようにドラッグします。
- *hillshade* レイヤを透過させるために レイヤプロパティ を開き、透過性 タブに移動します。
- 全体の透過率を 50% に設定します:
- レイヤプロパティ ダイアログの *OK* をクリックします。このような結果を得るでしょう:



- それをもたらした違いを見るために レイヤリストの *hillshade* レイヤの非表示にして、元のように表示させます。

このように陰影図を使用すると景観のトポグラフィを誇張することが可能です。その効果があなたにとって十分な強さだと思えない場合には、*hillshade* レイヤの透過度を変更すればいいですが、もちろん、陰影起伏がより明るくなるほど、その背後の色は薄暗くなります。ちょうど良いバランスを見つける必要があります。

完了したら地図を保存することを忘れないで下さい。

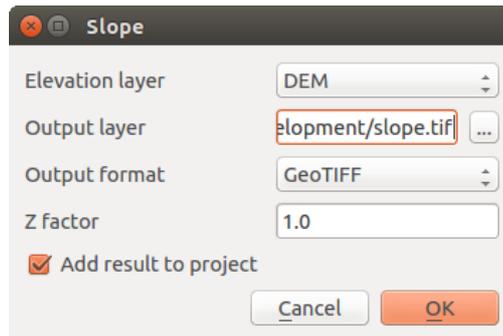
ノート: 次の2つの演習では新しい地図を使用して下さい。それにはDEM ラスタデータセットだけを読み込みます (`exercise_data/raster/SRTM/srtm_41_19.tif`)。これはラスタ解析ツールを使う作業を簡単にするためです。地図を `exercise_data/raster_analysis.qgs` として保存します。

8.3.3 Follow Along: 傾斜の計算

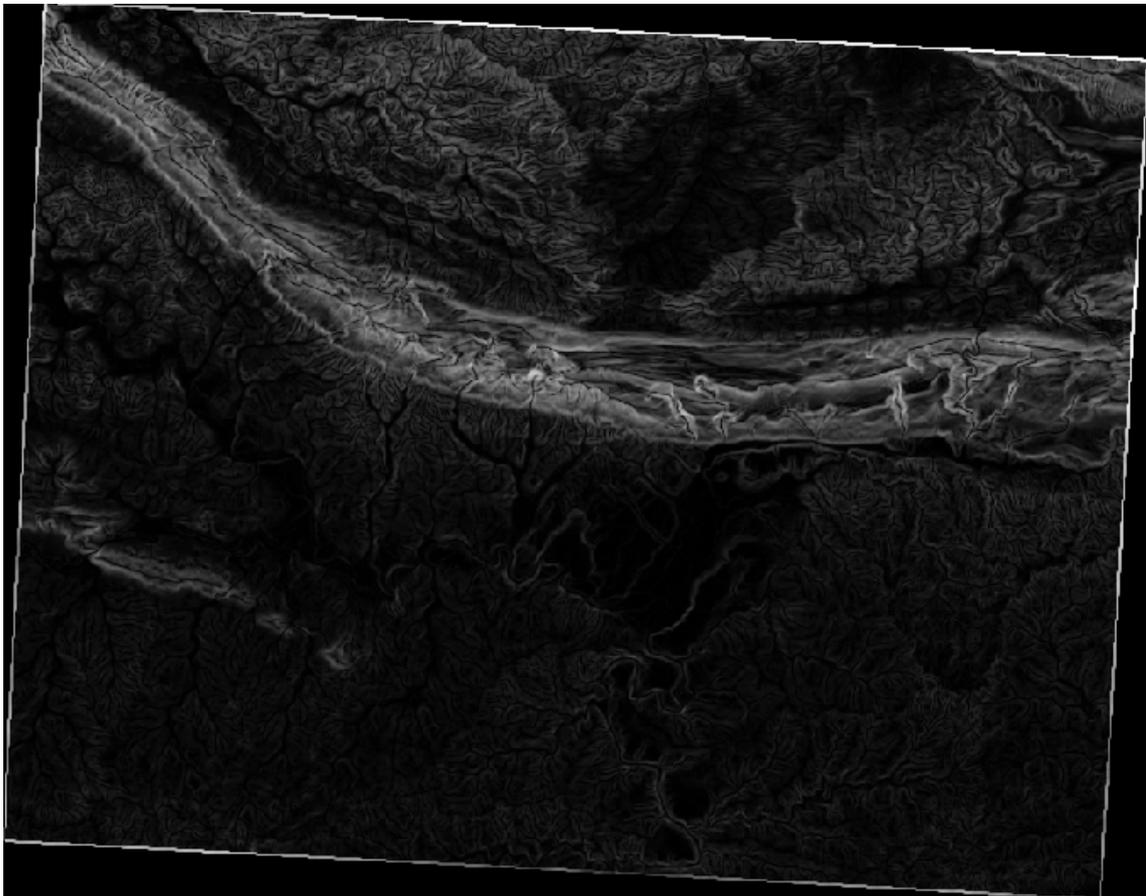
地形の傾斜がどれほど急であるかを知るのも有用なことです。たとえば、あなたが土地に家を建てたい場合には比較的平坦な土地を必要とします。

地形の傾斜を知るには *DEM* (地形モデル) の 傾斜 モードを使用します。

- 前と同じようにツールを開きます。
- モード オプションで 傾斜 を選択します:



- 保存場所を `exercise_data/residential_development/slope.tif` に設定します
- 終了時にキャンバスに... チェックボックスをオンにします。
- *OK* をクリックし、処理が完了したら *閉じる* をクリックしてダイアログを閉じます。あなたの地図に新しいラスタが読み込まれて表示されます。
- レイヤリストで新しいラスタを選択して *ヒストグラムをフルデータセットに伸ばす* ボタンをクリックします。そうすると地形の傾斜が、平坦な地形は黒のピクセルで、急な地形は白のピクセルで表示されます:



8.3.4 Try Yourself 斜面方位の計算

地形の斜面方位はそれが面する方位を指します。この研究は南半球で実施されているので、建物は日当たりがよいように理想的には北向き斜面に建てるのが好ましいです。

- *DEM* (地形モデル) ツールの *斜面方位* モードを使って地形の斜面方位を計算します。

結果をチェックする

8.3.5 Follow Along: ラスタ計算機の使用

不動産仲介業者の問題を振り返ってみましょう。この前 ベクタ分析 レッスンの中で取り組みました。買手は建物を購入してその地所に小さなコテージを建てたいとしましょう。私たちは南半球では開発に理想的な小地所は北向きで傾斜が5度未満の場所であることを知っています。しかし、もし傾斜が2度未満の場合、斜面方位は重要ではありません。

幸いにも、あなたは既に斜面方位だけでなく傾斜を示すラスタを持っていますが、両方の条件が同時に満たされている場所を知る方法がありません。この分析はどのように行うことができるでしょうか？

その答えはラスタ計算機です。

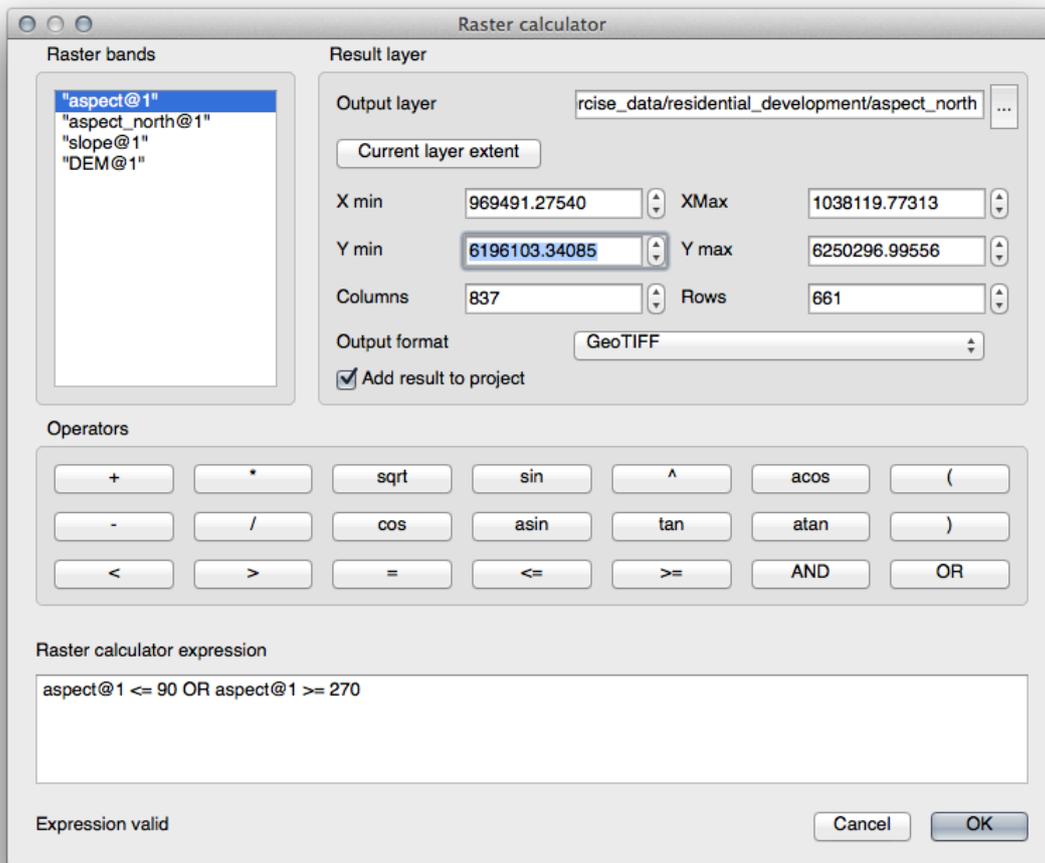
- このツールを起動するには ラスタ > ラスタ計算機... をクリックします。
- *aspect* データセットを利用するために左側にある ラスタバンド リストの *aspect@1* アイテムをダブルクリックします。それは下の ラスタ演算式 テキストフィールドに現れます。

北が0(ゼロ)度ですので、地形が北に面するには斜面方位は270度より大きく、90度未満でなければいけません。

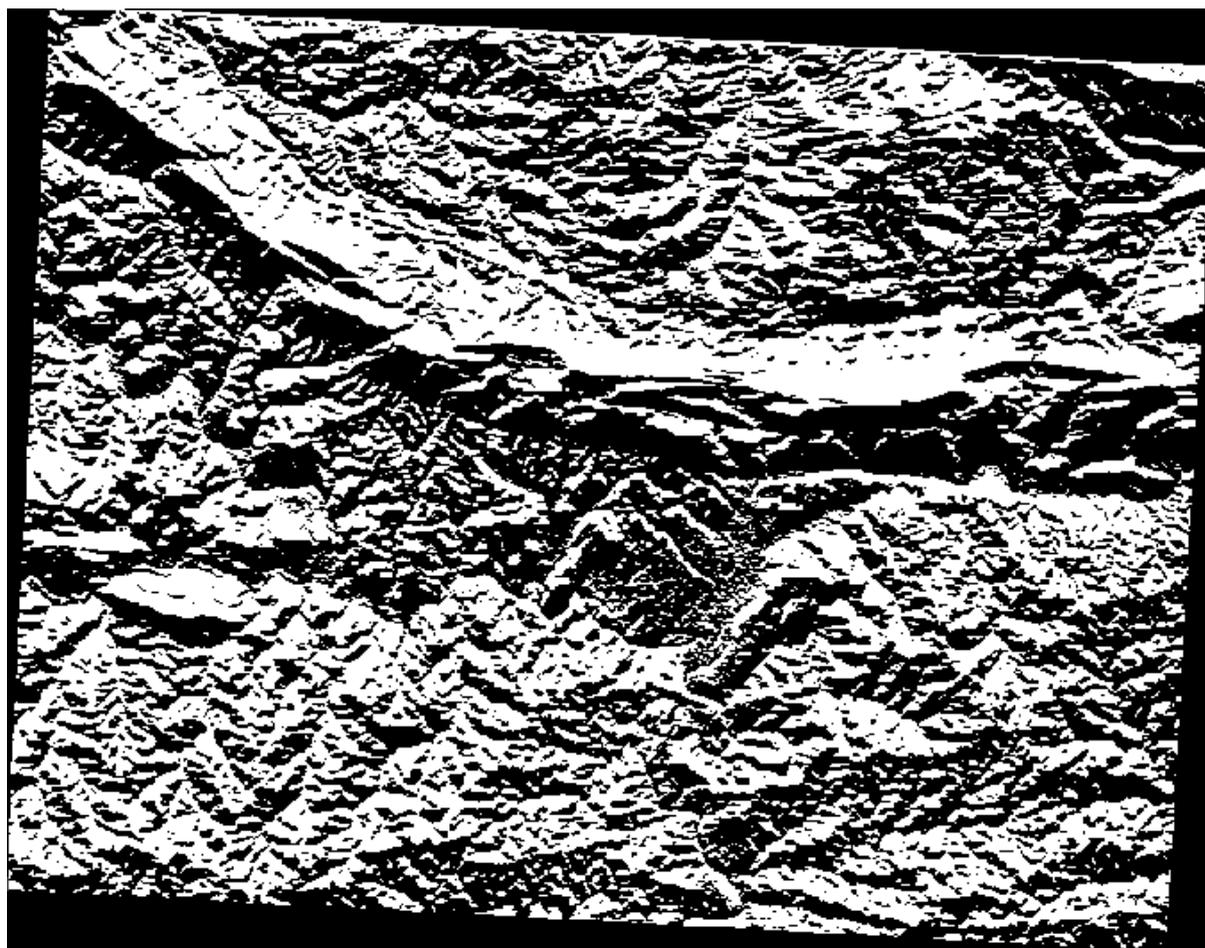
- ラスタ演算式 フィールドにこの式を入力します:

```
aspect@1 <= 90 OR aspect@1 >= 270
```

- 出力 ファイル を `exercise_data/residential_development/` ディレクトリの `aspect_north.tif` に設定します。
- 結果をプロジェクトに追加する ボックスがチェックされていることを確認します。
- *OK* をクリックして処理を開始します。



あなたの結果はこのようになります:



8.3.6 Try Yourself

斜面方位の次は *DEM* レイヤの新しい 2 つの独立した分析を行います。

- 1 番目は斜面傾斜が 2 度以下のすべての地域を識別します。
- 2 番目は似ていますが、傾斜が 5 度以下です。
- `exercise_data/residential_development/` に `slope_lte2.tif` と `slope_lte5.tif` として保存します。

結果をチェックする

8.3.7 Follow Along: ラスター解析結果を組み合わせる

今、あなたは *DEM* レイヤから作られた 3 つの新しいラスタを持っています:

- `aspect_north`: 地形が北に面している
- `slope_lte2`: 斜面傾斜が 2 度以下
- `slope_lte5`: 斜面傾斜が 5 度以下

これらのレイヤでは条件が満たされている場所は 1 に等しく、そうでない場所は 0 に等しいです。したがって、これらのラスタの 1 つを他の 1 つと乗算すれば両方が 1 に等しい場所が得られます。

満たされるべき条件は、5 度以下の傾斜で地形は北に面していなければならない。しかし、2 度以下の斜面では地形の方位は問題ではない。

したがって、傾斜が 5 度以下 AND 地形は北に面している OR 傾斜は 2 度以下 である場所を見つける必要があります。こ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:

- ラスタ計算機 をもう一度開きます。
- ラスタバンド リストと 演算子 ボタン、そしてキーボードを使って ラスタ演算式 テキストエリアにこの式を作成します:

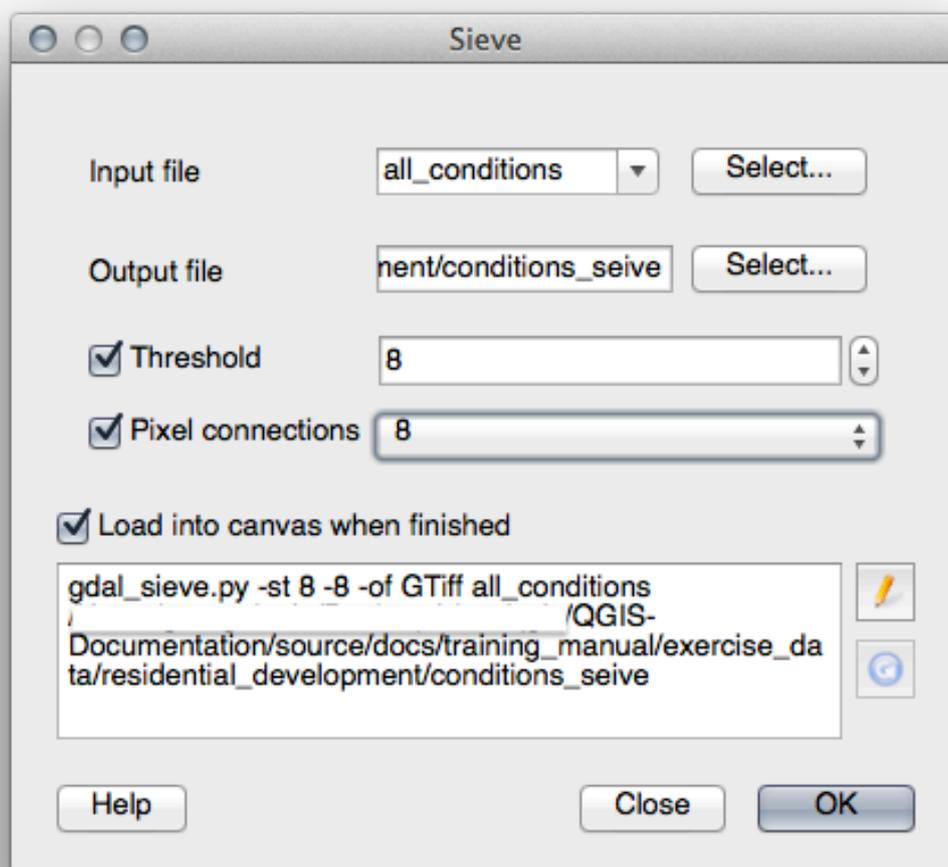
$$(\text{aspect_north}@1 = 1 \text{ AND } \text{slope_lte5}@1 = 1) \text{ OR } \text{slope_lte2}@1 = 1$$
- exercise_data/residential_development/ に all_conditions.tif として出力を保存します。
- ラスタ計算機 の OK をクリックします。あなたの結果は:



8.3.8 Follow Along: ラスタの簡素化

上の画像からわかるように、複合解析は条件を満たしている多くの非常に小さなエリアを残しました。しかし、これらは何かを建築するには小さすぎるので私たちの解析にとって本当に有用ではありません。小さく使用できないエリアをすべて取り除きましょう。

- ふるい ツールを開きます (ラスタ → 解析 → ふるい)。
- 入力ファイル を all_conditions に、出力ファイル を all_conditions_sieve.tif に設定します (exercise_data/residential_development/ ディレクトリ)。
- しきい値 と ピクセルの連結 の両方を 8 に設定して、ツールを実行します。

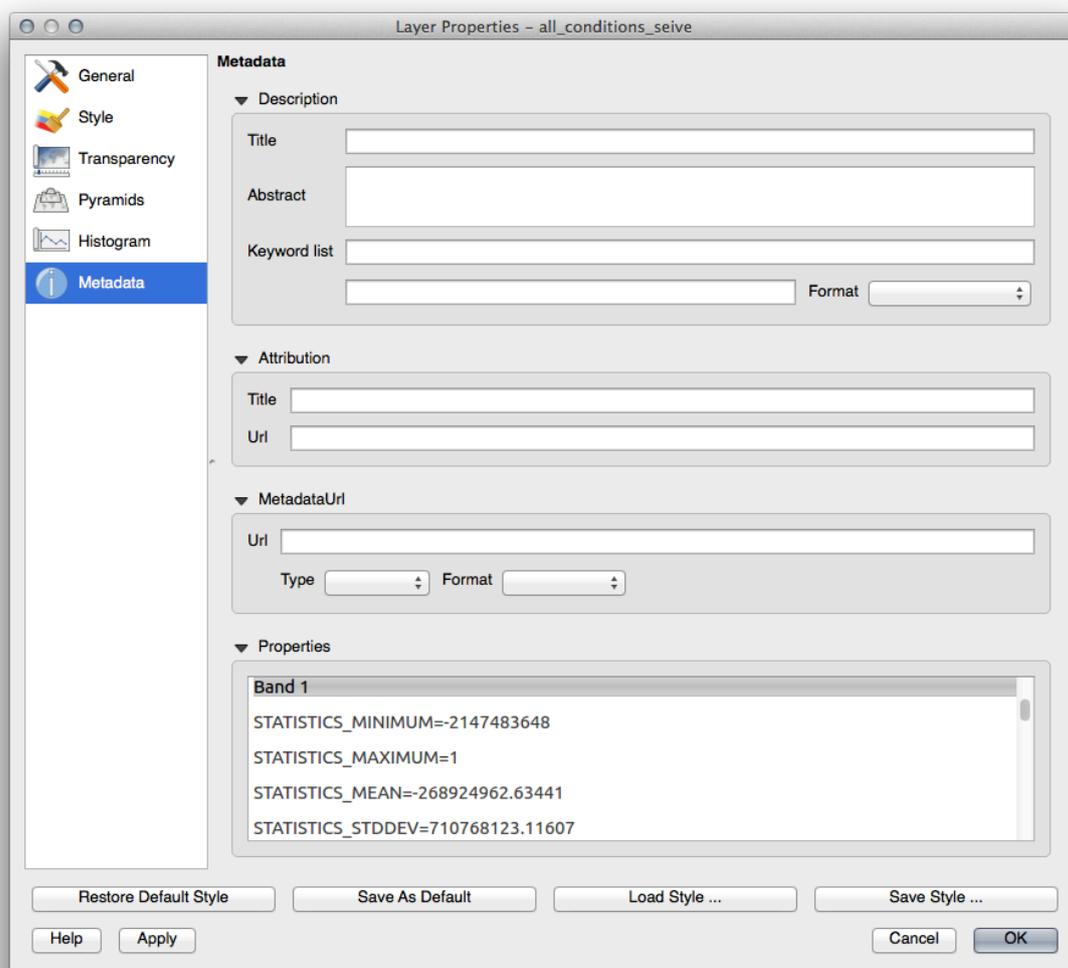


処理が完了すると、新しいレイヤはキャンバスに読み込まれます。しかし、データを表示するためにヒストグラムストレッチツールを使用しようとする次のようになります:



どうなっているのでしょうか? 答えは新しいラスタファイルのメタデータにあります。

- レイヤプロパティ ダイアログのメタデータ タブでメタデータを表示します。プロパティ セクションを見て下さい。



このラスタは派生元のものと同様に 1 と 0 の値しか持たないのに STATISTICS_MINIMUM の値はとても絶対値の大きな負の数になっています。データを調べたところ、この数は NULL 値として作用することがわかりました。私たちは除外されなかったエリアだけを求めているので、これらの NULL 値を 0 に設定しましょう。

- ラスタ計算機 を再度開いて次の計算式を作成します:

```
(all_conditions_sieve@1 <= 0) = 0
```

これは既存のすべてのゼロの値を維持しつつ、負の数をゼロにし、すべての 1 の値をもつエリアをそのままにします。

- exercise_data/residential_development/ に all_conditions_simple.tif として出力を保存します。

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。

8.3.9 In Conclusion

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、斜面方位の計算をしました。またこれらの結果をさらに分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。

8.3.10 What's Next?

あなたは2つの分析結果を持っています。潜在的に適した小地所を示すベクタ分析の結果と潜在的に適した地形を示すラスタ分析の結果です。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。

Chapter 9

Module: 分析の完了

分析は、ベクタとラスタの2パートを持ちます。このモジュールでは、これらを組み合わせかたを説明します。分析を終わらせ、最終結果を描きます。

9.1 Lesson: ラスタのベクタへの変換

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分析方法を使用するときに、ラスターとベクターデータの両方を利用できます。これは、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持っている柔軟性を向上させます。

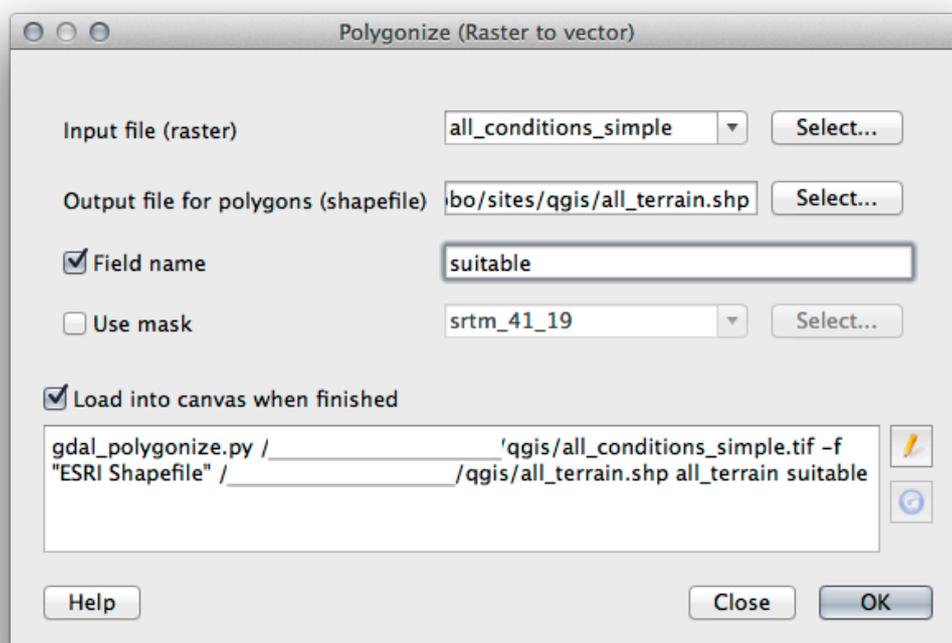
ラスタとベクタの分析を組み合わせるには、データのタイプを一方からもう一方に変換する必要があります。前のレッスンのラスタ結果をベクタに変換してみましょう。

****このレッスンの目標:****分析を完了するためにラスタ結果をベクタにすること。

9.1.1 Follow Along: ラスターからベクタ ツール

最新のモジュール `raster_analysis.qgs` から地図を起動します。前の練習中に `all_conditions_simple.tif` が計算されているはずで

- ラスター->変換->ポリゴン化(ラスターからベクトル)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:



- フィールド名を 適切 に (ラスタの値を記述する) 変更します。
- exercise_data/residential_development 下に all_terrain.shp として保存します。

今、ラスタのすべての値を含むベクトルファイルを持っていますが、興味ある領域は適当なものだけです。すなわち、それらのポリゴンの 適当 の値が 1 であるもの。このレイヤーのスタイルを変更すればそれを明確に可視化できます。

9.1.2 Try Yourself

ベクタ分析のモジュールに戻って参照ください。

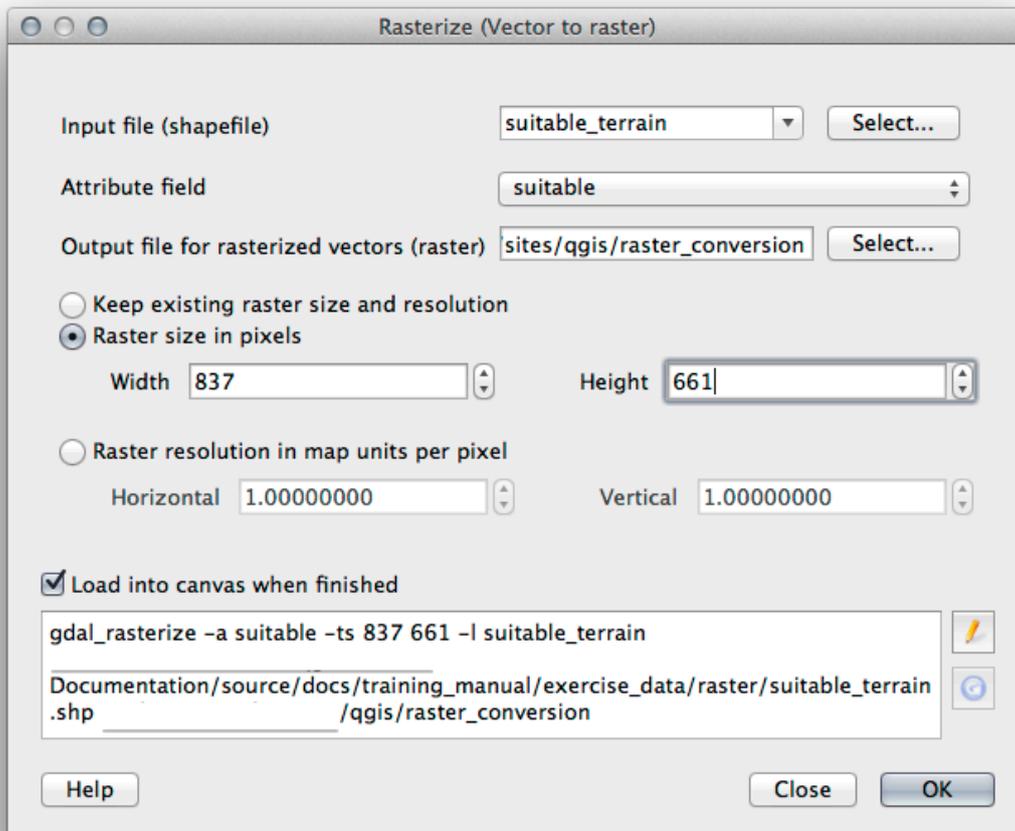
- suitable が値 1 であるポリゴンのみを含む新しいベクトルファイルを作成します。
- 新 し い ファイ ル を exercise_data/residential_development/ に suitable_terrain.shp として保存します。

結果の確認

9.1.3 Follow Along: ベクタをラスタに ツール

現在の問題には不必要ではありますが、上記で実行されるものとは反対の変換について知っておくと便利です。ちょうど前のステップで作成した suitable_terrain.shp ベクトルファイルをラスタに変換します。

- ラスター->変換->ラスタライズ (ベクトルからラスタ) をクリックしてこのツールを起動し、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。



- 入力ファイルは *all_terrain* ;
- 出力ファイル... は `exercise_data/residential_development/raster_conversion.tif` です。
- 幅 と 高さ はそれぞれ 837 と 661 です。

ノート: 出力画像のサイズは、ここではベクトル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 (レイヤーのプロパティ 中の *Metadata* 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。

9.1.4 In Conclusion

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。

9.1.5 What's Next?

今はベクトル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。

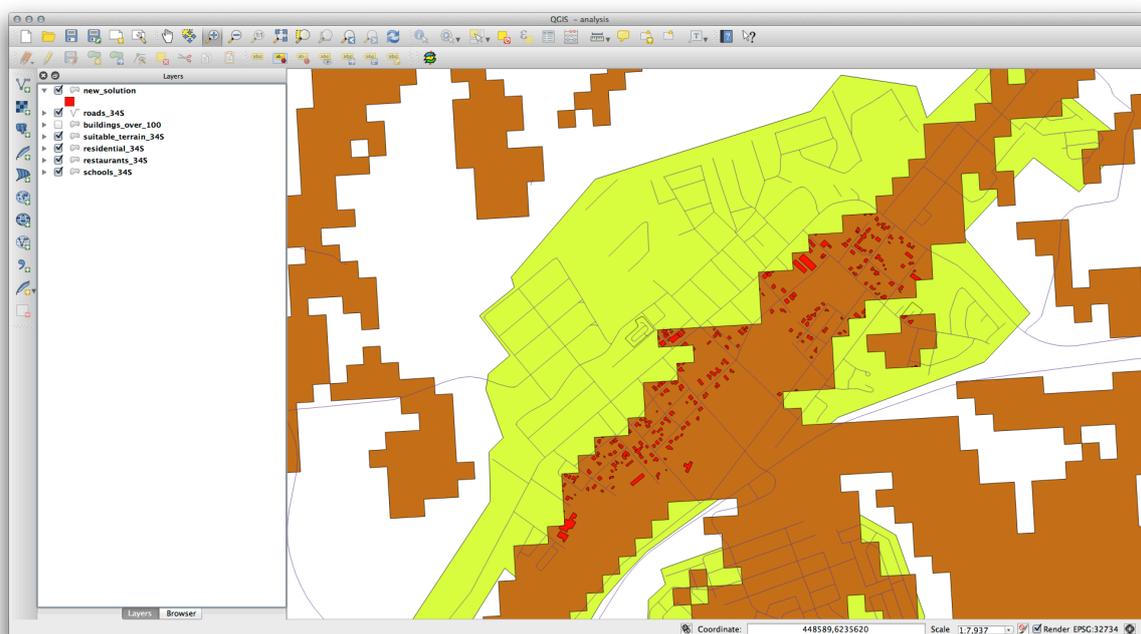
9.2 Lesson: 分析を組み合わせる

ベクトル化されたラスタ分析の結果を使用して、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トル化された地形の結果を使用します。

9.2.1 Try Yourself

- 現在の地図を保存します (`raster_analysis.qgs`)。
- 以前ベクトル分析をした地図を開きます (:kbd:`analysis.qgs`としてファイル保存したもの)。
- レイヤリストでこれらのレイヤを有効にします:
 - `hillshade`,
 - `solution` (または `buildings_over_100`)
- 前に作業した時から既に地図に読み込まれているはずであるこれらのレイヤに加え、`suitable_terrain.shp` データセットも追加します。
- もしレイヤに見つからないものがある場合は、`exercise_data/residential_development/` から見つけて下さい。
- `Intersect` ツール (ベクトル->ジオプロセッシングツール) を使用して、`suitable_terrain` レイヤと交差する建物のみが含まれる `new_solution.shp` と呼ばれる新しいベクトルレイヤーを作成します。

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:



ノート: 交差 ツールからどんな結果も得られない場合は各レイヤの CRS 設定をチェックして下さい。比較しているレイヤの両方が同じ CRS でなければなりません。要求される CRS を用いて新しいシェープファイルとしてレイヤを保存することによって片方のレイヤを投影変換する必要があるかもしれません。私たちの例では `suitable_terrain` レイヤは WGS 84 / UTM 34S に投影変換され、`suitable_terrain_34S` と名付けられました。

9.2.2 Try Yourself 結果の検査

new_solution レイヤの各建物を見て下さい。 *new_solution* レイヤのシンボロジをアウトラインだけに変更して、それらを *suitable_terrain* レイヤと比較して下さい。建物のいくつかを見て何に気づきましたか? それらは *suitable_terrain* レイヤと交差しているからといってすべて適当ですか? その理由は? どれが不適当だと考えますか?

結果をチェックする

9.2.3 Try Yourself 分析結果の改良

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分析結果を改良しましょう。

私たちの分析では *suitable_terrain* レイヤの中に完全に収まる建物だけが選ばれるようにしたい。これはどのように達成できるでしょうか? 1 つ以上のベクタ解析ツールを使用します。そして私たちの建物の大きさは 100 平方メートルを超えていることを覚えておいて下さい。

結果をチェックする

9.2.4 In Conclusion

あなたは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 (無理なく、分析に支えられて) 提供できます。

9.2.5 What's Next?

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。

9.3 課題

マップコンポーザを用いて分析結果を表す新しいマップを作して下さい。これらのレイヤを含めて下さい:

- *places* (ラベル付き),
- *hillshade*,
- *solution* (または *new_solution*),
- *roads* と
- *aerial_photos* または *DEM* のいずれか。

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みなさい。

9.4 Lesson: 補足の実習

このレッスンでは、QGIS での完全な GIS 解析を通して案内します。

ノート: Lesson developed by Linfiniti and S Motala (Cape Peninsula University of Technology)

9.4.1 問題の状態

あなたはケープ半島とその周辺で、珍しいフィンボスの植物種に適した生息地である領域を見つける任務を課されています。ケープ半島内での調査地域の範囲は次のとおりです。Melkbosstrand の南、ストランドの西。問題の種は次の嗜好を示すことを植物学者が示してくれています

- 東向き斜面に生息します。
- 15 % と 60 % の間の勾配の斜面に生息します。
- 年間総降水量 1200 ミリメートル超の領域に生息します。
- どの集落からも少なくとも 250 メートル離れたところで発見されます。
- それが発生した植生の面積は、面積が少なくとも 6000 平方メートルである必要があります。

ケープ自然保護ボランティアとして、あなたは自宅に最も近い適当な土地の一角でその植物を探すことに合意しました。GIS のスキルを使用してどこを見に行くべきか決定しましょう。

9.4.2 ソリューションの概要

自宅に最も近い候補領域を見つけるために、この問題を解決するために、利用可能なデータを使用する必要があります (`exercise_data/more_analysis` で利用可能)。ケープタウン (この問題の基になる) に住んでいない場合は、ケープタウンの領域の任意の家を選択できます。解決策に含まれるのは:

- DEM を分析し、東向き斜面および該当する勾配を有する斜面を見つける。
- 降雨ラスタを分析し、該当する降雨量の領域を見つける。
- ゾーニングベクトルレイヤーを分析し、集落から離れていて該当するサイズである領域を見つける。

9.4.3 マップの設定

- 画面の一番右下にある「CRS ステータス」ボタンをクリックします。表示される画面の *CRS* タブの下に *world* 座標参照系 ボックスが表示されるでしょう。
- このボックスで、システム座標投影->ユニバーサル横メルカトル (*UTM*) に移動します。
- Select the entry *WGS 84 / UTM zone 33S (with the EPSG code 32733)*.
- *OK* をクリックします。マップは *UTM33S* の CRS です。
- プロジェクトに名前を付けて保存 ツールバーボタンをクリックして地図を保存するか、ファイル → プロジェクトに名前を付けて保存... メニュー項目を使用します。
- お手元のコンピュータ上のどこかに *Rasterprac* というディレクトリを作成し、地図を保存してください。作成したレイヤーもこのディレクトリに保存します。

9.4.4 地図へデータを読み込む

データを処理するために、マップキャンバスに必要なレイヤー (通りの名前、ゾーン、降雨、DEM) をロードする必要があります。

ベクタにとって ...

- ベクトルレイヤーを追加 ボタンをクリックするか、レイヤー->ベクタレイヤーを追加... メニュー項目を使用します。
- 表示されるダイアログで、ファイルのラジオボタンが選択されているかを確認します。
- ブラウズ ボタンをクリックします。
- 表示されるダイアログで、`exercise_data/more_analysis/streets` ディレクトリを開きます。

- *Street_Names_UTM33S.shp* を選択します。
- 開く をクリックします。

ダイアログが閉じて、ファイルのパスが *Browse* ボタン隣のテキストフィールドに指定された元のダイアログが表示されています。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できます。そうしなければ手動でもこのフィールドにファイルのパスを入力できます。

- 開く をクリックします。ベクトルレイヤーが地図にロードされます。その色は自動的に割り当てられます。それは後に変更されます。
- レイヤーを *Streets* にリネームします。
- レイヤー *list* (デフォルトでは、画面の左側に沿った窓) でそれを右クリックします。
- 現れるダイアログで *名前変更* をクリックして名前を変更し、終わったら *Enter* キーを押します。
- ベクトル追加処理を繰り返しますが、今回は *Zoning* ディレクトリで *Generalised_Zoning_Dissolve_UTM33S.shp* ファイルを選択します。
- 名前を *Zoning* に変更します。

ラストでは …

- ラスターレイヤーを追加 ボタンをクリックするか、レイヤー->ラスターレイヤーを追加... メニュー項目を使用してください。
- 該当するファイルに移動し、それを選択し、開く をクリックしてください。
- 2 つのラスターファイル各々にこれを行います。したいファイルは、次のとおりです。 *DEM/再投影/DEM* と *降雨/再投影/rainfall.tif*。
- 降雨ラスタの名前を *Rainfall* (先頭大文字) に変更します。最初にそれらをロードする際は、画像がグレーの長方形になります。これは後に変更されますので心配無用です。
- マップを保存します。

適切に何が起きているのかを確認するためには、レイヤーのためのシンボルを変更する必要があります。

9.4.5 ベクタレイヤーのシンボロジの変更

- レイヤーリスト 内で 街路 レイヤーを右クリックします。
- 表示されているメニューで *プロパティ* を選択します。
- 表示されているダイアログで *スタイル* タブに切り替えます。
- 街路 レイヤーの現在の色を示す四角が付いた *変更* とラベルのついたボタンをクリックしてください。
- 表示されるダイアログで新しい色を選択します。
- *OK* をクリックします。
- レイヤープロパティ ダイアログの *OK* をもう一度クリックします。これは 街路 レイヤーの色を変更します。
- *Zoning* レイヤーについて同様のプロセスに従って、そのための適切な色を選択してください。

9.4.6 ラスタレイヤーのシンボロジの変更

ラスターレイヤーシンボロジはどこか違います。

- *Rainfall* ラスタの *プロパティ* ダイアログを開きます。
- *Style* タブに切り替えます。このスタイルダイアログがベクトルレイヤーに使用されるバージョンとは非常に異なっていることに気づくでしょう。

- 標準偏差を使用する が選択されていることを確認します。
- 関連付けられたボックスで値を 2.00 に変更します (デフォルトでは 0.00 に設定されているはず)です。
- コントラスト強調 見出しの下に、*Current* ドロップダウンリストの値を *MinMax* にストレッチ に変更します。
- *OK* をクリックします。「降水量」ラスタは、見える場合は、画素毎に異なる輝度値が見えるように、色を変更しなければなりません。
- DEM のためにこのプロセスを繰り返し、ストレッチするために使用される標準偏差を 4.00 に設定します。

9.4.7 レイヤ順の変更

- レイヤーの一覧 中で、レイヤーをクリックおよび上下にドラッグして地図上に表示される順序を変更します。
- QGIS の新しいバージョンでは、レイヤリスト 下に レンダリング順序を制御 チェックボックスがあるかもしれません。それがチェックされていることを確認してください。

これですべてのデータがロードされ正常に表示されますので、分析を開始できます。クリッピング操作が最初に行われるならば、最高です。使用するつもりのない領域の値を計算することに処理能力を無駄使いたないようにするためです。

9.4.8 正しい地区の検索

- ベクタレイヤ `admin_boundaries/Western_Cape_UTM33S.shp` を地図にロードします。
- `Districts` にリネームします。
- レイヤリスト 内で `Districts` レイヤを右クリックします。
- 表示されるメニューで、クエリ... メニュー項目を選択します。クエリ *Builder* ダイアログが表示されます。

今、地区の次のリストだけを選択するためのクエリを作成します：

- Bellville,
- Cape,
- Goodwood,
- Kuils River,
- Mitchells Plain,
- Simons Town,そして
- Wynberg.
- フィールド リスト中で `NAME_2` フィールドをダブルクリックします。以下の *SQL* の *where* 句 のテキストフィールドに表示されます。
- = ボタンをクリックし、= 記号が *SQL* クエリに追加されます。
- (現在空) 値 リストの下 *ALL* ボタンをクリックします。短い遅延の後、これは選択したフィールドの値 (`NAME_2`) を持つ 値 リストを移入します。
- 値 リストの中で `Bellville` をダブルクリックします。前述のように、これは *SQL* クエリに追加されます。

複数の地区を選択するためには、*OR* ブール演算子を使用する必要があります。

- *OR* ボタンをクリックすると、それが *SQL* クエリに追加されます。

- 上記と同様のプロセスを使用して、既存の SQL クエリに次の行を追加します。

```
"NAME_2" = 'Cape'
```

- 別の:kbd:OR 演算子を追加し、その後、同様の方法で、上記の地区のリストを通して作業を続けます。
- 最終的なクエリはこうであるべきです

```
"NAME_2" = 'Bellville' OR "NAME_2" = 'Cape' OR "NAME_2" = 'Goodwood' OR  
"NAME_2" = 'Kuils River' OR "NAME_2" = 'Mitchells Plain' OR "NAME_2" =  
'Simons Town' OR "NAME_2" = 'Wynberg'
```

- OK をクリックします。地図に表示される地区は現在、上記のリストのものに限定されています。

9.4.9 ラスタのクリップ

今、興味のある領域を持っている、このエリアへのラスタをクリップできます。

- 見えるレイヤーは *DEM*、*Rainfall*、*Districts* レイヤーだけであることを確認してください。
- *Districts* は見えるように最上になければなりません。
- ラスター->抽出->*Clipper* メニュー項目を選択することにより、クリッピングダイアログを開きます。
- 入力ファイル (ラスタ) レイヤー ドロップダウンリストで *DEM* レイヤーを選択します。
- 選択... ボタンをクリックして、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。
- *Rasterprac* ディレクトリに移動します。
- ファイル名を入力します。
- ファイルを保存します。データ値なし チェックボックスはチェックなしのままにしておきます。
- 正しいラジオボタンが選択されていることを確保することによって 範囲 クリッピングモードを使用してください。
- 地区を含む領域が選択されるように、キャンパス内のエリアをクリックしてドラッグ。
- 終了時にキャンパスにロード ボックスをチェックします。
- OK をクリックします。
- クリッピング操作が完了した後、*Clipper* ダイアログを閉じないでください。(そうすると、すでに定義されているクリッピング領域を失う原因となります。)
- 入力ファイル (ラスタ) ドロップダウンリストで *Rainfall* ラスタを選択し、別の出力ファイル名を選択します。
- 他のオプションを変更しないでください。以前に描いた既存のクリッピング領域を変更しないでください。すべてのものを同じままに残し、OK をクリックしてください。
- 第二のクリッピング操作が完了したら、*Clipper* ダイアログを閉じることができます。
- マップを保存します。

9.4.10 マップをクリーンアップします。

- レイヤーリスト から元の 降雨 と *DEM* レイヤーを削除します :
- これらのレイヤ上で右クリックし、削除 を選択します。
 - 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。
- 街路 レイヤのラベルを無効にします。
 - ラベリング ボタンをクリックします。
 - このレイヤーにラベル付け ボックスのチェックをオフにします。

- OK をクリックします。
- 再度 *Streets* をすべて表示します。
 - レイヤリスト でレイヤを右クリックします。
 - クエリ を選択します。
- 表示された クエリ ダイアログで、 *Clear* ボタンをクリックし、 *OK* をクリックします。
- データがロードされるまで待ちます。すべての街は今、表示されます。
- (*ラスタレイヤのシンボルを変更する*参照) 以前のようにラスタシンボルを変更します。
- マップを保存します。
- 今 レイヤーリスト 中のベクトルレイヤーを、横にあるボックスをオフにして非表示できます。これにより、地図のレンダリングが速くなりますし、時間の節約になります。

陰影起伏を作成するには、この目的のために書かれたプラグインを使用する必要があります。

9.4.11 ラスタ地形解析 プラグインの有効化

このプラグインは QGIS 1.8 にデフォルトで含まれています。しかし、それはすぐに表示されない場合があります。それがシステム上でアクセス可能かどうかを確認するには：

- プラグイン->プラグインを管理... メニュー項目をクリックしてください。
- ラスタ地形分析プラグイン の隣のボックスが選択されていることを確認します。
- *OK* をクリックします。

今 ラスタ->地形分析 メニュー項目を経由して、このプラグインにアクセスできるようになります。

プラグインは、時には特定の Python モジュールがシステムにインストールされているに依存してもよいことに注意してください。プラグインが欠落している依存関係を訴えながら働くことを拒否すべきである、援助のためのあなたの家庭教師や講師を依頼してください。

9.4.12 陰影図の作成

- レイヤーリスト 中で *DEM* がアクティブレイヤーであることを確認してください(すなわち、それはクリックされたことにより強調表示されている)。
- ラスタ->地形分析->*Hillshade* メニュー項目をクリックして *Hillshade* ダイアログを開きます。
- 出力レイヤーのための適切な場所を指定し、それを *hillshade* と呼びます。
- 結果をプロジェクトに追加する をチェックします。
- *OK* をクリックします。
- 処理が完了するのを待ちます。

新しい *hillshade* レイヤーが レイヤー list に現れています。

- レイヤー list 中で *hillshade* レイヤーを右クリックし、*Properties* ダイアログを呼び出します。
- 透明度 タブをクリックして透明度スライダーを 80 % に設定します。
- ダイアログで *OK* をクリックします。
- 透明な陰影起伏がクリップされた DEM 上に重ね合わされたときに効果に注意されたいです。

9.4.13 傾斜

- ラスター → 地形分析 メニュー項目をクリックします。
- クリップされた DEM を入力レイヤーとして 傾斜 分析タイプを選択します。
- 出力の目的のために適切なファイル名と場所を指定します。
- 結果をプロジェクトに追加する をチェックします。
- OK をクリックします。

勾配画像が計算され、地図に追加されました。しかし、いつものように単なる灰色の四角形です。何が起きているかを適切に確認するには、次のようにシンボルを変更します。

- (いつものようにレイヤーの右クリックメニューを経由して) レイヤー *Properties* ダイアログを開きます。
- スタイル タブをクリックします。
- (カラーマップ ドロップダウンメニューで) グレースケール となっているところを 疑似カラー に変更します。
- 標準偏差を使用 ラジオボタンが選択されていることを確認します。

9.4.14 傾斜方位

- 傾斜を計算するのと同じアプローチを使用しますが、最初のダイアログボックスで 傾斜方向 を選択します。

定期的にマップを保存することを忘れないで下さい。

9.4.15 ラスタの再階級

- ラスター → ラスター計算機 メニュー項目をクリックしてください。
- 出力レイヤーのための場所として *Rasterprac* ディレクトリを指定します。
- プロジェクトに結果を追加 ボックスが選択されていることを確認します。

左の ラスター *bands* リストに、レイヤー *list* 中のすべてのラスタレイヤが表示されます。傾斜レイヤーが傾斜 という名であれば、それは 傾斜@1、と表示されます。

傾きが 15 と 60 度の間にあることが必要です。したがって 15 未満 60 超はすべて排除しなければなりません。

- インタフェースのリスト項目とボタンを使用して以下の式を作成します：

```
((slope@1 < 15) OR (slope@1 > 60)) = 0
```

- 適切な場所とファイル名で 出力 *layer* フィールドを設定してください。
- OK をクリックします。

同じアプローチを使用して正しい向き (東向き 45 と 135 度の間) を見つけます。

- 次の式を組み立てます：

```
((aspect@1 < 45) OR (aspect@1 > 135)) = 0
```

- 同じように、正しい雨量 (1200mm より大きい) を見つけてください。次の式を構築します。

```
(rainfall@1 < 1200) = 0
```

すべてのラスタを再分類した今、それらは (それらは正しく地図に追加されていると仮定して) 自分の地図に灰色の長方形として表示が表示されます。(1 と 0 は true または false を意味する) 2 つのクラスだけでラスタデータを正しく表示するには、それらのシンボルを変更する必要があります。

9.4.16 再階級されたレイヤのスタイルの設定

- いつものように、レイヤーのプロパティ ダイアログ中でスタイル タブを開きます。
- *band* から最小/最大値ロード 見出しの下の 実際 (遅い) ラジオボタンを選択します。
- 読み込み ボタンをクリックします:

今 :guilabel: カスタム最小/最大値 フィールドにそれぞれ `:kbd:'0` と `1` を取り込む必要があります。(そうでない場合、そのデータの再分類と間違いだった、と再びその部分の上に移動する必要があります。)

- コントラスト強調 見出しの下で *Current* ドロップダウンリストを `:guilabel:'MinMax` ヘストレッチするに設定します。
- *OK* をクリックします。
- 3 つのすべてのためにこれを行うには、ラスタを再分類し、作業を保存することを忘れないでください!

残る唯一の基準は、その領域が都市地域から 250m 離れてなければならないことです。計算領域が農村地域の端から 250m 以上であることを確実にすることによって、この要件を満たします。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。

9.4.17 農村地域の検索

- レイヤリスト ですべてのレイヤを非表示にします。
- *Zoning* を再表示します。
- その上で右クリックして *Query* ダイアログを起動します。
- 次のクエリをビルドします:

```
"Gen_Zoning" = 'Rural'
```

行き詰まったら *Streets* クエリを構築するための以前の説明書を参照してください。

- それを実行したら クエリ ダイアログを閉じます。

Zoning レイヤーからポリゴンのコレクションが表示されるはずですが、新しいレイヤファイルにこれらを保存する必要があります。

- *Zoning* の右クリックメニューで 名前を付けて保存... を選択します。
- *Zoning* ディレクトリの下にレイヤーを保存します。
- 出力ファイル名を `rural.shp` とします。
- *OK* をクリックします。
- レイヤをマップに追加します。
- ベクター → ジオプロセッシングツール → ディゾルブ をクリックします。
- 選択地物のみ使用 ボックスのチェックはないままで、入力ベクトルレイヤーとして *rural* レイヤーを選択します。
- ディゾルブ *field* の下で、—すべてをディゾルブ— を選択します。
- *Zoning* ディレクトリの下にレイヤーを保存します。
- *OK* をクリックします。TOC (「目次」 レイヤー *list* に言及) に新しいレイヤーを追加するかどうかを確認するダイアログが表示されます。
- はい をクリックします。
- ディゾルブ ダイアログを閉じます。
- *rural* と *Zoning* レイヤーを外します。
- マップを保存します。

今、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。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これを行います。

9.4.18 負のバッファを作成します

- メニュー項目 ベクトル → バッファ → ジオプロセッシングツール をクリックしてください。
- 表示されるダイアログで、入力ベクトルレイヤーとして *rural_dissolve* レイヤーを選択します (選択された地物のみ使用 はチェックすべきではありません)。
- バッファ距離 ボタンを選択し、関連するフィールドに値 -250 を入力してください。負の値は、バッファが内部バッファでなければならないことを意味しています。
- バッファ結果をディゾルブする ボックスにチェックします。
- 他の地方のベクトルファイルと同じディレクトリに出力ファイルを設定します。
- 出力ファイル名を *rural_buffer.shp* とします。
- 保存 をクリックします。
- OK をクリックして処理が完了するのを待ちます。
- 現れたダイアログで はい を選択します。
- バッファ ダイアログを閉じます。
- *rural_dissolve* レイヤーを削除します。
- マップを保存します。

3 つの既存のラスタと同じ分析に農村地帯を組み込むためには、それは同様にラスタライズする必要があります。しかし、ラスタは、分析のために互換性があるようにするために、同じサイズにする必要があります。そのため、ラスタライズする前に次の 3 つのラスタと同じエリアにベクトルをクリップする必要があります。ベクトルは別のベクトルによってクリップできるのみなので、最初にラスタと同じサイズのバウンディングボックスポリゴンを作成する必要があります。

9.4.19 バウンディングボックスベクタの作成

- メニュー項目 レイヤー ->新規->新規シェープファイルのレイヤー... をクリックしてください。
- *type* 見出しの下で *Polygon* ボタンを選択します。
- CRS を指定 をクリックし、基準座標系を :kbd:`WGS 84/UTM ゾーン 33S : EPSG : 32733` に設定します。
- OK をクリックします。
- 新しいベクトルレイヤー ダイアログでも OK をクリックします。
- *Zoning* ディレクトリにベクタを保存します。
- 出力ファイル名を *bbox.shp* とします。
- 新しい *bbox* レイヤーと再分類ラスタのレイヤーを除くすべてのレイヤーを非表示にします。
- レイヤーリストで *bbox* レイヤーが強調表示されていることを確認してください。
- 表示>ツールバー メニュー項目に移動し、*Digitizing* が選択されていることを確認します。その後、その上に鉛筆やキーと、ツールバーのアイコンが表示されるはずですが、編集を切り替え ボタンです。
- 編集を切り替え ボタンをクリックして 編集モード に入ってください。これにより、ベクトルレイヤーが編集できるようになります。
- 地物を追加 ボタンをクリックします、編集を切り替え ボタンの近くにあるはずですが、これは、二重の矢印ボタンの後ろに隠れてすることができます。 *Digitizing* ツールバーの非表示のボタンもしそうなら、表示するために二重矢印をクリックしてください。

- 地物を追加 ツールをアクティブにして、ラスタの角を左クリックしてください。それが正確であることを保証するために、マウスのホイールでズームインする必要があるかもしれません。このモードでは、地図をパンするには、マウスの中央ボタンかマウスホイールをクリックしたまま地図内でドラッグします。
- 第四及び最終点に関しては、形状を確定するために右クリックします。
- シェープ ID の任意の番号を入力します。
- OK をクリックします。
- 編集を保存 ボタンをクリックします。
- 編集セッションを停止するには、:guilabel: 編集を切り替え ボタンをクリックしてください。
- マップを保存します。

これでバウンディングボックスができましたので、農村地域のバッファレイヤーをクリップするためにそれを使用できます。

9.4.20 ベクタレイヤのクリップ

- *bbox* と *rural_buffer* レイヤーだけが、後者を上にして、表示されていることを確認してください。
- ベクタ>ジオプロセッシングツール>クリップ メニュー項目をクリックしてください。
- 表示されたダイアログで、入力ベクトルレイヤーを *rural_buffer* に、クリップレイヤーを *bbox* に、両方とも 選択地物のみ使用 ボックスをチェックしないで、設定します。
- *Zoning* ディレクトリの下に、出力ファイルを置きます。
- 出力ファイル名を *rural_clipped* とします。
- OK をクリックします。
- TOC にレイヤーを追加するように求められたら、:guilabel: Yes をクリックしてください。
- ダイアログを閉じます。
- 3 つのベクトルを比較し、自分のために結果を参照してください。
- *bbox* と *rural_buffer* レイヤーを削除し、地図を保存します。

これでラスタライズする準備ができました。

9.4.21 ベクタレイヤのラスタ化

作成した新しいラスタのピクセルサイズを指定する必要がありますので、まず既存のラスタの 1 の大きさを知っておく必要があります。

- 3 つの既存のラスタのいずれかの *Properties* ダイアログを開きます。
- メタデータ タブに切り替えます。
- メタデータテーブルの *Dimensions* 見出しの下 X と Y 値を書き留めておきます。
- プロパティ ダイアログを閉じます。
- ラスター->変換->*Rasterize* メニュー項目 をクリックしてください。データセットがサポートされていないことについて警告を受けることがあります。それをクリックして消し、無視します。
- 入力レイヤとして *rural_clipped* を選択します。
- *Zoning* ディレクトリ内の出力ファイルの場所を設定します。
- 出力ファイル名を *rural_raster.tif* とします。
- 新しい *size* ボックスをチェックし、X と Y に以前にノートにとった値を入力します。
- キャンバスにロードします ボタンをチェックします。

- 実行されるコマンドを表示するテキストフィールドの隣にある鉛筆アイコンをクリックします。既存のテキストの最後に、スペースを追加し、それからテキスト `-burn 1` を追加します。これは、新しいラスタに既存のベクタを「焼き」、ベクタで覆われた領域に新たな 1 の値を与えるようラスタライズ機能に告げる (画像の残りの部分、自動的に 0 にされる、とは反対に)。
- OK をクリックします。
- それが計算された後、新たなラスタが地図に表示されるはずですが。
- 新しいラスタはグレーの長方形のようになります - 再分類ラスタのために行ったように、表示スタイルを変更できます。
- マップを保存します。

今、全 4 つの基準それぞれを別々のラスタに持っていますので、領域がすべての基準を満たすかを確認するためにそれらを組み合わせる必要があります。それをするために、ラスタは互いに乗算されます。これが発生するとき、すべての重なる画素が値 1 を持つ場合は値 1 を保持しますが、画素で 4 つのラスタのいずれかに 0 値がある場合はその結果においては 0 になります。このようにして、結果には重なる領域だけが含まれます。

9.4.22 ラスタの結合

- ラスタ → ラスタ計算機... をクリックします。
- (それらと呼ばれるものに応じて、そのレイヤーのための適切な名前を) 次の式を構築します。

```
[Rural raster] * [Reclassified aspect] * [Reclassified slope] *
[Reclassified rainfall]
```

- 出力場所を `Rasterprac` ディレクトリに設定します。
- 出力ラスタの名称を `cross_product.tif` とします。
- 結果をプロジェクトに追加する ボックスがチェックされていることを確認します。
- OK をクリックします。
- 他の再分類ラスタのスタイルを設定するのと同じ方法で、新しいラスタのシンボルを変更します。新しいラスタが正しく、すべての基準が満たされている領域を表示します。

最終的な結果を得るには、6000m² より大きい領域を選択する必要があります。しかし、これらの領域を正確に計算することはベクタレイヤでのみ可能なので、ラスターをベクタ化する必要があります。

9.4.23 ラスタのベクタライズ

- メニュー項目 ラスター->変換->*Polygonize* をクリックしてください。
- `cross_product` ラスタを選択します。
- 出力場所を `Rasterprac` に設定します。
- `candidate_areas.shp` と改名します。
- 終了時にキャンパスにロードします ボタンがチェックされていることを確認します。
- OK をクリックします。
- 処理が完了したらダイアログを閉じます。

ラスタのすべてのエリアはベクトル化されているので、1 の値を持っている領域だけを選択する必要があります。

- 新しいベクターのため クエリ ダイアログを開きます。
- クエリのビルド:

```
"DN" = 1
```

- OK をクリックします。
- クエリが完了した後 *candidate_areas* ベクトルを保存することで、結果から新しいベクトルファイルを作成します (この値を持つ領域のみ 1 が表示されています)。このためのレイヤーの右クリックメニュー中の 名前を付けて保存... 機能を使用してください。
- Rasterprac ディレクトリにファイルを保存します。
- 候補領域のみ.shp と名前を付けます。
- マップを保存します。

9.4.24 各ポリゴンの面積の計算

- 新規ベクタレイヤの右クリックメニューを表示します。
- 属性テーブルを開く を選択します。
- テーブルの下に沿った 編集モード切り替え ボタンをクリックするか、Ctrl キー+E を押します。
- テーブル底部に沿った フィールド calculator を開く ボタンをクリックするか、Ctrl キー+I を押します。
- 表示されるダイアログ中の 新しいフィールド 見出しの下に、フィールド名 面積 を入力します。出力フィールド型は整数、フィールド幅は 10 でなければなりません。
- フィールド演算式 に、タイプします:

```
$area
```

計算された値で、このフィールドの計算は、ベクトルレイヤーの各ポリゴンの面積を計算し、次いで (面積 と呼ばれる) 新しい整数列を移入することを意味します。

- OK をクリックします。
- *id* と呼ばれる別の新しいフィールドに同じことを行います。フィールド電卓 *expression* 中にタイプします:

```
$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。

- 再び 編集モード切り替え をクリックし、編集内容を保存するようプロンプトが表示されたら、そうします。

9.4.25 与えられたサイズの面積を選択

今このエリアは知られています:

- 6000 メートル m² よりも大きなポリゴンのみを選択するように (いつものように) クエリを作成します。クエリは次のとおりです。

```
"area" > 6000
```

- 選択を *solution.shp* と呼ばれる新しいベクタレイヤとして保存します。

今はもう解領域が出ているので、家に最も近いものを選択します。

9.4.26 あなたの家のデジタイズ

- 以前のように新しいベクトルレイヤーを作成しますが、今回は *type* 値が *Point* となるよう選択します。
- それが正しい CRS であることを確認します!

- 新規レイヤを `house.shp` とします。
- 新規レイヤ作成の完了
- (新規レイヤが選択された状態で) 編集モードにします。
- 街路をガイドとして使用しながら、家または居住の他の現在の場所がある点をクリックします。家を見つけるのを助けるために他のレイヤーを開く必要があります。近くのどこにも住んでいない場合は、単に家が考えられるかもしれない街の中のどこかをクリックします。
- シェープ ID の任意の番号を入力します。
- *OK* をクリックします。
- 編集内容を保存し、編集モードを終了します。
- マップを保存します。

どれが家に最も近いかを決定するため、ソリューションエリアのポリゴンの重心(「質量中心」)を見つける必要があります。

9.4.27 ポリゴンの中心点の算出

- ベクトル->ジオメトリツール->ポリゴン *centroids* メニュー項目をクリックします。
- 入力レイヤーを `solution.shp` と指定します。
- 出力場所を `Rasterprac` と与えます。
- 出力先ファイル `solution_centroids.shp` を呼び出します。
- *OK* をクリックし、TOC に結果を追加し(レイヤーリスト)、その後ダイアログを閉じます。
- 新しいレイヤーを、見えるようレイヤーの順序の一番上にドラッグします。

9.4.28 あなたの家に最も近い重心の計算

- ベクトル->分析ツール->距離 *matrix* メニュー項目をクリックしてください。
- 入力レイヤはあなたの家、ターゲットレイヤは `solution_centroids` である必要があります。これらの両方とも、ユニークな ID フィールドとして `id` フィールドを使用する必要があります。
- 出力マトリクス型は *linear* になるはずですが。
- 適切な出力場所と名前を設定します。
- *OK* をクリックします。
- テキストエディタでファイルを開きます(またはスプレッドシートにインポート)。最短距離に関連付けられたターゲット ID に注意してください。同距離に複数あることもあります。
- 家に最も近い唯一のソリューション領域を選択するために、QGIS でクエリを作成します(`id` フィールドを使用してそれを選択する)。

これは、研究の質問への最終的な答えです。

提出のために、お好みの魅力的なラスタ(例えば *DEM* とか 傾斜ラスタ、など)の上に半透明の陰影起伏のレイヤを、また、最も近いソリューション領域(複数可)のポリゴンだけでなく、家を含めます。出力図を作成する際は地図作成のためのすべてのベストプラクティスに従ってください。

Chapter 10

Module: 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方法をお見せします。

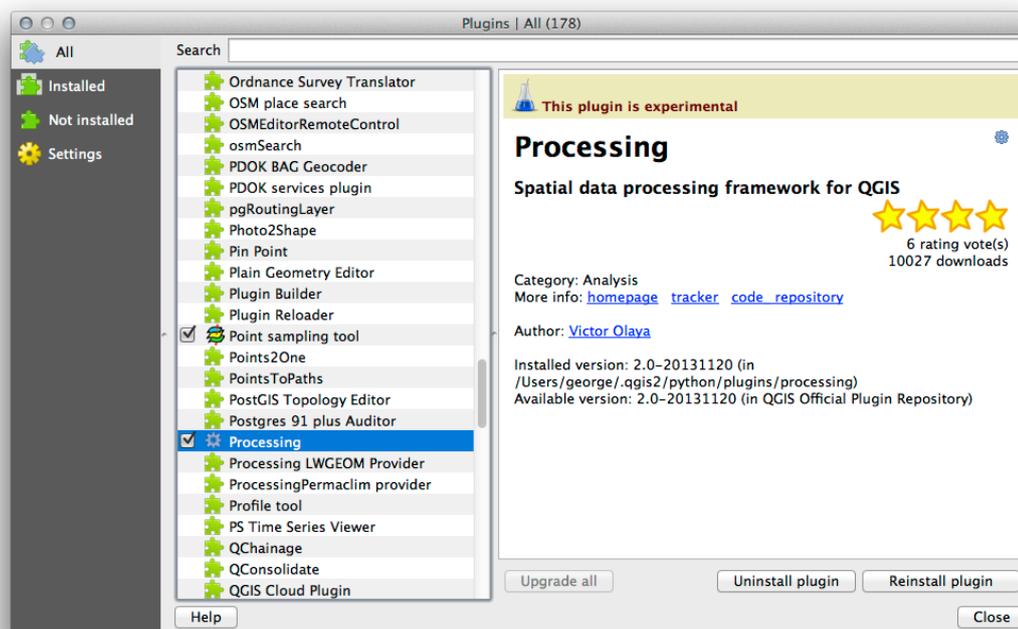
10.1 Lesson: プラグインのインストールと管理

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。

10.1.1 Follow Along: プラグインの管理

- 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。
- 開かれたダイアログで *Processing* プラグインを探しましょう。

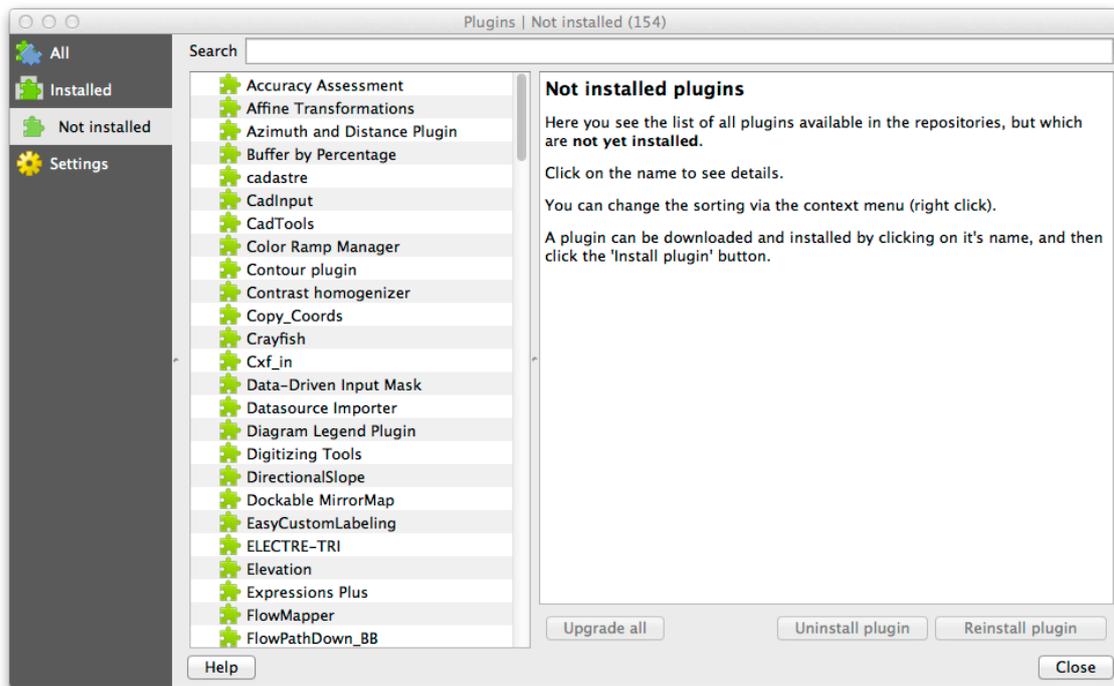


- このプラグインの隣のボックスをクリックしてチェックを外し、プラグインをアンインストールします。
- *Close* をクリックします。
- メニューを見ると *Processing* メニューがなくなったことに気付くでしょう。あなたが以前使っていたプロセッシングの機能の多くが消えてしまいました! それらは プロセッシング プラグインの一部だからです。再び使用するには有効化される必要があります。
- プラグインマネージャ をもう一度開き、 プロセッシング プラグインの隣のチェックボックスをクリックして再有効化し、 *閉じる* をクリックします。
- プロセッシング メニューが再び利用可能になったはずですが。

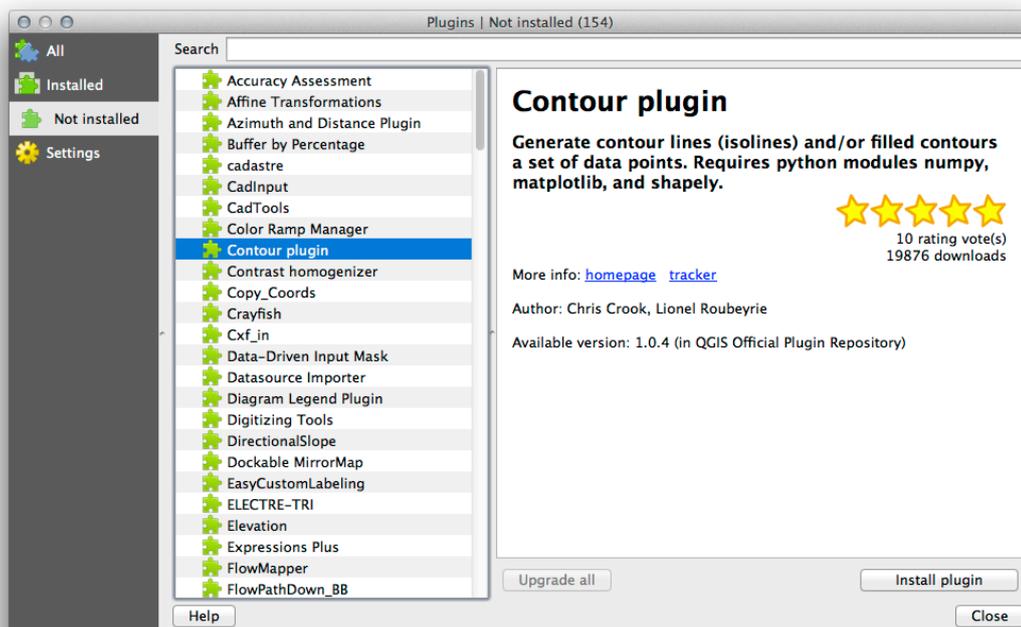
10.1.2 Follow Along: 新しいプラグインのインストール

有効化・無効化できるプラグインはインストールされているものです。

- 新しいプラグインをインストールするには プラグインマネージャ ダイアログの インストールされていない オプションを選択します。あなたがインストール可能なプラグインはここにリストされます。このリストはあなたのシステム設定によって異なります。



- 表示されている一覧の中から任意のプラグインを選択するとその情報を見ることができます。



- プラグイン情報パネルの下にある プラグインをインストール ボタンをクリックするとプラグインをインストールすることができます。

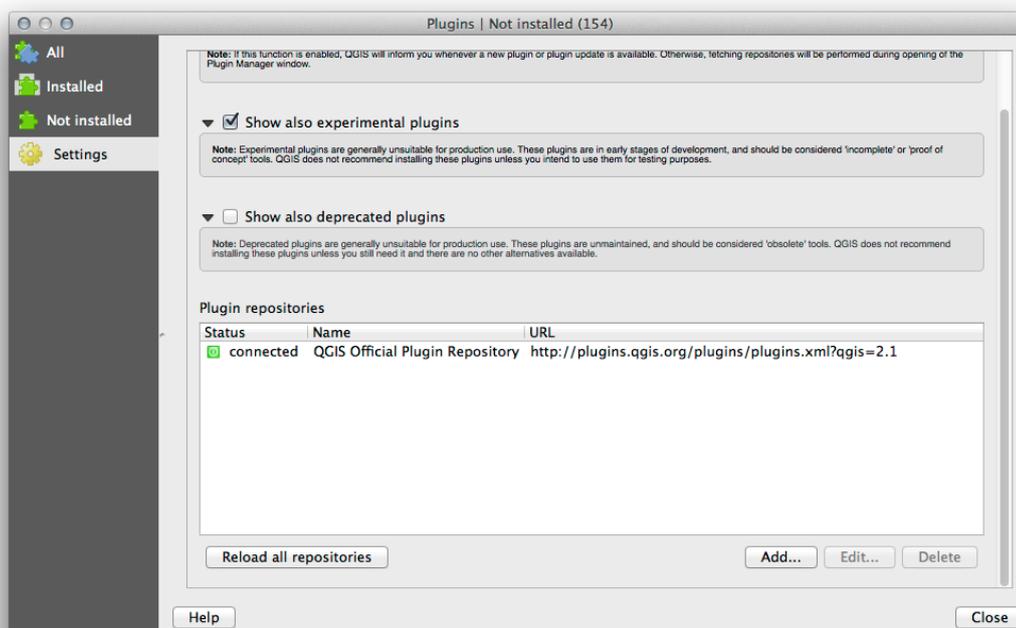
10.1.3 Follow Along: 追加的なプラグインリポジトリの設定

あなたが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で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だけが

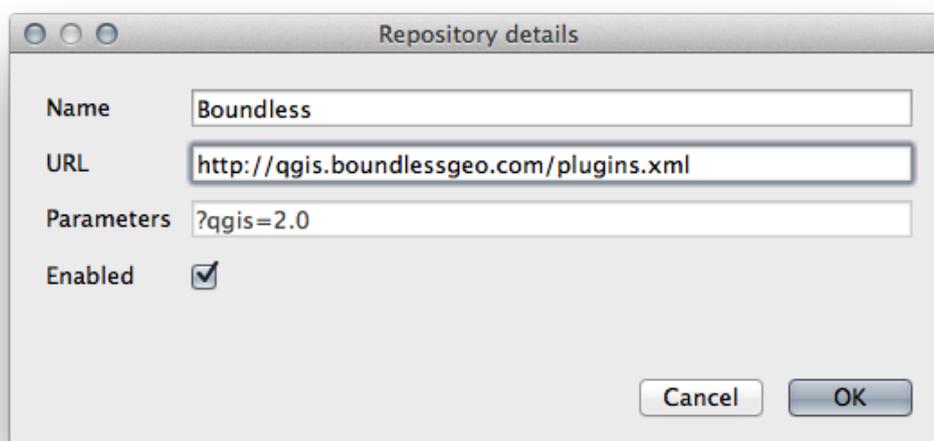
有効になっていて、公式プラグインにだけアクセスすることができます。これらは通常あなたが最初に利用しようと思うプラグインです。なぜなら、これらはよくテストされておりデフォルトで QGIS に含まれているものがあるためです。

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。まず追加的なリポジトリの設定をします。

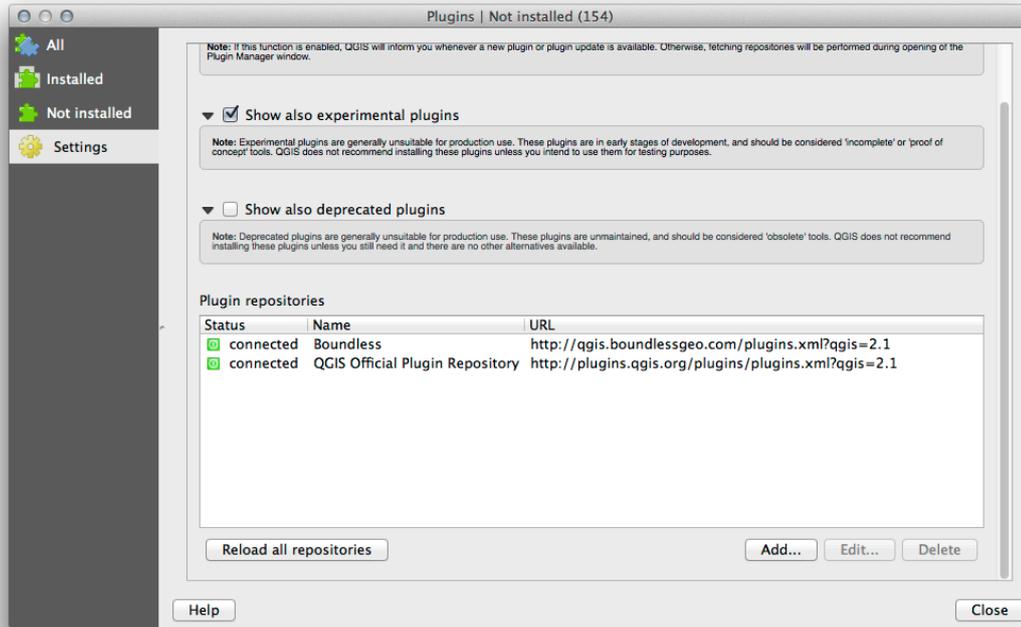
- プラグインマネージャ ダイアログの 設定 タブを開きます:



- 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。
- 設定したい新しいリポジトリの名前と URL を入力します。 利用可にする 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。



- 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。



- 開発の初期段階にあるプラグインも表示するには 実験的プラグインも表示する チェックボックスを チェックします。
- インストールされていない タブに戻ると追加的なプラグインが利用可能になったことがわかるでしょう。
- リスト中のプラグインをクリックしてから プラグインをインストール ボタンをクリックすればインストールできます。

10.1.4 In Conclusion

QGIS でプラグインをインストールするのは単純で効果的です!

10.1.5 What's Next?

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。

10.2 Lesson: 役に立つ 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。

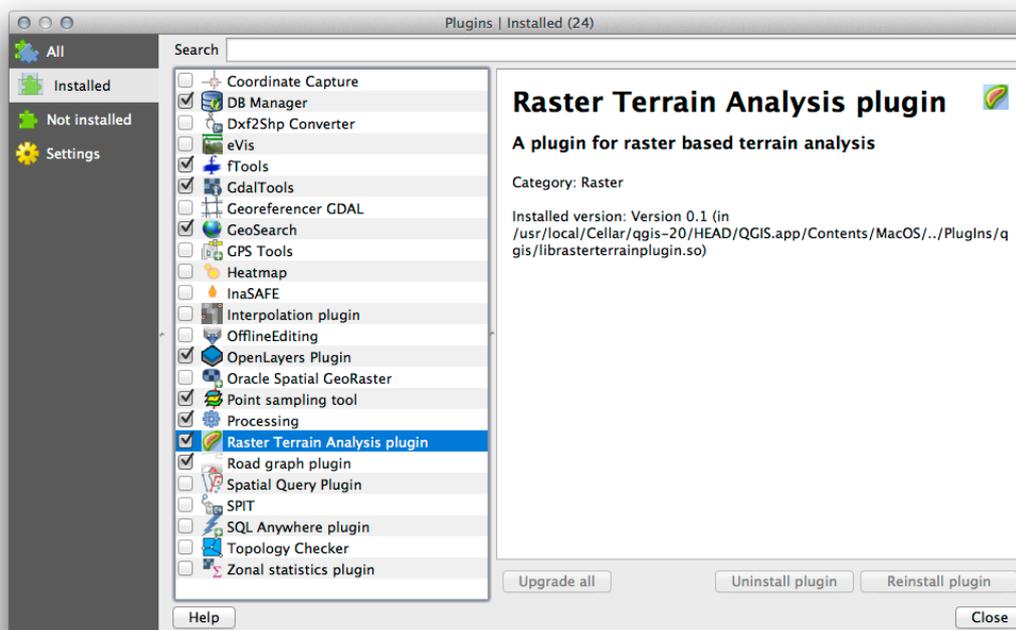
このレッスンの目標: プラグインインターフェイスに慣れ、いくつかの便利なプラグインを使ってみる。

10.2.1 Follow Along: ラスタ地形解析プラグイン

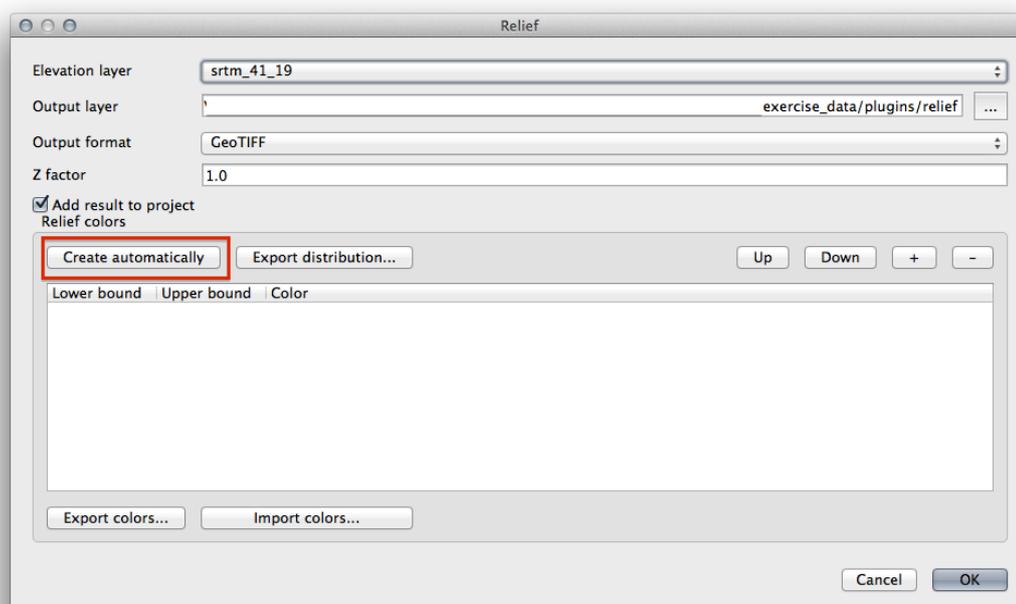
- 新しい地図に `exercise_data/raster/SRTM` の `srtm_41_19.tif` ラスタデータセットを読み込みます。

ラスタ解析に関するレッスンから、あなたは既にラスタの解析機能を知っています。あなたは GDAL ツール(ラスタ → 解析 からアクセス可能)を使用しました。しかし、ラスタ地形解析プラグインについても知っておく方がいいでしょう。これは QGIS の新しいバージョンでは標準機能なので個別にインストールする必要はありません。

- プラグインマネージャを開き、ラスタ地形解析プラグインが有効になっていることをチェックします。

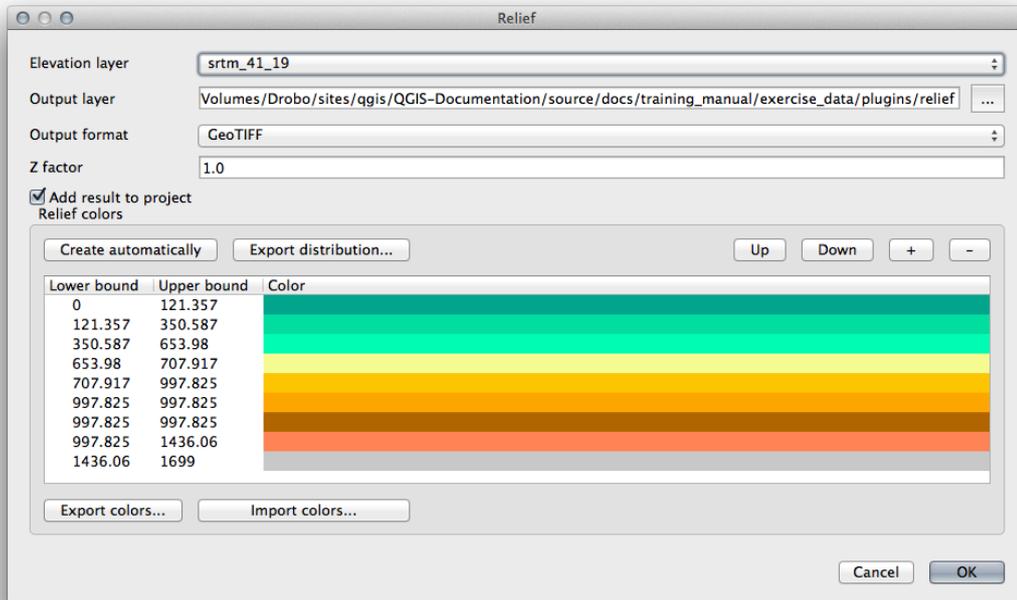


- ラスタメニューを開くと、地形解析サブメニューがあります。
- 地形解析 → レリーフ をクリックして、次のオプションを入力します:



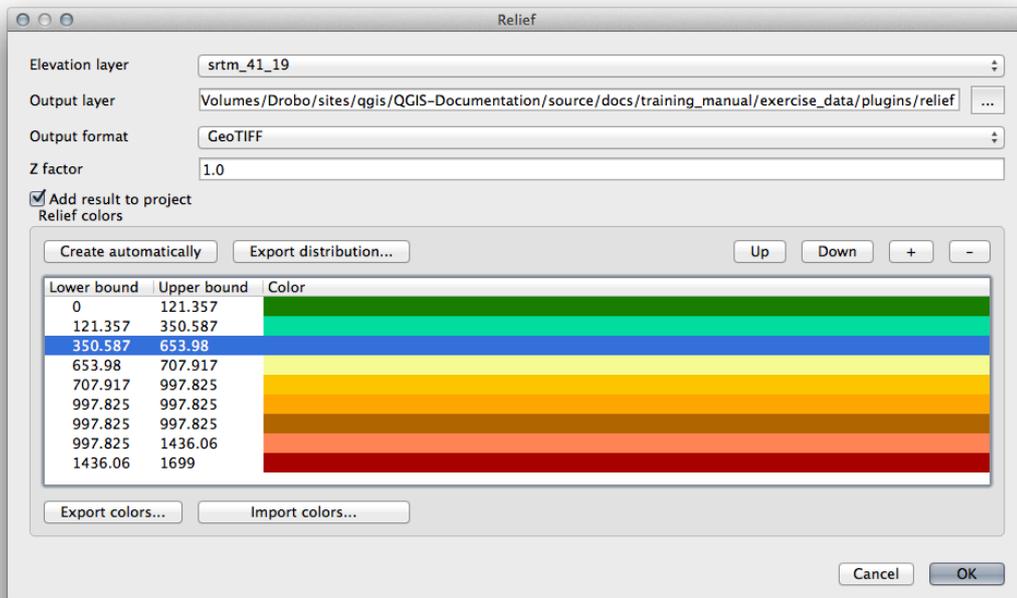
- 新しいファイルを exercise_data/plugins/relief.tif に保存します (必要なら新しいフォルダを作成します)。

- 出力形式 と Zファクタ は変更しません。
- 結果をプロジェクトに追加する ボックスはチェックされていることを確認します。
- 自動的に作成する ボタンをクリックします。下にリストが作成されます:

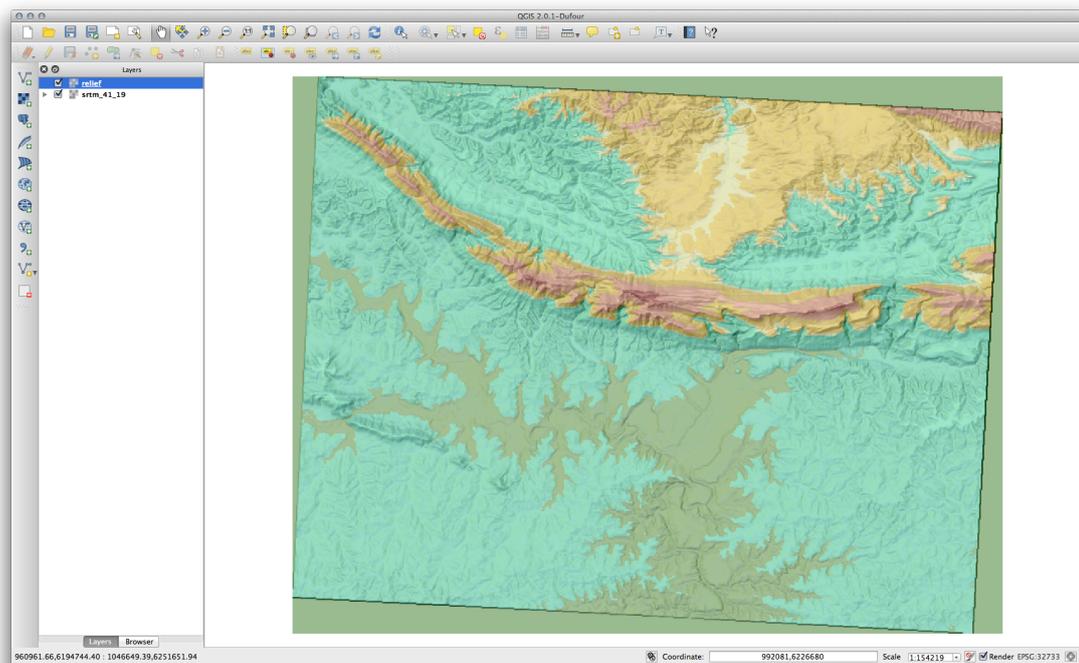


これらはプラグインがレリーフを作成するのに使う色です。

- お好みに応じてこれらの色は変更することができます。変更するには各行の色の横棒をダブルクリックします。例えば:



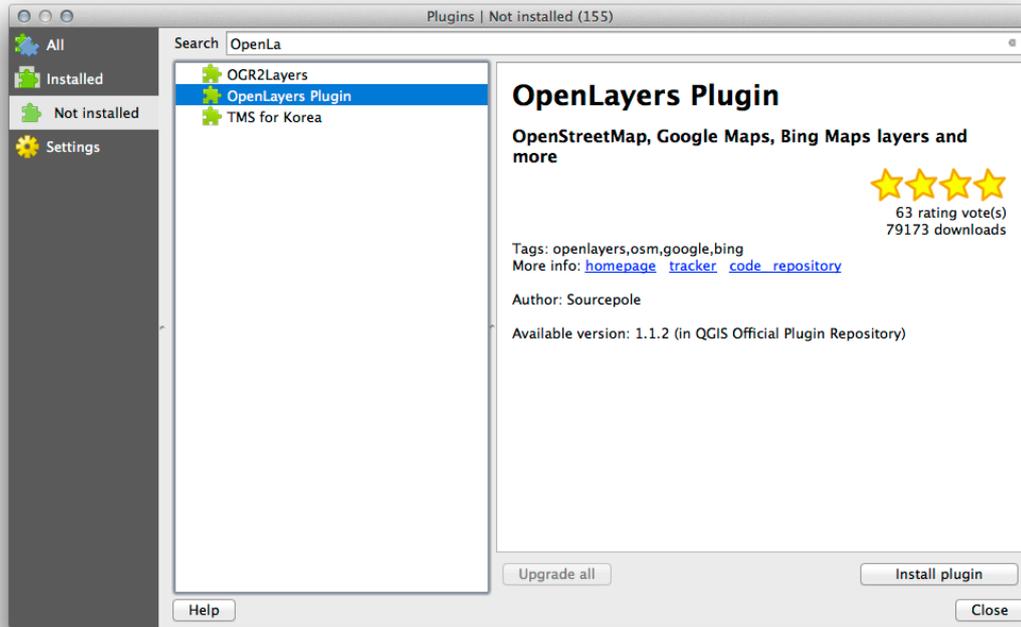
- OK を押すとレリーフが作成されます:



これは他のラスタレイヤの上のオーバーレイとして半透明の陰影図を使用したときと同様の効果を実現しています。このプラグインの利点は、レイヤのみでこの効果を作成することです。

10.2.2 Follow Along: OpenLayers プラグイン

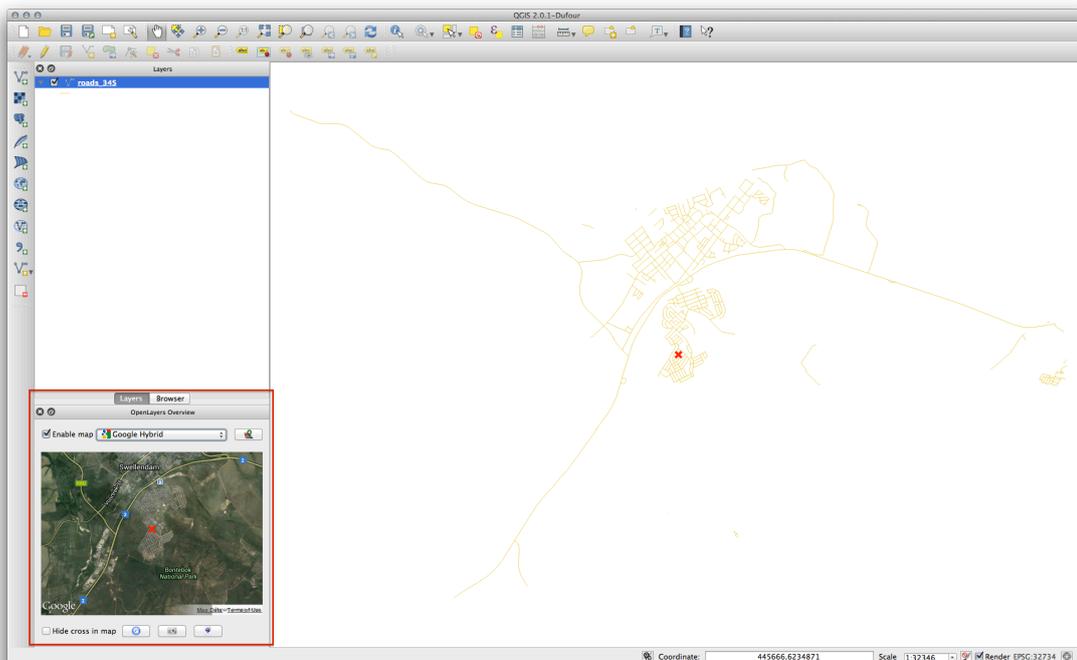
- 新しい地図を起動し、それに *roads.shp* レイヤーを追加します。
- Swellendam エリア上で拡大します。
- プラグイン *Manager* 使用し、*Filter* フィールド中に単語 *OpenLayers* を入力して新しいプラグインを見つけます。
- フィルタリングされたリストからプラグイン *OpenLayers* をを選択します。



- プラグインをインストール ボタンをクリックしてインストールします。
- 終わったら、プラグイン *Manager* を閉じます。

それを使用する前に、地図とプラグインの両方が正しく設定されていることを確認してください：

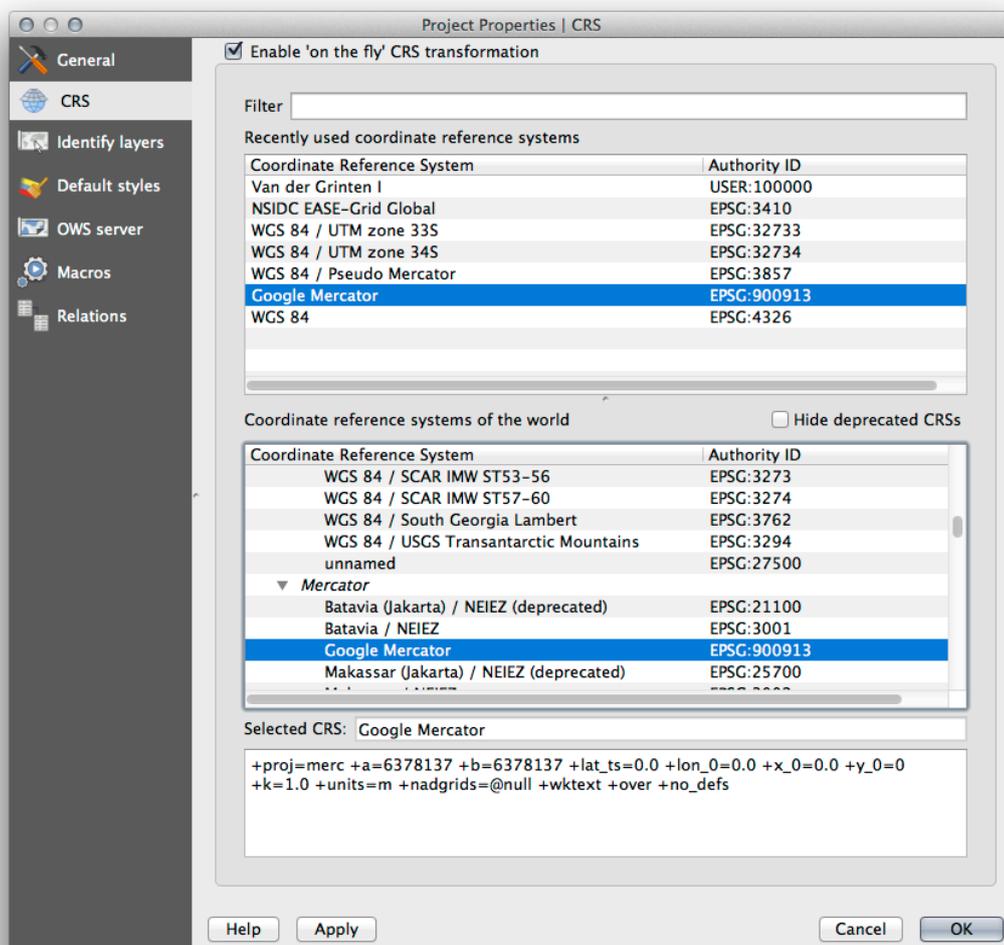
- ウェブ-> *OpenLayers* をプラグイン -> *OpenLayers* を *Overview* をクリックすることで、プラグインの設定を開きます。
- 希望の地図の種類を選択するパネルを使用します。この例では、「ハイブリッド」型地図を使用しますが、ご希望であれば他のものを選択できます。



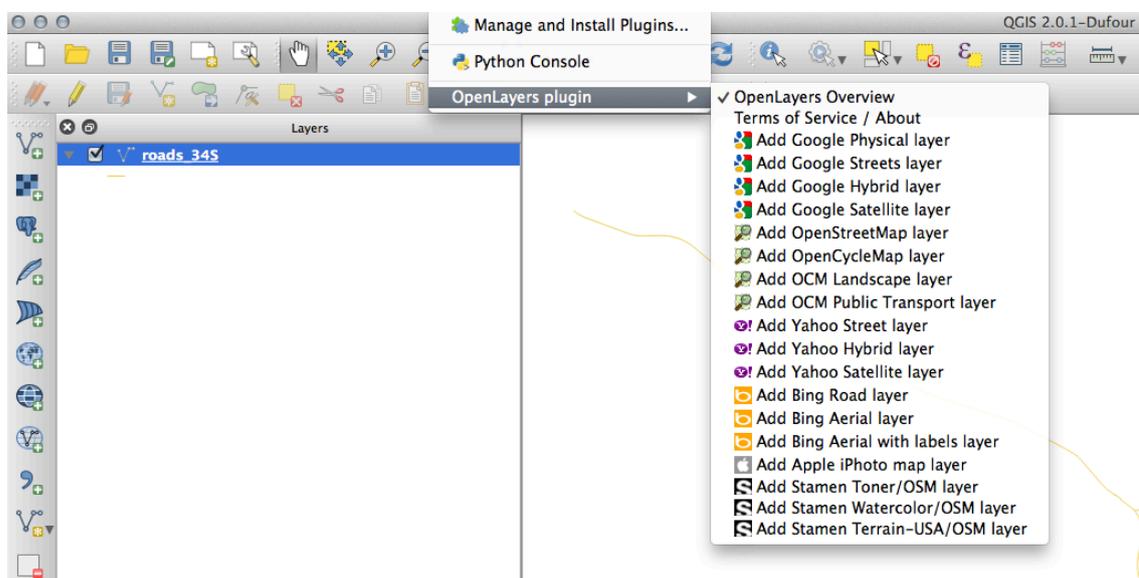
- メニューから プロジェクト->プロジェクト *Properties* を選択して プロジェクト *Properties* ダイアロ

グを開きます。

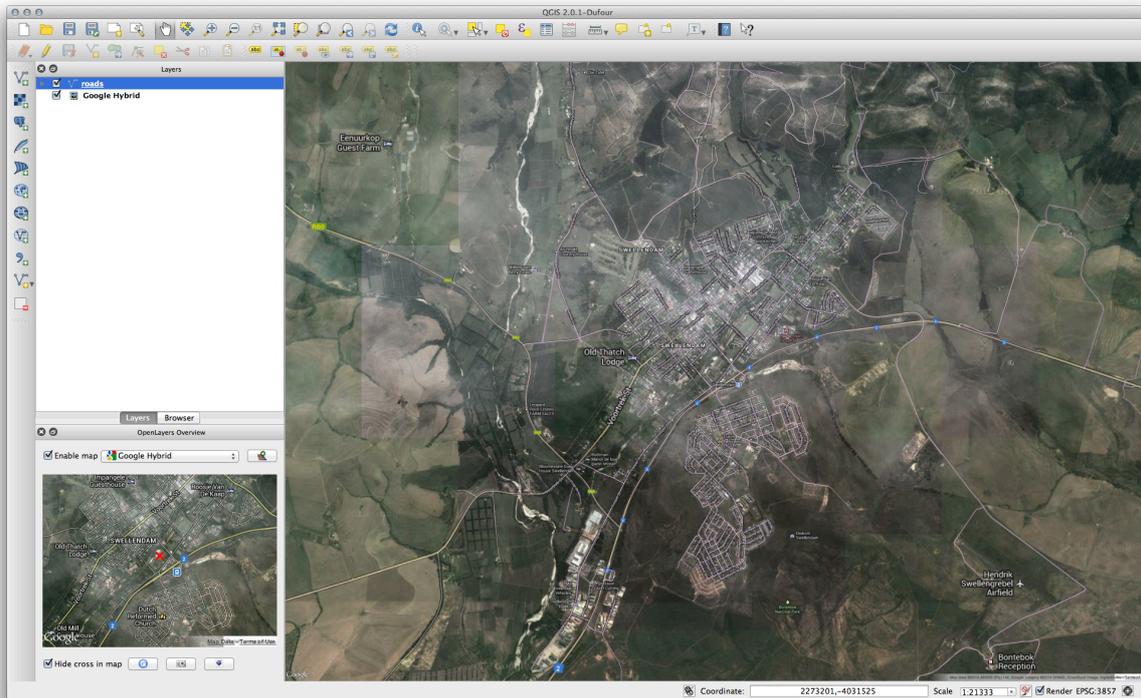
- 「オンザフライ」投影を有効にし、Google メルカトル図法を使用します。



- 今、あなたの地域のグーグルマップを与えるためにプラグインを使用します。プラグイン -> *OpenLayers* をプラグイン-> *Google ハイブリッド Layer* を追加 をクリックしてそれを追加できます。



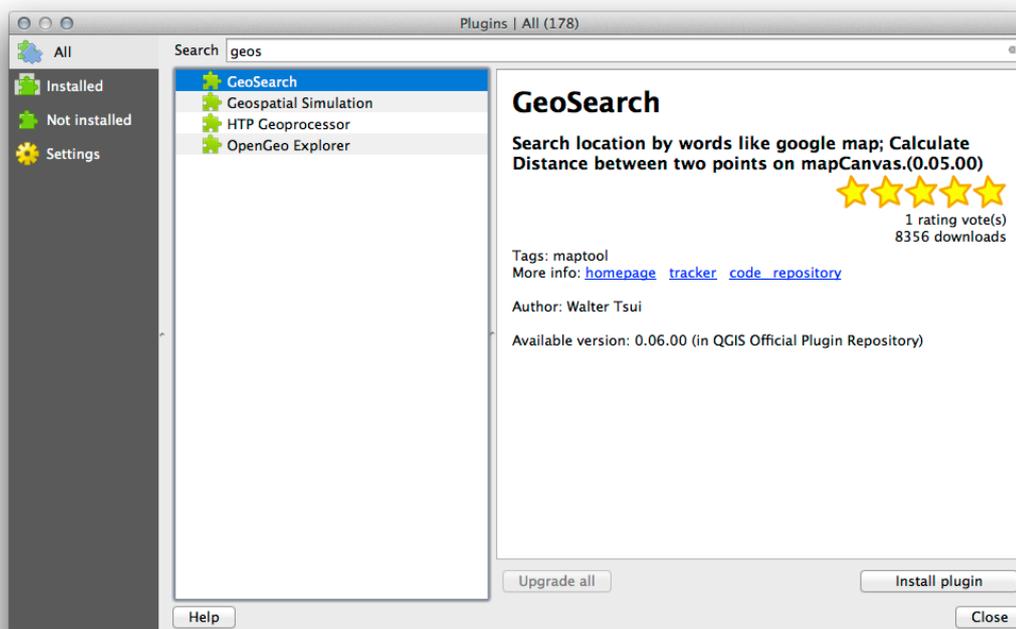
これは背景として使用でき、またはマップ上にある場所を見つける手助けするには Google から新しいラスタ画像をロードします。ここでは、オーバーレイとして当社独自のベクトル道路レイヤで、そのようなレイヤは、次のとおりです。



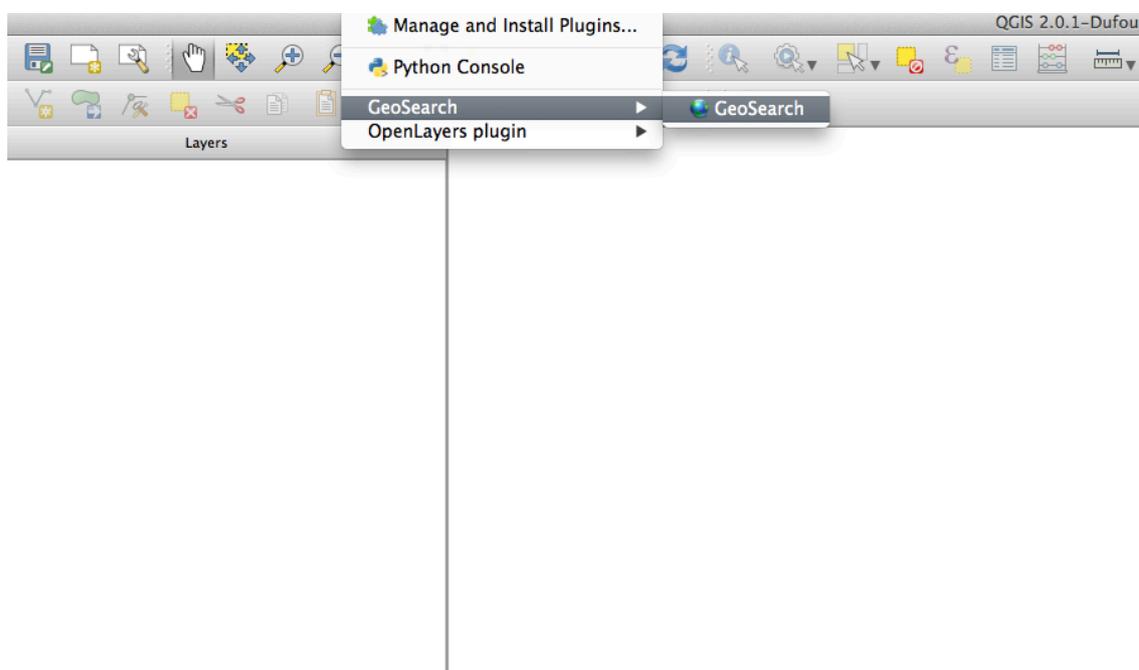
ノート: 背景レイヤーの上にそれが見えるように、Google のレイヤーの上に自分の道路レイヤーをドラッグする必要があります。また、再中心マップに道路レイヤの程度まで拡大する必要があります。

10.2.3 Follow Along: GeoSearch プラグイン

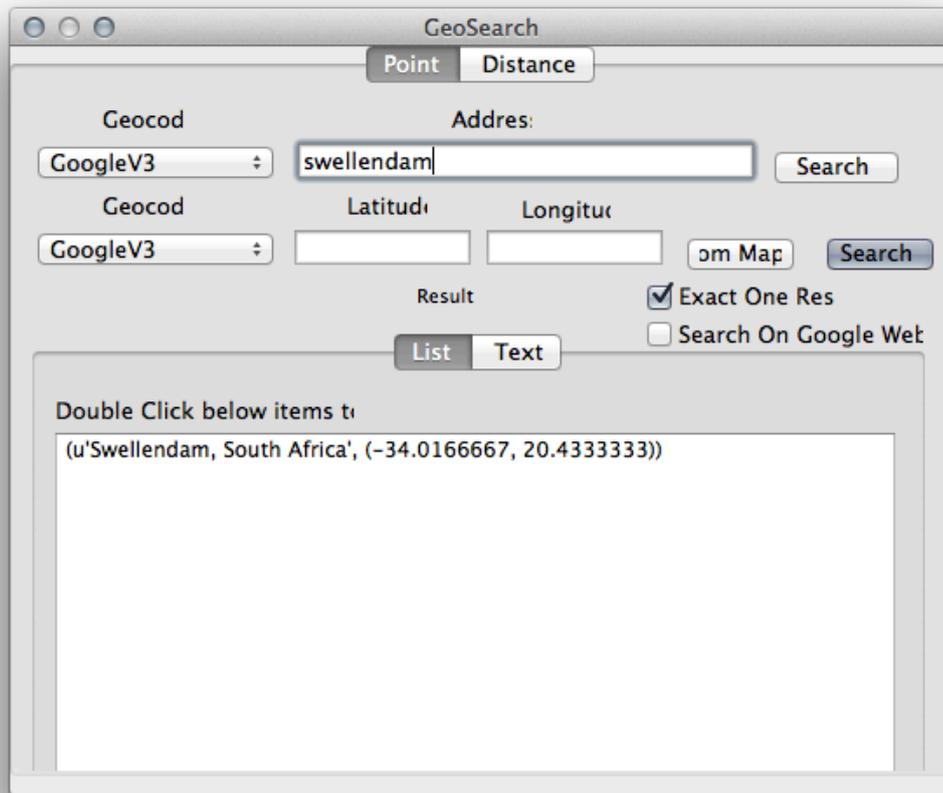
- なしのデータセットを使用して新しいマップを開始します。
- プラグイン *Manager* を開き GeoSearch プラグインについてフィルタし、*Plugin* をインストール をクリックしてインストールします。



- プラグイン *Manager* を閉じます。
- 今、地名を検索するために GeoSearch プラグインを使用できます。プラグイン -> *GeoSearch* プラグイン -> *GeoSearch* をクリックして GeoSearch ダイアログを開きます。



- majorUrbanName **検索!** GeoSearch ダイアログで、マップ上でそれを見つけます：



10.2.4 In Conclusion

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。

10.2.5 What's Next?

次はリモートサーバにホストされているレイヤをリアルタイムで使う方法を見ていきます。

Chapter 11

Module: オンラインリソース

マップ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。あなたがインターネットに接続されている限り、ロードすることが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)。

11.1 Lesson: Web Mapping Services

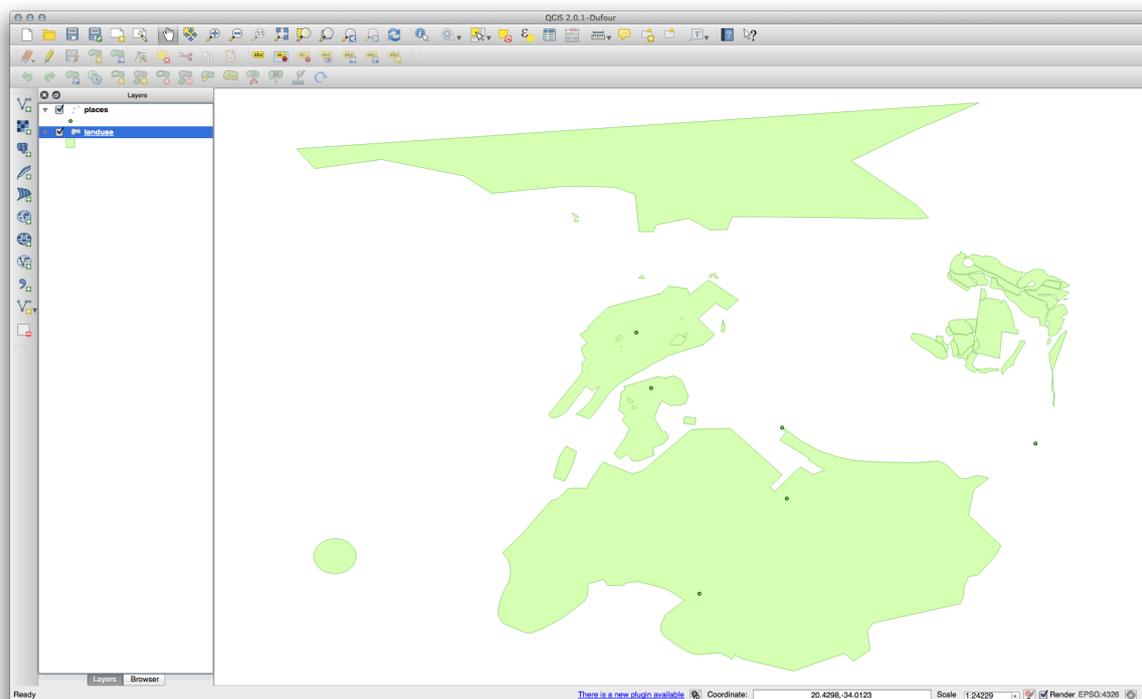
Web Mapping Service (WMS) は、リモートサーバ上でホストされたサービスです。ウェブサイトと同じ様に、サーバへ接続することでアクセスができます。QGIS を使用すると、既存のマップに直接 WMS をロードすることができます。

プラグインのレッスンから、それが例えば、Google からの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、これは一度オフのトランザクションです: 画像はダウンロードしたら変更されません。WMS は、地図上でパンやズームインするとき、自動的にそのビューを更新するライブサービスだということが異なっています。

****このレッスンの目標:**** WMS を使用して制限を知ること。

11.1.1 Follow Along: WMS レイヤーの読み込み

この演習では、コースの開始時に作られた基本地図を使用するか、または単に新しい地図を開始し、その中にいくつかの既存のレイヤーを読み込むことができます。この例では、新しいマップを使用して、元のロード *places* と *landuse* レイヤーやシンボルを調整します:



- 新しいマップ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マップを使用します。
- WMS レイヤーを追加するために開始する前に、最初に「オンザフライ」投影を無効化してください。これは、レイヤーがもはや適切に重複しないように発生することがありますが、心配しないでください：後で修正します。
- WMS レイヤーを追加するため、WMS レイヤーの追加 ボタンをクリックします。

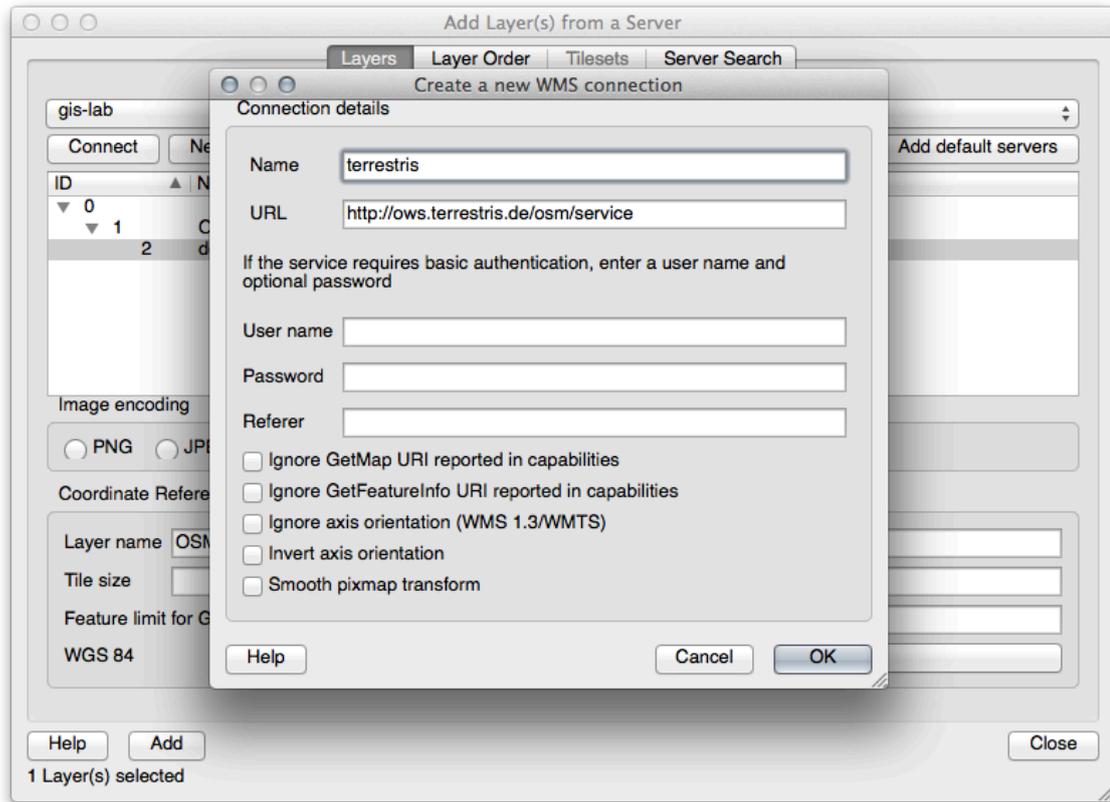


コースの冒頭で SpatiaLite データベースに接続した方法を思い出してください。landuse、places、および water レイヤーは、そのデータベース内にあります。これらのレイヤーを使用するには、まずデータベースに接続する必要があります。WMS を使用するレイヤーは、リモートサーバ上にあることを除いて、同様です。

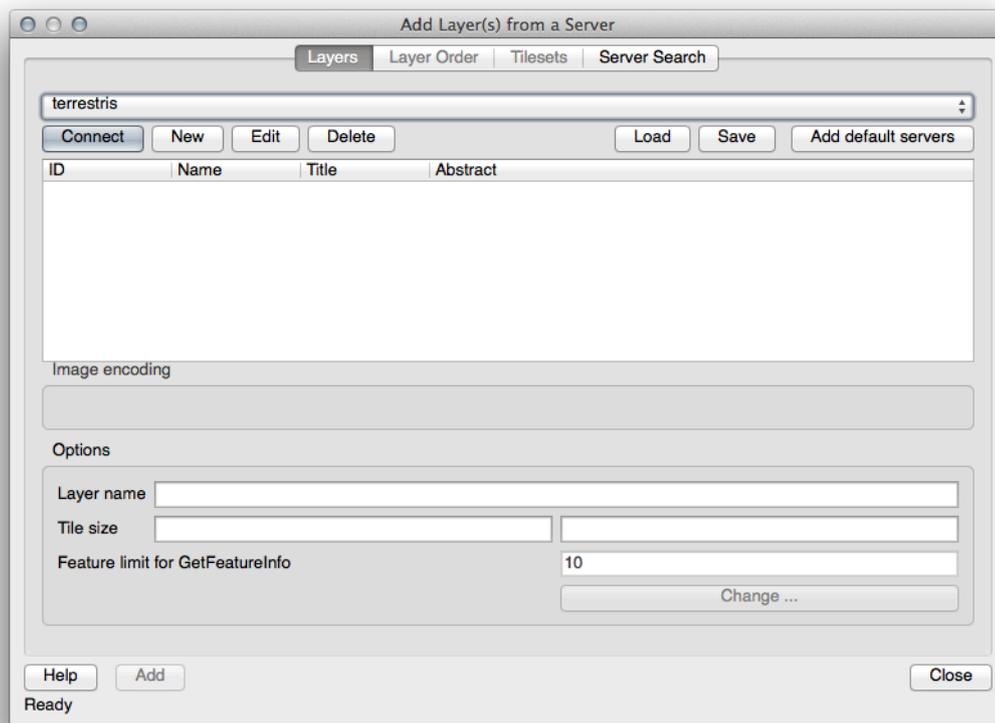
- ボタン WMS への新しい接続を作成するには、New をクリックしてください。

WMS アドレスを続ける必要があります。インターネット上で利用可能ないくつかの無料の WMS サーバがあります。これらの一つは テレストリス、です、OpenStreetMap データセットを使用しています。

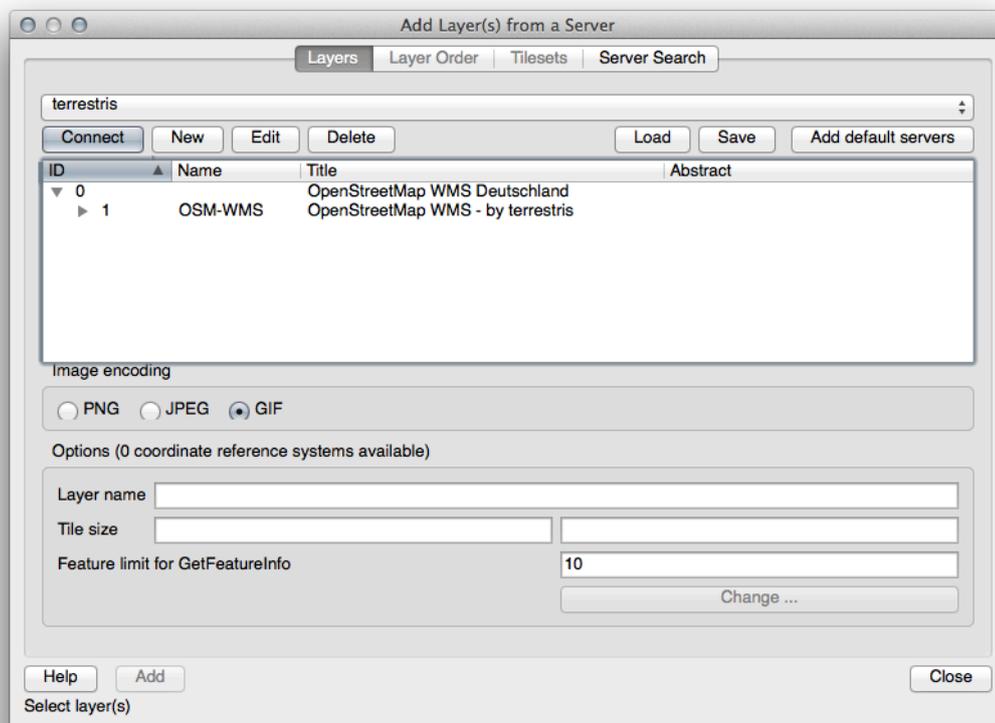
- この WMS を利用するには、このように、今表示しているダイアログで設定します。



- 名称 フィールドの値は terrestris とすべきです。
- :guilabel:の値は URL フィールドは :kbd:'http://ows.terrestris.de/OSM/service' になります。
- Click OK. You should see the new WMS server listed:

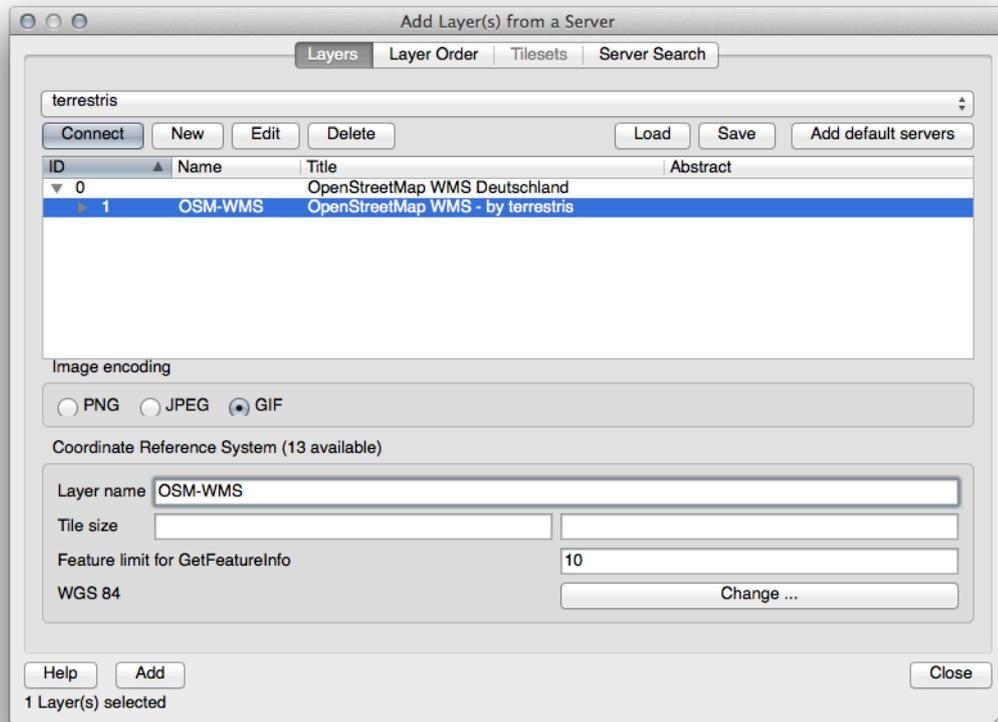


- *Connect* をクリック。以下のリストに、今これらの新しいエントリがロードされるはずですが。



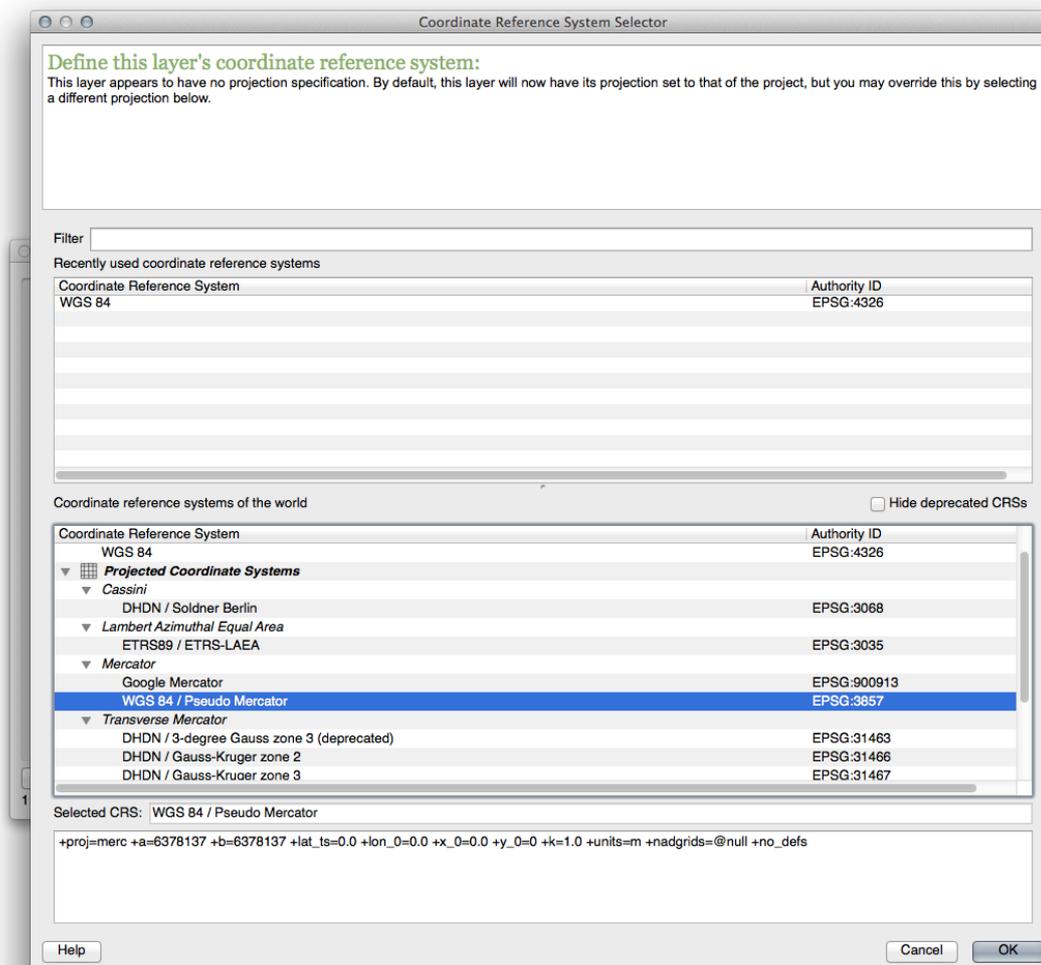
WMS サーバにホストされたレイヤがすべてあります。

- OSM-WMS レイヤーを一度クリックしてください。これはその 参照座標系 を表示するでしょう：



地図に WGS 84 を使用していないので、選択する必要がありますすべての CRS を見てみましょう。

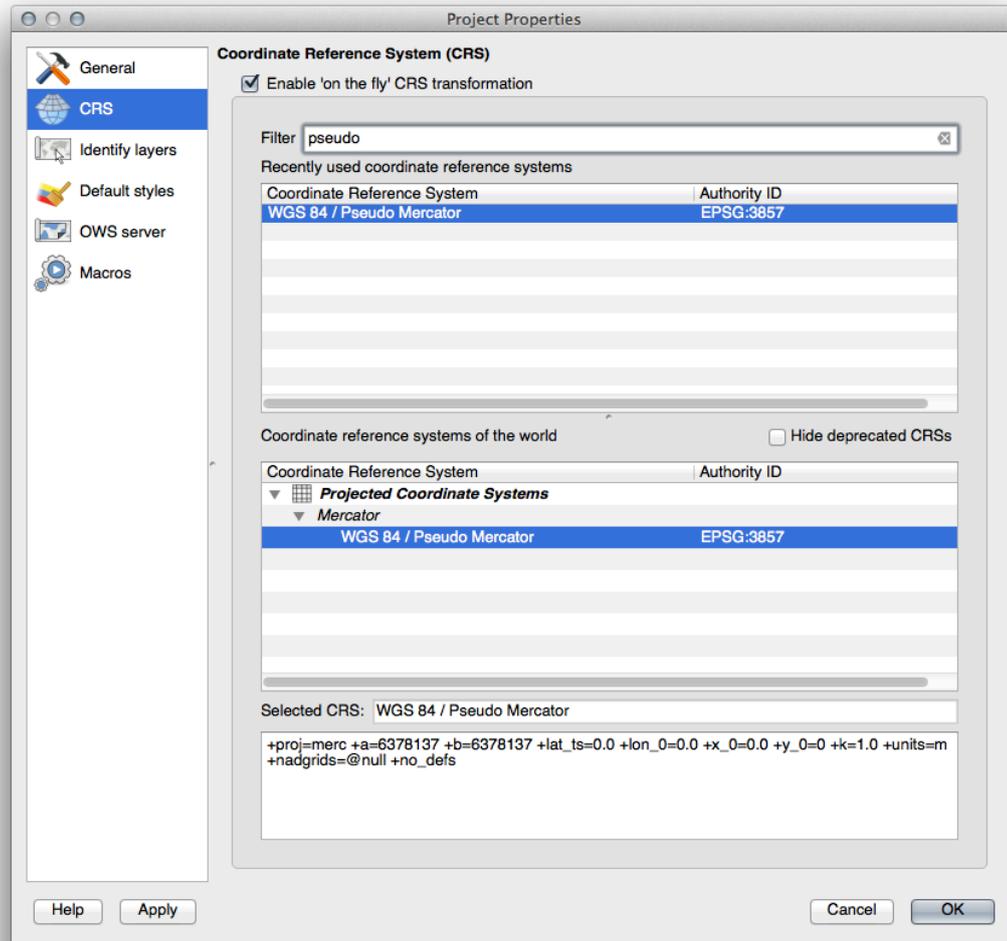
- *Change* ボタンをクリックします。標準 参照座標系 *Selector* ダイアログが表示されます。
- 投影 CRS を希望するので、*WGS 84* /疑似 *Mercator* を選択してみましょう。



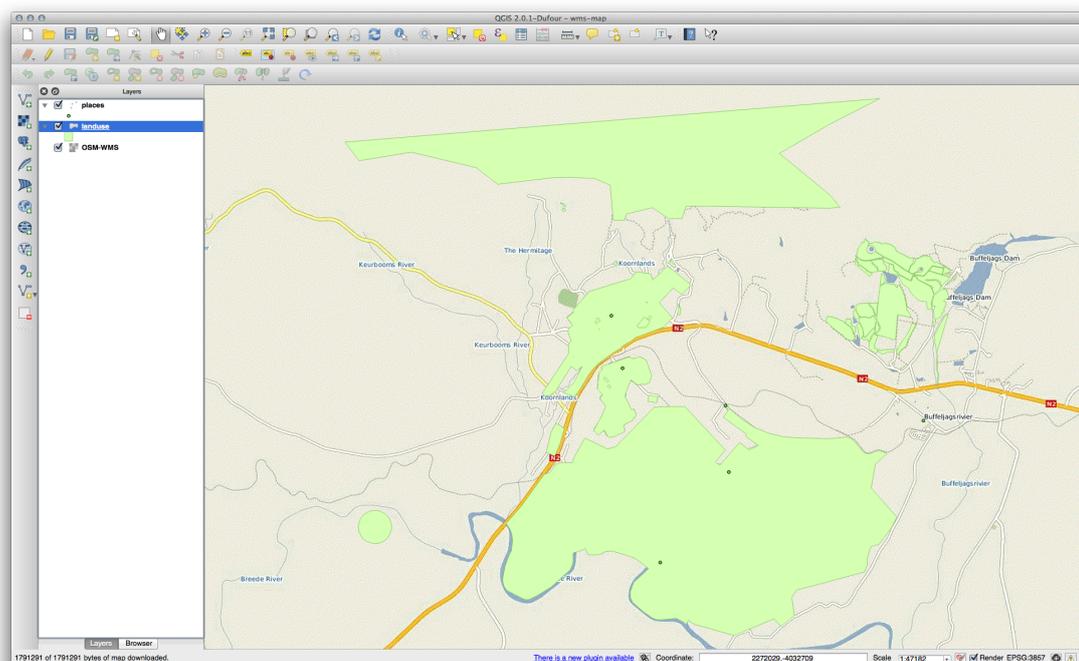
- Click *OK*.
- *Add* をクリックすると、新しいレイヤーが *OSM-WMS* として地図に表示されます。
- In the *Layers list*, click and drag it to the bottom of the list.

レイヤーが正しく配置されていないことがわかります。これは、「オンザフライ」投影が無効になっているためです。それを再度有効にしましょう、ただし *OSM-WMS* レイヤーと同じ投影を使用して、それは *WGS 84* /擬似 *Mercator* です。

- “オンザフライ”再投影を有効にします。
- *CRS* タブ (:*guilabel:* ‘プロジェクト *Properties* ダイアログ) で、 *Filter* フィールドに値 *pseudo* を入力します :



- リストから *WGS 84 / Pseudo Mercator* を選択します。
- Click *OK*.
- 今、レイヤー *list* 中の独自のレイヤーの一つに右クリックして、レイヤー範囲にズームをクリックします。Swellendam エリアが表示されるはずです：



WMS レイアの街と私たち自身の通りが重なってどのように注意してください。それは良い兆候です！

WMS の性質と限界

今までに、この WMS レイヤー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。しかも、それはベクトルで構成されているように見えるにもかかわらず、ラスタのようですが、そのシンボルを変更できません。何故でしょうか？

これは WMS がどのように動作するかです：画像として受け取るものは、紙上の通常地図に類似した、地図です。通常起こることは、ベクトルレイヤーがあり、それを QGIS が地図としてレンダリングすることです。しかし、WMS を使用して、それらのベクトルレイヤーは地図としてそれをレンダリングした画像と、その地図を送信 WMS サーバ、上にあります。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。

シンボルを心配する必要はありませんので、これは、いくつかの利点を有しています。これは、すでにできあがっていますし、任意の有能設計された WMS 上で良く見えるはずで

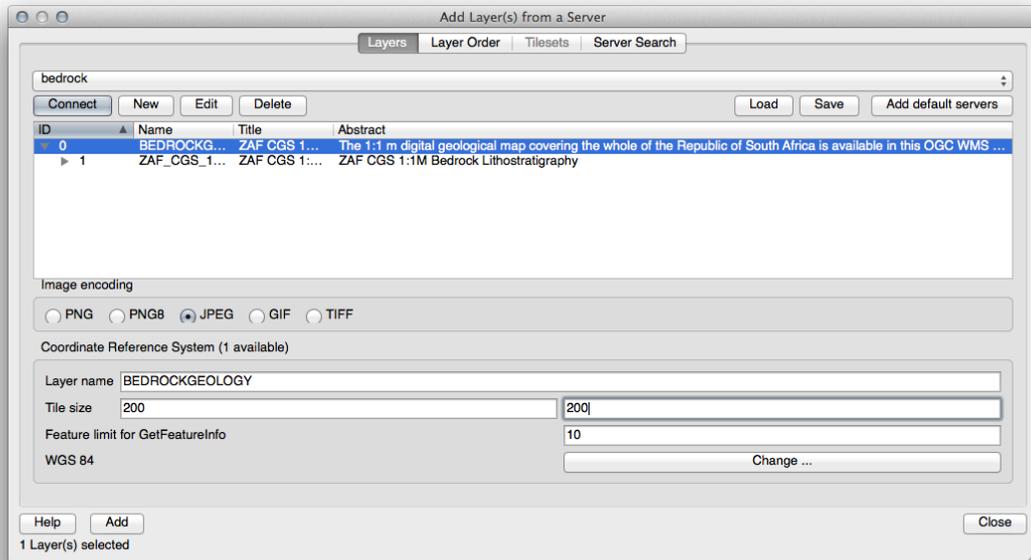
他方、気に入らなかったとしてもシンボルは変更できません、と物事が WMS サーバ上で変更した場合、その後、彼らは同様に地図に変更します。時々 WMS スタイルの地図の一部として Web 機能別にベクターレイヤーを与える代わりに、サービス (WFS)、およびないを使用したい理由です。

しかしこれは、次のレッスンで説明します。まずは、*terrestris* WMS サーバから別の WMS レイアを追加してみましょう。

11.1.2 Try Yourself

- レイヤリストで *OSM-WMS* レイアを隠します。
- この URL: http://196.33.85.22/cgi-bin/ZAF_CGS_Bedrock_Geology/wms にある “ZAF CGS 1M 岩盤岩相層序” WMS サーバを追加します。
- `guiabel`: ロードマップに ‘BEDROCKGEOLOGY’ 層を (`:guiabel`: 使用 ‘以前のように WMS Layer’ ボタンを追加します) `guiabel`: ‘WGS 84/世界 Mercator’ 投影をマップの他の部分と、それは同じでありますことをチェックすることを忘れないでください！

- より速くロードされるよう、その *Encoding* を *JPEG* に、その *Tile size* オプションを 200 掛ける 200 に設定することをお勧めします



結果の確認

11.1.3 Try Yourself

- 背景で不要にレンダリングされるのを防ぐため、すべての WMS レイヤを非表示にします。
- Add the “OGC” WMS server at this URL: <http://ogc.gbif.org:80/wms>
- *bluemarble* レイヤを追加します。

結果の確認

11.1.4 Try Yourself

WMS を使用しての難しさの一部は、良い（無料）サーバーを見つけることです。

- directory.spatineo.com <<http://directory.spatineo.com/>> _ (または他の場所でオンライン) で新しい WMS を探します。それは、関連する手数料や制限があってはならない、と以上のカバレッジを持っている必要があります | `majorUrbanName` | 学習の場。

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。

結果の確認

11.1.5 In Conclusion

WMS を使用して、既存の地図データのための背景として、非アクティブマップを追加することができます。

11.1.6 Further Reading

- *Spatineo* ディレクトリ <<http://directory.spatineo.com/>> _
- Geopole.org
- OpenStreetMap.org list of WMS servers

11.1.7 What's Next?

今、背景として、非アクティブ地図を追加したことを、それが（例えば、前に追加される他のベクトルレイヤーなど）の地物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。リモートサーバー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。それは、次のレッスンのトピックです。

11.2 Lesson: Web Feature Services

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することができます。

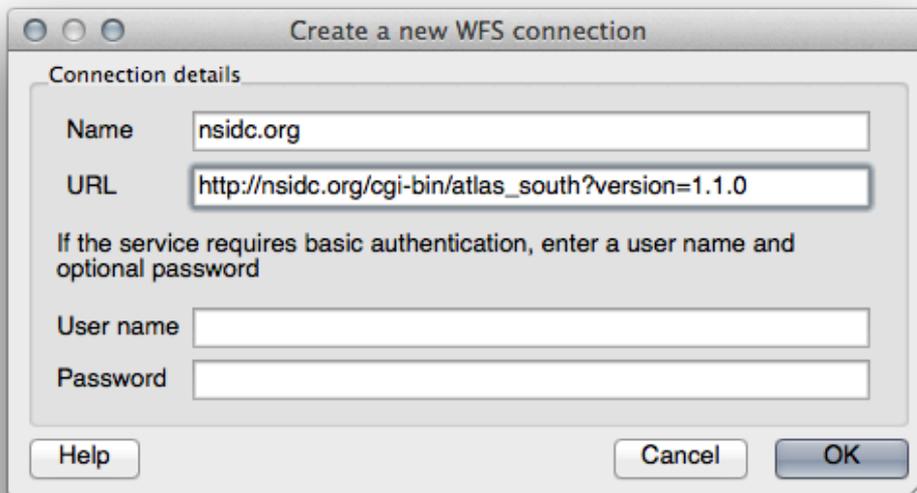
このレッスンの目標: WFS を使用して WMS との違いを理解します。

11.2.1 Follow Along: WFS レイヤーの読み込み

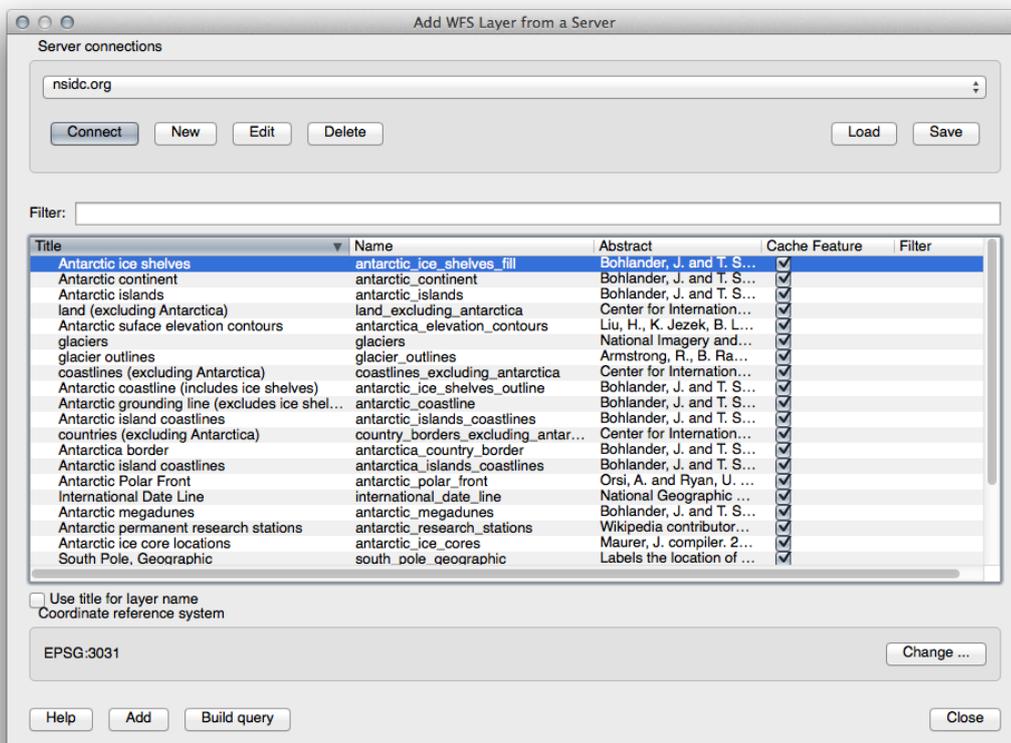
- 新しい地図を開始します。これはデモを目的としており保存されません。
- “オンザフライ” 再投影がオフになっていることを確認して下さい。
- WFS レイヤーの追加 ボタンをクリックします:



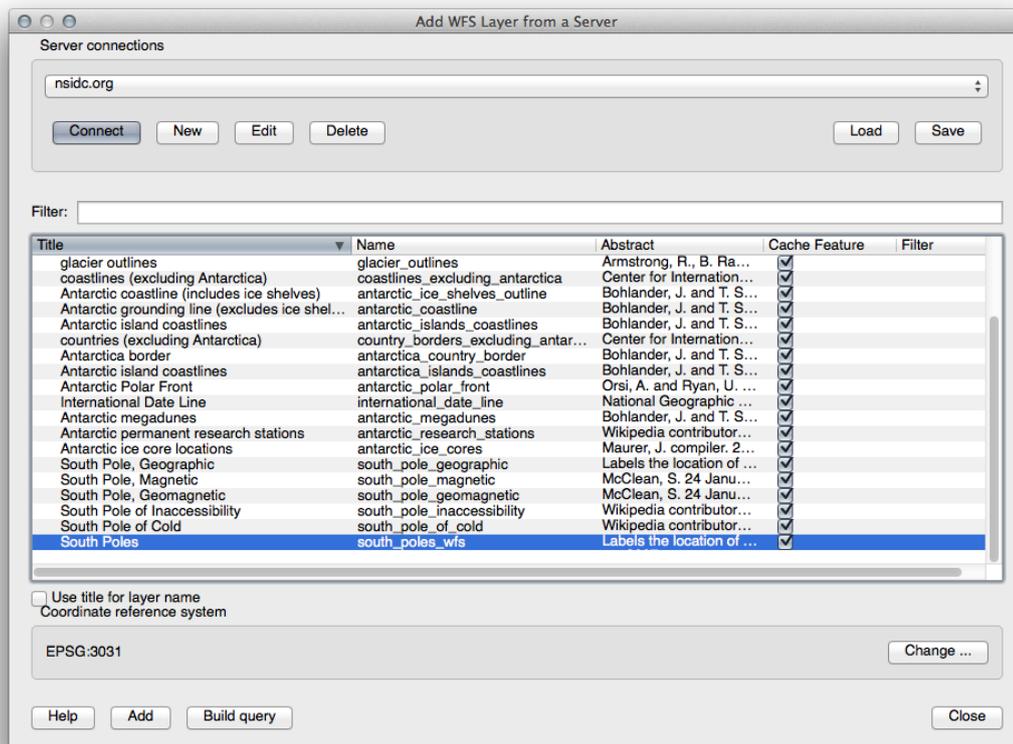
- 新規 ボタンをクリックします。
- 表示されるダイアログで 名称 に `nsidc.org`, URL に `http://nsidc.org/cgi-bin/atlas_south?version=1` と入力します。



- **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。
- **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:

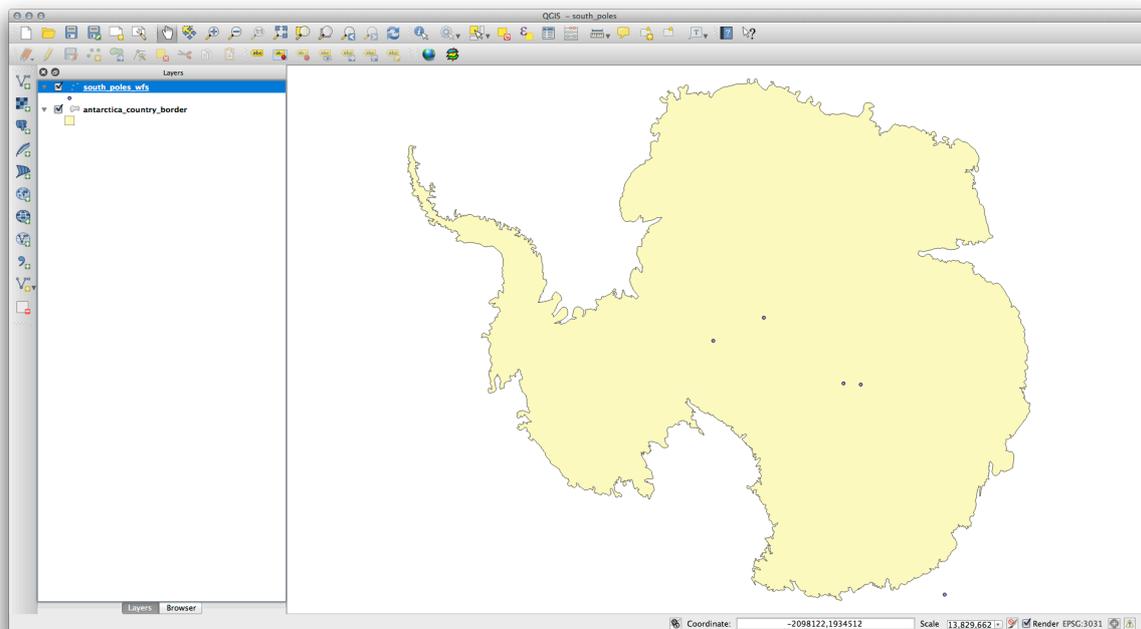


- *south_poles_wfs* レイヤを見つけます。
- レイヤをクリックして選択します:



- 追加 をクリックします。

レイヤの読み込みにしばらく時間がかかることがあります。読み込みが完了したら地図に表示されます。南極大陸の輪郭の上に表示するようになります (同じサーバで *antarctica_country_border* の名前でも利用可能です):

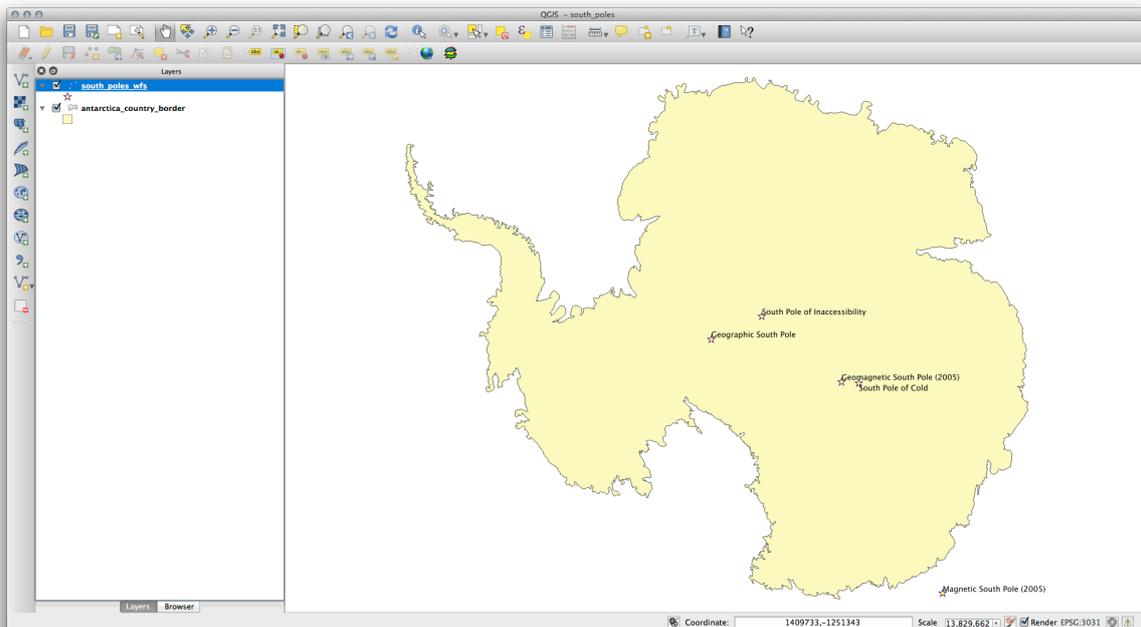


WMS レイヤとはどのように違うのですか? それはレイヤの属性を見ると明らかになります。

- *south_poles_wfs* レイヤの属性テーブルを開きます。このように見えるはずですが:

Id	NAME
0	Geographic South Pole
1	Magnetic South Pole (2005)
2	Geomagnetic South Pole (2005)
3	South Pole of Inaccessibility
4	South Pole of Cold

ポイントは属性を持つのでシンボロジを変更し、ラベルを付けることができます。例を示します:



- このレイヤの属性データを活用するためにレイヤにラベルを追加します。

WMS レイヤからの違い

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、シンボロジを変更しデータの分析を行うことができます。しかし、非常に多くのデータの送信にはコストがかかります。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。

11.2.2 Follow Along: WFS レイヤのクエリ

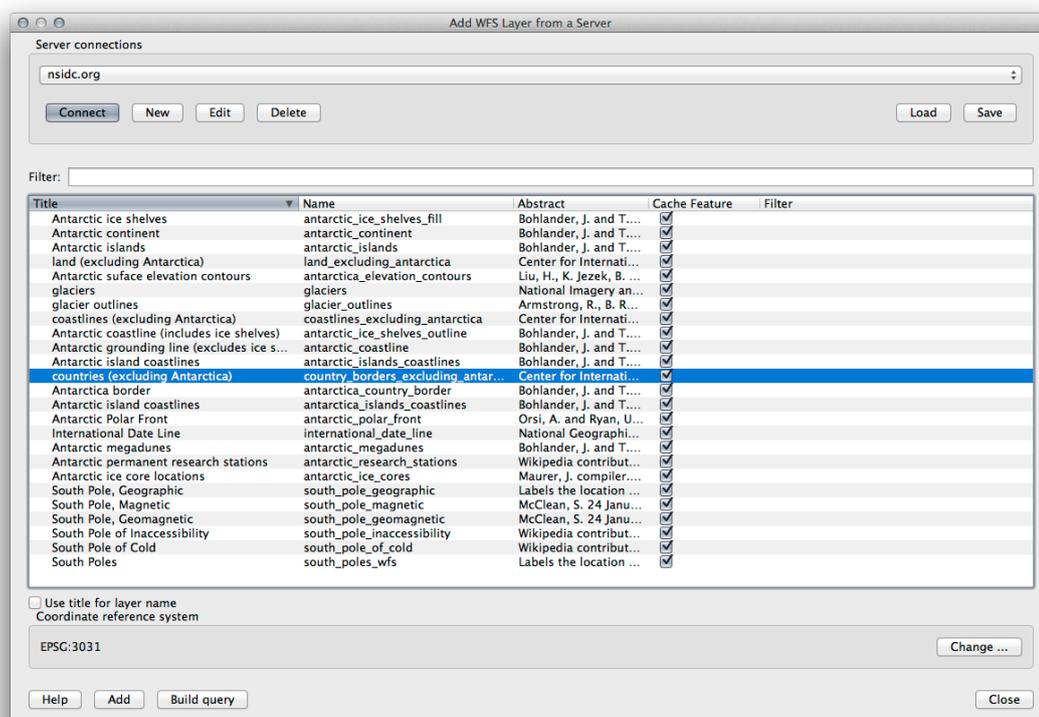
WFS レイヤをロードした後に問い合わせをすることは可能ですが、ロードする前に問い合わせをする方が多くの場合に効率的です。そのようにして必要な地物だけを要求すればはるかに少ない帯域幅の使用で済むことになります。

たとえば、現在使用している WFS サーバに *countries (excluding Antarctica)* というレイヤがあります。既に読み込まれている *south_poles_wfs* レイヤに対する南アフリカ共和国の位置を知りたいとしましょう(そしておそらく *antarctica_country_border* レイヤに対する位置も)。

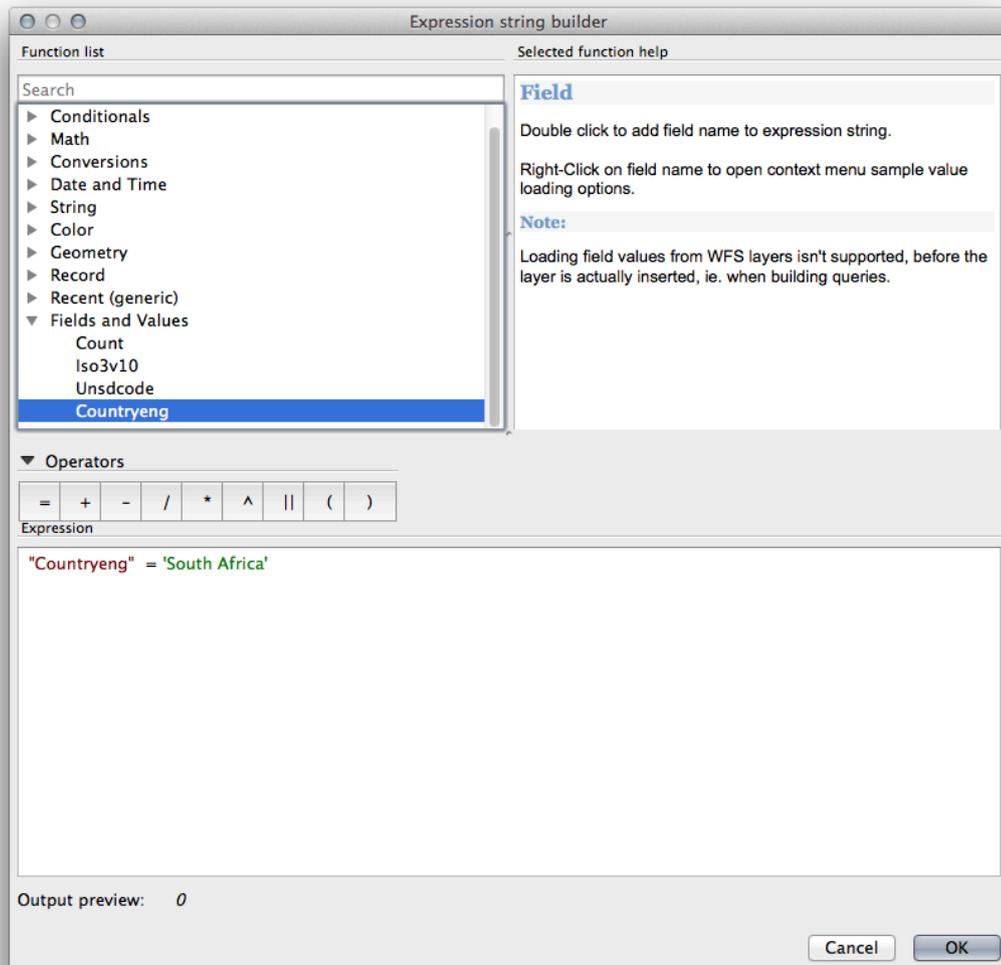
これを行うには2つの方法があります。 *countries ...* レイヤの全体をロードしてから、いつものようにクエリを作成することができますが、世界中の国のデータを送信してから南アフリカのデータだけを使用するのは少し帯域幅の無駄と思われる。あなたの接続によっては、このデータセットの読み込みに数分かかることがあります。

サーバからレイヤを読み込む前にフィルターとしてクエリを作成することもできます。

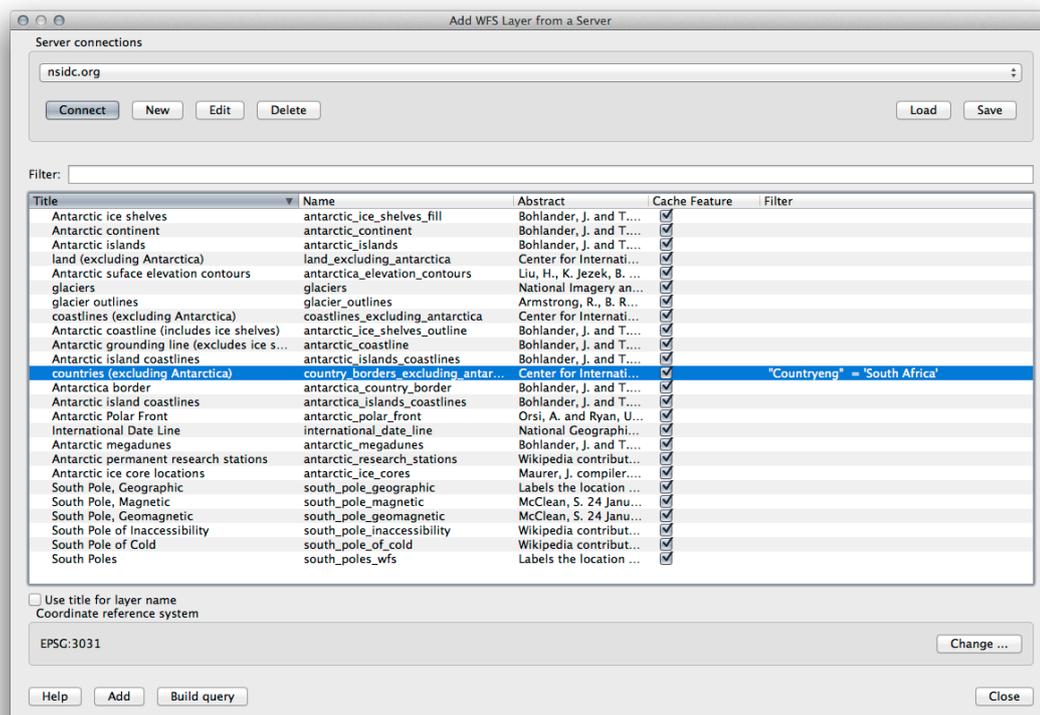
- WFS レイヤの追加 ダイアログで前に使用したサーバに接続して、利用可能なレイヤリストを表示させます。
- *Filter* フィールドの *countries ...* レイヤの隣をダブルクリックするか *クエリ作成* をクリックします:



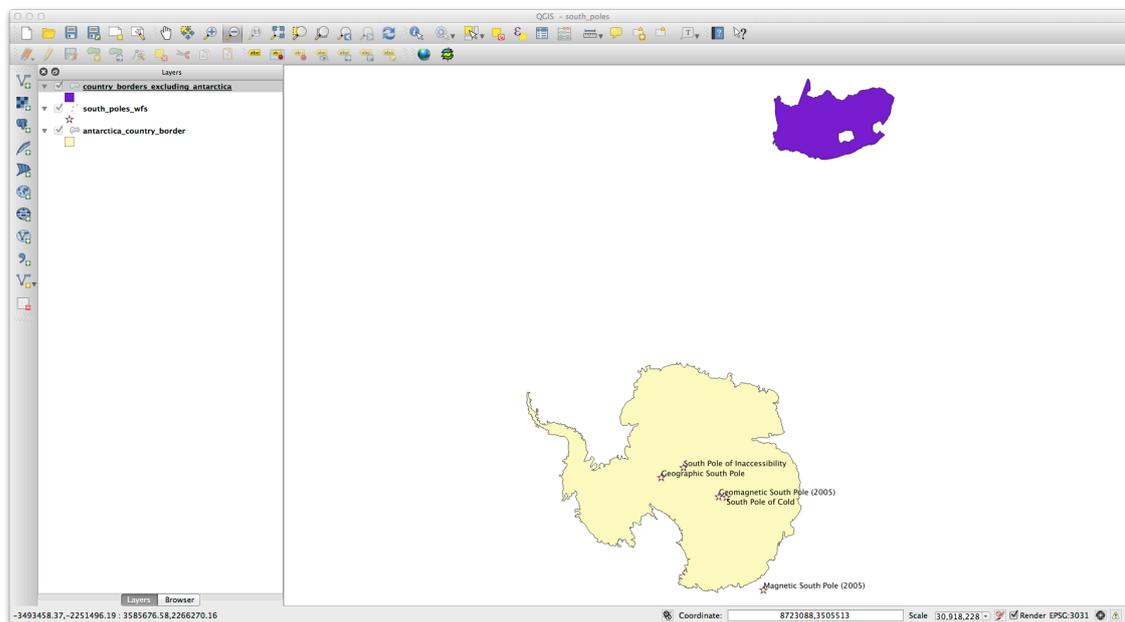
- 表示されるダイアログでクエリ `"Countryeng" = 'South Africa'` を作成します:



- *Filter* の値として表示されます:



- 上のように *countries* レイヤを選択した状態で **追加** をクリックします。Countryeng の値が South Africa の国だけがレイヤから読み込まれます:



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

あなたのニーズが非常に具体的であれば、あなたが必要とする WFS サーバを見つけることは稀です。WFS サービスが比較的新しい理由に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りよくありま

せん。

したがって、あなたが出会うであろう 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。

11.2.3 In Conclusion

あなたが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を考慮すると (速度の問題そして容易に利用可能な公開 WFS サーバの不足へとつながります)、必ずしも WMS の代わりに WFS が使用できるとは限りません。

11.2.4 What's Next?

次は有名な GRASS GIS のためのフロントエンドとして QGIS を使用する方法を説明します。

Chapter 12

Module: GRASS

GRASS (Geographic Resources Analysis Support System) 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 Lesson: GRASS Setup

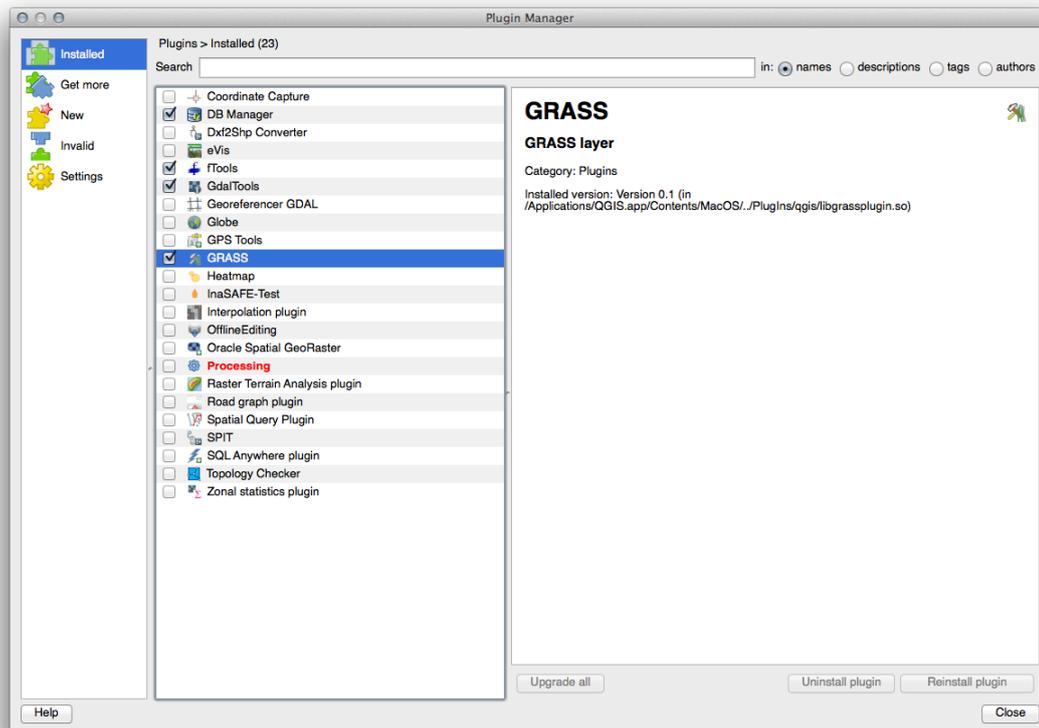
Using GRASS in QGIS requires you to think of the interface in a slightly different way. Remember that you're not working in QGIS directly, but working in GRASS *via* QGIS.

The goal for this lesson: To begin a GRASS project in QGIS.

12.1.1 Follow Along: Start a New GRASS Project

To launch GRASS from within QGIS, you need to activate it as with any other plugin. First, open a new QGIS project.

- In the *Plugin Manager*, enable *GRASS* in the list:



The GRASS toolbar will appear:



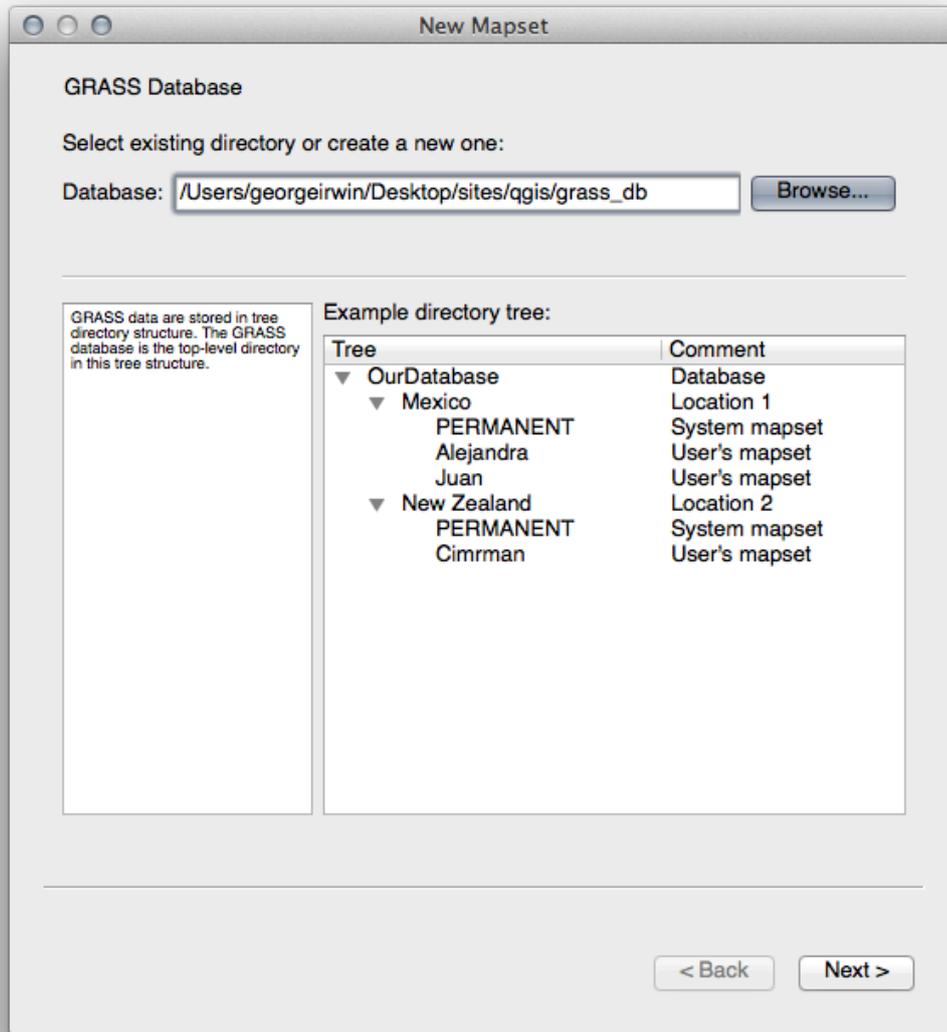
Before you can use GRASS, you need to create a **mapset**. GRASS always works in a database environment, which means that you need to import all the data you want to use into a GRASS database.

- Click on the *New mapset* button:



You'll see a dialog explaining the structure of a GRASS mapset.

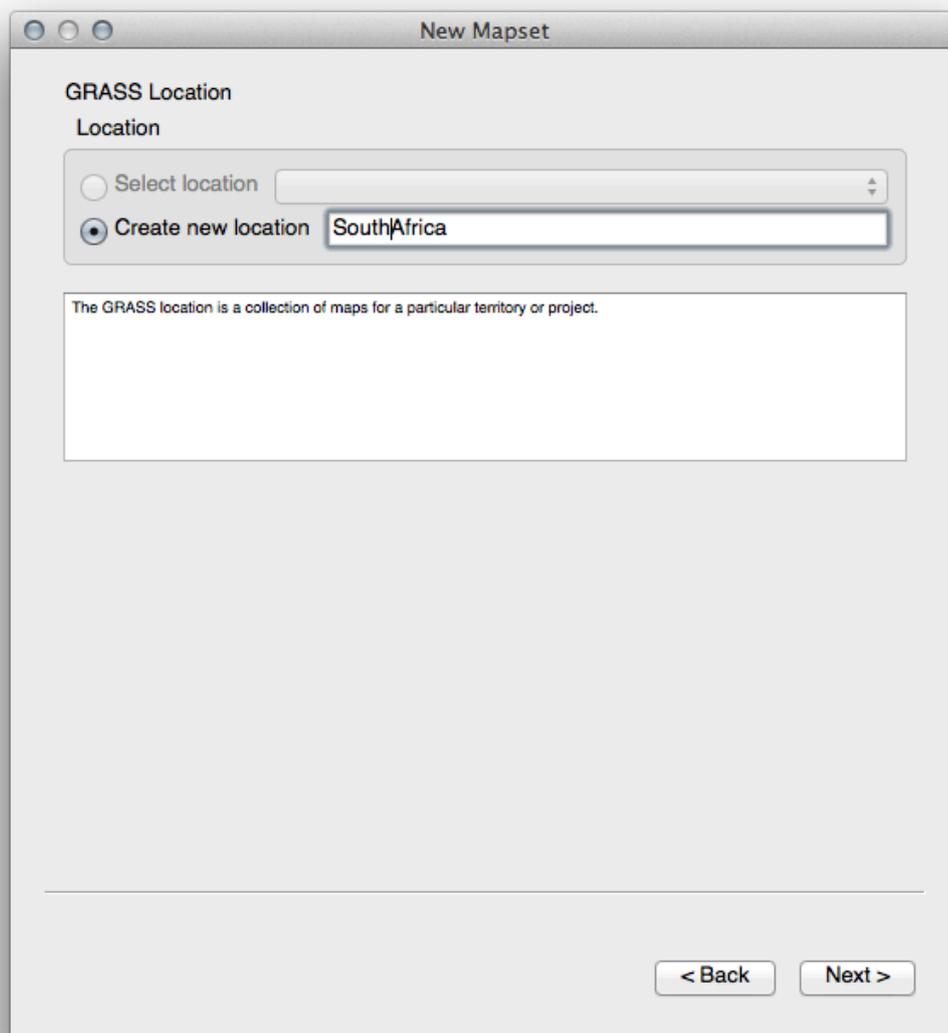
- Create a new directory called `grass_db` in *exercise_data*.
- Set it as the directory that will be used by GRASS to set up its database:



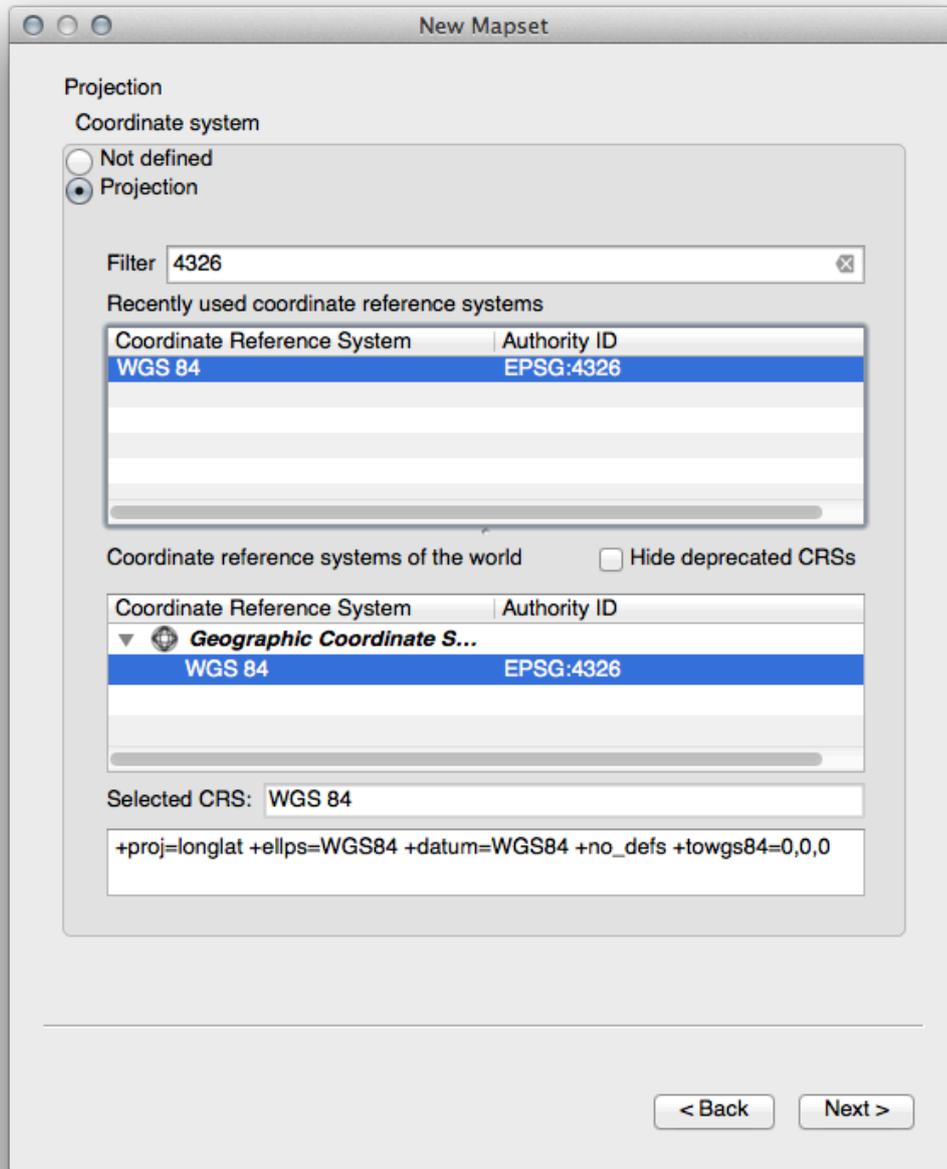
- Click *Next*.

GRASS needs to create a “location”, which describes the maximum extents of the geographic area you’ll be working in.

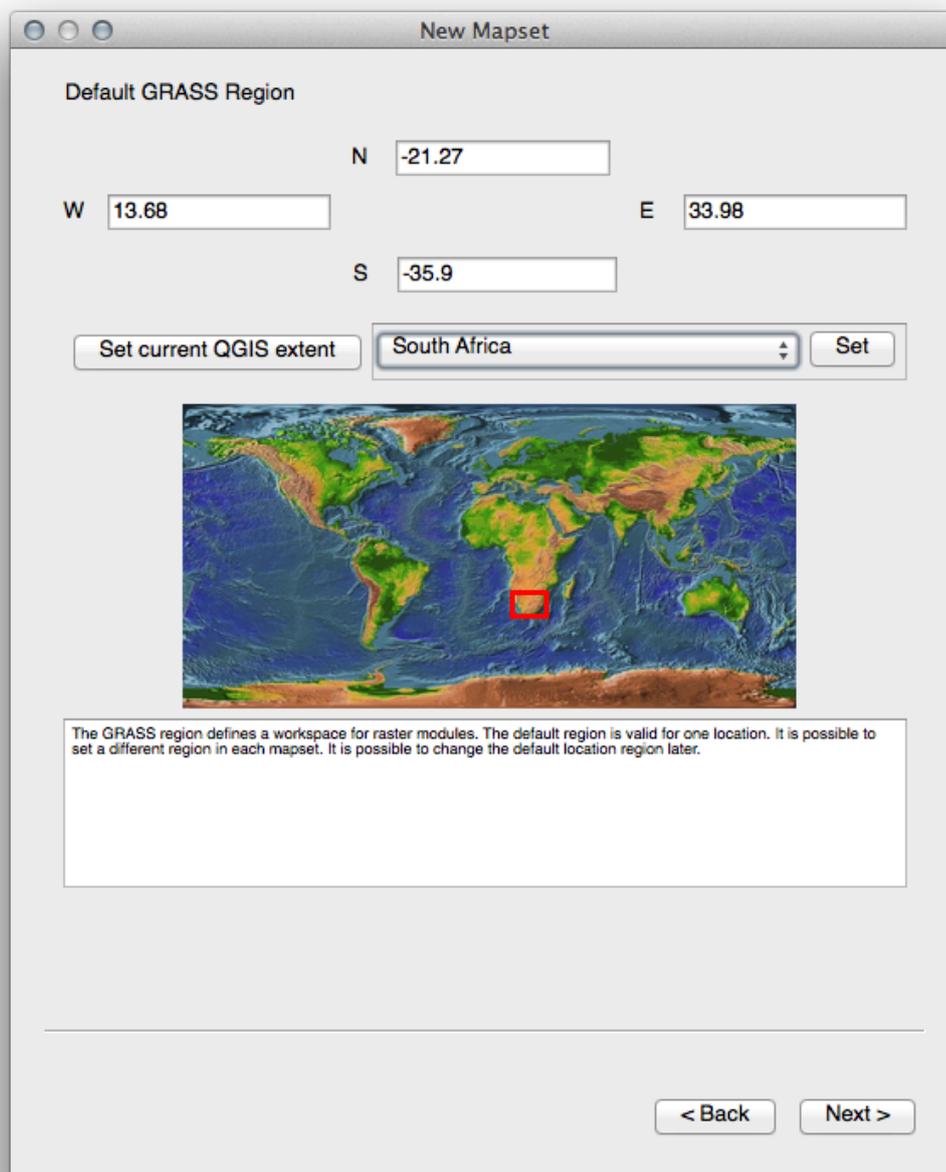
- Call the new location `South_Africa`:



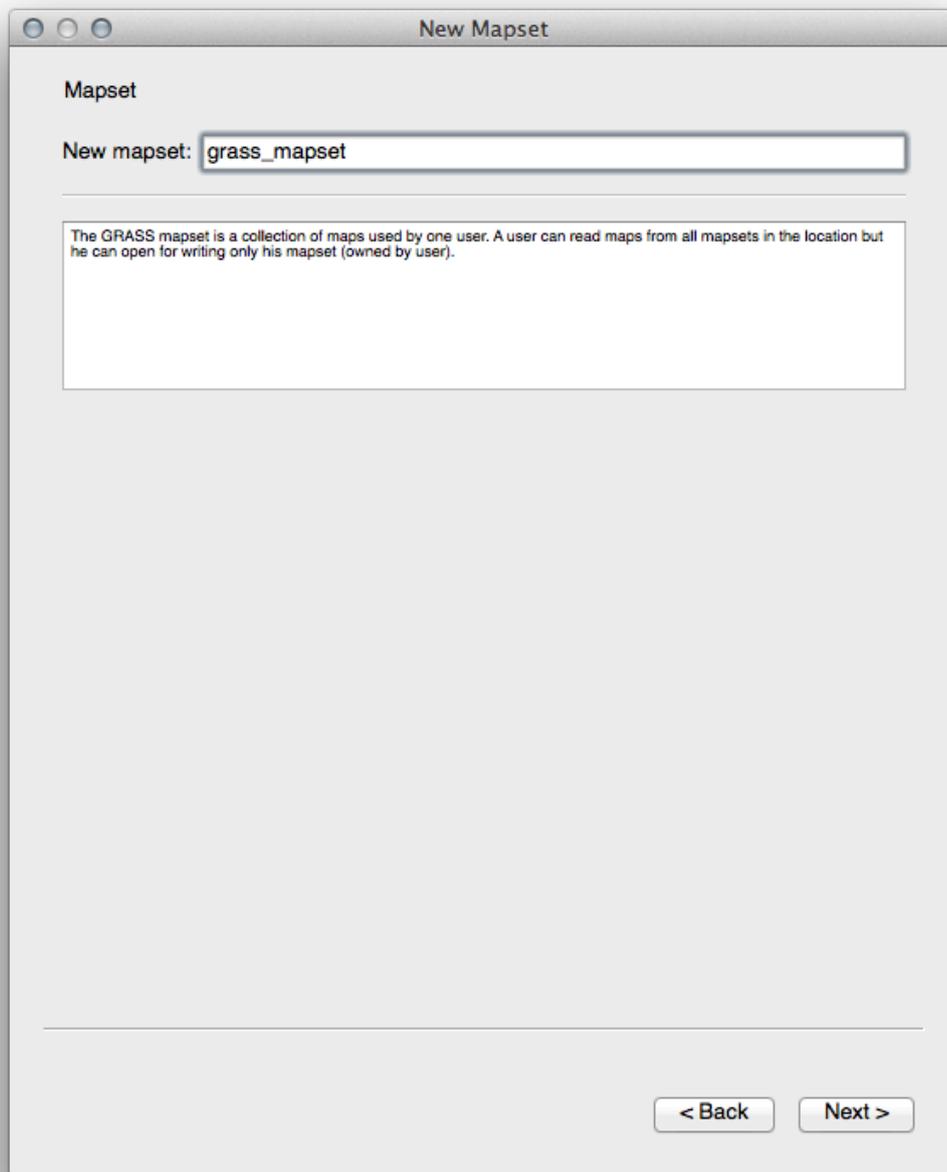
- Click *Next*.
- We'll be working with WGS 84, so search for and select this CRS:



- Click *Next*.
- Now select the region *South Africa* from the dropdown and click *Set*:



- Click *Next*.
- Create a mapset, which is the map file that you'll be working with.



Once you're done, you'll see a dialog asking you to confirm that the settings it displays are correct.

- Click *Finish*.
- Click *OK* on the success dialog.

12.1.2 Follow Along: Loading Vector Data into GRASS

You'll now have a blank map. To load data into GRASS, you need to follow a two-step process.

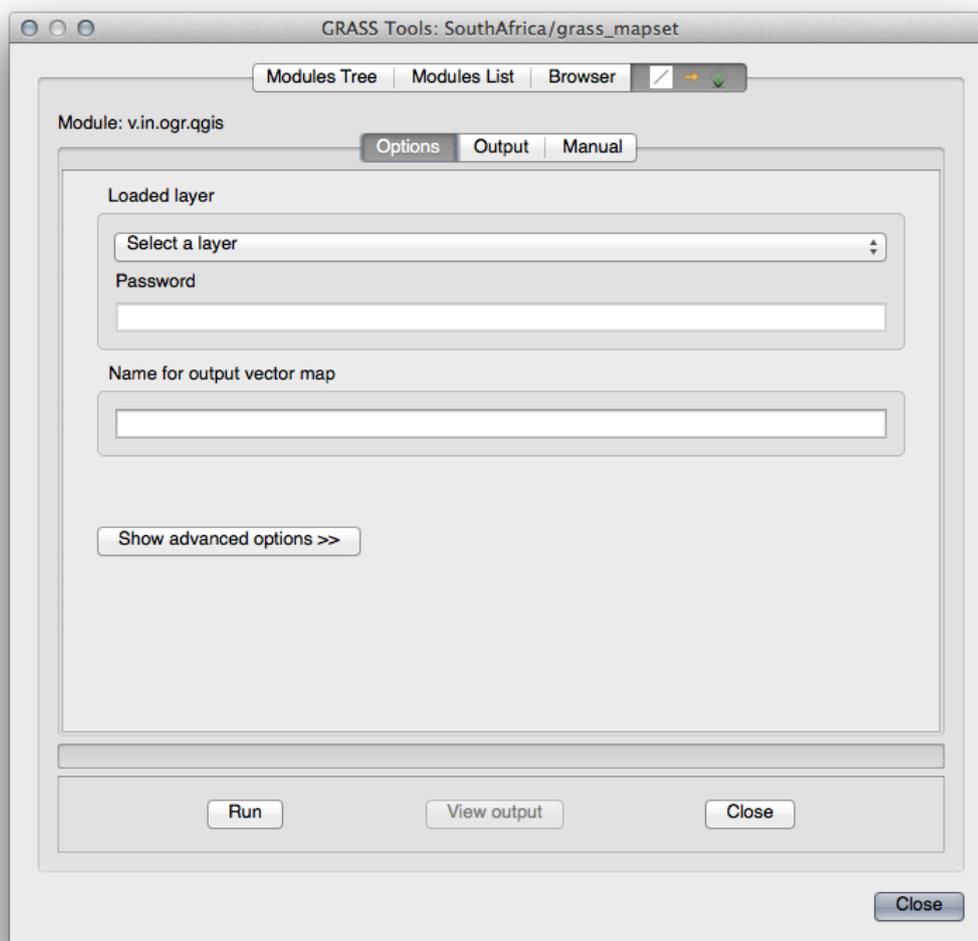
- Load data into QGIS as usual. Use the `roads.shp` dataset (found under `exercise_data/epsg4326/`) for now.
- As soon as it's loaded, click on the *GRASS Tools* button:



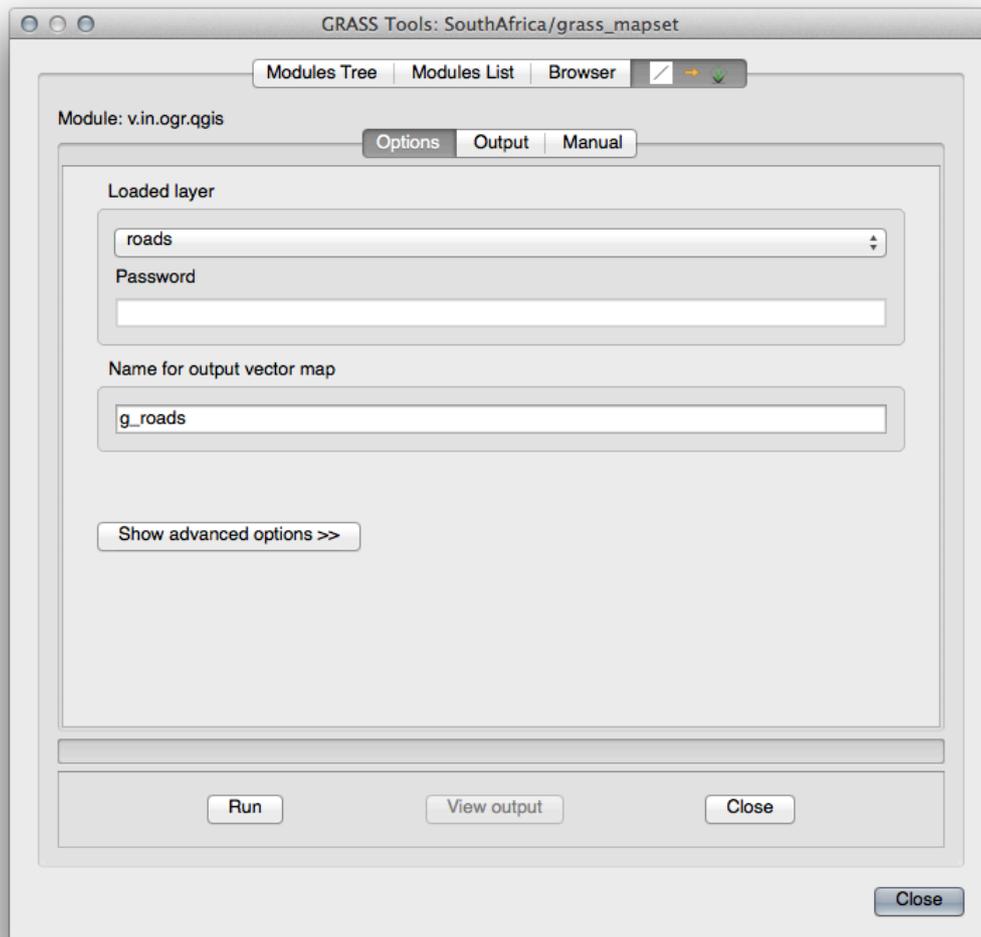
- In the new dialog, select *Modules list*.
- Find the vector import tool by entering the term `v.in.ogr.qgis` in the *Filter* field.

The `v` stands for “vector”, `in` means its a function to import data into the GRASS database, `ogr` is the software library used to read vector data, and `qgis` means that the tool will look for a vector from among the vectors already loaded into QGIS.

- Once you’ve found this tool, click on it to bring up the tool itself:



- Set the loaded layer to *roads* and its GRASS version’s name to `g_roads` to prevent confusion.



ノート:  Note the extra import options provided under *Advanced Options*. These include the ability to add a WHERE clause for the SQL query used for importing the data.

- Click *Run* to begin the import.
- When it's done, click the *View output* button to see the newly imported GRASS layer in the map.
- 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。
- Remove the original *roads* layer.

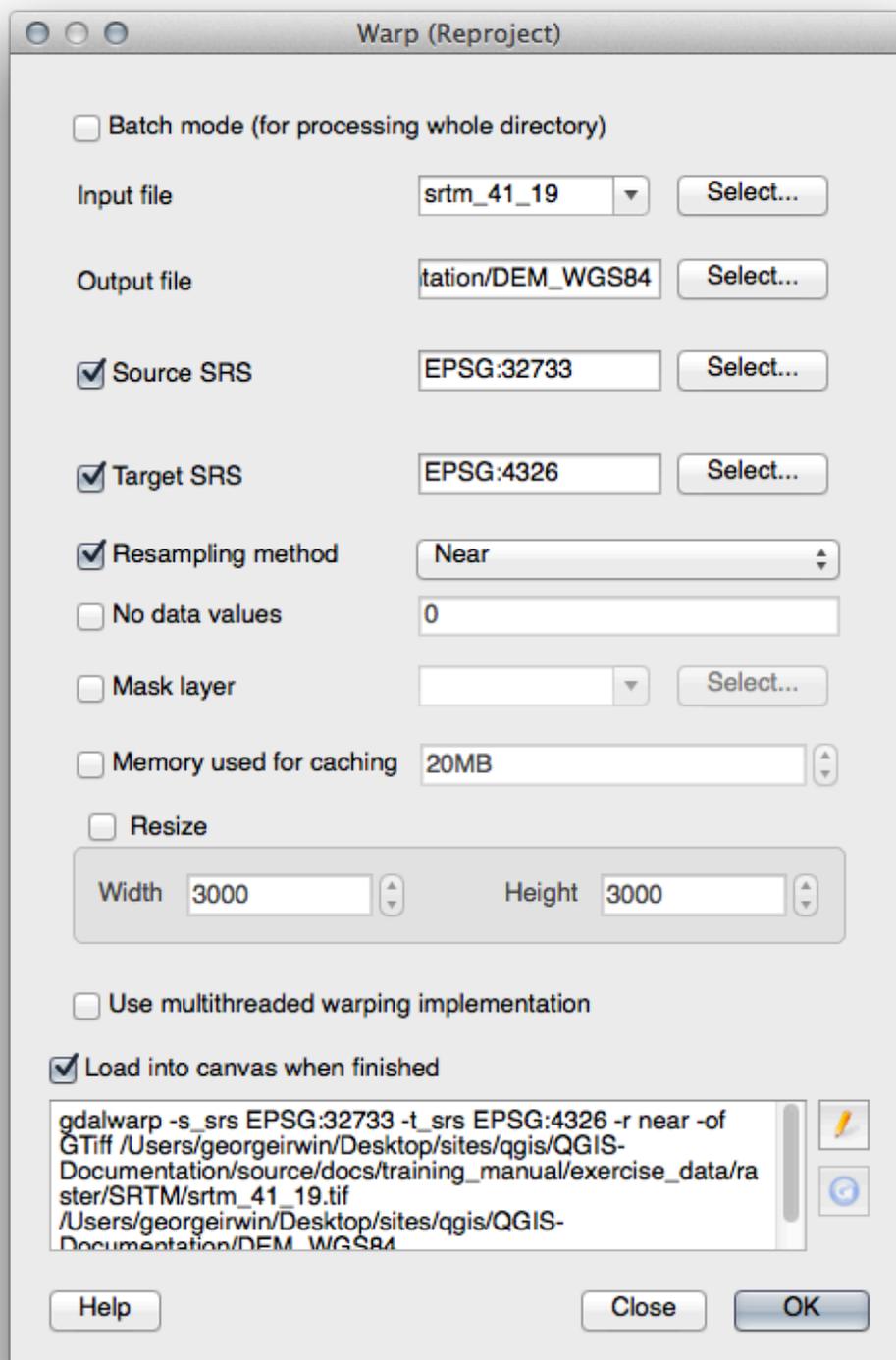
Now you are left with only the imported GRASS layer as displayed in your QGIS map.

12.1.3 Follow Along: Loading Raster Data into GRASS

Recall that our DEM is in the Projected CRS UTM 33S / WGS 84, but our GRASS project is in the Geographic CRS WGS 84. So let's re-project the DEM first.

- Load the *srtm_41_19.tif* dataset (found under *exercise_data/raster/SRTM/*) into the QGIS map as usual, using QGIS' *Add Raster Layer* tool.

- Re-project it using GDAL Warp tool (*Raster* → *Projections* → *Warp (Reproject)*), setting it up as shown:

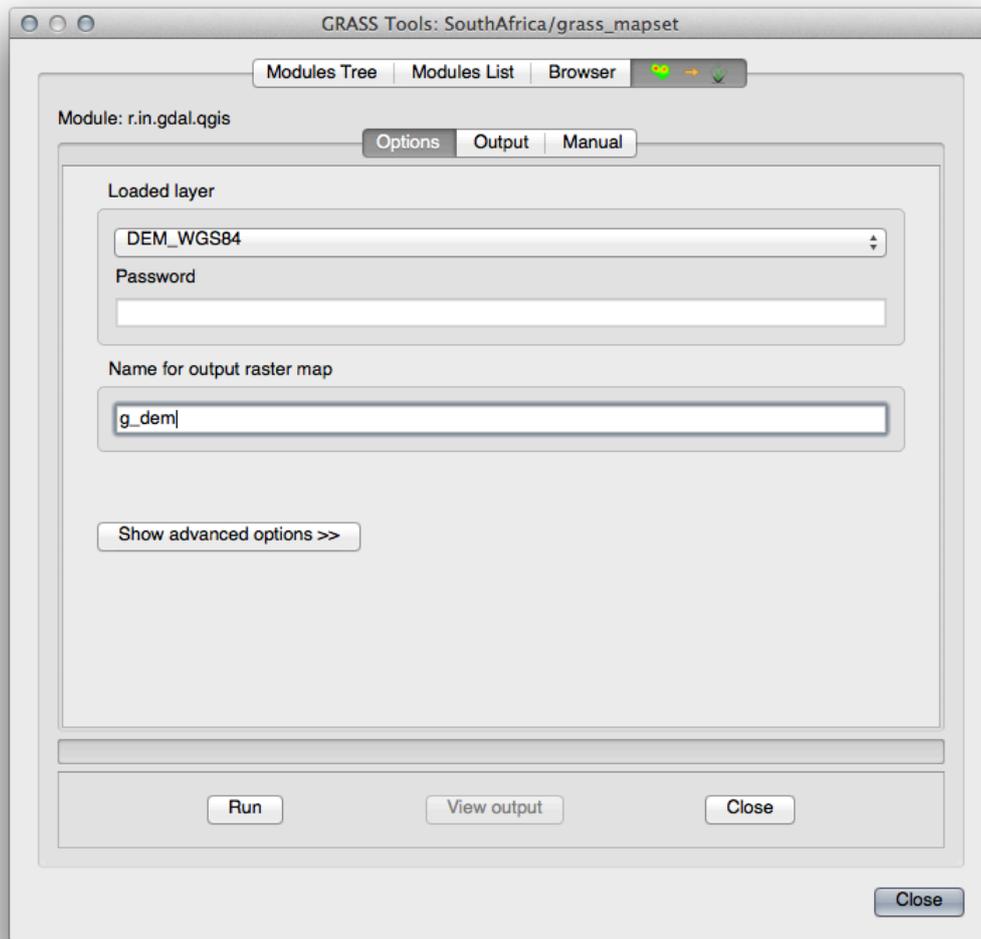


- Save the raster under the same folder as the original, but with the file name `DEM_WGS84.tif`. Once it appears in your map, remove the `srtm_41_19.tif` dataset from your *Layers list*.

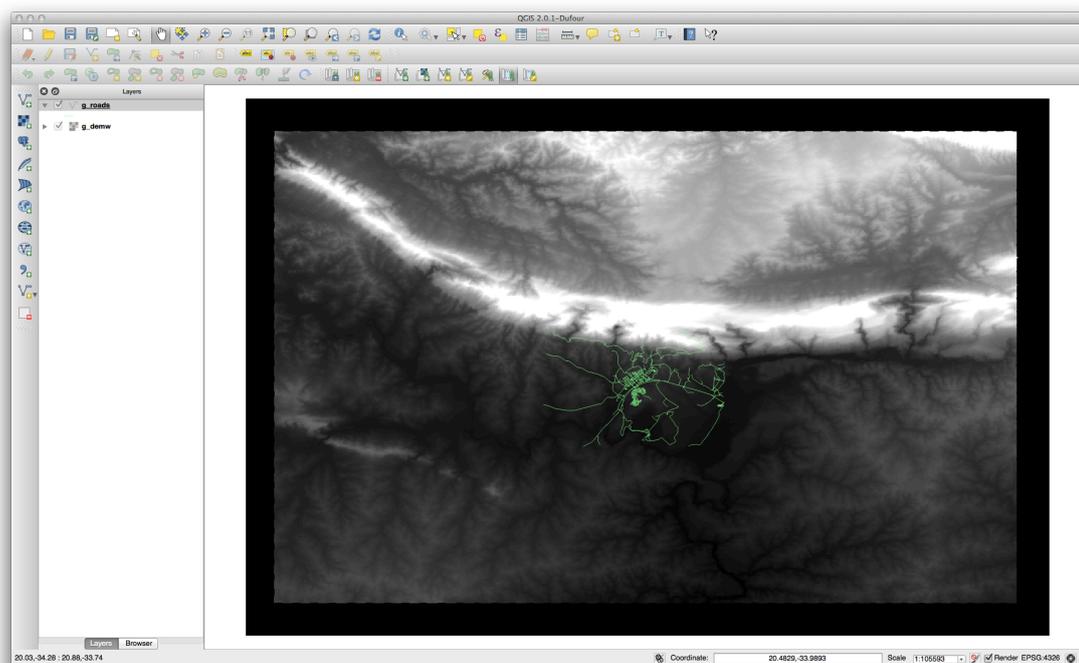
Now that it's reprojected, you can load it into your GRASS database.

- Open the *GRASS Tools* dialog again.

- Click on the *Modules List* tab.
- Search for `r.in.gdal.qgis` and double click the tool to open the tool's dialog.
- Set it up so that the input layer is `DEM_WGS84` and the output is `g_dem`.



- Click *Run*.
- When the process is done, click *View output*.
- *Close* the current tab, and then *Close* the whole dialog.



- You may now remove the original *DEM_WGS84* layer.

12.1.4 In Conclusion

The GRASS workflow for ingesting data is somewhat different from the QGIS method because GRASS loads its data into a spatial database structure. However, by using QGIS as a frontend, you can make the setup of a GRASS mapset easier by using existing layers in QGIS as data sources for GRASS.

12.1.5 What's Next?

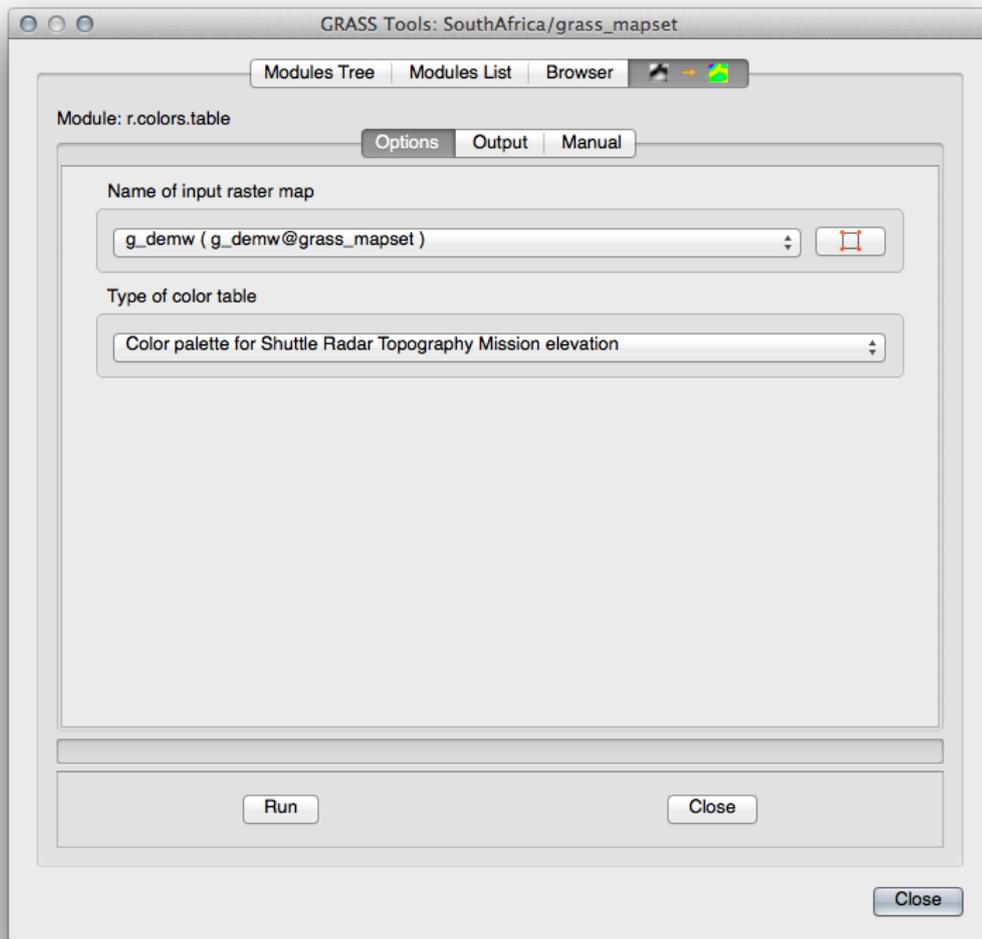
Now that the data is imported into GRASS, we can look at the advanced analysis operations that GRASS offers.

12.2 Lesson: GRASS ツール

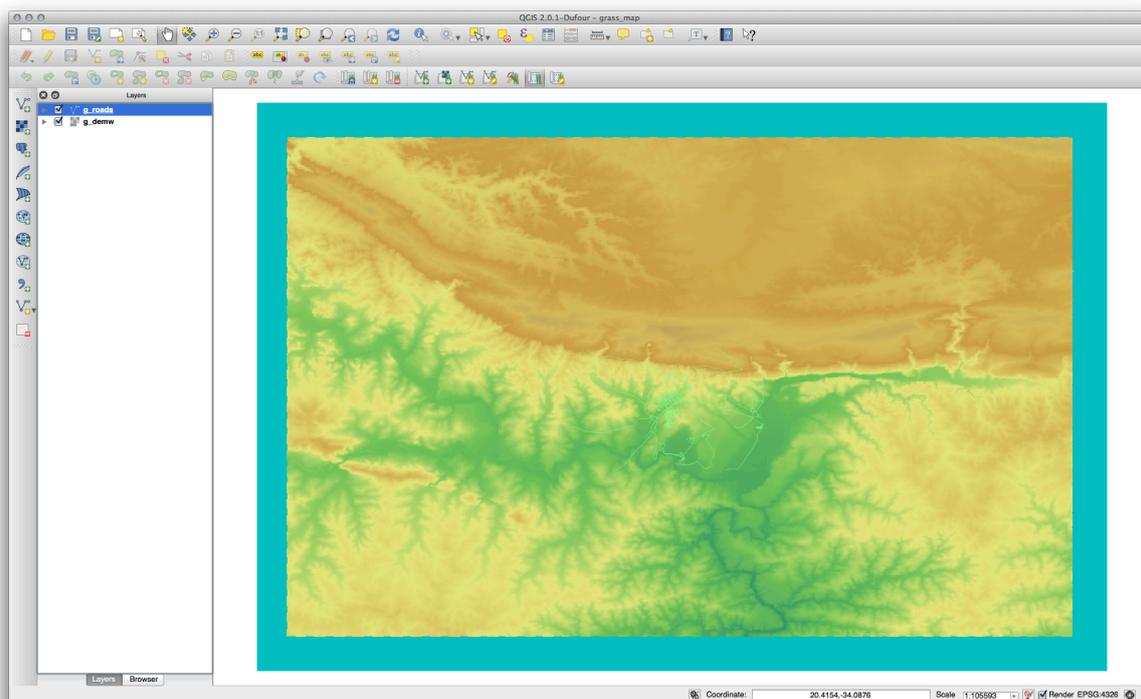
このレッスンでは、あなたに GRASS の機能についてのアイデアを与えるために選り抜きのツールを紹介します。

12.2.1 Follow Along: ラスタの色の設定

- GRASS ツール ダイアログを開きます。
- モジュールリスト タブの フィルター フィールドで検索することによって `r.colors.table` モジュールを探します。
- ツールを開き、このように設定します:



ツールを実行するとラスタの色が変更されます:



12.2.2 Follow Along: 3D でデータの視覚化

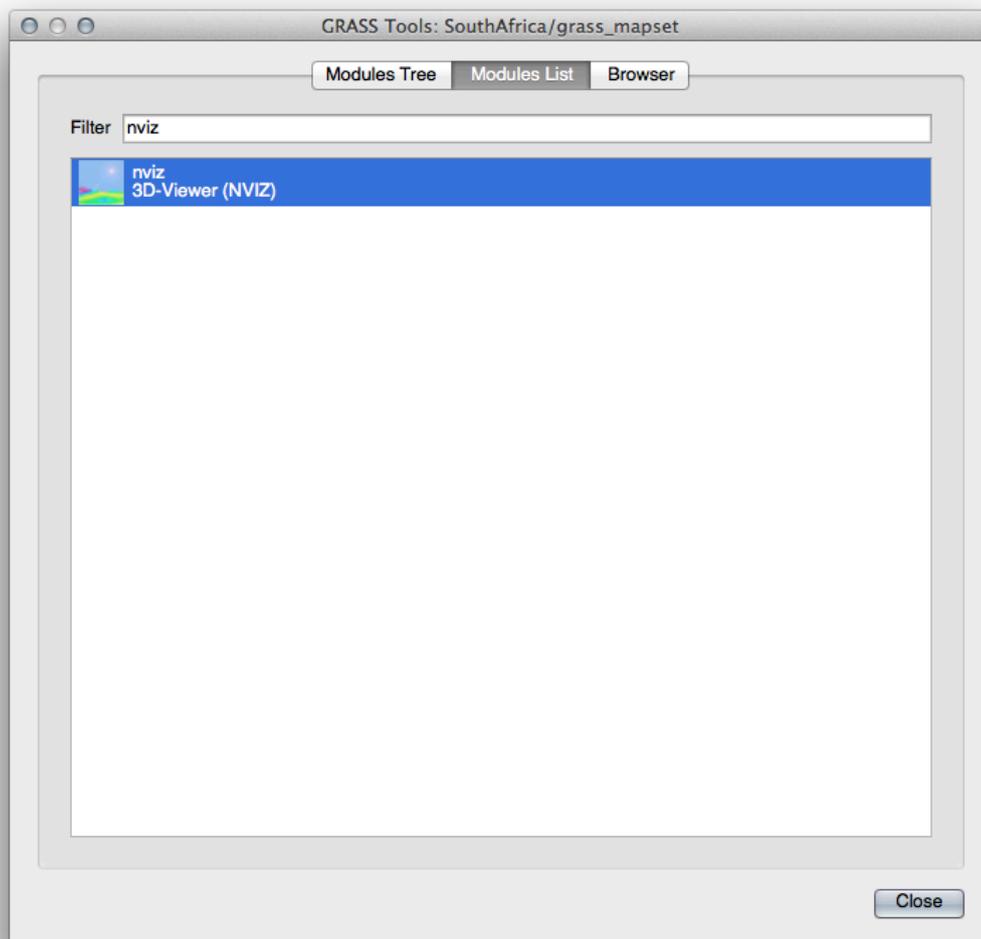
GRASS では DEM を使用して 3 次元でデータを視覚化することができます。使用しようとするツールは GRASS 領域で動作します。現時点ではあなたが前に設定した南アフリカ全体の範囲に設定されています。

- ラスタデータセットのみをカバーする範囲に再定義するために、このボタンをクリックします:

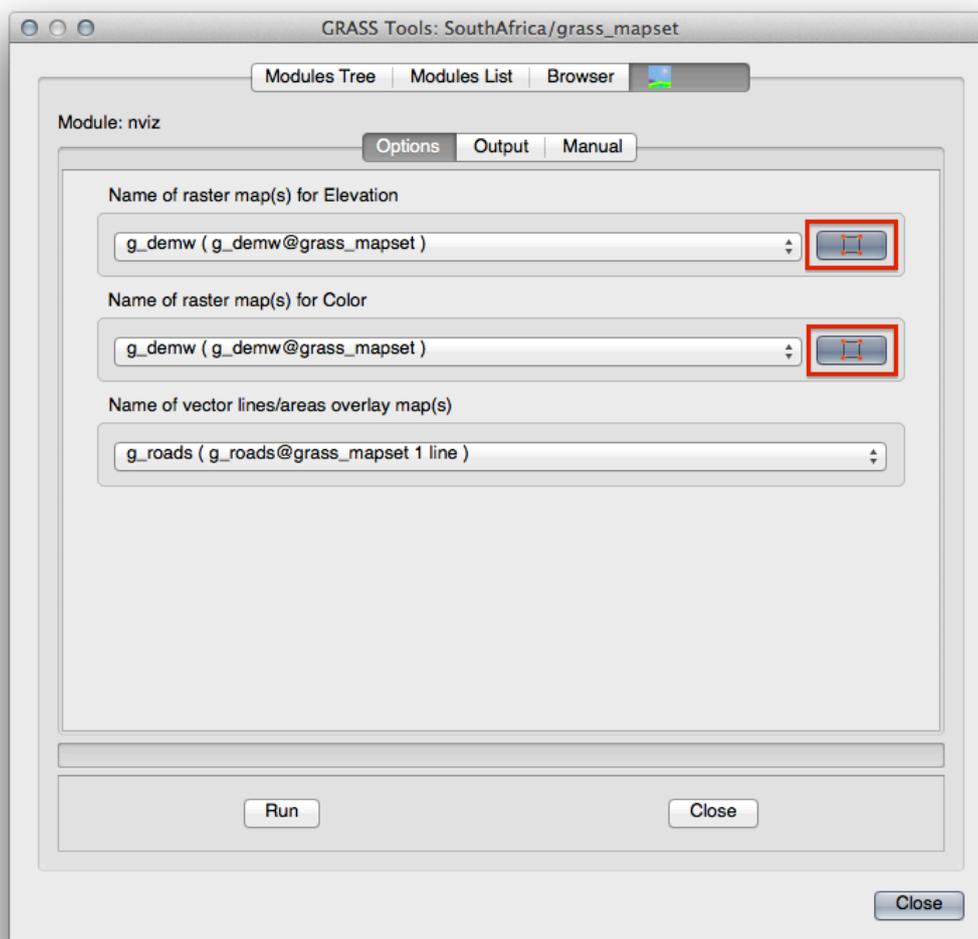


このツールがアクティブになっているとカーソルは QGIS のマップキャンバス上にある時に十字になります。

- このツールを使用して、GRASS ラスタの縁を囲む長方形を描きます。
- それができたら *GRASS Region* 設定 ダイアログの *OK* をクリックします。
- *nviz* ツールを検索します:

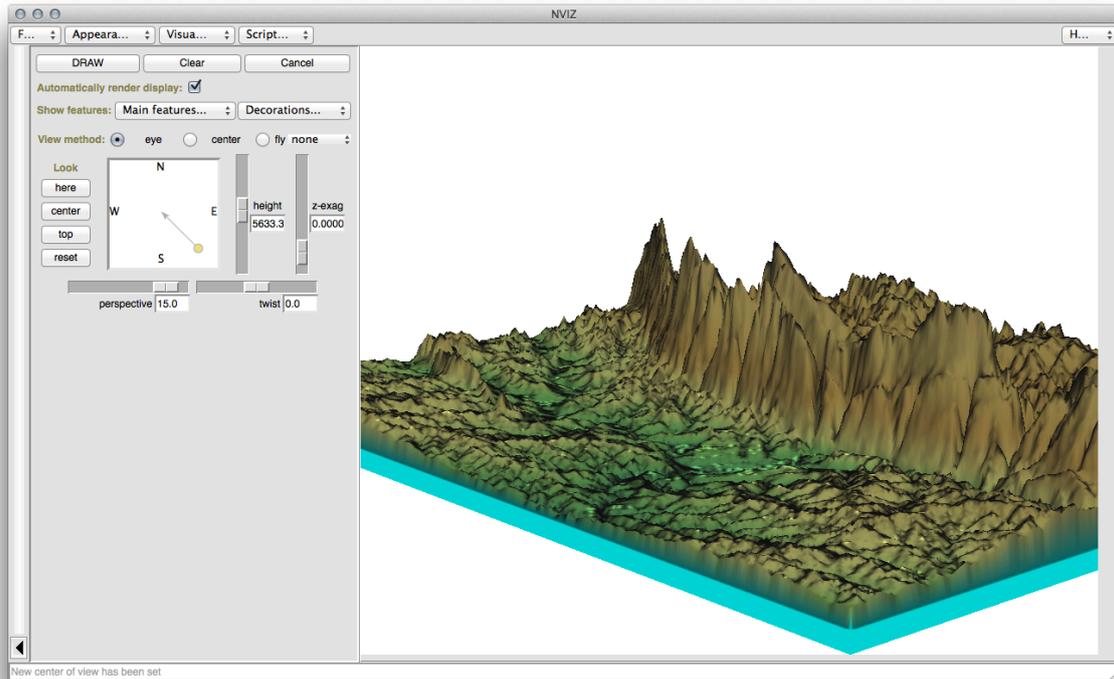


- このように設定します:



- 2つのラスタ選択ドロップダウンメニューの右側の 地図の領域を使用します ボタンを両方とも有効にして下さい。これで NVIZ がラスタの解像度を正しく評価できるようになります。
- 実行 ボタンをクリックします。

NVIZ は選択されたラスタとベクタを用いて 3D 環境を構成します。これはハードウェアに依存し時間がかかる場合があります。それが終わったら新しいウィンドウで 3D でレンダリングされた地図が表示されます:



height や *z-exag* , *View method* の設定を変えてデータの表示を変更してみましょう。ナビゲーション方法に慣れるのに少し時間がかかるかもしれません。

試用を終えたら、NVIZ ウィンドウを閉じます。

12.2.3 Follow Along: Mapcalc ツール

- GRASS ツールの モジュールリスト タブで *calc* を検索します。
- モジュールのリストから *r.mapcalc* を選択します (*r.mapcalculator* ではありません。それはより基本的なものです)。
- 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コレクションに対して実行される一連の分析を構築することができます。これらのツールを使います:



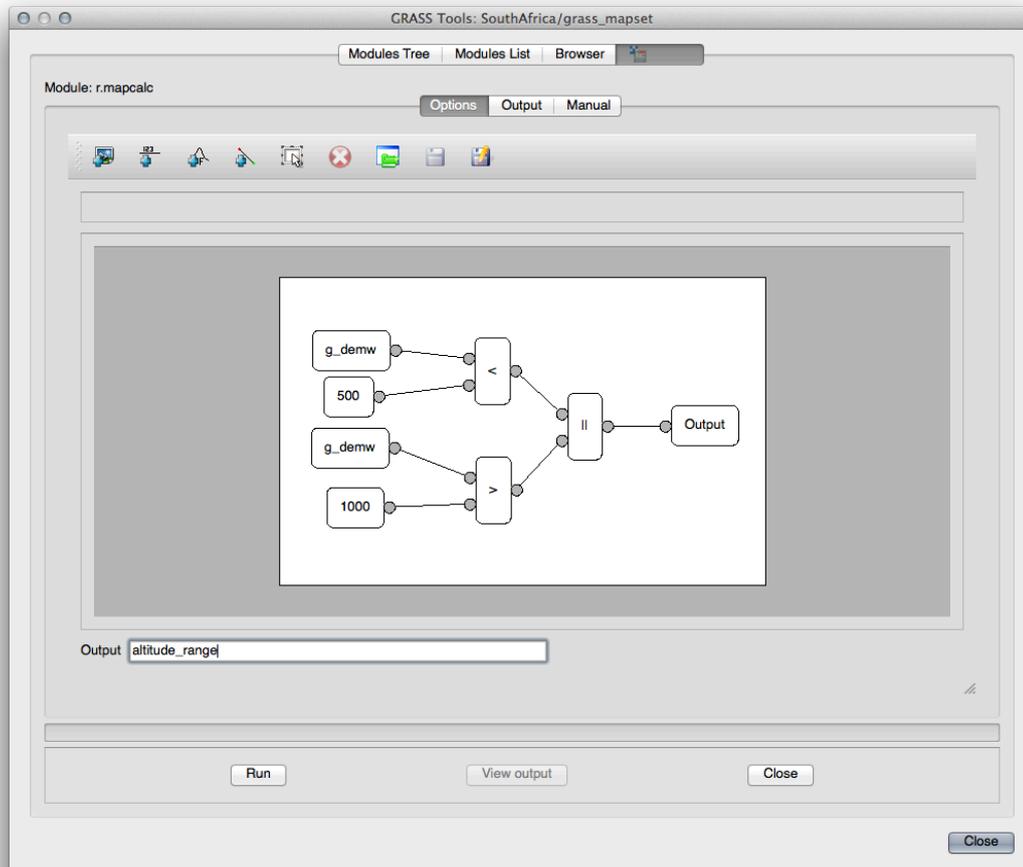
順番に:

- 地図の追加: 現在の GRASS マップセットからラスタファイルを追加します。
- 定数値を追加: 関数で使用する定数値を追加します。
- 演算子または機能を追加: 入力と出力に接続する演算子や関数を追加します。
- 接続の追加: 要素を接続します。このツールを使用して1つのアイテムの赤いドットから他のアイテムの赤いドットへクリック、ドラッグします。正しくコネクタ線に接続されたドットは灰色になります。線やドットが赤の場合、それ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。

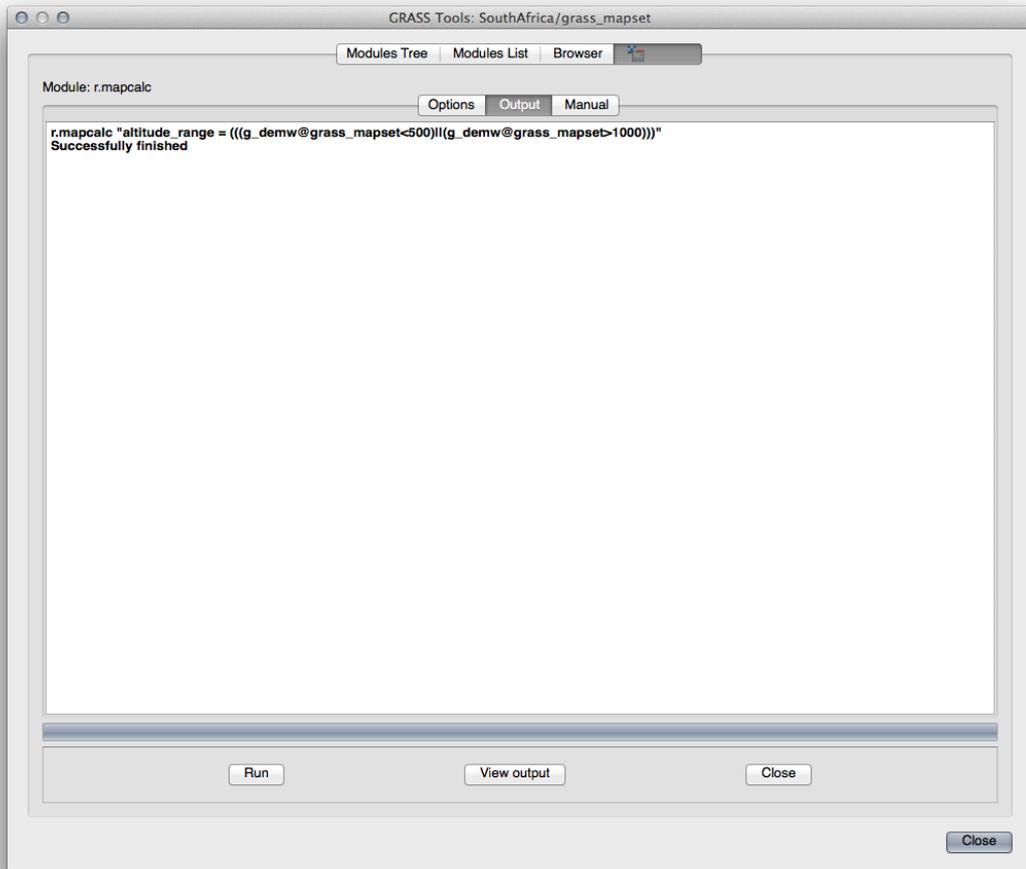
- 選択したアイテムを削除: 現在の Mapcalc シートから選択したアイテムを削除します (ラスタはマップセットからは削除されません)。

これらのツールを使用して:

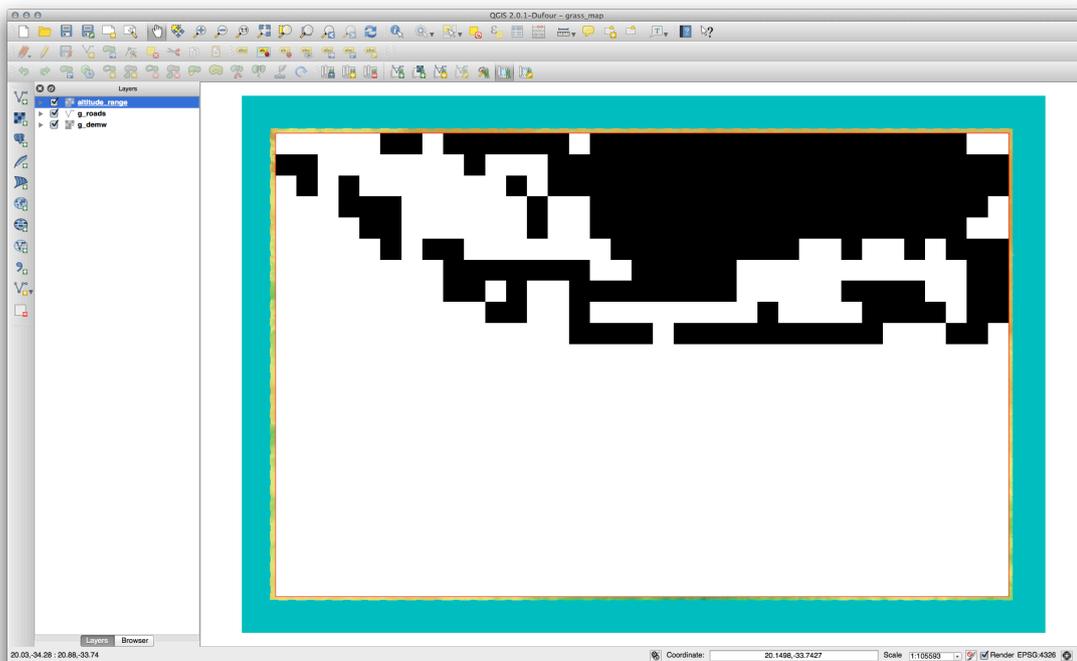
- 次のアルゴリズムを作成します:



- 実行 をクリックすると、出力はこのようになります:



- 出力を見る をクリックすると出力が地図に表示されます:



これは地形が 500 メートルより低いか 1000 メートルより高いすべての地域を示しています。

12.2.4 In Conclusion

このレッスンでは、GRASS が提供する多くのツールのうちのわずかしかがカバーしていません。自分で GRASS の機能を探索するには *GRASS ツール ダイアログ* を開き、*モジュールリスト* を下へスクロールします。より構造化されたアプローチがよければ *モジュールツリー* タブではツールが種類ごとに整理されています。

Chapter 13

Module: 評価

このセクション用に自分のデータを使います。次のものがが必要です：

- POI (ポイント名と複数のカテゴリ) の点ベクタデータセット
- 道路の線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イメージ
- DEM (持っていない場合 [この URL](#) からダウンロード可能)

13.1 ベースマップの作成

任意のデータ分析を行う前に、ベースマップが必要でしょう。これはコンテキストを持つ分析結果を提供します。

13.1.1 ポイントレイヤの追加

- ポイントレイヤーで追加します。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：



- 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、地図の主な特徴ではありません。
- カテゴリに基づいて異なる色にポイント自体を分類します。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。



-  セクションと同じことをやります。
- 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。

- 単一のポイント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、または大規模の町の名）、ポイントは何も割り当てないでください。



- レイヤーをシンボル化するためにポイントシンボルを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 *Calculator* 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移入します。

13.1.2 ラインレイヤーの追加

- 道路レイヤーを追加して、そのシンボルを変更します。道路にラベルを付けしないでください。



- 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。



- 複数のシンボルレイヤー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が、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、道路のレイヤーは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シンプルな記号。（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ー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。



- 上記の  セクション中と同じ操作を行います。
- すべてのシンボルが複数のシンボルレイヤー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 ポリゴンレイヤーの追加

- 土地利用レイヤーを追加し、そのシンボルを変更します。



- 土地利用に応じて層を分類します。ソリッドカラーを使用してください。



- 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保持。これは背景の一部になることに留意してください！



- 「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、ルールベースの分類を使用します

13.1.4 ラスター背景を作成します。

- 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 ベースマップを完成

- 上記のリソースを使用して、レイヤーのいくつかまたはすべてを使用して、基本図を作成します。この地図は、すべての基本的なユーザーを配向させるために必要な情報だけでなく、「シンプル」/視覚的に統一さを含める必要があります。

13.2 データを分析

- 一定の基準を満たす資産を探しています。
- 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。
 - ターゲットプロパティは、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、例えば病院のように、いくつかのポイントカテゴリから一定の距離内になければなりません

13.2.1



- 検索結果のラスタ分析を含めます。例えばその側面や傾きとして、ラスタの少なくとも一つの派生プロパティを考えます。

13.3 最終的な地図

- 地図 *Composer* を使用して分析結果を組み込んだ最終地図を作成してください。
- 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。

Chapter 14

Module: 森林アプリケーション

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 Lesson: 林業モジュール発表

林業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。この後のレッスンの演習では、読者はすでに QGIS での基本的な操作の多くはできるものと仮定し、前に使用されていないツールだけが詳細に提示されます。

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

ノート: このモジュールで使用す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あり、ここからダウンロード<http://qgis.org/downloads/data/training_manual_exercise_data.zip> ‘_’ できます。zip ファイルをダウンロードし、forestry\ フォルダを:kbd:exercise_data\ フォルダに解凍します。

林業関連のサンプルデータ(林業マップ、森のデータ)は、‘EVO-HAMK 林業学校’<<http://www.hamk.fi/tietoa-hamkista/kartat-ja-toimipaikat/Sivut/evo.aspx> によって提供されています> ‘_’。データセットは、レッスンのニーズに適応するように変更されました。

一般的なサンプルデータ(航空写真、の LiDAR データ、基本的なマップが)フィンランドのオープンデータサービスの国土調査から得られ、および演習の目的のために適応されています。オープンデータファイルダウンロードサービスは、ここで、<http://www.maanmittauslaitos.fi/en/file_download_service> _ 英語にアクセスすることができます。

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、こ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。独自のデータにここに記載されている技術を使用する前に、必ず適切にバックアップをとっていることを確認!

14.2 Lesson: 地図を地理参照する

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。その地域についての以前の情報は、数年さかのぼるとアナログで（つまり紙で）集めた可能性がある、またはおそらくそれはデジタル化されたが残されているのはその調査データの紙判しかない。

ほとんどの場合は、例えば、後で後の調査と比較するために、GIS でその情報を使用したいと思うでしょう。これは、GIS ソフトを用いて手で情報をデジタル化する必要があることを意味します。しかし、最初に重要なステップ、紙の地図をスキャンし地理参照しなければ、デジタル化は開始できません。

このレッスンの目標：QGIS で地理参照ツールを使用する方法を学ぶ。

14.2.1 地図をスキャン

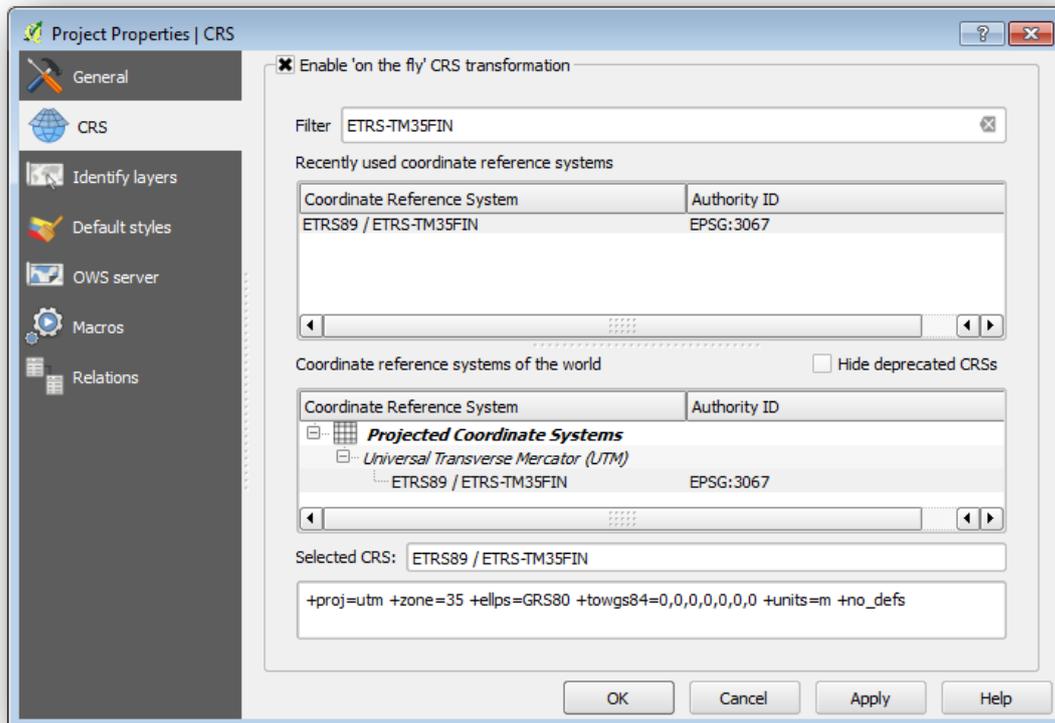
すべき最初のタスクは、地図をスキャンすることです。地図が大きすぎる場合は、部分に分けてそれをスキャンできますが、各部分について前処理とジオリファレンスの作業を繰り返す必要があることを心に留めておいてください。ですからできれば、可能な限り少ない部分に分けて地図をスキャンします。

このマニュアルで提供される一つは、画像ファイルとして地図をスキャンするために、独自のスキャナを使用する異なる地図を使用しようとしている場合は、300 DPI の解像度が行います。地図が色付きの場合は色付き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ー（例えば、林分、道路、等高線...）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。 `exercise_data/forestry data` フォルダ内に `rautjarvi_map.tif` として見つかります

14.2.2 Follow Along: スキャンした地図を地理参照する

QGIS を開き、**menuselection:プロジェクト->プロジェクトのプロパティ->CRS** でプロジェクトの CRS を `ETRS89/ETRS-TM35FIN` 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。別の CRS である古いデータを扱うので、*'on the fly'* CRS 変換を有効にする がチェックされていることを確認してください。



map_digitizing.qgs として QGIS プロジェクトを保存します。

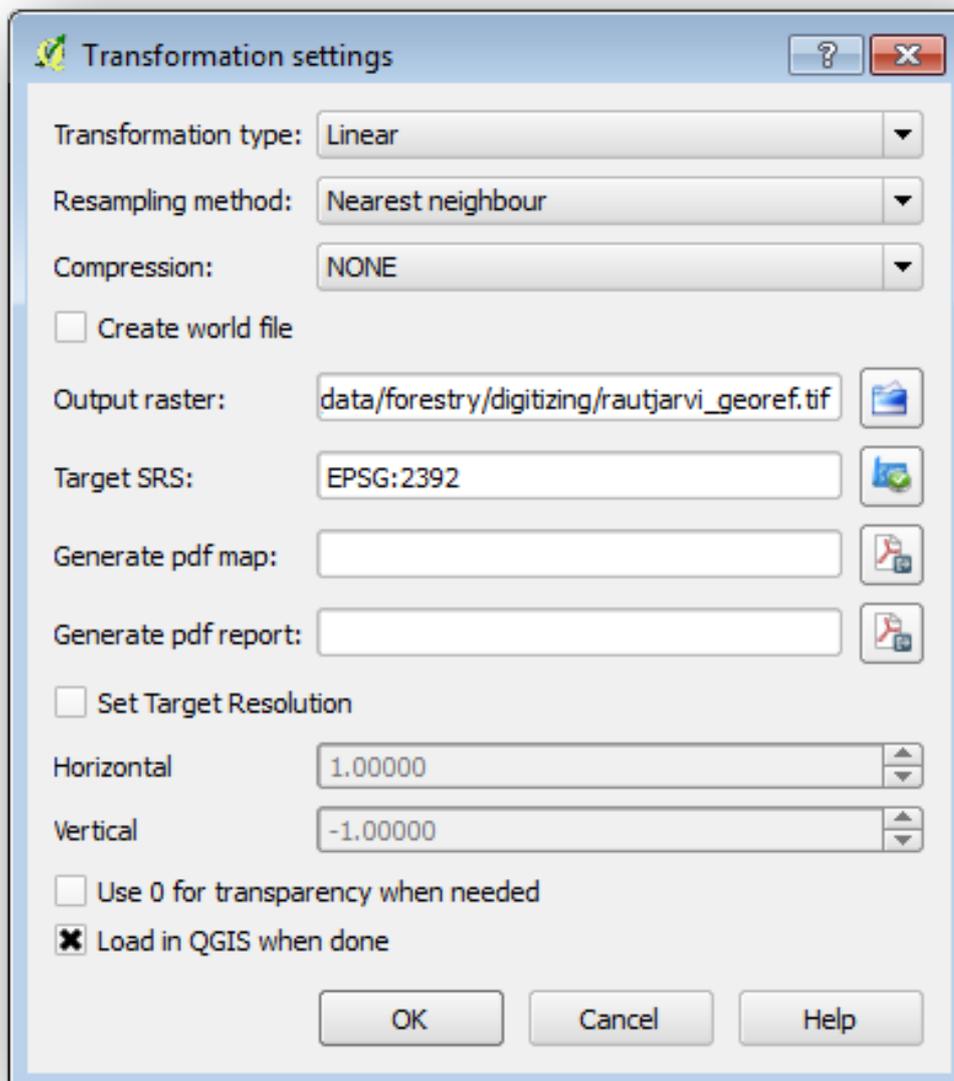
QGIS から地理参照 プラグイン を使用します。この プラグイン はすでに QGIS にインストールされています。前のモジュールで行ったように、プラグイン マネージャを使用して プラグイン を有効にします。プラグイン の名前は *Georeferencer GDAL* です。

地図を地理参照するには：

- 地理参照ツールを開きます ラスタ->*Georeferencer*->*Georeferencer*。
- 地理参照する画像として地図画像ファイル rautjarvi_map.tif を追加します、ファイル-> ラスタを開く。
- 求められたら KKJ/フィンランドゾーン 2 CRS を検索して選択します。それは昔この地図が作成された 1994 年にフィンランドで使用されていた CRS です。
- OK をクリックします。

次は地図を地理参照するための変換設定を定義する必要があります。

- 設定 ->変換設定 を開きます。
- 出力ラスタ ボックスの隣にあるアイコンをクリックして、フォルダに移動し、フォルダ exercise_data\林業\digitizing を作成し、ファイルの名前を rautjarvi_georef.tif とします。
- 下記に示すように、残りのパラメータを設定します。



- OK をクリックします。

地図には、地図中の座標をマークするいくつかの十字線が含まれていて、それらはこの画像を地理参照するために使用します。地理参照のためのウィンドウ内の画像を検査するには、QGIS で通常そうであるようにズームやパンツールを使用できます。

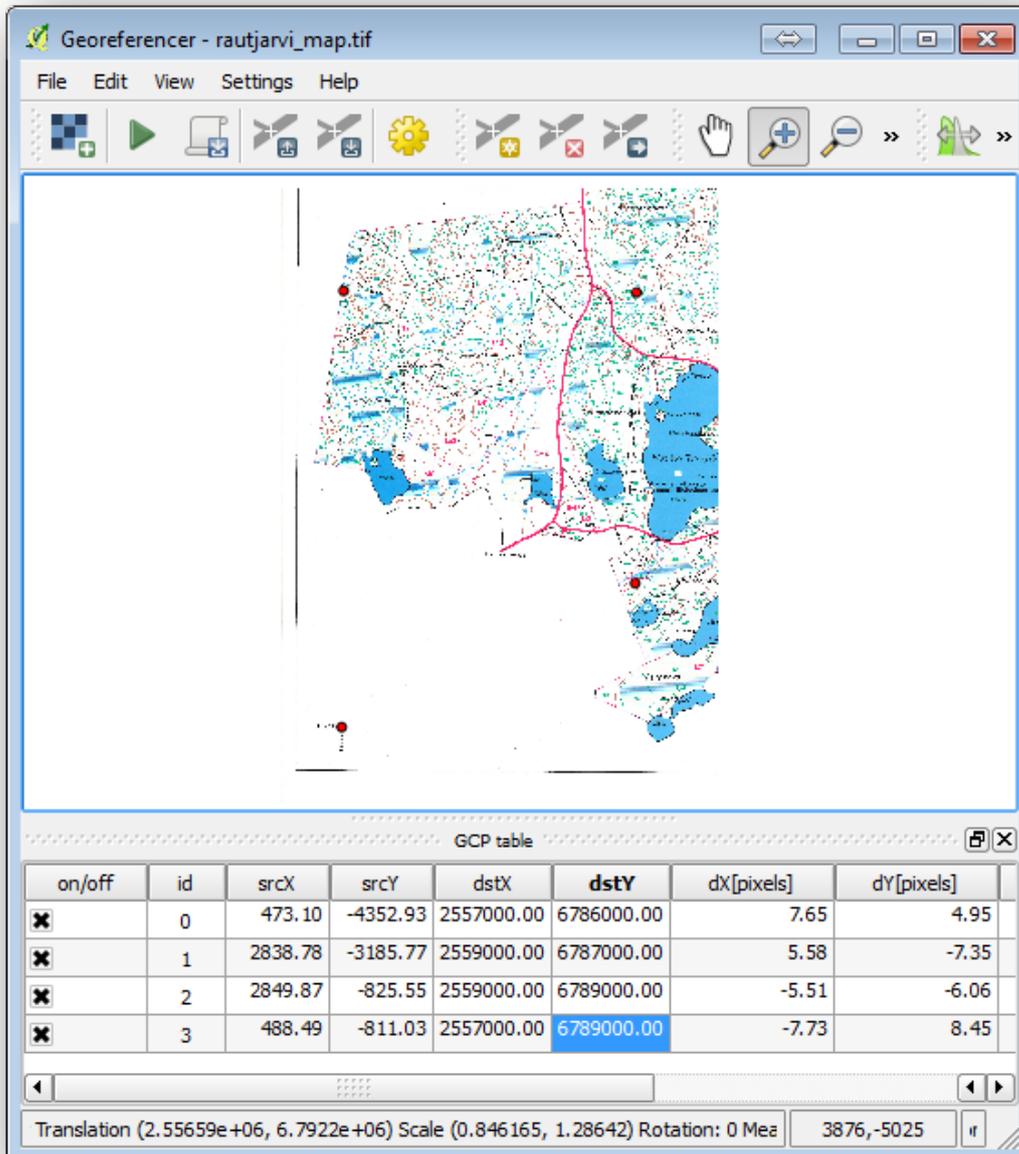
- 地図の左下隅にズームインし、座標対、x と y を持つ十字があることに注意してください。座標は前述のように `KKJ / フィンランドゾーン 2 CRS` です。このポイントを地図を地理参照するための第 1 の地上基準点として使用します。
- `point` ツールを追加し、(必要に応じてパンとズーム) 十字線の交点をクリックし選択します。
- 地図座標を入力 ダイアログで地図に表示される座標 (X : 2557000 と Y : 6786000) を書き込みます。
- OK をクリックします。

地理参照のための最初の座標の準備ができました。

黒い線の画像で他の十字線を探すと、北と東の両方の方向で互いに 1000 メートル離れています。最初のポ

イントとの関係でそれらのポイントの座標を計算できるはずですが、

画像にズームアウトして、他の十字を見つけるまで右に移動し、何キロメートル移動してきたか見積もります。地上基準点はできるだけ互いに離して取得してください。1つめをやったのと同じ方法で、少なくとも3つの地上基準点をデジタル化します。終わりにはこのようになっている必要があります：



3つの地上基準点がすでにデジタル化されていると、地理参照の誤差をポイントから出てくる赤い線として見るすることができます。GCPテーブルの中でもdx[ピクセル]とdy[ピクセル]列でピクセル単位の誤差を見ることができます。ピクセル単位の誤差が10ピクセルよりも大きくあるべきではありません。そうである場合は、問題がないかどうか、デジタル化された点および入力した座標を見直してください。ガイドとして上記の画像を使用できます。

満足いく基準点ができたら、後で必要になった場合のために地上基準点を保存し：

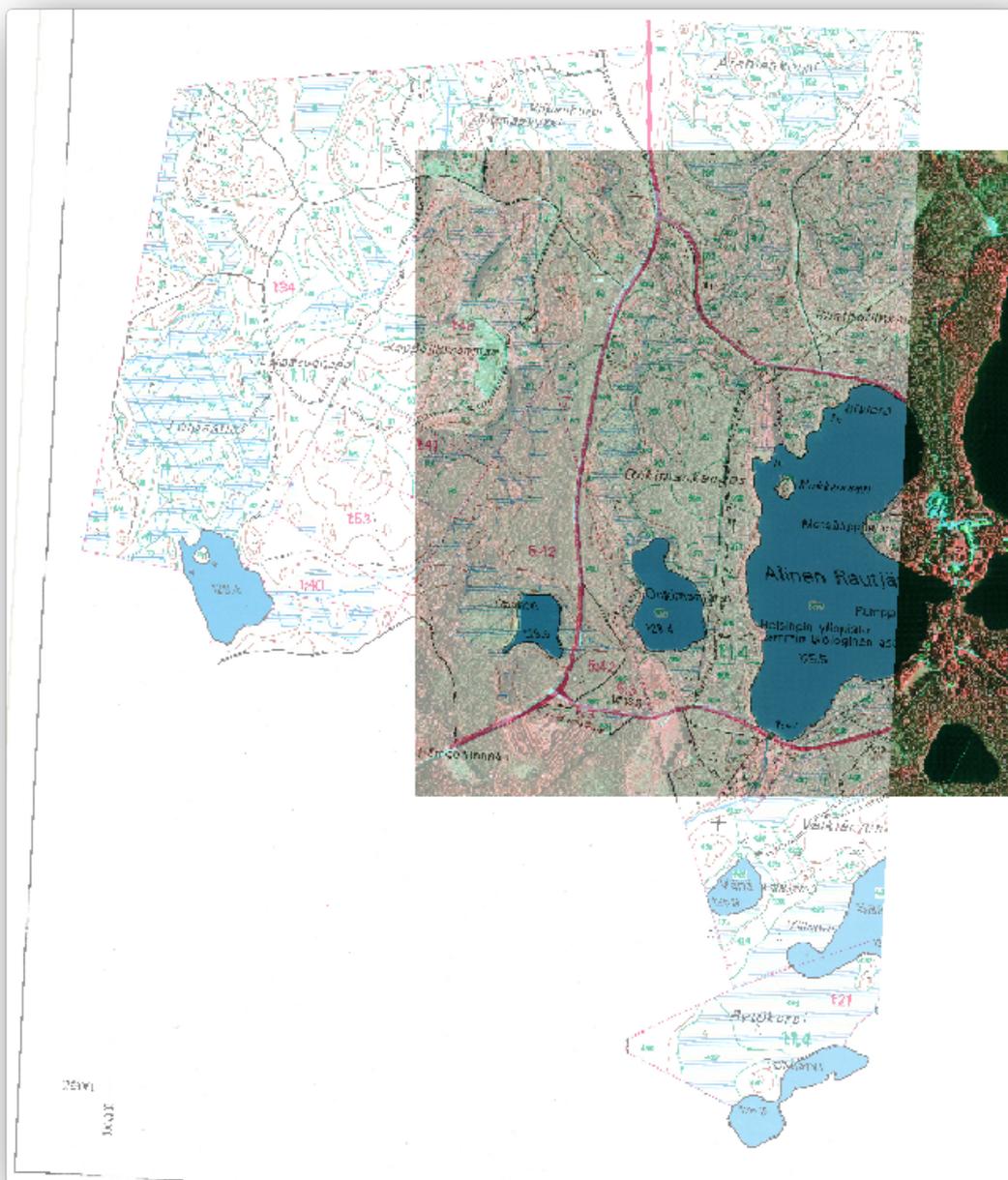
- ファイル->名前をつけてGCP点を保存...。
- フォルダ exercise_data\forestry\digitizing 中で、ファイルに rautjarvi_map.tif.points という名前を付けます。

最後に、地図を地理参照します：

- ファイル → 地理参照を開始。
- 地理参照の設定を編集したときに既にファイルを rautjarvi_georef.tif と名前つけていることに注意。

今、地理参照ラスターとして QGIS プロジェクトの地図を見ることができます。ラスターがわずかに回転しているように見えますが、それはデータが KKJ/フィンランドゾーン 2 でありプロジェクトが ETRS89 / ETRS-TM35FIN であるというだけの理由であることに注意してください。

exercise_data\forestry フォルダの、rautjarvi_aerial.tif という名前の空撮画像を開き、データが適切に地理参照されていることを確認できます。地図とこの画像は非常によく一致している必要があります。地図の透明度を 50 % に設定して空撮画像と比較します。



QGIS プロジェクトへの変更を保存し、次のレッスンはこの時点から続きます。

14.2.3 In Conclusion

ご覧になったように、紙の地図を地理参照することは、比較的簡単な操作です。

14.2.4 What's Next?

次のレッスンでは、地図内の森林をポリゴンとしてデジタル化し、それらに調査データを追加します

14.3 Lesson: 林分をデジタル化する

地理参照された地図を単純な背景画像として使用しようとしている場合を除き、次の自然なステップは、それからの要素をデジタル化することです。そのようなことは、既に `../ create_vector_data / create_new_vector` でベクトルデータを作成する演習の中で、学校のフィールドをデジタル化したときに行ってきました。このレッスンでは緑の線として地図に表示される林分の境界をデジタル化しようとしていますが、空撮画像を使用する代わりに地理参照された地図を使用します。

このレッスンの目標：林分をデジタル化し、最終的にそれらに目録データを追加して、デジタル化作業を支援するための技術を学びます。

14.3.1 Follow Along: 林分境界を抽出する

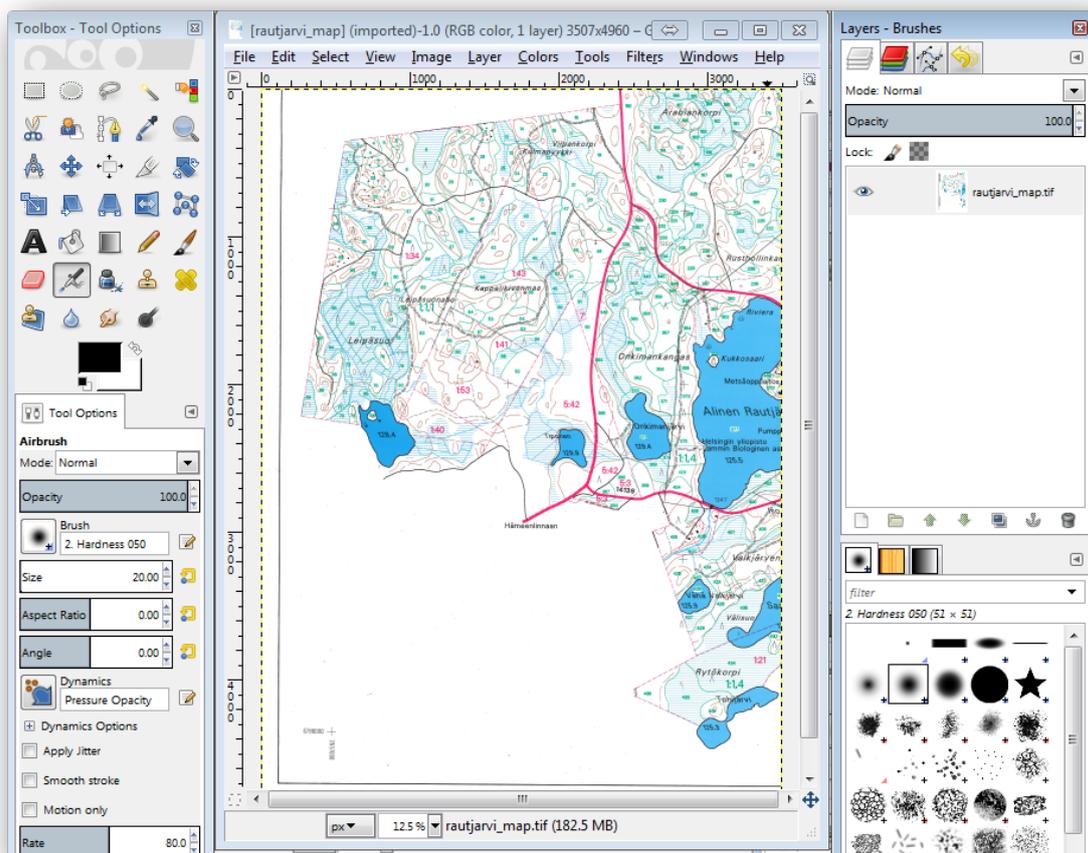
前のレッスンで保存された、`map_digitizing.qgs` プロジェクトを QGIS で開きます。

地図をスキャンして地理参照してしまえば、画像をガイドとして見ることで直接デジタル化を開始できるでしょう。デジタル化に使用しようとしている画像が、例えば航空写真である場合には、それが最も正しい方向である可能性が高いでしょう。

デジタル化するために使用しているものが良い地図であれば、この場合もそうですが、情報が各要素の種類ごとに異なる色の線で明確に表示されているものです。これらの色は、_ ‘GIMP <<http://www.gimp.org/>> のような画像処理ソフトウェアを使用すれば、個々の画像として比較的容易に抽出できます。下で見るように、このような独立した画像は、デジタル化を支援するために使用できます。プロセッシング

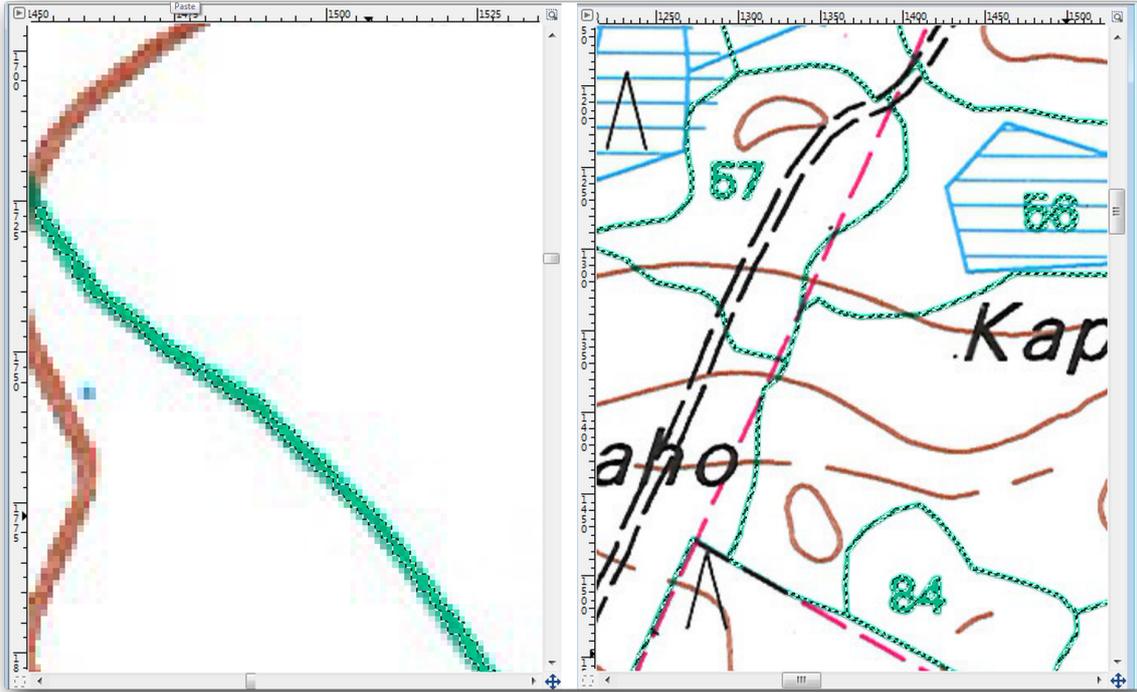
最初のステップは、GIMP を使用して、林分、つまりオリジナルのスキャン地図で見ることができた緑がかった線すべて、だけを含む画像を得ることです。

- GIMP を開きます (それが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、先生に尋ねます)
- `exercise_data / forestry` フォルダの中で、オリジナルの地図画像 `rautjarvi_map.tif` を開きます ファイル -> 開く。林分は緑色の線として (各ポリゴン内に緑で 林分の数も) 表されていることに留意されたいです。



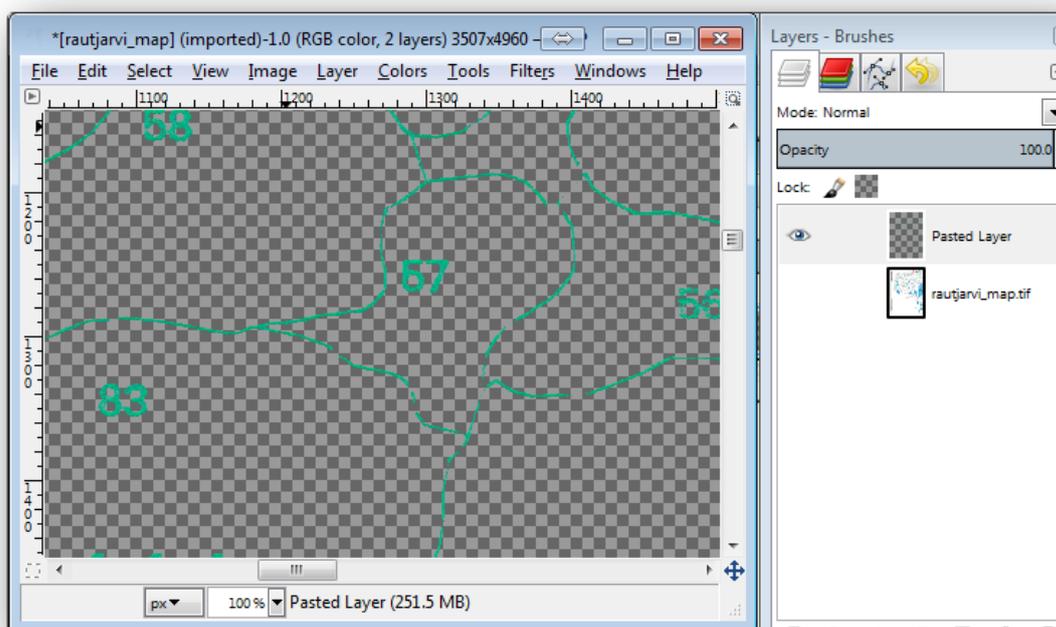
今は、林分の境界（緑がかったピクセル）を構成するされている画像内のピクセルを選択できます。

- ツール 選択 -> 色により を開きます。
- アクティブツールで、林分の線が線を構成する画素を区別するのに十分なくらい近くなるように、画像にズームインします (*Ctrl* キー+マウス *wheel*)。下の左の画像を参照してください。
- ツールは、いくつかのピクセルの色の値を収集するように行の中央にマウスカーソルをクリックしてドラッグ。
- マウスクリックを解除し、数秒待つ。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。
- 緑がかったピクセルが画像全体で選択されているかを確認するためにズームアウト。
- 結果に満足できない場合は、クリックやドラッグ操作を繰り返します。
- ピクセルの選択は、以下の右の画像のようになるはずでず。



選択を完了したら、新しいレイヤー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。

- 選択された画素をコピー (*Ctrl+C*) します。
- そして、ピクセルを直接貼り付けます (*Ctrl+V*)、GIMP には貼り付けられたピクセルが新しい一時的なレイヤーとして、レイヤー-ブラシ パネル中の 浮動選択 (貼り付け レイヤー) として表示されます。
- その一時 レイヤー を右クリックして 新しいレイヤー へ を選択します。
- 貼り付け レイヤー のみが表示されるよう、元の画像 レイヤー の隣の「目」のアイコンをクリックしてそれを非表示にしてください。



- 最後に、ファイル->エクスポート... を選択し、ファイルの種類を選択 (拡張子で) を *TIFF* 画像として設定し、digitizing フォルダを選択してそれに rautjarvi_map_green.tif と名前を付けます。尋ねられたとき非圧縮を選択します。

道路や地形等高線を表す茶色のものを表す黒線を抽出し、例えば、画像内の他の要素と同様の処理を行うことができます。しかし、私たちのために林分で十分です。

14.3.2 Try Yourself 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、データの残りの部分とそれを使用できるようにするには、この新しいイメージをジオリファレンスする必要があります。

この画像は Georeferencer ツールに関する限りは基本的に元の地図画像と同じ画像なので、これ以上地上基準点をデジタル化する必要はないことに注意してください。ここでは、忘れてはならないいくつかのものがあります：

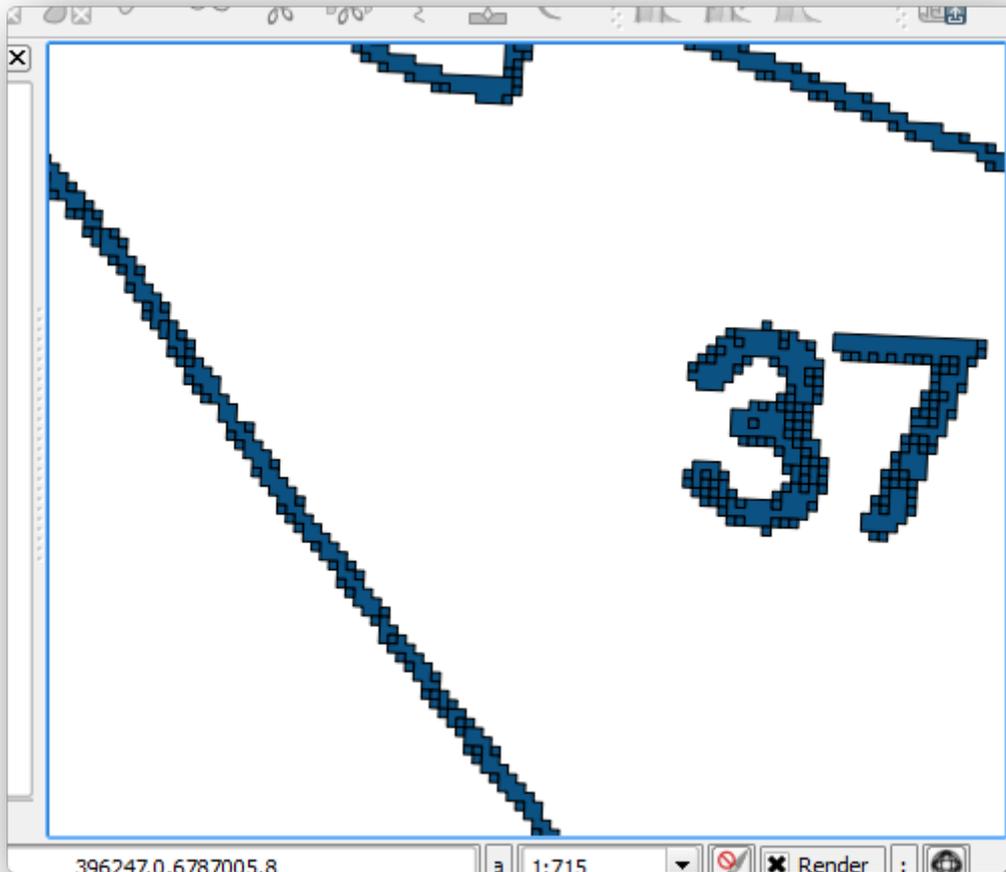
- この画像も、もちろん、KKJ/フィンランドゾーン 2 CRS です。
- 保存された地上基準点を使用する必要があります、ファイル->GCP 点をロード。
- 変換設定を確認することを忘れないでください。
- 出力ラスタに digitizing フォルダ中で rautjarvi_green_georef.tif と名前を付けます。

新しいラスタは、元の地図とうまく合っていることを確認してください。

14.3.3 Follow Along: デジタル化のための支援ポイントを作成します

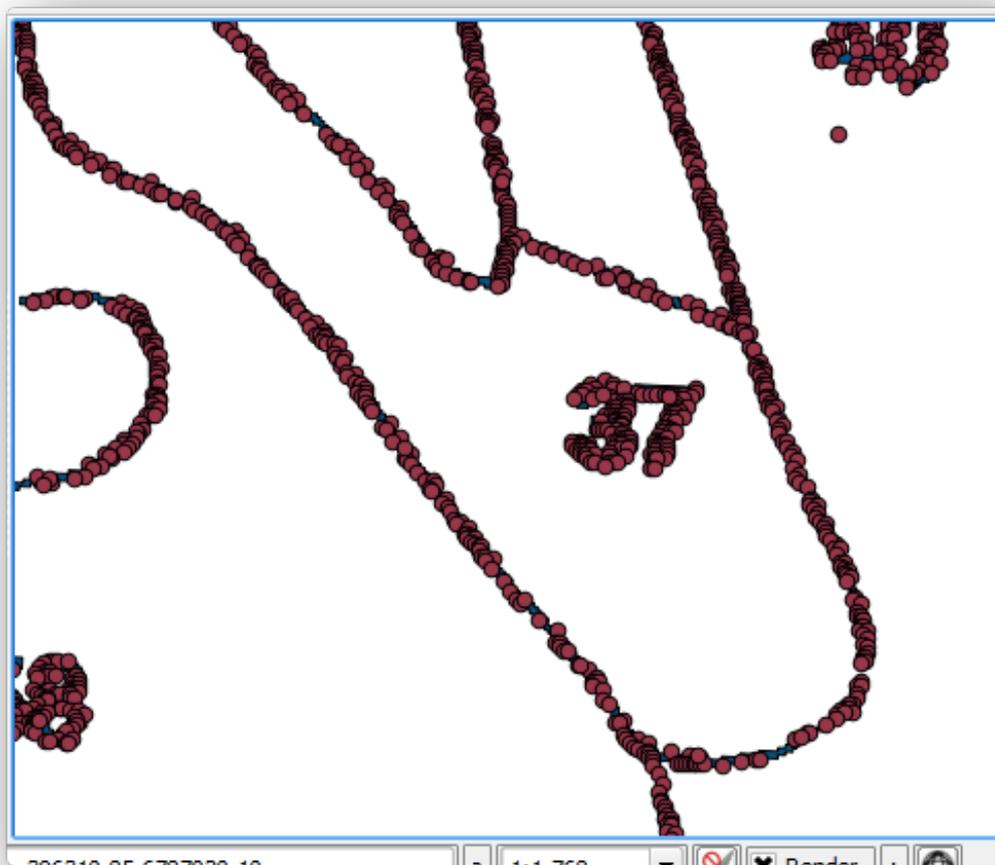
QGIS でデジタル化するツールを心の中に持って、それをデジタル化の間にそれらの緑のピクセルにスナップすると便利だろうとすでに考えているかもしれませんが。それはまさに次にしようとしていること、QGIS で利用可能なスナップツールを使用して、デジタル化するときに、林分の境界をたどる支援するために、後でそれらを使用するためにこれらのピクセルからポイントを作成です。

- ラスタ ->変換 -> *Polygonize*(ベクタ から ラスタ) 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。どのようにか覚えていない場合、../ complete_analysis / raster_to_vector で確認できます。
 - digitizing フォルダ内に rautjarvi_green_polygon.shp と名前を付けて保存します。
- ズームインとポリゴンがどのように見えるかを参照してください。このような何かが得られるでしょう：



これらのポリゴンのうち、ポイントを獲得するために、次のいずれかのオプションは、自分の重心を得ることです：

- ベクトル ->ジオメトリツール ->ポリゴン重心 を開きます。
- ツールの入力ファイルとしてたった今得たポリゴン レイヤ を設定します。
- digitizing フォルダ内で 出力 に 内部 green_centroids.shp と名前を付けます。
- 結果を キャンバス に追加 をチェックします。
- ポリゴンの重心を計算するためのツールを実行します。



今、TOC から *rautjarvi_green_polygon* レイヤ を削除できます。

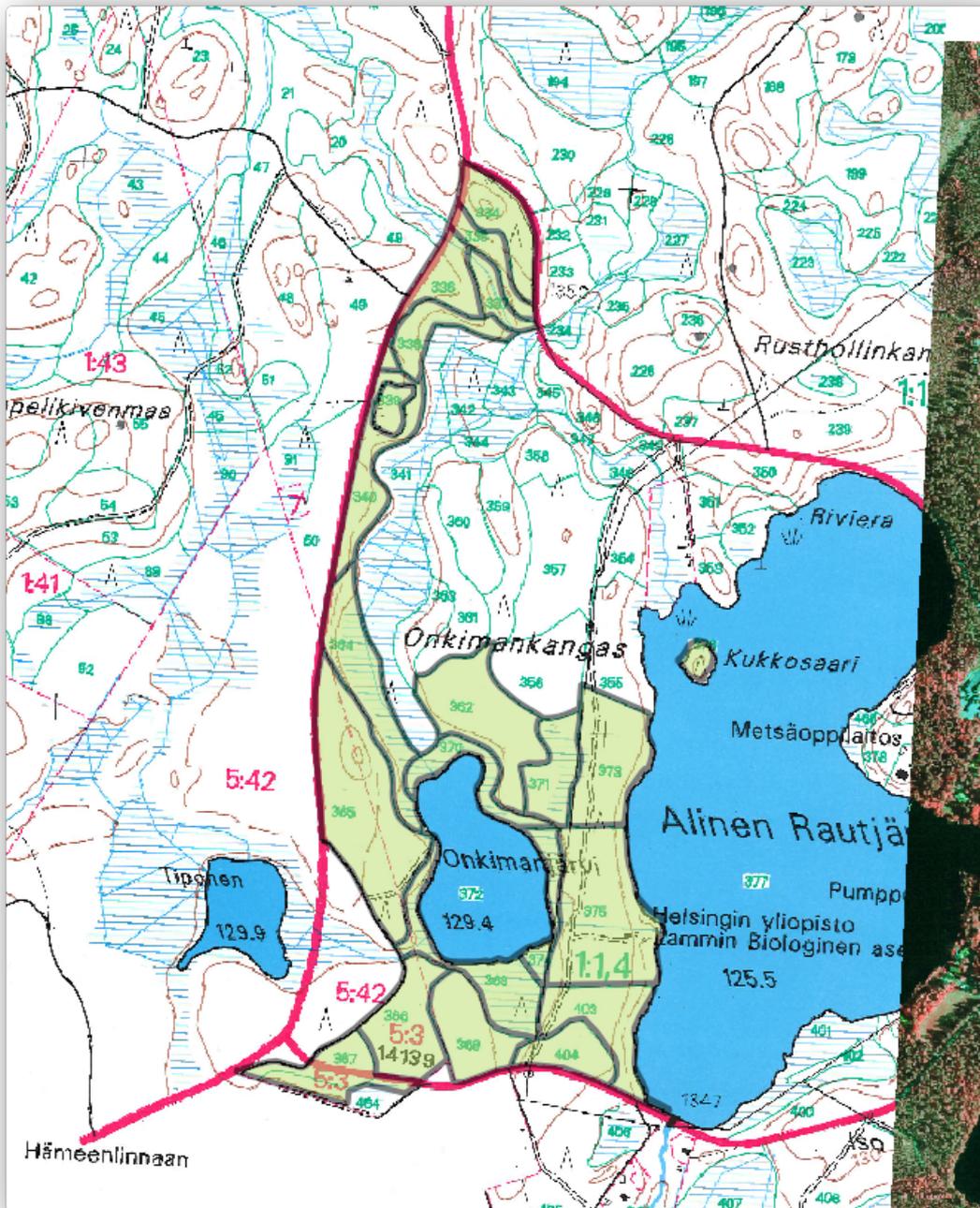
重心レイヤーのシンボルをこのように変更します：

- *green_centroids* のために レイヤプロパティ を開きます。
- *Style* タブに移動します。
- *Unit* を地図単位に設定してください。
- *Size* を 1 に設定してください。

お互いにポイントを区別する必要はありません、それらがスナップツールがそれらを使用するそこにする必要があるだけ。これらのポイントを使用することで、それ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 Follow Along: 林分 をデジタル化

今、実際のデジタル化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、この演習のために、すでにデジタル化された関心領域の一部で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 の半分のデジタル化を完了するだけでしょう：



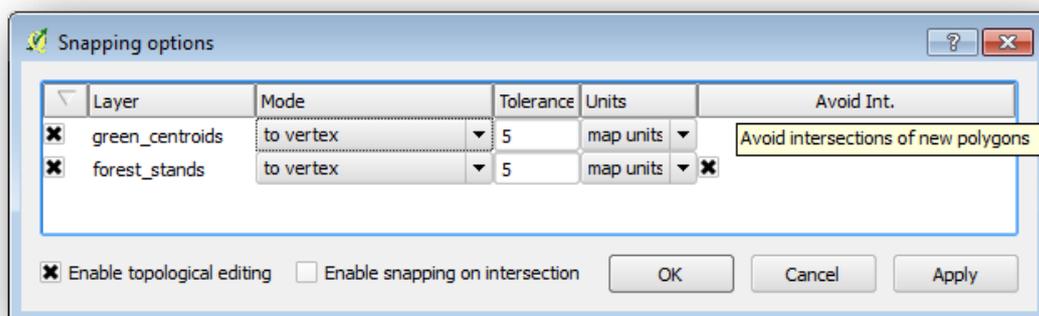
- ファイルマネージャのブラウザを使用して digitizing フォルダに行きます。
- 地図に forest_stands.shp ベクタ ファイルをドラッグ&ドロップします。

どのポリゴンがすでにデジタル化されているかが容易に見えるように、新しいレイヤのシンボルを変更します。

- ポリゴンの塗りを緑に。
- ポリゴン境界を 1mm に。
- そして透明度を 50 % に設定します。

過去のモジュールを覚えていれば今、私たちは、セットアップおよびスナップオプションを有効にする必要があります：

- 設定 -> スナップオプション... に行きます。
- green_centroids と forest_stands レイヤ でスナップを有効にします。
- それらのレイヤーの 許容誤差 を 5 地図単位に設定してください。
- forest_stands レイヤ の 交差を回避 ボックスを チェックします。
- トポロジー編集を有効 をチェックします。
- Apply をクリックします。



デジタル化と重心 レイヤ またはデジタル化されたポリゴンの任意の 頂点 でのポイントの一つに十分近い取得されるたびにこれらのスナップ設定では、ピンクの十字がスナップされる点に表示されます。

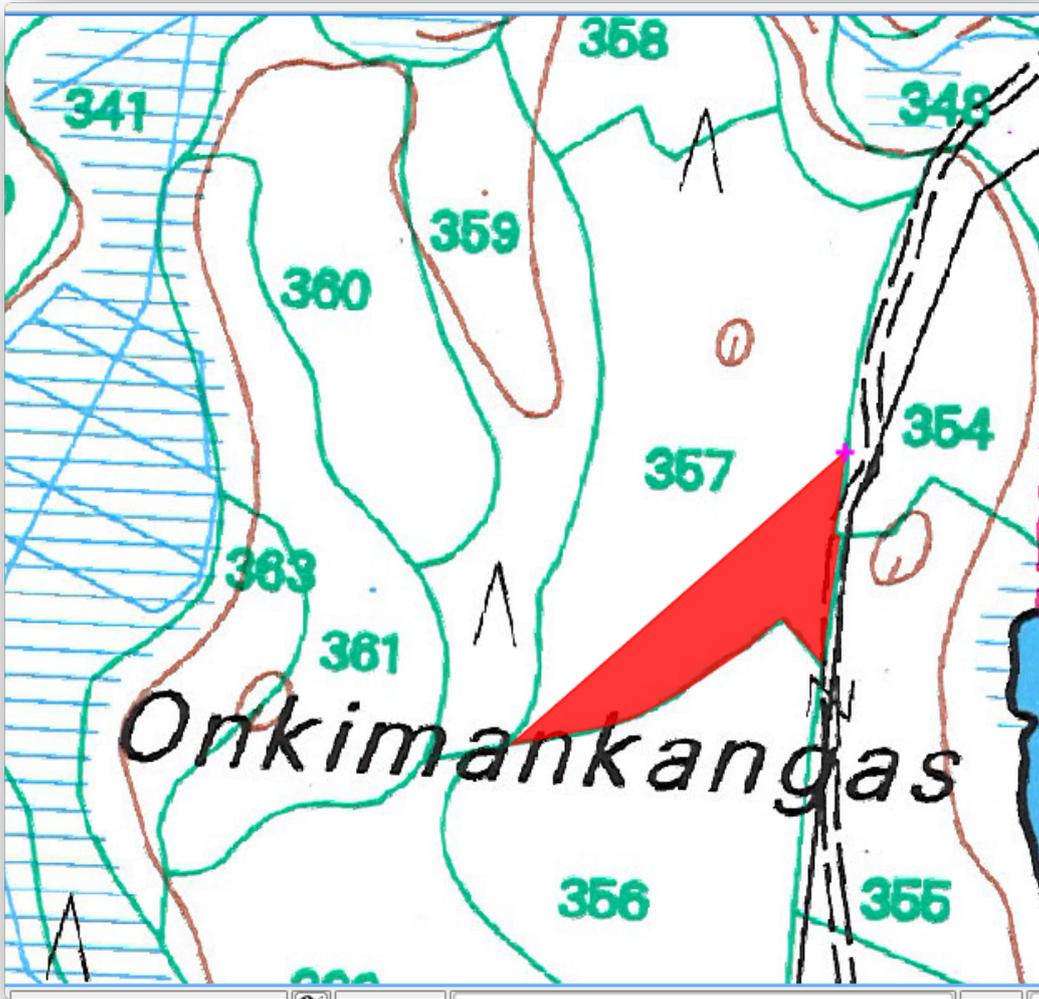
最後に、forest_stands`と `:kbd:`rautjarvi_georef`を除くすべてのレイヤーの可視性をオフにします。地図画像がこれ以上の透明されていないことを確認してください。

デジタル化を開始する前に注意すべき重要な物事のカップル：

- 国境のデジタル化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、ノード 2 つだけでデジタル化してください。一般に、できるだけ少ないノードを使用してデジタル化してください。
- ポリゴンは、例えばいくつかのコーナーで、正確であることが必要と感じた場合、または特定のノードで他のポリゴンと接続したいときのみズームインして範囲を閉じてください。
- デジタル化しながら、マウスの中ボタンを使用してズームイン/アウトおよびパンしてください。
- 一度に 1 つのポリゴンをデジタル化してください。
- 1 つのポリゴンをデジタル化した後、地図から見える林分 ID を書き込みます。

今、デジタル化を開始できます：

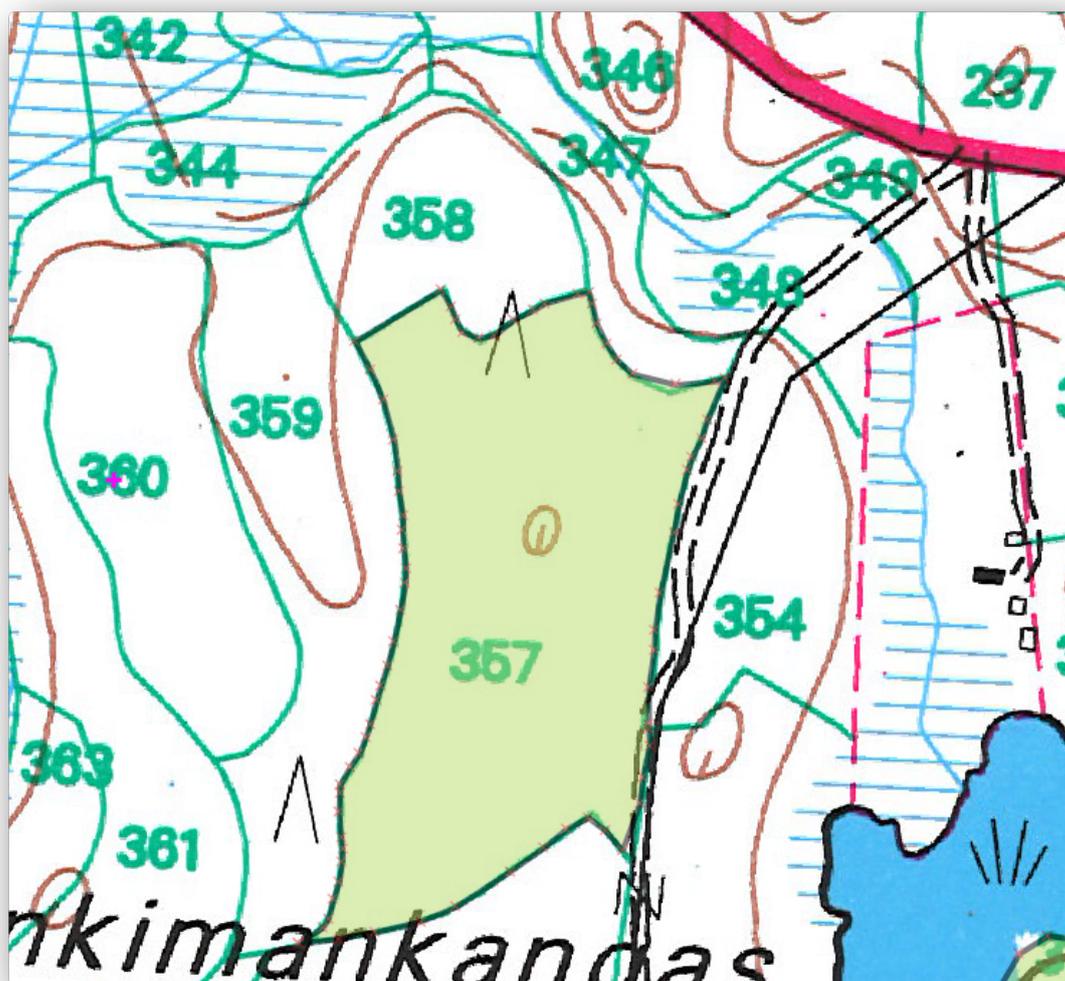
- マップウィンドウで林分番号 357 の位置を確認します。
- forest_stands.shp レイヤについて編集を有効にします。
- :guilabel:`地物を追加`ツールを選択します。
- ドットの一部を接続することにより、林分 357 をデジタル化開始します。
- スナップを示すピンク十字に留意されたいです。



- 完了したら、右のそのポリゴンをデジタイズ終了します。
- 林分 id を入力します (この場合は 357)。
- Click *OK*.

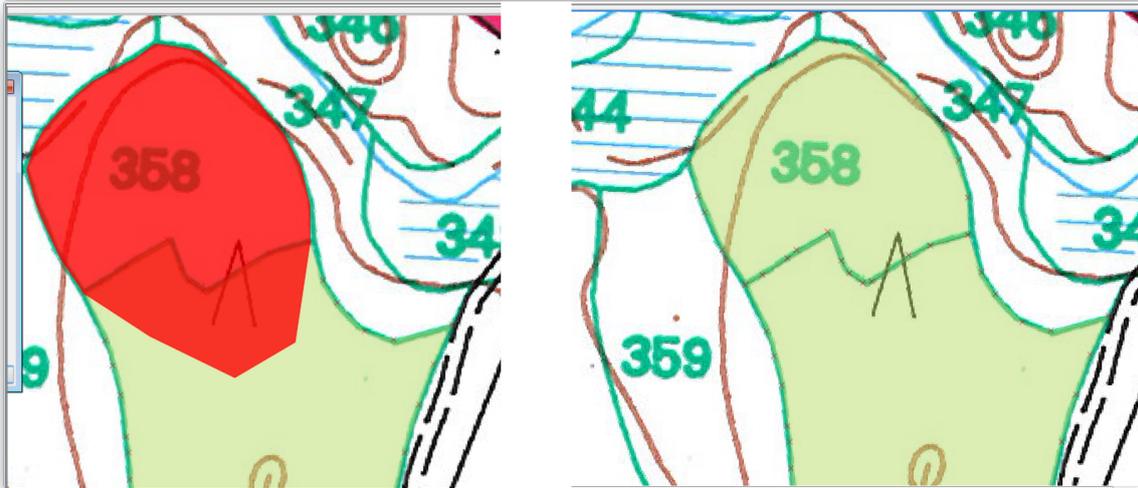
デジタル化し終えてもポリゴン id の入力を求められなかった場合は、設定->オプション-> *Digitizing* に行き、地物作成後まで 属性 フォームのポップアップを抑制 がチェックされていないことを確認してください。

デジタル化されたポリゴンは次のようになります。



今、第二のポリゴンのために、林分番号 358 を拾います。forest_stands レイヤ 交差を回避 がチェックされていることを確認してください。このオプションでは、既存のポリゴンを超えるデジタル化した場合、新しいポリゴンが既存のポリゴンの境界を満たすようにトリミングされるように、デジタル化でポリゴンを交差することはできません。この特性を使用すれば共通の境界が自動的に得られます。

- 林分 357 と共通する角の一つで林分 358 をデジタル化し始めます。
- 両方の林分のための他の共通の角に到達するまで、次に正常に続きます。
- 最後に、共通の境界が交差されていないことを確認したポリゴン 357 内のいくつかのポイントをデジタル化。下の左の画像を参照してください。
- 右の林 358 の編集を終了します
- id を 358 と入力します。
- OK をクリック、右の画像で見ることができるよう、新しいポリゴンは林分 357 と共通の境界を示すべきです。



既存のポリゴンを重ねたポリゴンの一部は自動的にトリミングされていて、それがあつたことを意図したように、共通の境界が残されています。

14.3.5 Try Yourself 林分をデジタル化完了

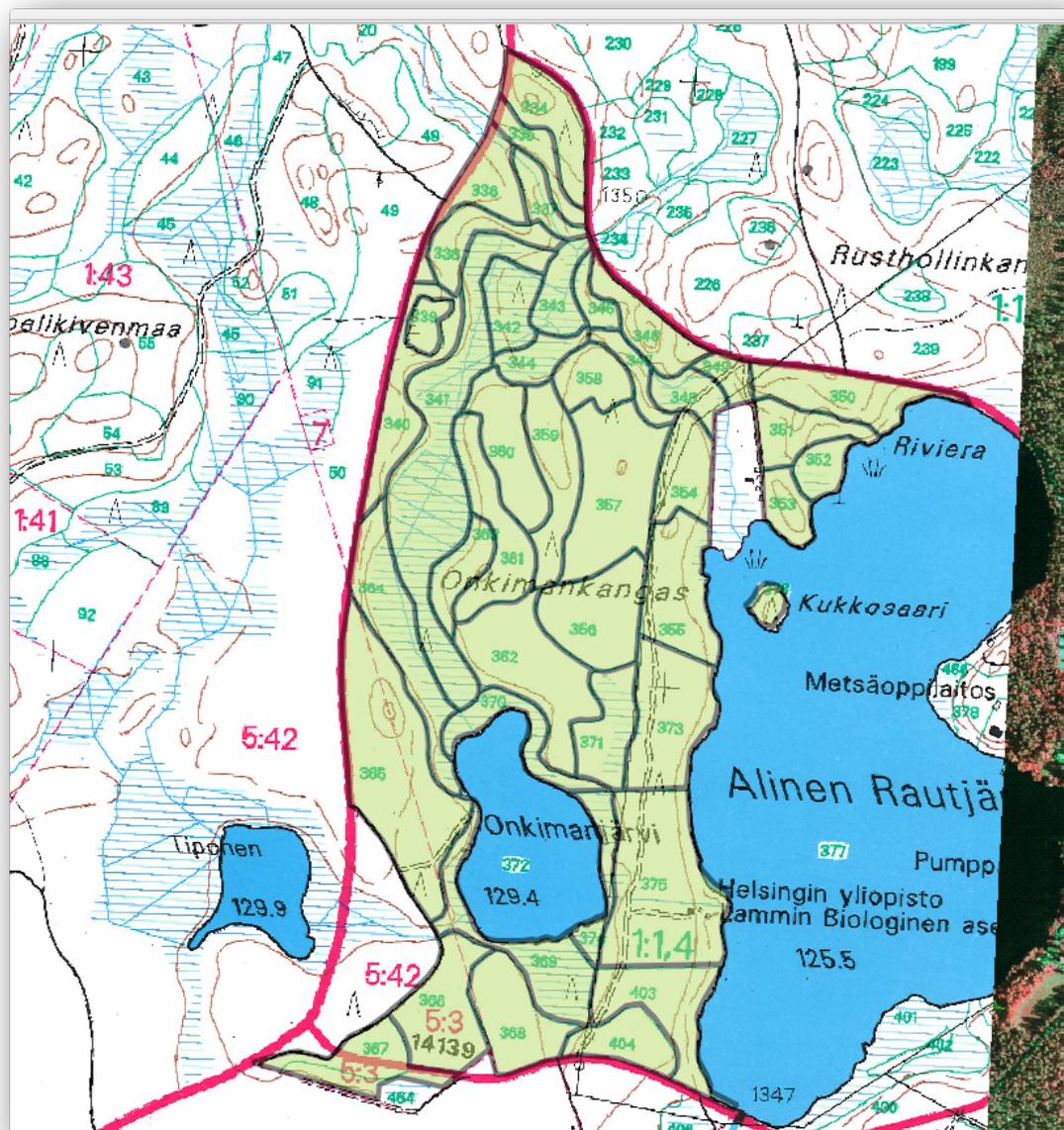
今、2つ林分の準備ができています。そして、続行する方法について良いアイデア。あなたは主要道路と湖によって制限されているすべての林分をデジタル化するまで、自分でデジタル化を続行します。

それは多くの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ル化するのに慣れるでしょう。それは約15分かかります。

デジタル化中に、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。../ create_vector_data / topo_editing で必要なツールについて学びましたが、今それらについて読み直しに行くにはおそらく良い瞬間です。

トポロジカル編集を有効がアクティブになっていると、共通の境界は、両方のポリゴンの同じ時間に編集されているように、2つのポリゴンに共通のノードを移動できます、ことを覚えておいてください。

結果は次のようになります。

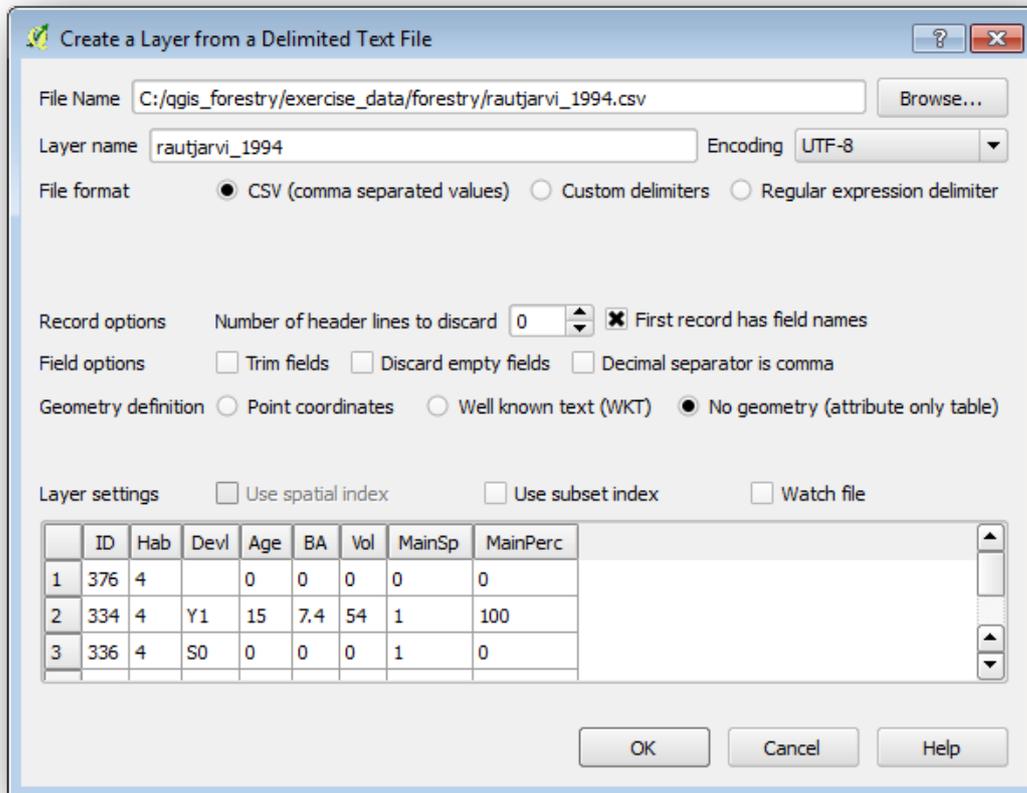


14.3.6 Follow Along: 林分データを結合

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。

テキストエディタで `exercise_data\forestry` ディレクトリ中の `rautjarvi_1994.csv` ファイルを開き、目録データファイルが林分の番号である ID と呼ばれる属性を持っていることに注意してください。これらの番号はポリゴンに入力した林分の ID と同じであり、ベクタファイルにテキストファイルからのデータをリンクするために使用できます。同じフォルダにある `rautjarvi_1994_legend.txt` ファイルでこの目録データのメタデータを見ることができます。

- QGIS で `.csv` レイヤ -> 区切りテキストレイヤーを追加... ツールを開きます。ダイアログで、それを次のように設定します。



.csv ファイルからデータを追加するには：

- forest_stands レイヤ のための レイヤプロパティ を開きます。
- 結合 タブに移動します
- ダイアログボックスの下部にあるプラス記号をクリックします。
- 結合 レイヤ として rautjarvi_1994.csv を、 結合 フィールドとして ID を選択します。
- Target フィールドも id に設定されていることを確認してください。
- OK を 2 回クリックします。

テキストファイルからのデータは、今ベクタファイルにリンクする必要があります。起こったことを確認するには、forest_stands レイヤの属性テーブルを開きます。目録データファイルからのすべての属性は、今お使いのデジタル化されたベクタレイヤにリンクされていることがわかります。

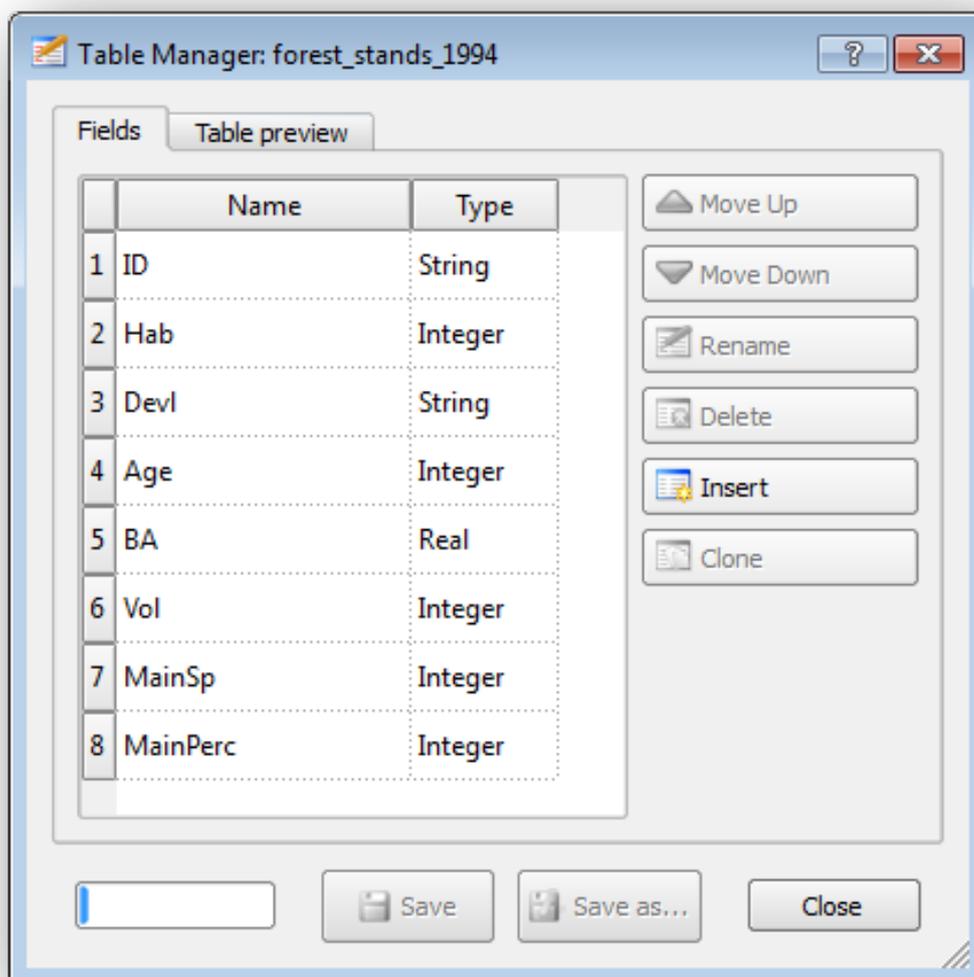
14.3.7 Try Yourself 属性 名の名前を変更してエリアと境界を追加する

.csv ファイルからのデータがちょうどベクタファイルにリンクされています。データが実際にベクタファイルに記録されるように、このリンクを恒久的なものにするには新しいベクタファイルとして forest_stands レイヤを保存する必要があります。属性テーブルを閉じ、forest_stands レイヤを右クリックして forest_stands_1994.shp としてそれを保存します。

まだ追加していなかった場合は、新しい forest_stands_1994.shp を地図で開きます。そして、属性テーブルを開きます。追加した列の名前はぜんぜん有用でないことに気づきます。これを解決するには：

- 前に他のプラグインに行ったようにテーブル マネージャー プラグインを追加します。

- プラグインが有効になっていることを確認します。
- TOC でレイヤー `forest_stands_1994.shp` を選択します。
- 次に、ベクタ->テーブル マネージャー->テーブル マネージャー に行きます。
- ダイアログボックスを使用し、`.csv` ファイル中のものと一致するように列の名前を編集します。



- `Save` をクリックします。
- `Yes` を選択してレイヤースタイルを維持します。
- テーブル マネージャ ダイアログを閉じます。

これらの 林分 に関連する情報を収集完了するには、林分の面積と周囲を計算することがあります。../complete_analysis / analysis_exercise 内のポリゴンのための領域を計算しました。する必要がある場合は、そのレッスンに戻り 林分の領域を計算し、新しい属性に面積と名前を付け、計算値はヘクターであることを確認してください。

今 :kbd: forest_stands_1994.shp レイヤは準備され、利用可能なすべての情報が満載です。

それに後で戻ってくる必要がある場合に、現在の地図の表示を維持するために、プロジェクトを保存します。

14.3.8 In Conclusion

これにはマウスを数回クリックする必要がありますが、今、デジタル形式と QGIS での使用のために準備中で古い目録データを持っています。

14.3.9 What's Next?

まっさらな新しいデータセットで異なる分析を開始できますが、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かもしれません。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 Lesson: 林分を更新する

古い目録の地図から情報をデジタル化し、林分に対応する情報を追加し終わりましたから、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す。

その森林地域からの航空写真に従って最初から新しい林分をデジタル化します。前のレッスンでデジタル化した森林地図は空中色赤外線 (CIR) の写真から作成されました。赤外光は、代わりに青色光の記録された画像のこのタイプは、広く植生領域を研究するために使用されます。また、このレッスンで CIR の写真を使用します。

林分をデジタル化した後、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：林分の新しいセットを CIR 航空写真からデジタル化し、他のデータセットからの情報を追加します。

14.4.1 古い林分を現在の航空写真と比較する

フィンランドの国土調査庁は、空撮画像などの地理データ、伝統的な地形図、DEM、LiDAR データ、など、各種のダウンロードを可能にするオープンデータポリシーを持っています。サービスは '[ここ<http://www.maanmittauslaitos.fi/en/file_download_service>](http://www.maanmittauslaitos.fi/en/file_download_service)' で英語でもアクセスできます。この演習で使用した空撮画像は、そのサービスからダウンロードした 2 枚のオルソ補正 CIR 画像 (M4134F_21062012 と M4143E_21062012) から作成されています。

- QGIS を開き、プロジェクト -> プロジェクトのプロパティ -> CRS でプロジェクトの CRS を ETRS89 / ETRS-TM35FIN に設定します。
- 「その場で」 CRS 変換を有効 がチェックされていることを確認してください。
- exercise_data\forestry\ フォルダから、デジタル化された湖を含む CIR の画像 rautjarvi_aerial.tif を追加します。
- 続いて QGIS プロジェクト digitizing_2012.qgs を保存します。

CIR の画像は 2012 年からです。ほぼ 20 年後の状況と 1994 年に作成された林分を比較できます。

- forest_stands_1994.shp レイヤーを追加します。
- ポリゴンを通して見るようにそのスタイルを設定します。
- 古い林分が視覚的に均質な森として解釈する可能性があるものをフォロー（またはしない）方法を確認してください。

エリアの周りにズームおよびパンします。おそらく、古い林分でおお静止画に対応するもの、しないものがわかるでしょう。

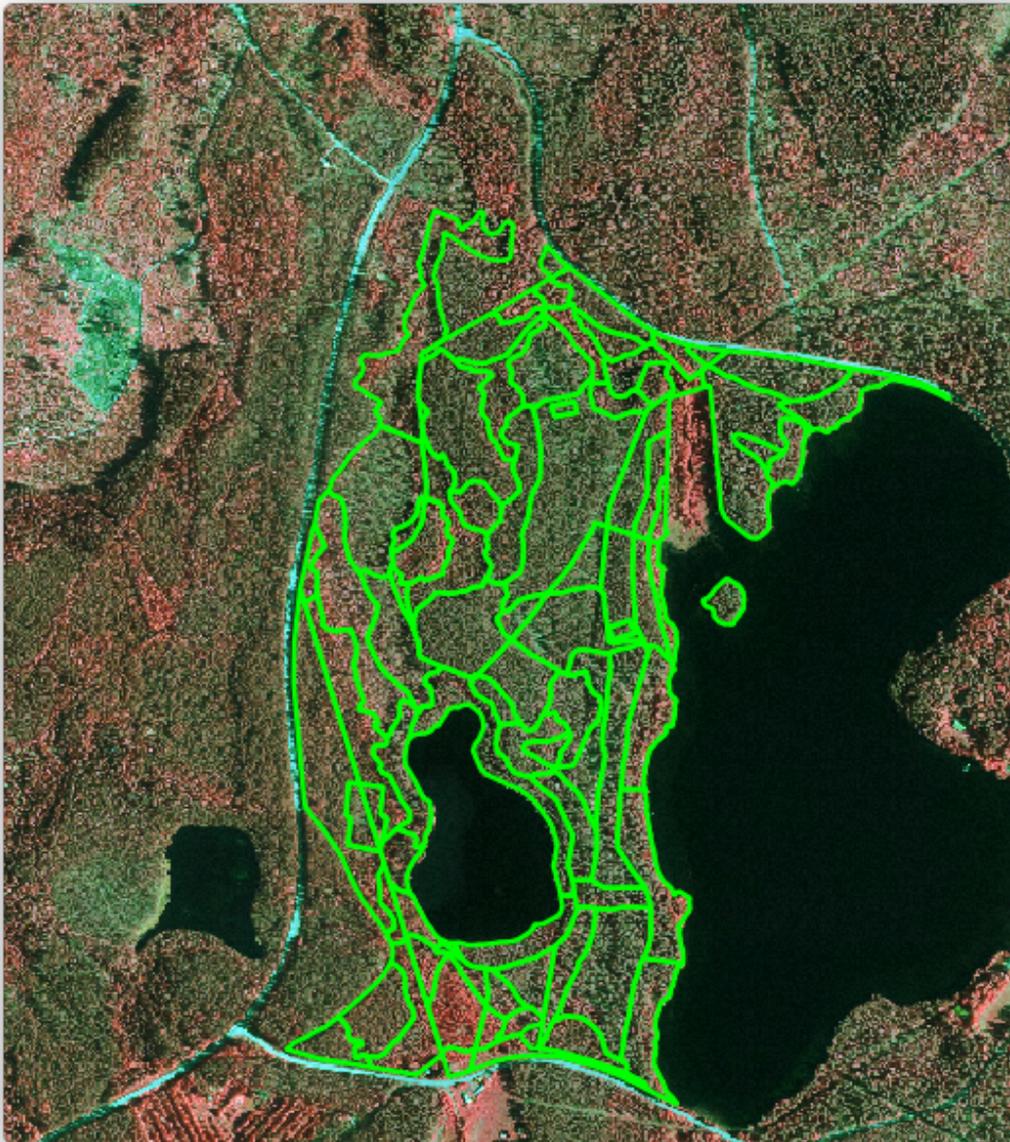
いくつかの 20 年が経過していると別の森林施業が (... 薄く、収穫を) 行われているように、これは、通常の状態です。林分はそれらをデジタル化したが、時間が経過しているとして、いくつかの森には、さまざまな方法で開発してきた人に 1992 年に均質な背中を見ていることも可能です。それとも単に森林インベントリーのための優先順位は、彼らが今日であることを異なっていました。

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 CIR 画像の解釈

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領域を、デジタル化してみましょう。前の練習でのようにすでに林分のほとんどが含まれているベクトルファイルで開始できますので、全領域をデジタル化する必要はありません。

- forest_stands_1994.shp レイヤーを外します。
- forest_stands_2012.shp レイヤーを追加します、exercise_data \林業\フォルダにあります。
- ポリゴンは何の塗りつぶしを持っていないし、境界が表示されるように、このレイヤのスタイリングを設定します。



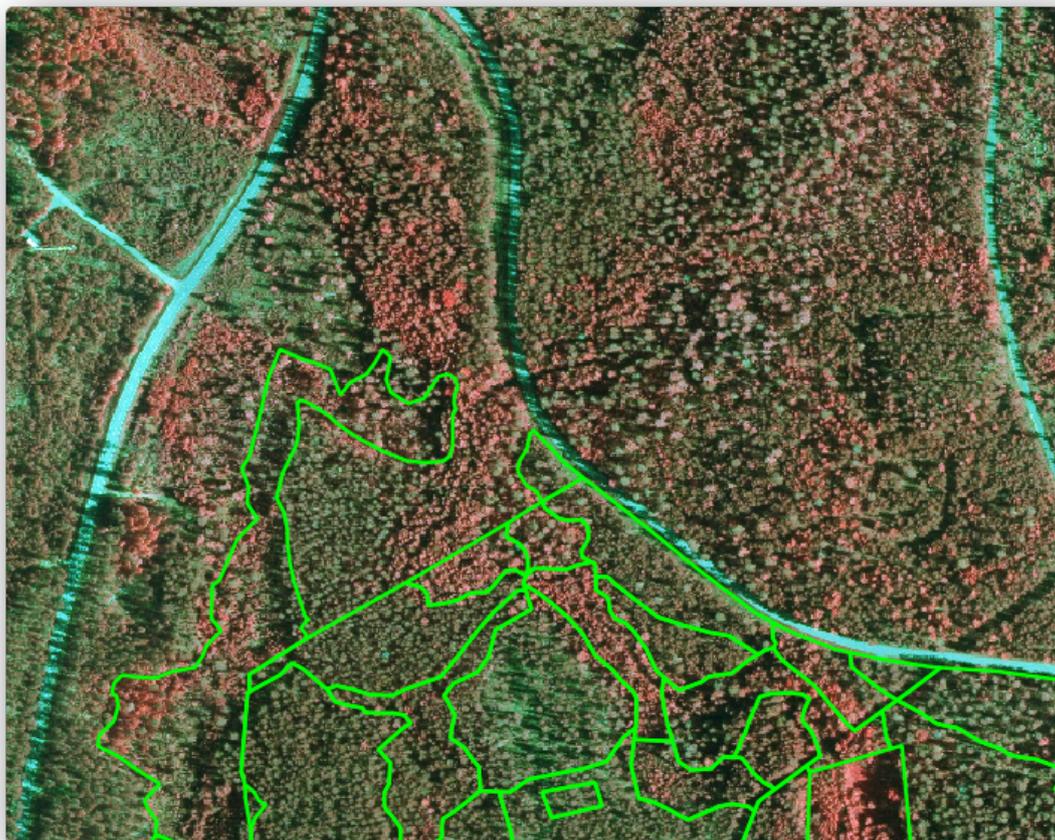
目録エリアの北に領域がまだ不足していることがわかります。不足している林分をデジタル化することは必要になります。

しかし、開始する前に、すでにデジタル化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費やしてください。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、いくらか林業の知識を持っている場合それが役立ちます。

画像から特定できるものについていくつかのアイデア：

- どの森林が落葉（フィンランドのほとんどは白樺林）樹林で（この地域では松やトウヒ）針葉樹林であるか。CIR 画像において、落葉樹林はしばしば暗緑色本、一方、針葉樹林は明るい赤色来ます。
- いつ林分の年齢が変化するか、樹冠の大きさを見ることで、画像内で識別できます。
- 別の林分の密度は、例えば、最近間引きが行われた林分は明確に樹冠の間にスペースを示すでしょうし、その周りの他の林分から区別することは容易なはずで。
- 青みを帯びた領域は不毛の地形、道路や市街地などの成長を開始していない作物を示しています

- 林分を特定しようとするとき、画像に近すぎるズームを使用しないでください。この画像には 1:3000 と 1:5000 の間の縮尺が十分であるはずですが、下の画像を参照（1:4000 縮尺）。

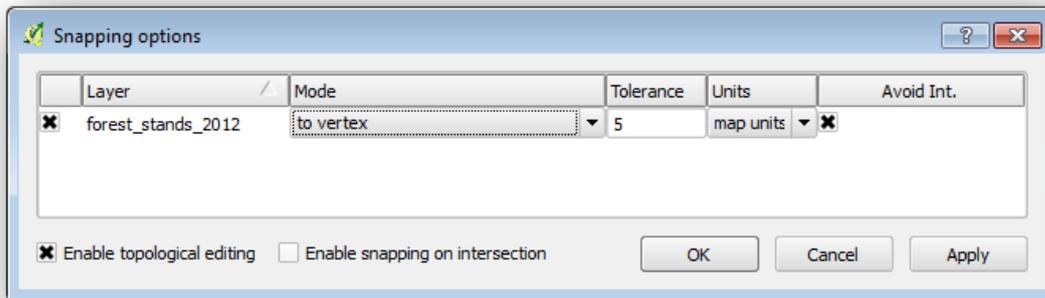


14.4.3 Try Yourself CIR 画像から林分をデジタル化

林分をデジタル化するときにはしかし、あまりにも詳細なことはいけない...、樹種、林齢の観点から、可能な限り均一で密度を林分を取得しようとするべきか、小さな数百人を作ることになります林分はそれがすべてでは有用ではない。もいずれか大きい（せいぜい3ヘクタール）ではないが小さすぎる（少なくとも0.5ヘクタール）林業の文脈において意味ではなく、ある林分を取得しようとする必要があります。

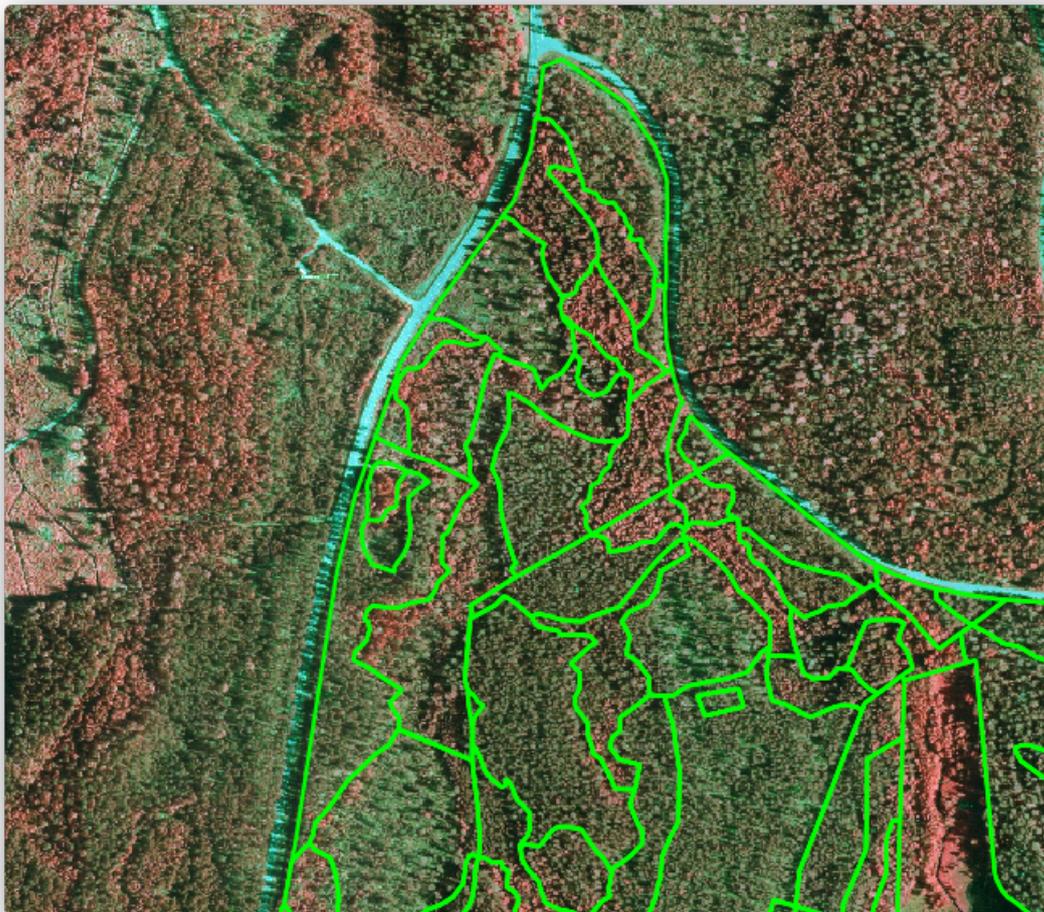
この示唆を心に持って、今、不足している林分をデジタル化できます。

- forest_stands_2012.shp の編集を有効にします。
- 画像のようにスナップとトポロジオプションを設定します。
- Apply が OK をクリックすることを忘れないでください。



前のレッスンで行ったように、スナップされる点レイヤーがないことが違うだけで、デジタル化を開始します。この地域のためには約 14 の新しい林分を取得する必要があります。デジタル化しながら、Stand_id フィールドに 901 から始まる番号を記入します。

作業が完了したら、レイヤーは次のようになります。



今、別の森林を定義するポリゴンの新しいセットは、CIR 画像から解釈できるように、現在の状況を表しています。しかし明らかにまだ森林目録データが欠落しています、よね？そのためにはまだ、森林を訪問し、各林分に対する森林属性を推定するために使用するいくつかのサンプルデータを取得する必要があります。次のレッスンでそれを行う方法について説明します。

現時点では、まだこの地域のために考慮されるべき保全規制について持っているいくつかの追加情報を使用して、ベクトルレイヤを改善できます。

14.4.4 Follow Along: 保全情報で林分を更新する

作業している地域のために、森林計画をしている間、以下の保全規制を考慮しなければならないことを研究されてきました。

- 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。
- 地域に流れに沿って成長している特別な関心の水辺林は保護されなければなりません。フィールドへの訪問では、流れの両側 20 メートル保護されなければならないことがわかりました。

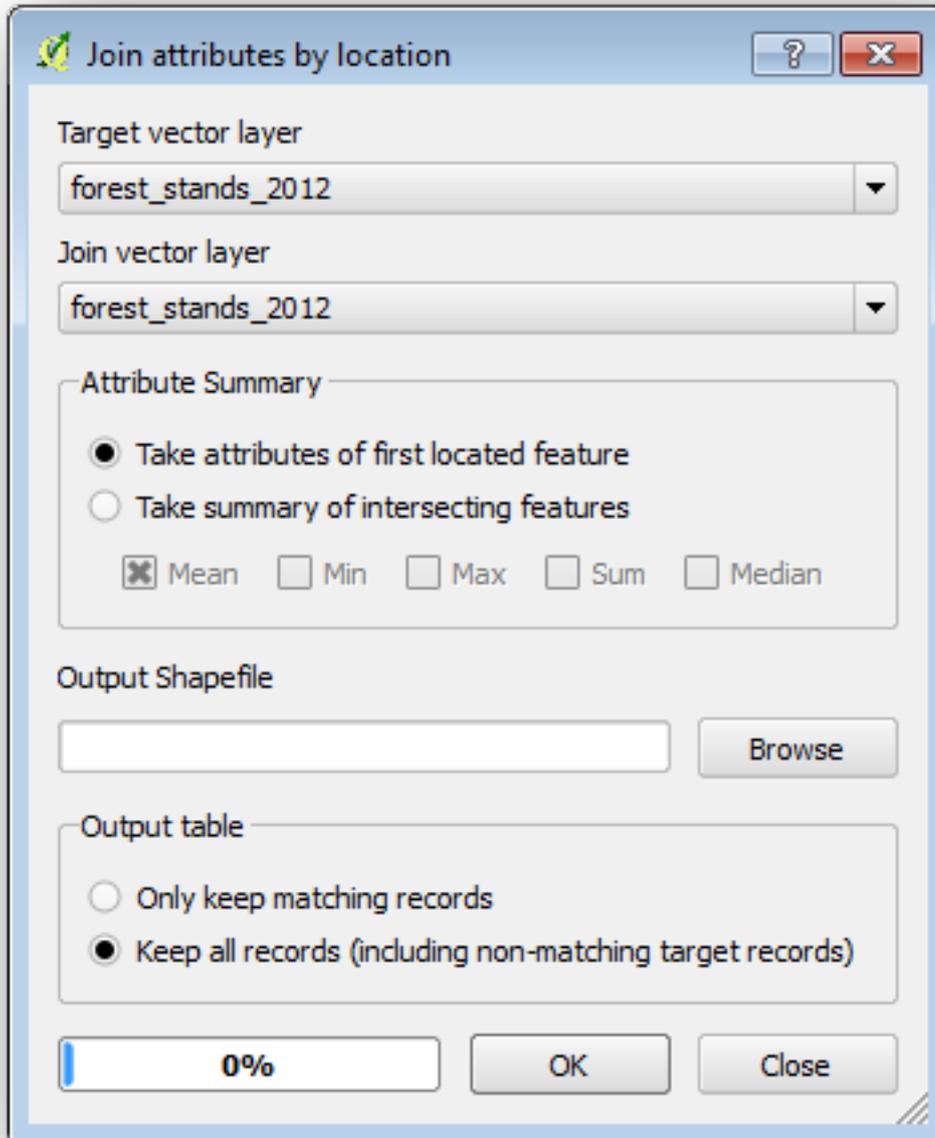
リスの場所や湖に向けて北エリアで実行されているデジタル化された流れを含む他の情報を含む 1 つのベクトルファイルを持っています。 `exercise_data\forestry\` フォルダから、ベクトルファイル `squirrel.shp` と `stream.shp` を追加します。

リスの場所の保護のために、保護する必要がある点の位置に関する情報が含まれている新しい林分に新しい属性 (列) を追加しようとしています。その情報は、森林施業が計画されるときはいつでも、後で利用できるようになり、野外チームは作業を開始する前に手つかずで残す必要がある領域をマークできるでしょう。

- リス レイヤのための属性テーブルを開きます。
- もしエゾモモンガのように定義されている 2 つの場所があることを確認することができ、及び領域を保護することは、場所から 15 メートルの距離で示されています。

林分にリスについての情報を結合するには、位置によって属性を結合を使用できます：

- ベクタ ->データ管理ツール ->位置によって属性を結合を開きます。
- ターゲットベクトル *layer* として `:kbd:' forest_stands_2012.shp'` レイヤを設定してください。
- 結合ベクトル *layer* として `squirrel.shp` ポイントレイヤを選択。
- 出力ファイルに `stands_squirrel.shp` と名前を付けます。
- 出力 *table* で `:guilabel:'すべてのレコードを保管 (非マッチングの対象レコードを含む)'` を選択します。だから、空間的にリスの場所に関連しているものだけを維持するのではなく、レイヤ中のすべての林分を保つ。
- *OK* をクリックします。
- TOC にレイヤを追加するか求められたとき *Yes* を選択します。
- ダイアログボックスを閉じます。

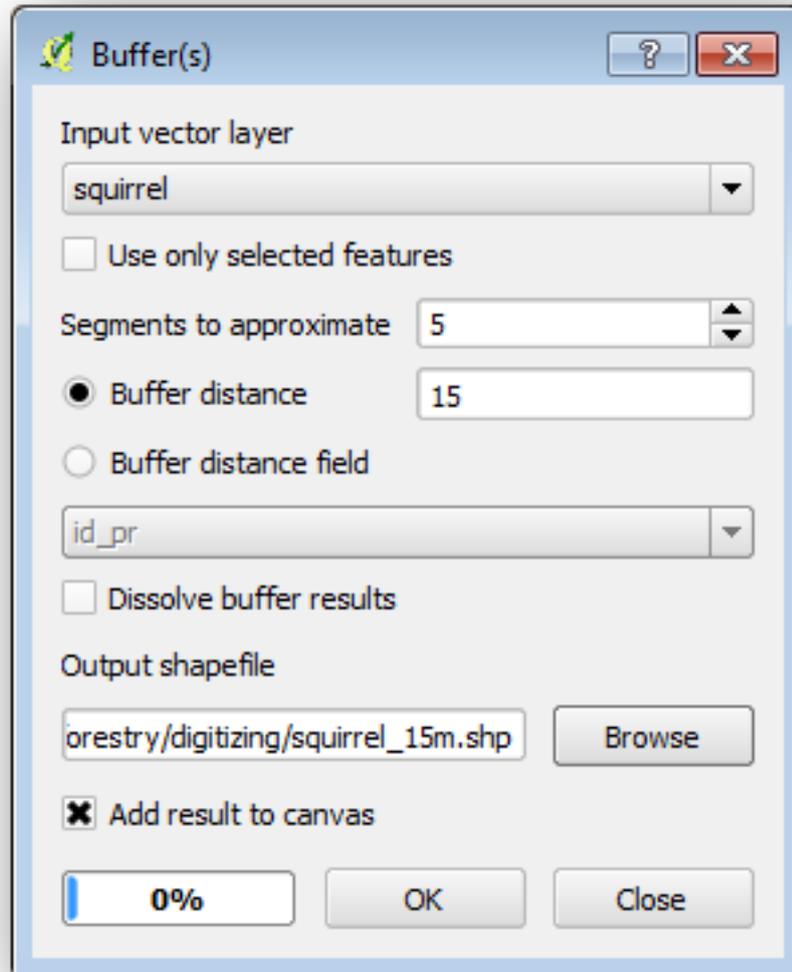


エゾモモンガに関連した保護情報に対応した新しい属性がある `stands_squirrel` 今、新しい林分レイヤ。新しいレイヤーのテーブルを開き、`Protection` 属性のための情報を持つ林分が一番上になるように整理します。リスが配置された場所 2 つの林分を今持っている必要があります。

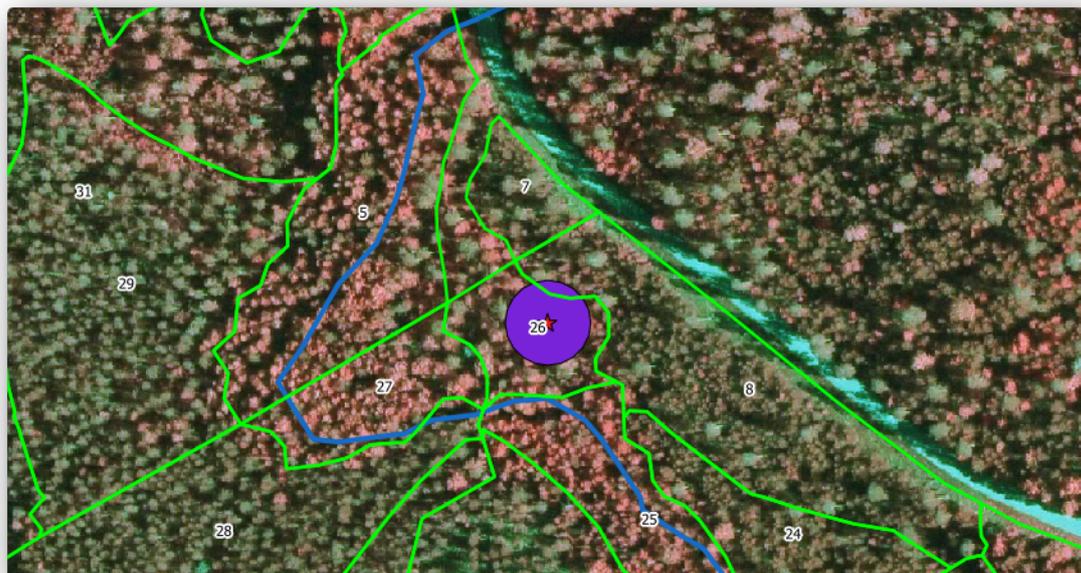
	Stand_id	id_pr	Protection	Distance
83	78	2	liito-orava	15
22	26	1	liito orava	15
0	1	NULL	NULL	NULL
1	33	NULL	NULL	NULL
2	32	NULL	NULL	NULL

この情報は十分かもしれませんが、リスに関連する領域が保護されるべきかを見てください。リスの場所を中心に 15 メートルのバッファを残す必要があることを知っています：

- ベクトル -> ジオプロセッシングツール -> *Buffer* を開きます。
- squirrel レイヤーのための 15 メートルのバッファを作成します。
- 結果に squirrel_15m.shp と名前を付けます。

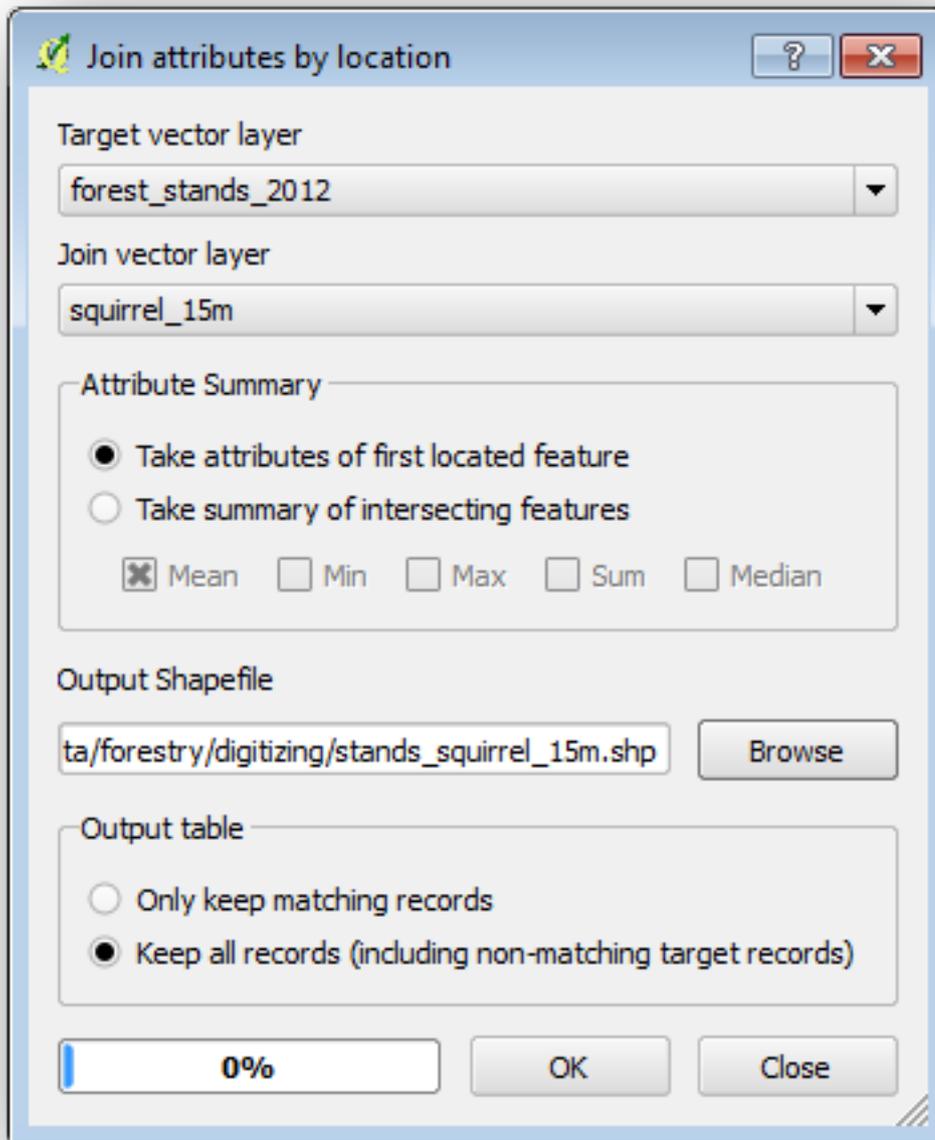


地域の北部の場所にズームインするとき、バッファ領域が同様に立つ近隣に延びていることがわかります。これは、森林施業は、そのスタンドに場所を取るだろうたび、保護された場所も考慮すべきであることを意味します。



以前の分析からは、保護のステータスに関する情報を登録することスタンドを取得できませんでした。この問題を解決するには：

- guilabel：実行「再び location」ツールで属性が参加。
- しかし、今回は結合レイヤとして squirrelrel_15m レイヤを使用しています。
- stands_squirrelrel_15m.shp として出力ファイルに名前を付けます。



この新しいレイヤの属性テーブルを開き、今あなたが保護の場所についての情報を持っている3つの林分を持っていることに注意してください。森の中の情報は、データを考慮に入れるべき保護の考慮事項があることを森林管理に指示します立っています。KBD：それから彼または彼女はから位置を取得することができます 'squirrel' データセットを、そしてフィールドでのオペレーターがリス環境を乱さないことができるように場所を中心に対応するバッファをマークするエリアをご覧ください。

14.4.5 Try Yourself 流れへの距離で林分を更新する

保護されたリスの位置について示されているように、同じアプローチに従って、今、林分を野外で識別されたストリームに関連した保護情報に更新できます。

- この場合、バッファはその周り 20 メートルであることを覚えておいてください。

- ターゲットとして `stands_squirrel_15m` レイヤ同じベクトルファイル内のすべての保護情報を持っているので、使用したいです。
- 出力に `forest_stands_2012_protect.shp` と名前を付けます。

新しいベクタレイヤーの属性テーブルを開き、今、流れに関連付けられた水辺林を保護するための保護措置の影響を受けている林分のためのすべての保護情報を持っていることを確認します。

QGIS プロジェクトを保存します。

14.4.6 In Conclusion

林分をデジタル化する CIR 画像をどのように解釈するかを見てきました。もちろんそれは、より正確なスタンドを作るためにいくつかの練習を取るだろうし、通常の土壌マップのような他の情報を用いて、より良い結果を与えるだろうが、あなたは今、このタイプのタスクのための基礎を知っています。そして、他のデータセットからの情報を追加すると、非常に簡単な作業であることをなりました。

14.4.7 What's Next?

林分は、将来的に森林施業を計画するために使用されるデジタル化され、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、インベントリにだけのデジタル化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、森のパラメータの全体的な見積もりを取得します。

14.5 Lesson: 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ル化しましたが、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、目録に全体の森林面積の調査を設計し、そのパラメータを推定できます。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、明らかに、サンプルで使用されるプロットし、目的を達成するために収集されるデータの種類を目標を定義することが重要です。個々の場合のために、それらは、森林管理目的のタイプに依存するであろう。そして、林業の知識を持つ誰か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：森林地域を調査するために体系的なサンプリングプロットのデザインを作成します。

14.5.1 森林目録

森林目録にはいくつかの方法が、それらのそれぞれが異なる目的や条件に合った、があります。たとえば、森林の目録への1つの非常に正確な方法は（唯一の樹種を検討している場合）森林を訪問し、すべての木とその特性のリストを作成することです。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き、一般的には適用されません。

森を知るための最も一般的な方法は、つまり、それをサンプリング森で異なる場所で測定を行うと、森全体にその情報を一般化しています。これらの測定は、多くの場合、容易に測定することができる小さな森林地域です*サンプルプロット*で作られています。サンプルプロットは、ex のために（可変サイズ、矩形、円形 EX 用）（例：50 M2、0.5 ヘクタール）任意のサイズ及び形状にすることができ、そして異なる方法で森の中に配置することができる。ランダムに、体系的、線に沿って）サンプルプロットの大きさ、形状及び位置は、通常、統計的、経済的で実用的な考慮事項を以下決定されます。何の林業の知識を持っていない場合は、この Wikipedia の記事<http://en.wikipedia.org/wiki/Forest_inventory> _ を読んで興味があるかもしれません。

14.5.2 Follow Along: 体系的サンプリングプロットデザインを実装します

作業している森のために、管理者は、体系的なサンプリング・デザインは、この森のために最も適切であると判断したと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ますことを決定しました（この場合のために、68 %の確率で平均誤差 $\pm 5\%$ ）。可変サイズのプロットは、成長および成熟林分のため、この在庫のための最も効果的な方法であることが決定されているが、4 メートルの半径のプロットは苗林分に使用される固定されました。

実際には、単に後でフィールド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ります。

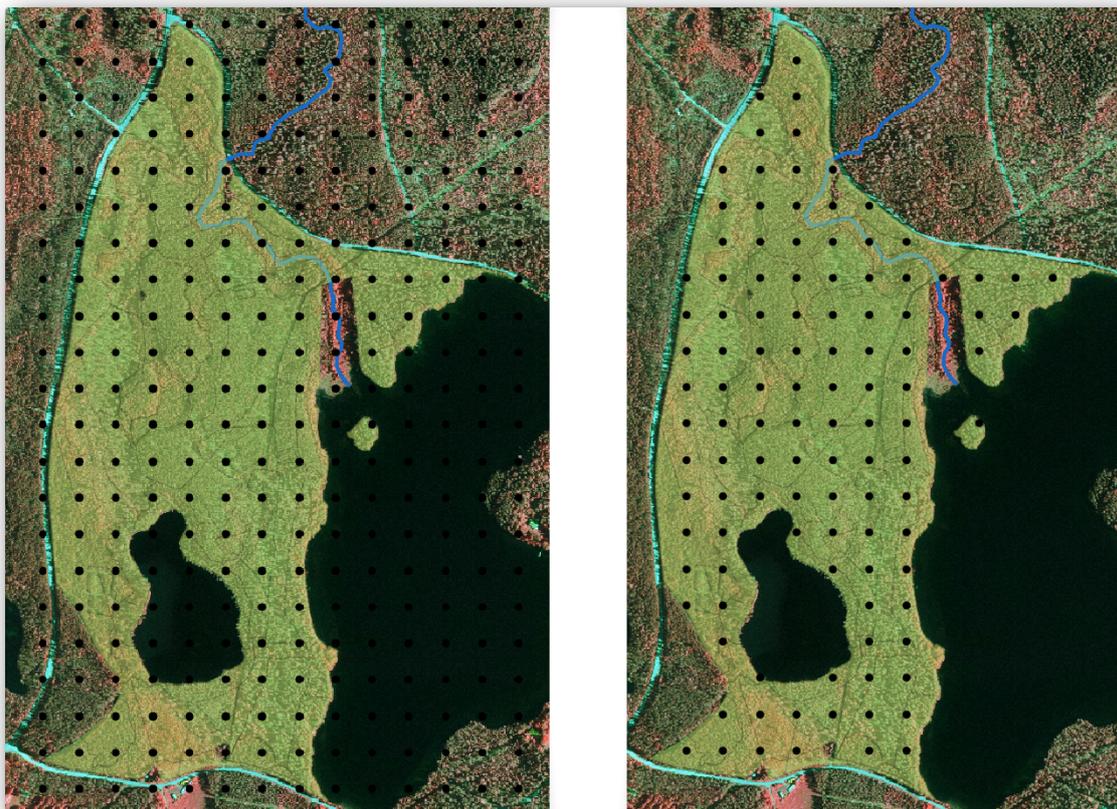
- 前のレッスンから `digitizing_2012.qgs` プロジェクトを開きます。
- `forest_stands_2012` 以外のすべてのレイヤーを削除します。
- 今、プロジェクトを `forest_inventory.qgs` として保存します

今、お互いから 80 メートルを隔て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。

- ベクトル -> 研究のツール -> 規則的 *points* を開きます。
- *Area* 定義で 入力バウンダリ *Layer* を選択します。
- `forest_stands_2012` レイヤーを入力レイヤーとして設定します。
- グリッド *Spacing* 設定で、この時点 *spacing* を使用 を選択し、それを 80 に設定します。
- 出力を 林業\サンプリング\フォルダに `systematic_plots.shp` として保存します。
- 結果をキャンバスに追加 をチェックします。
- *OK* をクリックします。

ノート: 提案された 規則点 は、選択したポリゴンレイヤーの広がりコーナーを左上隅から始まる体系的なポイントを作成します。この規則的なポイントにいくつかのランダム性を追加したい場合は、(80 が我々の点間の距離である) 0 と 80 の間でランダムに計算された数を使用して、としてそれを書くことができます ツールのダイアログで コーナーから最初のインセット (*LH* 側) パラメータ。

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、林分レイヤーの全体の範囲を使用していることに気づきます。しかし、お使いの森林面積の内側に実際にそれらの点にのみ興味を持っています (下の画像を参照):



- ベクトル -> ジオプロセッシングツール -> *Clip* を開きます。
- 入力ベクトル *layer* として *systematic_plots* を選択します。
- クリップ *layer* として *:kbd: 'forest_stands_2012* を設定します。
- *systematic_plots_clip.shp* として結果を保存します。
- 結果をキャンバスに追加 をチェックします。
- *OK* をクリックします。

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を持っています。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なお準備することができます。少なくとも、ポイントのために意味のある名前を追加し、その GPS デバイスで使用できる形式にエクスポートする必要があります。

サンプルプロットの命名で始めることができます。チェックした場合 自動で生成され *id* フィールド 正規 *points* ツール 森林面積の内側プロットの属性 *table* を、デフォルトを持っていることがわかります。ポイントにラベルを付けて地図でそれが見えるようにし、これらの番号をサンプルプロットの命名の一部として使用できるか考えます。

- *systematic_plots_clip* の レイヤーのプロパティ]-> [*Labels* を開きます。
- ラベルこの層 *WITH* をチェックし、*ID* フィールドを選択します。
- バッファ オプションに行ってテキスト *buffer* を描画 をチェックし、*Size* を *:kbd:1* に設定します。
- *OK* をクリックします。

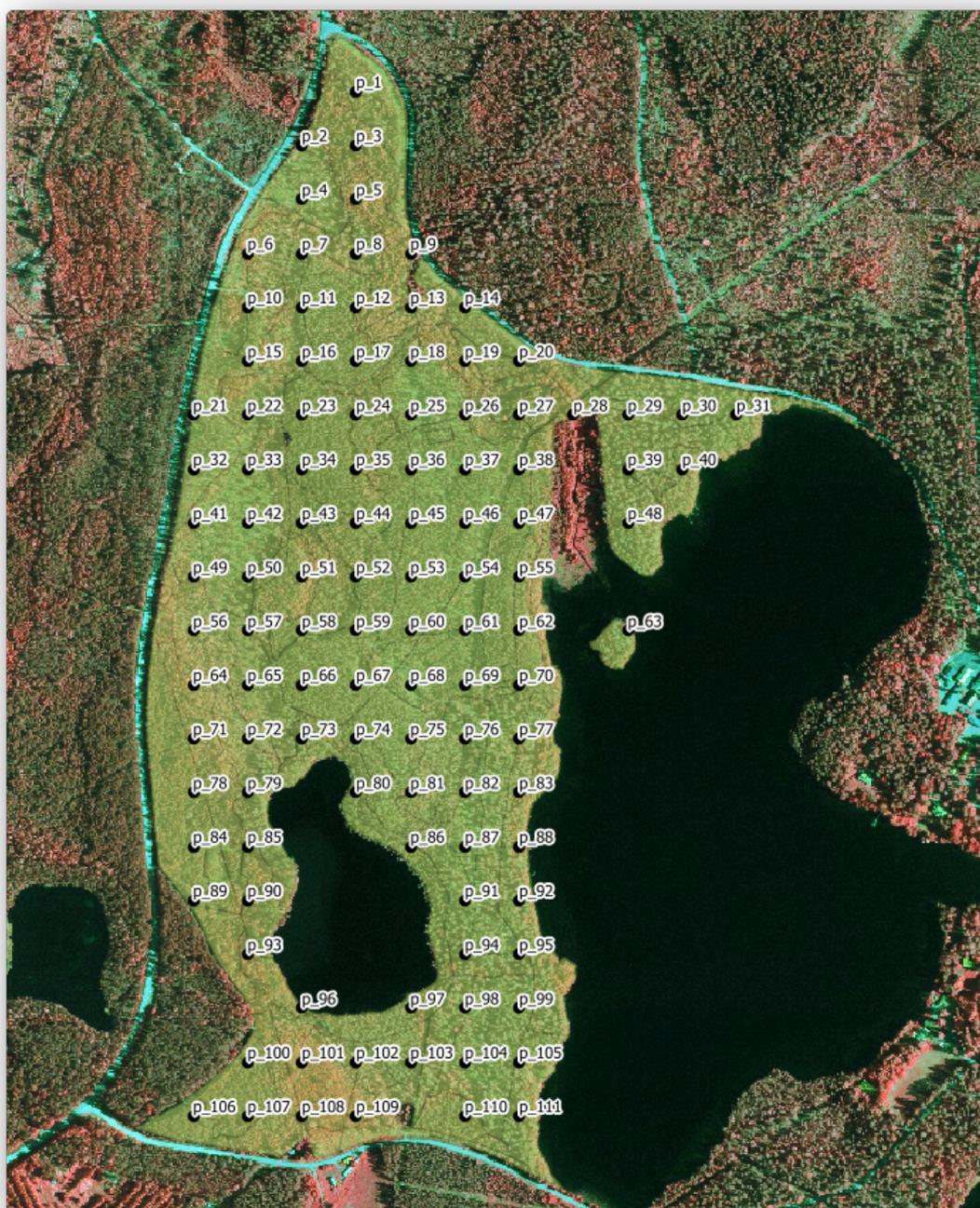
今、地図上のラベルを見てください。ポイントが作成され、最初に西から東、それから北から南へと番号付けられていることがわかります。再び属性テーブルを見れば、テーブル内の順序もそのパターンに従っていることがわかります。異なる方法でサンプルプロットに名前付けする理由がない限り、名前を西東/南北ファッションで名前付けすることは論理的な順序に従っており、良いオプションです。

ノート: 別の方法でそれらを注文するか、名前を付けたい場合は、任意の異なる方法で行と列を注文し、組み合わせることができるようにスプレッドシートを使用することができます。

それにもかかわらず、id フィールド中の数値はそれほど良いものではありません。命名は p_1、p_2 ... のようなものであれば良かったでしょう。systematic_plots_clip レイヤーのための新しい列を作成できます。

- systematic_plots_clip のための 属性テーブル に行きます。
- 編集モードを有効にします。
- Open the *Field calculator* and name the new column Plot_id.
- 出力フィールド type を:kbd:テキスト (文字列) に設定します。
- *Expression* フィールドで、この数式 CONCAT('P_', \$rownum) を構築、書き込み、コピーします。機能 list も内部の要素をダブルクリックできますことを覚えておいてください。STRING と:kbd:concat 関数は下にあります:guilabel:Record の:kbd:\$rownum パラメータは下にあります。
- OK をクリックします。
- 編集モードを無効にし、変更を保存します。

今、意味のあるプロット名を持つ新しい列を持っています。:kbd:'systematic_plots_clip' レイヤーについて、ラベル付けに使用するフィールドを新しい:kbd:'Plot_id'フィールドに変更します。



14.5.3 Follow Along: GPX 形式としてサンプルプロットを書き出す

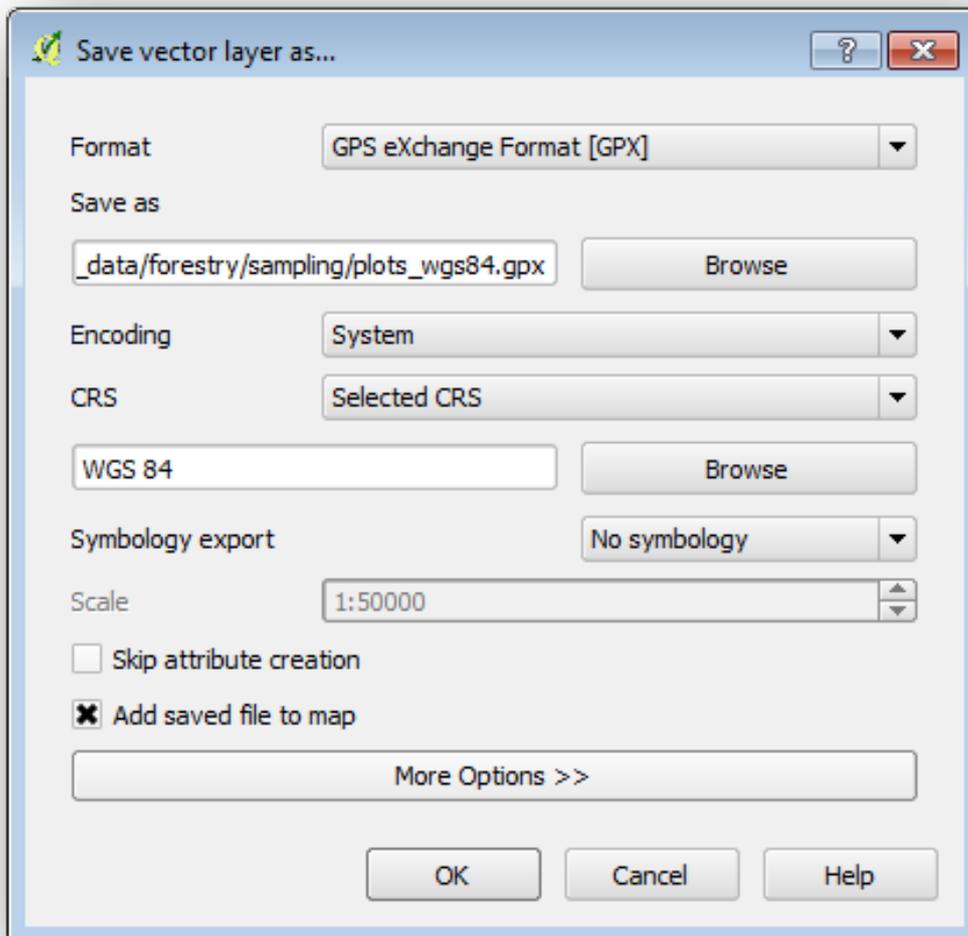
野外チームはあなたが計画したサンプルのプロットを見つけるために GPS デバイスを使用しているでしょう。次のステップは、GPS を読むことができる形式に作成されたポイントを書き出すことです。QGIS は、特殊なソフトウェアのほとんどで読み取ることができる標準の GPS データ形式である、http://en.wikipedia.org/wiki/GPS_Exchange_Format GPS 交換フォーマット (GPX) にポイントやライン、ベクトルデータを保存できます。データを保存する際に CRS を選択して注意する必要があります。

- `:kbd:'systematic_plots_clip'` を右クリックし、`:guiabel:'名前をつけて保存'` を選択します。
- `Format` で `:guiabel:GPS 交換フォーマット [GPX]` を選択します。

- plots_wgs84.gpx として出力を保存します。
- CRS 中で 選択された CRS を選択します。
- WGS 84 (EPSG : 4326) を参照します。

..note:: GPX 形式はこの CRS だけを受け入れる、別のものを選択した場合、QGIS には、エラーを与えませんが、空のファイルが得られます。

- OK をクリックします。
- 表示されるダイアログで、waypoints レイヤーのみを選択します (レイヤーの残りの部分は空である)。



目録サンプルプロットは、GPS ソフトウェアのほとんどで管理できる標準形式になりました。フィールドのチームは現在、自社のデバイスにサンプルプロットの位置をアップロードすることができます。plots_wgs84.gpx はだけ保存したファイルは、その特定のデバイス自身のソフトウェアと使用して行われることとなります。他のオプションが使用することで *GPS Tools* プラグインを、それは最も可能性の高い特定の GPS デバイスで動作するようにツールを設定する伴うだろう。自身のデータを操作し、ツールがどのように動作するかを確認したい場合は、:guilabel:' QGIS ユーザー Manual' 中の 'GPS データでの作業' <http://docs.qgis.org/2.2/en/docs/user_manual/working_with_gps/index.html> 'セクション' でそれについての情報を見つけることができます。

ここで QGIS プロジェクトを保存します。

14.5.4 In Conclusion

ちょうど森林目録に使用する体系的なサンプリングのデザインを作成する方法を簡単に見ました。サンプリングデザインの他のタイプを作成すると、サンプルプロットの座標を計算するために、QGIS、スプレッドシートやスクリプト内の異なるツールの使用を含むが、一般的な考え方は同じままになります。

14.5.5 What's Next?

次のレッスンでは、QGIS でアトラス機能を使用して、野外チームがそれらに割り当てられたサンプルプロットにナビゲートするために使用している詳細地図を自動的に作成する方法について説明します。

14.6 Lesson: アトラスツールで詳細な地図を作成します

体系的なサンプリングのデザインが準備できて、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フィールドのデータ形式を持っています。野外チームは、すべてのサンプルプロットに自分の道が簡単に見つかるよう、いくつかの地上情報をサンプルプロットの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ることができると多くの詳細地図を要求してきました。アトラスツールを使用することで、多く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 でアトラスツールを使用して、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 Follow Along: 地図コンポーザを準備

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できるようになるには、まずフィールドワークのために有用と考えるすべての要素が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、適切にスタイルされることですが、印刷された地図を完成する他の要素の多くを追加する必要もあります。

前のレッスンから QGIS プロジェクトを開きます `forest_inventory.qgs`。少なくとも以下のレイヤーを持っている必要があります。

- `forest_stands_2012` (50%の透明度、緑塗りつぶしと暗緑色の境界線を有する)。
- `systematic_plots_clip`。
- `rautjarvi_aerial`。

プロジェクトを新しい名前 `map_creation.qgs` で保存します。

印刷可能なマップを作成するには、コンポーザマネージャーを使用することを思い出してください：

- プロジェクト->コンポーザマネージャー... を開きます。
- コンポーザマネージャー ダイアログで。
- 追加 ボタンをクリックして、コンポーザに `forest_map` と名前を付けます。
- OK をクリックします。
- 表示 ボタンをクリックします。

地図が A4 用紙のために、紙や余白に合うように、プリンタオプションを設定します。

- コンポーザ->ページ設定 を開きます。
- サイズ は A4 (217 X 297 ミリメートル)。
- 向き は 風景。
- マージン (*millimeters*) はすべて 5 に設定。

印刷組版 ウィンドウで、*Composition* タブ (右のパネル) に行き、紙と品質 に対するこれらの設定がプリンタ用に定義されたものと同じであることを確認してください：

- *Size* : A4 (210x297mm)。
- *Orientation* : Landscape。
- 品質 : 300dpi。

さまざまな要素を配置するキャンパスのグリッドを利用する場合はマップを構成することは簡単です。コンポーザグリッドの設定を確認します。

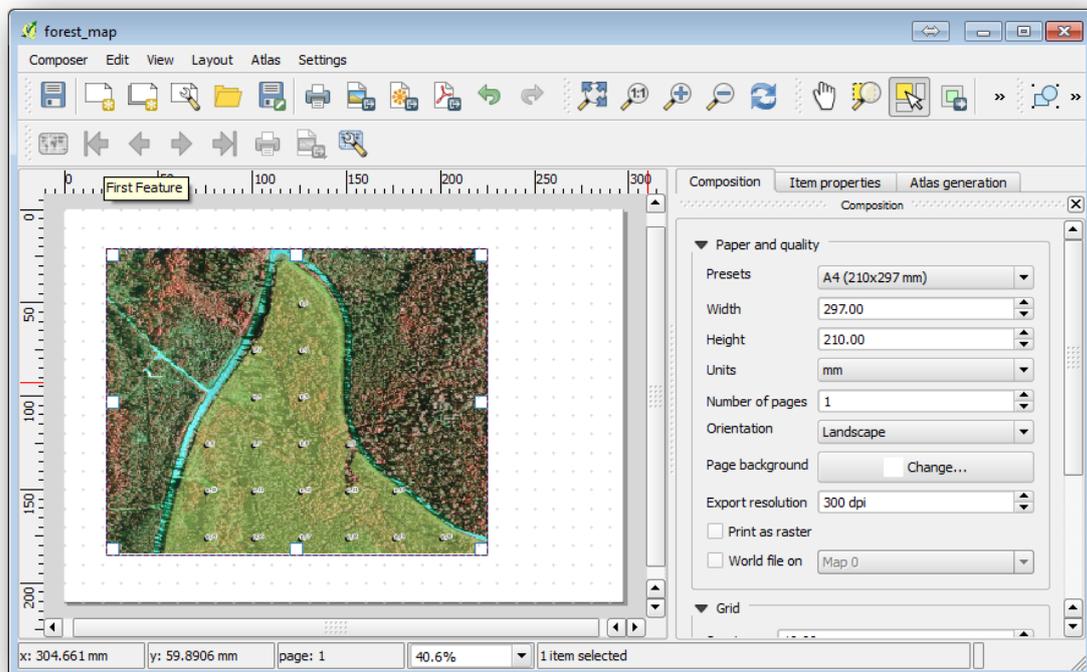
- *Composition* タブ中で *Grid* 地域を展開します。
- 余白が 10 mm に設定されていることを確認してください。
- そして *Tolerance* は 2mm に設定されています。

グリッドの使用を有効にする必要があります。

- 表示メニューを開きます。
- *grid* を表示 をチェックします。
- *grid* にスナップ をチェックします。
- *guides* を使用するためのオプションがデフォルトでチェックされていて、コンポーザ内の要素を移動しているときに赤いガイドラインを見ることができることに注意してください。

今すぐ地図キャンパスへの要素の追加を開始できます。レイヤーシンボルに変更を加えるとそれがどのように見えるか確認できるよう、最初にマップ要素を追加します。

- 新しい地図を追加 ボタンをクリックします .
- 地図はそれのほとんどを占めるようにキャンパス上のボックスをクリックしてドラッグします。



マウスカーソルがキャンパ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。より多くの精度を持つようにしたい場合は、グリッドを変更 *Spacing* 設定。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、常にそれを *View* メニューでチェックをオンオフできます。

14.6.2 Follow Along: 背景マップを追加します

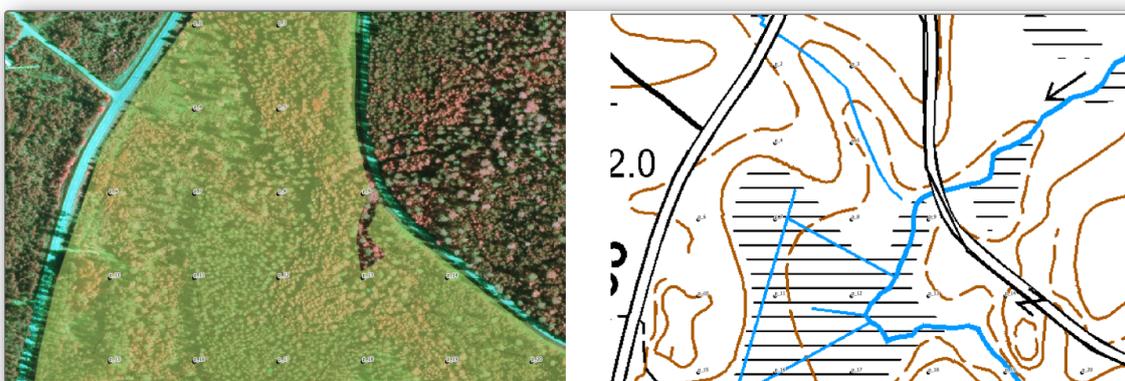
コンポーザを開いたまま、地図に戻ります。地図の内容ができるだけ明確になるように、いくつか背景データを追加したり、いくつかスタイルを作成したりできます。

- 背景 ラスタ `basic_map.tif`、`exercise_data\林業\フォルダ`に見つけることができる、を追加します。
- プロンプトが表示されたら `ETRS89/ETRS-TM35FIN CRS` をラスタのために選択します。

見ることができるように背景地図にはすでにスタイルが付いています。地図作成のラスタを使用する準備ができてこのタイプは非常に一般的です。いくつかのベクトルレイヤーをスタイリングして良い結果を得ることを心配気にする必要がないようにそれは、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。

- 今、プロットの約 4 または 5 行を見ることができるよう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リングは最高ではありませんが、それはどのようにマップの作曲に見えるん?:



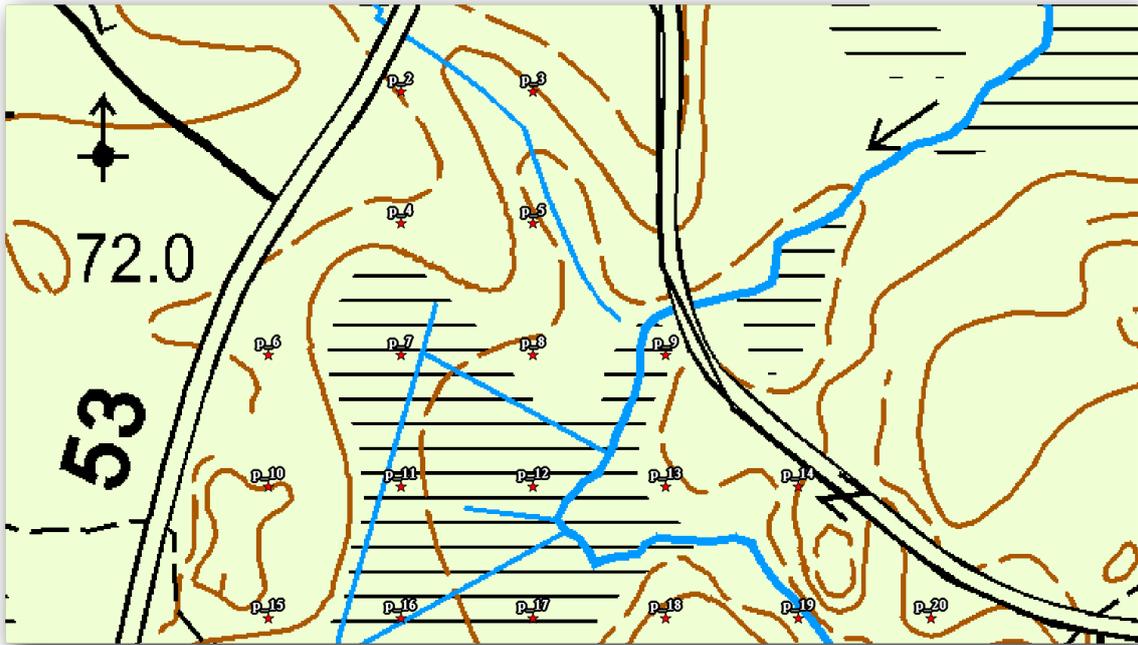
最後の演習中に、白いバッファは空撮画像の上では OK でしたが、今で背景画像はほとんどが白であり、かろうじてラベルを見ることができます。コンポーザでどのように見えるか確認もできます。

- 印刷 *Composer* ウィンドウに行きます。
- コンポーザの地図要素を選択するため  ボタン を使用します。
- 項目 *properties* タブに移動します。
- 範囲 の下で `:guilabel'地図キャンパス範囲に設定'` をクリックしてください。
- 要素を再描画する必要がある場合は、メイン *properties* 下の *プレビュー* を更新 をクリックしてください。

野外チームのためにプロット番号ができるだけはっきり見えるようにしたい、これは明らかに十分ではありません。

14.6.3 Try Yourself レイヤーのシンボルを変更します

作業してきた *Module: 基礎的な地図作成* シンボルとし、中 ラベル付き `../vector_classification/index` 。利用可能なオプションとツールのいくつかについて、リフレッシュする必要がある場合は、それらのモジュールに戻ります。目標は、のようにはっきりと見える可能のようにプロットの位置と自分の名前を取得することですが、常に背景地図要素を参照できます。この画像からいくつかのガイダンスを取ることができます。

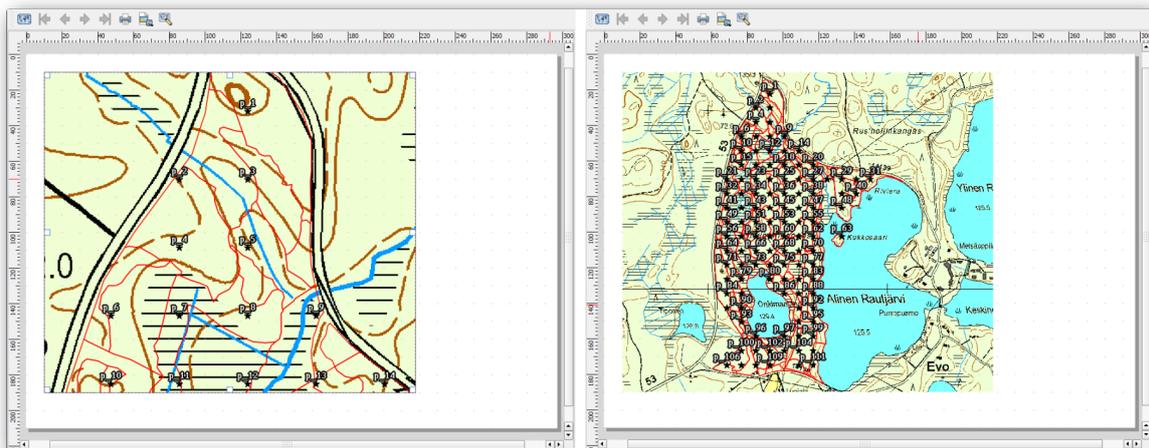


後で forest_stands_2012 レイヤーの緑のスタイリングを使用するでしょう。それを維持し、林分の境界だけを表示するようそれを可視化するには：

- forest_stands_2012 を右クリックし *Duplicate* を選択します
- たとえば充填と赤のボーダーがない別のスタイルを定義するために使用できる、forest_stands_2012 copy という名前の新しいレイヤーを取得します。

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。

プリントコンポーザ ウィンドウに何度も戻り、地図がどのように見えるかを確認します。詳細地図を作成する目的のために、全体の森林地域の縮尺(下の左画像)ではなくより近い縮尺(下の右画像)で良く見えるシンボルを探しています。地図やコンポーザでズームを変更するたびに、プレビューを更新と地図キャンバス extent 設定を使用することを忘れないでください。

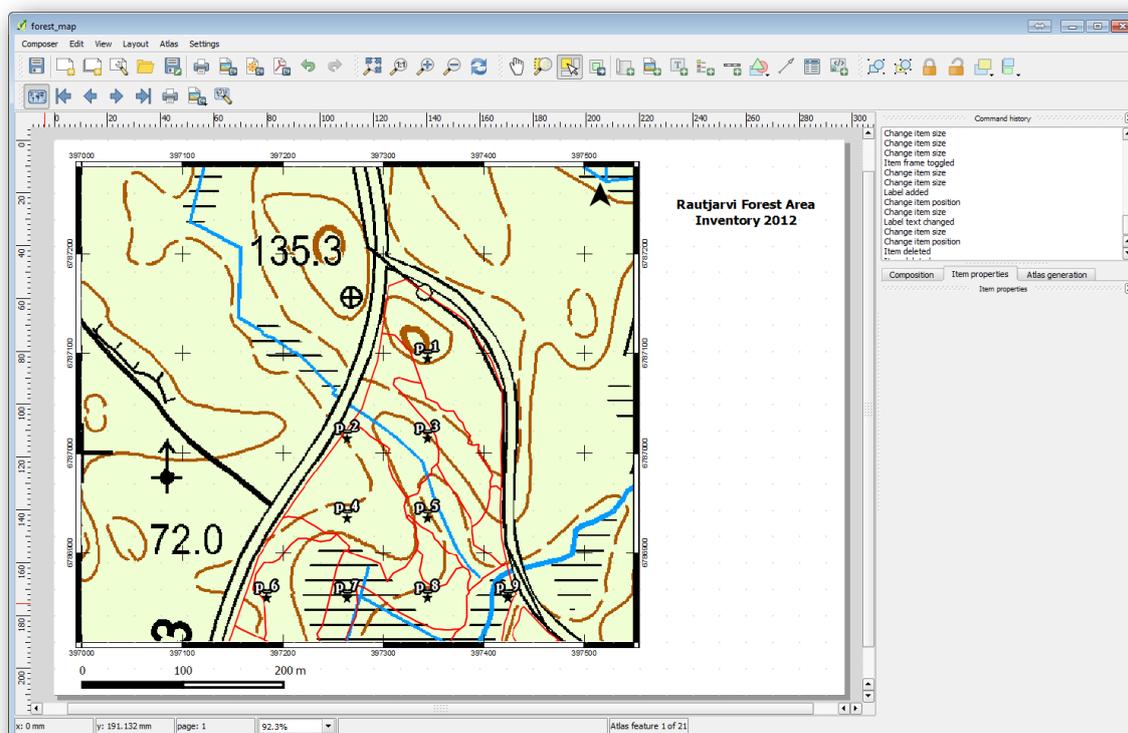


14.6.4 Try Yourself 基本地図テンプレートを作成する

満足なシンボル体系を得たら、印刷された地図にいくつかのより多くの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ます。

- タイトル。
- スケールバー。
- マップのグリッドフレーム。
- グリッドの両側に調整します。

Module: マップの作成 既に同様の組成を作成しました。必要とするように、そのモジュールに戻ります。参考のために、この例の画像を見ることができます。



地図を画像として書き出し、それを見てください。

- コンポーザ-> 画像として書き出し。
- 例えば JPEG フォーマット を使用してください。

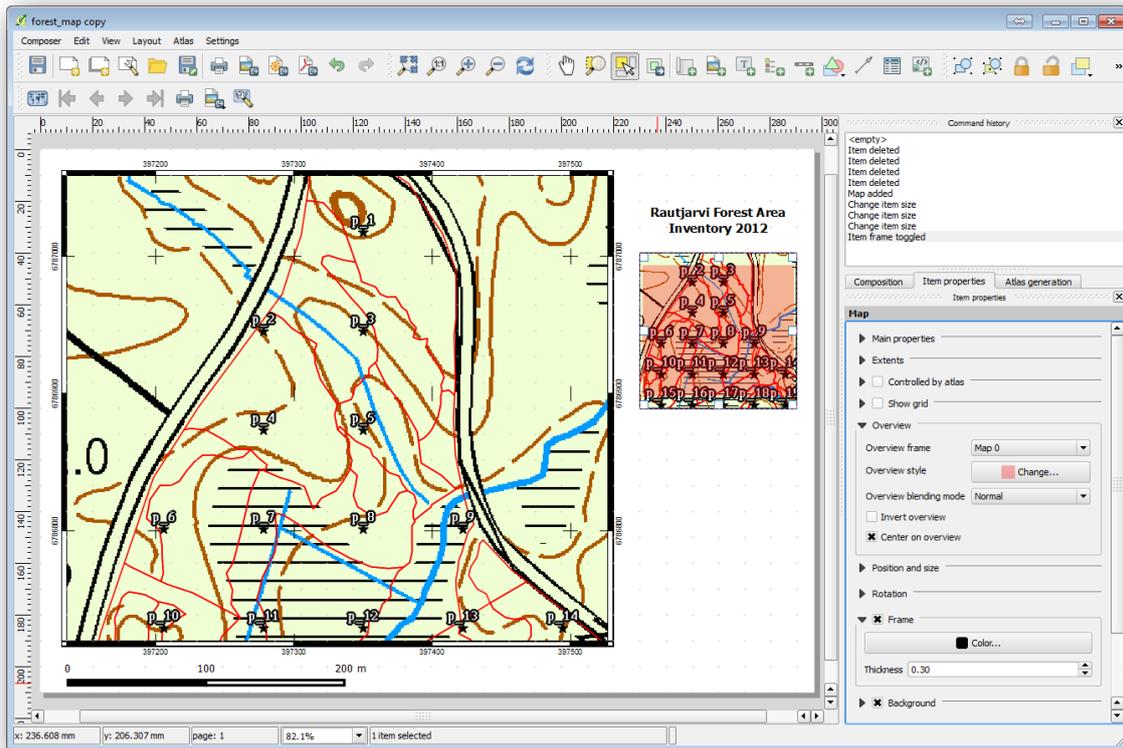
これは、印刷されたときに、それがどのように見えるかです。

14.6.5 Follow Along: コンポーザに多くの要素を追加します

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、キャンパスの右側に空間がたくさんあります。そこに行くことができる他に何見てみましょう。ここでの地図の目的のためには凡例は本当は必要はありませんが、概要地図といくつかのテキストボックスには、地図に値を追加することができます。

概観マップは、フィールドのチームは、一般的な森林エリア内の詳細マップを置くのに役立ちます。

- タイトルテキストのすぐ下で、キャンパスに別の地図要素を追加します。
- アイテムのプロパティ タブで、*Overview* ドロップダウンを開きます。
- 概要フレーム を地図:guiabel: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。
- *Frame* オプションが黒色で、*Thickness* が 0.30 であることもチェックします。



概観地図は本当に望むものである森林地域の概観を与えていないことに注意してください。この地図が全体の森林面積を表現したいとそれが唯一の背景地図と表示したい forest_stands_2012 レイヤーを、サンプルプロットを表示しません。そしてまた、レイヤーの可視性や順序を変更するたびに、それはもう変更されないように、そのビューをロックします。

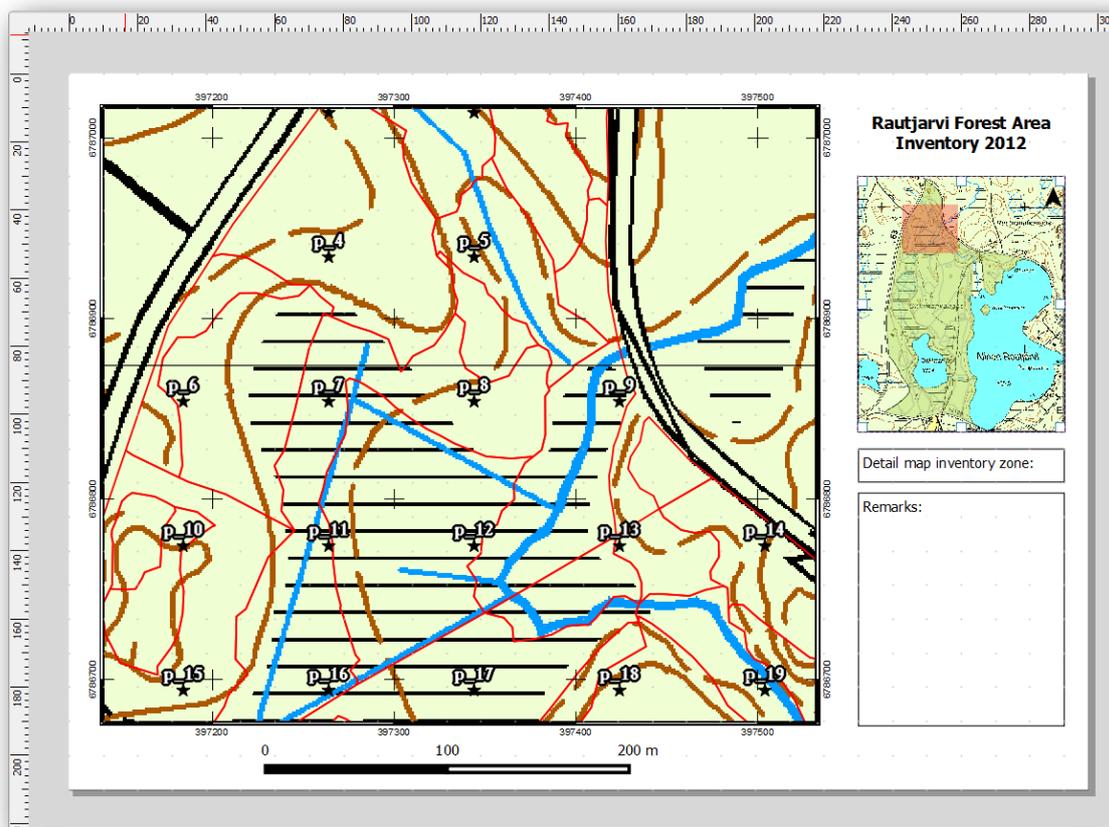
- 地図に戻りますが、印刷 composer を閉じないでください。
- forest_stands_2012 レイヤーを右クリックし、レイヤー範囲にズーム をクリックします。
- basic_map と forest_stands_2012 以外のすべてのレイヤーを無効にします。
- 印刷 composer に戻ります。
- 選択された小地図では、地図キャンバス extent を設定 をクリックし、その範囲を地図ウィンドウで見えるものに設定してください。
- メイン properties 下の 地図 item のためにレイヤーをロック をチェックすることにより、概観マップのビューをロックします。

今概観地図は、予想するより以上のものであり、そのビューはもう変更しないでしょう。しかし、もちろん、詳細地図は今まだ林分境界もサンプルのプロットも表示していません。修正しましょう：

- 再度、地図 ウィンドウに移動し、見えるようにしたいレイヤーを選択します (systematic_plots_clip、forest_stands_2012 copy、Basic_map)。
- 目に見えるサンプルプロットの数行を持って再び拡大します。
- プリントコンポーザ ウィンドウに戻ります。
- コンポーザで大きな地図を選択します (🖱️)。
- 項目 properties 中で プレビューを更新 と 地図キャンバス範囲に設定 をクリックしてください。

大きな地図だけが、現在の地図ビューを表示している、小さな索引地図は、それをロックしたときに持っていた同じ景色を保っていることに注意してください。

索引図は、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注意。



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：」含む、もう一方は「備考：」。それらが上の画像で見えるように配置してください。

また、概観マップに北矢印を追加できます。

- *image* を追加 ツールを使用してください 。
- 概観マップの右上にクリックしてください。
- アイテムのプロパティで 検索ディレクトリ を開き、矢印画像を参照します。
- 画像回転の下で 地図で Sync をチェックし、Map 1 (概要地図) を選択します。
- 背景 のチェックを外します。
- 小さな地図上良く見えるサイズに矢印画像のサイズを変更します。

基本的な地図コンポーザーが準備できたところで、今度はアトラスツールを利用してこの形式で必要と考えるだけ多くの詳細地図を作成したいですね。

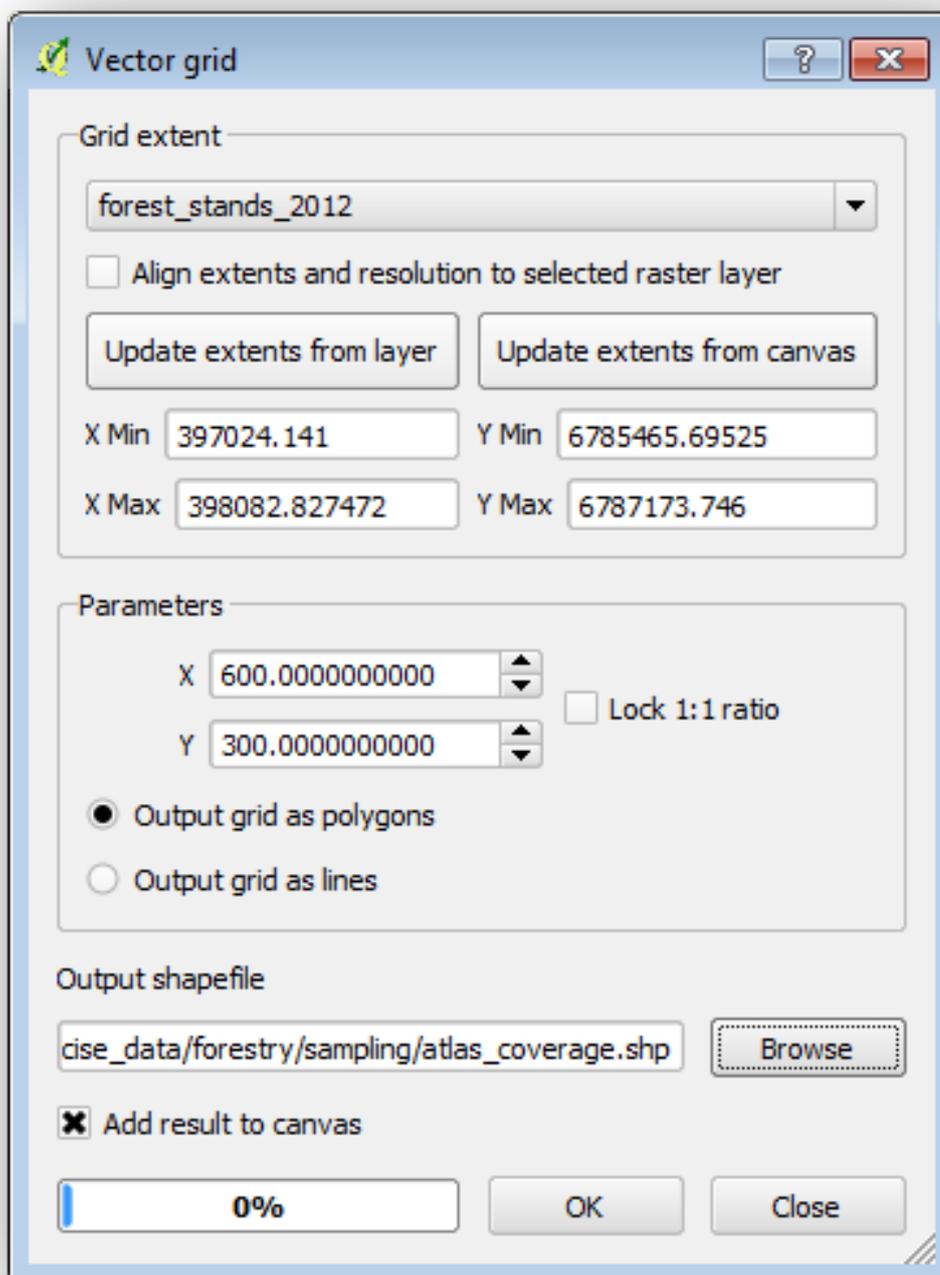
14.6.6 Follow Along: アトラスカバレッジを作成します

アトラスカバレッジは、カバレッジ内のすべての機能の1枚の地図を詳細地図を生成するために使用されるだけでベクトルレイヤーです。次に何をしますかのアイデアを取得するには、ここでは詳細地図のフルセットは、森林面積のためであります：



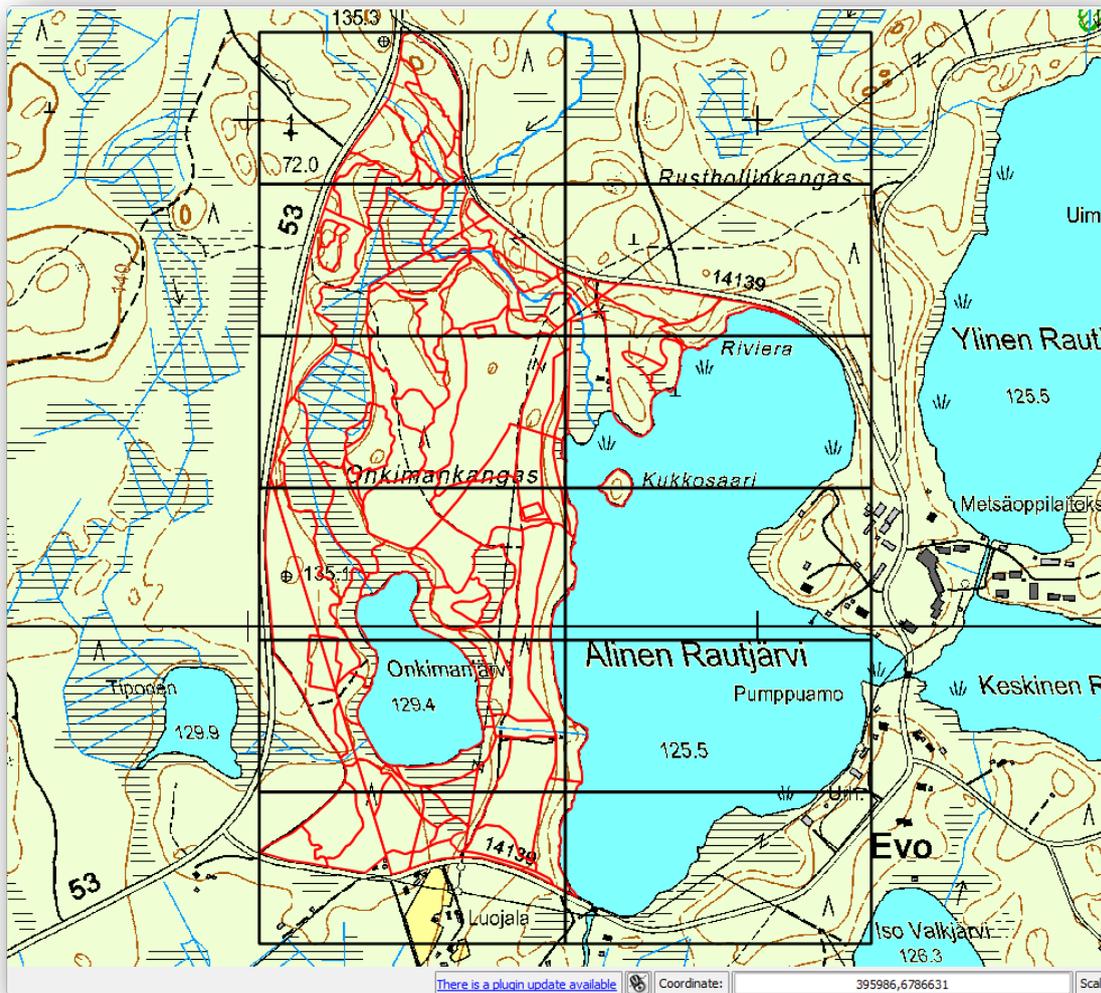
カバレッジでは、既存のレイヤーかもしれないが、通常は特定の目的のための1つを作成するために、より多くの意味があります。のは、森林面積をカバーするポリゴンのグリッドを作成してみましょう：

- QGIS 地図ビューでベクトル -> 研究のツール -> ベクトル *grid* を開きます。
- この画像に示すようにツールを設定します。



- atlas_coverage.shp として出力を保存します。
- 新しい atlas_coverage レイヤーを、ポリゴンが塗りを持たないようにスタイル設定します。

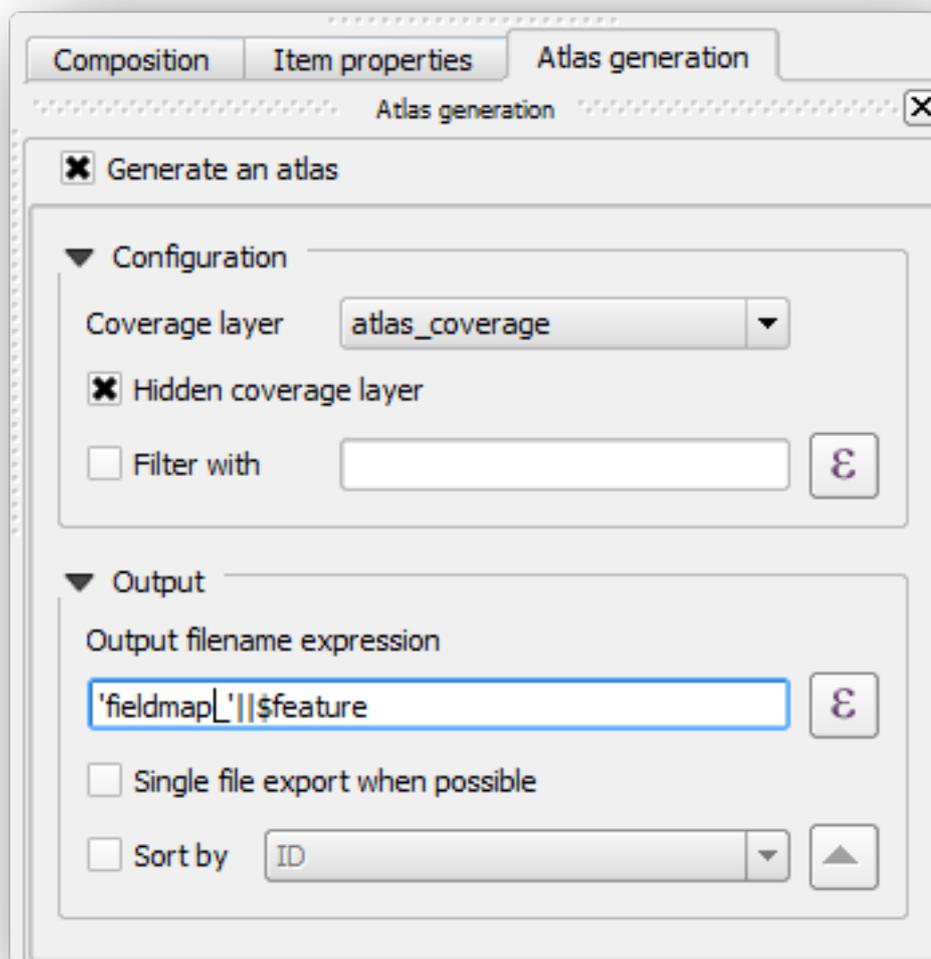
新しいポリゴンが全体の森林地域をカバーしていると、彼らは（各ポリゴンから作成された）各地図の何が含まれているかのアイデアを与えます。



14.6.7 Follow Along: アトラスツールの設定

最後のステップは、アトラスツールを設定することです：

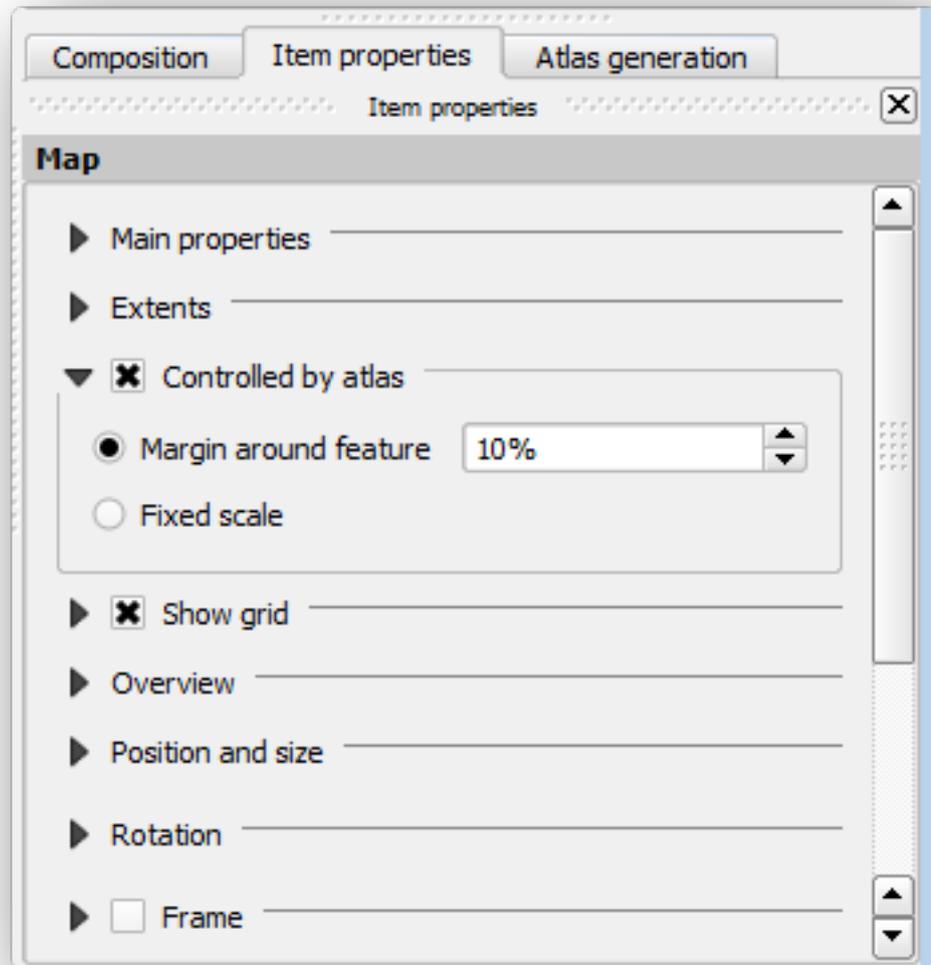
- プリントコンポーザに戻ります。
- 右側のパネルでアトラス *generation* タブに行きます。
- 次のようにオプションを設定します。



細部マップの焦点として atlas_coverage 内部のそれは地物（ポリゴン）を使用するアトラスツールに指示します。これは、レイヤー内のすべての地物の出力 1 つの地図になります。隠しカバレッジレイヤーは出力地図でポリゴンを表示しないようにアトラスに伝えます。

もう一つする必要があります。すべての出力地図に対してどの地図要素が更新されるつもりかをアトラスツールに伝える必要があります。今までは、おそらく、すべての地物のために変更される地図は、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であると推測でき、それがキャンバスでより大きい地図要素であります：

- より大きな地図要素を選択します。
- 項目 *properties* タブに移動します。
- リストで、*atlas* によって制御 をチェックします。
- そして地物周りにマージン *g* を 10 % と設定します。ビューの範囲は多角形よりも 10 % 大きくなります、これは詳細地図が 10% overlap を持っていることを意味します。



今、地図がどのように見えるか確認するためにアトラスマップのプレビューツールを使用できます。

- ボタン  を使用して、アトラスのプレビューを有効に、またはアトラス ツールバー が表示されていない場合は アトラス ->プレビュー *Atlas* を通じて。
- 作成される地図を移動するには、アトラス・ツール・バーまたは *Atlas* メニューの矢印を使用できます。

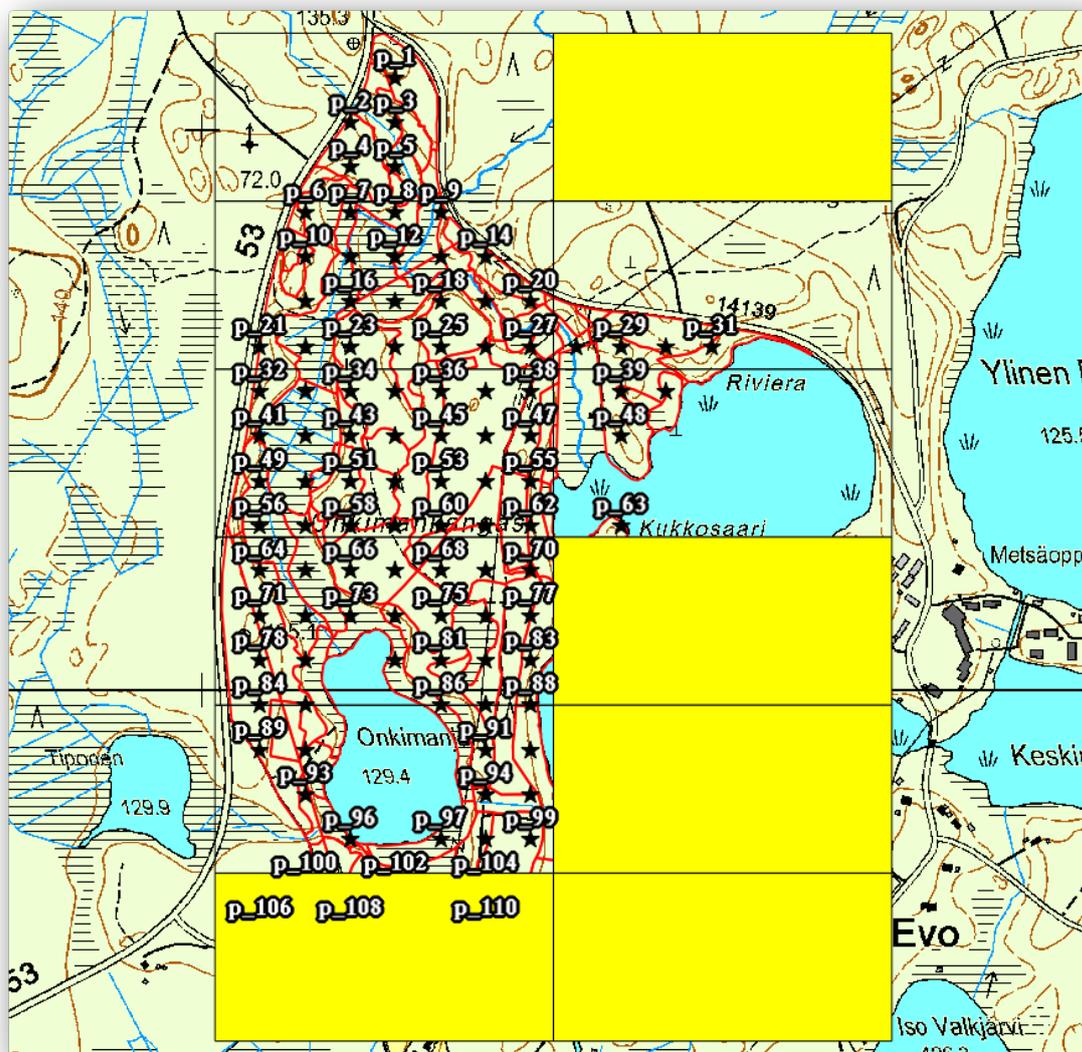
それらのいくつかは関心のない領域をカバーすることに注意してください。それについて何かをして、それらの無用な地図を印刷しないことで、木をいくらか節約しましょう。

14.6.8 Follow Along: カバレッジレイヤーの編集

それらの領域のために関心のないポリゴンを削除するだけでなく、カバレッジレイヤー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。

- マップビューに戻ります。
- `atlas_coverage` レイヤーの編集を有効にします。

- 下の画像に（黄色で）選択されたポリゴンを選択します。
- 選択したポリゴンを削除します。
- 編集を無効にして、編集内容を保存します。



印刷 *Composer* に戻って、アトラスのプレビューがレイヤーに残された唯一のポリゴンを使用することを確認できます。

使用しているカバレッジレイヤーは、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、アカウントに持っている野外チームのためのいくつかの発言で、たとえばポリゴン領域のゾーンコードとフィールドを追加でき、それらを作成することです：

- atlas_coverage レイヤーのための 属性テーブル を開きます。
- 編集を有効に。
- 以下の二つのフィールドを作成し、移入する  電卓を使用します。
- Zone という名前の、タイプ 整数 (整数) のフィールドを作成します。
- Expression ボックスで \$rownum を書き込み/コピー/構築します。
- 名前が Remarks、タイプがテキスト (文字列)、幅が 255 の別のフィールドを作成します。

- *Expression* ボックスで 'NO remarks.' と書き込みます。これは、すべてのポリゴンに対してすべてデフォルト値を設定します。

森林管理は、地域を訪れる際に役に立つかもしれない地域に関するいくつかの情報を持っています。例えば、橋の有無、湿地や保護種の位置。atlas_coverage レイヤーはおそらくまだ編集モードです、Remarks フィールド中に次のテキストを対応するポリゴンに追加します (セルを編集するにはそのセルをダブルクリックしてください):

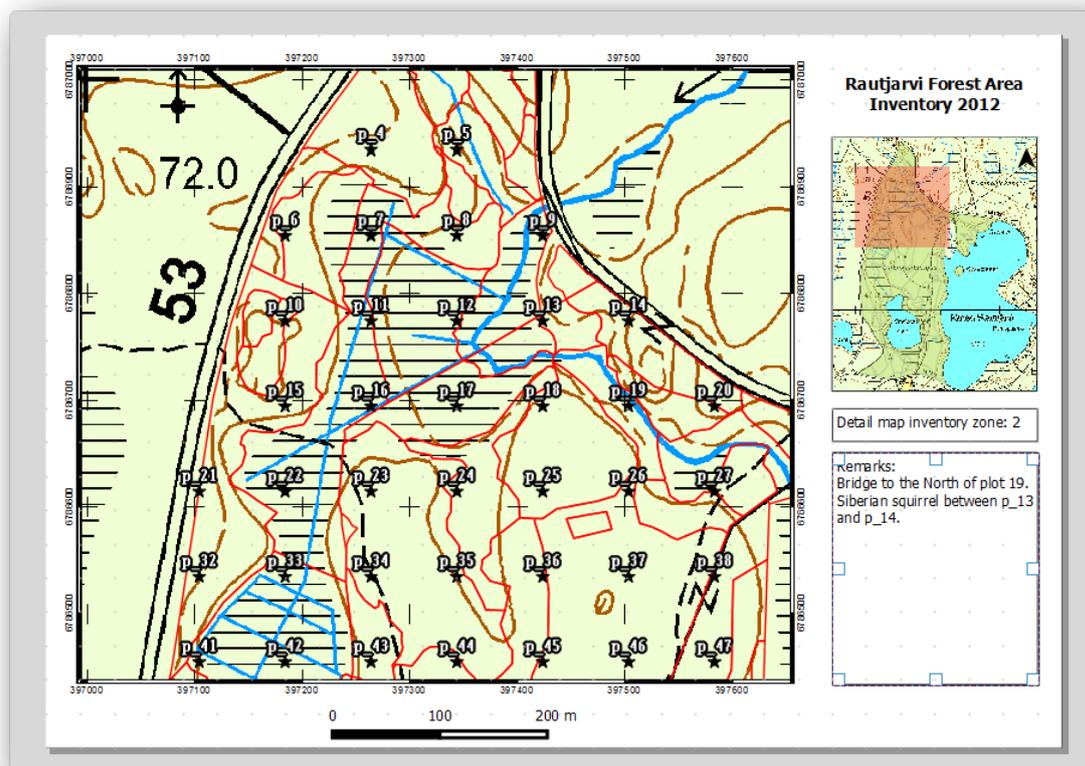
- Zone 2 : プロット 19 の北に橋。p_13 と p_14 間にシベリアリス。。
- Zone 6 : の場合:kbd:湖の北へ沼に移するのは困難。
- Zone 7 : の場合:kbd:シベリアリスを p_94.南東に。
- 編集を無効にして、編集内容を保存します。

atlas_coverage レイヤーの属性テーブルはほぼ準備ができて、今、テキストラベルの一部がそこから情報を使用したいアトラスツールを伝える必要があります。

- プリントコンポーザに戻ります。
- 詳細な地図を... を含むテキストラベルを選択します。
- フォント サイズを 12 に設定してください。
- ラベル内のテキストの末尾にカーソルを設定します。
- 項目 *properties* タブで、メイン *properties* の内側で 式を挿入 をクリックしてください。
- 機能 *list* で フィールドと *Values* の下のフィールド *zone* をダブルクリックします。
- OK をクリックします。
- 項目プロパティ 中のボックス内にテキストは 詳細マップの調査ゾーン : [% "ゾーン" %] が表示されるはずです。:kbd:[% 「ゾーン」 %] は atlas_coverage レイヤーからの対応する地物について、フィールド *zone* の値によって置換されることに注意。

異なるアトラスプレビューマップを見て、ラベルの内容をテストします。

テキストとラベルに同じ操作を行います 備考: フィールドを使用して、ゾーン情報を聖霊降臨祭。式を入力する前に、破断線を残すことができます。下の画像では、ゾーン 2 のプレビューのために結果を見ることが出来ます:



すぐに作成されるすべてのマップを閲覧して楽しむことがアトラスプレビューを使用してください！

14.6.9 Follow Along: 地図を印刷します

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。アトラス --> アトラスを PDF として書き出し... や アトラス>アトラス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：

- 右パネルのアトラス作成 タブに移動します。
- 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。このオプションを使用すると、すべての地図に 1 つのファイルを取得するチェックされていない場合、これは、PDF ファイルにまとめ、すべての地図を配置します。
- コンポーザ->PDF として書き出し... を開きます。
- PDF ファイルを exercise_data\林業\samplig\ map_creation\ フォルダ中に :kbd: inventory_2012_maps.pdf として保存します。保存

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。

同じように簡単にすべての地図（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）ここに作成される画像のサムネイルを見ることができるようのために別々の画像を作成できます。



印刷コンポーザで、組版テンプレートとして `:kbd:' forestry_atlas.qpt'` として `exercise_data\林業\map_creation\` フォルダ中に地図を保存します。 `:menuselection: Composer->Template` として保存を使用します。何度も何度もこのテンプレートを使用できます。

印刷 *Composer* を閉じ、QGIS プロジェクトを保存します。

14.6.10 In Conclusion

自動的に別のプロットに移動を助けるためにフィールドで使用されるように、詳細マップを生成するために使用できるテンプレート・マップを作成するために管理しています。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場合に利益が来ると、先ほど保存したテンプレートを使用できます。

14.6.11 What's Next?

次のレッスンではレーザー測量データを使用して DEM を作成し、それからそれを使用してデータや地図の視認性を高める方法を示します。

14.7 Lesson: 森林パラメータの計算

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用し、最初に森林全体に対して、それから前にデジタル化した林分に対して、森林のパラメータを計算しましょう。

このレッスンの目標：一般と林分レベルで森林のパラメータを計算します。

14.7.1 Follow Along: 調査結果を追加

野外チームは森林を訪問し、提供する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。

ほとんどの場合、情報は野外で紙のフォームに収集され、それからスプレッドシートに入力されます。サンプルプロット情報は QGIS で簡単に開くことができる `.csv` ファイルに凝縮されています。

調査の設計に関するレッスンからの QGIS プロジェクトを続けます、おそらくそれには `forest_inventory.qgs` という名前を付けました。

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。

- レイヤー ->区切りのテキストレイヤーを追加... に行きます。
- exercise_data\林業\結果\にあるファイル systematic_inventory_results.csv を参照します。
- 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。
- 座標のフィールドを X と Y フィールドに設定します。
- OK をクリックします。
- 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。
- 新しいレイヤの属性テーブルを開いてデータを見てみましょう。

exercise_data\林業\結果\ フォルダに位置するテキストファイル legend_2012_inventorydata.txt 内で、サンプルプロットの測定に含まれているデータの種類を読み取ることができます。

先ほど追加した systematic_inventory_results レイヤーは、実際にはテキスト情報を .csv ファイルで仮想的に表現したものにすぎません。続行する前に、調査の結果を実際のシェープファイルに変換します。

- systematic_inventory_results レイヤーを右クリックします。
- exercise_data\林業\結果\ フォルダを参照します。
- sample_plots_results.shp ファイルに名前を付けます。
- 保存したファイルを地図に追加 をチェックします。
- プロジェクトから systematic_inventory_results レイヤを削除してください。

14.7.2 Follow Along: 森林全体のパラメータ推定

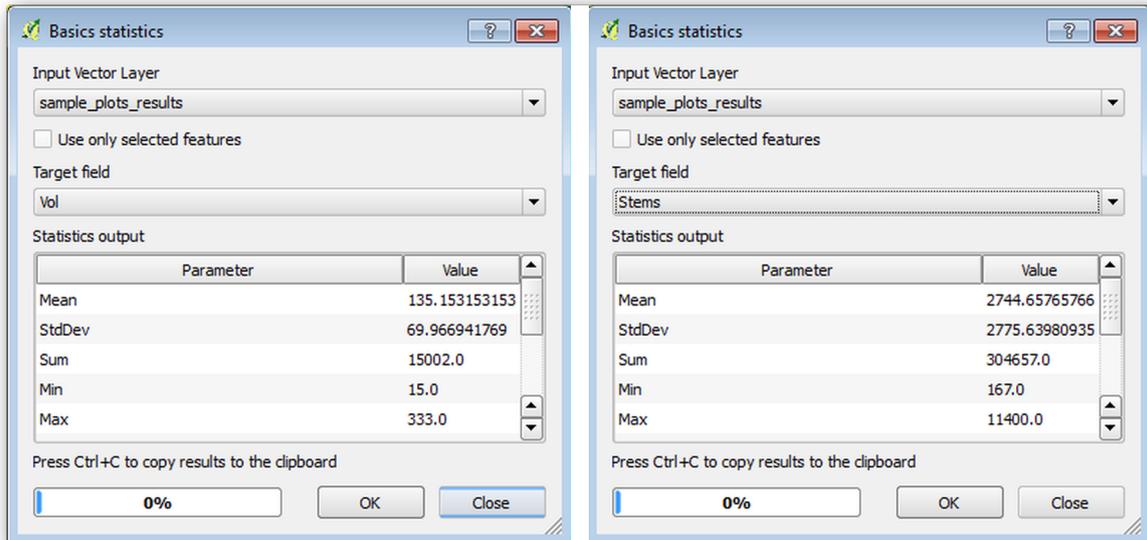
調査結果から、ヘクタール当たりの体積や茎数など、いくつかの興味深いパラメータのこの森林領域全体に対する平均値を計算できます。体系的なサンプルプロットは等しい面積を表しているため、sample_plots_results レイヤーからヘクタール当たり体積と茎数の平均値を直接に計算できます。

基本統計 ツールを使用して、ベクターレイヤ内のフィールドの平均値を計算できます。

- ベクタ ->分析ツール->基本統計 を開きます。
- sample_plots_results を入力ベクタレイヤとして選択します。
- ターゲット field として Vol を選択します。
- OK をクリックします。

森の中の平均体積は次のとおりです。135.2 立方メートル/ ha。

同じように茎数の平均を計算できます、2745 茎/ha。



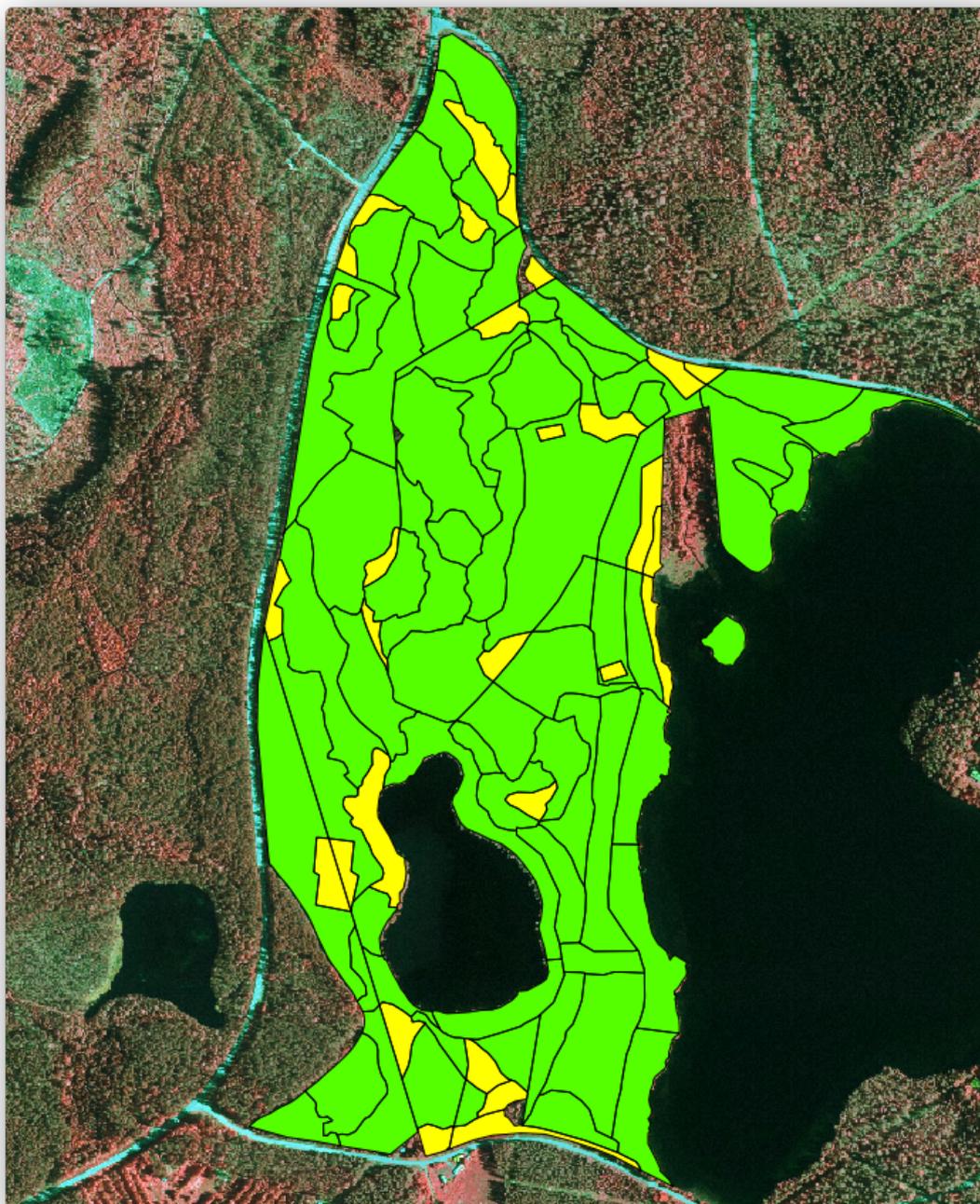
14.7.3 Follow Along: 林分のパラメータを推定する

別の森林は、以前にデジタル化された林分のための推定値を算出するために、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、任意のサンプルプロットを取得していないと、それらのために情報を取得することはありません。野外チームは、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していたように、体系的な調査を計画するときには、いくつかの余分なサンプルプロットを計画している可能性があります。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できるかもしれません。それにもかかわらず、計画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。

必要なのは林分のそれぞれの中に落ちてい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組み合わせたいときは、空間的結合を行います：

- ベクトル->データ管理->位置により属性を結合 ツールを開きます。
- ターゲット ベクタレイヤとして forest_stands_2012 を設定します。それに対する結果が欲しいレイヤです。
- ベクタレイヤを結合として sample_plots_results を設定します。それから推定値を計算したいレイヤです。
- 交わる地物を集計 をチェックします。
- 平均 だけ計算するようチェックしてください。
- 結果に forest_stands_2012_results.shp と名前をつけ、それを exercise_data\林業\結果\フォルダに保存します。
- 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。
- OK をクリックします。
- 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ーを追加し受け入れます。
- 位置によって属性を結合 ツールを閉じます。

forest_stands_2012_results の属性テーブルを開き、得た結果を確認します。林分 の数が計算のための値として NULL を持っていることに注意してください、これらはサンプルプロットを有していないものです。それらすべてを選択し、地図でそれらを見直し、それらはより小さな林分のいくつか：



前に行ったように、今森林全体の同じ平均値を計算しましょう、今回は計算のためのベースとして林分のために得た平均値を使用するだけでできます。80 × 80m 直前の状況で、各サンプルプロットは、理論上の林分を表現することを忘れないでください。今、個別の代わりに林分のそれぞれの面積を考慮しなければなりません。その方法は、再び、例えば、体積の M3/ha であるパラメータの平均値は、林分の総体積に変換されます。

最初に林分対する面積を計算し、それからそれらのそれぞれに対する総量と茎数を計算する必要があります。

- 属性テーブル 中で編集を有効にします。
- フィールド *calculator* を開きます。
- *area* と呼ばれる新しいフィールドを作成します。
- 出力フィールド *type* は 十進数 (実数) のままにしておきます。

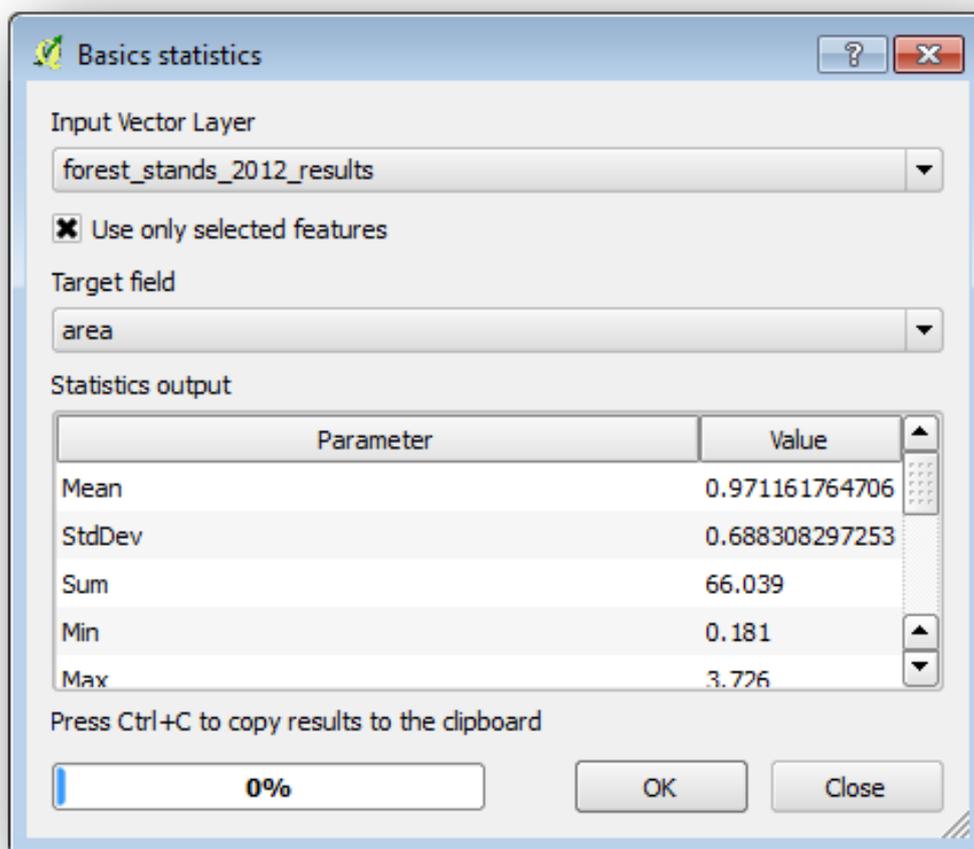
- *Precision* を 2 に設定してください。
- *Expression* ボックスに \$面積/10000 と書きます。これは林分の面積をヘクタールで計算します。
- *OK* をクリックします。

今、すべての林分のための全体積および推定茎の数を持つフィールドを計算します。

- *s_vol* と *s_stem* フィールドに名前を付けます。
- フィールドは整数でもよいし、実数も使用できます。
- 式 "エリア"*"MEANVol" と "エリア"*"MEANStems" をそれぞれ総体積と合計茎に対して使用します。
- 終了したら、編集内容を保存します。
- 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るのに十分でした。今すぐ見積もりを計算するには、林分の体積または茎数の合計を分割するために必要な情報を含む林分の面積の合計によって。

- *forest_stands_2012_results* レイヤーの属性 *table* で、情報を含むすべての林分を選択します。
- ベクタ ->分析ツール->基本統計 を開きます。
- 入力ベクトルレイヤー として *forest_stands_2012_results* を選択します。
- *area* を *Target field* として選択します。
- 選択地物のみ使用 をチェックしてください
- *OK* をクリックします。



おわかりのように、林分面積の総和は 66.04 ha です。行方不明の林分の面積は約 7 ha だけであることを注意してください。

同様に、これらの林分の合計体積が 8908 立方メートル/ ha であり、茎の総数は 179594 stems であることを計算できます。

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 立方メートル/ ha と
- 2719 stems/ha。

QGIS プロジェクトを保存します、forest_inventory.qgs。

14.7.4 In Conclusion

最初は森林の特性を考慮することなく、また林分への空撮画像の解釈を使用して、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、全体の森林のための森林の推定値の計算をすることができました。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 What's Next?

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でしょう。

14.8 Lesson: 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ーを作成します。

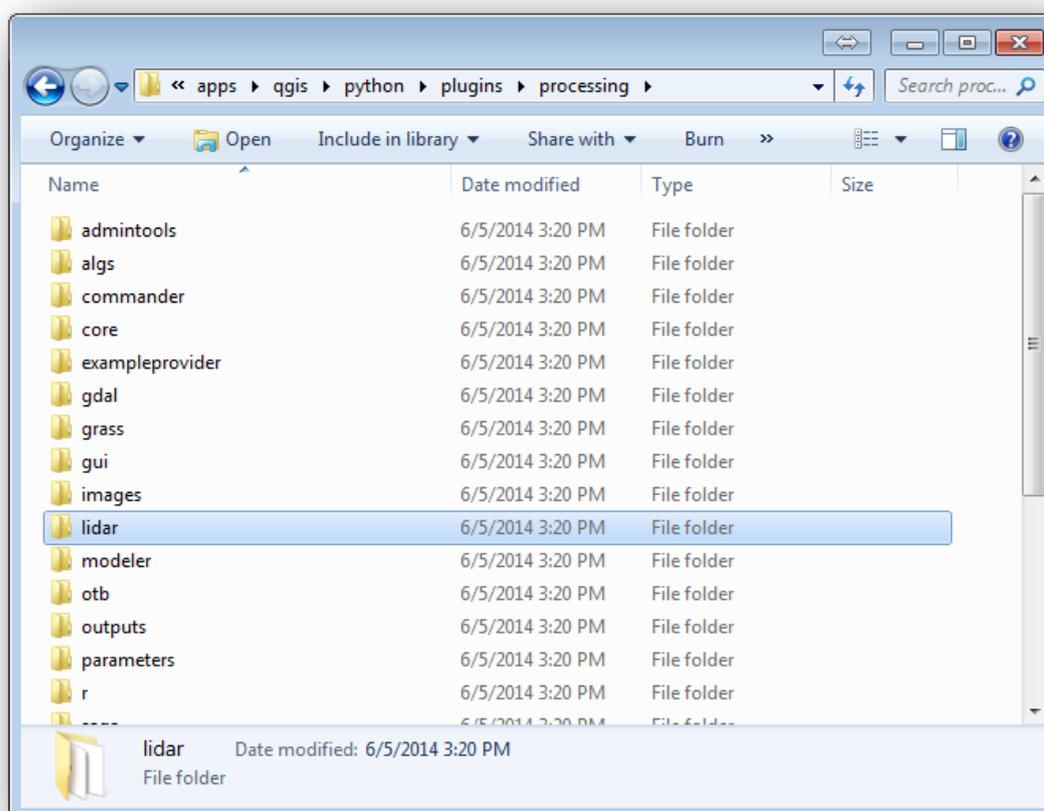
このレッスンの目標： LAStools をインストールし、レーザー測量データと陰影起伏ラスターから DEM を計算します。

14.8.1 Follow Along: Lastools のインストール

QGIS 内レーザー測量データを管理する処理フレームワークと *LAStools* <<http://rapidlasso.com/2013/09/29/how-to-install-lastools-toolbox-in-qgis>> によって提供されるアルゴリズムを使用して可能です。

レーザー測量点群からデジタル標高モデル (DEM) を取得して、プレゼンテーション目的のために、視覚的にはより直感的である陰影起伏ラスターを作成できます。最初に、*Processing* 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。

- すでにそれを開始している場合は、QGIS を閉じます。
- 古いライダープラグインは、フォルダ `C:/Program Files/QGISValmiera/apps/qgis/python/plugins/pro` 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。
- 名前 `lidar` のフォルダを持っている場合、それを削除します。これは、QGIS 2.2 および 2.4 の一部のインストールのために有効です。



- `exercise_data\林業\ライダー\` フォルダと、そこにファイル `QGIS_2_2_toolbox.zip` が見つけられます。それを開き `lidar` フォルダを解凍して、ちょうど削除したものと置き換えます。
- 別の QGIS のバージョンを使用している場合は、このチュートリアル <http://rapidlasso.com/2013/09/29/how-to-install-lastools-toolbox-in-qgis/> 中のより多くのインストール手順を参照できます。

今コンピュータに LAStools を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 <http://lastools.org/download/lastools.zip> で取得し、お使いのシステム内のフォルダ、例えば `:kbd:'c:\lastools\` に `lastools.zip` ファイルの内容を解凍します。lastools フォルダのパスにはスペースや特殊文字は使用できません。

ノート: lastools フォルダ内部の `LICENSE.txt` ファイルをお読みください。LAStools の一部はオープンソースであり、他は、クローズドソースであり、ほとんどの商用および政府の使用のライセンスが必要です。教育と評価の目的のためには、必要なだけ LAStools を使用してテストできます。

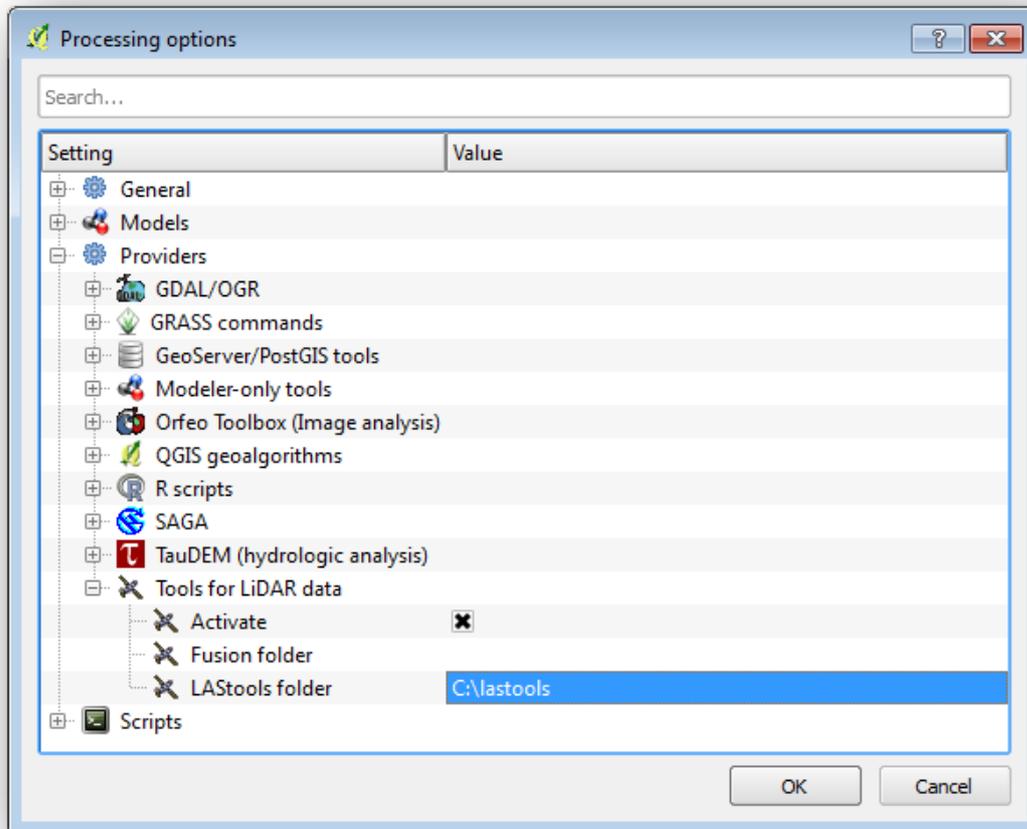
プラグインと実際のアルゴリズムは、今お使いのコンピュータにインストールして使用することはほとんど準備ができている、それらの使用を開始するために処理フレームワークを設定する必要があるだけです。

- QGIS で新しいプロジェクトを開きます。
- プロジェクトの CRS を ETRS89 / ETRS-TM35FIN に設定してください。
- プロジェクトを `forest_lidar.qgs` として保存します。

QGIS で LAStools 設定するには：

- 処理 -> オプションと `configuration` に行きます。
- 処理 `options` ダイアログで、プロバイダに、その後 `LiDAR data` のためのツールに行きます。

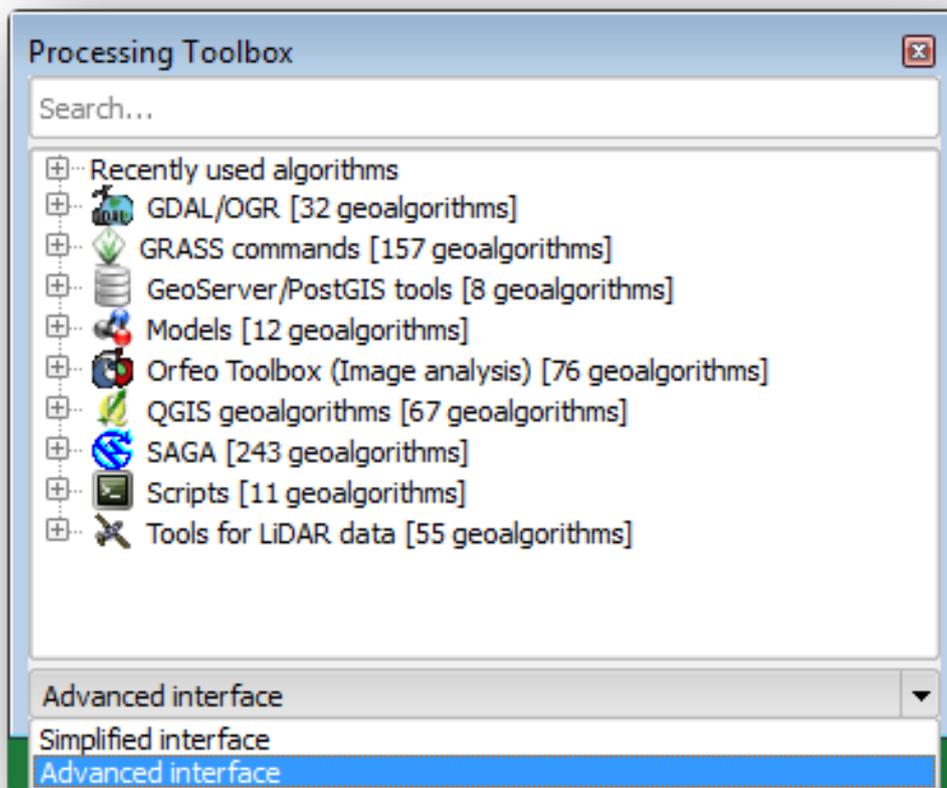
- *Activate* をチェックします。
- *LAStools* フォルダ について `:kbd:`C:\lastools\`` (または *LAStools* を展開したフォルダ) を設定します。



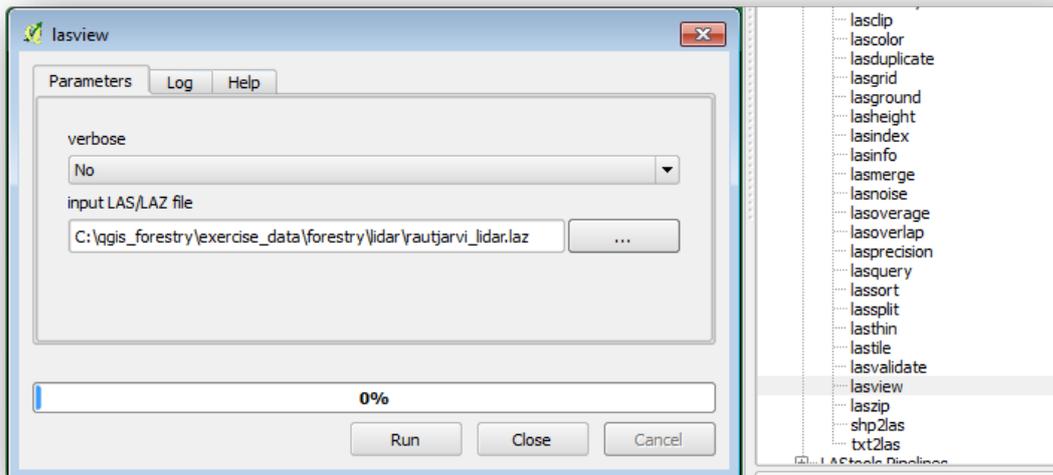
14.8.2 Follow Along: LAStools と DEM の計算

いくつかの SAGA アルゴリズムを実行するために、`../ vector_analysis/spatial_statistics` 中の *Processing* ツールボックスをすでに使用してきました。今、*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。

- 処理->*Toolbox* 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ース を選択します。
- レーザー測量 *data* カテゴリのツール、表示されるはずですが。

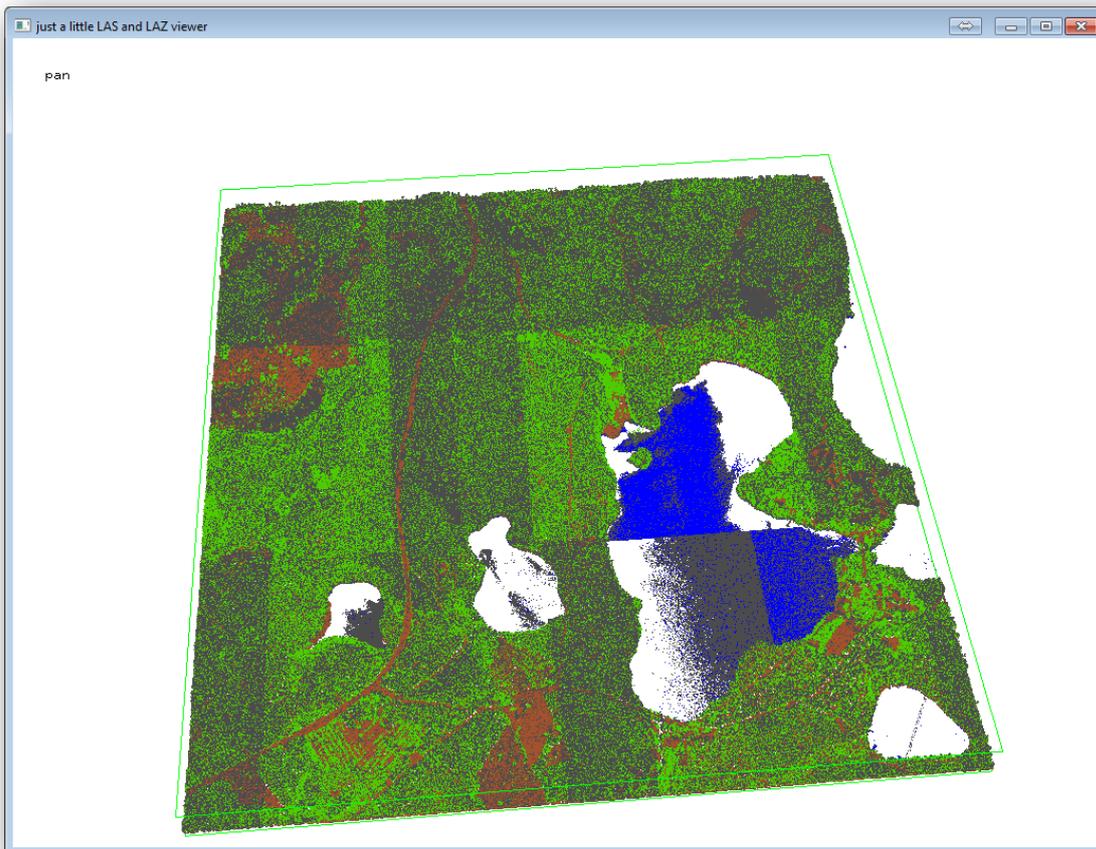


- 利用可能なツールが見えるようにそれを展開し、さらに *LAStools* カテゴリを展開します（アルゴリズムの数は変化する場合があります）。
- *lasview* アルゴリズムが見つかるまで下にスクロールし、見つかったらダブルクリックして開きます。
- 入力 LAS / LAZ の file で *exercise_data*\林業\ライダー\ を参照し、*rautjarvi_lidar.laz* ファイルを選択します。



- *Run* をクリックします。

今、ほんの少し LAS と LAZ ダイアログウィンドウ *viewer* 内に LiDAR データを見ることができます：



そこには、このビューア内行うことができます多くのものがありますが、今それがどのように見えるかを確認するための LiDAR 点群をパンするビューアをクリックしてドラッグできます。

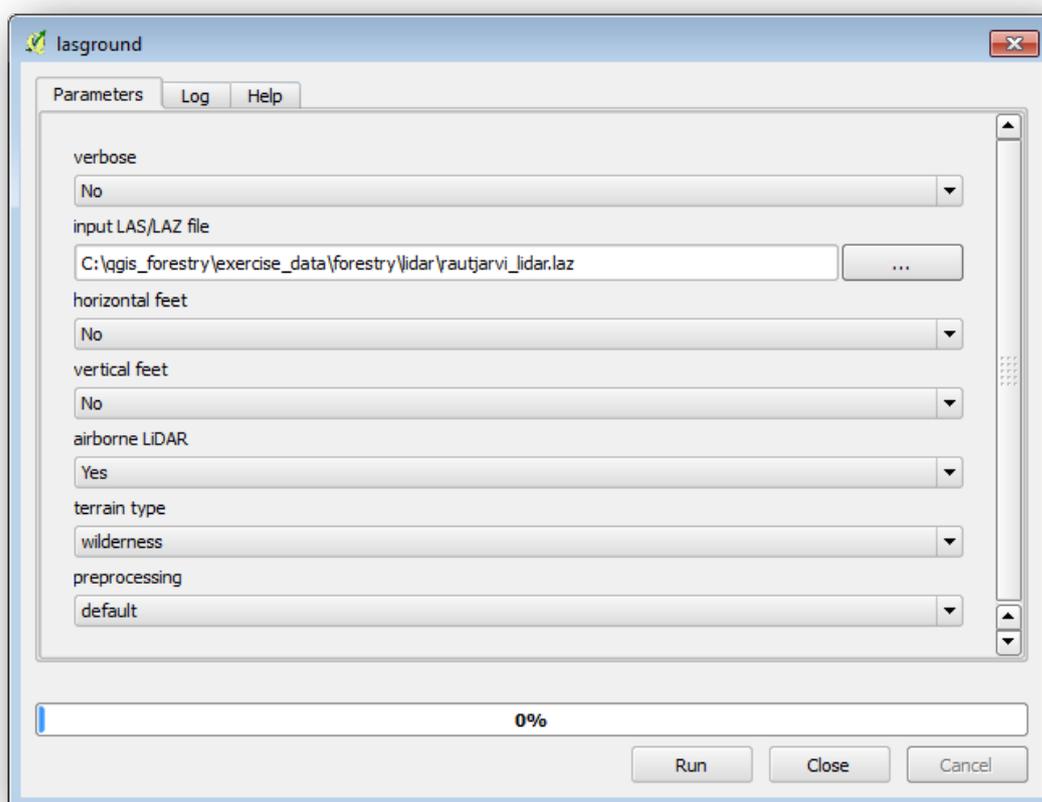
ノート: LAStools がどのように動作するかについての詳細をお知りになりたい場合は、ツールのそれぞれ

について `c:\lastools\bin\` フォルダ内の README テキストファイルを読むことができます。チュートリアルと他の材料は、'[Rapidlasso ウェブページ<http://rapidlasso.com/>](http://rapidlasso.com/)' でご利用いただけます。

- 準備ができたら、ビューアを閉じます。

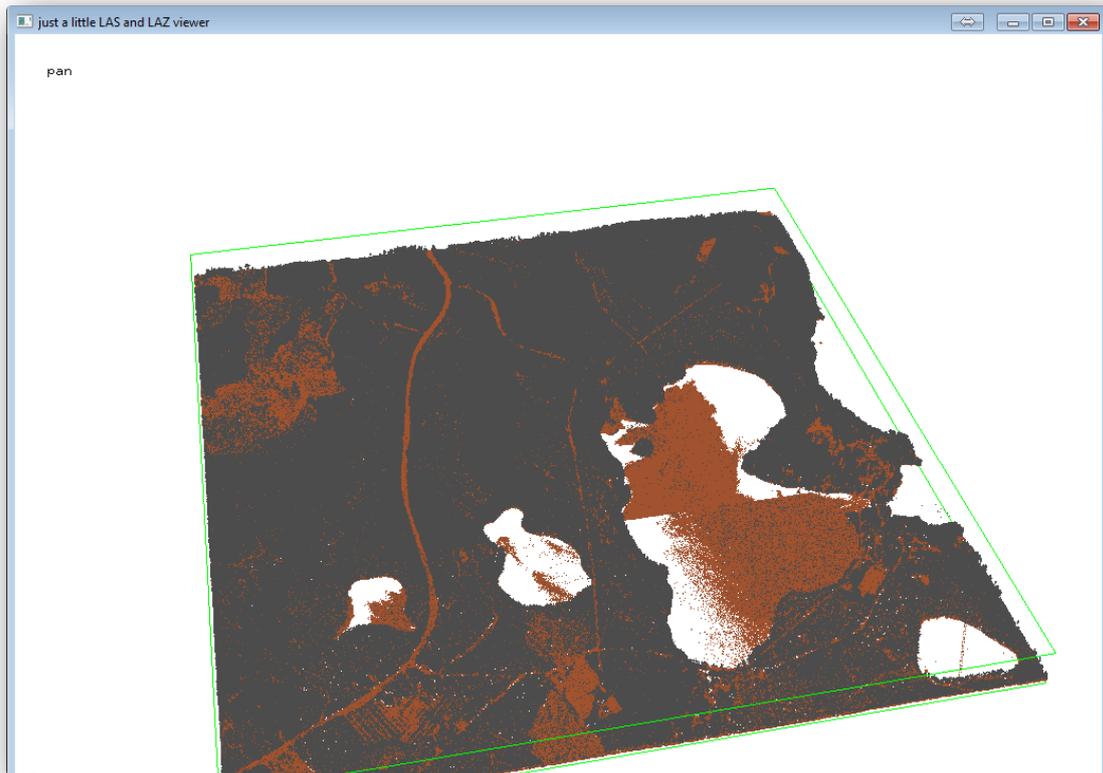
LASTools と DEM の作成は 2 段階で行うことができます。第 1 は点群を ground と NO ground 点に分類すること、次は ground 点のみ使用して DEM を計算すること。

- 処理 *Toolbox* に戻ります。
- 検索... ボックス注意して、`lasground` と書いてください。
- 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：



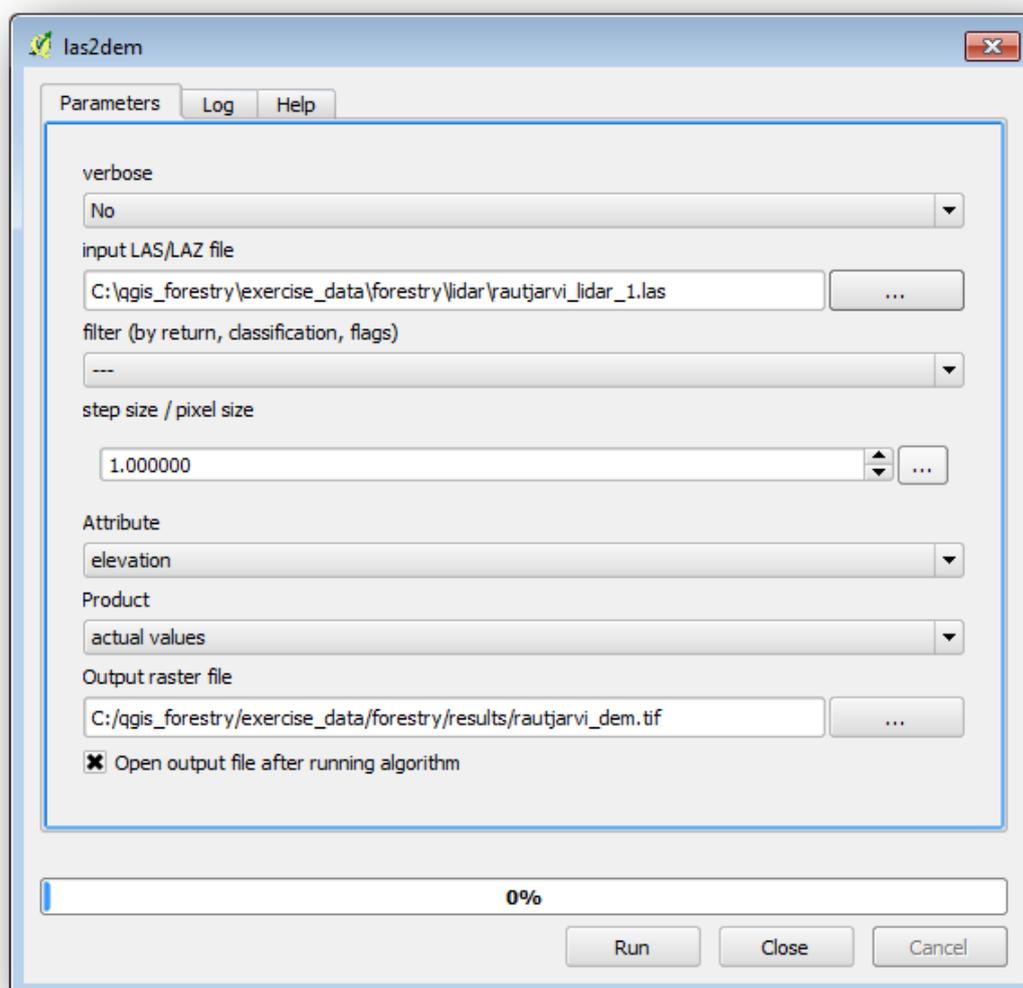
- 出力ファイルは `rautjarvi_lidar.laz` が配置されているのと同じフォルダに保存され、`rautjarvi_lidar_1.las` と名前が付けられます。

それを確認したい場合は、`lasview` で開くことができます。



茶色の点が基準に分類される点で、灰色のものはそれ以外です。文字 `g` をクリックして地上基準点だけを視覚化したり、文字 `u` で分類されていない点だけを見ることができます。文字 `a` をクリックすると再びすべての点を見ることができます。コマンドについてこれ以上は `lasview_README.txt` ファイルを確認してください。興味のある方は、レーザー測量ポイントを手動で編集することについてのこのチュートリアル<<http://www.rapidlasso.com/2014/03/02/tutorial-manual-lidar-editing/>> も、ビューアの中に別の操作を表示します。

- 再びビューアを閉じます。
- 処理 *Toolbox* で:kbd:‘ las2dem‘ を検索します。
- *las2dem* ツールを開き、この画像に示すように設定します。



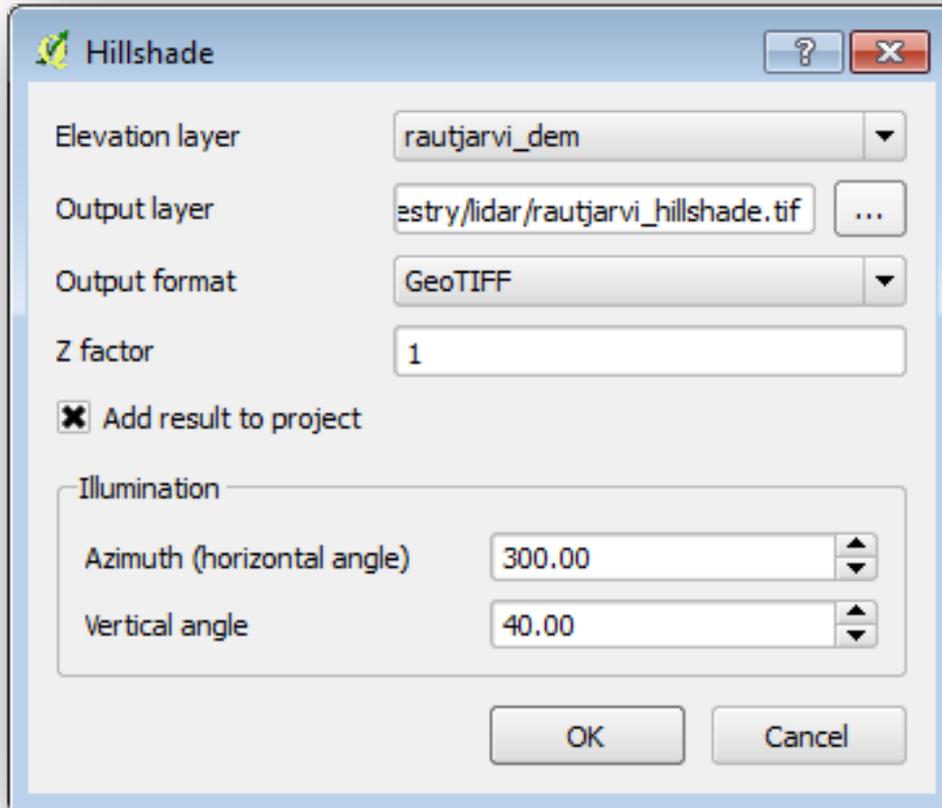
結果 DEM は、出力ラスタファイル という一般的な名前で地図に追加されます。

ノート: *lasground* と `guiabel:' 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、無免許で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 Follow Along: 地形陰影起伏を作成します

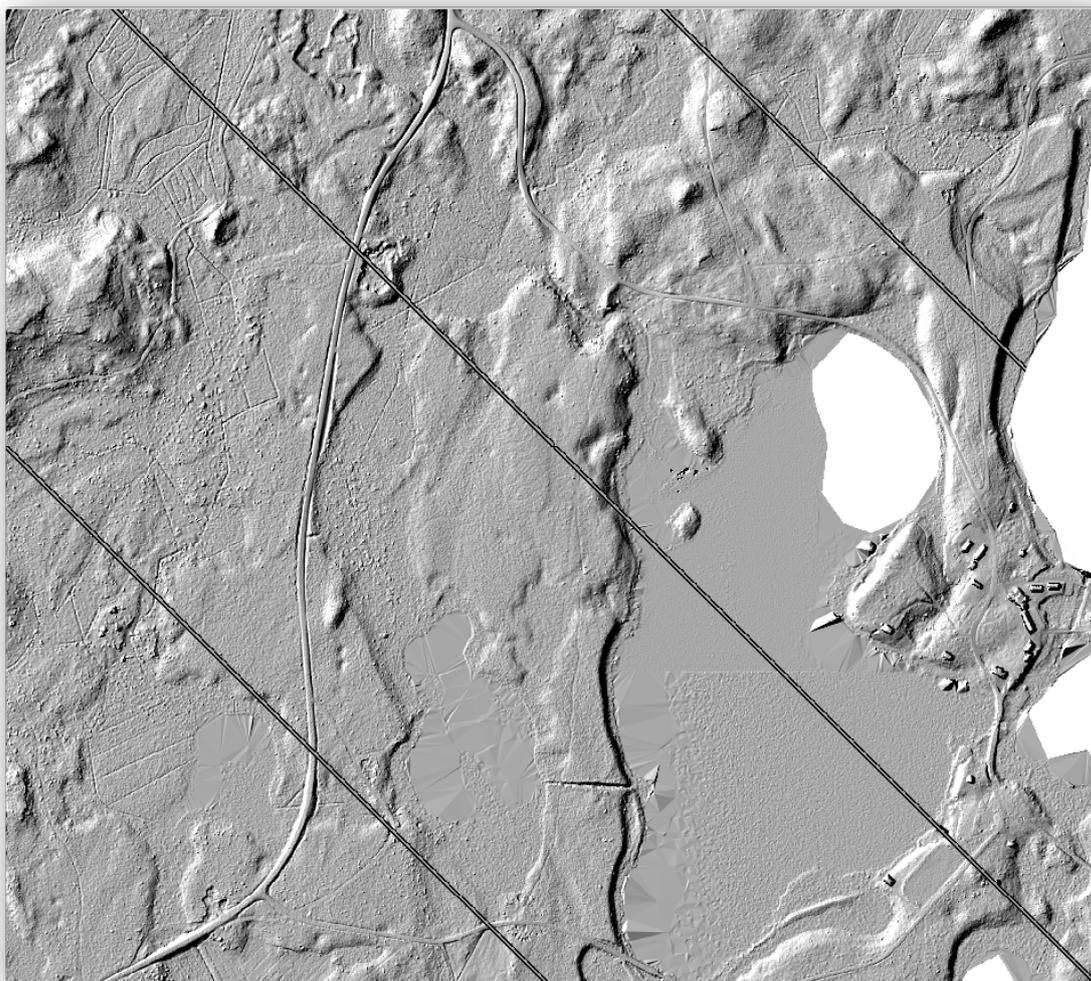
可視化の目的のために、DEM から生成された陰影起伏は、地形のより良い可視化を提供します：

- ラスター->地形分析->*Hillshade* を開きます。
- 出力 *layer* として:`kbd:exercise_data\林業\ライダー\`を参照し、ファイルに:`kbd:' hillshade.tif'` と名前を付けます。
- デフォルトの設定でパラメータの残りの部分を残します。



- プロンプトが表示されたら ETRS89 / ETRS-TM35FIN を CRS として選択します。

陰影起伏ラスタ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 In Conclusion

レーザー測量データを使用して DEM を取得することは、特に森林地域では、あまり労力なしに良好な結果を与えます。準備ができたレーザー測量派生の DEM や、SRTM の 9 メートルの解像度の DEM <<http://www.cgiar-csi.org/data/srtm-90m-digital-elevation-database-v4-1>>のような他のソースも使用できます。いずれにせよ、それらを使用して、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できます。

14.8.5 What's Next?

このモジュールの次の、そして最後のステップ、レッスンでは、結果の地図のプレゼンテーションを作成するために、陰影起伏ラスタと森林目録の結果を使用します。

14.9 Lesson: 地図プレゼンテーション

現在の状況にそれを更新し、GIS プロジェクトとして古い森林調査をインポートした前のレッスンでは、フィールドワークやフィールド測定から算出された森林のパラメータの地図を作成し、森林調査を設計しました。

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で、特定の数字を見なくても、誰でも容易に結果が何かが一目でわかるようになるでしょう。

このレッスンの目標： 調査結果を提示する地図を、陰影起伏ラスタを背景に使用して作成します。

14.9.1 Follow Along: 地図データを準備する

パラメーター計算のレッスンからの QGIS プロジェクトを開きます、 `forest_inventory.qgs`。少なくとも以下のレイヤーを保管してください：

- `forest_stands_2012_results`.
- `basic_map`.
- `rautjarvi_aerial`.
- `lakes` (それを持っていない場合は、`exercise_data \林業\フォルダ`からそれを追加します)

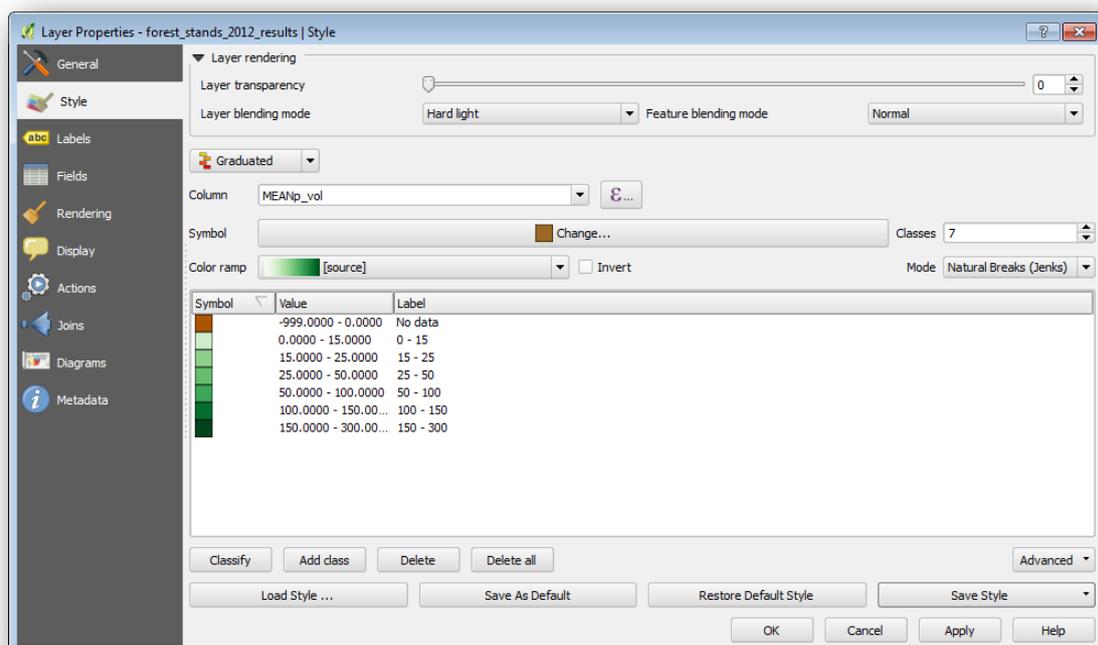
地図で 林分の平均体積を提示しようとしています。 `forest_stands_2012_results` レイヤの属性テーブルを開いた場合、見ることができます NULL 値を情報なしで 林分のために。例えば、に NULL 値 -999 は、それらの負の数は、それらのポリゴンのデータが存在しないことを意味することを知ってスタイリングにも、これらの 林分を取得できるようにするには、変更する必要があります。

`forest_stands_2012_results` レイヤに対して：

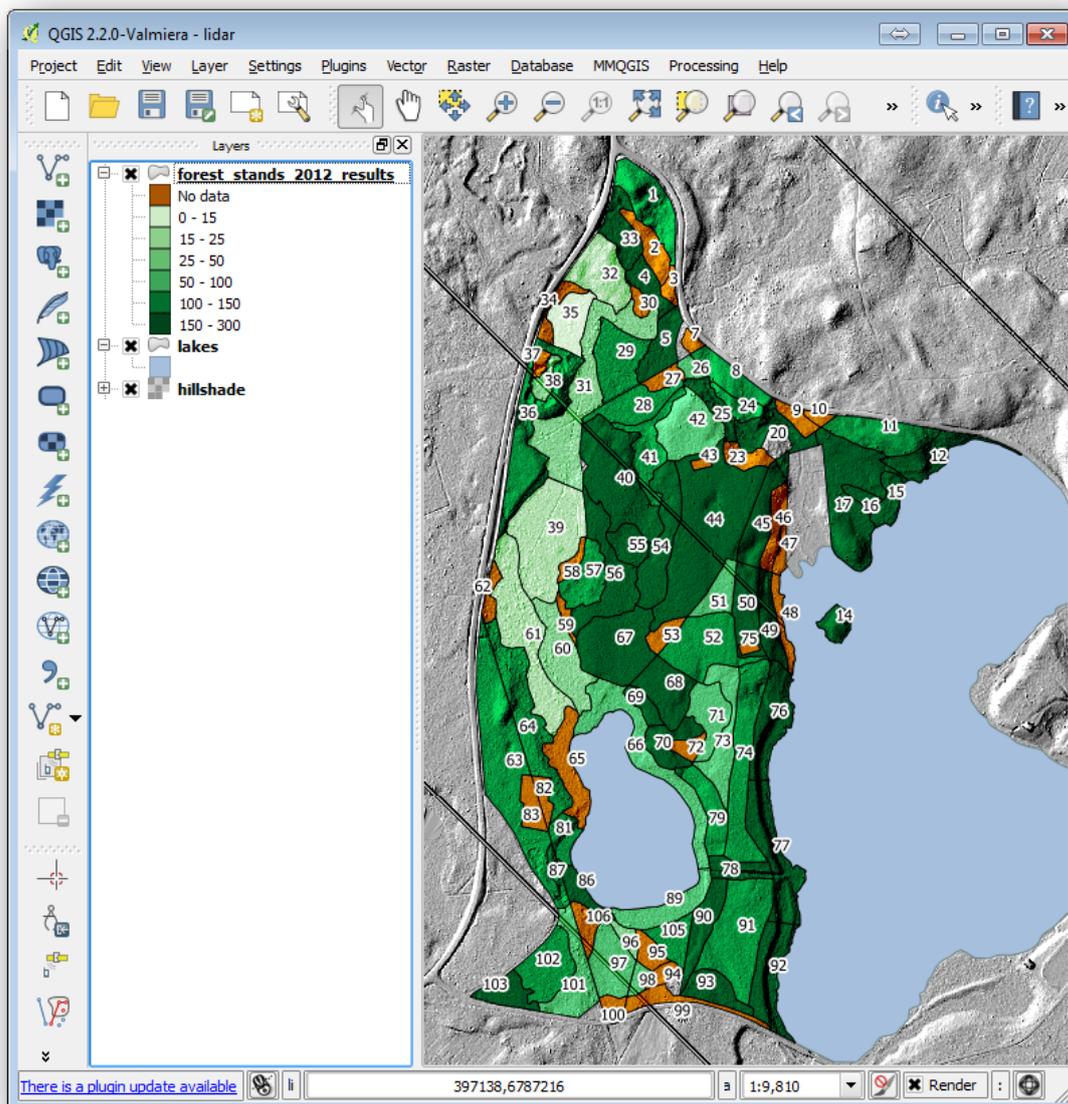
- 属性テーブルを開き編集を可能にします。
- NULL 値を持つポリゴンを選択します。
- 選択された地物だけに対して、計算機を使用して `MEANVol` フィールドの値を -999 に更新します。
- 編集を無効にし、変更を保存します。

今、保存されたスタイルをこのレイヤーに使用できます。

- `Style` タブに行きます。
- `Load Style` をクリックします。
- `exercise_data \林業\結果\フォルダ`から `:kbd:forest_stands_2012_results.qml` を選択します。
- `OK` をクリックします。

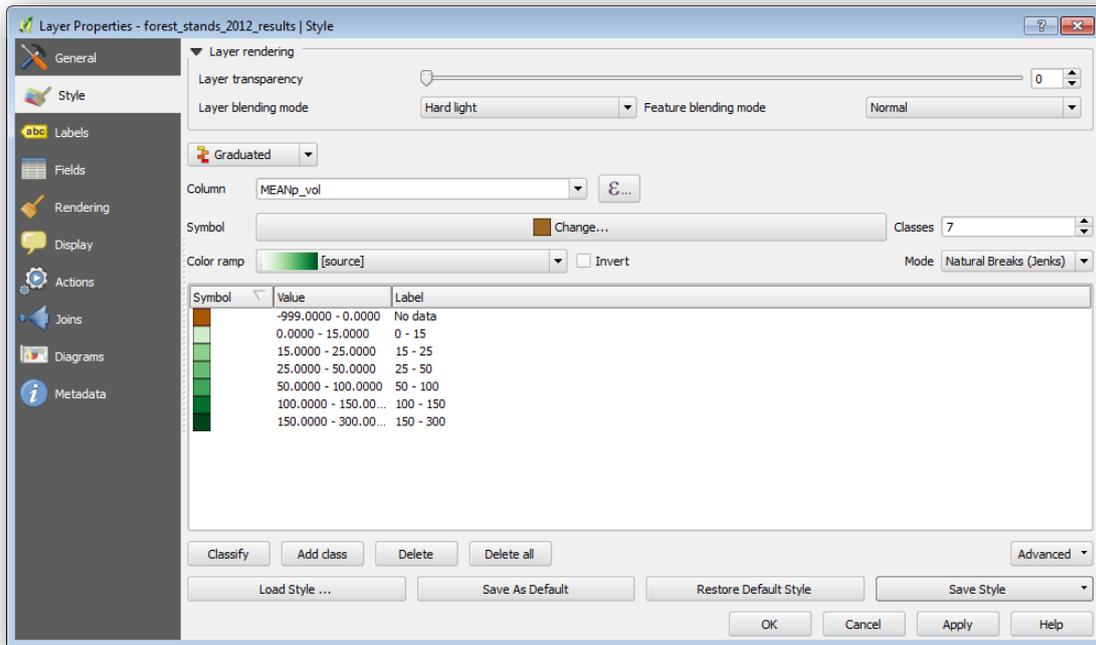


地図は次のようになります。



14.9.2 Try Yourself **さまざまなブレンドモードを試す**

ロードしたスタイル:

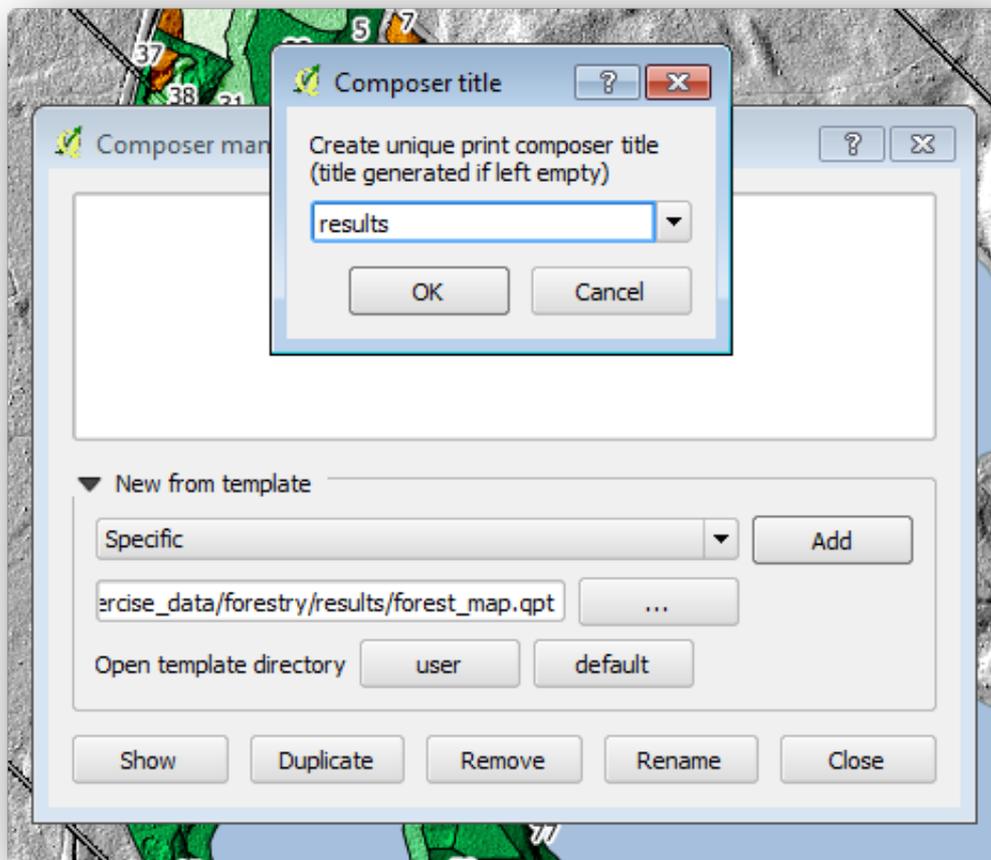


ハード light モードには レイヤブレンドモード が使用されます。異なるモードは、この場合には、陰影起伏ラスタの基礎となると、上にあるレイヤを組み合わせると異なるフィルタを適用し、林分が使用されていることに注意してください。これらのモードについては ‘[ユーザガイド <http://docs.qgis.org/2.2/en/docs/user_manual/working_with_vector/vector_properties.html>](http://docs.qgis.org/2.2/en/docs/user_manual/working_with_vector/vector_properties.html)’ で読むことができます。

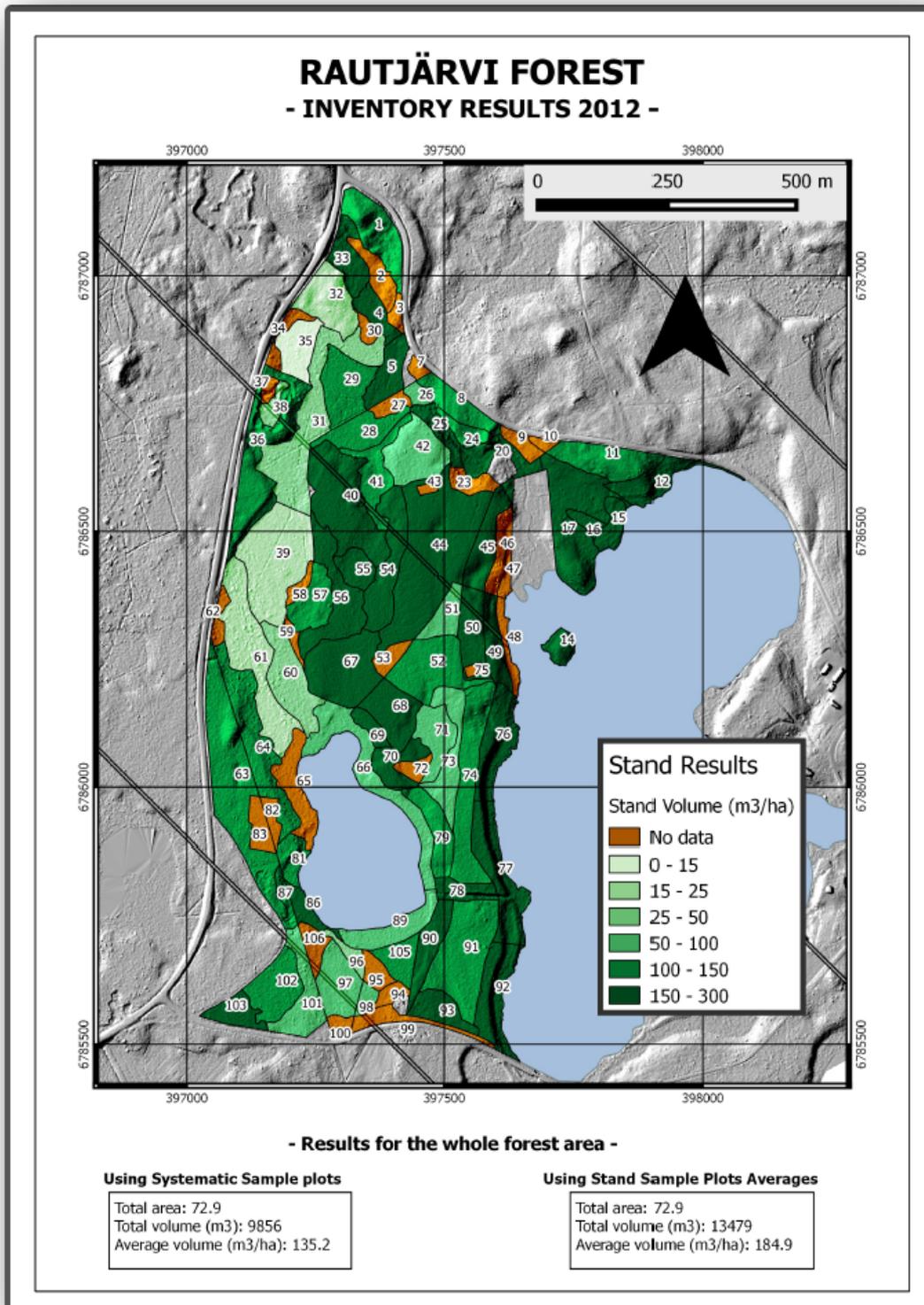
異なるモードで試してみて、地図の違いを参照してください。そして、最終的な地図に好きな方いずれかを選択してください。

14.9.3 Try Yourself 地図の結果を作成するためにコンポーザテンプレートを使用する

結果を提示するために予め準備したテンプレートを使用してください。テンプレート forest_map.qpt は exercise_data\林業\結果\フォルダに位置しています。プロジェクト->作曲マネージャ... ダイアログを使用して、それをロードします。



地図コンポーザを開き、満足な結果を得るために、最終的な地図を編集します。
 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 In Conclusion

このモジュールを介して、基本的な森林調査が計画され、QGIS で提示され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、このマニュアルが必要な特定の結果をどうしたら達成できるか探るための良い出発点になってくれるのではないかと。

Chapter 15

Module: 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 Lesson: 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 テーブル

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では、テーブルは、縦の列（名前で識別される）と横の行のモデルを使用して構成されたデータ要素（値）のセットです。テーブルには指定された数の列がありますが、任意の数の行を持つことができます。各行は、候補キーとして識別された特定の列部分に現れる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る。

15.1.3 列/フィールド

列は、特定の単純型のデータ値の集合であり、表の各行に対して1つです。列は、行を構成する構造を提供します。フィールドという用語は、多くの場合、フィールド（またはフィールド値）を使用して、1つの行と1つの列の交差点に存在する単一項目を具体的に参照する方が正しいと考えられますが、列と互換的に使用されます。 -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*

データ型の多くの種類があります。のが最も一般的に焦点を当ててみましょう：

- 文字列 - 自由形式のテキストデータを格納します
- 整数 - 整数を格納します
- 実数 - 小数を保存します
- 日付 - 誰も忘れないよう、ホルストの誕生日を格納します
- ブール - シンプルな真/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。フィールドに何も無い場合、フィールドコンテンツは**'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)
```

使用できるデータ型はもっと多くあります - [PostgreSQL のマニュアルをご確認ください!](http://www.postgresql.org/docs/current/static/datatype.html) <http://www.postgresql.org/docs/current/static/datatype.html> <_

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用してみましょう。住所のデータベースを作成したいとします。

Try Yourself

簡単な住所を構成し、データベースに格納したいプロパティを書き留めます。

[結果をチェック](#)

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。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)

同じ名前の通りや都市の友達が多いとしましょう。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、都市名が変更された場合は、データベースを更新するために多くの作業を行う必要があります。

15.1.9 Try Yourself

重複を低減し、データ構造を正規化するために、上記の理論 ‘people’ テーブルを再設計します。
 データベースの正規化の詳細についてはこちらをご覧ください<http://en.wikipedia.org/wiki/Database_normalization>
 -
[結果をチェック](#)

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いて概念の説明を探してるが、その教科書には索引がなかったとします。必要な情報が見つかるまで、1つの表紙から読み始め、書籍全体を通して作業を進めなければなりません。ブックの裏にある索引は、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)
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 連番

連番は、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 は連番です - 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 エンティティ関係作図

正規化されたデータベースでは、通常、多くの関係 (テーブル) があります。エンティティ関係図 (ER 図) は、関係間の論理依存関係を設計するために使用されます。正規化されていない ‘people’ テーブルをレッスンの前半から考えてみましょう:

```
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)
```

その後、「キー」:kbd:streets.id と:kbd:' people.streets_id' を使用して2つのテーブルをリンクできます。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ダイアグラムは、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あることを示しています。

Try Yourself

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。

結果をチェック

15.1.13 制約、主キーと外部キー

リレーション内のデータがモデラーのデータの格納方法と一致するように、データベースの制約が使用されます。たとえば、郵便番号の制約により、数字が:kbd:1000 と:kbd:9999 の間に入ることが保証されます。

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という連番です。

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・ダイアグラムでは、テーブル間の結合は、通常、プライマリキーにリンクする外部キーに基づいています。

人の例を見てみると、テーブルの定義によれば、street カラムはストリートテーブルのプライマリ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 |
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

プロセス中に何か問題（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 In Conclusion

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 What's Next?

今、データベースが理論的にはどのように動作するかを見てきましたので、カバーしてきた理論を実装するための新しいデータベースを作成してみましょう。

15.2 Lesson: データモデルの実装

私たちはすべての理論をカバーしたので新しいデータベースを作成してみましょう。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

ノート: このドキュメントの範囲外ではありますが、Mac ユーザは [Homebrew](#) を用いて PostgreSQL をインストールすることができます。Windows ユーザはここにあるグラフィカルなインストーラを使うことができます: <http://www.postgresql.org/download/windows/> このドキュメントでは Ubuntu で QGIS を動作させているユーザを想定しています。

Ubuntu で:

```
sudo apt-get install postgresql-9.1
```

このようなメッセージを取得するはず:

```
[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 and 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 には非常に良い [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 ユーザになり、新しいデータベースユーザを作成します:

```
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 ユーザでの bash プロンプトでデータベースユーザを作成します。ユーザ名は 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が:

```
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。bash シェルプロンプトから実行します:

```
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が:

```
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
```

‘ここで<http://www.postgresql.org/docs/9.0/html/postgresql_cheat_sheet.html> オンラインで
利用可能 - ‘PSQL はカンニングペーパー’<http://www.postgresql.org/docs/9.0/html/postgresql_cheat_sheet.html>
‘_も参照してください。PDF> ‘_。

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 データベースに接続します:

```
psql address
```

streets テーブルを作成します:

```
create table streets (id serial not null primary key, name varchar(50));
```

serial と varchar はデータ型です。serial は新しいレコードのそれぞれに id を自動的に設定するために PostgreSQL に整数シーケンス (自動番号割り当て) を開始させます。varchar(50) は PostgreSQL に長さ 50 文字の文字列フィールドを作成させます。

コマンドが ; で終わっていることに気づきましたか。すべての SQL コマンドはこのように終わるべきです。Enter キーを押すと psql は次のように報告します:

```
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
```

streets.id を使用する主キー streets_pkey を持つテーブルが正しく作成されました。

注: ; を入力せずに Enter キーを押すと address-# のようなプロンプトが表示されます。PG はさらなる入力を期待しています。コマンドを実行するには ; を入力して下さい。

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。

Try Yourself

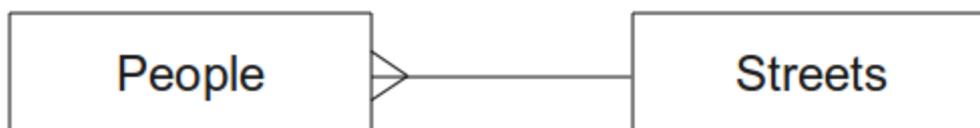
上記のアプローチを使用して people というテーブルを作成します:

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。

[結果をチェックする](#)

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);
```

`people` テーブルの `street_id` フィールドは `streets` テーブルの有効な街路 `id` と一致しなければならないことを指示します。

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` テーブルの `name` 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

テーブルを取り除きたい場合は `drop` コマンドを使用します:

```
drop table streets;
```

ノート: 現在の例では上記のコマンドは動作しないでしょう。なぜでしょうか。 [こちら](#) を参照して下さい。

`people` テーブルに同じ `drop table` コマンドを使う場合は正しく削除されるでしょう:

```
drop table people;
```

ノート: 実際にそのコマンドを入力して `people` テーブルを削除した場合は、再度作成して下さい。次の演習で必要になります。

15.2.11 pgAdmin III 上の単語

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。pgAdminIII 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-get 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 In Conclusion

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 What's Next?

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 Lesson: モデルへのデータの追加

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移植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

どのようにテーブルにデータを追加しますか？SQL の INSERT 文は、このための機能を提供します：

```
insert into streets (name) values ('High street');
```

注意すべき物事のカップル：

- テーブル名 (街路) の後に、移したい列の名前を列挙します (この場合は名前列のみ) 。
- values キーワードの後にフィールドの値のリストを置きます。
- 文字列は単一引用符で囲む必要があります。
- id 列には値を挿入していないことに注意。それは連番であり自動生成されるためです。
- id を手動で設定すると、データベースの整合性に深刻な問題を引き起こす可能性があります。

成功した場合 0 1 を挿入 と表示されるはずで

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)
```

Try Yourself

INSERT コマンドを使用して 街路 テーブルに新しい道路を追加してください。

結果を確認します

15.3.2 制約によるとシーケンシングデータの追加

15.3.3 Try Yourself

人物オブジェクトを以下の詳細を持つ people テーブルに追加してみましょう：

```
Name: Joe Smith
House Number: 55
Street: Main Street
Phone: 072 882 33 21
```

ノート：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。

この時点では、streets 表にあるメインストリートのレコードを最初に作成せずにこれを実行しようとすると、エラーレポートが出ているはずで

気づいているはずで

- その名前を使用して街路を追加できません
- 最初に街路テーブルに街路レコードを作成しないと、街路:kbd:id を使用して街路を追加できません

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。

結果を確認する

15.3.4 データを選択します

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います

15.3.5 更新データ

いくつかの既存のデータに変更を加えたい場合は? 例えば、街路名が変更される:

```
update streets set name='New Main Road' where name='Main Road';
```

このような更新ステートメントを使用するときは非常に注意してください - WHERE 句に複数のレコードが一致した場合、それらはすべて更新されます!

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;
```

それは UPDATE 1 を返す必要があります。

ノート: WHERE 文の基準は大文字と小文字が区別されます。Main Road は Main road と同じではありません

15.3.6 データの削除

テーブルからオブジェクトを削除するために、DELETE コマンドを使用してください:

```
delete from people where name = 'Joe Smith';
```

今度は私達の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 Try Yourself

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 In Conclusion

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、変更する、および/またはそのデータを格納する新しいモデルを作成することを忘れないでください。

15.3.9 What's Next?

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 Lesson: 検索

:kbd:`SELECT ...` コマンドを書くとき、これは一般的にはクエリと言われますが、情報のデータベースに問い合わせしています。

****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。

ノート: 前のレッスンでそうしなかった場合は、以下の人々オブジェクトを:kbd:`people` テーブル追加します。外部キー制約に関連した何らかのエラーを受け取る場合は、まず街のテーブルに「主要道路」オブジェクトを追加する必要があります

```
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)
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

だいたいデータベース内のすべてのレコード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:kbd: `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;
```

name	house_no
Joe Bloggs	3
Roger Jones	33

(2 rows)

フィルタ (:kbd: `WHERE`句を使用して定義される) はソート (:kbd: `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)

ここでは、:kbd: `LIKE`句を使用し、:kbd: `s`を持つすべての名前を見つけてます。このクエリは大文字小文字が区別されることに気づくでしょう、だから:kbd: `Sally Norman`エントリは返されていません。

大文字小文字関係なく文字列を検索したい場合は、:kbd: `ILIKE`句を使用すれば大文字小文字を無視した検索ができます:

```
select name, house_no from people where name ilike '%r%';
```

name	house_no
Roger Jones	33
Sally Norman	83

(2 rows)

そのクエリは、:kbd: `r`か:kbd: `R`を自分の名前に持つ `people`オブジェクトすべてを返します。

15.4.3 結合

代わりに ID の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、あなたは、単一のクエリで 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;
```

ノート：結合によって、情報が由来する 2 つのテーブルは、この場合は人と通りですが、常に状態になります。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、人と街のすべての可能な組み合わせのリストを取得しますが、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！

ここでは、正しい出力がどのように見えるかです：

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み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 SELECT

サブ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)

それでは、このデータのサブ選択を表示してみましょう。:kbd: 'street_id'番号 kbd: '1' に住む人だけを表示したい:

```
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サブ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、人々オブジェクトが人々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)

```

ノート: Because we have not used an ORDER BY clause, the order of your results may not match what is shown here.

Try Yourself

通りの名前で人を要約し、実際の通りの名前の代わりに、street_ids を示しました。

: ref: 結果を確認します /

15.4.6 In Conclusion

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 What's Next?

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 Lesson: 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;
```

はじめの `create view roads_count_v as` の部分だけが異なります。そのビューからデータを選択することができます:

```
select * from roads_count_v;
```

結果:

```
count |      name
-----+-----
      1 | Main Road
      2 | High street
      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することができます:

```
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。)

ORDER BY 句を追加したのでビューの行はきれいに並べ替えられています:

```
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 In Conclusion

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 What's Next?

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 Lesson: ルール

ルールは“query tree”に書き換えることができます。一つの一般的な使用法は、更新可能なビューを含めて実装することです。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 Materialised ビュー (ルールのベースビュー)

people テーブルにある phone_no の変更すべてを people_log テーブルにログとして記録したいとする。その場合には新しいテーブルを設定する

```
create table people_log (name text, time timestamp default NOW());
```

次のステップでは、people テーブル内の phone_no の変更すべてを people_log テーブルにログとして記録するルールを作成

```
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;
```

people テーブルが正しく更新されたことを確認してください

```
select * from people where id=2;
```

```
id | name | house_no | street_id | phone_no
---+-----+-----+-----+-----
  2 | Joe Bloggs | 3 | 2 | 082 555 1234
(1 row)
```

今、作成したルールによって、people_log テーブルは次のようになります

```
select * from people_log;
```

```
name | time
-----+-----
Joe Bloggs | 2014-01-11 14:15:11.953141
(1 row)
```

ノート: `time` フィールドの値は、現在の日付と時刻に依存します。

15.6.2 In Conclusion

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することができます。

15.6.3 What's Next?

次のセクションは、これらのデータベースの概念を使い、GIS データに適用した PostGIS を使用する、空間データベースを紹介します。

Chapter 16

Module: 空間データベースの概念と PostGIS

空間データベースはデータベース内にジオメトリを格納できるようにするだけでなく、これらのジオメトリを用いてクエリを実行してレコードを取得する機能を提供します。このモジュールで私たちは PostgreSQL の拡張である PostGIS を使って空間データベースをセットアップし、シェープファイルからデータをデータベースへインポートし、PostGIS の地理的な関数を利用する方法を学びます。

このセクションで作業をしている間、あなたは [Boston GIS user group](#) から入手可能な PostGIS cheat sheet のコピーを持っていても構いません。もう一つの有用なリソースは [オンライン PostGIS ドキュメント](#) です。

Boundless Geo から 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も入手可能です:

- [Introduction to PostGIS](#)
- [Spatial Database Tips and Tricks](#)

[PostGIS online](#) も参照して下さい。

16.1 Lesson: 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。

ノート: ここでは PostGIS バージョン 2.1 を使用を前提としています。インストールとデータベースの設定方法は以前のバージョンと異なりますが、この資料の残りの部分はまだ有効です。インストールおよびデータベースの設定については、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-get install postgis
$ sudo apt-get install postgresql-9.1-postgis
```

本当に簡単です...

ノート: 使用している Ubuntu のバージョンや設定されているリポジトリによって、PostGIS 1.5 もしくは 2.x がインストールされます。psql や他のツールで `select PostGIS_full_version();` クエリを発行すれば、インストールされたバージョンを知ることができます。

PostGIS 最新版をインストールするには、以下のコマンドが使用できます。

```
$ sudo apt-add-repository ppa:sharpie/for-science
$ sudo apt-add-repository ppa:sharpie/postgis-nightly
$ sudo apt-get update
$ sudo apt-get install postgresql-9.1-postgis-nightly
```

16.1.2 Windows でのインストール

Windows へのインストールは少し複雑ですが、それでも大変ではありません。PostGIS のスタックをインストールするには、オンラインである必要があることに注意してください。

まず [このダウンロードページ](#) を参照します。

そして [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細な情報は [PostGIS のウェブサイト](#) 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 のウェブサイトのダウンロード<<http://postgis.net/install/>> _の MacOSX を含む他のプラットフォーム上と他の 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を持っています

16.1.4 PostGIS を使用するためのデータベースの設定

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;"
```

ノート: PostGIS の 1.5 と PostgreSQL のバージョン 9.1 よりも小さいを使用している場合は、データベースの PostGIS の拡張をインストールするためのステップの異なるセットを実行する必要があります。これを行う方法については、_ “ PostGIS のドキュメント <http://postgis.net/docs/postgis_installation.html#create_new_db>を参照してください。このマニュアルの以前のバージョン<http://manual.linfiniti.com/en/postgis/spatial_functions.html#install-plpgsql> _でいくつかの命令もあります。

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:kbd: ‘address’データベースは今地理空間的、PostGIS のおかげで有効になっています。これからのセクションでは、このにたくさんより深く掘り下げるために行くが、迅速に少しテイスターを与えてみましょう。テキストからポイントを作成したいとしましょう。まず、ポイントに関連する機能を見つけるために、psql のコマンドを使用します。まだ:kbd: ‘address’データベースに接続されていない場合は、今接続してください。それから実行:

```
\df *point*
```

これが探しているコマンドです :kbd: *st_pointfromtext*。下向き矢印を使用してリストを上から下まで見たあと、:kbd: ‘q’を押して終了し psql のシェルに戻ります。

このコマンドを実行してみましょう:

```
select st_pointfromtext('POINT(1 1)');
```

結果:

```
st_pointfromtext
-----
010100000000000000000000F03F000000000000F03F
(1 row)
```

注目すべき3点:

- POINT(1 1), を使って、ポイントの位置を 1,1 (EPSG:4326 を想定) と定義しました。
- 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。

結果をテキストに戻すために、テキストを返す何かのための機能リストを探してクイックスキャンできます:

```
\df *text
```

今探しているクエリは:kbd:'st_astext' です。それを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)
```

ここでは、文字列:kbd:'POINT(1,1)' を入力し、:kbd:'st_pointfromtext()' を使用してポイントに変え、:kbd:'st_astext()' で人間が読める形式に戻します。これは元の文字列を返しました。

本当に PostGIS の使用の詳細に入る前の最後の例

```
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 の機能に加えて、拡張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。

まずは、psql の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
```

この結果はこのとおりであるべきです:

```
Table "public.spatial_ref_sys"
  Column |          Type          | Modifiers
-----+-----+-----
srid    | integer                | not null
auth_name | character varying(256) |
auth_srid | integer                |
srtext  | character varying(2048) |
```

```
proj4text | character varying(2048) |
Indexes:
"spatial_ref_sys_pkey" PRIMARY KEY, btree (srid)
```

(入門セクションから学んだように) このテーブルを表示および操作するために、標準の SQL クエリを使用できます - 何をしているか知らないならば、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ない。

興味を持たれる可能性がある一つの SRID は EPSG : 4326 です - WGS 84 楕円体を使用した緯度/経度の地理基準系。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
srttext      | GEOGCS["WGS 84",DATUM["WGS_1984",SPHEROID["WGS
84",6378137,298.257223563,AUTHORITY["EPSG","7030"]],TOWGS84[0,
0,0,0,0,0],AUTHORITY["EPSG","6326"]],PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],UNIT["degree",0.01745329251994328,
AUTHORITY["EPSG","9122"]],AUTHORITY["EPSG","4326"]]
proj4text    | +proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
```

The `srttext` is the projection definition in well known text (you may recognise this from `.prj` files in your shapefile collection).

16.1.7 In Conclusion

今 PostGIS の機能は PostgreSQL のコピーにインストールされています。これを使用すると、PostGIS の大規模な空間関数を使用できるようになります。

16.1.8 What's Next?

次はデータベースにおける空間フィーチャの表現方法について学習しましょう。

16.2 Lesson: 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。

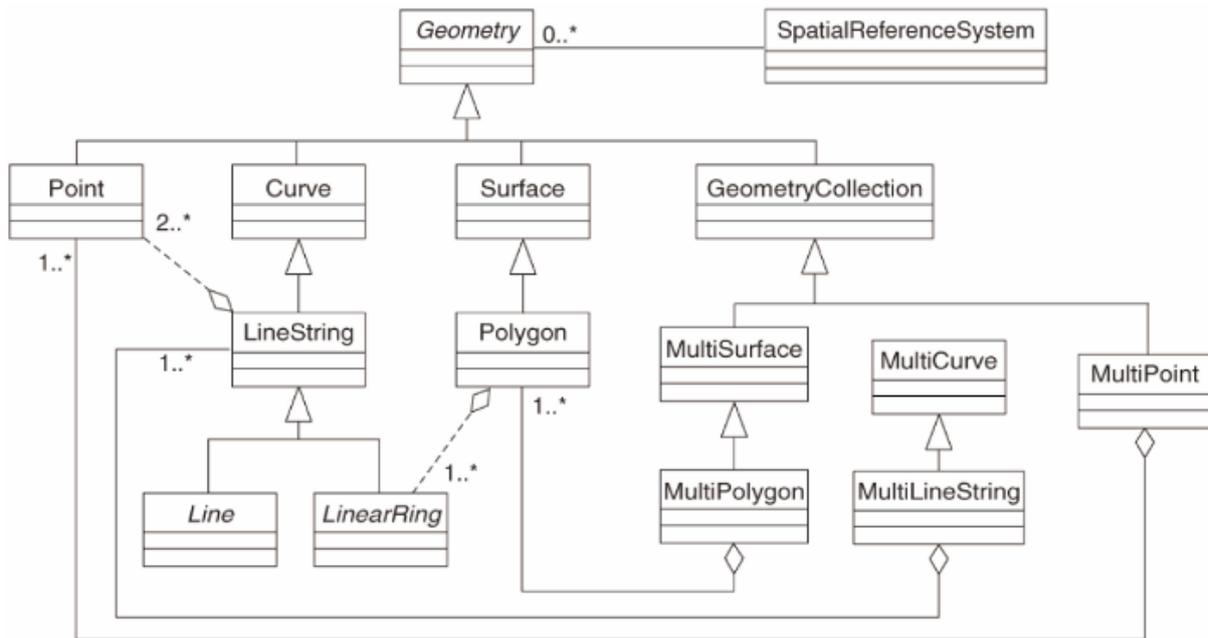
このレッスンの目標: SFS モデルとは何か、それをどうやって使うかを学習します。

16.2.1 OGC とは

The Open Geospatial Consortium (OGC), an international voluntary consensus standards organization, originated in 1994. In the OGC, more than 370+ commercial, governmental, nonprofit and research organizations worldwide collaborate in an open consensus process encouraging development and implementation of standards for geospatial content and services, GIS data processing and data sharing. - *Wikipedia*

16.2.2 SFS モデルとは

The Simple Feature for SQL (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カルな方法で、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。



モデルはポイントやラインストリング、ポリゴンのタイプ(そしてそれらの集合)で地理空間データを定義します。

詳細については [OGC Simple Feature for SQL 標準](#)を参照して下さい。

16.2.3 ジオメトリフィールドをテーブルに追加する

people テーブルにポイントフィールドを追加してみましょう:

```
alter table people add column the_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の型はどのタイプのジオメトリであるかを指定していないことがわかるでしょう。そこで制約が必要となります:

```
alter table people
add constraint people_geom_point_chk
    check(st_geometrytype(the_geom) = 'ST_Point'::text OR the_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 Try Yourself

cities 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。

[結果をチェックする](#)

16.2.6 geometry_columns テーブルの設定

この時点で geometry_columns テーブルにもエントリを追加する必要があります:

```
insert into geometry_columns values
('','public','people','the_geom',2,4326,'POINT');
```

なぜでしょう? `geometry_columns` はデータベースの中のどのテーブルがジオメトリデータを持っているかをアプリケーションが認識するために使われます。

ノート: 上記の `INSERT` ステートメントでエラーが発生する場合にはまずこのクエリを実行します:

```
select * from geometry_columns;
```

If the column `:kbd:'f_table_name'` contains the value `:kbd:'people'`, then this table has already been registered and you don't need to do anything more.

2 の値は次元の数を指しています。この場合は 2 です: `x` と `y`.

4326 の値は私たちが使っている投影法を指しています。WGS 84 は 4326 の数字で参照されます (EPSG に関する以前の解説を参照して下さい)。

Try Yourself



新しい `cities` レイヤーのための適切なエントリを `geometry_columns` に追加して下さい

[結果をチェックする](#)

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

私たちのテーブルにジオメトリを格納することができるようになりました:

```
insert into people (name,house_no, street_id, phone_no, the_geom)
values ('Fault Towers',
       34,
       3,
       '072 812 31 28',
       'SRID=4326;POINT(33 -33)');
```

ノート: 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。

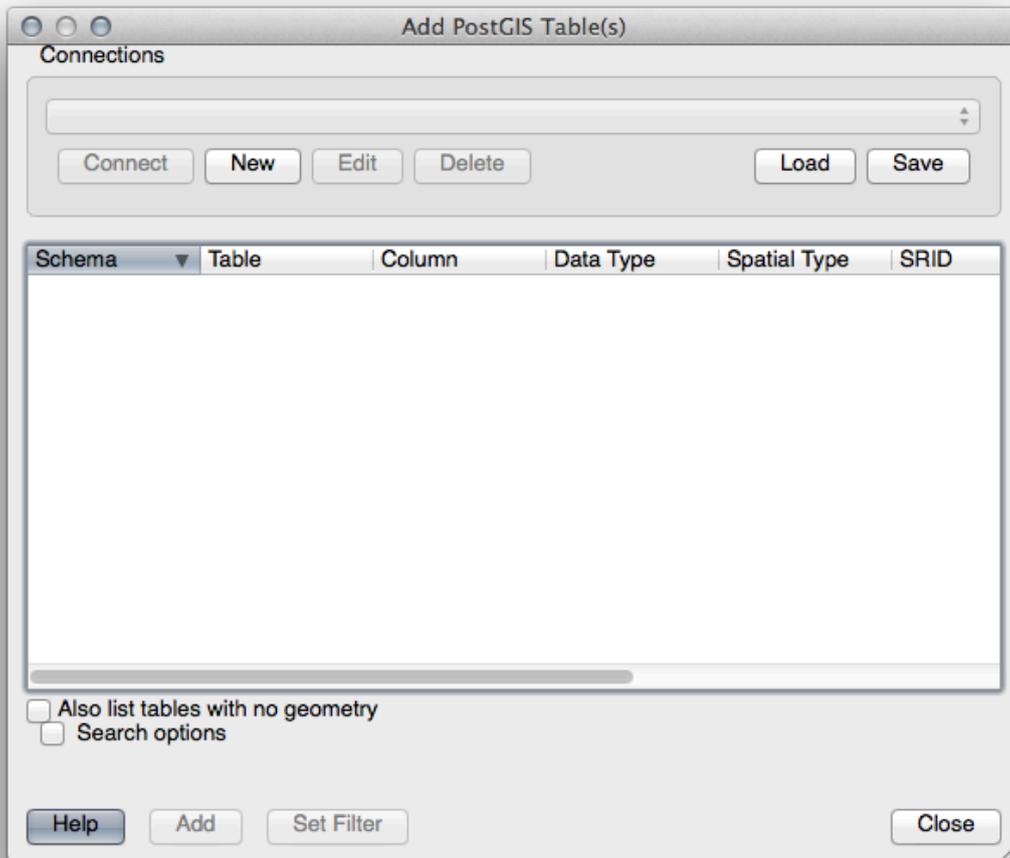
もしグラフィカルなインターフェイスを使用していれば、たとえば、各ポイントの投影法は自動で指定されます。つまり以前行ったようにデータセットに投影法を指定しておけば、すべてのポイントに対して正しい投影法を指定しなくてもよいのです。

では QGIS を開いて `people` テーブルを表示します。そしてデータベースでレコードの編集/追加/削除を試し、選択クエリを実行してデータがどのように変更されたかを見ます。

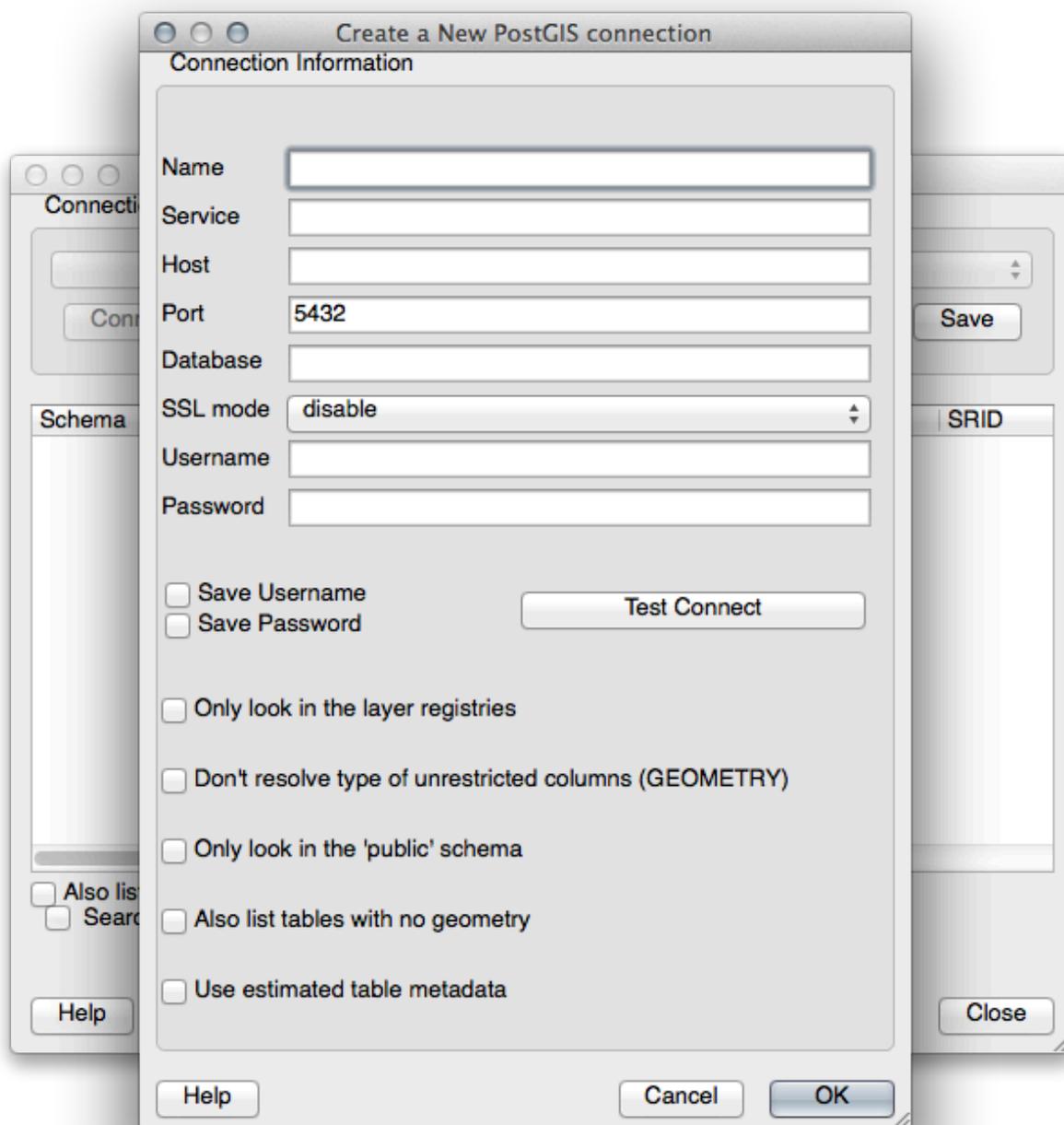
QGIS で PostGIS レイヤーを読み込むには レイヤー → *PostGIS* レイヤーの追加 メニューオプションまたは次のツールバーボタンを使用します:



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:
```

QGIS が address データベースを見つけたかどうか、そしてユーザ名とパスワードが正しいことを確認するには、接続テストをクリックします。正しく動作したら ユーザ名の保存 と パスワード保存 の横にあるチェックボックスをチェックします。そして OK ボタンをクリックしてこの接続を作成します。

PostGIS レイヤの追加 ダイアログに戻り 接続 をクリックし、いつものようにプロジェクトにレイヤを追加します。

Try Yourself

人の名前と街路の名前、位置 (the_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。
結果をチェックする

16.2.8 In Conclusion

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 What's Next?

次はデータベースヘ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 Lesson: インポートとエクスポート

もちろん、簡単にデータを移行する方法はなく、そこから多く使用されることがありません。しかし、幸いなことに簡単にデータを移動でき、PostGIS に出力するツールがいくつかあります。

16.3.1 shp2pgsql

shp2pgsql は、ESRI シェープファイルをデータベースへインポートするためのコマンドラインツールです。Unix では、新しい PostGIS のテーブルをインポートするために、次のようなコマンドを使用できます:

```
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ために空間インデックス *in situ* を作成するための既知の問題です。エラーを回避するためには、`-I` パラメータを外します。これは空間インデックスが直接作成されないことを意味します。データがインポートされた後にデータベースに空間インデックスを作成する必要があります (空間インデックスの作成は、次のレッスンで説明します.)。

16.3.2 pgsq2shp

pgsq2shp は、PostGIS のテーブル、ビュー、または SQL の select クエリをエクスポートするためのコマンドラインツールです。Unix では次のように実行します

```
pgsq2shp -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 は、PostGIS を多くのデータフォーマットに変換する強力なツールです。ogr2ogr は、GDAL/OGR ソフトウェアの一部であり、個別にインストールする必要があります。テーブルを PostGIS から GML へエクスポートするには、このコマンドを使用することができます

```
ogr2ogr -f GML export.gml PG:'dbname=<databasename> user=<username>  
host=<hostname>' <Name of PostGIS-Table>
```

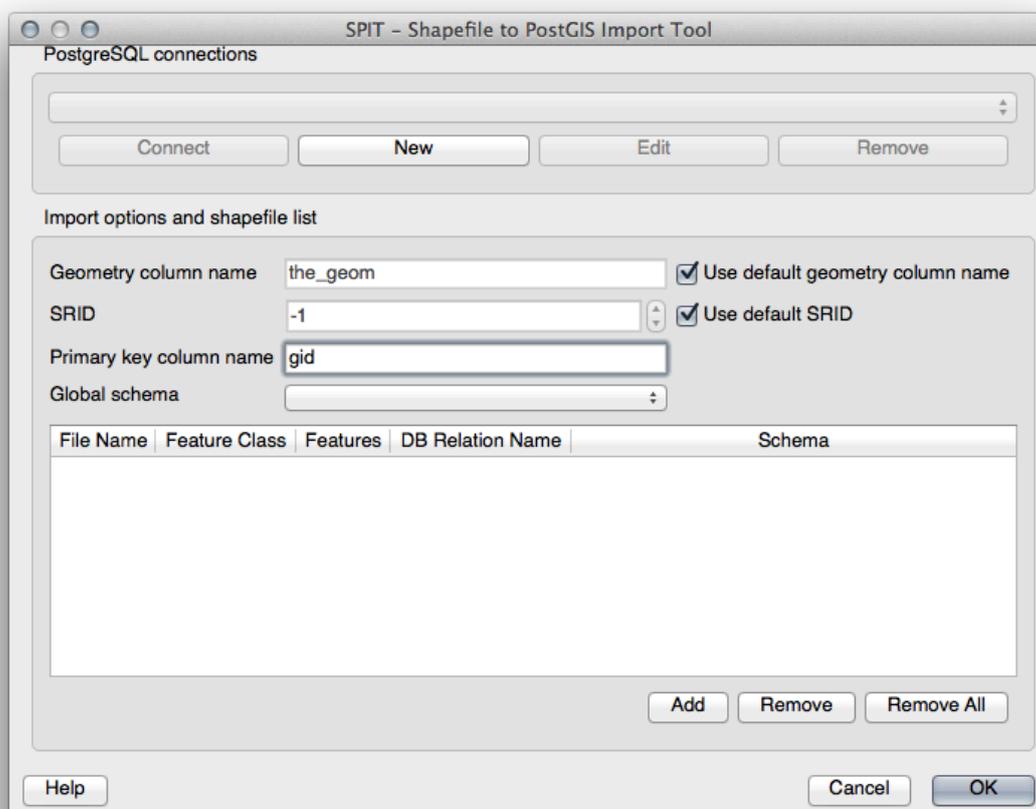
16.3.4 SPIT

SPIT は QGIS と一緒に配布される QGIS プラグインです。PostGIS に ESRI シェープファイルをアップロードするために SPIT を使用することができます。

Plugin Manager を介して SPIT プラグインを追加したら、このボタンを探します：



このボタンをクリックするか、メニューから *Database -> Spit -> Import Shapefiles to PostgreSQL* のように多たどり、SPIT ダイアログを表示できます。



Add ボタンをクリックすることでファイルブラウザウィンドウを開き、データベースにシェープファイルを追加することができます。

16.3.5 DB Manager

Database メニュー内の *DB Manager* という別のオプションに気づいているかもしれません。これは PostGIS を含む空間データベースと対話する統合的なインターフェースを持った QGIS 2.0 の新しいツールです。このツールもインポートしたデータベースから他のフォーマットにエクスポートすることができます。次のモジュールは、主にこのツールを使うことを念頭に置いているので、ここでは簡単にそれを説明します。

16.3.6 In Conclusion

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。

16.3.7 What's Next?

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 Lesson: 空間検索

地理空間情報のクエリは、その他のデータベースのクエリと変わりなく、同じように利用できます。PostGIS を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は地理空間関数が、一般的な地理空間以外の関数と同様に導入できることを明らかにすることです。

16.4.1 地理空間オペレータ

ある地点 (X,Y) から 2 度距離がある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(the_geom, 'SRID=4326;POINT(33 -34)') < 2;
```

結果:

id	name	house_no	street_id	phone_no	the_geom
6	Fault Towers	34	3	072 812 31 28	01010008040C0

(1 row)

ノート: 上記 `geom` value は当サイトのスペースを残すため削除されました。human-readable coordinates を確認するためには、"View a point as WKT" セクションと類似した操作で確認することができます。

上述のクエリが 2 度という空間内にある地点をすべてかえすということはどうしたらわかりますでしょうか? 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?

結果をご確認ください

16.4.2 地理空間インデックス

地理空間のインデックスも定義することができます。地理空間のインデックスはクエリを高速化します。地理空間のインデックスをジオカラムに作成するには：

```
CREATE INDEX people_geo_idx
ON people
USING gist
(the_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     |
  the_geom   | geometry               |
```

Indexes:

```
"people_pkey" PRIMARY KEY, btree (id)
"people_geo_idx" gist (the_geom) <-- new spatial key added
"people_name_idx" btree (name)
```

Check constraints:

```
"people_geom_point_chk" CHECK (st_geometrytype(the_geom) = 'ST_Point'::text
OR the_geom IS NULL)
```

Foreign-key constraints:

```
"people_street_id_fkey" FOREIGN KEY (street_id) REFERENCES streets(id)
```

16.4.3 Try Yourself



そのジオメトリ列が空間索引付けされたように、都市のテーブルを変更します。

結果の確認

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。

開始のため、新しいデータベース（第一に psql シェルが存在する）を作成します：

```
createdb postgis_demo
```

Remember to install the postgis extensions:

```
psql -d postgis_demo -c "CREATE EXTENSION postgis;"
```

次に、提供されたデータを :kbd: `exercise_data / PostGIS の` ディレクトリの中にインポートします。手順については前のレッスンを参照していただきますが、新しいデータベースへの新しい PostGIS の接続を作成する必要があることを覚えておいてください。端末から、または、SPIT を経由して、インポートできます。次のデータベーステーブルにファイルをインポートします：

- `points.shp` を `building` に
- `lines.shp` を `road` に

- polygons.shp を region に

いつものように、PostGIS のレイヤーを追加 ダイアログでこれらの 3 つのデータベースレイヤーを QGIS にロードします。それらの属性テーブルを開くと、id フィールドと PostGIS のインポートによって作成された gid フィールドの両方があることに注意。

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、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取得する:

```
SELECT a.id, a.name, st_astext(a.the_geom) as point
FROM building a, region b
WHERE st_within(a.the_geom, b.the_geom)
AND b.name = 'KwaZulu';
```

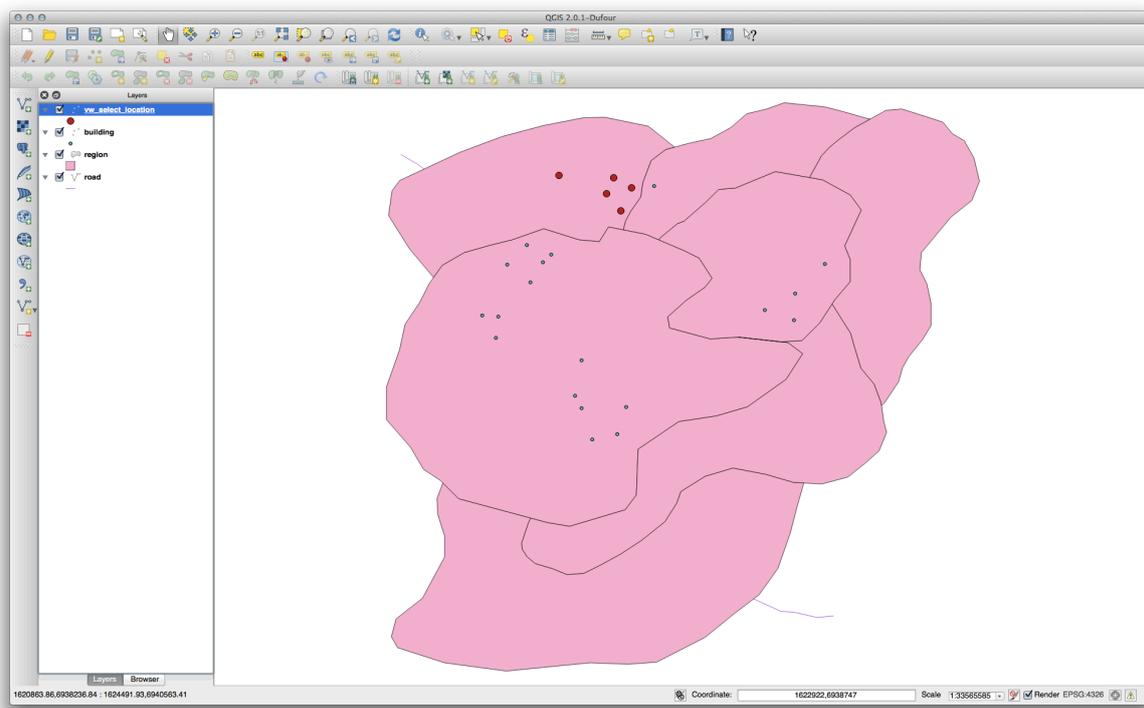
結果:

```
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)
```

あるいは、それからビューを作成する場合:

```
CREATE VIEW vw_select_location AS
SELECT a.gid, a.name, a.the_geom
FROM building a, region b
WHERE st_within(a.the_geom, b.the_geom)
AND b.name = 'KwaZulu';
```

レイヤーとしてビューを追加し、QGIS で表示:



近傍の選択

北海道地区に隣接するエリアの名称すべてのリストを表示します。

```
SELECT b.name
FROM region a, region b
WHERE st_touches(a.the_geom, b.the_geom)
AND a.name = 'Hokkaido';
```

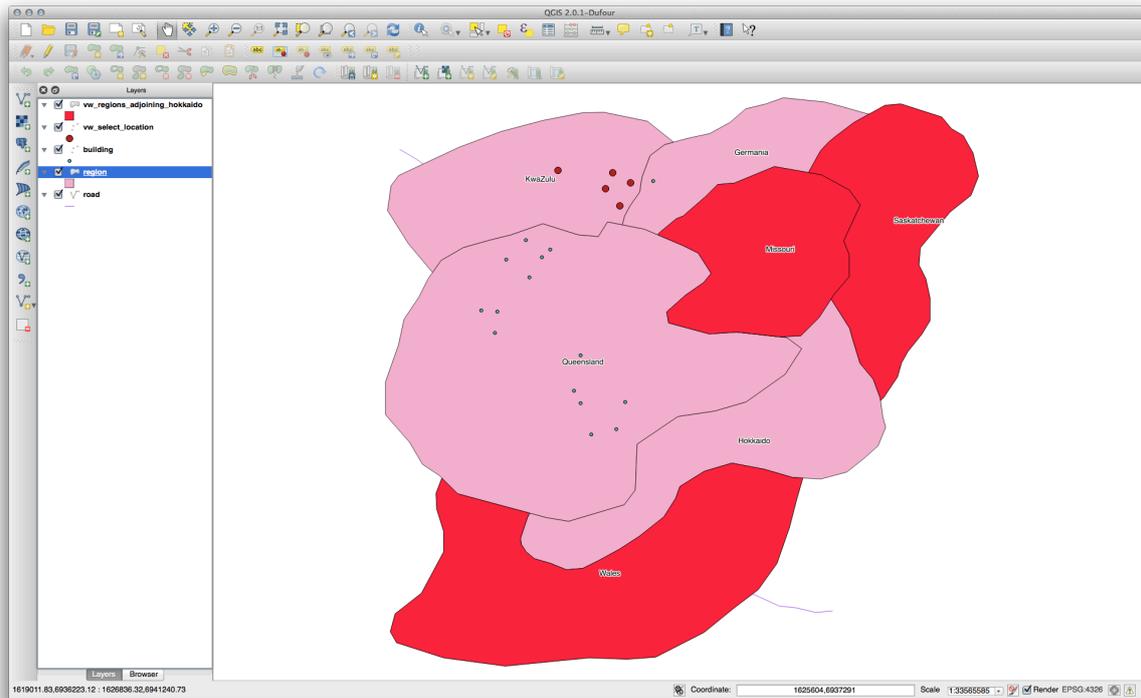
結果:

```
name
-----
Missouri
Saskatchewan
Wales
(3 rows)
```

見た目として:

```
CREATE VIEW vw_regions_adjoining_hokkaido AS
SELECT b.gid, b.name, b.the_geom
FROM region a, region b
WHERE TOUCHES(a.the_geom, b.the_geom)
AND a.name = 'Hokkaido';
```

QGIS では:

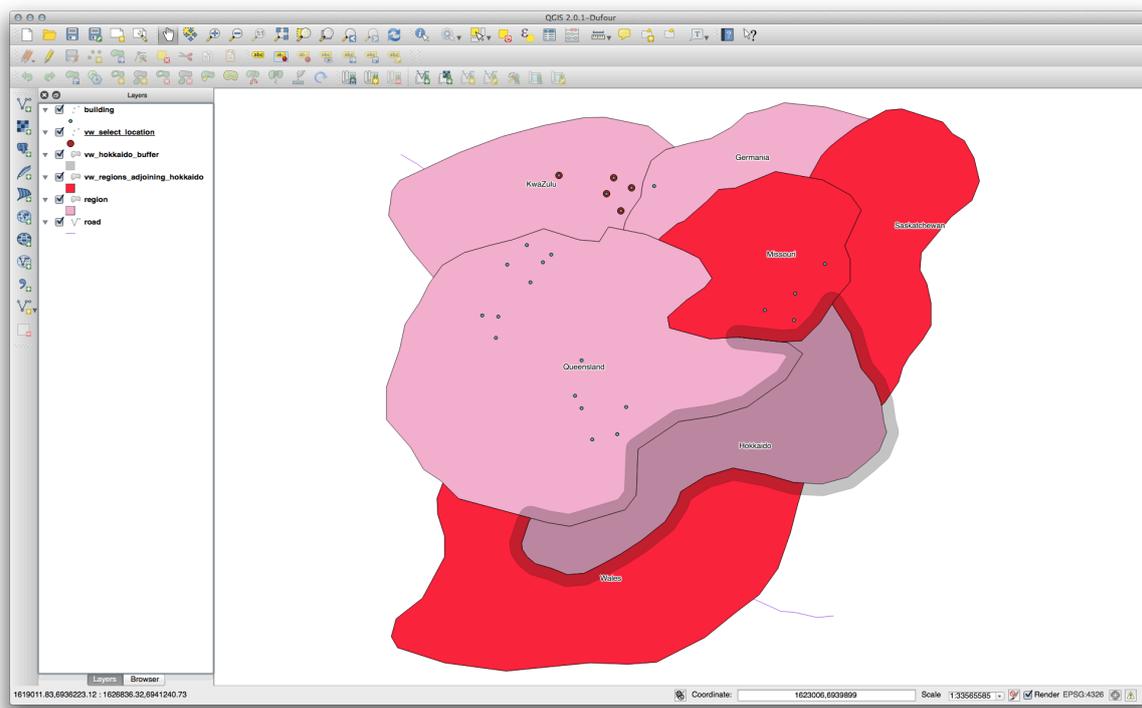


不足している地域（クイーンズランド州）に注意してください。これは、トポロジエラーに起因する可能性があります。このようなアーティファクトは、データの潜在的な問題のために警告することがあります。データが持つかもしれない異常に巻き込まれることなく、この謎を解決するために、代わりにバッファ交差を使用できます：

```
CREATE VIEW vw_hokkaido_buffer AS
  SELECT gid, ST_BUFFER(the_geom, 100) as the_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：

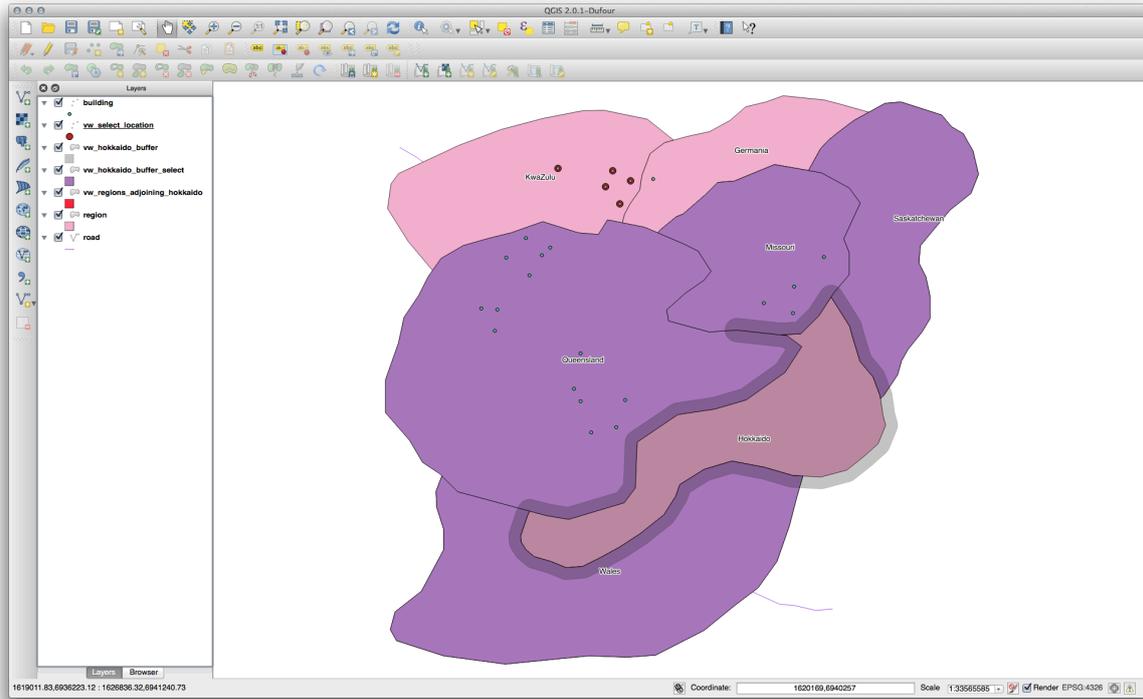


バッファを使用して選択します:

```
CREATE VIEW vw_hokkaido_buffer_select AS
  SELECT b.gid, b.name, b.the_geom
    FROM
      (
        SELECT * FROM
          vw_hokkaido_buffer
      ) a,
      region b
  WHERE ST_INTERSECTS(a.the_geom, b.the_geom)
        AND b.name != 'Hokkaido';
```

このクエリでは、元のバッファ・ビューは、他のテーブルがされるように使用されます。これは別名 a を与えられ、その幾何形状フィールド region テーブル (別名 b) a.the_geom が、それと交差するすべての多角形を選択するために使用されます。しかし北海道自体はこの select 文から除外されます。必要なのは北海道自体ではなく、それに隣接する領域だけなので。

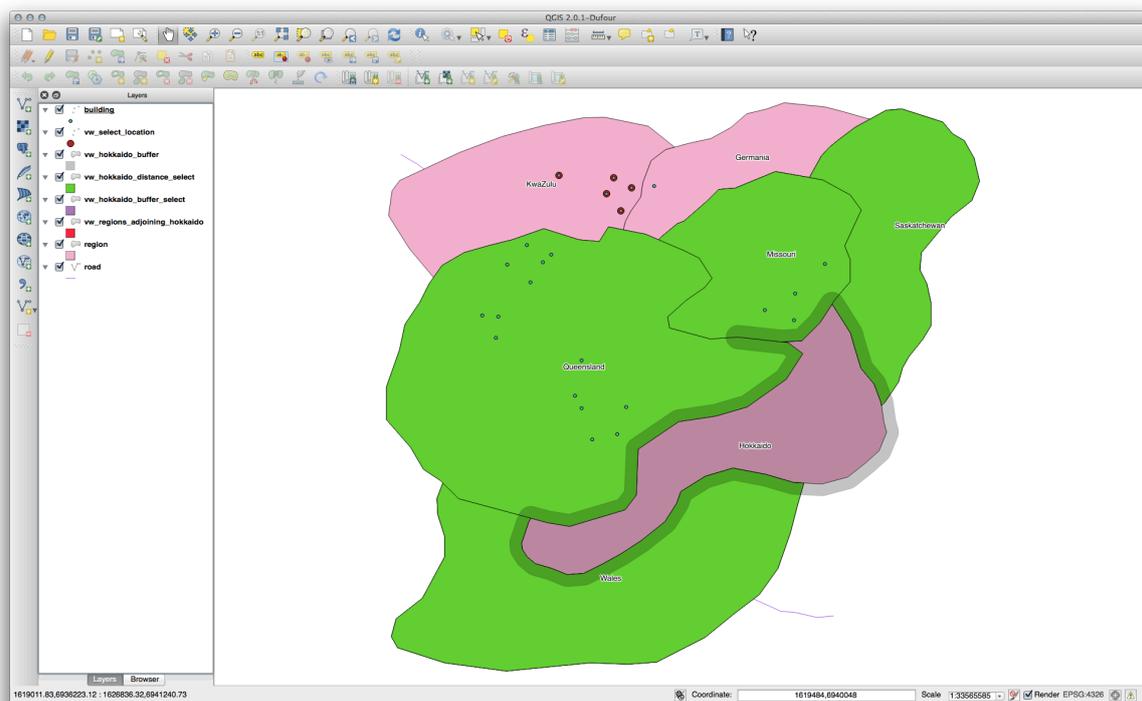
QGIS では:



バッファを作成するための余分な工程を経ることなく、与えられた距離内のすべてのオブジェクトを選択することも可能である:

```
CREATE VIEW vw_hokkaido_distance_select AS
SELECT b.gid, b.name, b.the_geom
FROM region a, region b
WHERE ST_DISTANCE (a.the_geom, b.the_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

クィーンズランド州に所在する建物の所在地の村の名前をリストにします。

```
SELECT DISTINCT a.name
FROM building a, region b
WHERE st_within(a.the_geom, b.the_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.the_geom, b.the_geom)) as
text, ST_SHORTESTLINE(a.the_geom, b.the_geom) AS the_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.the_geom, b.the_geom)) as
text, ST_LONGESTLINE(a.the_geom, b.the_geom) AS the_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.the_geom) as the_geom
```

```

FROM road a
  WHERE a.id = 1;

CREATE VIEW vw_region_centroid AS
  SELECT a.gid as gid, ST_CENTROID(a.the_geom) as the_geom
  FROM region a
  WHERE a.name = 'Saskatchewan';

SELECT ST_PERIMETER(a.the_geom)
FROM region a
  WHERE a.name='Queensland';

SELECT ST_AREA(a.the_geom)
FROM region a
  WHERE a.name='Queensland';

CREATE VIEW vw_simplify AS
  SELECT gid, ST_Simplify(the_geom, 20) AS the_geom
  FROM road;

CREATE VIEW vw_simplify_more AS
  SELECT gid, ST_Simplify(the_geom, 50) AS the_geom
  FROM road;

CREATE VIEW vw_convex_hull AS
  SELECT
    ROW_NUMBER() over (order by a.name) as id,
    a.name as town,
    ST_CONVEXHULL(ST_COLLECT(a.the_geom)) AS the_geom
  FROM building a
  GROUP BY a.name;

```

16.4.5 In Conclusion

PostGIS のから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照会する方法を見てきました。

16.4.6 What's Next?

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 Lesson: ジオメトリの構成

このセクションではシンプルなジオメトリが SQL でどのように構成されているか少し掘り下げます。実際には、複雑なジオメトリをデジタイジングツールを使用して作るためには、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が、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。

このレッスンの目的 PostgreSQL/PostGIS で空間要素を直接作成する方法をよく理解する。

16.5.1 ラインストリングの作成

address データベースに戻って、他に一致する街路テーブルを取得してみましょう。すなわち、ジオメトリ、インデックスと GEOMETRY_COLUMNS テーブル中のエントリに制約を持ちます。

16.5.2 Try Yourself

- タイプ ST_LineString のジオメトリ列を持つように 街路 テーブルを修正します。
- 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、ラインストリングまたは null ではありません追加されているすべてのジオメトリを防止するための制約を追加します。
- 新しいジオメトリカラムに空間インデックスを作成します。

結果の確認

では、ラインストリングを streets テーブルに挿入しましょう。この場合は、既存の道路のレコードを更新します

```
update streets set the_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道路の要素をもう少し追加します。

16.5.3 ポリゴンの作成

ポリゴンを作成することも簡単です。覚えておくことは、定義により、ポリゴンは最低4つの頂点を持ち、最初と最後の頂点は同じ場所になるということです:

```
insert into cities (name, the_geom)
values ('Tokyo', 'SRID=4326;POLYGON((10 -10, 5 -32, 30 -27, 10 -10))');
```

ノート: 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になります。例えば

```
insert into cities (name, the_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 に都市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。
- 都市テーブルの主キーを参照する人々への外部キー列を追加します。
- QGIS を使用していくつかの都市をキャプチャします。
- それぞれが関連付けられている通り、街を持っていることを確認して、いくつかの新しい人々のレコードを挿入する SQL を使用してください。

更新された人々のスキーマは次のようになります:

```
\d people
```

```
Table "public.people"
  Column | Type | Modifiers
```

```

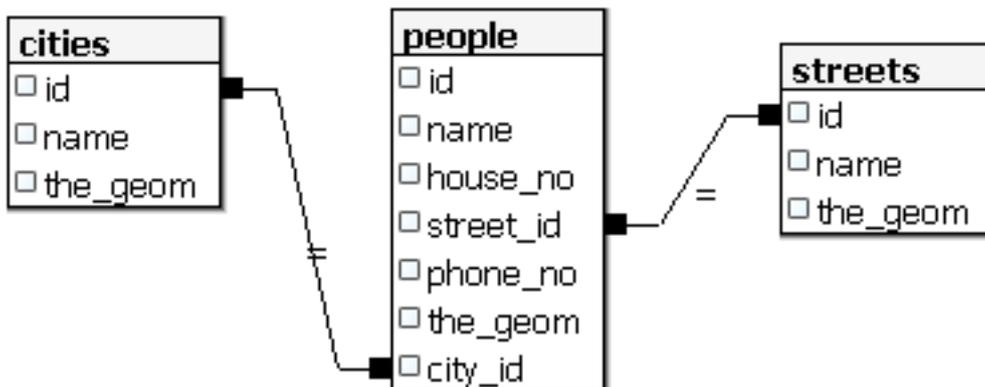
-----+-----+-----
id      | integer          | not null
        |                  | default nextval('people_id_seq'::regclass)
name    | character varying(50) |
house_no | integer          | not null
street_id | integer          | not null
phone_no | character varying  |
the_geom | geometry         |
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(the_geom) =
                             'ST_Point'::text OR the_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)

```

結果の確認

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 Try Yourself

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。

16.5.7 サブオブジェクトへのアクセス

SFS-モデルの機能を使用すると、SFS ジオメトリのサブオブジェクトにアクセスするためのさまざまなオプションを持っています。テーブル myPolygonTable 内のすべてのポリゴンジオメトリの最初の頂点を選択したいときは、このような方法でこれを実行する必要があります:

- ラインストリングにポリゴン境界を変換する:


```
select st_boundary(geometry) from myPolygonTable;
```
- その結果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(geometry) as myGeometry
  from myPolygonTable) as foo;
```

16.5.8 データプロセッシング

PostGIS には、すべての OGC SFS / MM 標準では機能に準拠サポートしています。すべてのこれらの関数は “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() 関数を使用できます。空のジオメトリを回避するために、以下を使用:

```
where not st_isempty(st_intersection(a.the_geom, b.the_geom))
```

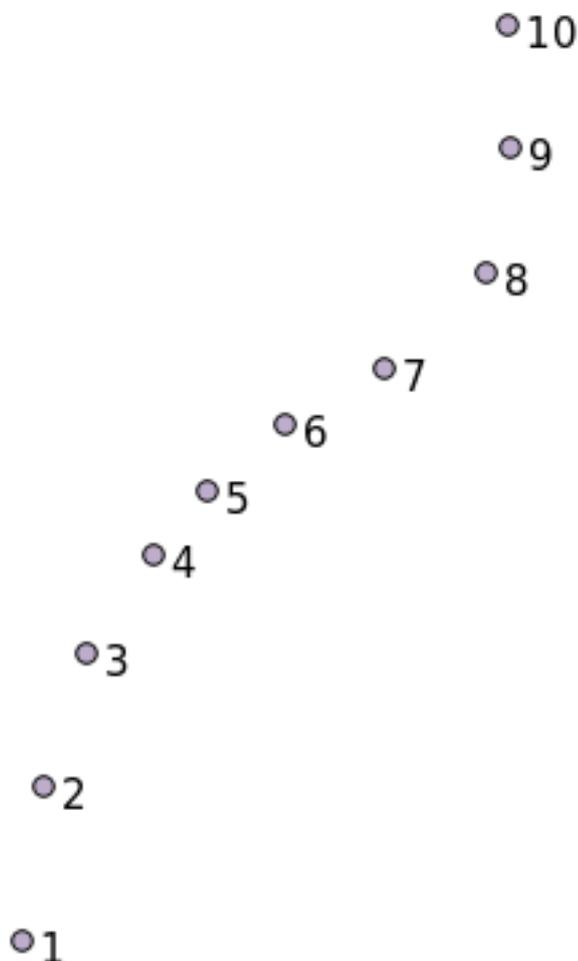


```
select st_intersection(a.the_geom, b.the_geom), b.*
from clip as a, road_lines as b
where not st_isempty(st_intersection(st_setsrid(a.the_geom, 32734),
  b.the_geom));
```



16.5.10 他の形状から建物のジオメトリ

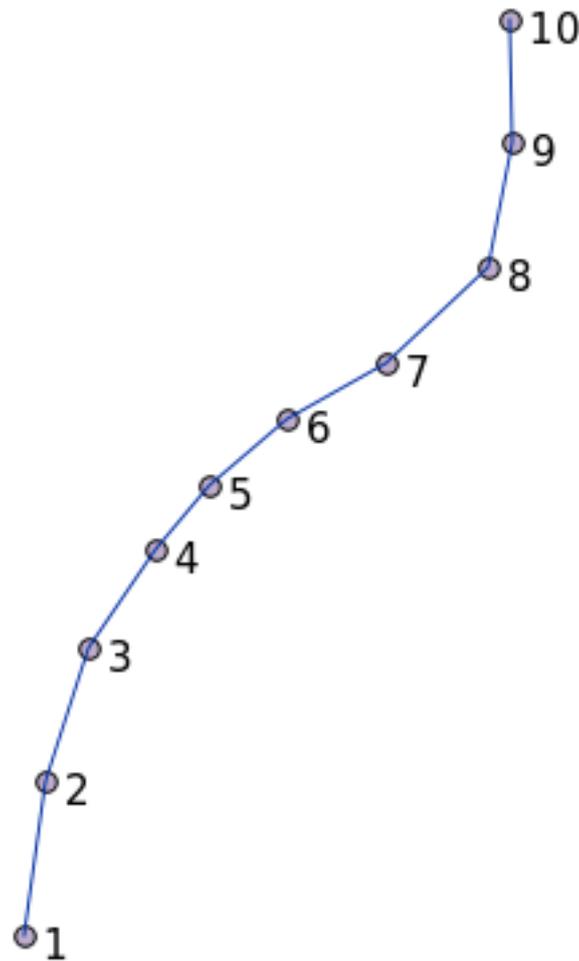
与えられたポイントテーブルから、ラインストリングを生成します。ポイントの順序は、その `id` によって定義されます。別の並べ方は、GPS 受信機でウェイポイントをキャプチャするときを得るような、タイムスタンプになりうるかもしれません。



「ポイント」と呼ばれる新しいポイントレイヤーからラインストリングを作成するため、次のコマンドを実行できます:

```
select ST_LineFromMultiPoint(st_collect(the_geom)), 1 as id
from (
  select the_geom
  from points
  order by id
) as foo;
```

もちろん、これを行うのは現実世界の意味はほとんどないでしょうが、どのように機能するかを確認するには、「人」レイヤーにもこのコマンドを実行できます。



16.5.11 ジオメトリクリーニング

You can get more information for this topic in [this blog entry](#).

16.5.12 テーブル間の違い

同じ構造を持つ2つのテーブル間の差を検出するには、PostgreSQLのEXCEPTキーワードを使用できます:

```
select * from table_a
except
select * from table_b;
```

結果として、table_aからtable_bに格納されていないすべてのレコードが得られるでしょう。

16.5.13 表領域

Postgresは表領域を作成することにより、ディスク上のデータを保存する場所を定義できます:

```
CREATE TABLESPACE homespace LOCATION '/home/pg';
```

データベースを作成するときは、その後、例えば、使用する表領域を指定することができます:

```
createdb --tablespace=homespace t4a
```

16.5.14 In Conclusion

PostGIS の文を使用して、より複雑な形状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場合、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。通常、実際に手動でこれらのステートメントを入力する必要はありませんが、そうでなければ不可解と思われるエラーが発生した場合は特に、GIS を使用した場合、その構造の一般的な考えを持つことに役立ちます。

Chapter 17

QGIS プロセッシングガイド

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 プロセッシングフレームワークを使用する方法について説明します。これはプロセッシングフレームワークや、フレームワークが依存するアプリケーションなどについての知識を必要とするものではありません。QGIS の基本的な知識が必要とされるのみです。

このガイドは自習用に設計されたプロセッシングワークショップを実行するためのものです。

このガイドの例では、QGIS 2.0 を使用し、部分的なアップグレードは 2.10 になっています。それらは動作しないかもしれないか、それ以外のバージョンでは利用できないかもしれません。

このガイドは、漸進的な複雑さの一連の小さな演習で構成されています。処理フレームワークを使用したことがない場合は、最初から開始する必要があります。以前の経験がある場合は、自由にレッスンをスキップしてください。彼らは多かれ少なかれ互いに独立しており、それぞれ新しい概念や新しい要素を導入しています。これは章のタイトルと各章の初めの短い紹介で示されています。これにより、特定のトピックを扱うレッスンを簡単に見つけることができます。

すべてのフレームワークコンポーネントとその使用方法の体系的な説明のために、QGIS マニュアルの該当する章を確認することをお勧めします。このガイドに沿ってサポートテキストとしてください。

このガイドのすべての演習では、QGIS のウェブサイト<<http://qgis.org/downloads/data/>> からダウンロードできる無料のデータセットを使用しています。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複数のフォルダが含まれています。それぞれに QGIS 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準備が整います。

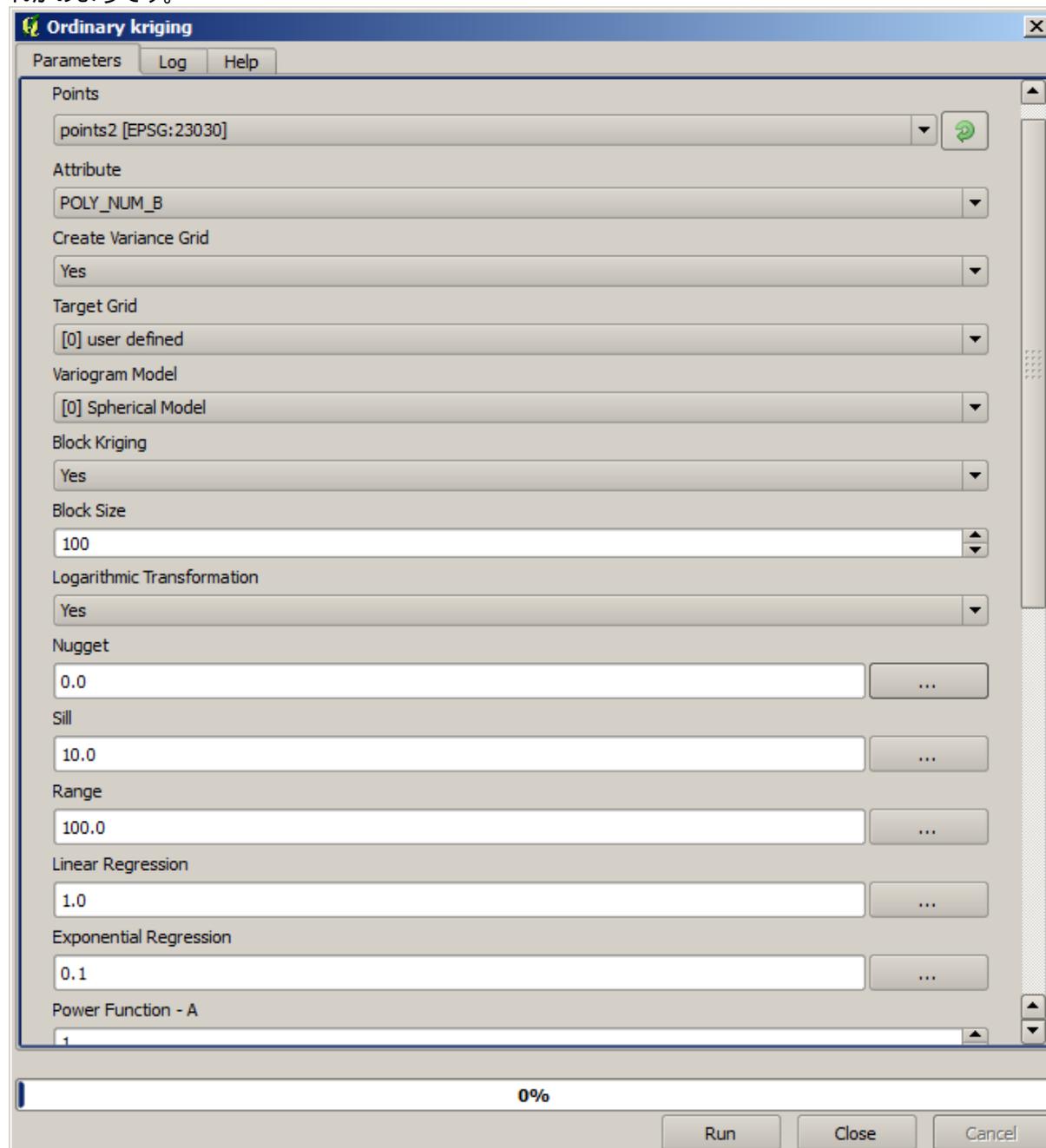
楽しんで!

17.2 開始前の重要な警告

ちょうど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お教えしないように、また CAD チュートリアルは建物のための梁のサイズを計算する方法をお見せしないように、このガイドでは、空間分析をお教えしません。その代わりに、QGIS 処理フレームワーク、空間的な分析を行うための強力なツールを使用する方法を説明しますが、分析の種類を理解するために必要とされる必須概念を学ぶためかどうかはあなた次第です。それらがなければ、したいと誘惑されても、フレームワークとそのアルゴリズムを使用するにはポイントがありません。

例を挙げてより明確にこれをお見せしましょう。

点の集合および各点で与えられた変数値の値を与えられると、それらから*クリギング* `geostat` を使用し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] ダイアログボックスは、以下のいずれかのようなものです。



それは複雑に見えます、よね？

このマニュアルを読むことによって、そのモジュールを使用する方法、たくさんの点レイヤーからラスタレイヤを作成するために1回の実行でバッチ処理でそれを実行する方法、入力レイヤーでいくつかの点を選択された場合は何が起こるか、などを学ぶでしょう。しかし、パラメータ自体は説明されていません。地球統計学の十分な知識を持つベテランの分析者は、これらのパラメータを理解するのに何の問題もありません。そうでない方、敷衍、レンジ、または*ナゲット*という概念をよく知らない方は、*クリギング*モジュールを使用しないでください。それはおそらく前にも聞いていない、または少なくとも十分な長勉強していない、そのような空間的自己相関や `semivariograms` などの概念、についての学習が必要です、それ以上に、はるか*クリギング*モジュールを使用する準備ができています。最初に勉強し、理解して、その後、実際にそれを実行し、分析を実行するために戻って QGIS に来る必要があります。これを無視すると、誤った結果と貧しい（そして役に立たない可能性が高い）分析になります。

すべ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、ほとんどそれらのすべてにおいて、それらが基づいている基本的な分析の考え方を理解することが必要とされます。その知識がなければ、それらを使用するても大した結果は出ない可能性が高いでしょう。

空間分析の良い基盤を持たずに `geoalgorithms` を使用するの、文法や構文については何も知らずに、そして物語について何の知識もなしに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ないが、まったく価値がない可能性があります。このガイドを読んだ後はもう空間解析を行って健全な結果を得ることが可能であると愚かにも考えないでください。空間的分析を勉強する必要もあります。

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
 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・ロングリーへの総合ガイド

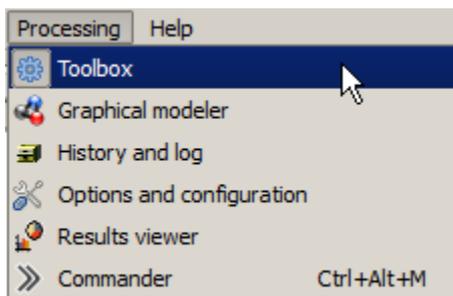
それは <<http://www.spatialanalysisonline.com/>> ‘ ‘ここにオンラインで入手できます

17.3 プロセッシングフレームワークを構成する

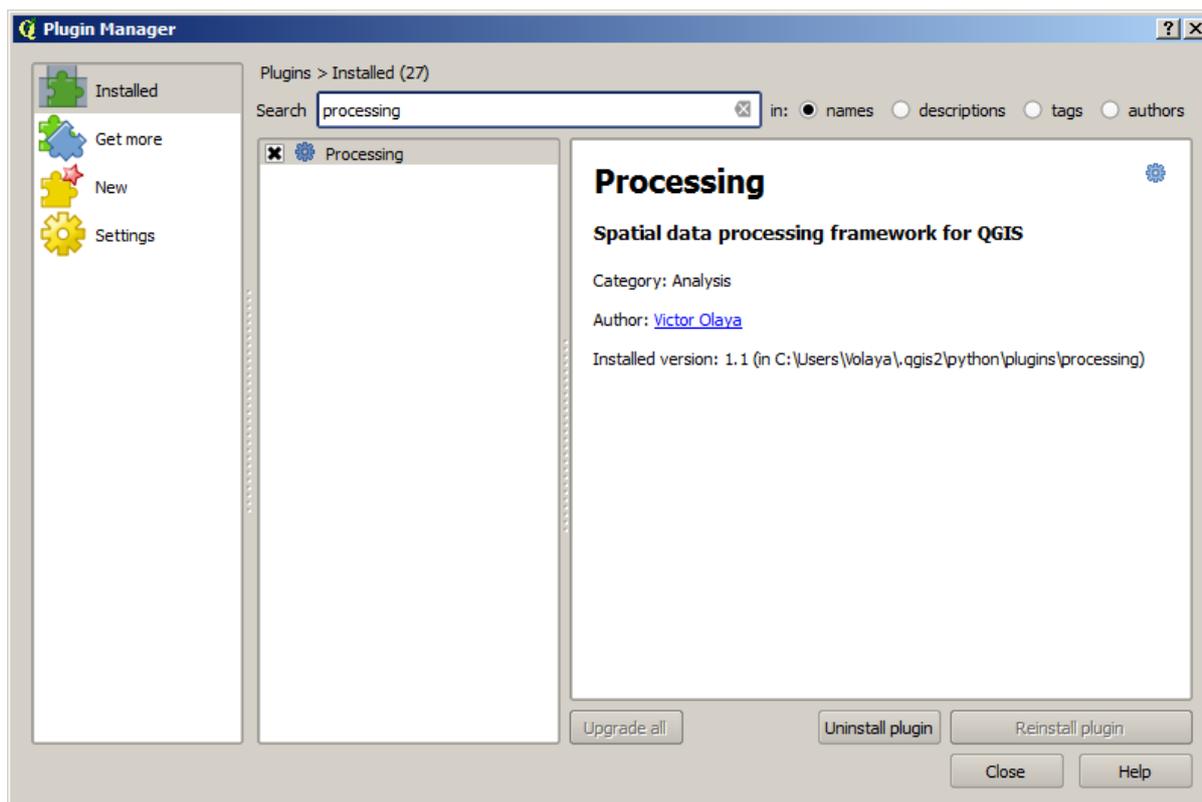
プロセッシングフレームワークを使う前に環境設定をします。設定項目は多くないので簡単です。

その後、我々は、利用可能なアルゴリズムのリストを拡張するために使用されている外部アプリケーションを構成する方法を示しますが、今は我々だけでフレームワーク自体で作業しようとしています。

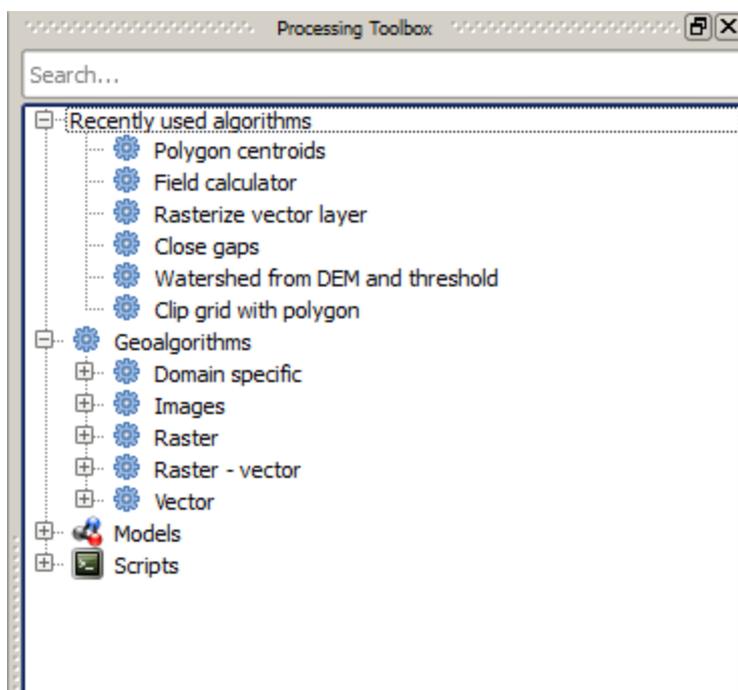
処理フレームワークを使用すると、QGIS 2.0 以降を実行している場合、それは QGIS に含まれているので、それはすでに、システムにインストールする必要があります、ということの意味コア QGIS プラグインです。それがアクティブである場合は、メニューバーに 処理 と呼ばれるメニューが表示されるはずですが。そこには、すべてのフレームワークコンポーネントへのアクセスがあります。



メニューを見つけれない場合、プラグインマネージャでそれを有効にすることで、プラグインを使用可能にする必要があります。



作業するのに使用する主な要素は、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ドッキングされ、ツールボックスが表示されます。



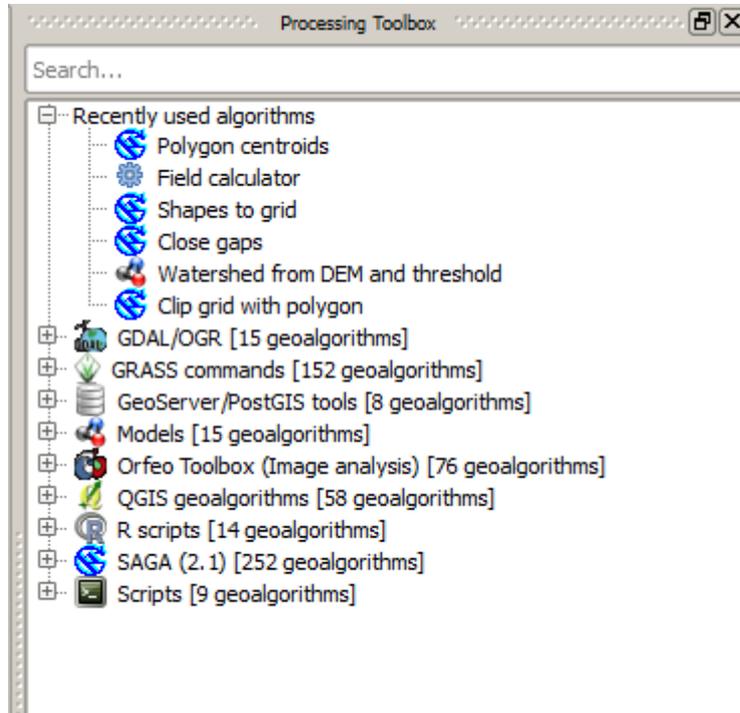
ツールボックスは、グループに分け、すべての利用可能なアルゴリズムのリストが含まれています。高度なインターフェースと簡体インターフェース：これらのアルゴリズムの表示と整理の二つの方法があります。

デフォルトでは、アルゴリズムは、彼らが実行する操作の種類に応じたグループ、簡易モードが表示されます。ツールボックスに表示されるアルゴリズムのいくつかは、他の外部アプリケーション（それらのほとんどは、実際には、やる）に依存しますが、これらのアプリケーションへの言及は表示されません。アルゴリズムの原点は、処理フレームワークを介してアルゴリズムを使用して簡素化するファサードで、このモードでは隠されています。

このガイドの最初の例は、単なる簡易モードを使用しています。アドバンスモードでは、いくつかの追加機能とアルゴリズムを持っていますが、それは呼ばれているアプリケーションを理解する必要があり、そう、彼らはより高度なトピックであり、後に説明します。

ツールボックスの底部にセレクトを使用することによって簡略化された、高度なインタフェースの間で変化することができます。

上級者トモードで使用する場合、ツールボックスはこのように見えます。



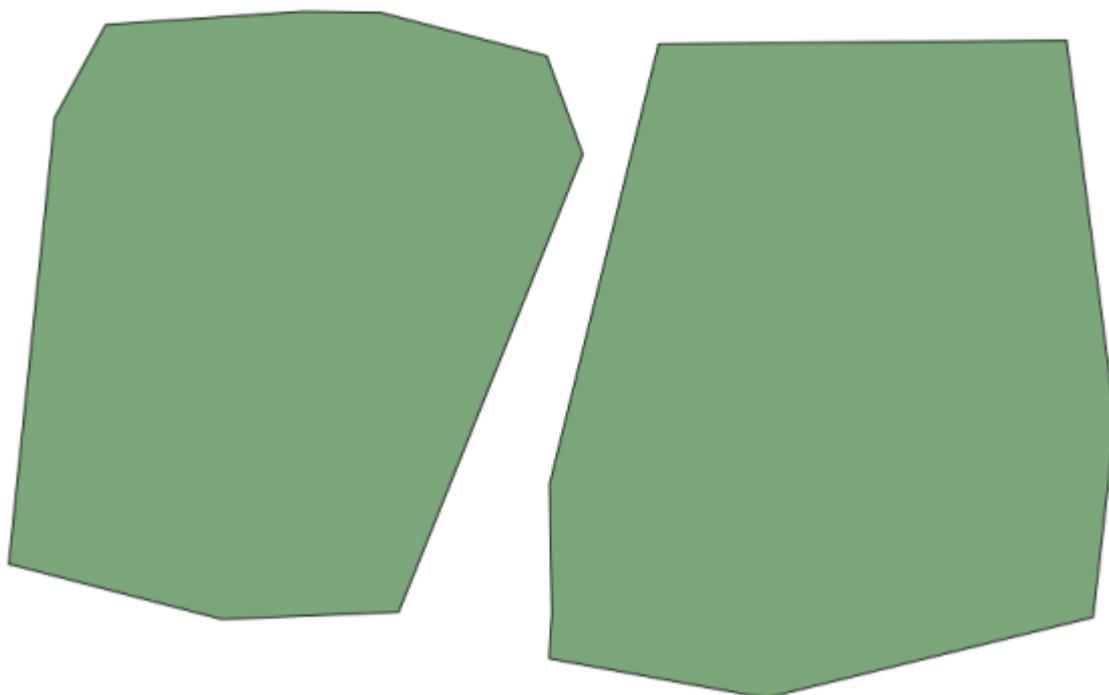
この点に達している場合は、今、geoalgorithms を使用する準備が整いました。今では何かを設定する必要はありません。すでに、次のレッスンで行います私たちの最初のアルゴリズムを、実行できます。

17.4 最初のアルゴリズムを実行します。ツールボックス

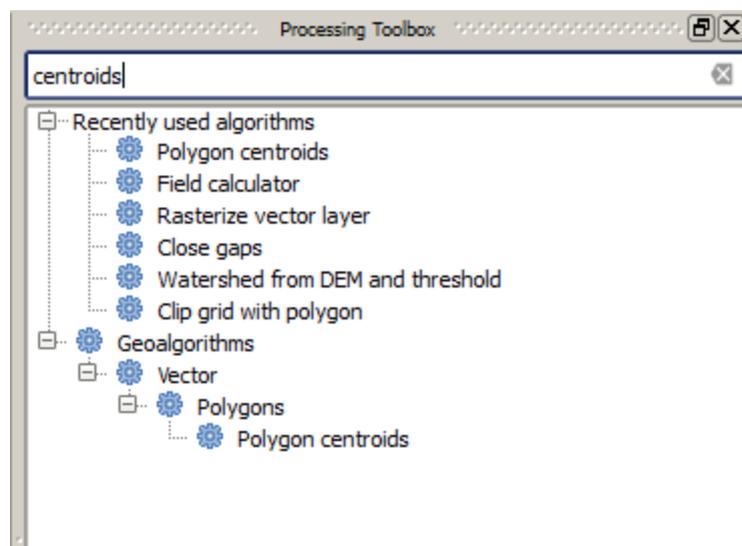
ノート: このレッスンでは、最初のアルゴリズムを実行し、それから、最初の結果を取得します。

すでに述べたように、処理フレームワークは、他のアプリケーションからのアルゴリズムを実行できますが、それはまた、実行すべき外部ソフトウェアを必要としないネイティブのアルゴリズムが含まれています。処理フレームワークの探索を開始するために、それらのネイティブのアルゴリズムのいずれかを実行するつもりです。特に、ポリゴンの集合の重心を計算するつもりです。

まず、このレッスンに対応する QGIS プロジェクトを開きます。これは、2つのポリゴンを持つだけの単層が含まれています

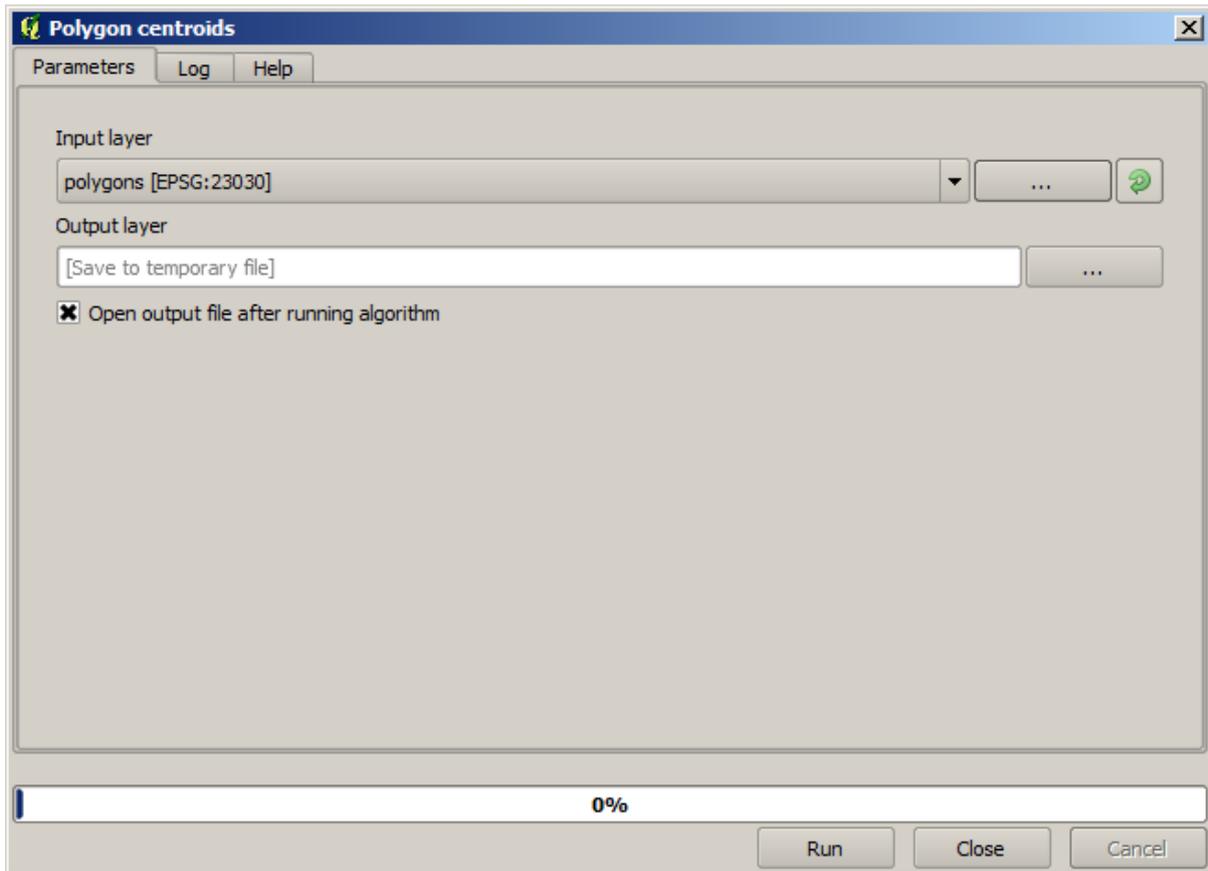


今、ツールボックスの上部にあるテキストボックスにアクセスしてください。これは検索ボックスで、それにテキストを入力すると、入力したテキストを含むものだけが表示されるので、アルゴリズムのリストをフィルタします。“centroids”を入力するとこのようなものが表示されるはずです。



検索ボックスは、探しているアルゴリズムを見つける非常に実用的な方法です。

アルゴリズムを実行するには、ツールボックスでその名前をダブルクリックする必要があるだけです。重心アルゴリズムをダブルクリックすると、次のダイアログが表示されます。



アルゴリズムはすべて同様のインターフェースを持っていて、基本的に記入しなければならない入力パラメータと、どこに保存するか選択する必要がある出力が含まれています。この場合は、持っている唯一の入力はポリゴンでのベクトルレイヤーです。

入力として ポリゴン レイヤーを選択します。アルゴリズムは、重心レイヤーである単一の出力を有しています。ファイルパスを入力するか、一時的なファイル名に保存：データ出力が保存されている定義するための2つのオプションがあります。

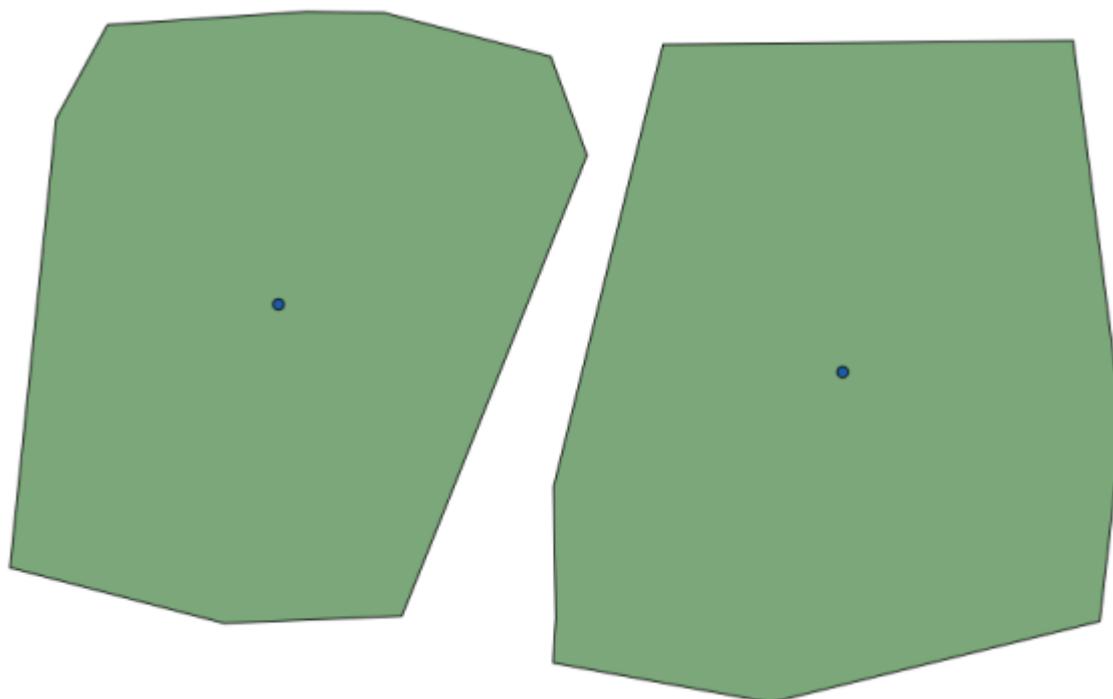
目的地を設定し、一時ファイルに結果を保存しないようにしたい場合には、出力の形式は、ファイル名の拡張子によって定義されます。形式を選択するには、単に対応するファイルの拡張子を選択します（または、あなたが代わりに、直接ファイルパスを入力している場合は、それを追加します）。入力したファイルパスの拡張子は、サポートされているもののいずれか、デフォルトの拡張子（ベクトルものため、通常はラスタレイヤのためのテーブル用 .dbf、 “.tif” と “.shp” と一致しない場合）ファイルパスに追加され、その拡張子に対応するファイル形式は、レイヤまたはテーブルを保存するために使用されます。

後で使用するためにそれらを保存する必要がないので、このガイドのすべての演習では、我々は、一時ファイルに結果を保存します。したい場合は permanent 場所に保存して自由に感じます。

警告: QGIS を閉じると一時ファイルが削除されます。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する場合は、後でプロジェクトをバック開こうとすると、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。

アルゴリズムを実行するためのアルゴリズムダイアログを設定したら、実行 を押します。

次のような出力が得られます。



出力が入力と同じ CRS を有しています。Geoalgorithms は、すべての入力層が同じ CRS を共有し、任意の再投影を行わない前提としています。いくつかの特別なアルゴリズム（例えば、再投影のもの）の場合を除き、出力も同じ CRS ことがあります。我々はすぐにこの詳細が表示されます。

さまざまなファイル形式を使用（例えば、拡張子として “shp” と “geojson” を使用）してそれを保存し、自分自身を試してみてください。それが生成された後レイヤーは QGIS にロードしたくない場合にも、出力パスボックスの下に発見されるチェックボックスをオフにチェックできます。

17.5 より多くのアルゴリズムとデータタイプ

ノート： このレッスンでは、3つのアルゴリズムを実行する他の入力タイプを使用する方法を学習し、自動的に指定したフォルダに保存されるように出力を設定します。

このレッスンのためには、テーブルとポリゴンレイヤーが必要になります。テーブル内の座標に基づいてポイントレイヤーを作成し、各ポリゴン内のポイントの数をカウントしていきます。このレッスンに対応する QGIS プロジェクトを開くと、X と Y 座標を持つテーブルがありますが、ポリゴンレイヤーは何も見つかりません。心配しないで、処理 geoalgorithm を使用して作成されます。

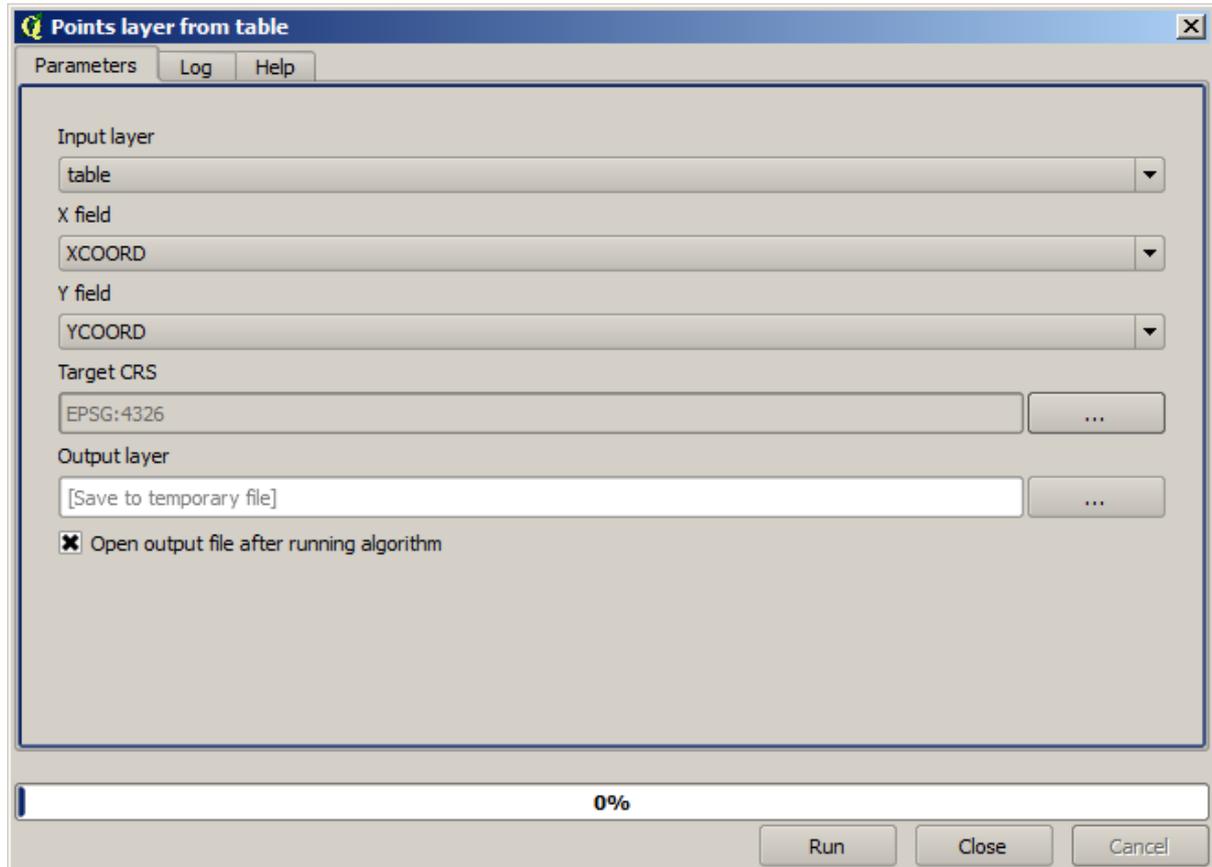
まずやることは、テーブルからポイントレイヤー アルゴリズムを使用して、テーブル内の座標からポイントレイヤーを作成することです。これで、検索ボックスを使用する方法を知っているので、それを見つけることは難しいことはありません。それを実行し、その次のダイアログを取得し、それをダブル-クリックしてください。

このアルゴリズムは、前のレッスンから 1 のように、ただ 1 つの出力を生成し、それは 3 つの入力を持っています：

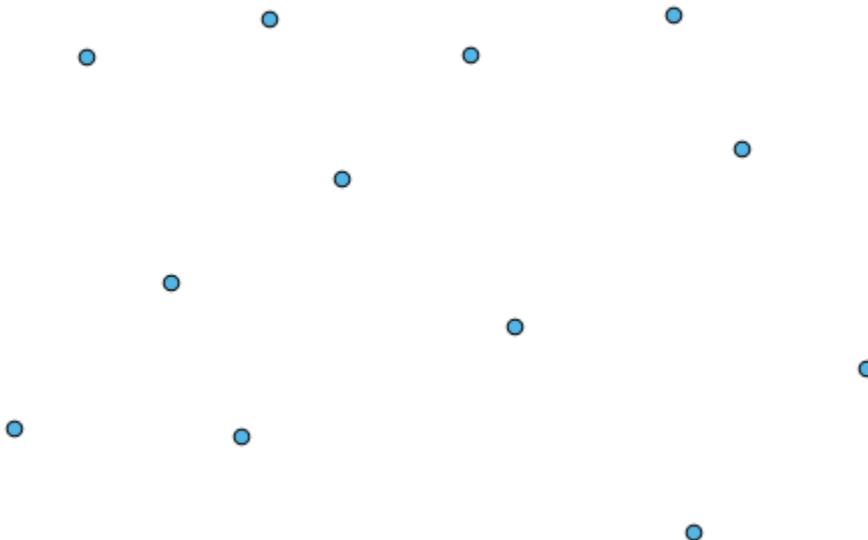
- 表：座標を持つテーブル。ここで授業のデータからテーブルを選択する必要があります。
- X と Y のフィールド：これら 2 つのパラメータは、最初の 1 にリンクされています。対応するセレクトクは、選択されたテーブルで利用可能なこれらのフィールドの名前が表示されます。Y パラメータのための X パラメータ、及び YCOORD フィールドに XCOORD フィールドを選択します。
- CRS：このアルゴリズムでは何も入力レイヤーをとらないので、それに基づいて出力レイヤーへ CRS を割り当てることはできません。代わりに、手動でテーブルの使用中の座標という CRS を選択する

ように求められます。左-手側のボタンをクリックして QGIS CRS セレクタを開き、選択するために出力 CRS として EPSG:4326 を。表中の座標はその CRS にあるのでこの CRS を使用しています。

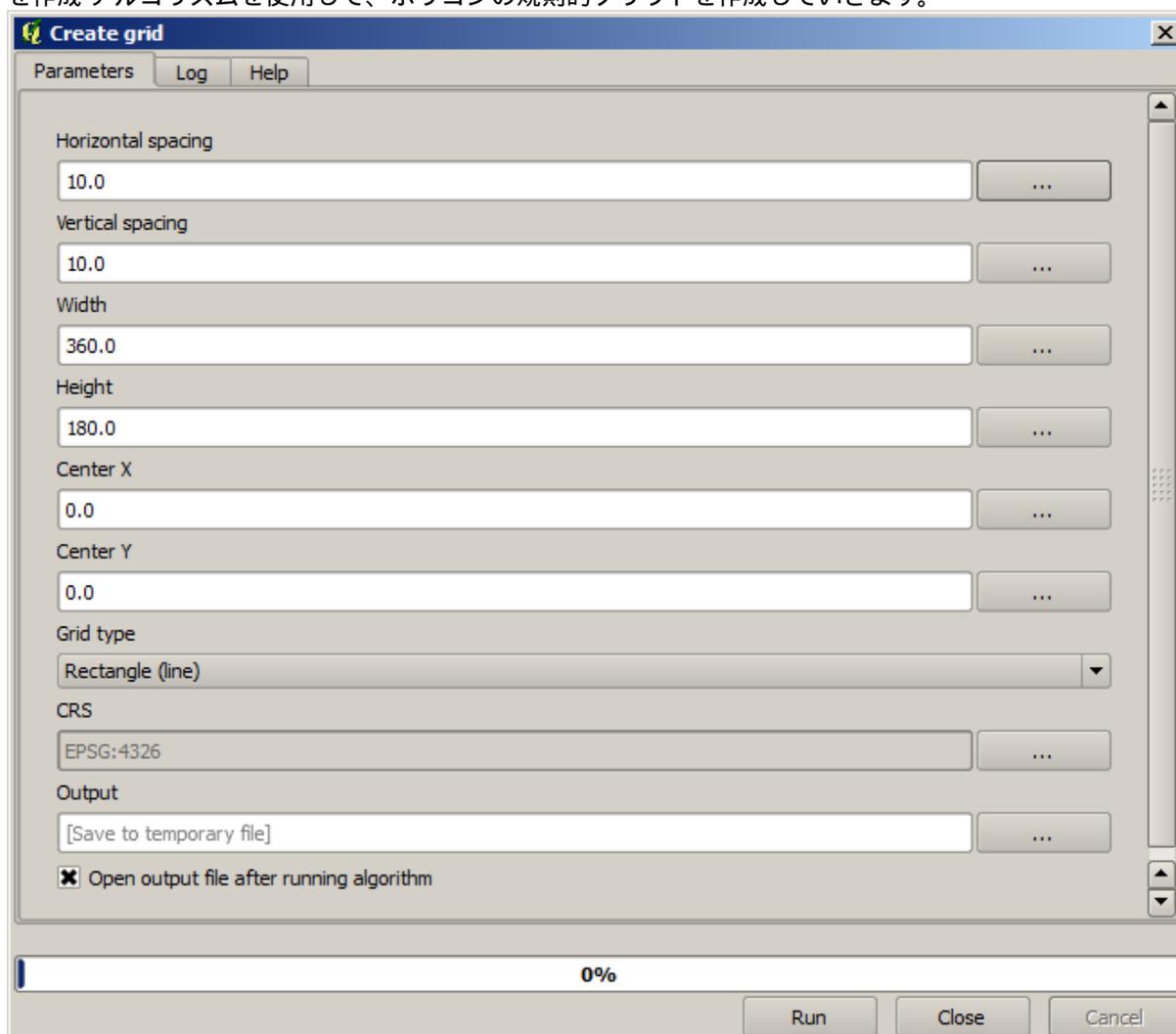
ダイアログは次のようになります。



今、次のレイヤーを得るために 実行 ボタンを押します (新たに作成されたポイント周辺の地図を再入力するフルズームする必要があるかもしれません)。

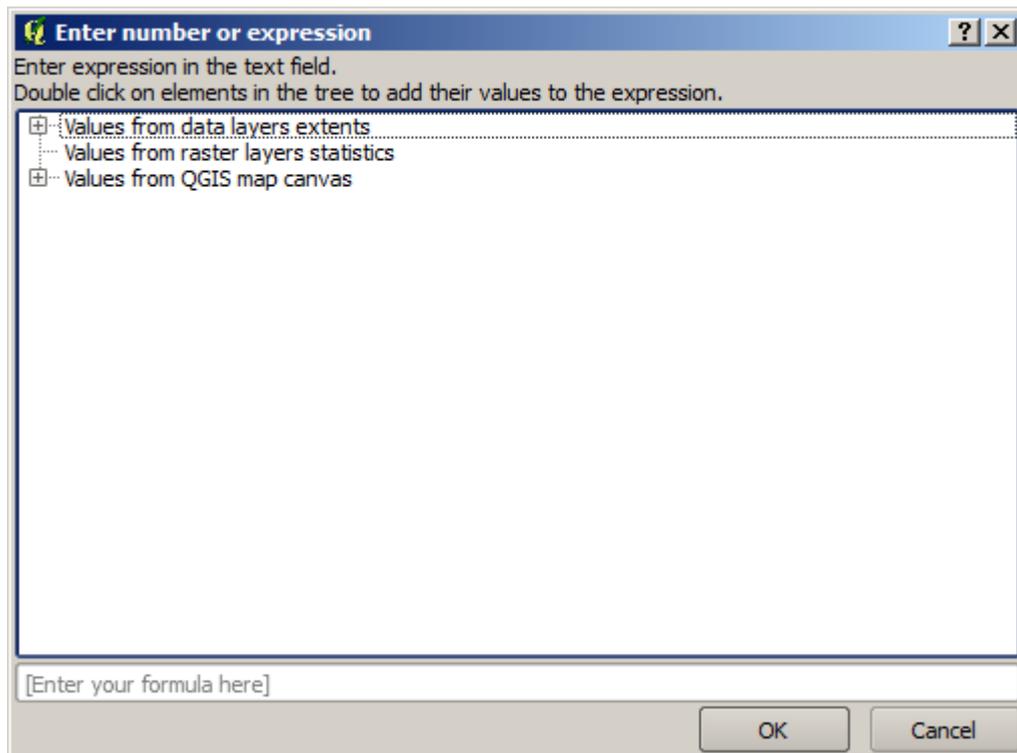


必要とする次のものはポリゴンレイヤです。次のパラメータ] ダイアログボックスを持っている グリッドを作成 アルゴリズムを使用して、ポリゴンの規則的グリッドを作成していきます。



警告: オプションは、QGIS の最近のバージョンでは単純です。X と Y のための最小値と最大値を入力する必要があるだけです (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、すべての数字です。 - 右側次の図のようなダイアログボックスに取得するには、対応するボックスに直接入力するか、右側のボタンをクリック : 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。



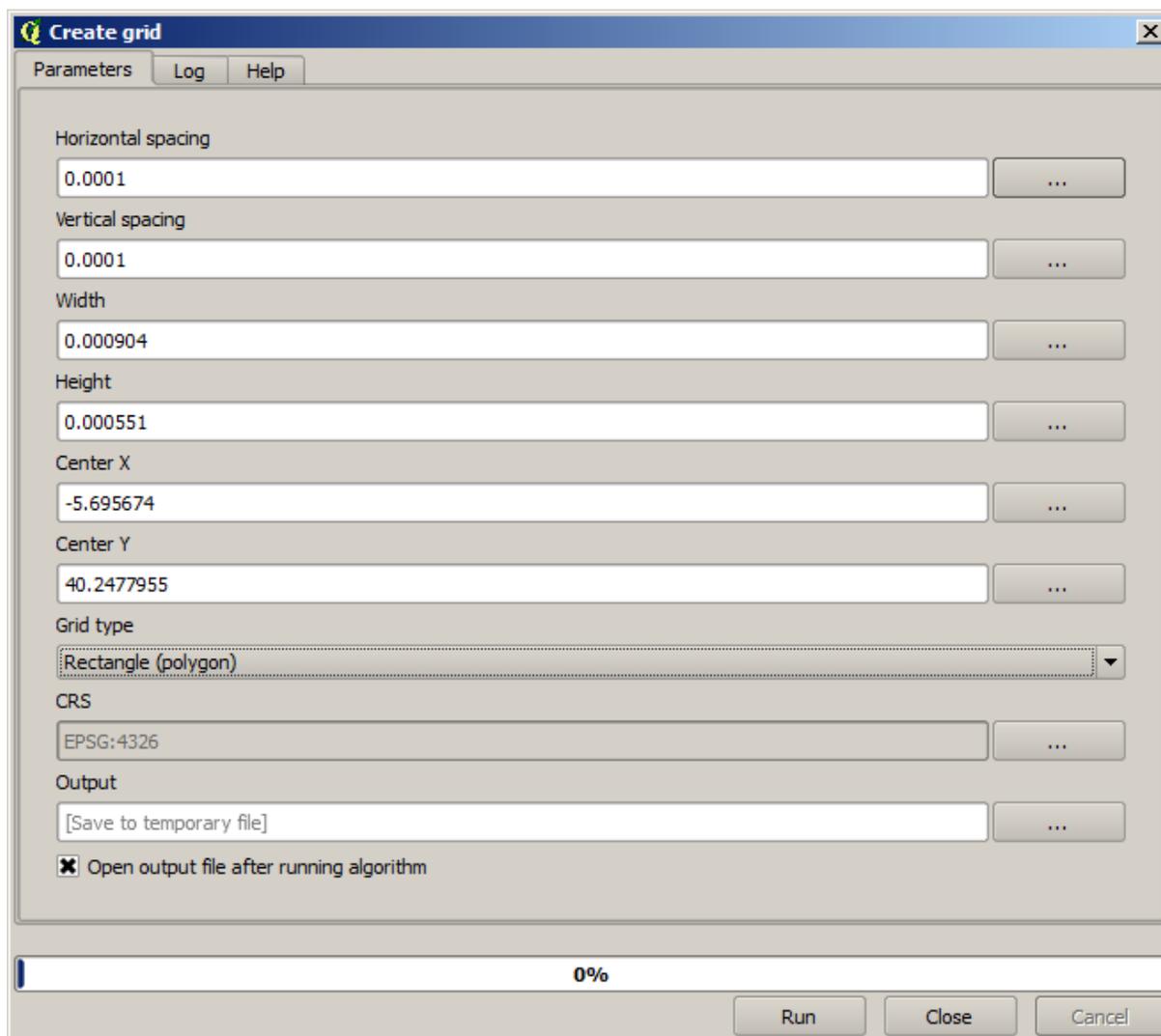
このような “ $11 * 34.7 + 4.6$ ” としての式を入力することができ、その結果が計算され、パラメータダイアログに対応するテキストボックスに置かれますので、ダイアログは、シンプルな電卓が含まれています。また、それはあなたが使用できる定数が含まれており、他の層からの値が入手できます。

この場合、入力点レイヤーの範囲をカバーするグリッドを作成するので、これらのアルゴリズムは、に要するパラメータであるので、グリッドとその幅と高さの中心座標を計算するためにその座標を使用する必要がありますグリッドを作成します。数学の少して、自分が電卓ダイアログと入力点レイヤーからの定数を使用していることをやろう。

タイプフィールドに 長方形（ポリゴン）を選択します。

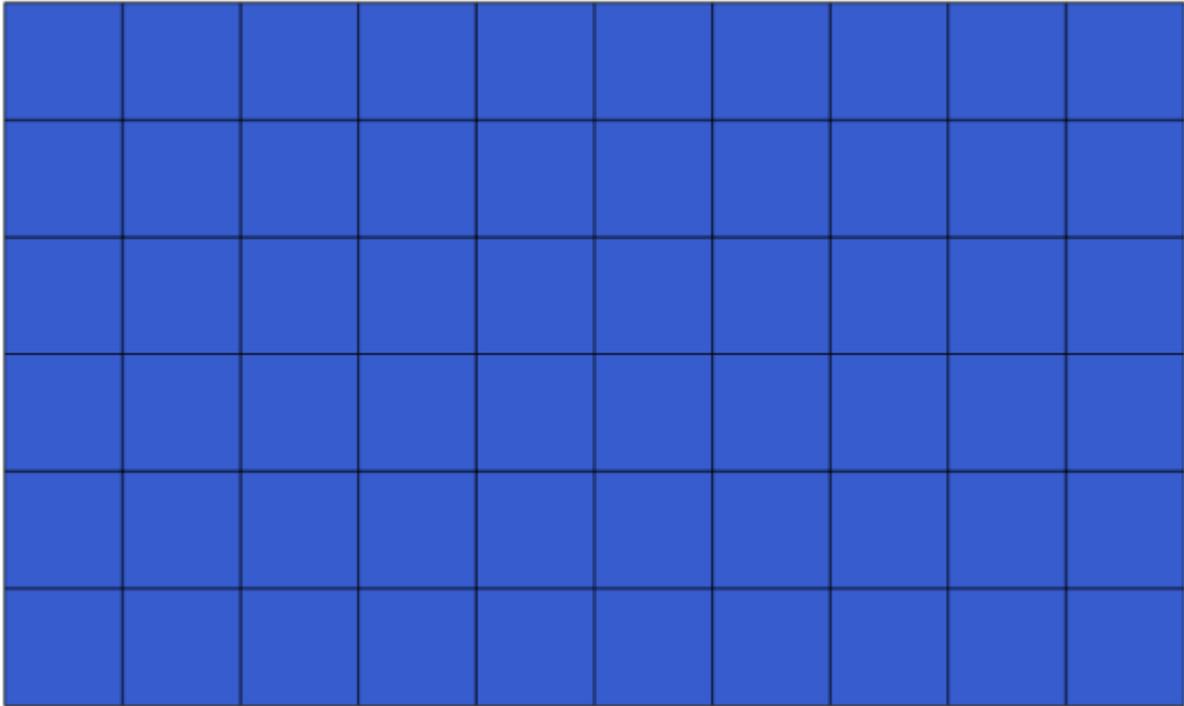
最後のアルゴリズムの場合のように、私たちはここにも CRS を入力する必要があります。私たちは前に行ったように、ターゲット CRS として 4326 : EPSG を選択します。

最後に、あなたは、このようなダイアログのパラメータを持っている必要があります。

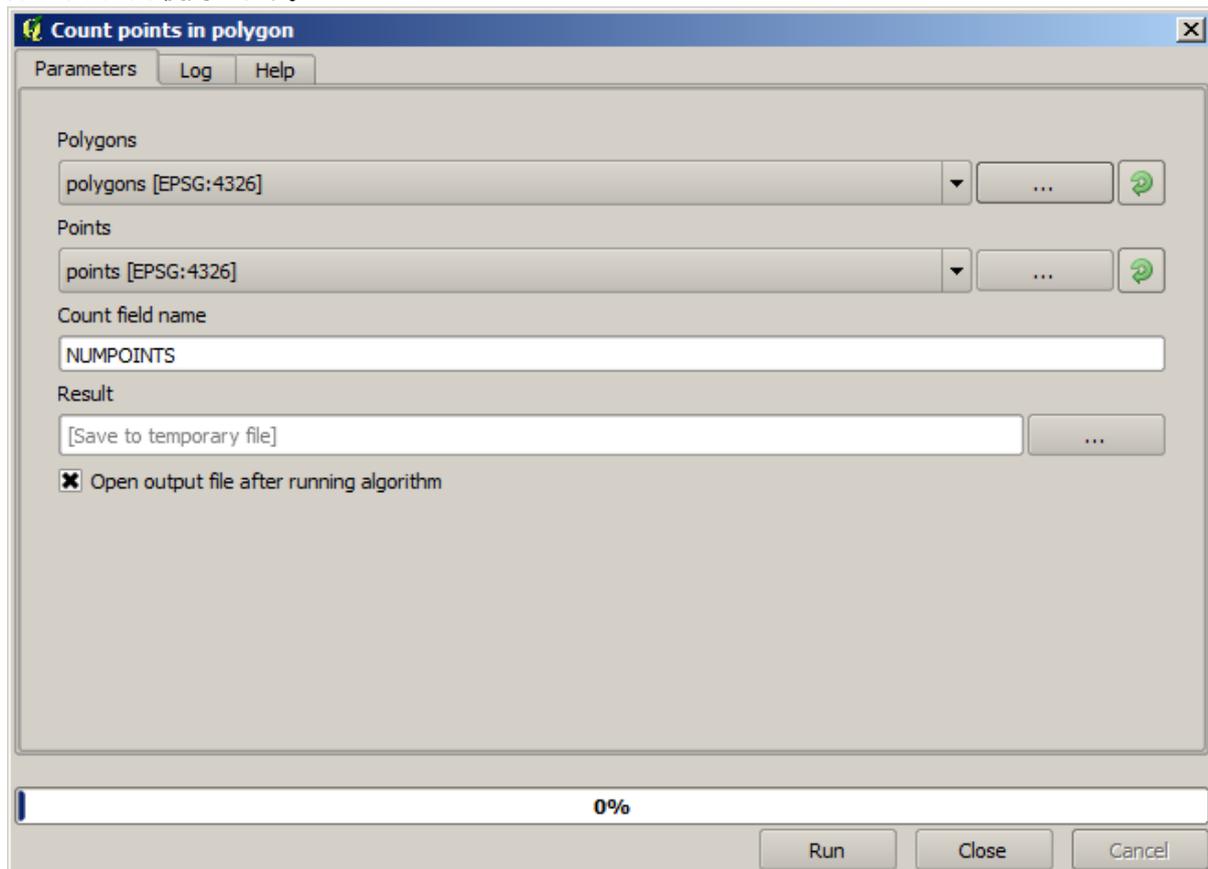


(ベター幅と高さ上の一点の間隔を追加する : 水平間隔 : 0.0001、垂直間隔 : 0.0001、幅 : 0.001004、高さ : 0.000651、センター X : -5.695674、センター Y : 40.2477955) X 中心の場合は少しトリッキーであり、参照 : $-5.696126 + ((-5.695222 + 5.696126) / 2)$

実行 を押すと目盛レイヤーが得られるでしょう。



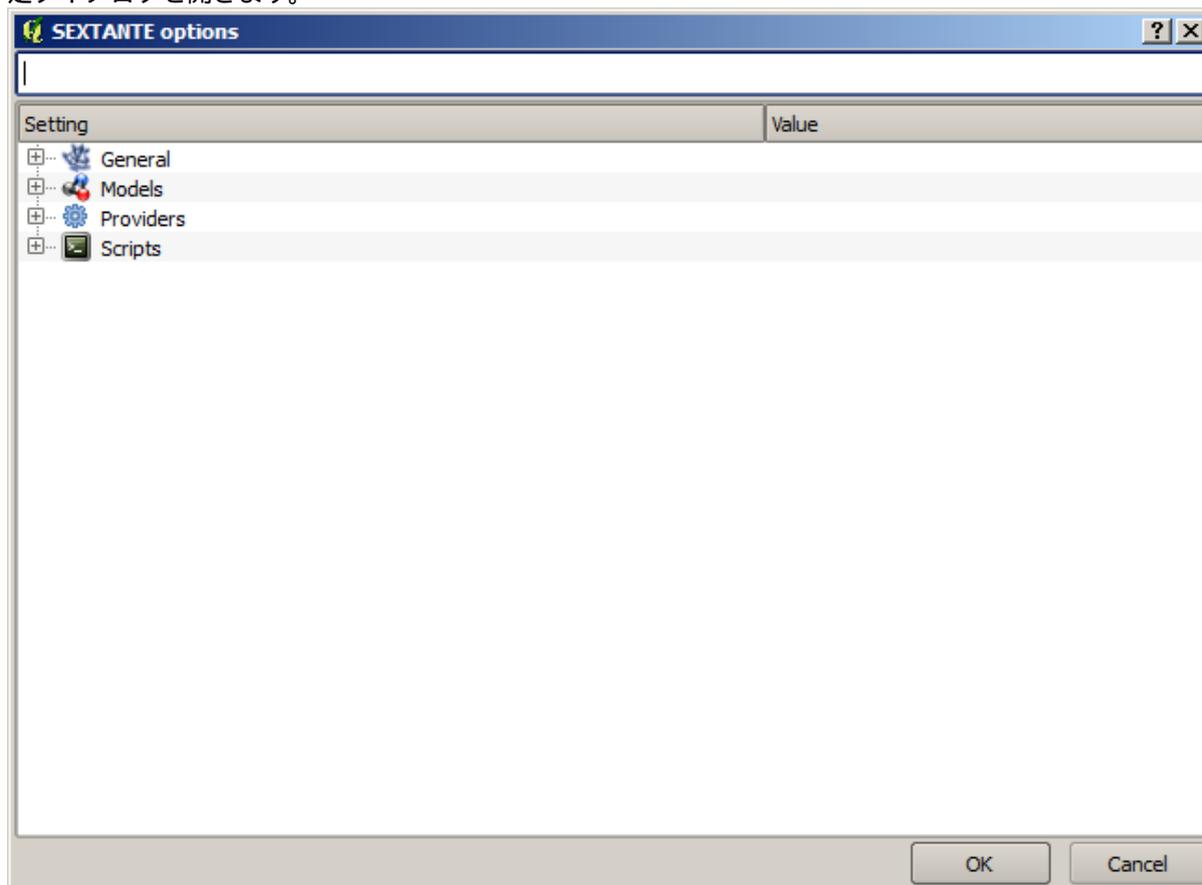
最後のステップは、その目盛りの長方形の各 1 の点を数えることです。ポリゴンでポイントをカウントアルゴリズムを使用します。



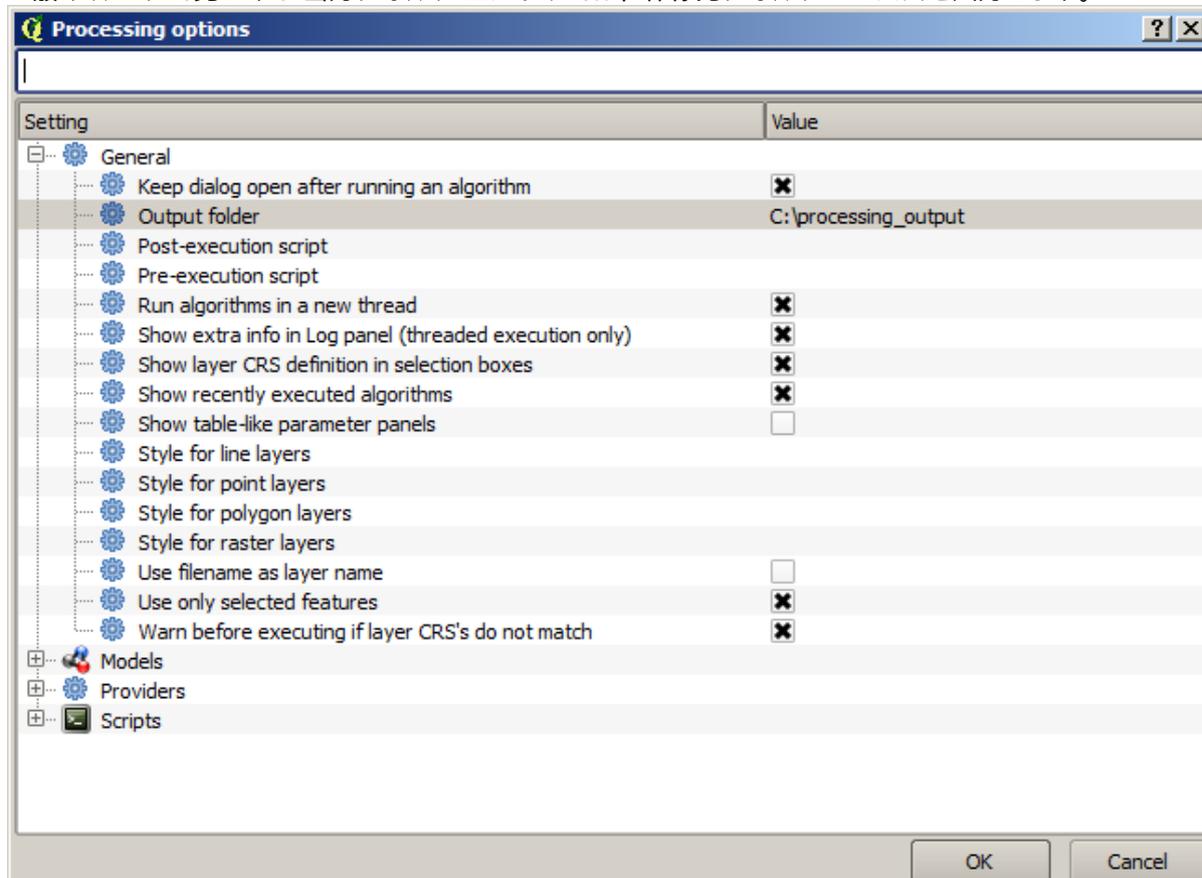
今、私たちは、私たちが探していた結果を持っています。

このレッスンを終了する前に、データを保存永続的にしたい場合にあなたの人生を楽しむための簡単なヒントです。すべての出力ファイルが指定したフォルダに保存したい場合は、フォルダ名を毎回入力する必要はありません。代わりに、処理メニューに移動し、オプションと設定項目を選択します。これは、設

設定ダイアログを開きます。



一般グループで見つける出力フォルダ エントリでは、保存先フォルダへのパスを入力します。



アルゴリズムを実行すると、単にフルパスの代わりにファイル名を使用します。ちょうど今使用しているアルゴリズムの出力パスとして `graticule.shp` を入力した場合例えば、上記のような構成で、結果は `D:\processing_output\graticule.shp` に保存されます。まだ結果を別のフォルダに保存したい場合には、完全パスを入力できます。

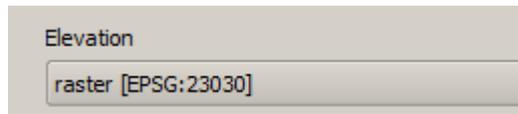
グリッドを作成 アルゴリズムを異なるグリッドサイズで、またグリッドの異なる種類で自分で試してみてください。

17.6 空間参照系。再投影

ノート: この課題で、私たちは空間参照系を利用したプロセッシングの方法を議論します。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 処理ユーザーのための混乱の偉大な源です、だからここに新しいレイヤーを作成するときそれらが地理アルゴリズムでどのように扱われるかについてのいくつかの一般的なルールがあります。

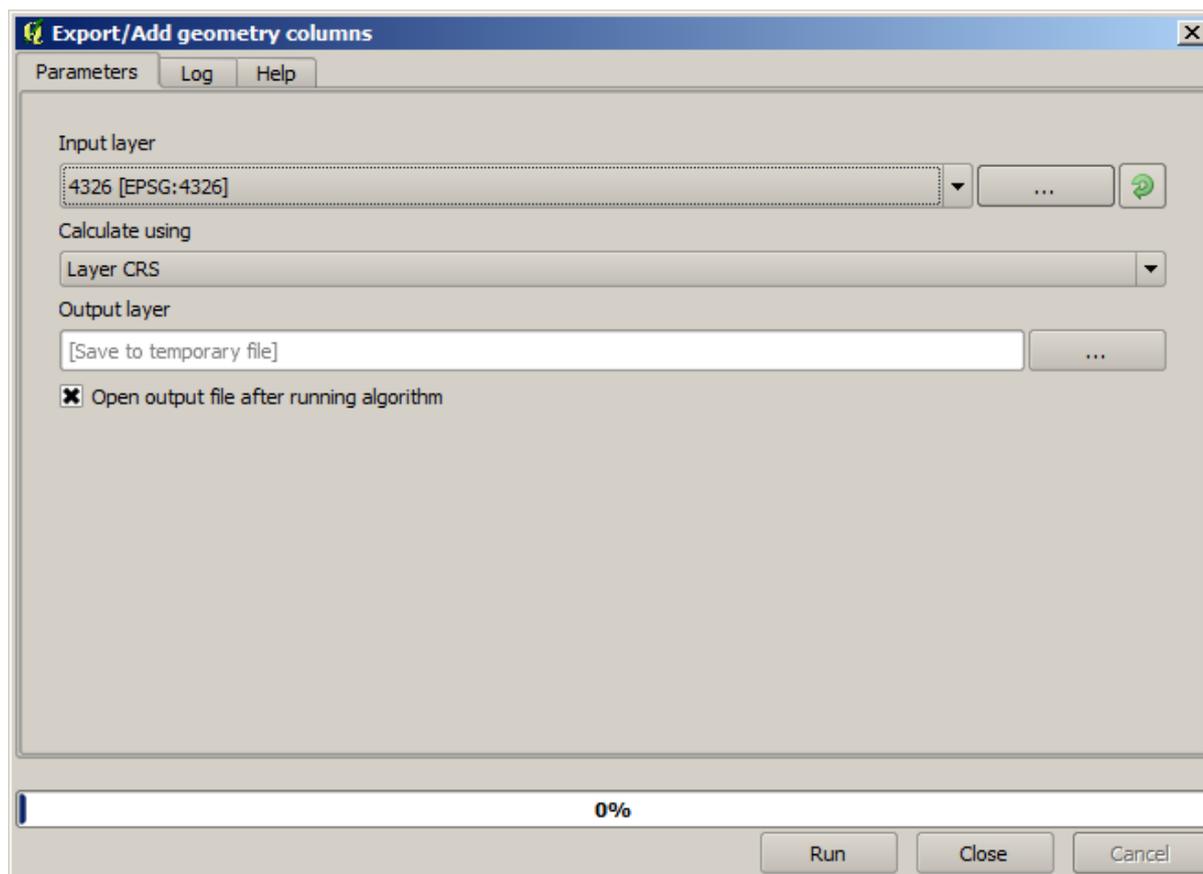
- 入力レイヤーが存在する場合、それは第一レイヤーの CRS を使用します。これは、それらが同じものを持っている必要があるため、すべての入力レイヤーの CRS を想定しています。CRS 年代不一致でレイヤーを使用している場合は、QGIS はそれについて警告を表示します。入力レイヤーの CRS は、パラメータダイアログで、その名前と一緒に表示されていることに注意してください。



- 何も入力レイヤーが存在しない場合、アルゴリズムでは、特定の CRS のフィールドが含まれていない限り（それが目盛アルゴリズムで最後のレッスンで起きたように）、プロジェクト CRS を使用します。

このレッスンに対応するプロジェクトを開き、23030 と 4326 という名前の 2 つのレイヤーが表示されます。それら両方に同じ点が含まれているが、異なるの CRS です (EPSG : 23030 と EPSG : 4326)。QGIS によりプロジェクト CRS (EPSG : 4326) にその場で再投影されたので、それらは同じ場所に表示されますが、それらは実際には同じレイヤーではありません。

ジオメトリ・カラムの出力/追加 を開くアルゴリズム。



このアルゴリズムは、ベクトルレイヤーの属性テーブルに新しい列を追加します。列の内容は、レイヤーの形状の種類によって異なります。点の場合には、各点の X 及び Y 座標で新しい列を追加します。

入力レイヤーのフィールドに見つかる使用可能なレイヤーのリストには、それに対応する CRS を持つ各レイヤーが表示されます。それが意味するのは、それらはキャンバス内の同じ場所に表示されますが異なる方法で処理されることです。4326 レイヤーを選択します。

アルゴリズムの他のパラメータは、アルゴリズムは、それが得られたレイヤーに追加される新たな値を計算するために、座標をどのように使用するかを設定することを可能にします。ほとんどのアルゴリズムは、そのようなオプションを持っている、とだけ座標を直接使用しないでください。彼らがそうであるように、単に座標を使用するには、レイヤー CRS オプションを選択します。これは、ほとんどすべての地理アルゴリズムがどのように機能するかです。

他の 2 つのレイヤーとまったく同じポイントで新しいレイヤーを取得する必要があります。4326 : 右のレイヤーの名前をクリックすると、そのプロパティを開くと、それが入力レイヤーの同じ CRS を共有していることがわかります、それは、EPSG です。レイヤーは QGIS にロードされると QGIS はすでにそれについて知っているので、レイヤーの CRS を入力するよう求められることはありません。

新しいレイヤー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わかります。

	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

そのオプションを選択したため、これらの座標値はレイヤーの CRS に与えられています。しかし、入力 CRS は出力レイヤーの CRS を設定するために使用されているので、別のオプションを選択した場合でも、レイヤーの出力 CRS は、同じだったでしょう。別のオプションを選択すると、入力 1 の CRS に異なるように変更したり、出力に含まれるレイヤーの CRS する結果のポイントの値が異なることが原因ではなくなります。

今、他のレイヤーを使用して同じ計算を行います。他のものと同じ場所に正確にレンダリングされたレイヤーを見つける必要があります、それは EPSG があります：それは入力レイヤーの一つだったので、23030 の CRS を。

その属性テーブルに行けば、私たちが作成した 1 層目のものとは異なる値が表示されます。

	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

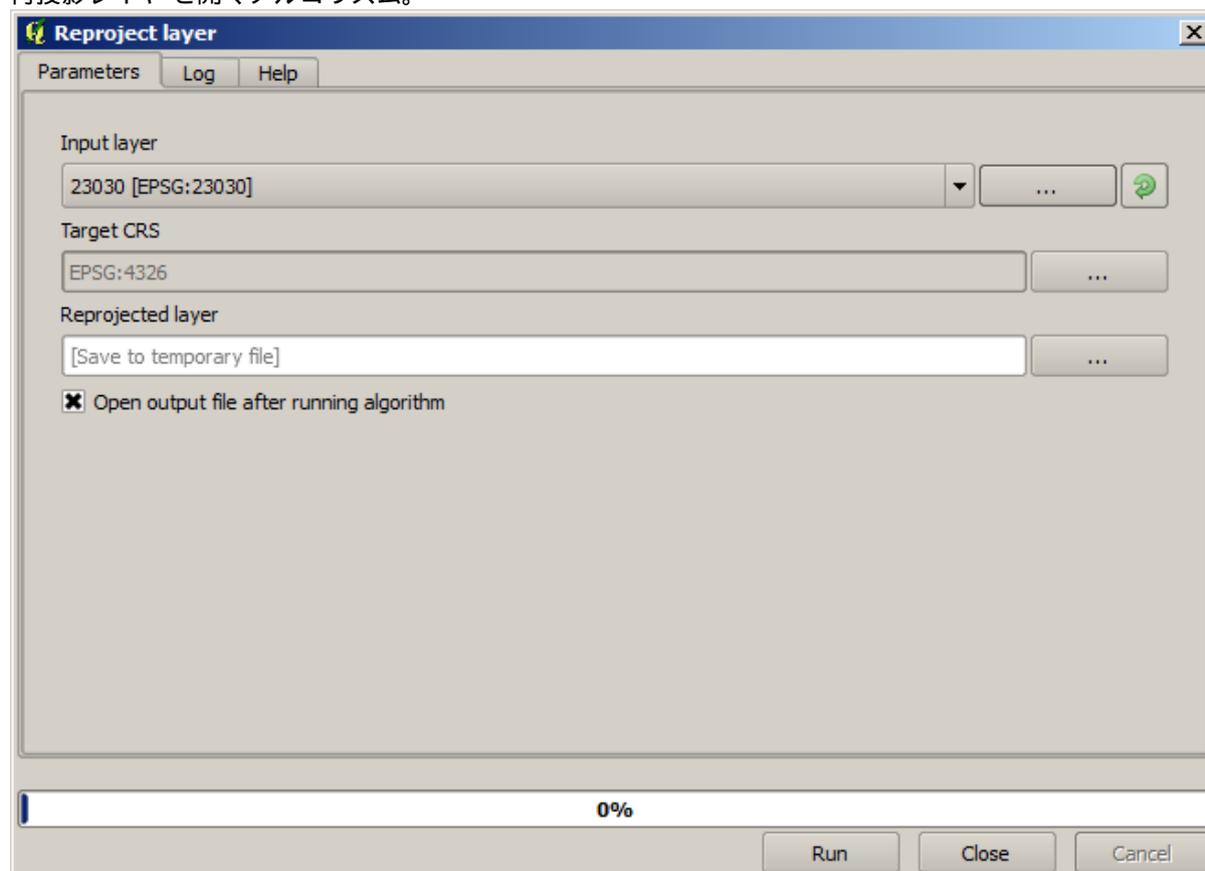
これは、元のデータが異なるため（それは異なる CRS を使用する）であり、これらの座標は、それから取られます。

このことから何を学ぶべき？これらの例の背後にある主なアイデアは、地理アルゴリズムが、それは、元のデータソースにあるように、レイヤーを使用し、完全に QGIS をレンダリングする前にやっているかもしれない再投影を無視するという事です。言い換えれば、キャンバスに何を参照してください信用していませんが、常に元のデータが使用されることを念頭に置いています。この場合は一度に 1 つのレイヤーを使用しているのでそれほど重要ではありませんが、それらの複数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のような）では、一致したり重なったりするように見えるレイヤーは、異なる CRS を持っているかもしれないため、互いに非常に遠くにあるかもしれません。

アルゴリズムは、（次に見る再投影アルゴリズムを除いて）何の再投影を行っていないので、レイヤーが CRS に合致していることを確認することは自己責任です。

CRS を扱う興味深いモジュールは再投影のモジュールです。それは入力レイヤー（再投影するもの）を有

しているのです、それは、特定の場合一し、それは出力いずれかのその CRS を使用しません。
再投影レイヤ を開くアルゴリズム。



入力として、レイヤーのいずれかを選択し、EPSG 選択：先の CRS として 23029 を。アルゴリズムを実行して、入力したものと同一の新しいレイヤーを、取得しますが、異なる CRS を持ちます。QGIS は、その場でそれを再投影されますが、その元の座標が異なっているので、それは、他のもののように、キャンバスの同じ領域に表示されます。入力として、この新しいレイヤーを使用して、コメントを追加座標が前に計算した 2 レイヤーの両方の属性テーブル内のものと異なっていることを検証するアルゴリズム* *エクスポートを実行して、/ジオメトリ列を追加していることがわかります。

17.7 選択

ノート：このレッスンでは、処理アルゴリズムが入力として使用されるベクトルレイヤーでの選択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 での他の分析プラグインとは異なり、処理 geoalgorithms 任意の「のみ選択した機能を使用する」のチェックボックスまたは同様に見つけることができません。選択に関する行動は、全プラグインとそのすべてのアルゴリズムのためではなく、各アルゴリズムの実行のために設定されています。ベクトル層を使用した場合のアルゴリズムは、以下の簡単なルールに従ってください。

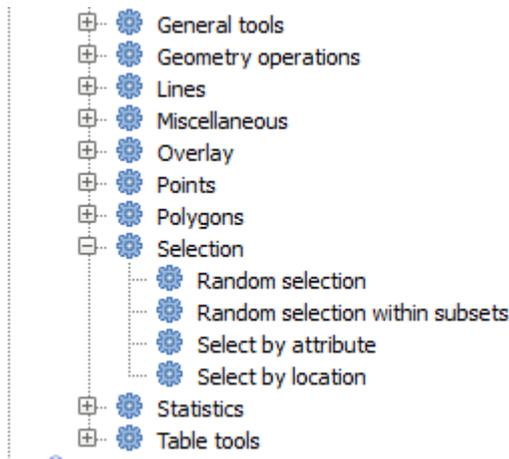
- レイヤーに選択がある場合は、選択した地物のみが使用されます。
- 選択範囲がない場合は、すべての地物が使用されます。

>オプション --> General'メニュー '処理：menuselection：あなたは、関連するオプションを選択解除すると、この動作を変更することができますので、予めご了承ください。

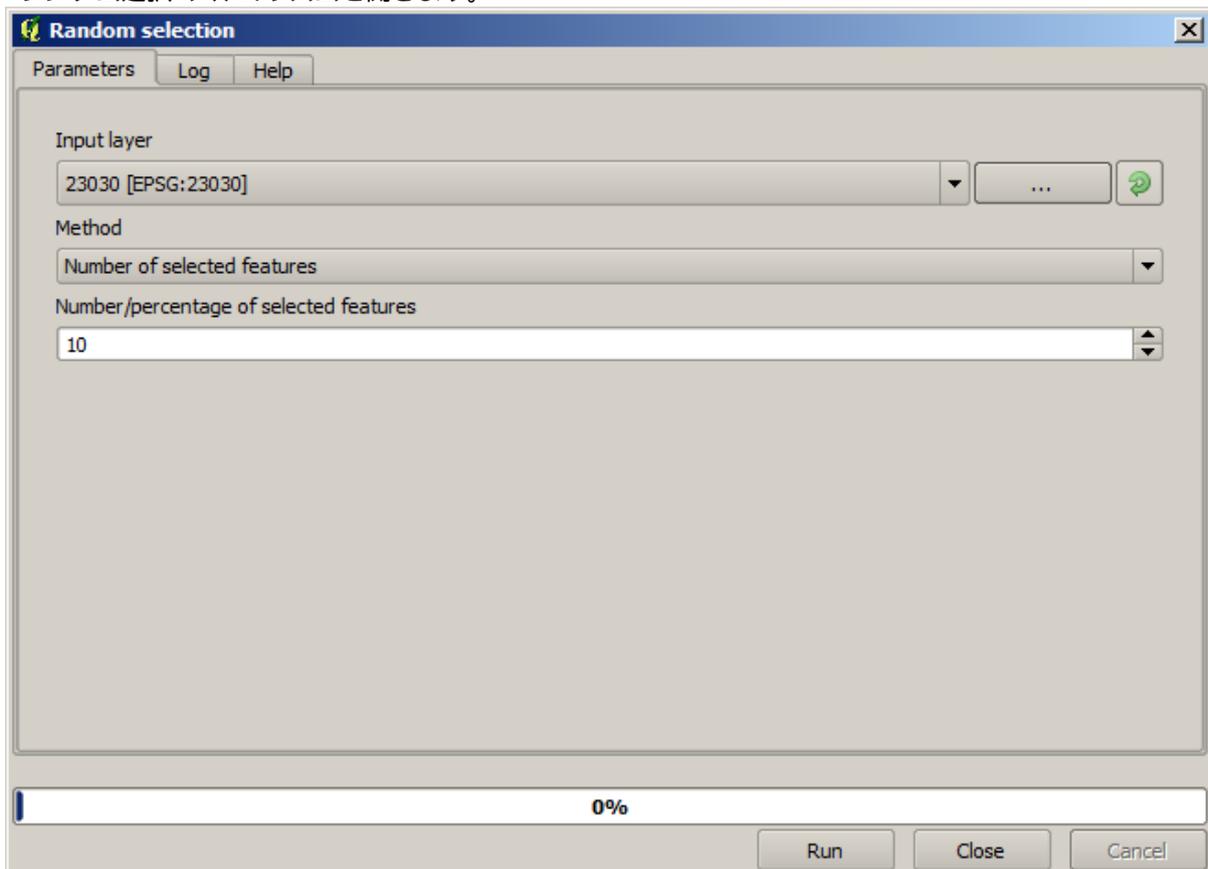
私たちが最後の章で使用される層のいずれかにいくつかのポイントを選択し、それらの上に再投影アルゴリズムを実行していることを自分でテストすることができます。得られる層は、元のレイヤーからすべて

のポイントが含まれていることになり、何も選択が、ありませんでしたし、あなたが取得する再投影層は、選択された点のみが含まれます。

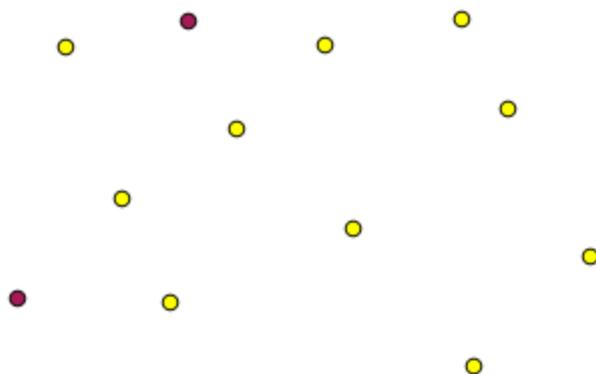
選択を作成するには、QGIS で利用可能な方法やツールのいずれかを使用できます。しかし、geoalgorithm を使用してもそれはできます。選択を作成するためのアルゴリズムは、*ベクトル/選択*中のツールボックスにあります



*ランダム選択*アルゴリズムを開きます。



デフォルト値を残して、それが現在のレイヤーから 10 点を選択します。



このアルゴリズムは、任意の出力を生成しないことがわかりますが、入力層（ない層自体が、その選択）を変更します。他のすべてのアルゴリズムは、新たな層を生成し、入力層を変更することはありませんので、これは珍しい行動です。

選択はデータ自体の一部ではありませんが、QGIS 内に存在するだけの何かですので、これらの選択アルゴリズムは、QGIS 内で開かれているレイヤーを選択するのに使用する必要があるだけで、対応するパラメータ値ボックスに見つかるファイル選択オプションでない。

私達はちょうど行った選択は、選択アルゴリズムの残りの部分によって作成されたもののほとんどのよう、また、QGIS から手動で行うことができますので、そのためのアルゴリズムを使用して上の点であるか疑問に思うことがあります。今、これはあまり意味がないかもしれないが、それ以降のモデルとスクリプトを作成する方法について説明します。（処理ワークフローを定義する）モデルの途中で選択をしたい場合は、唯一の `geoalgorithm` をモデルに追加でき、他の QGIS 要素と操作を追加できません。これは、いくつかの処理アルゴリズムは、他の QGIS 要素で利用できる機能を複製する理由です。

これまでに、単に選択が処理 `geoalgorithms` を使用して行うことができることを覚えて、その選択が存在する場合のアルゴリズムは、選択機能を使用すること、またはすべてがそうでない場合があります。

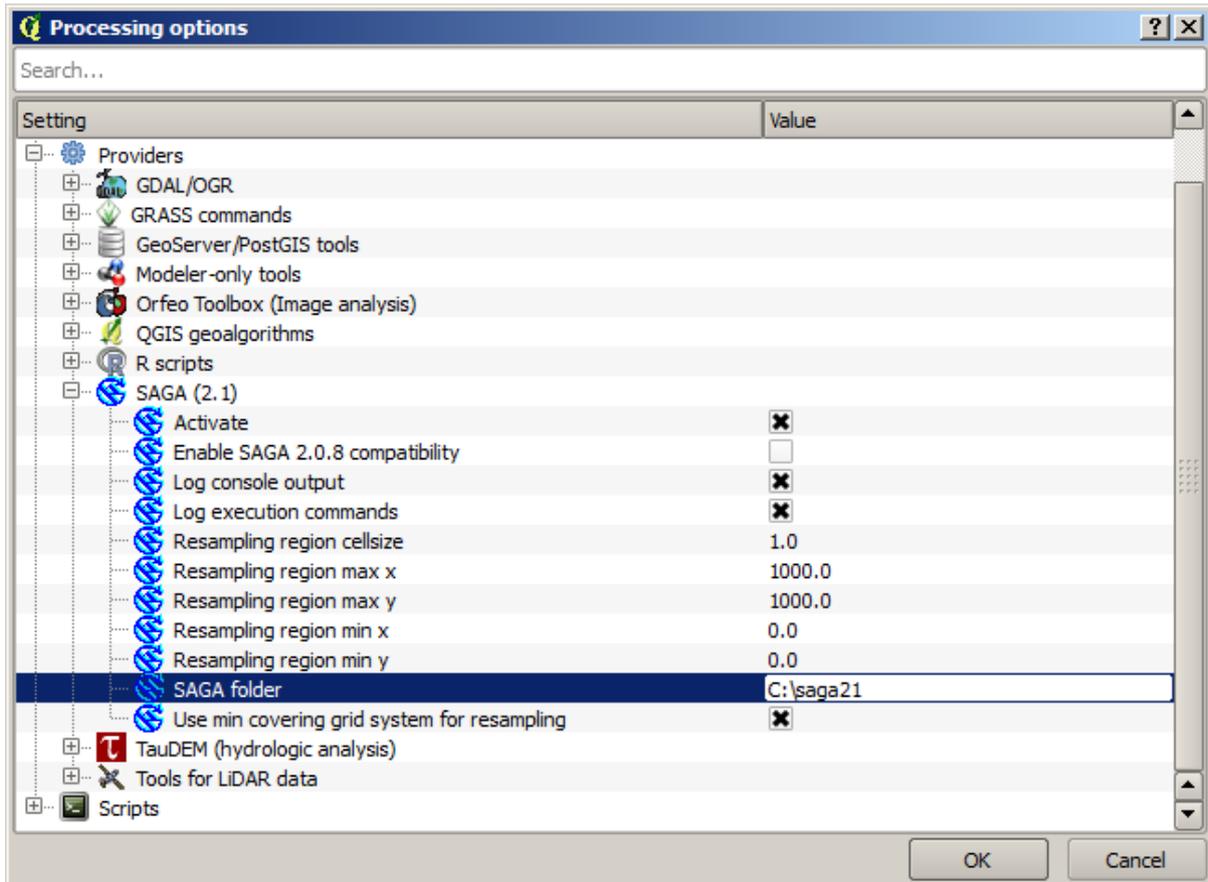
17.8 外部のアルゴリズムを実行します

ノート：このレッスンでは、メインアルゴリズムプロバイダーの一つであり、サードパーティのアプリケーション、特に 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。

これまでに実行したすべてのアルゴリズムは、処理フレームワークの一部です。すなわち、それらはネイティブアルゴリズムプラグインで実装され、ちょうど自分自身が実行されるプラグインのような QGIS によって運営されています。しかし、処理フレームワークの最大の特徴の一つは、それが外部のアプリケーションからのアルゴリズムを使用して、それらのアプリケーションの可能性を拡張できることです。QGIS から簡単にそれらを使用して、それらを実行するために、QGIS データを使用できるようにこのようなアルゴリズムは、ツールボックスに包んで含まれています。

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。特別な関心の一つのアルゴリズムプロバイダは、SAGA（自動地理空間分析のためのシステム）です。まず、QGIS が正しく SAGA を呼び出すことができるように、すべてのものを設定する必要があります。これは難しいことではありませんが、それは、それがどのように動作するかを理解することが重要です。それぞれの外部アプリケーションは、独自の構成を有しており、この後の同じマニュアルに、我々は他のもののいくつかについて話しますが、SAGA は、当社の主要なバックエンドになるだろうので、私たちはここでそれを説明します。

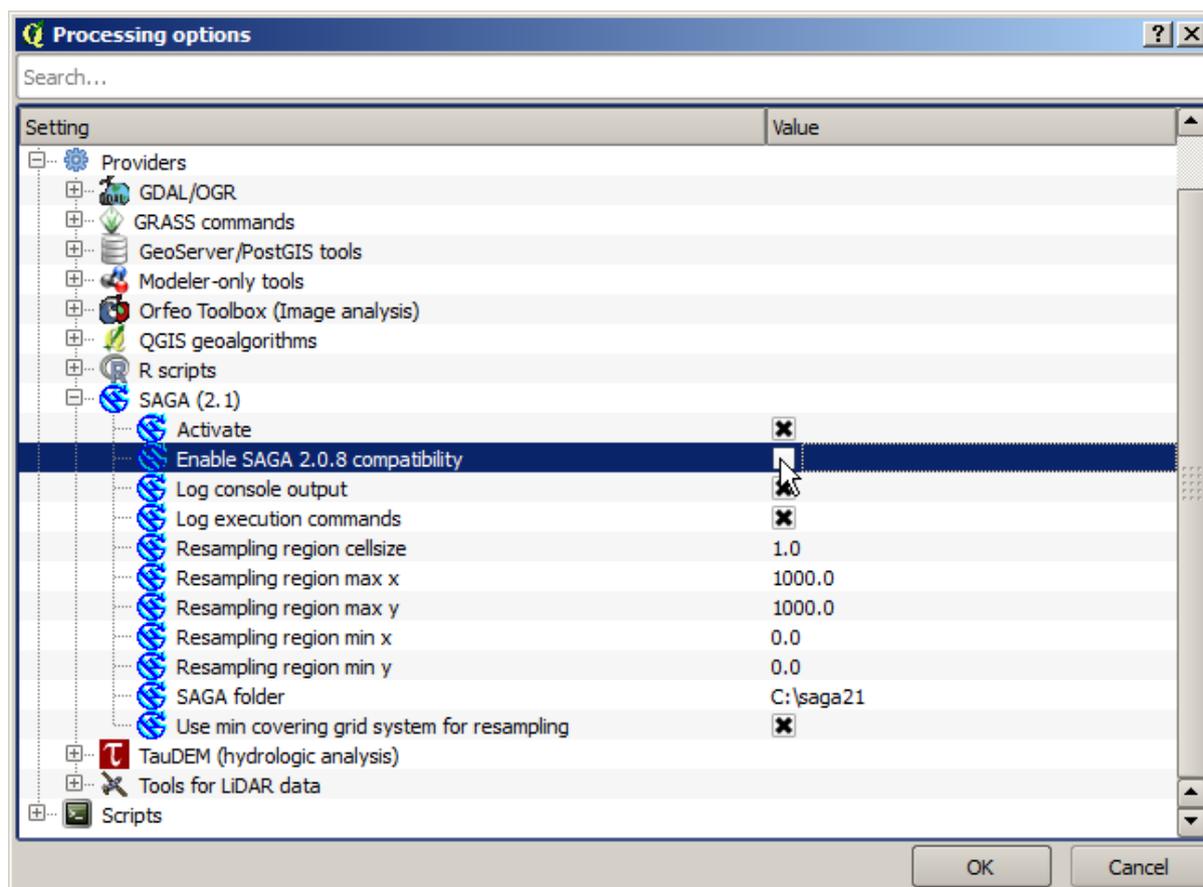
Windows である場合は、外部のアルゴリズムで動作するための最良の方法は、スタンドアロンのインストーラを使用して QGIS をインストールすることです。それは SAGA を含むすべての必要な依存関係を、インストールの世話をしますので、それを使用している場合は、行うには他には何もありません。設定ダイアログを開き、プロバイダ/SAGA グループに行くことができます。



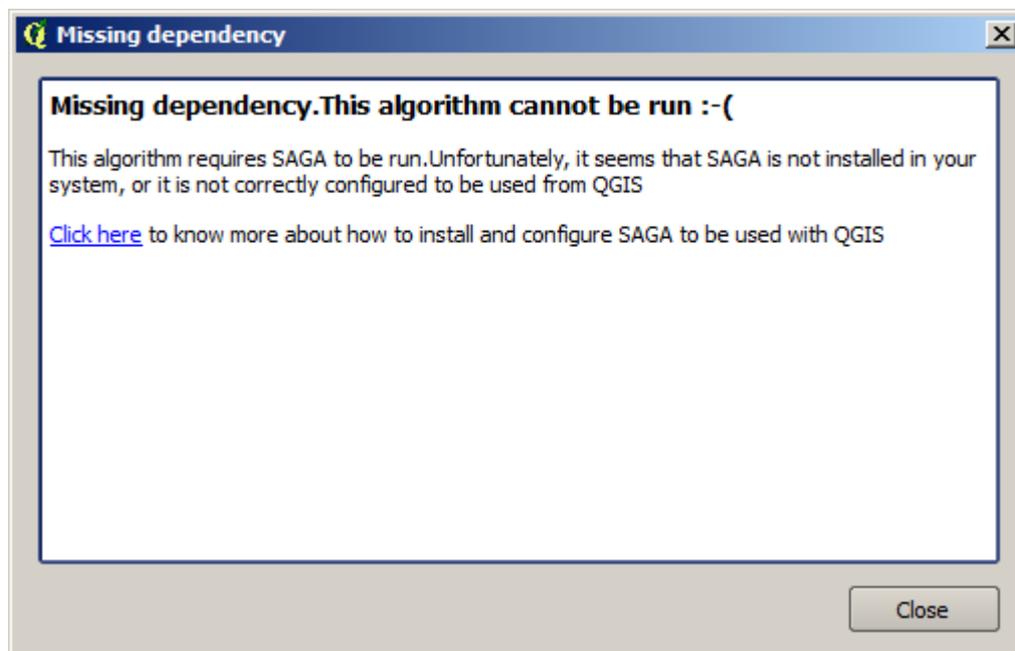
SAGA パスが既に設定され SAGA がインストールされているフォルダを指しすべきです

スタンドアロンインストーラを使用していない QGIS をインストールしている場合は、そこ（個別にインストールされている必要があります）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ている] です。

Linux を使用しているケースでは、処理構成で 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（ちょうどコンソールを開いて、それをチェックするために `saga_cmd` を入力）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。Linux では、SAGA のターゲットバージョンも 2.1 ですが、（そのよう OSGEO ライブ DVD など）一部のインストールでは、だけ 2.0.8 利用できる可能性があります。そこにいくつかの 2.1 のパッケージが用意されていますが、それらは一般的にインストールされていないといくつかの問題があるかもしれないので、より一般的と 2.0.8 の安定を使用することを好む場合は、下に、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、それを行うことができます* SAGA *基。

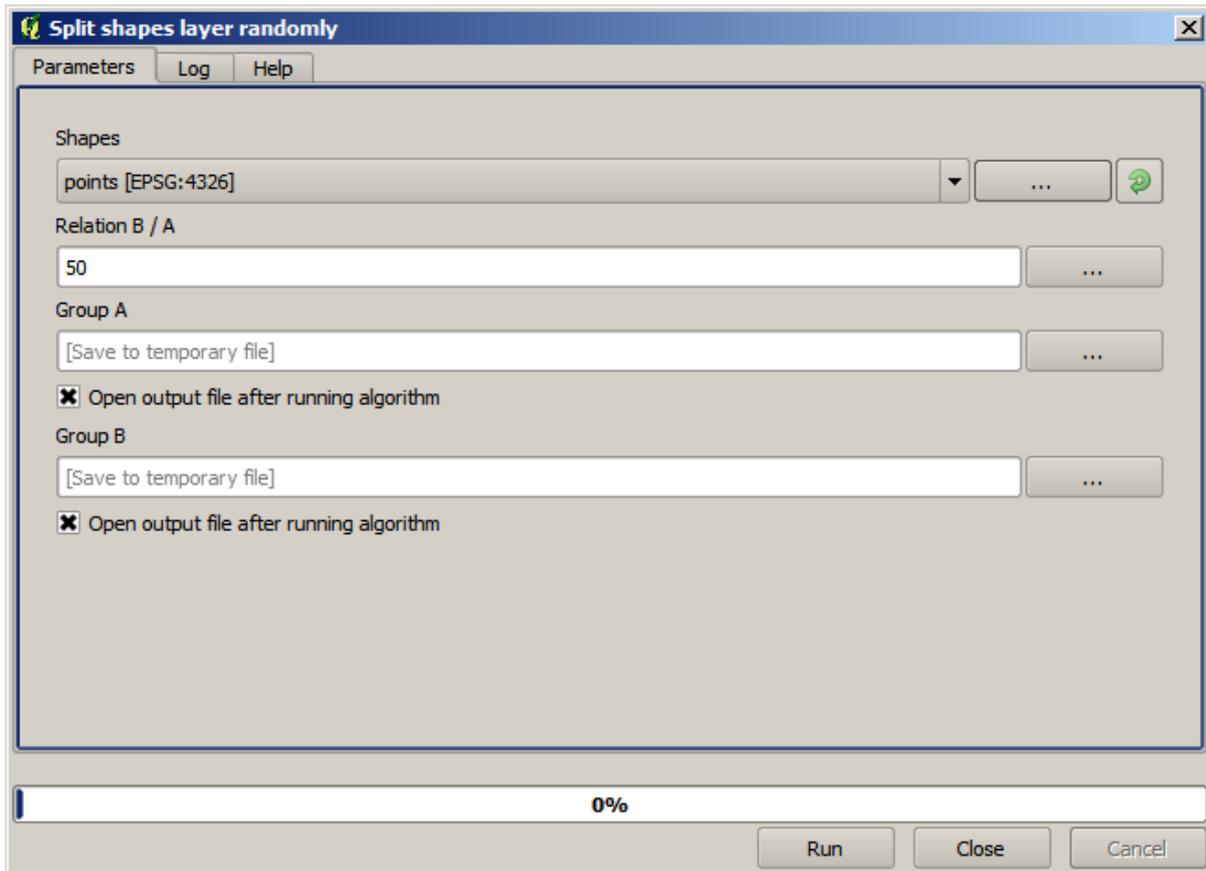


SAGA がインストールされたら、他のアルゴリズムと同様に、その名前を SAGA アルゴリズムをダブルクリックし起動できます。我々は単純化されたインターフェイスを使用しているため、SAGA 上または別の外部のアプリケーションに基づいているアルゴリズムは知らないが、二重に起こる場合 - インストールされていない彼らと、対応するアプリケーションのいずれかをクリックし、次のようなものが表示されますこの。

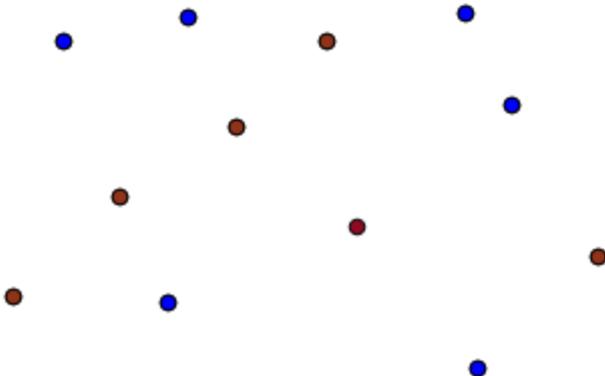


我々の場合には、その SAGA と仮定すると、正しくインストールされ、構成され、このウィンドウは表示されません、代わりにダイアログのパラメータを取得します。

シェイププレイヤーをランダムに分割と呼ばれるもの、SAGA - ベースのアルゴリズムで試してみましょう。

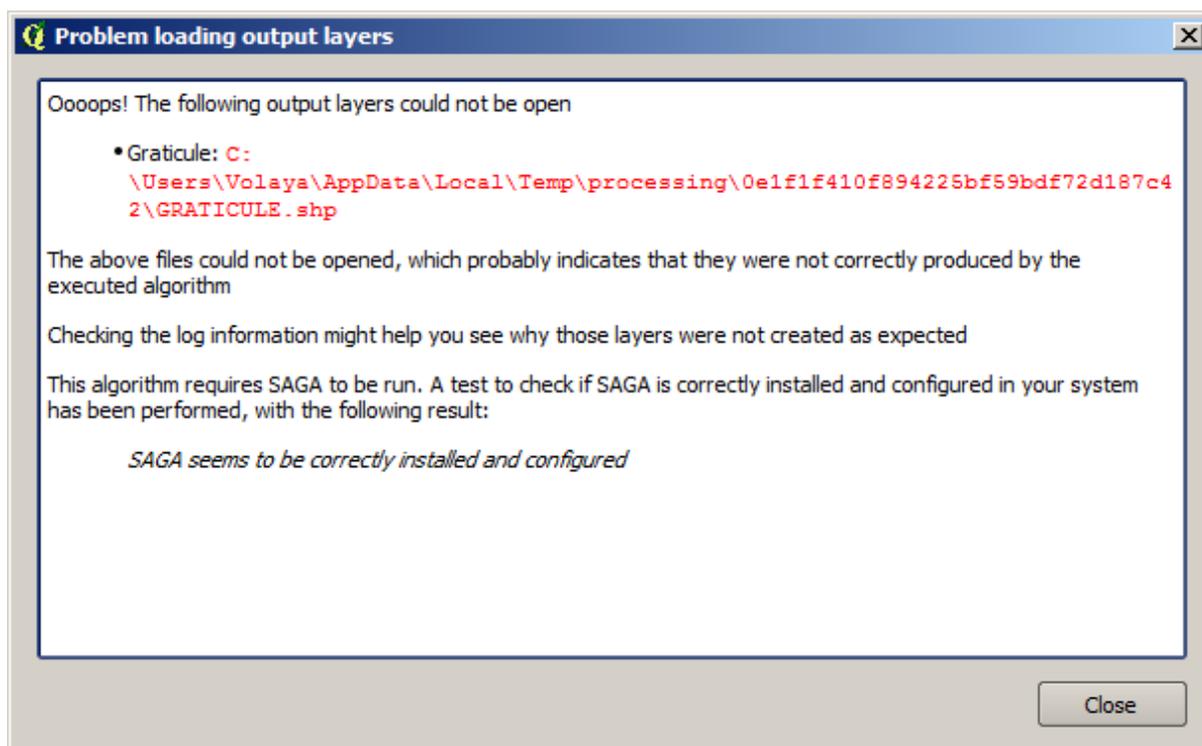


入力として、このレッスンに対応したプロジェクト内のポイントレイヤー、およびデフォルトのパラメータ値を使用して、このような何かを取得します（分割がランダムであるので、結果は異なる場合があります）。



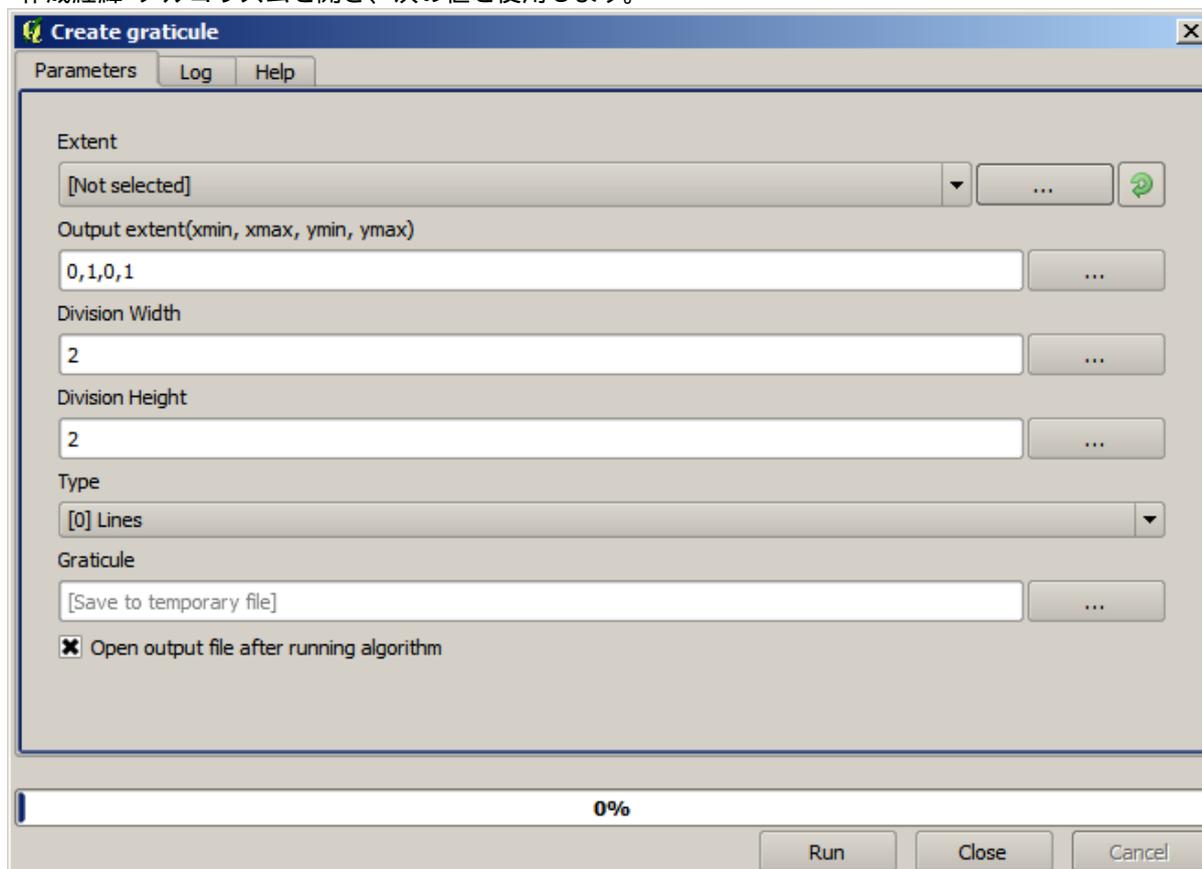
入力レイヤーは、2 レイヤー、それぞれポイントの同じ数に分割されています。この結果は、SAGA によって計算され、以降 QGIS で撮影したと QGIS プロジェクトに追加されました。

ベースのアルゴリズムと、我々は以前に実行を持っている他人の 1 - すべてがうまくいけば、この SAGA 間のいずれかの違いに気付くことはありません。しかし、SAGA は、何らかの理由で、結果を生成し、QGIS を期待しているファイルを生成しないことができない場合があります。その場合、問題 QGIS プロジェクトに結果を追加し、このようなエラーメッセージが表示されますが存在するであろう。



この種の問題は、SAGA（または私達は処理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る場合でも、起こるかもしれない、それらに対処する方法を知っておくことが重要です。のは、これらのいずれかのエラーメッセージを生成してみましょう。

*作成経緯*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ため、SAGA は、任意の出力を生成することが

できません。言い換えれば、パラメータ値が間違っているが、SAGA はそれらを取得し、目盛りを作成しようとするまで、彼らはチェックされません。それはそれを作成できませんので、それが期待されるレイヤーを生成しません、上記のようなエラーメッセージが表示されます。

ノート: SAGA >= 2.2.3 では、コマンドは自動的に間違った入力データを調整しますので、あなたがエラーを取得していないでしょう。エラーを誘発するために、分割のための負の値を使用します。

この種の問題を理解することは、それらを解決し、何が起きているかの説明を見つけましょう。エラーメッセージで見ることができるよう、テストは、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、SAGA との接続が正しく機能していることを確認するために行われます。これは、同様に SAGA にするだけでなく、他の外部アプリケーションに限らず適用されます。

次のレッスンでは、geoalgorithms 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、ご紹介し、そしてこのような問題が現れたときに詳細を取得する方法について説明します。

17.9 処理ログ

ノート: このレッスンでは、処理ログを説明します。

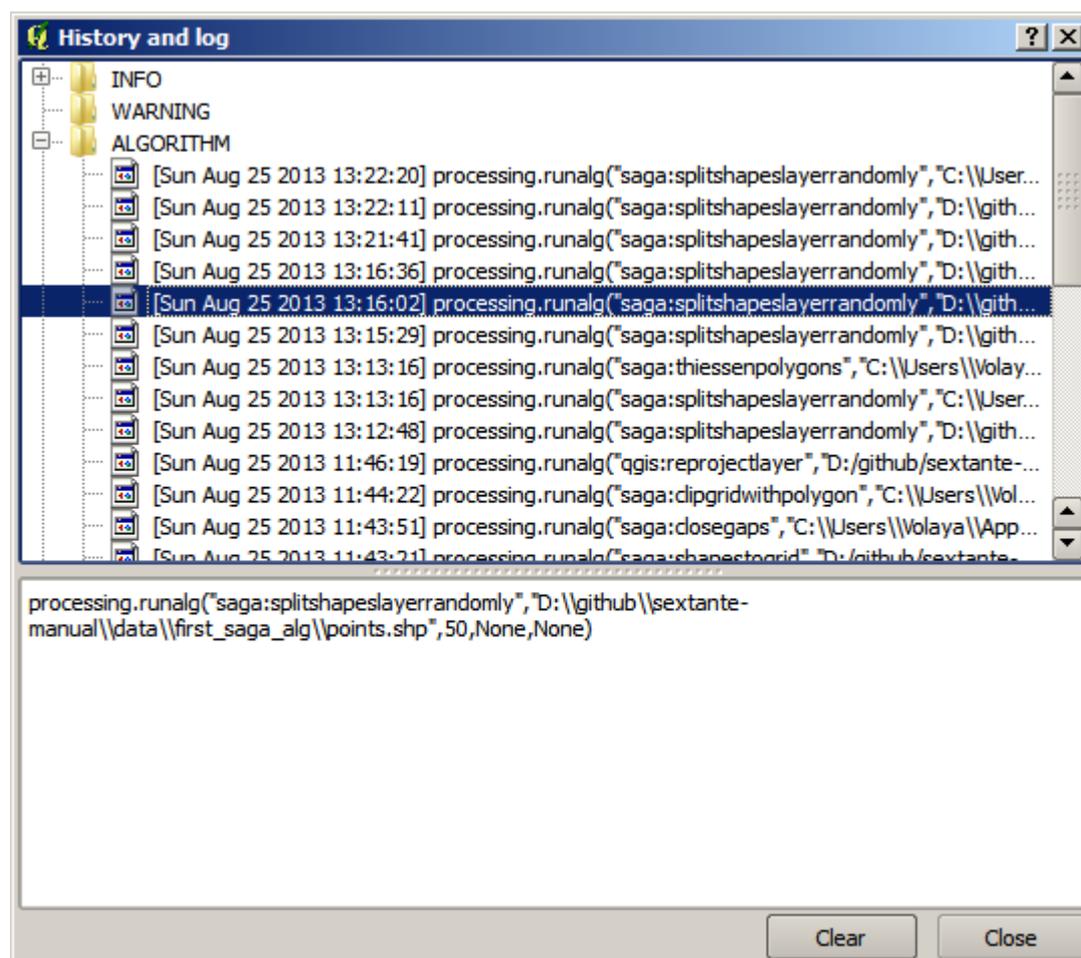
処理フレームワークで実行されるすべての分析は、QGIS のロギングシステムに記録されます。ロギングシステムはまた、いくつかの対話を実装しているので、以前の操作を実行します - これは、彼らが発生したときの問題を解決するために、処理ツールを用いて行われていた、とも再するかについての詳細を知ることができます。

ログを開くには、QGIS のステータスバーに、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これらのアルゴリズムは、通常、このエントリには、そのアプリケーションのコンソール出力をログ。あなたはそれを見ている場合は、私たちは実行（および入力データが正しくありませんでしたので、それが実行に失敗）SAGA アルゴリズムの出力はここに格納されていることがわかります。

これは、何が起きているか理解するのに便利です。上級ユーザーは、アルゴリズムが失敗した理由を見つけるために、その出力を分析できるようになります。上級ユーザーでない方は、これは外部のソフトウェアのインストール中に問題または提供されたデータの問題であるかもしれない、他の人が持っている問題を診断するのを助けるために有用であろう。

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差作成されますが、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がわからない場合は、定期的に警告のこのタイプをチェックすることをお勧めします。

処理メニューから、履歴セクションの下に、アルゴリズムを見つけることができます。それらは、（本書で後述する）コンソールから GUI から実行されていない場合でも実行されているすべてのアルゴリズムは、コンソールコールとして、このセクションに格納されます。それはアルゴリズムを実行するたび、コンソールコマンドがログに追加されたことを意味し、作業セッションの完全な履歴を持っています。これはその履歴がどのように見えるかです：



アルゴリズムの構文について学ぶために、コンソールで作業を開始するとき、これは非常に便利です。それは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を使用するでしょう。

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再実行できます。これは、すでに前にした作業を複製する簡単な方法です。

たとえば、次のようにしてみてください。このマニュアルの最初の章に対応するデータを開き、アルゴリズムが説明して実行します。今、ログダイアログに移動して、あなただけ実行したアルゴリズムに対応して、リスト内の最後のアルゴリズムを見つけます。ダブル - 新しい結果はちょうどあなたが、それは通常のダイアログを使用して、ツールボックスからそれを呼び出す実行したときのように、生成する必要があり、それをクリックしてください。

17.9.1 上級編

アルゴリズムを変更することもできます。ただそれをコピーし、プラグイン->Python の console を開き、インポートクラス -> [インポート処理 class をクリックし、それから貼り付けて分析（意志のテキストを変更します）を再実行する。結果のファイルを表示するには、iface.addVectorLayer（' /パス/ filename.shp', ' 凡例中のレイヤ名', ' OGR' ）とタイプします。それ以外の場合は、processing.runandload を使用できます。

17.10 ラスタ電卓。無データ値

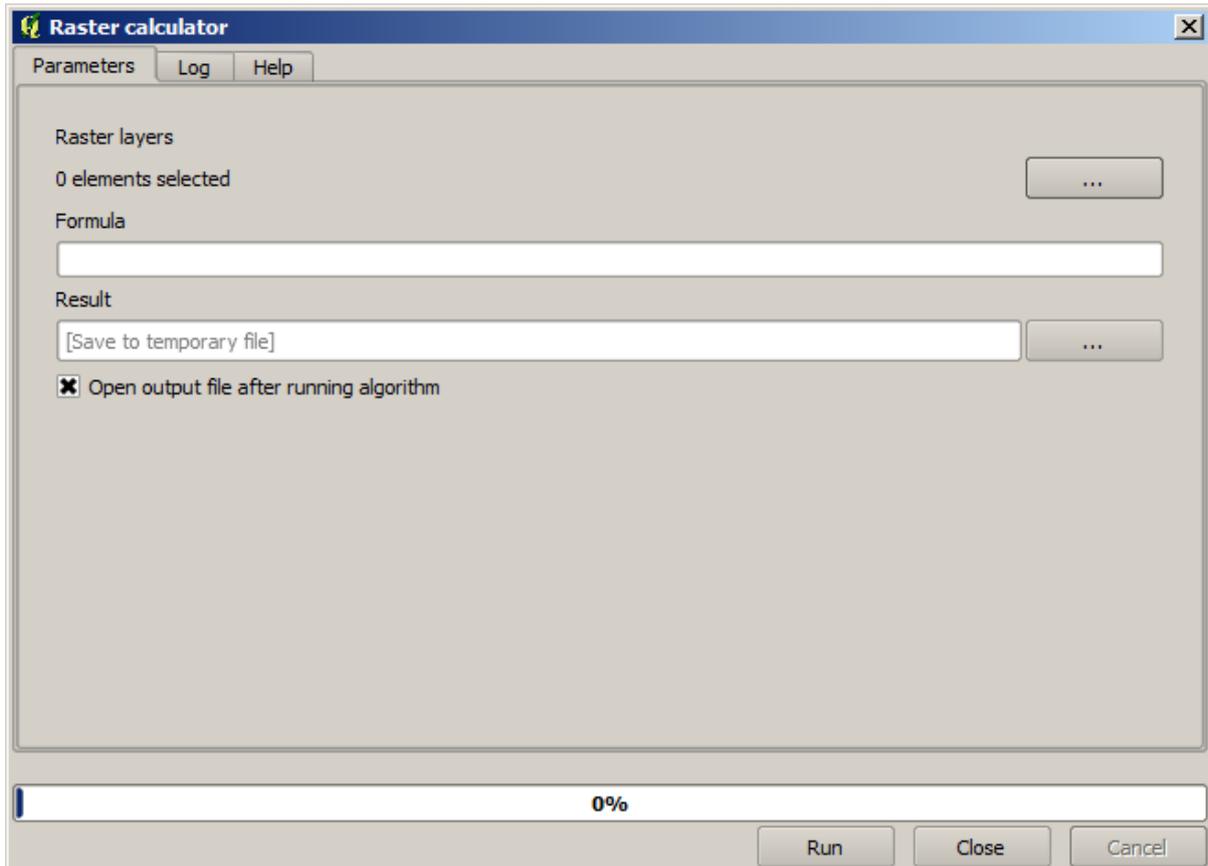
ノート: このレッスンでは、ラスタレイヤの上いくつかの操作を実行するために、ラスタ計算機を使用する方法について説明します。我々はまた、全く何を説明しません - データ値とどのように計算し、他のアルゴリズムは、それらに対処します

ラスタ電卓は、あなたが見つかる最も強力なアルゴリズムの一つです。それはすぐにあなたのツールボックスの重要な一部となります多くの異なる計算、および 1 のために使用することができ、非常に柔軟で汎用性の高いアルゴリズムです。

このレッスンでは、それらのほとんどはかなり単純、ラスタ電卓で、いくつかの計算を実行します。これは、私たちはそれを使用する方法を見て、それはそれを見つけるかもしれないいくつかの特別な状況を扱う方法をできるようにします。それを理解することは、一般的にそれに適用される特定の技術を理解するためにも、電卓を使用して、いつ後で期待される結果を取得することが重要です。

このレッスンに対応する QGIS プロジェクトを開き、あなたはそれがいくつかのラスタレイヤが含まれていることがわかります。

今すぐツールボックスを開き、ラスタ計算に対応するダイアログを開きます。



ノート: インターフェイスは、最近のバージョンで異なります。

ダイアログには、2つのパラメータが含まれています。

- 層は、分析に使用します。これは、複数の入力、あなたが望むようあなたに多くの層を選択することができる意味です。右のボタンをクリックして - 手側と、あなたが表示されますダイアログで使用したいレイヤーを選択します。
- 適用する式。式は、変数名として “... (...、B、C) アルファベット文字を使用して命名された上記パラメータで選択された層を使用するか、または $G1$ 、 $G2$ 、 $G3$ 。すなわち、式 $A + 2 * b$ が “ $G1 + 2 * g2$ ” と同じであり、第一層に第二層におけるプラス 2 倍の値を値の和を計算する、です。層の順序付けを使用すると、選択ダイアログに表示さと同じ順序です。

警告: 電卓は、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。必要とする計算式は、次のいずれかです。

$$h' = h * 3.28084$$

式フィールド内のレイヤフィールドとタイプ “* 3.28084” に DEM を選択します。

警告: 英語以外のユーザーの場合: 常に使用し、ありません 「『』」。

アルゴリズムを実行するために*実行*をクリックしてください。あなたは、入力層の同じ外観を有する層を得るが、異なる値となります。私たちが使用した入力層は、そのすべてのセル内の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。

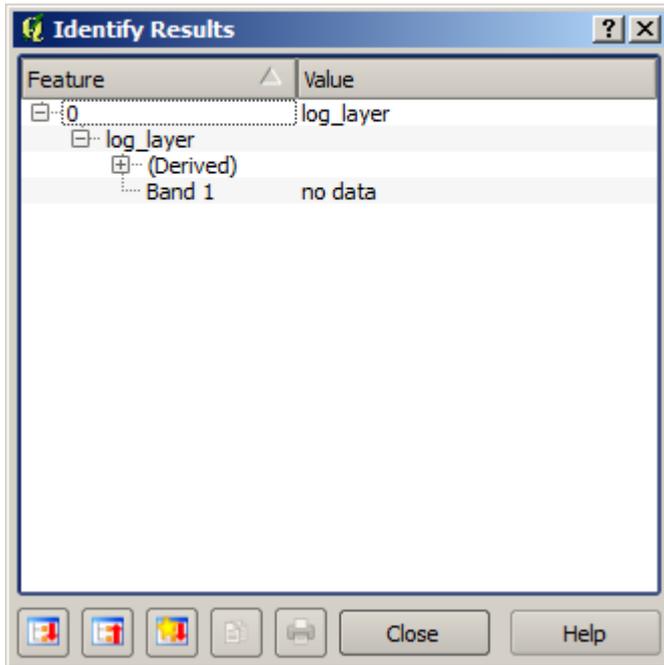
それでは、* accflow の*層の上に、この時間を別の計算を実行してみましょう。この層は、積算流量、水文学的パラメータの値が含まれています。その外部データ値 - それがないと、のみ所与流域の領域内でこれらの値を含んでいます。あなたが見ることができるように、レンダリングは、値が分散されている方法に起因する非常に有益ではありません。その流れの蓄積の対数を使用すると、はるかに有益な表現が得られます。私たちは、ラスタ計算機を使用していることを計算することができます。

のみ入力層として* accflow *レイヤーを選択し、再びアルゴリズムダイアログを開き、次の式を入力します。 “ログ (a) の “。

ここであなたが得る層です。



データ - あなたは、与えられた点での層の価値を知っている私たちが作成したレイヤーを選択し、流域の外でポイントをクリックする* *特定するツールを選択した場合、あなたはそれがありません含まれていることがわかります値。



次の演習のために我々は1つではなく、2つの層を使用しようとしている、と我々は唯一の第二の層で定義された流域内の有効な標高値とDEMを得ようとしています。電卓ダイアログを開き、入力層フィールドにプロジェクトの両方の層を選択します。対応するフィールドに次の式を入力します。

$a/a * b$

“a”は、積算流量の層を意味する（リストに表示される最初のものであるため）と“b”は、DEMを指します。外部データ値 - 私たちはここで、式の最初の部分にやっていると、流域内の1の値、およびなしで生じるであろう、それ自体で積算流量層を分割することです。データ値外部（ $DEM * NO_DATA = no_data$ ） - その後、我々は（ $DEM * 1 = DEM$ ）なし流域内のそれらの細胞の上昇値を取得するために、DEMを掛け結果のレイヤーはこれです。



この技術は、ラストレイヤ内*マスク*値に頻繁に使用される、あなたがラストレイヤによって使用される任意の矩形領域を他の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、ラストレイヤの標高ヒストグラムはあまり意味を持ちません。それは代わりに（上記彼の場合のように）流域に対応する値のみを使用して計算される場合、我々が得た結果は、実際には流域の構成についての情報を与える意味のあるものです。

データ値とそれらがどのように処理されます - 私たちは離れていないから、実行してきたこのアルゴリズムについて他の興味深いものがあります。あなたは私たちが掛けている層のエクステントを見ている場合 (あなたは二重のそれを行うことができます - 目次の層の自分の名前をクリックすると、そのプロパティを見て) あなたはそうでないことがわかりますフロー蓄積層によって覆われた範囲が完全 DEM の範囲ことが小さいので、同じ。

それは、これらの層が一致しないことを意味し、彼らは、1つのまたは両方の層をリサンプリングすることにより、それらのサイズとエクステントを均質化せずに直接掛けすることができません。しかし、我々は何もしていませんでした。QGISは、このような状況の世話をし、必要なときに自動的に入力層をリサンプリング。出力範囲は、入力層から計算された最小の被覆範囲、およびそれら cellsize 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、彼らは結果に影響を与える可能性があるため、あなたは常に、行われている追加の操作に注意する必要があります。この動作が希望されない場合があります場合には、手動のリサンプリングは、事前に適用されるべきです。複数のラスターレイヤを使用した場合、後の章では、我々は、アルゴリズムの動作の詳細が表示されます。

別のマスキングの練習でこのレッスンを終了してみましょう。当社は、1000年と1500メートルの間に標高とすべての分野での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ーはありませんが、電卓を使用して作成できます。

唯一の入力層と次式のように DEM を使用して計算を実行します

```
ifelse(abs(a-1250) < 250, 1, 0/0)
```

見ることができるように、我々は簡単な代数演算を行うためだけでなく、計算機を使用することができ、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、我々が仕事をしたい範囲内の1の値、およびそれ以外の細胞で無データを持っています。



無データ値が 0/0 の式から来ています。それは未定の値であるため、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。セルのデータ値がある - この小さなトリックを使えば、ノー何を知らなくても、無データ値を設定することができます。

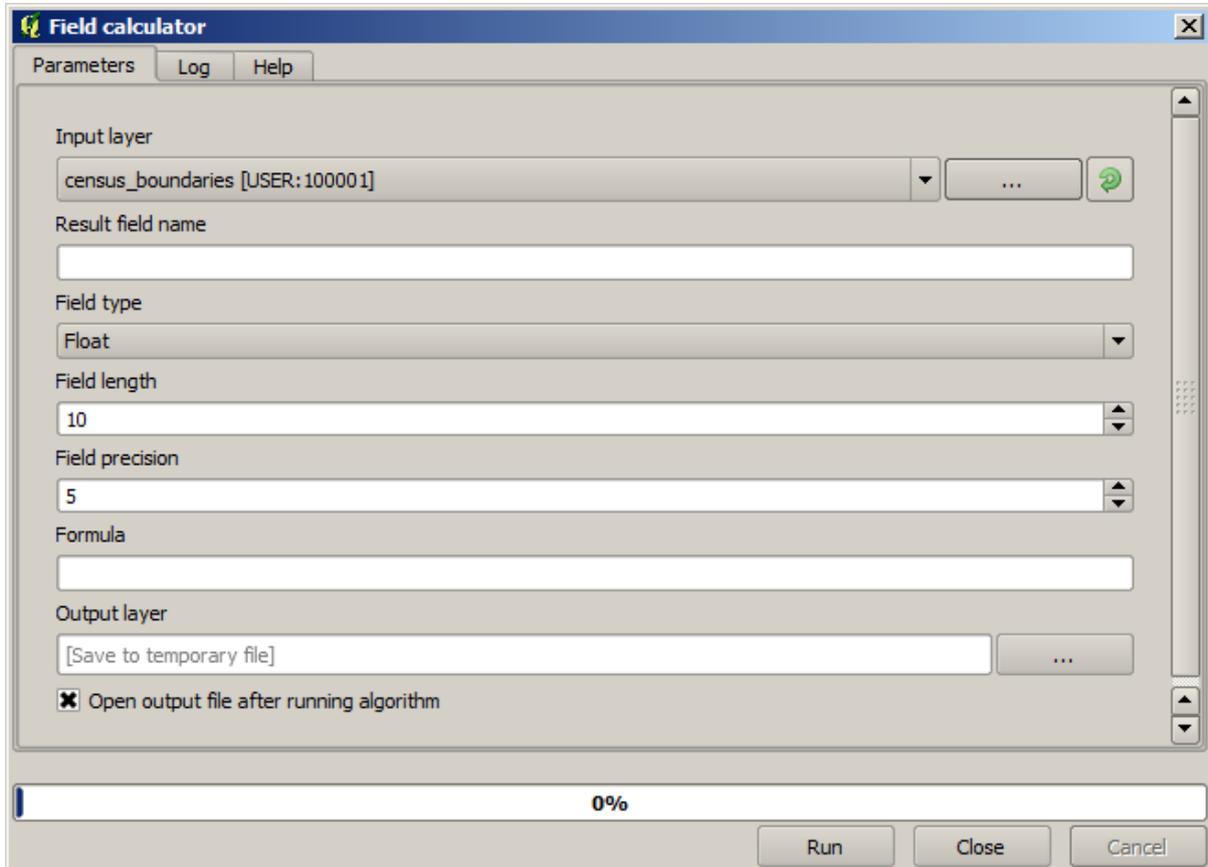
今、あなただけのプロジェクトに含まスロープ層を掛けする必要があります、あなたは望ましい結果を得るでしょう。

すべてのことは、電卓で、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 ベクトル計算機

ノート: このレッスンでは、ベクトル演算を使用して、数式に基づいてベクトル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。

数学的な表現を使用して、新しいラスターレイヤーを作成するには、ラスター計算機を使用する方法はすでに知っています。同様のアルゴリズムがベクトルレイヤーに対して使用可能であり、入力レイヤーと同じ属性を持つ新しいレイヤー、に加えて入力された式の結果を持つ新しいレイヤーを生成します。このアルゴリズムは、*フィールド電卓*と呼ばれ、以下の [パラメータ] ダイアログボックスを持っています。



ノート: インターフェースが大幅に変更された処理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。

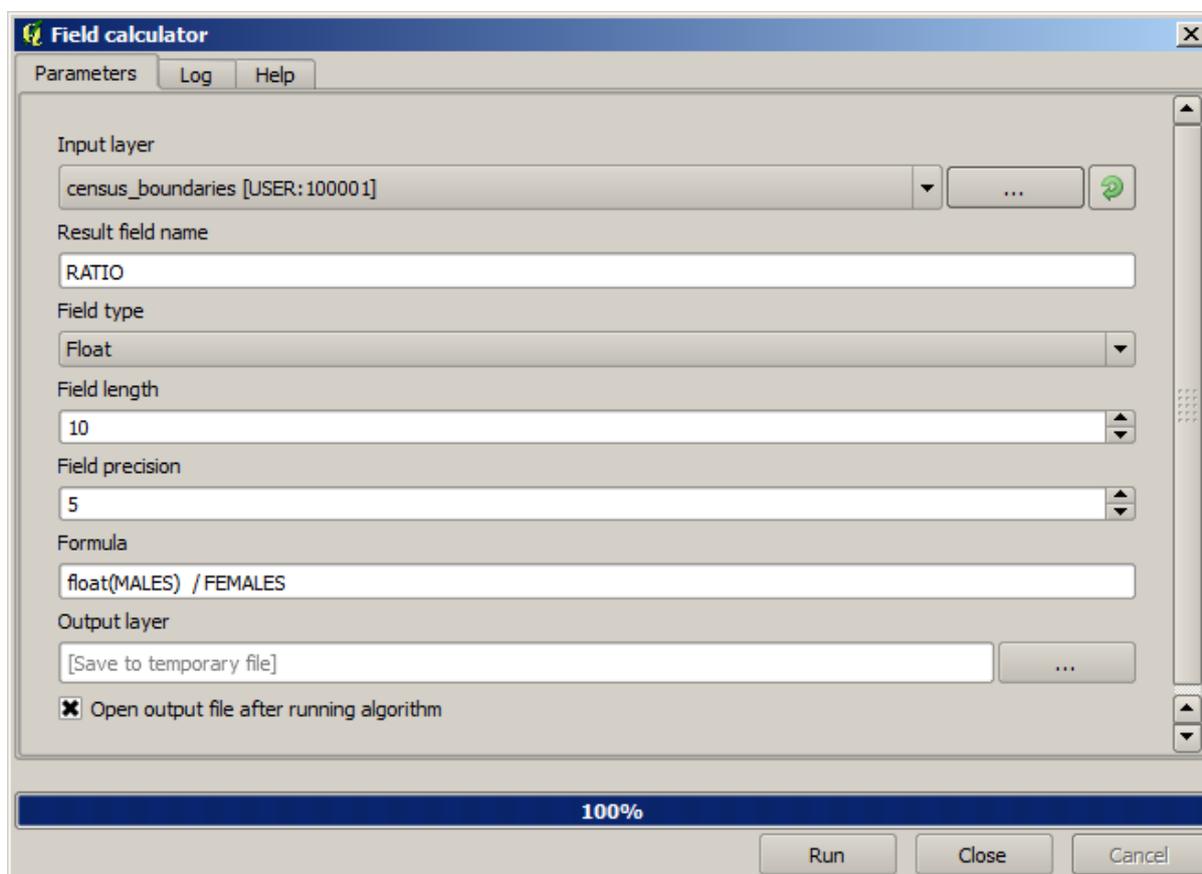
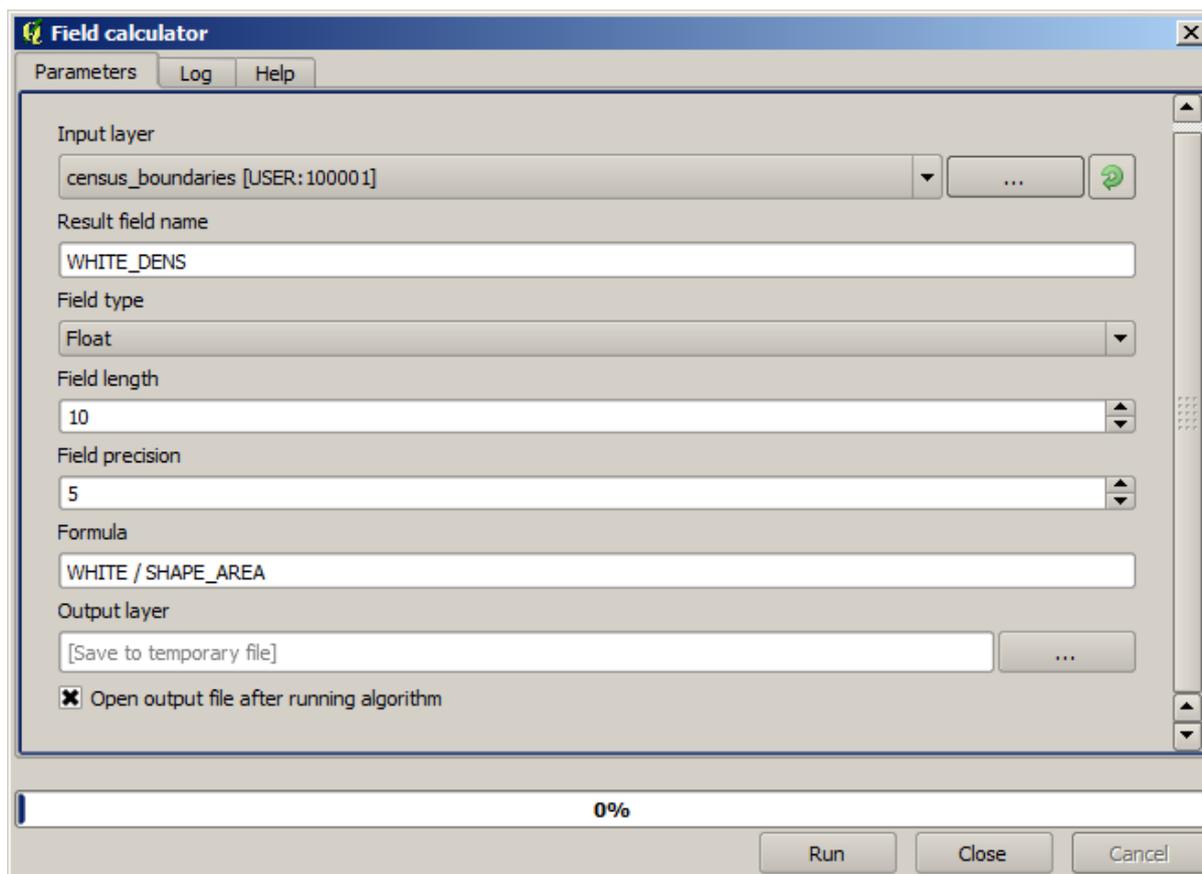
これは WHITE_DENS “ ” という名前の新しいフィールドを生成します。

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"
```

この時間は、パラメータウィンドウは *OK* の *ボタン* を押す前に、次のようになります。



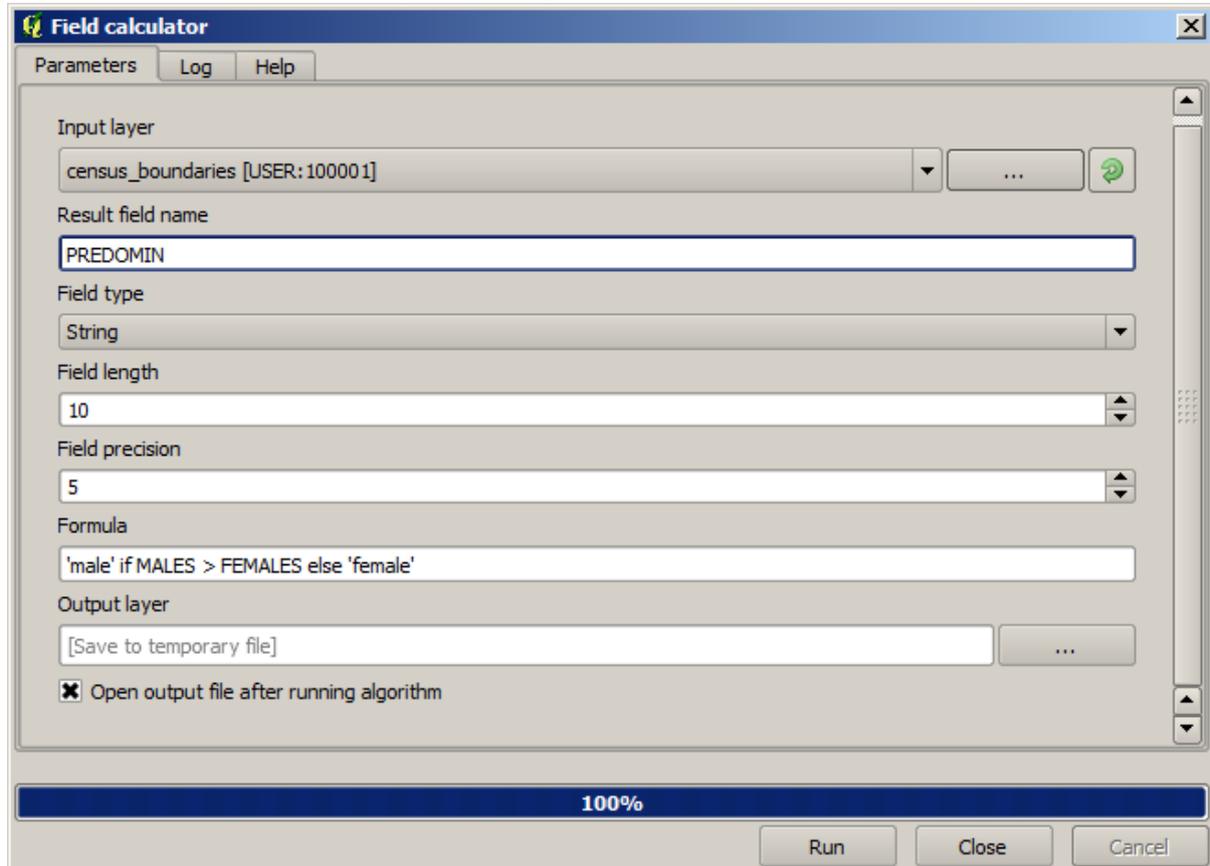
両方のフィールドが整数型であるため、以前のバージョンでは、結果が整数に切り捨てられることになり

まず、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 * 「男性」 / 「FEMALES」 “、。

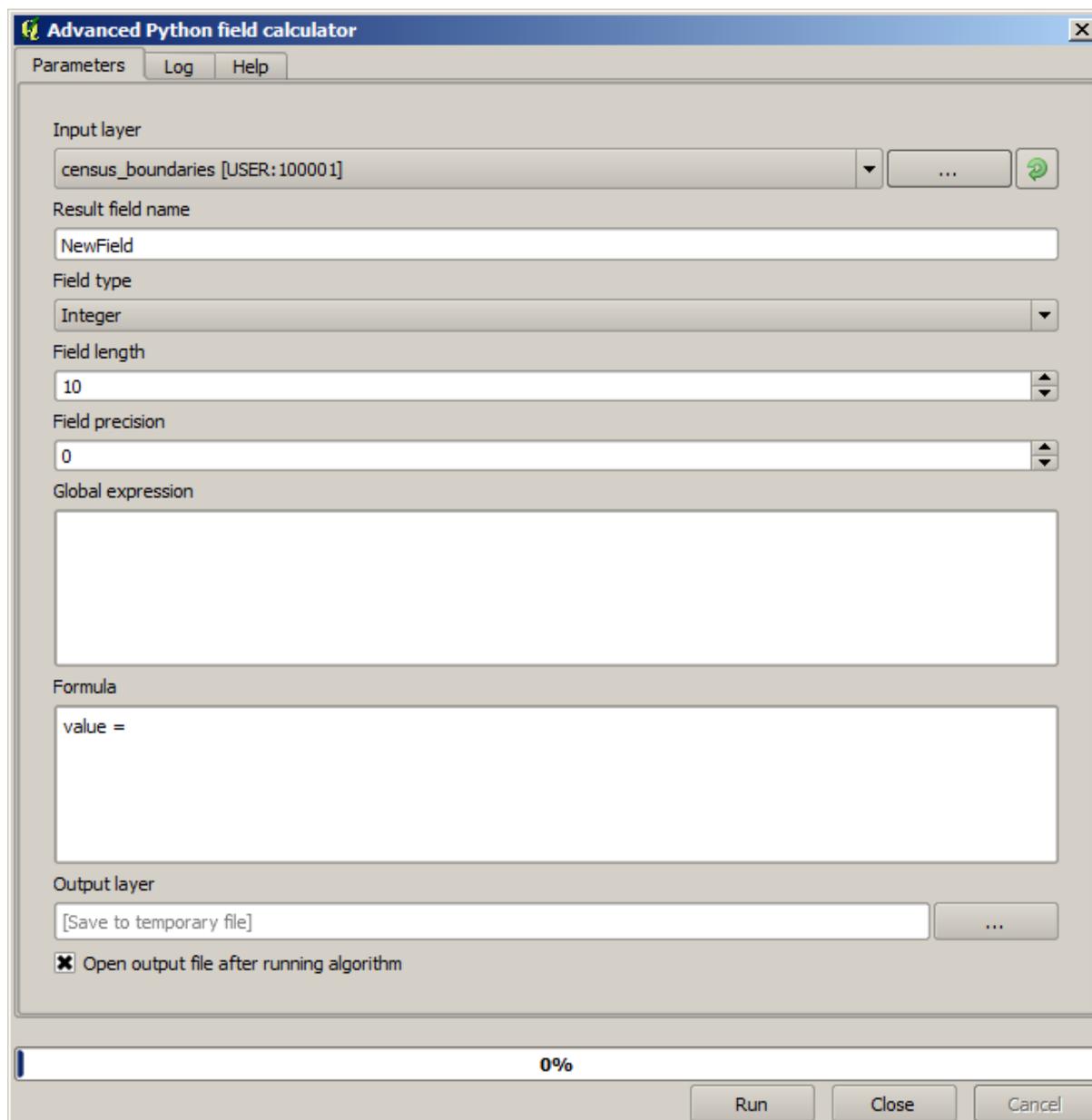
次の式を使用して、“male”または“female”テキスト文字列の代わりに、それらの比の値で新しいフィールドを持つように、条件付きの関数を使用できます:

```
CASE WHEN "MALES" > "FEMALES" THEN 'male' ELSE 'female' END
```

パラメータウィンドウは、次のようになります。



Python のフィールドの計算は、ここでは詳述しない*高度な Python のフィールドの電卓*、で提供されています



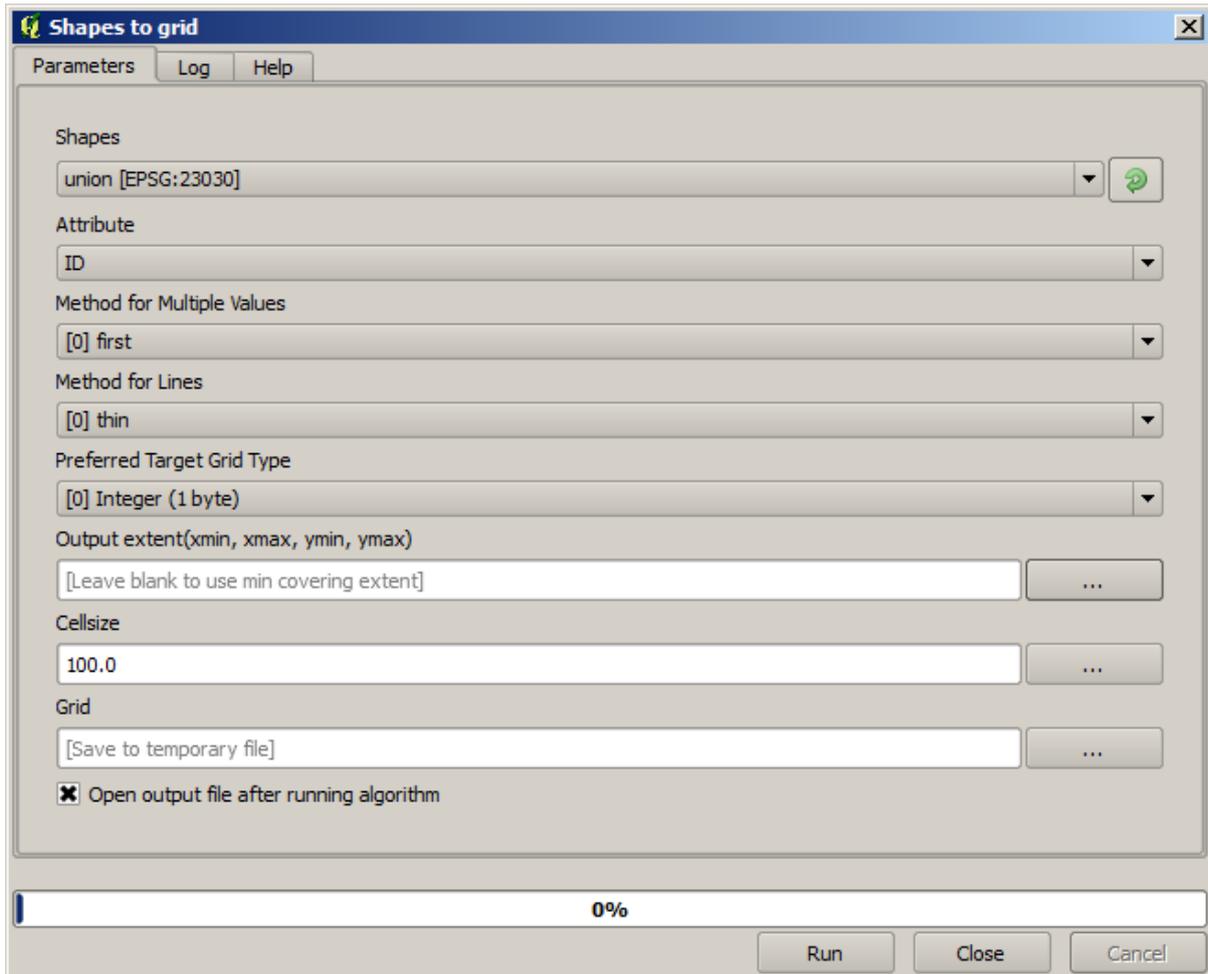
17.12 範囲の定義

ノート: 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、特にラスタのもの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、それらが行う分析によってカバーされる領域を定義するために、通常、得られたレイヤーの範囲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、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るが、同様にそれを行うための他のより実用的かつ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。

まずは、定義する範囲を必要とするアルゴリズムを開いてみましょう。ベクターレイヤーからラスタレイヤーを作成するラストライズアルゴリズムを開きます。

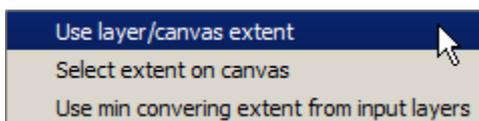


すべてのパラメータは、最後の二つのものを除いて、ラスター化されるべきレイヤーを定義し、ラスター化処理がどのように動作するかを設定するために使用されます。最後の二つのパラメータが、一方、出力レイヤーの特性を定義します。すなわち、それらは（必ずしも入力ベクトルレイヤーによって覆われ、同じ領域ではない）覆われている領域を定義することを意味し、（ベクトルレイヤーはセルサイズを有していないので、ベクターレイヤーから inferred できない）解像度/セルサイズ。

まず最初に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、カンマで区切りで、入力することです。

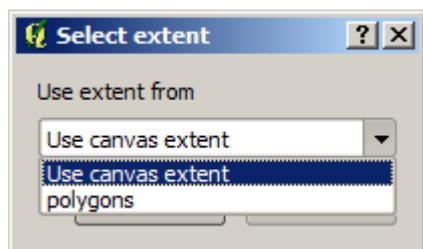


それは余分な説明を必要としません。これは最も柔軟なオプションですが、それは baainiyotteha あまり実用的でないこともあり、それが他のオプションが実装されている理由です。範囲テキストボックスの左側を - それらにアクセスするには、右側のボタンをクリックする必要があります。



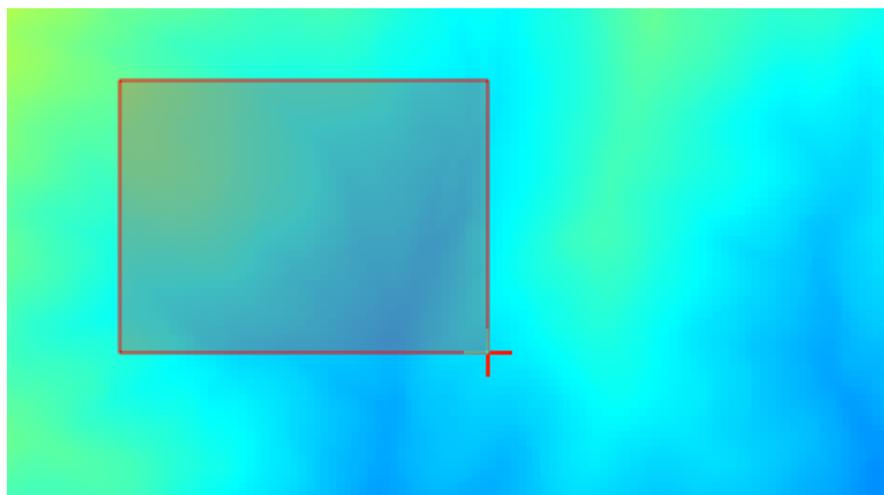
それらのそれぞれが何をするか見てみましょう。

第 1 のオプションは レイヤー/キャンバス範囲を使用 で、以下に示す選択ダイアログが表示されます。



ここでは、キャンバス（現在のズームでカバー範囲）の範囲、または拡張可能なレイヤーのいずれかを選択できます。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。

第2のオプションは、キャンバス上で範囲選択です。この場合、アルゴリズムダイアログが消え、所望の範囲を定義するために QGIS キャンバス上でクリック&ドラッグできます。

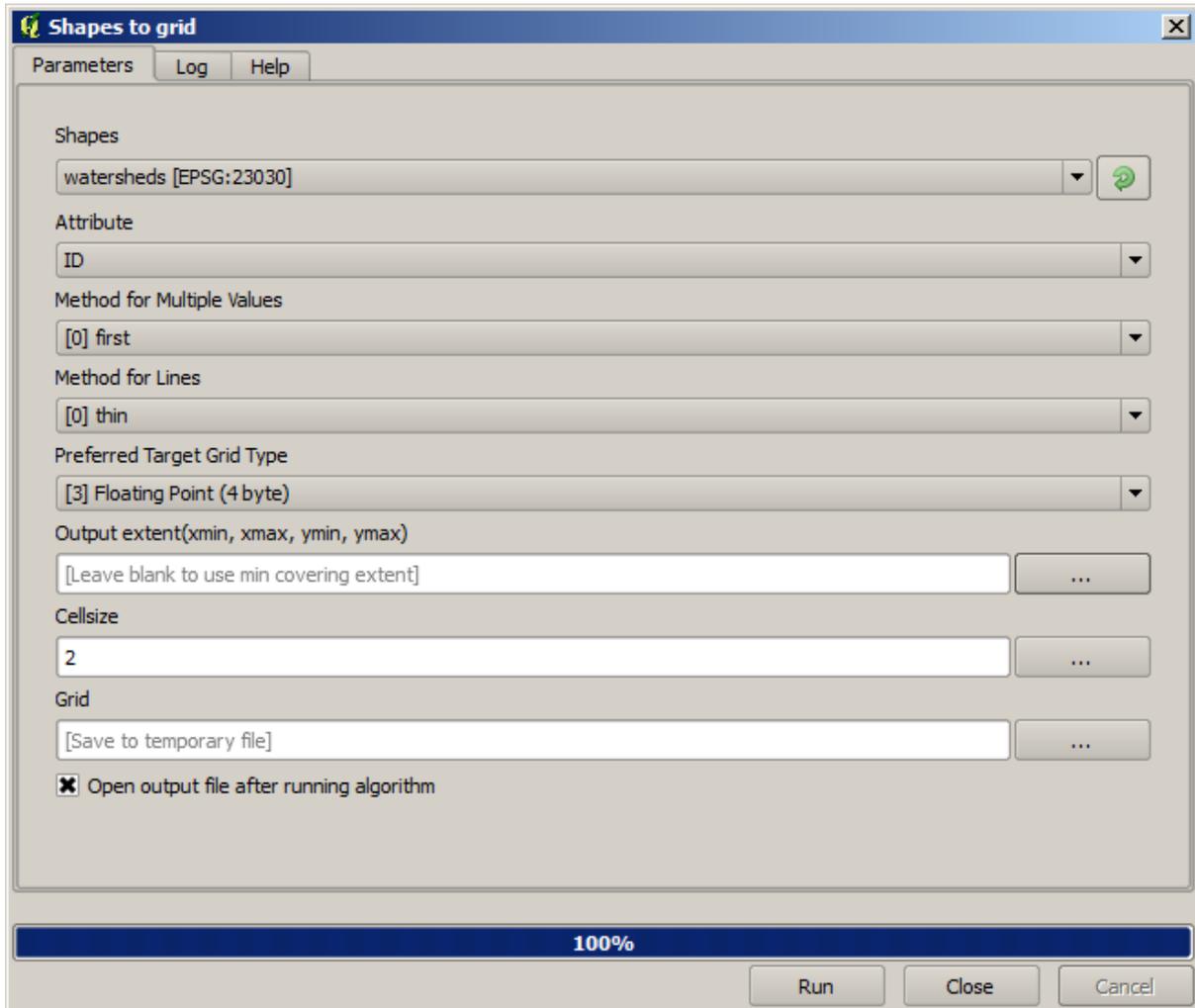


マウスボタンを放した後、ダイアログが再度表示されますし、テキストボックスには、すでに定義された範囲に対応する値を持つことになります。

最後のオプションは、入力レイヤー範囲をカバーする最小の使用で、デフォルトのオプションです。これは、アルゴリズムを実行するために使用されるすべてのレイヤーの範囲をカバーする最小を計算し、テキストボックスに任意の値を入力する必要はありません。単一の入力レイヤーの場合には、実行されているアルゴリズムのように、同じ程度には、すでに見たことレイヤー/キャンバス範囲を使用 で同じ入力レイヤーを選択することによって得ることができます。複数の入力レイヤーが存在する場合、それは一緒にそれらの全てから計算されるので、範囲をカバーする最小は、入力レイヤーの範囲のいずれにも該当しません。

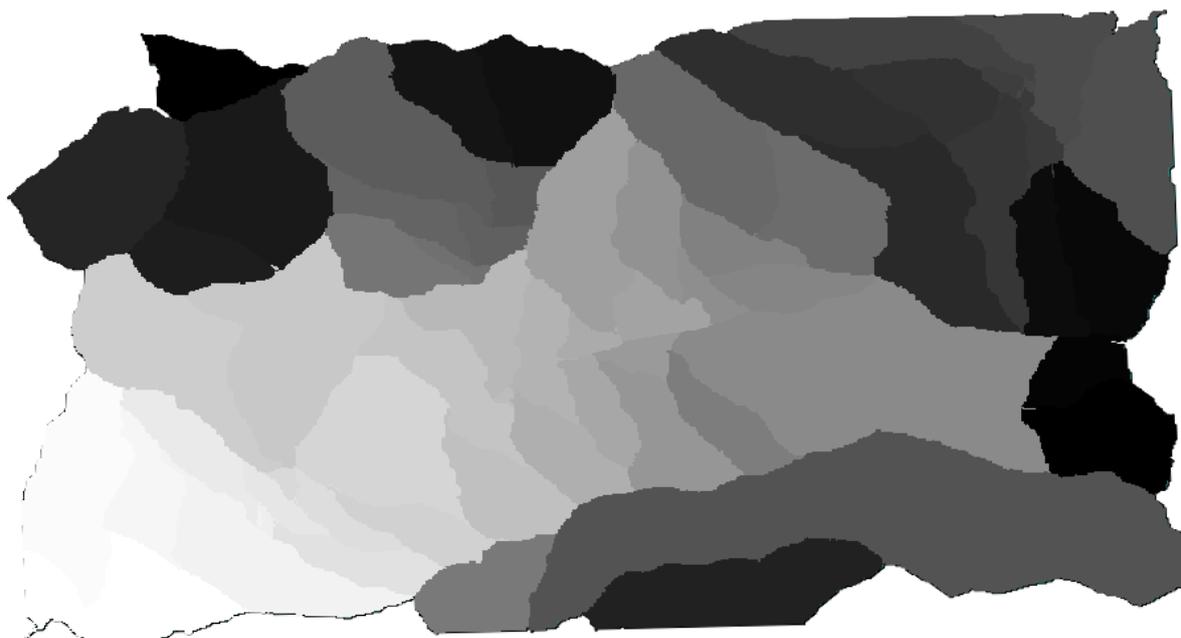
ラスタライズアルゴリズムを実行するために、この最後のメソッドを使用します。

次に示すようにパラメータダイアログボックスを入力し、*OK* を押してください。



ノート: この場合には、*NAME* は最大値=64 の整数なので、浮動小数点(4バイト)よりも 整数(1バイト)を使用するほうが良いです。これによってファイルサイズは小さくなり、計算がより高速になります。

元のベクトルレイヤーで覆われた領域を正確にカバーするラスターレイヤーが得られるでしょう。



場合によっては、最後のオプション 入力レイヤーから最小カバー範囲を使用 が使用できないことがあります。これは、入力レイヤーを持たず他のタイプのパラメーターだけ持つアルゴリズムで起こります。その場合は、手動で値を入力するか、他のオプションのいずれかを使用する必要があります。

選択が存在する場合、レイヤーの範囲は地物の全体集合のことであり、たとえラスタライズ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ーを作成しそれを入力として使用したいかもしれません。

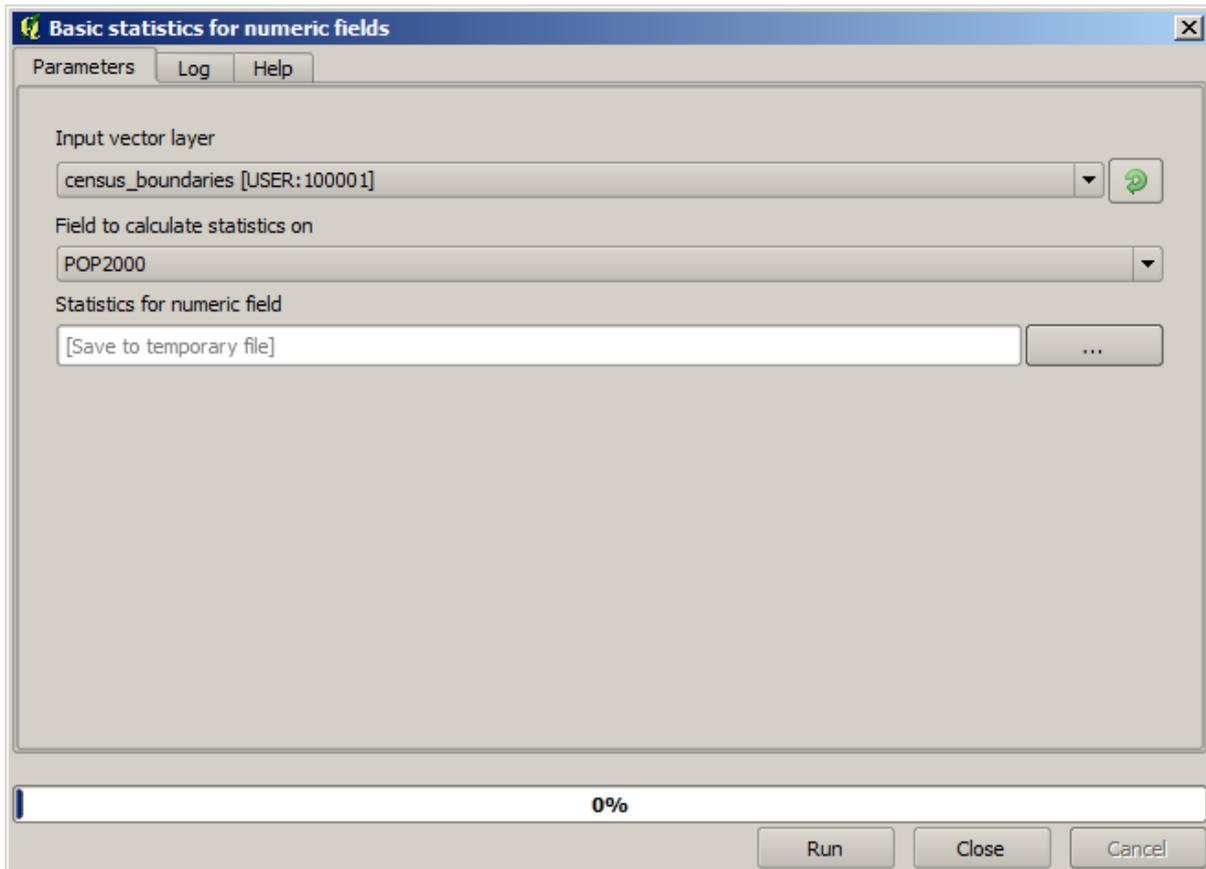
17.13 HTML 出力

ノート: 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。

我々はこれまでに生産されているすべての出力は、(ラスタまたはベクトル)層でした。しかし、いくつかのアルゴリズムは、テキストとグラフィックスの形式で出力を生成します。このすべての出力は HTML ファイルで包み、そうで表示されます - 処理フレームワークの別の要素であると呼ばれる*結果ビュー*。

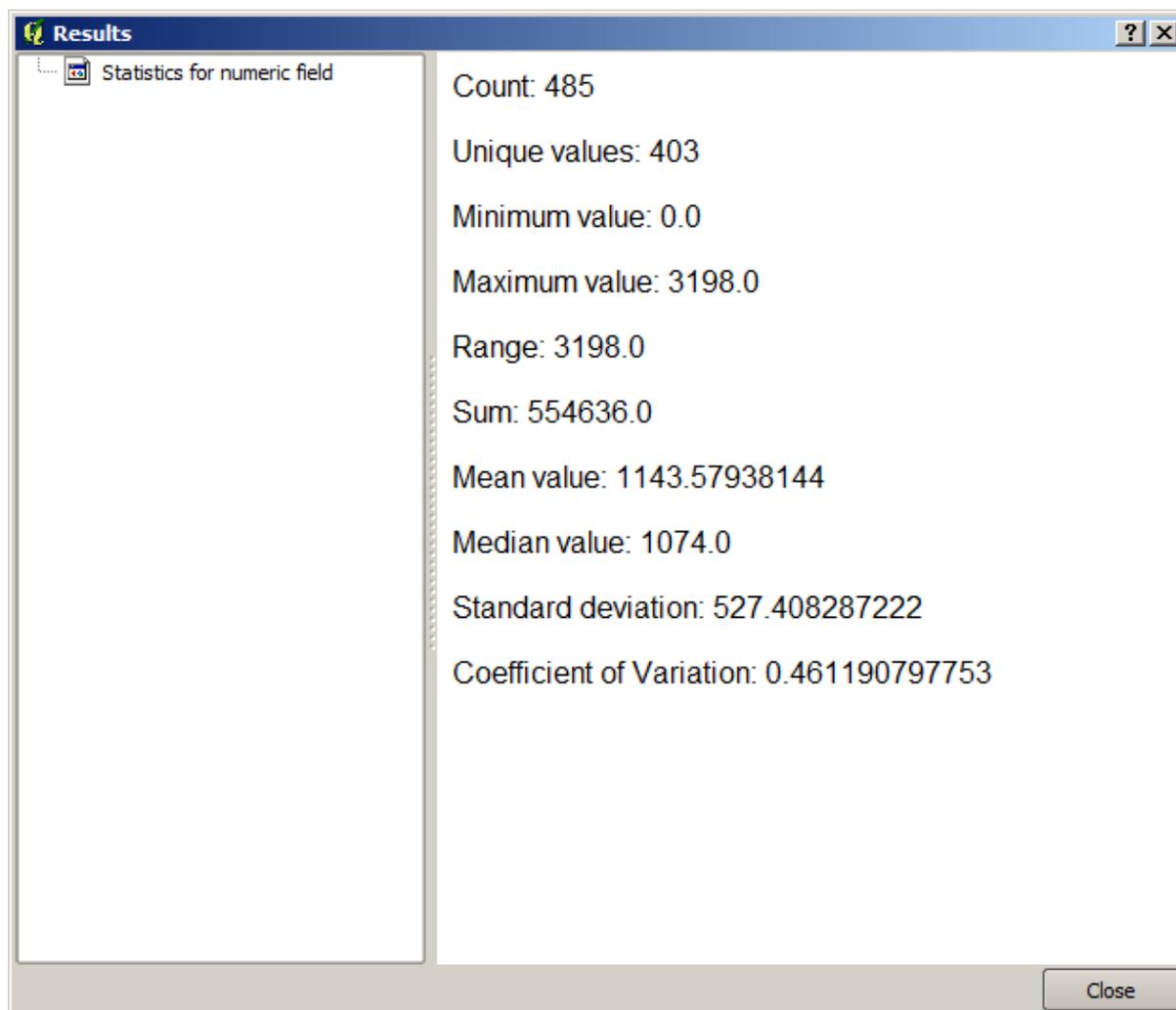
彼らがどのように機能するかを理解するために、これらのアルゴリズムのいずれかを見てみましょう。

このレッスンで使用するデータでプロジェクトを開き、数値フィールドの*基本的な統計情報を開く*アルゴリズム。



アルゴリズムはかなり単純で、あなただけ使用する層と、そのフィールド（数値 1）のいずれかを選択する必要があります。出力は、タイプの HTML のですが、対応するボックスは正確にあなたがラスタまたはベクトル出力の場合に見つけることができる 1 のように動作します。あなたは、ファイルパスを入力するか、一時ファイルに保存する場合は空白のままにすることができます。この場合、しかし、唯一の “html” と “htm” 拡張が許可されているので、別のものを使用して出力形式を変更する方法がありません。

入力として、プロジェクト内の唯一の層、および * POP2000 * フィールドを選択するアルゴリズムを実行し、アルゴリズムが実行されるとパラメータ] ダイアログが閉じられた後、次の図のような新しいダイアログが表示されます。



これは、**結果ビューアです。これは、現在のセッション、簡単にアクセス中に生成されたすべての HTML 結果を保持しますので、あなたがそれを必要とする時はいつでも、あなたはすぐにそれらを確認することができます。それは層で発生したとして、あなたは、一時ファイルに出力を保存した場合は、QGIS を閉じると、それは削除されます。あなたは非一時的なパスに保存されている場合は、ファイルが残りますが、それは*結果ビューア*次回オープン QGIS には表示されません。

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、アルゴリズムは、外部プロセスからのテキスト出力を取り込む場合にはそうです。他の場合には、出力は、テキストとして提示されているが、内部的に通常の数値の形で、いくつかの小さな出力に分割されます。私たちは、実行しているアルゴリズムは、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い、変数に格納されています。これは今、まったく重要性を持っていないが、我々は、グラフィカルモデラーに移動したら、それは私たちが他のアルゴリズムの数値入力としてこれらの値を使用できるようになりますことがわかります。

17.14 最初の分析例

ノート: 処理フレームワークの要素により精通できるので、このレッスンでは、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。

これですべてが設定されていて、私たちは外部のアルゴリズムを使用できることを、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。世界のデータ - いくつかの本当の持つ大規模な運動を仕事する時間です。

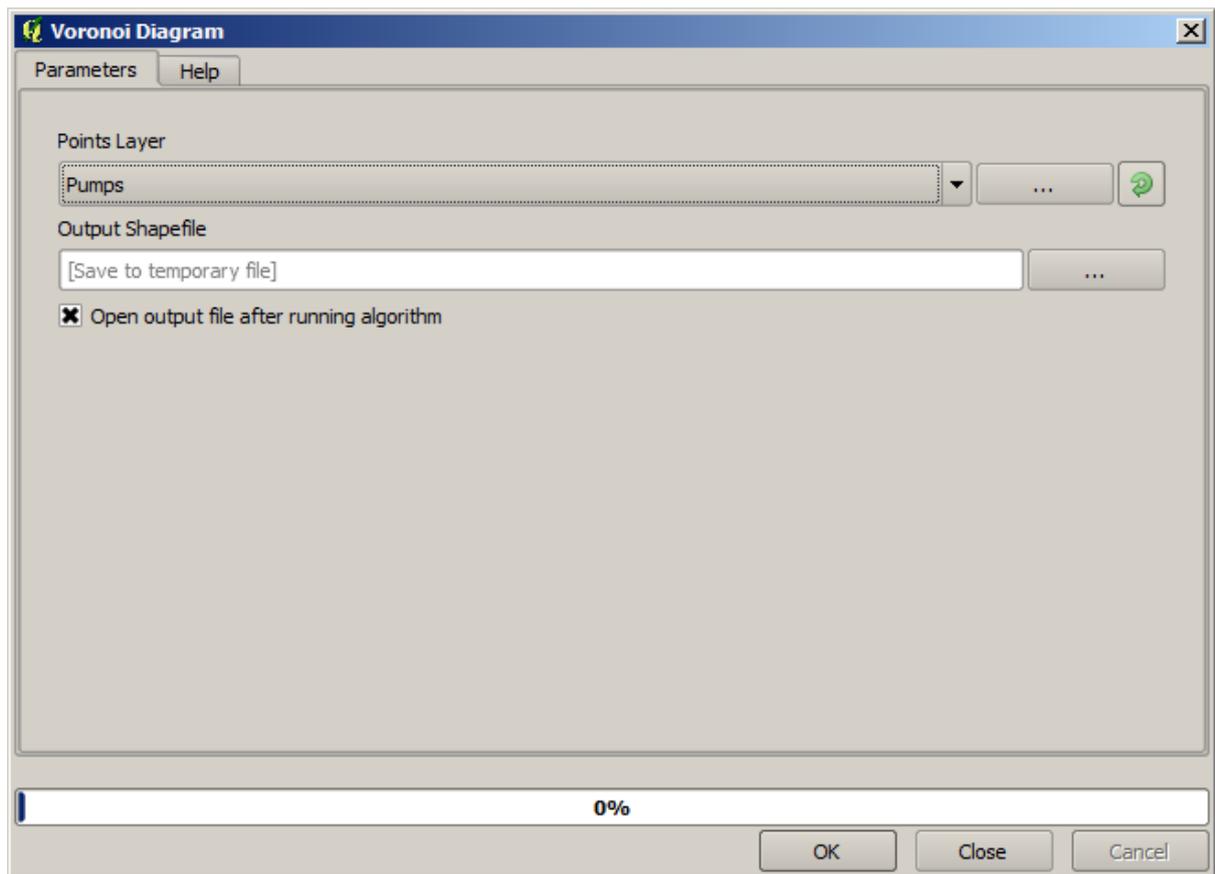
ここではジョン・スノーが 1854 年に使用した有名なデータセットを使用することになります、彼の画期的

な作品 (http://en.wikipedia.org/wiki/John_Snow_で、%の28Physician % 29) そしていくつかの興味深い結果が得られます。このデータセットの分析はかなり明白であり、良好な結果と結論で終わるための高度利用 GIS 技術は必要ありませんが、それはこれらの空間の問題を分析し、異なる処理ツールを使用することによって解決できるかを示すのに良い方法です。

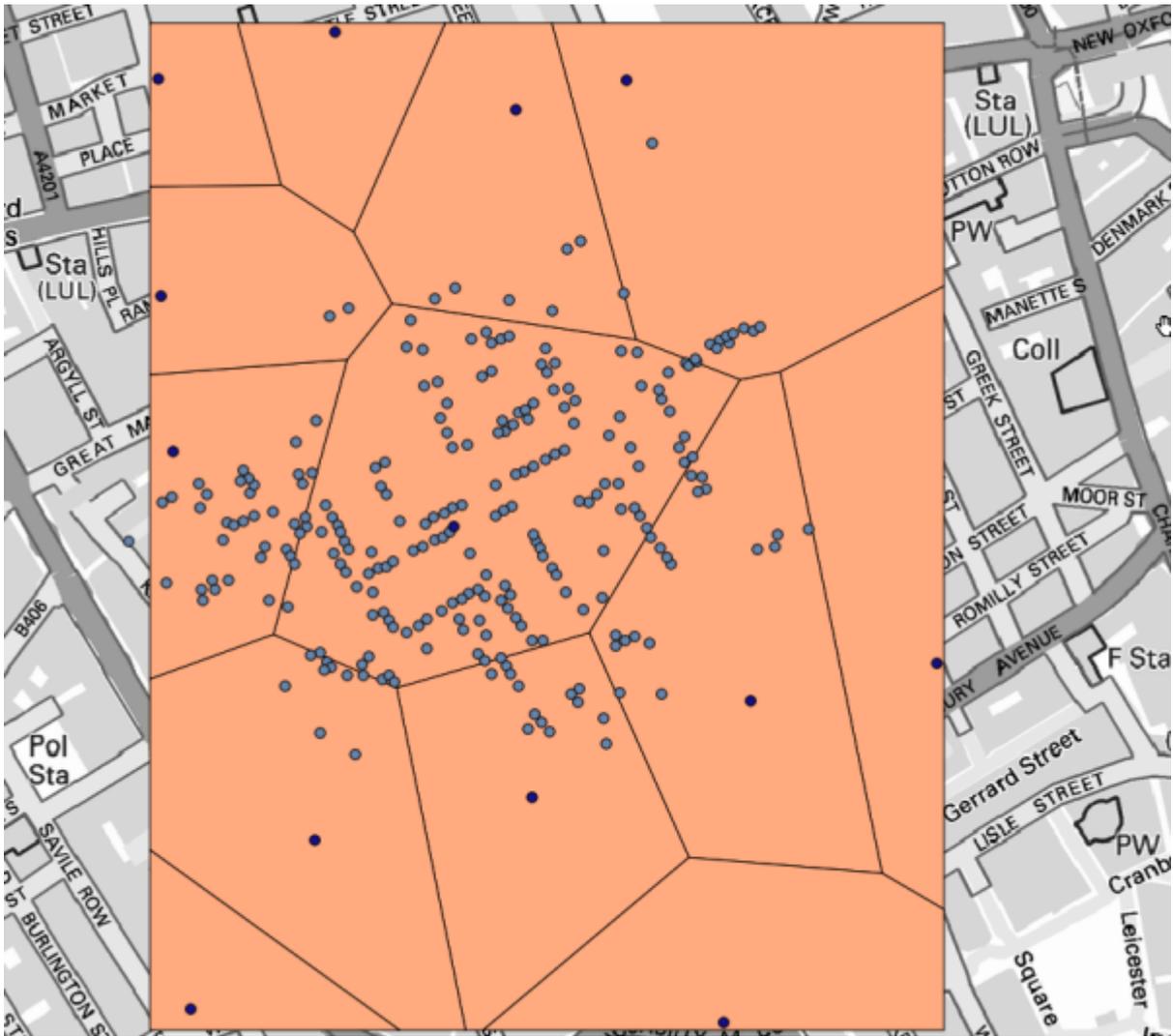
データセットは、コレラによる死亡とポンプの位置、および TIFF フォーマットで OSM レンダリングされたマップでシェープファイルが含まれています。このレッスンのための対応 QGIS プロジェクトを開きます。



まず最初に行うべきは、各ポンプの影響ゾーンを取得するためにポンプレイヤーのボロノイ図 (別名ティールセンポリゴン) を計算することです。そのために*ボロノイ図*アルゴリズムを使用できます。

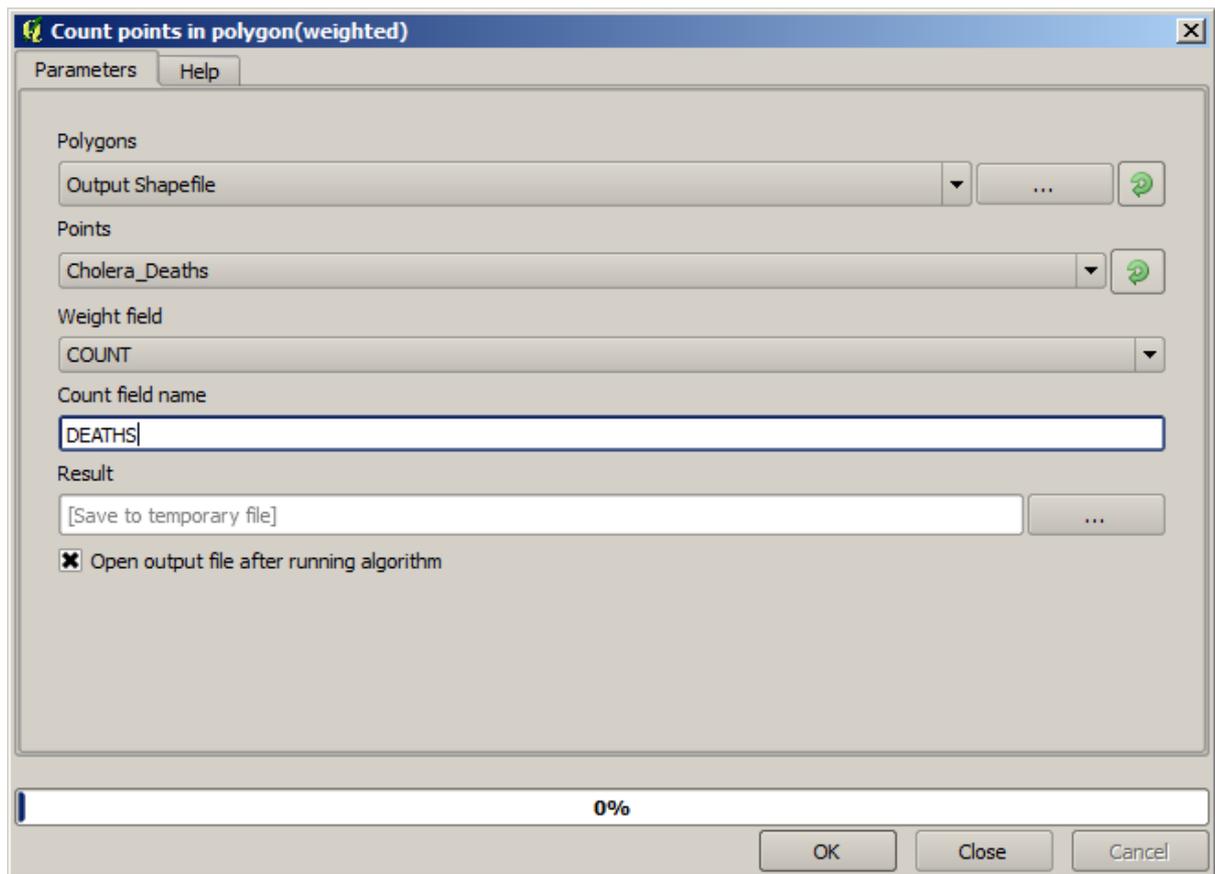


かなり簡単ですが、それですでに興味深い情報が得られます。

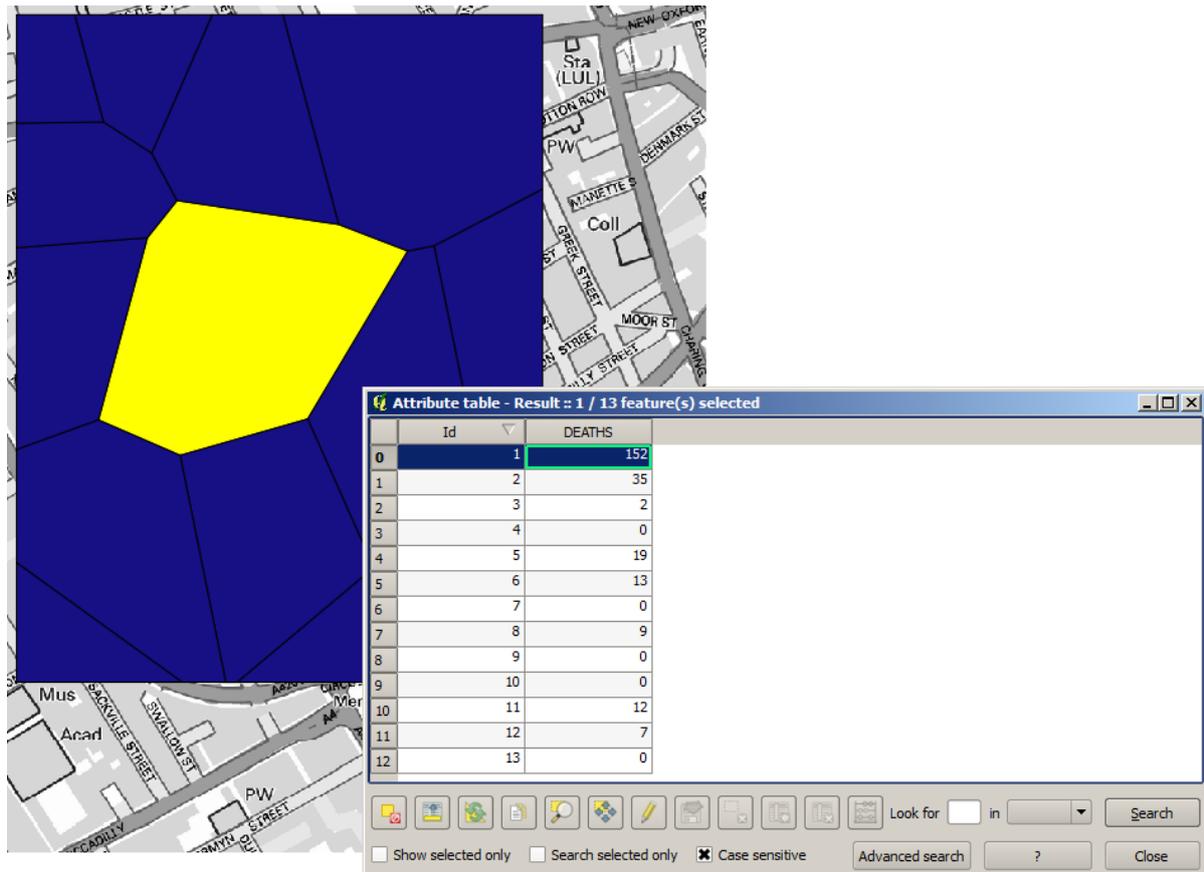


明らかに、ほとんどの場合は、多角形の 1 の範囲内にあります

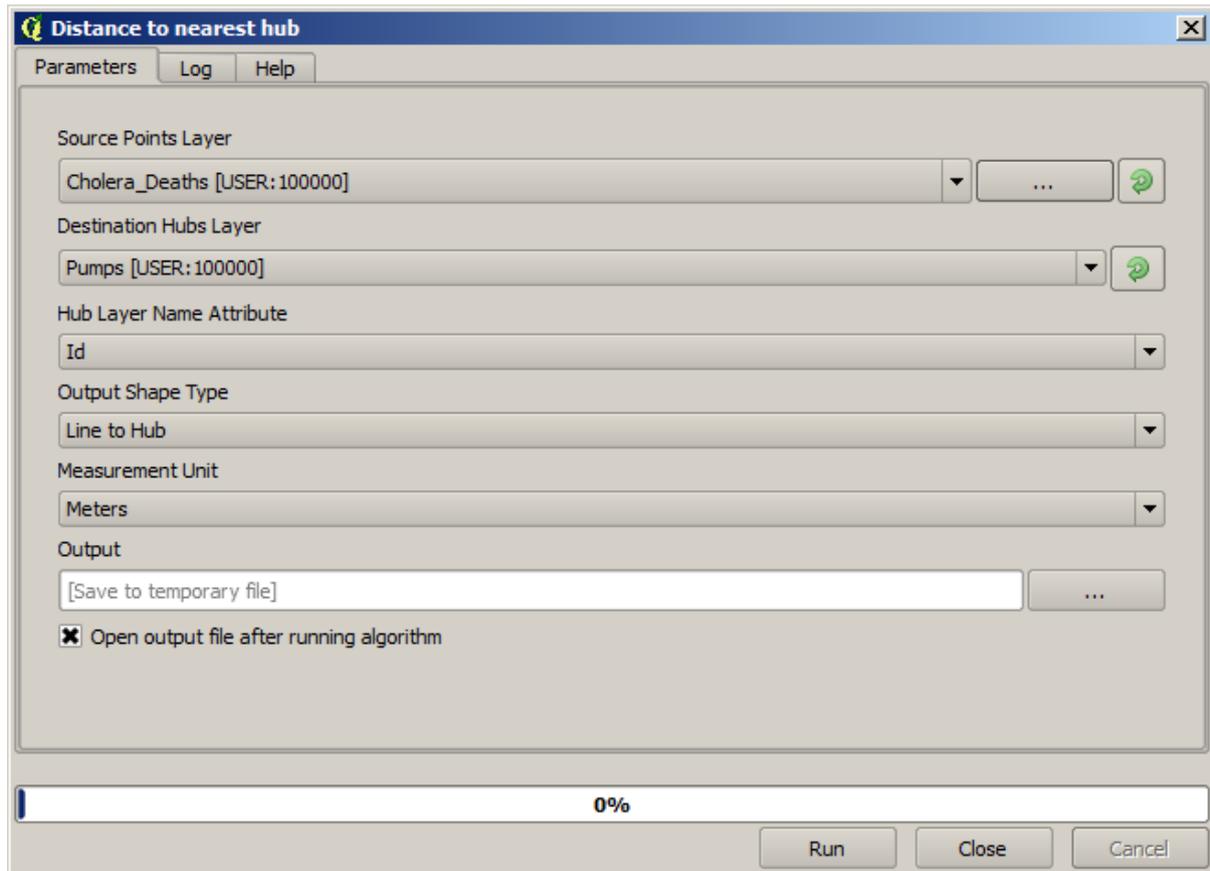
より定量的な結果を得るためには、各ポリゴンにおける死亡者数をカウントできます。各ポイントは、死亡が発生した建物を表しており、死亡者数は、属性に格納されているので、ポイントをカウントすることはできません。私たちは、重み付けされた回数を必要とするので、ポリゴンカウントポイント（加重）ツールを使用します。



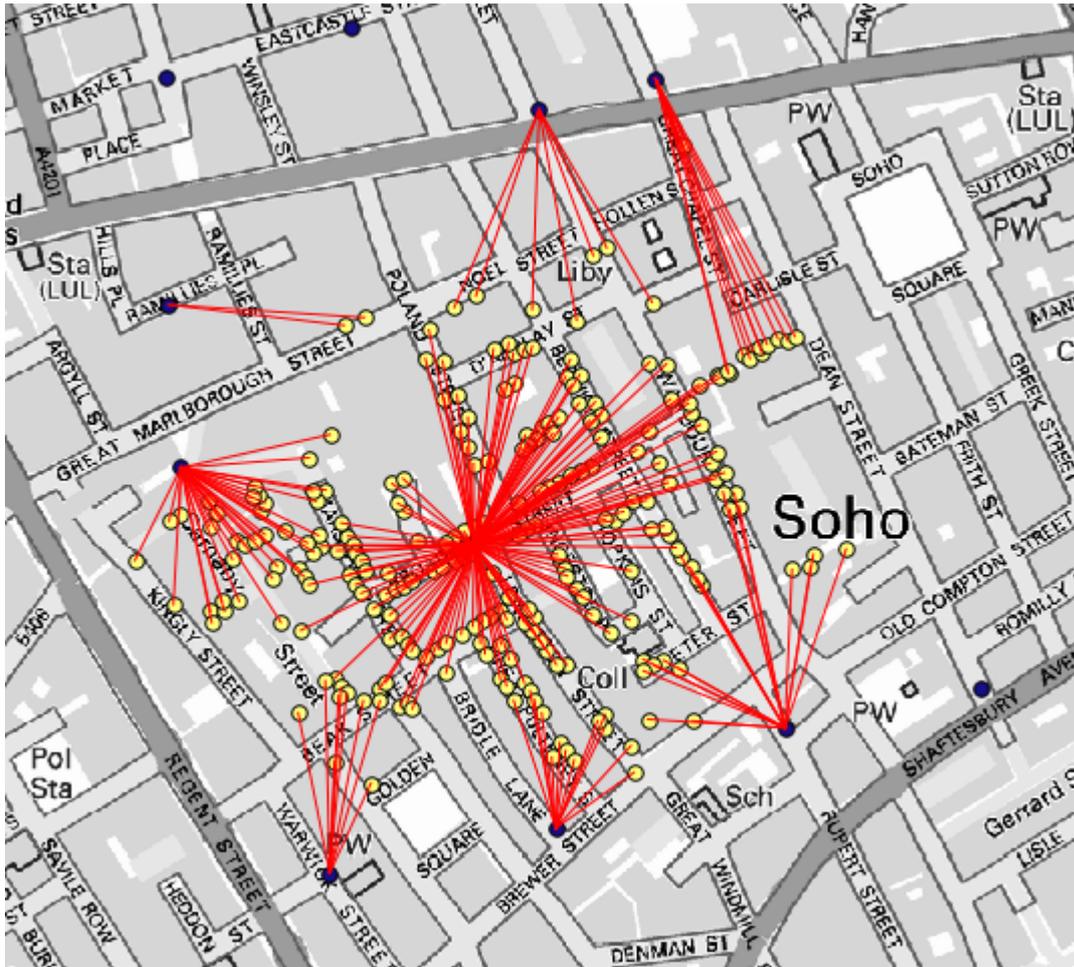
新しいフィールドは、* 死亡と呼ばれる、と我々は重みフィールドとして COUNT *フィールドを使用します。結果のテーブルは、明らかに第一のポンプに対応するポリゴンにおける死亡者数が他のものよりもはるかに大きいことを反映しています。



“Pumps”レイヤーにおけるポイントと “Cholera_deaths”レイヤーの各点の依存性を視覚化するもう一つの良い方法は、最も近いものに線を描画することです。これは、最寄のハブ*ツールへ*距離で行われ、次のような構成を使用することができます。

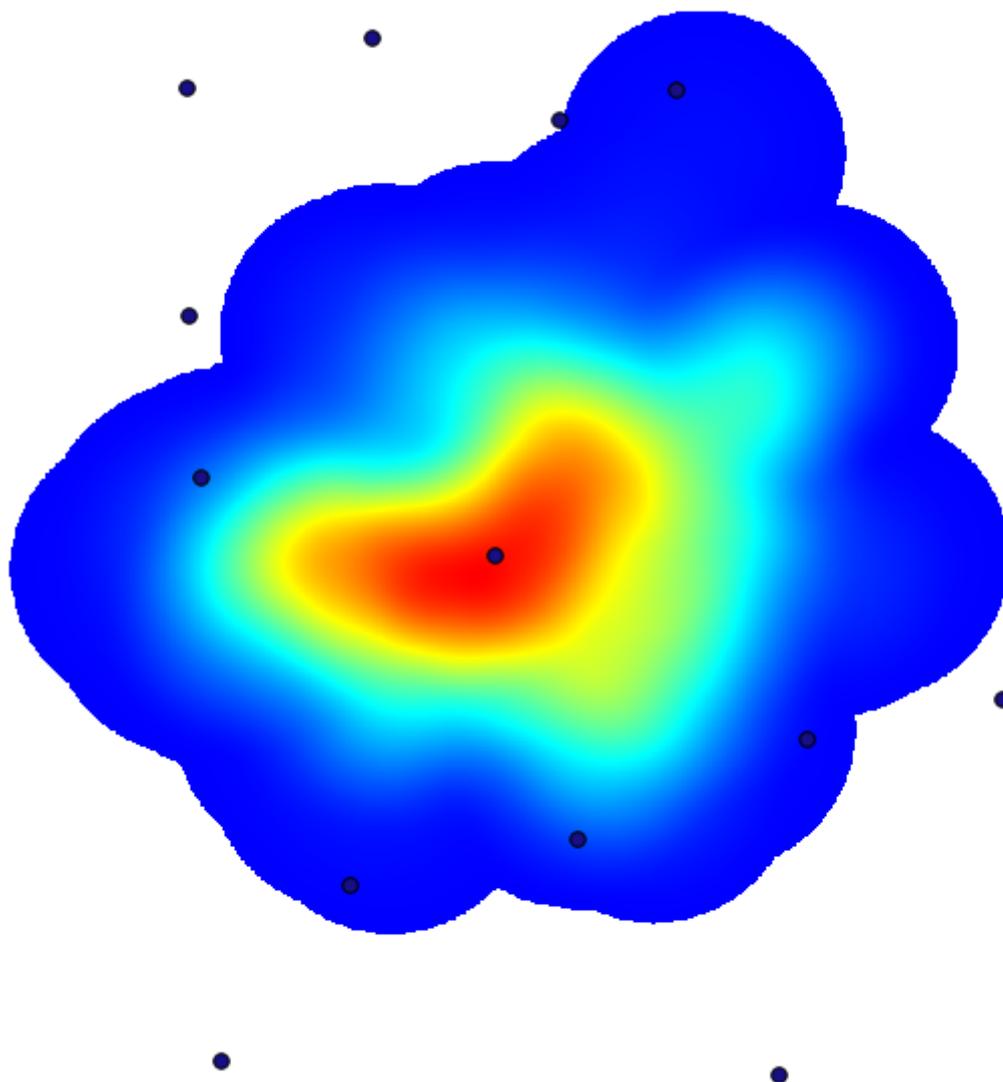


この結果は次の通りです:

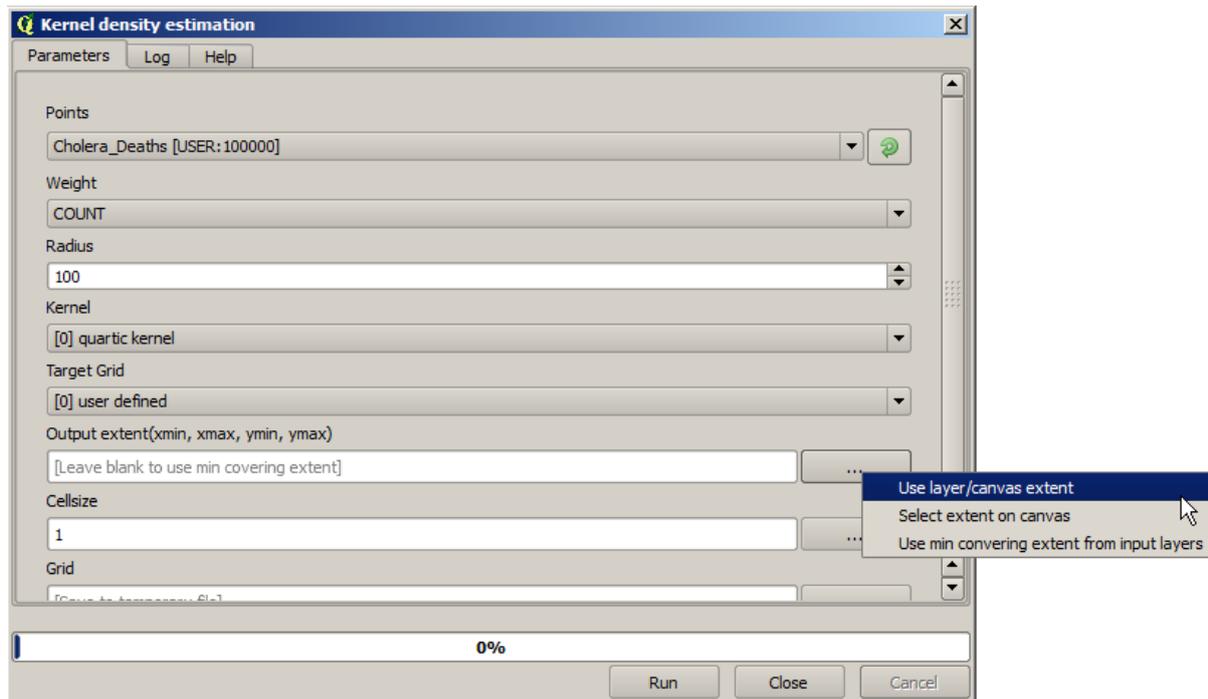


行数は、中央ポンプの場合の方が大きいですが、これは死亡数が、コレラの症例が発見された場所の数を表すものではないことを忘れないでください。それは、代表的なパラメータであるが、いくつかの場所が他よりも多くの例があるかもしれないことを考慮されていません。

密度レイヤーはまた、何が起きているかの非常にクリアな視界を提供します。それは*カーネル密度*アルゴリズムで作成できます。*Cholera_deaths*層、100の半径と重みフィールドとしての*数*フィールドを使用して、街のラスタレイヤの範囲とセルサイズは、このような何かを得ます。



出力範囲を取得するには、入力する必要はありません。右側のボタンをクリックし、* Use layer / canvas extent *を選択します。



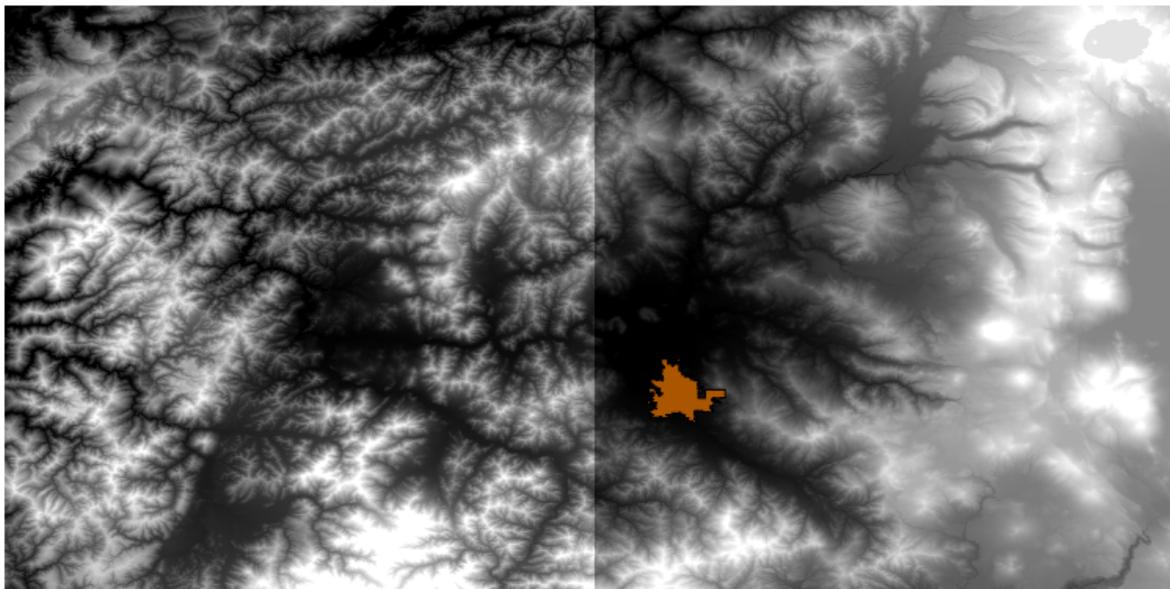
街のラスターレイヤーを選択し、その範囲は、自動的にテキストフィールドに追加されます。あなたにもそのレイヤーのセルサイズを選択し、セルサイズと同じことを行う必要があります。

ポンプレイヤーと組み合わせることで、死亡例最高濃度が検出されたホットスポットで明確に一つのポンプがあることを参照してください。

17.15 ラスターレイヤーをクリッピングし、マージ

ノート: 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つだけのベクトルレイヤー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ーを計算しようとしています。ベース DEM は2つのラスターレイヤーに分割され、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。



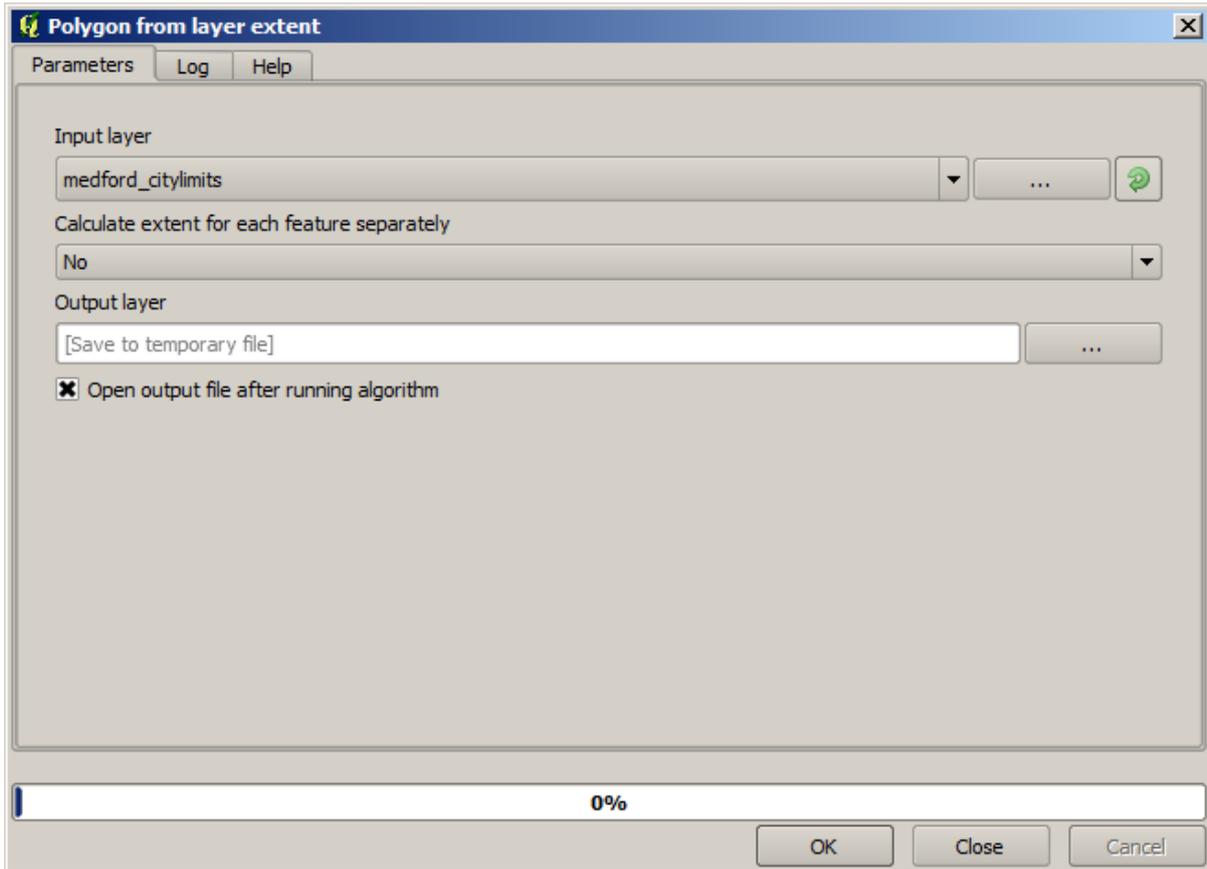
これらのレイヤーには二つの問題があります。

- 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）
- 彼らは、2つの異なるファイルにあります（市域は1つだけのラスターレイヤーに入るが、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）。

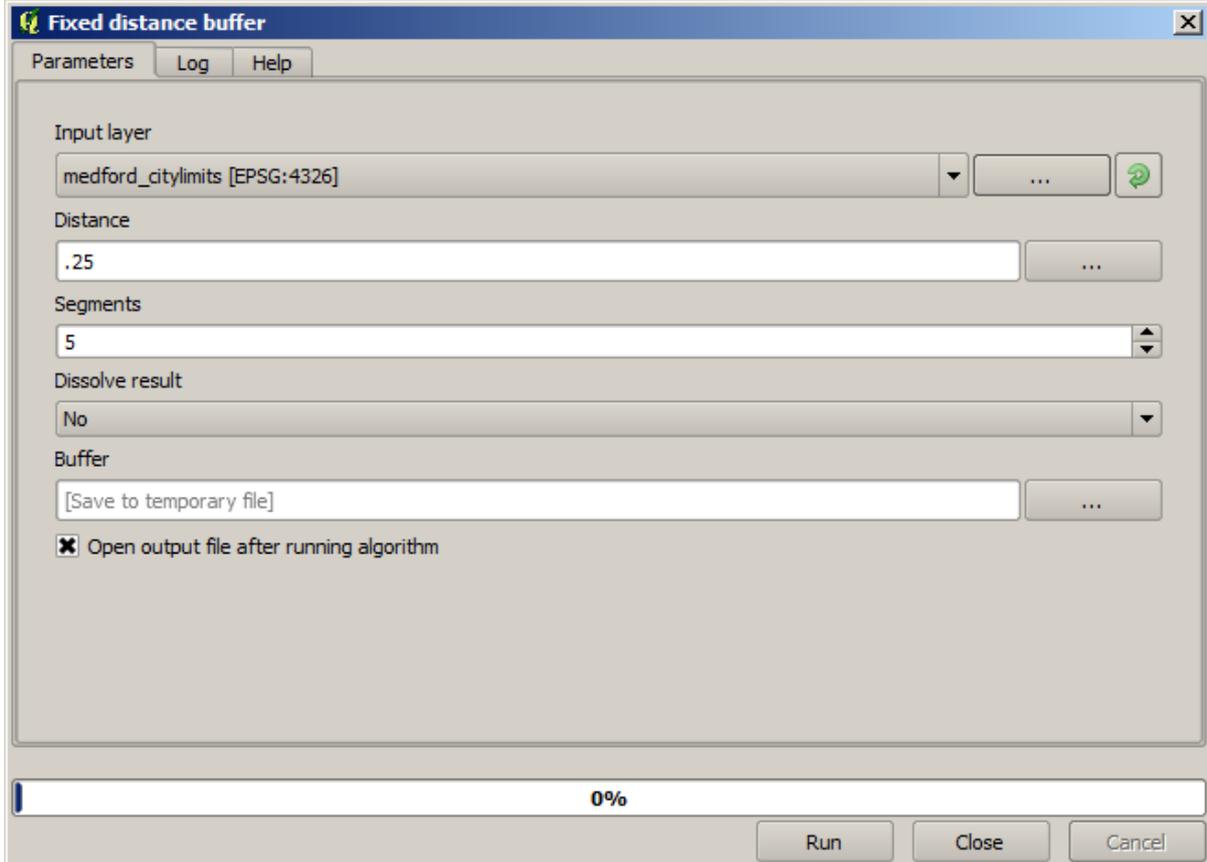
それらの両方が適切な地理アルゴリズムで簡単に解決できます。

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ーのバウンディングボックスを含むレイヤーを作成し、それから厳密に必要なよりもう少しカバーするラスターレイヤを有するように、それをバッファリングします。

バウンディングボックスを計算するために、レイヤーの範囲からポリゴン アルゴリズムを使用できます

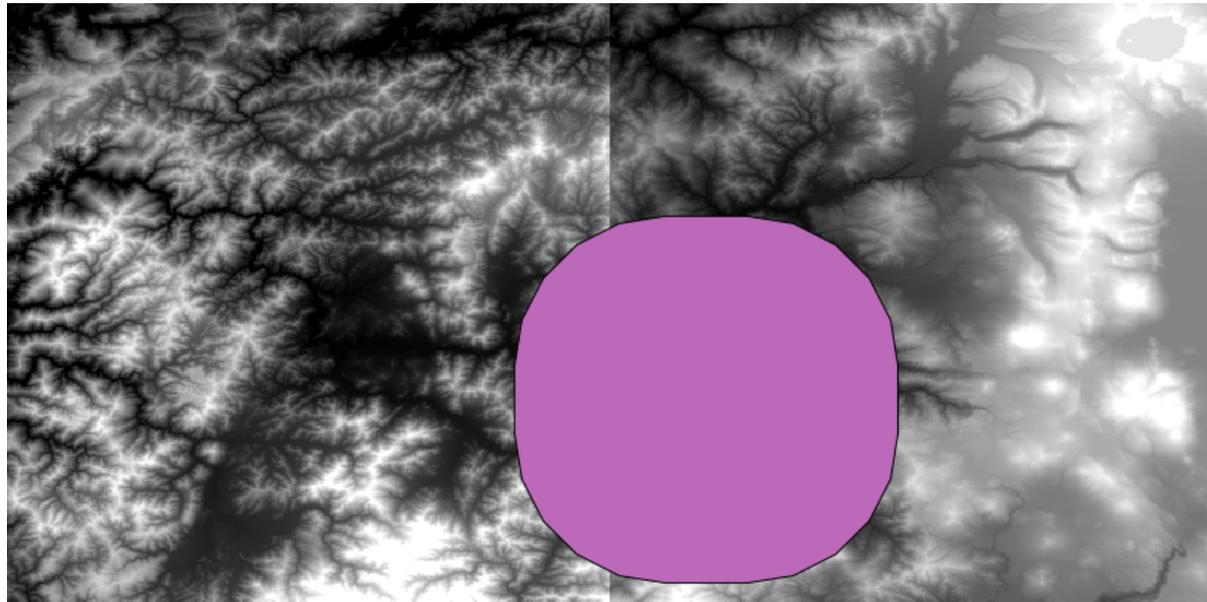


それをバッファリングするために、以下のパラメータ値で、固定距離バッファ アルゴリズムを使用します。

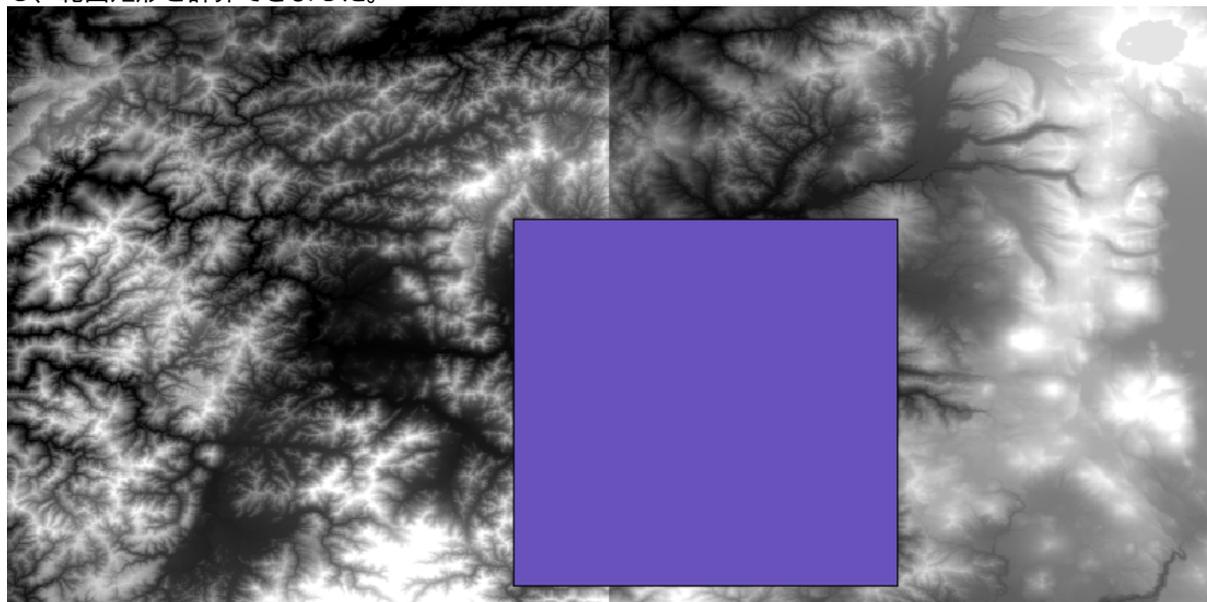


警告: 構文は、最近のバージョンに変更しました。距離とアークの頂点の両方を 0.25 に設定します

ここで上に示したパラメータを用いて得られた結果のバウンディングボックスであります



これは、丸みを帯びた箱ですが、簡単にレイヤーの範囲からポリゴン アルゴリズムを実行することによって、正方形の角度で同等のボックスを取得できます。最初の市域をバッファリングして、一ステップ省略し、範囲矩形を計算できました。



ラストは、ベクターと別の投影を有することがわかります。したがって、ワープ（再投影）ツールを使用して、さらに進む前にそれらを再投影する必要があります。

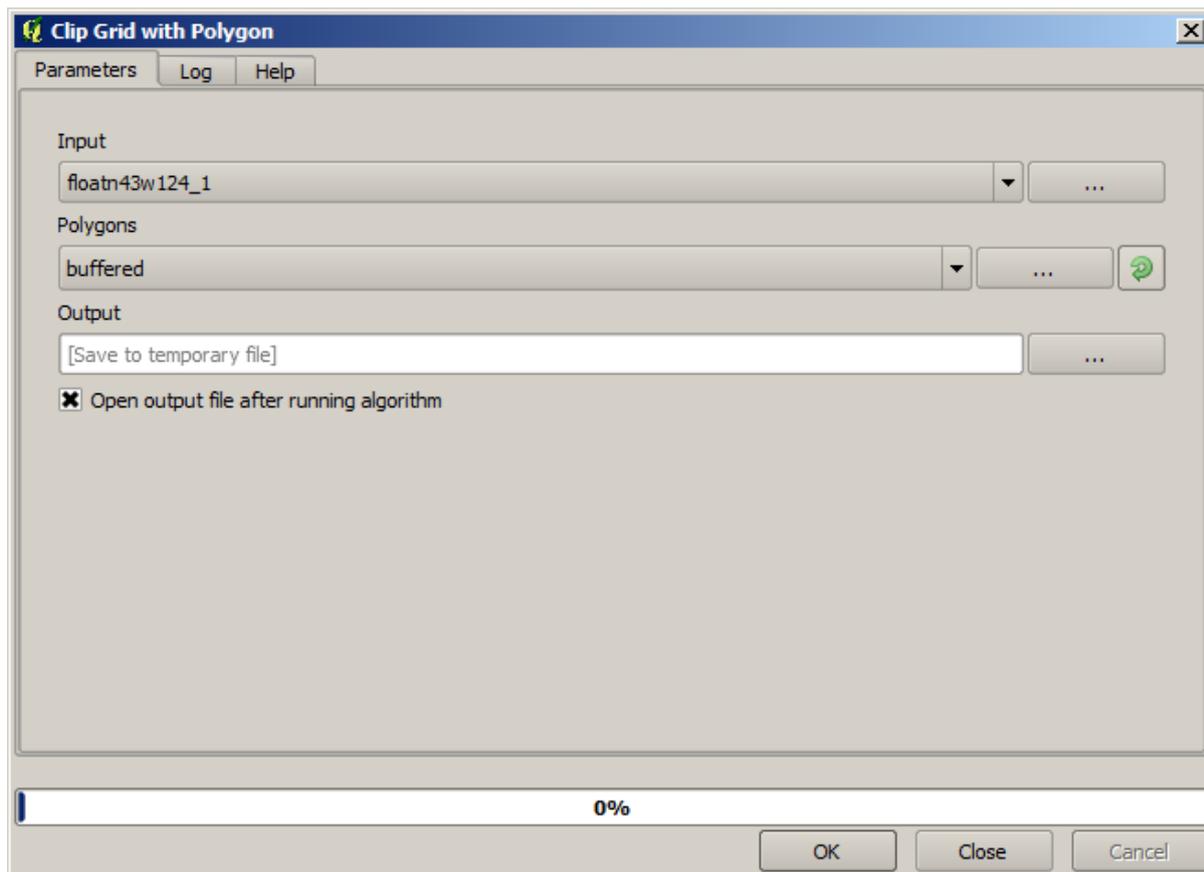
The screenshot shows the 'Parameters' dialog box for a tool in QGIS. The dialog has three tabs: 'Parameters', 'Log', and 'Help'. The 'Parameters' tab is active. The dialog contains the following fields and controls:

- Input layer:** A dropdown menu showing 'dem2 [EPSG:4269]' and a button with three dots.
- Source SRS (EPSG Code):** A text input field containing 'EPSG:4269' and a button with three dots.
- Destination SRS (EPSG Code):** A text input field containing 'EPSG:4326' and a button with three dots.
- Output file resolution in target georeferenced units (leave 0 for no change):** A text input field containing '0,000000' and a button with three dots.
- Resampling method:** A dropdown menu showing 'near'.
- Additional creation parameters:** An empty text input field.
- Output layer:** A text input field containing '[Save to temporary file]' and a button with three dots.
- Open output file after running algorithm:** A checked checkbox.

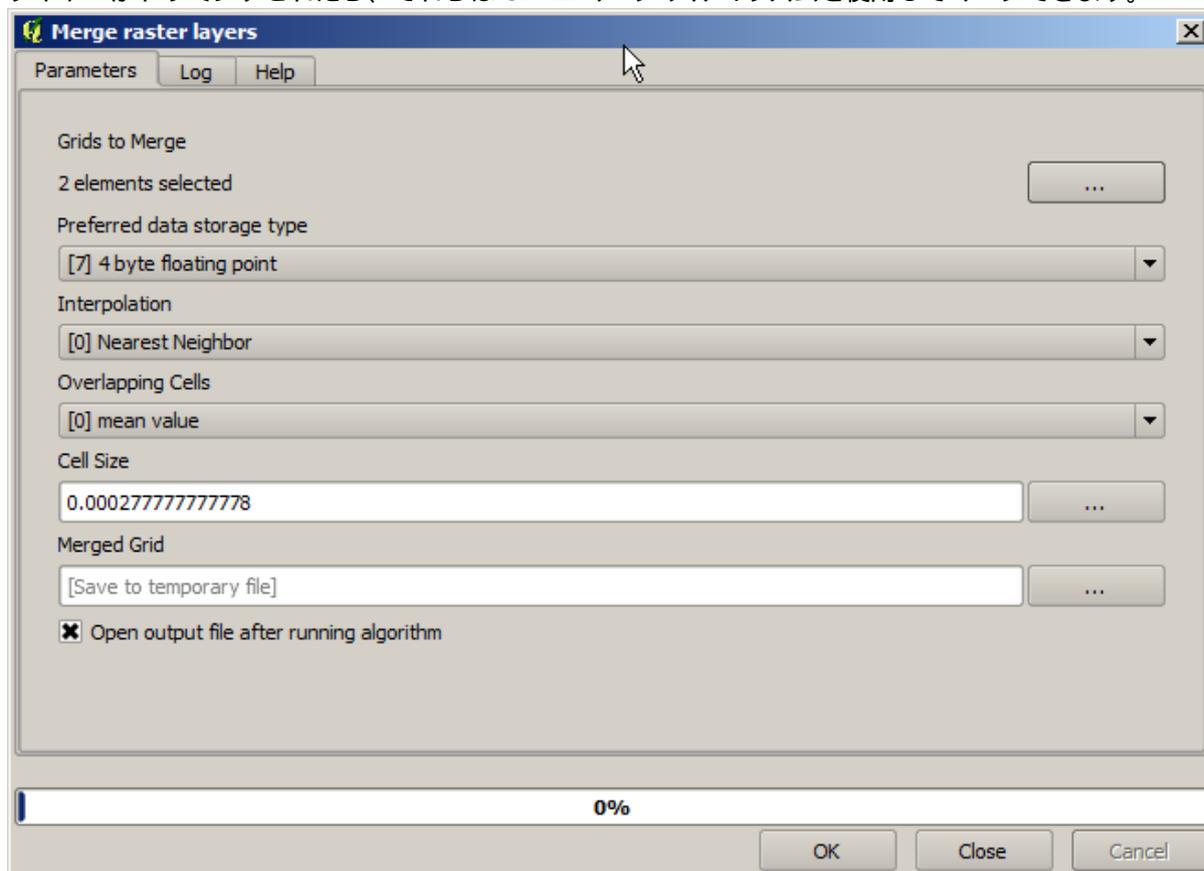
At the bottom of the dialog, there is a progress bar showing '0%' and three buttons: 'Run', 'Cancel', and 'Close'.

ノート: 最近のバージョンでは、より複雑なインターフェースを持っています。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。

入手したいラスターレイヤのバウンディングボックスが含まれるこのレイヤーでポリゴンでラスタをクリップアルゴリズムを使用して、ラスターレイヤの両方をトリミングできます。

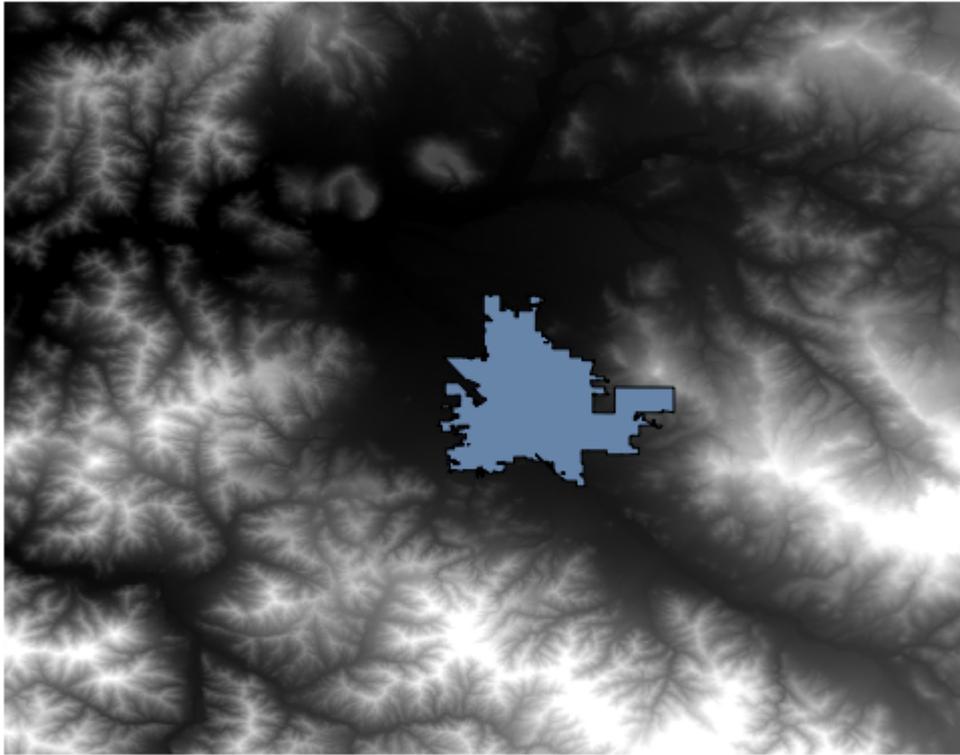


レイヤーがトリミングされたら、それらは GDAL マージ アルゴリズムを使用してマージできます。



ノート: 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ないようにします。しかしながら、マージするいくつかのレイヤーがあり、それらがかなり大きなサイズを持っている場合は、それが後工程に処理が困難であるよりも大きなレイヤー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、レイヤーは2つだけなので、今それを心配することはありません。

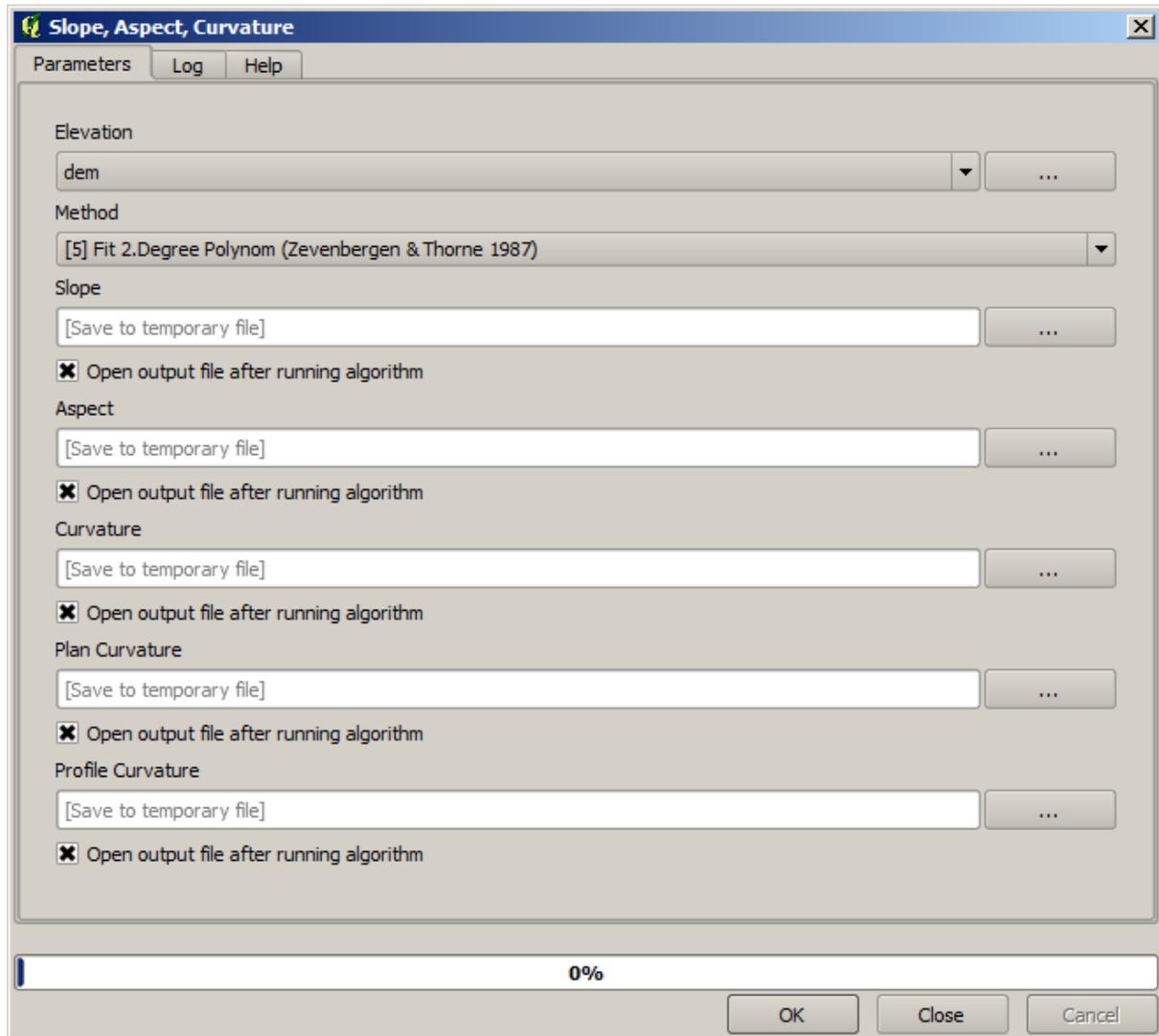
それによって、私たちが望む最後の DEM が得られます。



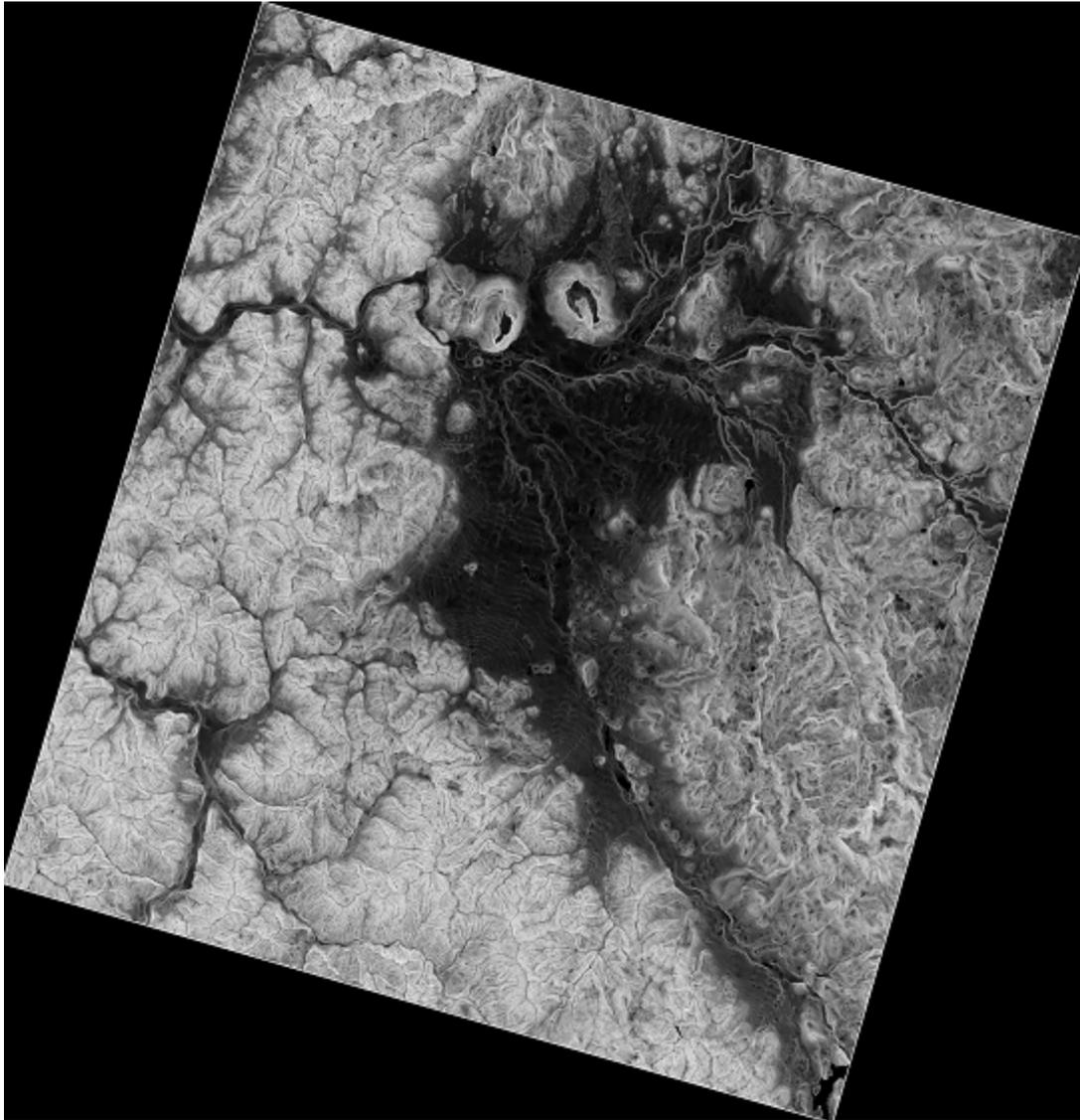
さて傾斜レイヤーを計算する時間になりました。

傾斜レイヤーは傾斜・方向・曲率 アルゴリズムを用いて計算できますが、標高値はメートル単位ですがセルサイズはメートルで表現されていないため、最後の工程で得られた DEM は入力として適していません (レイヤーは地理座標を持つ CRS 使用して)。再投影が必要とされています。ラスターレイヤを再投影するために、ワーブ (再投影) アルゴリズムを再び使用できます。単位 (例えば 3857) メートルで CRS に再投影、その後、正しく SAGA や GDAL のいずれかで、傾きを計算できます。

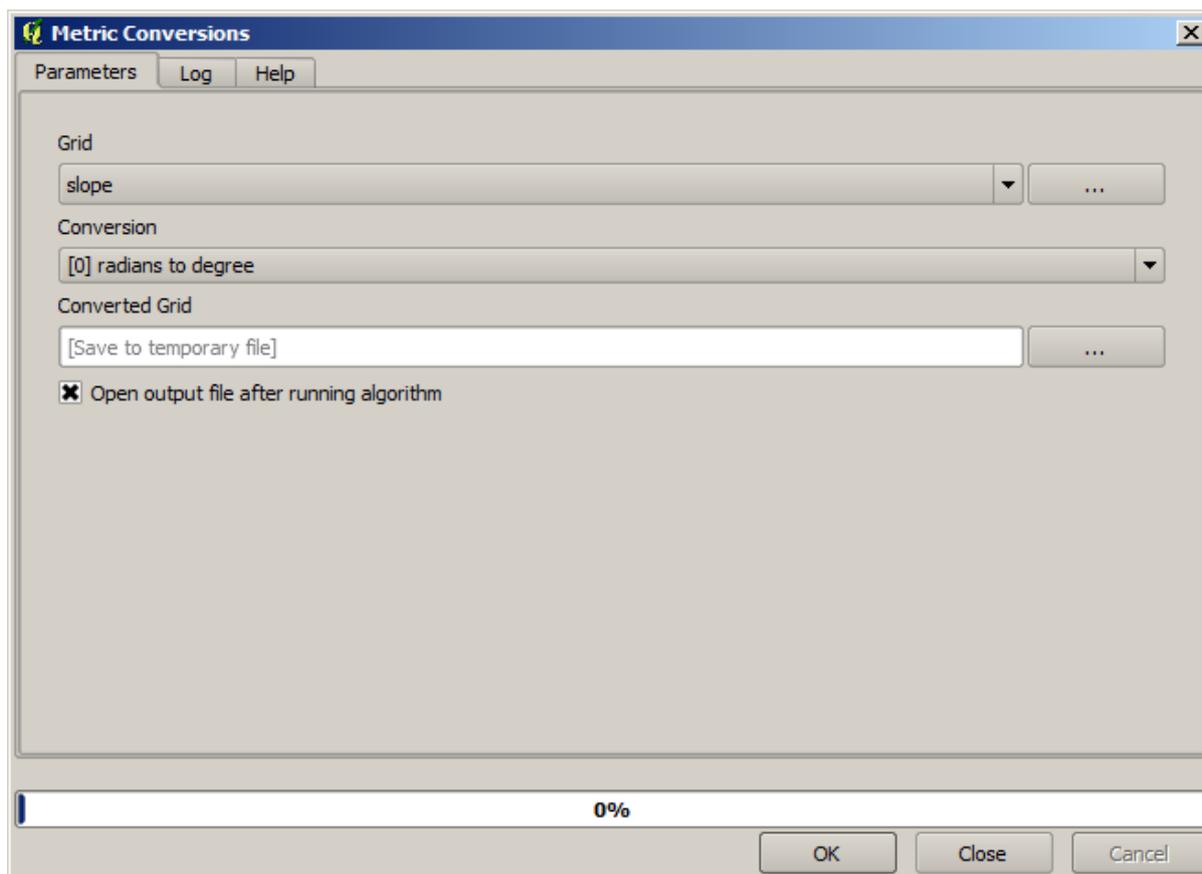
新 DEM では、傾きが計算できるようになりました。



そして、これが結果の傾斜レイヤーです。



傾斜・方向・曲率 アルゴリズムによって作成される傾斜は、度またはラジアンで表現できます。度は、より実用的で一般的な単位です。ラジアンでそれを計算した場合は、メトリック変換 アルゴリズムが変換を行うのに役立ちます（しかし、そのアルゴリズムが存在していた知らなかった場合は、すでに使用しているラスタ計算機を使用できたでしょう）。



*ラスタレイヤー*投影をし直すと再変換斜面レイヤーを再投影、望んでいた最終レイヤーを取得します。

警告: TODO : 画像を追加します。

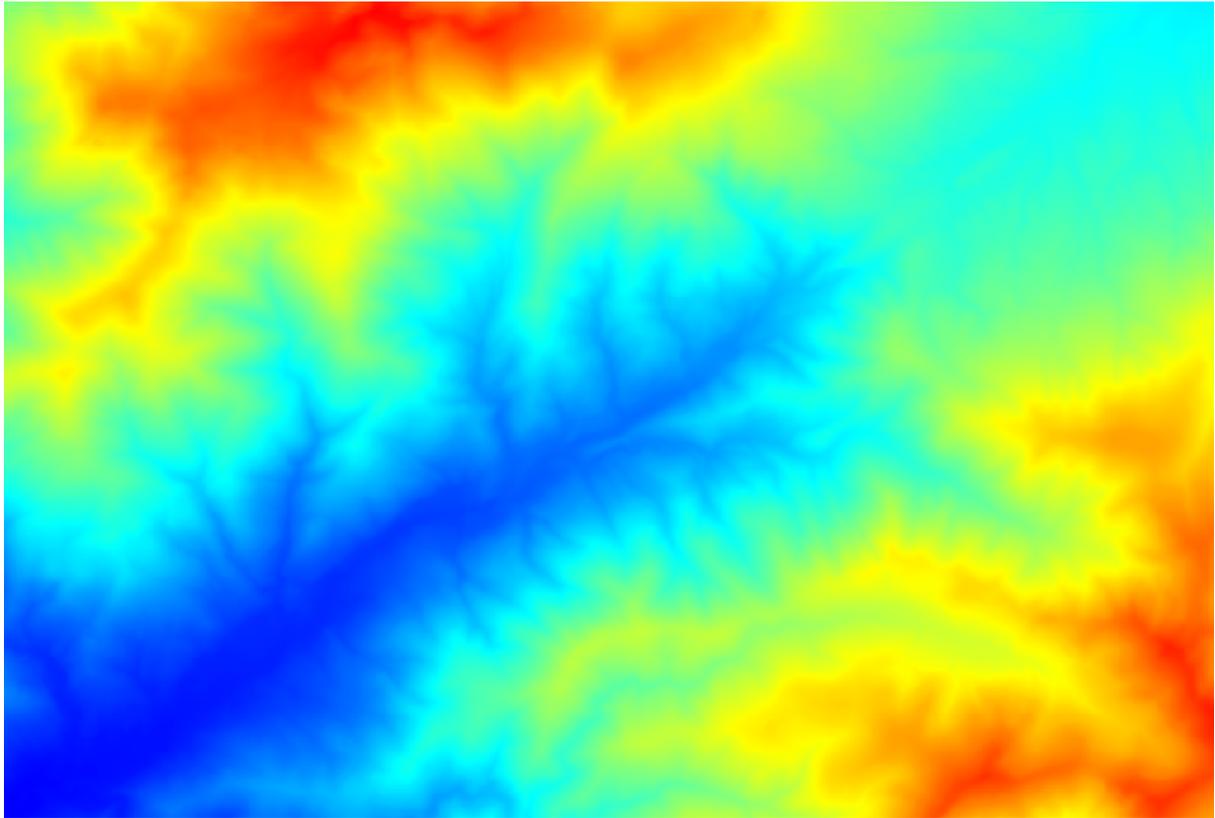
再投影プロセスは、最初のステップの 1 で計算バウンディングボックス外のデータを格納するための最終レイヤーを引き起こしている可能性があります。これは、ベース DEM を得るために行ったように、再びそれをクリッピングすることによって解決できます。

17.16 水文解析

ノート: このレッスンでは、ある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。

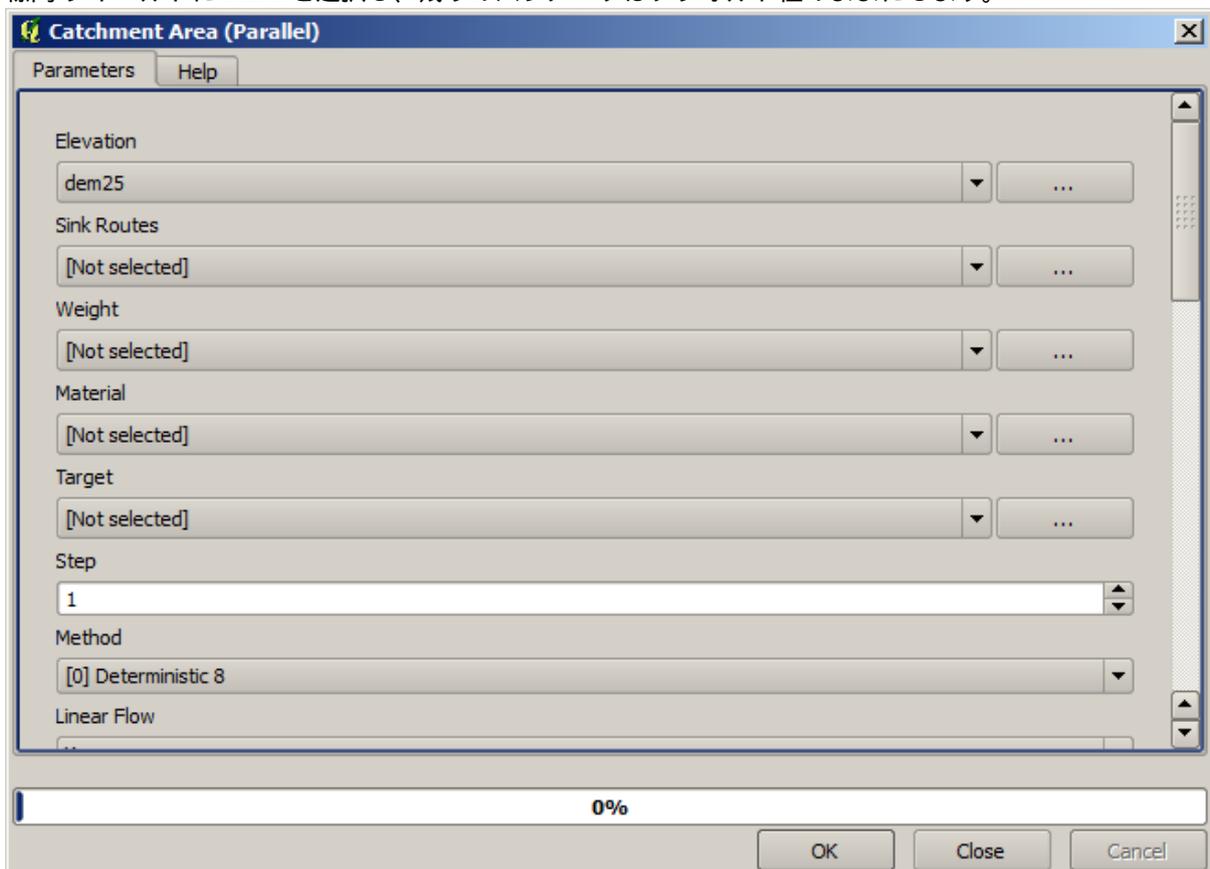
このレッスンでは、ある水文解析をしていきます。DEM を皮切りに、水路ネットワークを抽出し、流域を描き、いくつかの統計を計算していきます。

最初は、レッスンデータ、DEM だけが含まれている、を持つプロジェクトをロードすることです。



実行する最初のモジュールは 流域面積 です (一部 SAGA バージョンではそれは 累積流量 (トップダウン) と呼ばれる)。流域面積 という名前の他の誰にも使用できます。それらは下では異なるアルゴリズムを持っているが、結果は基本的に同じです。

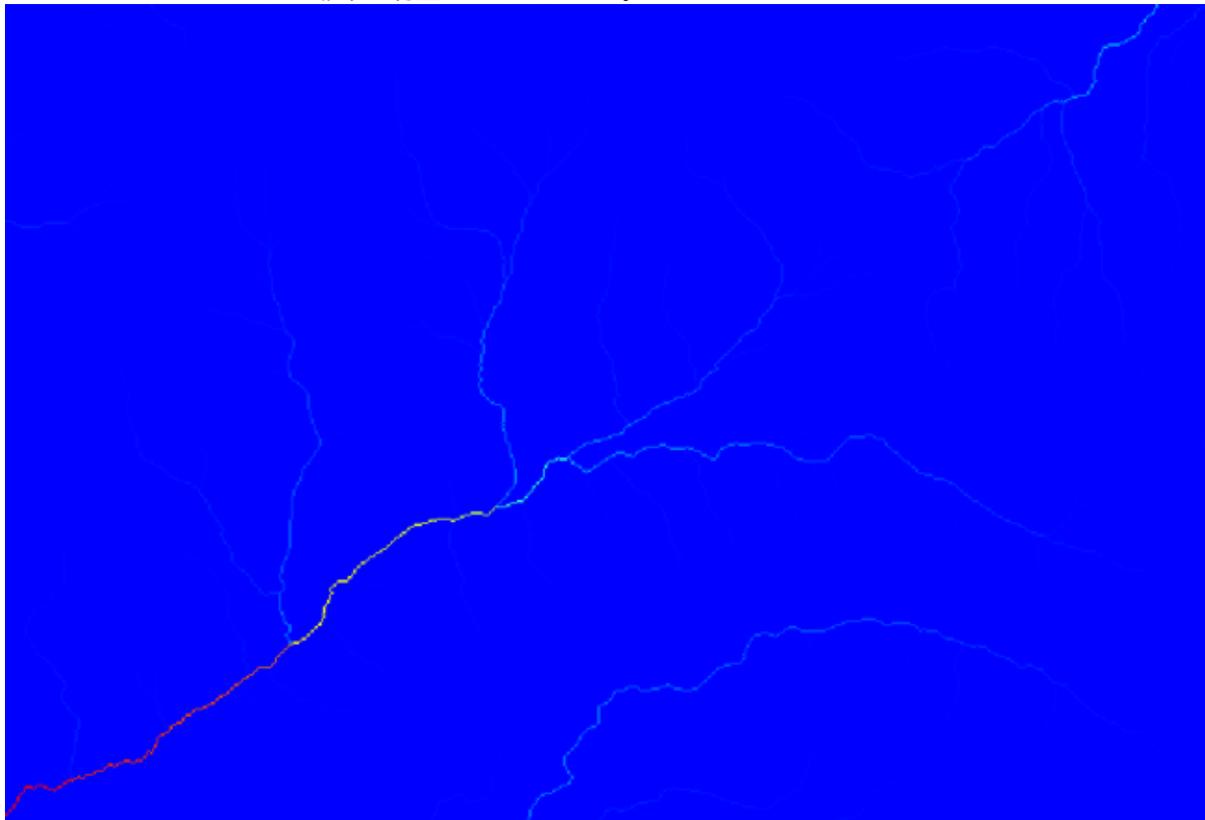
標高 フィールドに DEM を選択し、残りのパラメータはデフォルト値のままにします。



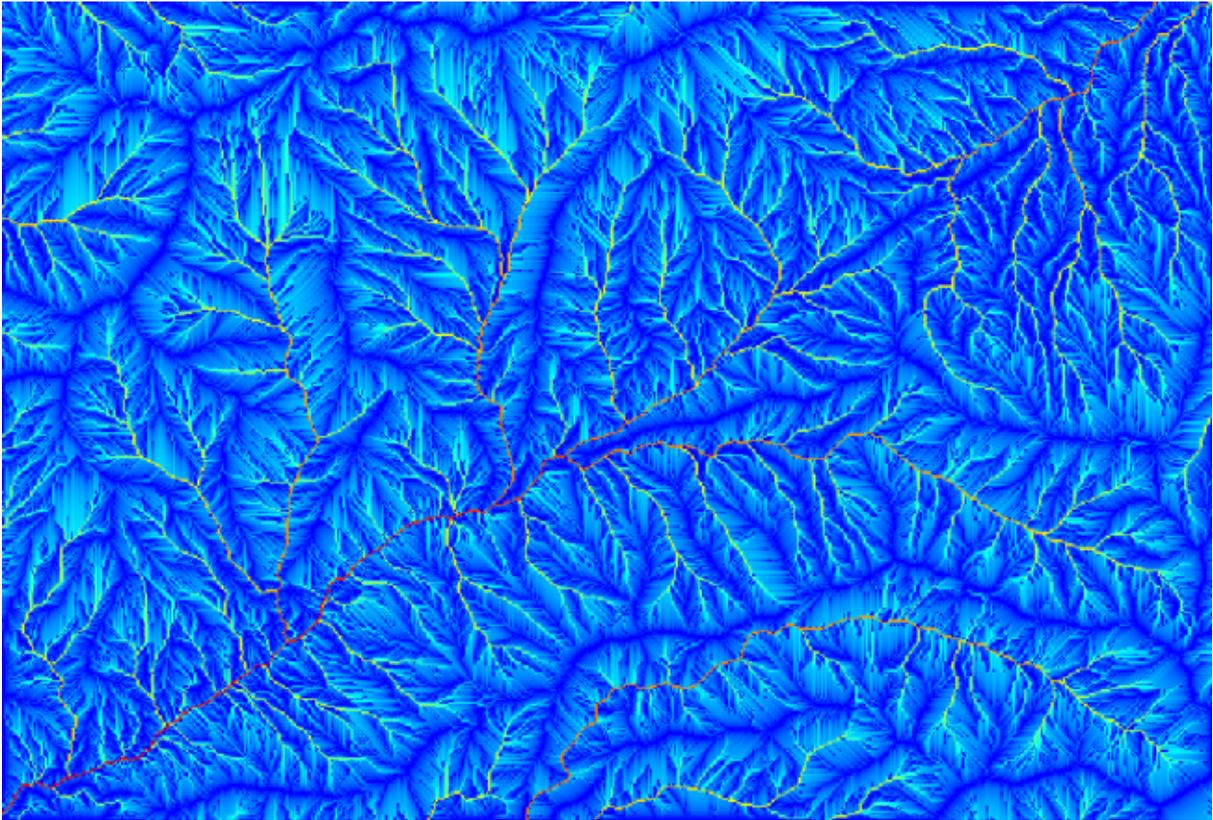
アルゴリズムには多くのレイヤーを計算するものもありますが、流域レイヤーだけが使用するレイヤーです。

ご希望なら他のレイヤーは取り除くことができます。

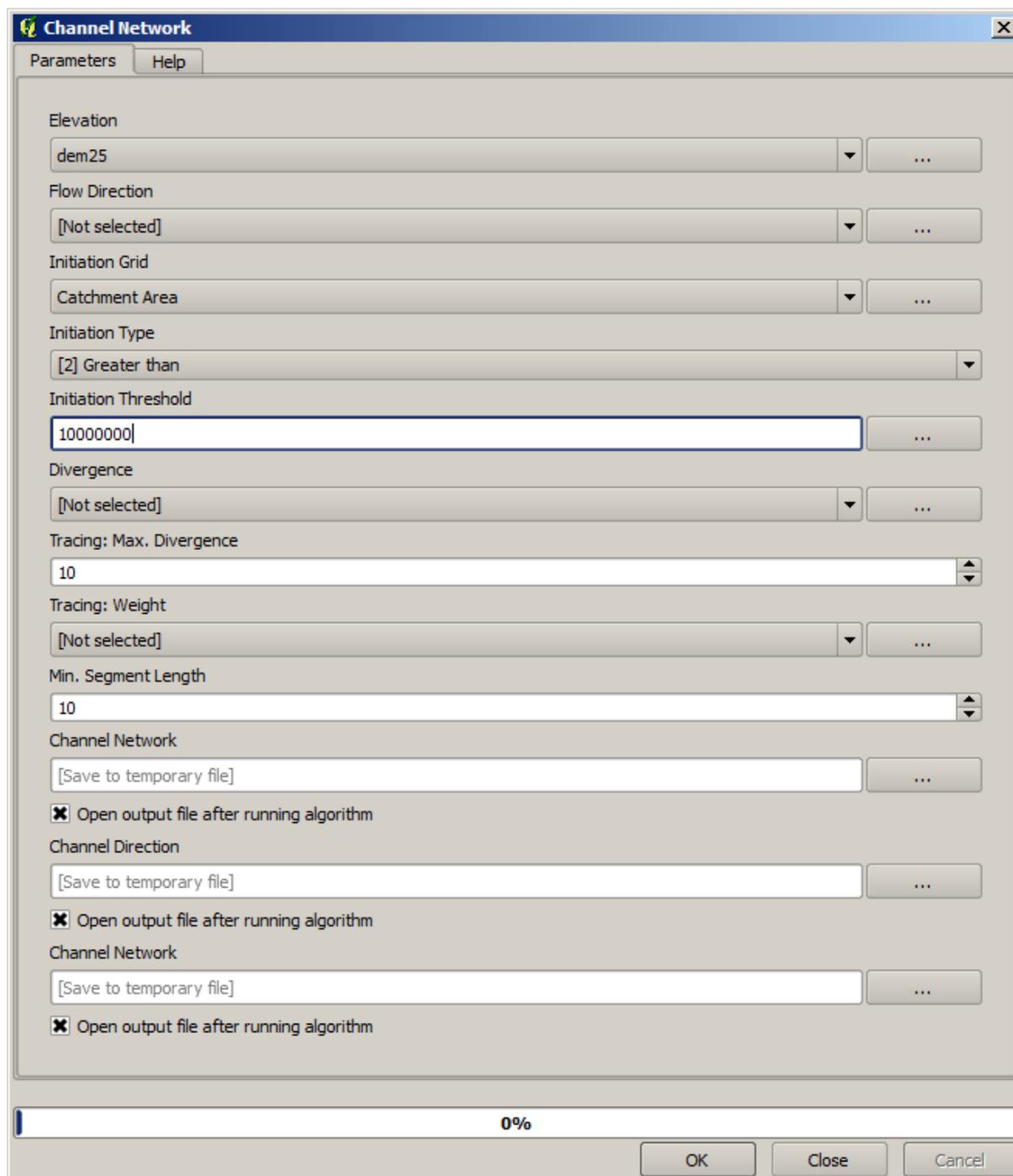
レイヤーのレンダリングは非常に有益ではありません。



なぜかを知るためにはヒストグラムを見ればよく、値が均等に分布していないことがわかり得るでしょう（非常に高い値を持つ少数のセル、水路ネットワークに対応したものがある）。流域面積値の対数を計算するとより多くの情報を伝えるレイヤーが作成されます（ラスタ計算機を使用してそれを行うことができます）。

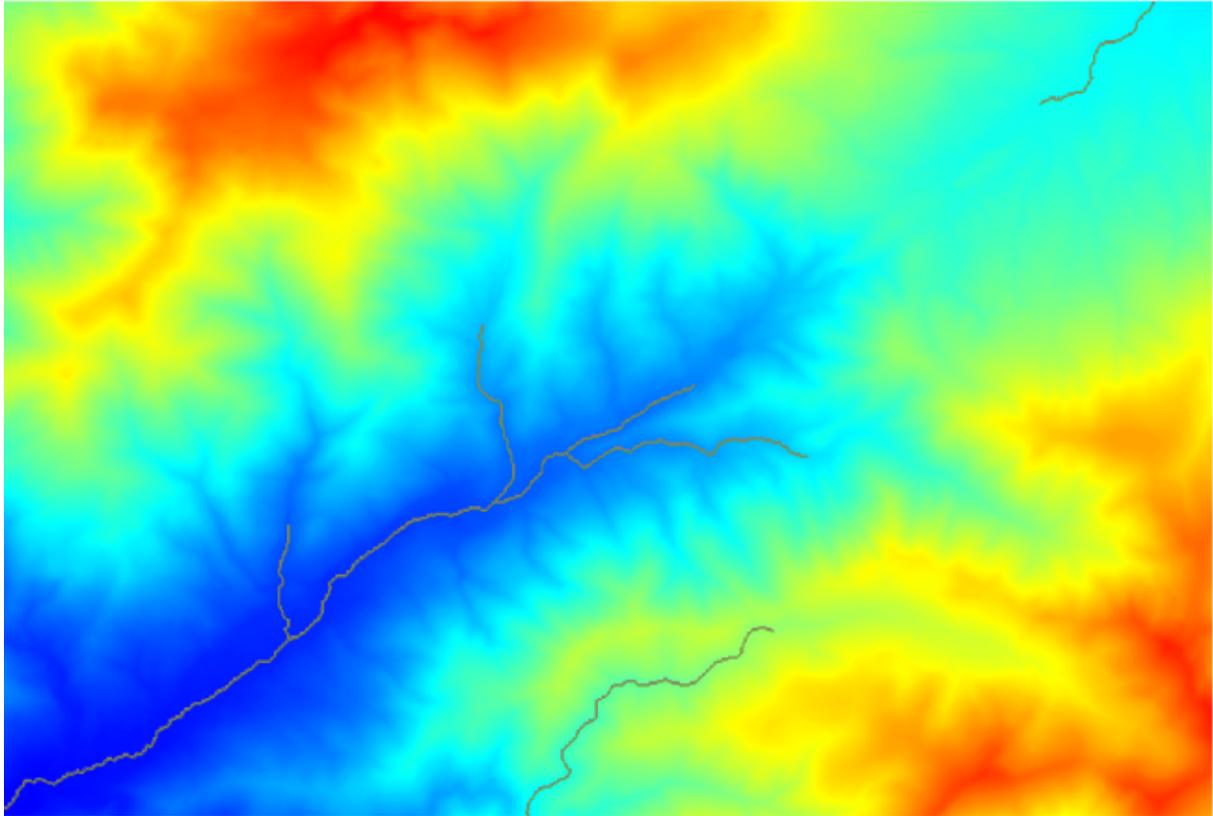


(また、累積流量として知られている)流域は、水路開始のためのしきい値を設定するために使用できます。これは、水路ネットワーク アルゴリズムを使用して行うことができます。それを設定しなければならない方法はこうなります (開始しきい値 が 10,000,000 より大きいことに注意してください)。

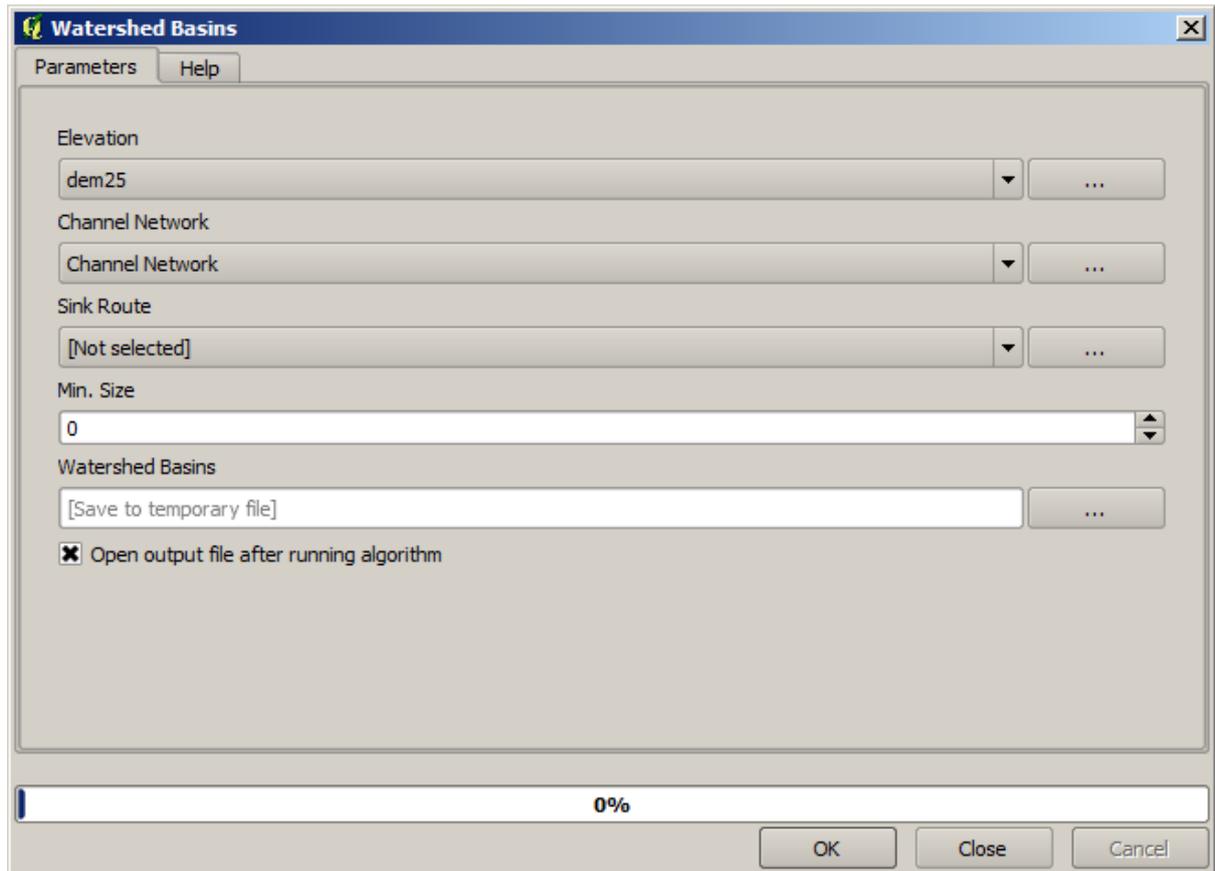


オリジナルの集水域レイヤ、対数のものではなく、を使用してください。そのレイヤはレンダリングする目的だけのためでした。

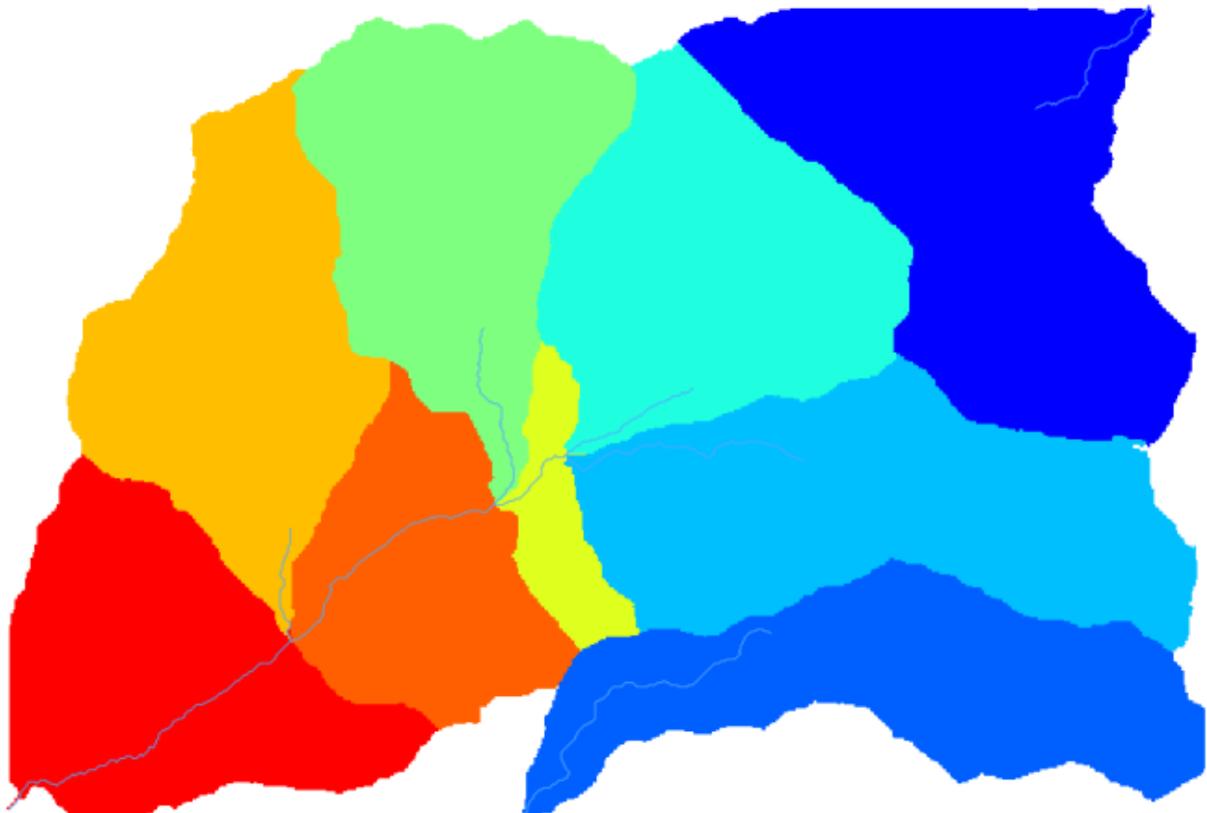
開始しきい値を増やすと、より疎らな水路ネットワークが得られます。減らすと、より密なネットワークが得られます。提案された値では、得られるものはこれです。



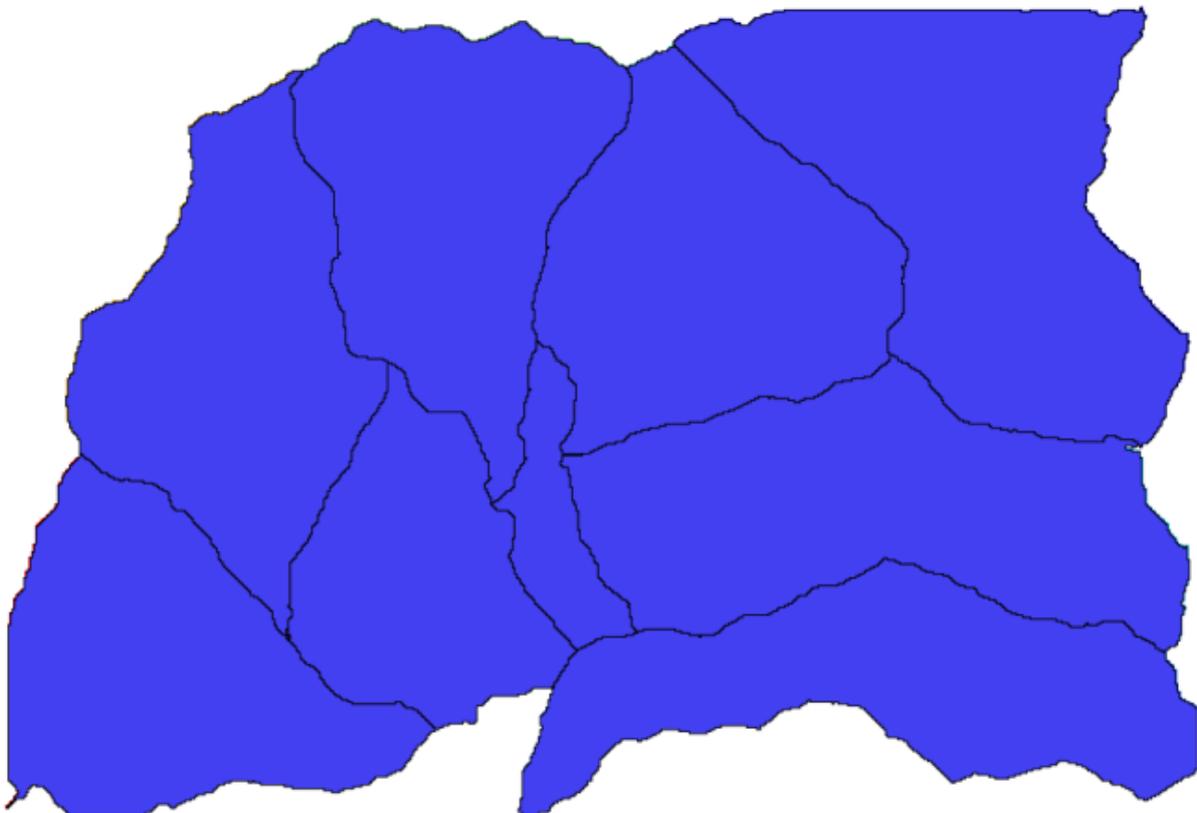
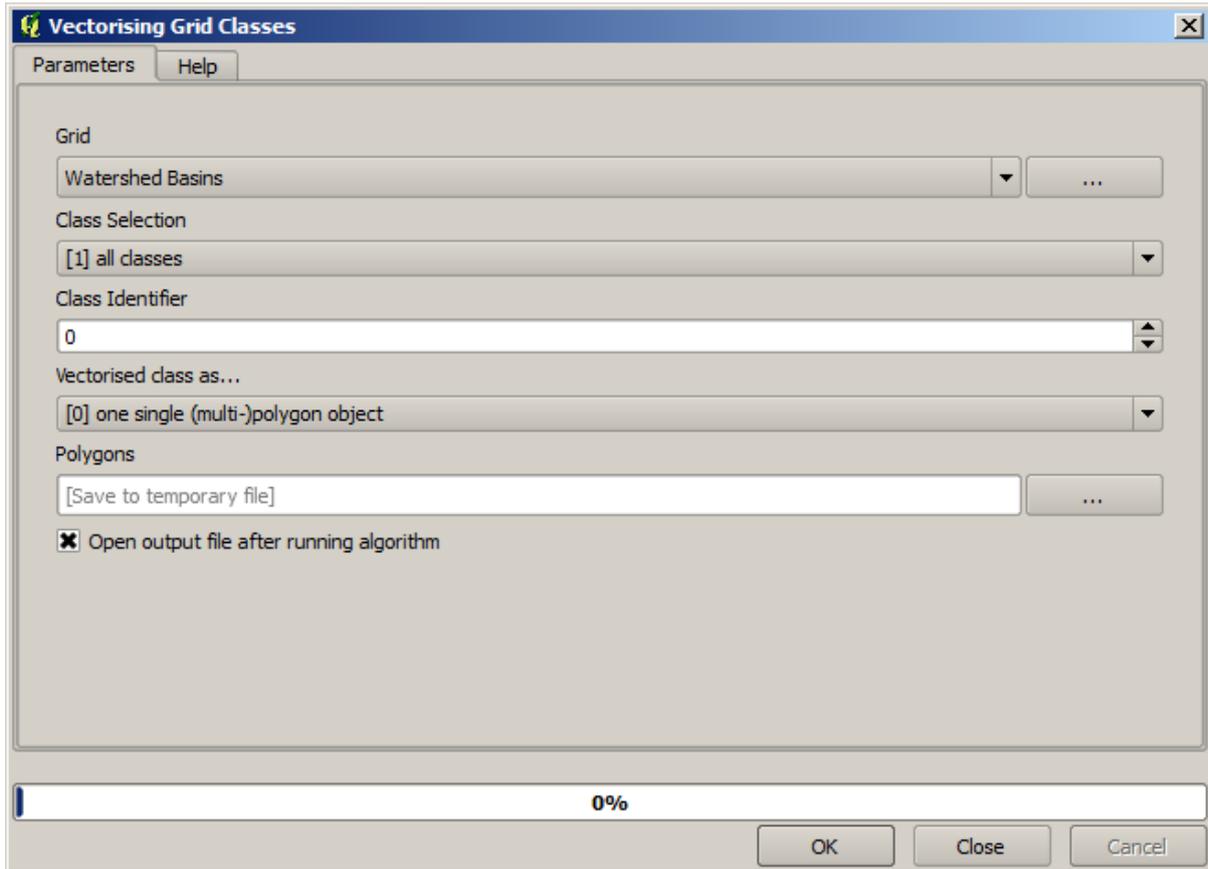
上の画像は得られたベクターレイヤーと DEM だけ表示していますが、同じ水路ネットワークを持つラスターのレイヤーがなければなりません。そのラスターレイヤーが、実際には、使用しているものになります。さてここで、吐出口ポイントとしてその中のすべての接合部を使用して、その水路のネットワークに対応する下位流域を描写する アルゴリズム 流域の流域を使用します。これは、対応するパラメータダイアログボックスをどのように設定する必要があるかです。



そして、得られるものはこれです。



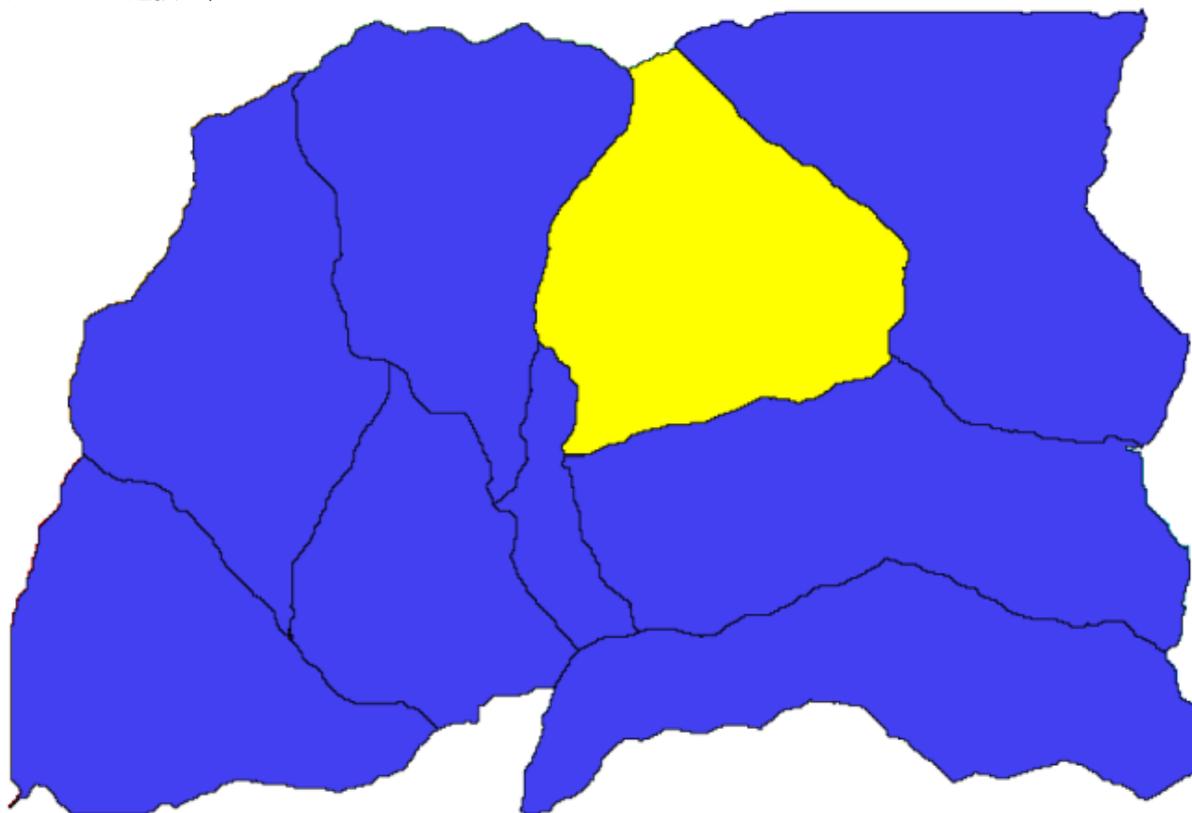
これがラスタの結果です。それはベクトル化グリッドクラス アルゴリズムを使用してベクトル化できます。



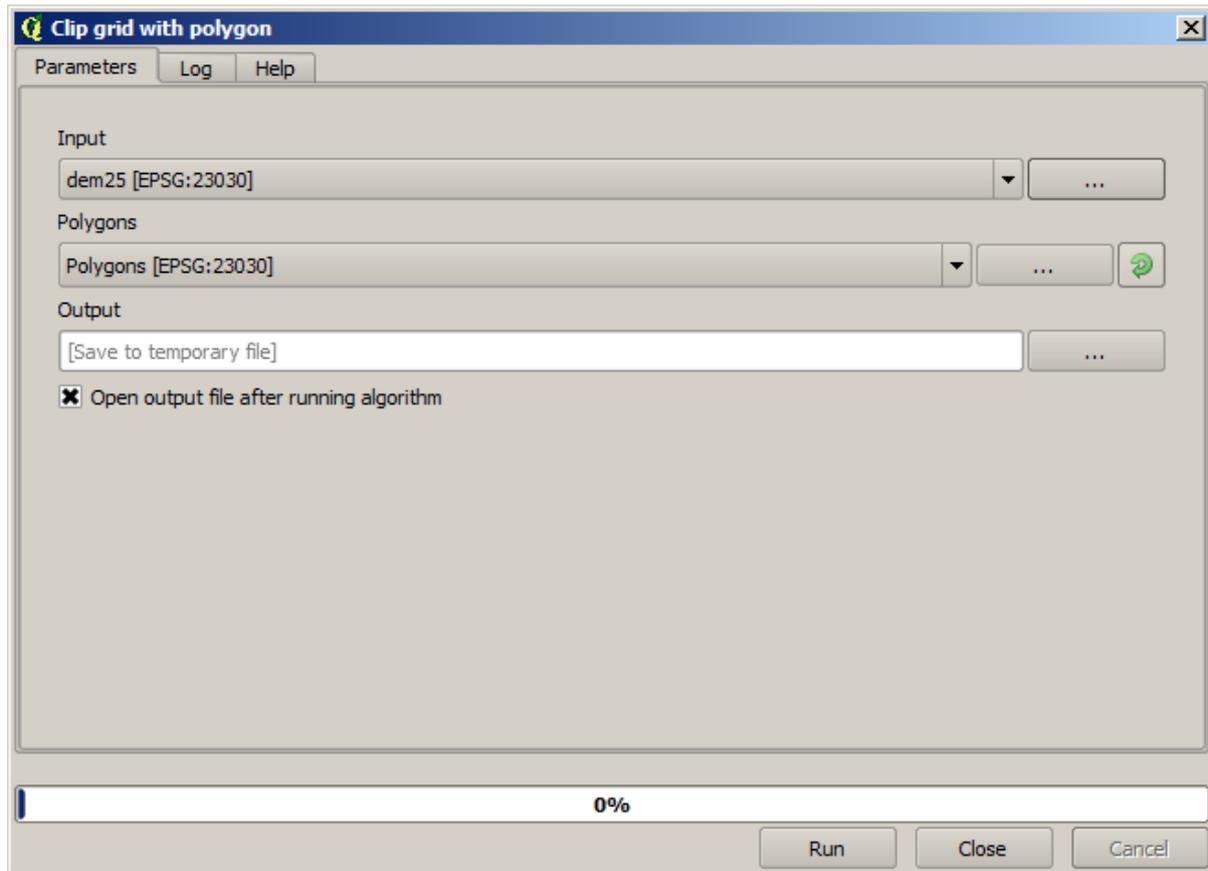
さて、下位流域の一つで標高値についての統計を計算してみましょう。アイデアは、ちょうどその下位流域内の標高を表しレイヤーを持っているし、それらの統計を計算モジュールにそれを渡すことです。

まずは、下位流域を表すポリゴンと元 DEM をクリップしましょう。ポリゴンでラスターをクリップ アルゴリズムを使用します。単一の下位流域ポリゴンを選択し、クリッピングアルゴリズムを呼び出す場合、アルゴリズムが選択を知っているため、そのポリゴンによってカバーされる領域に DEM をクリップできます。

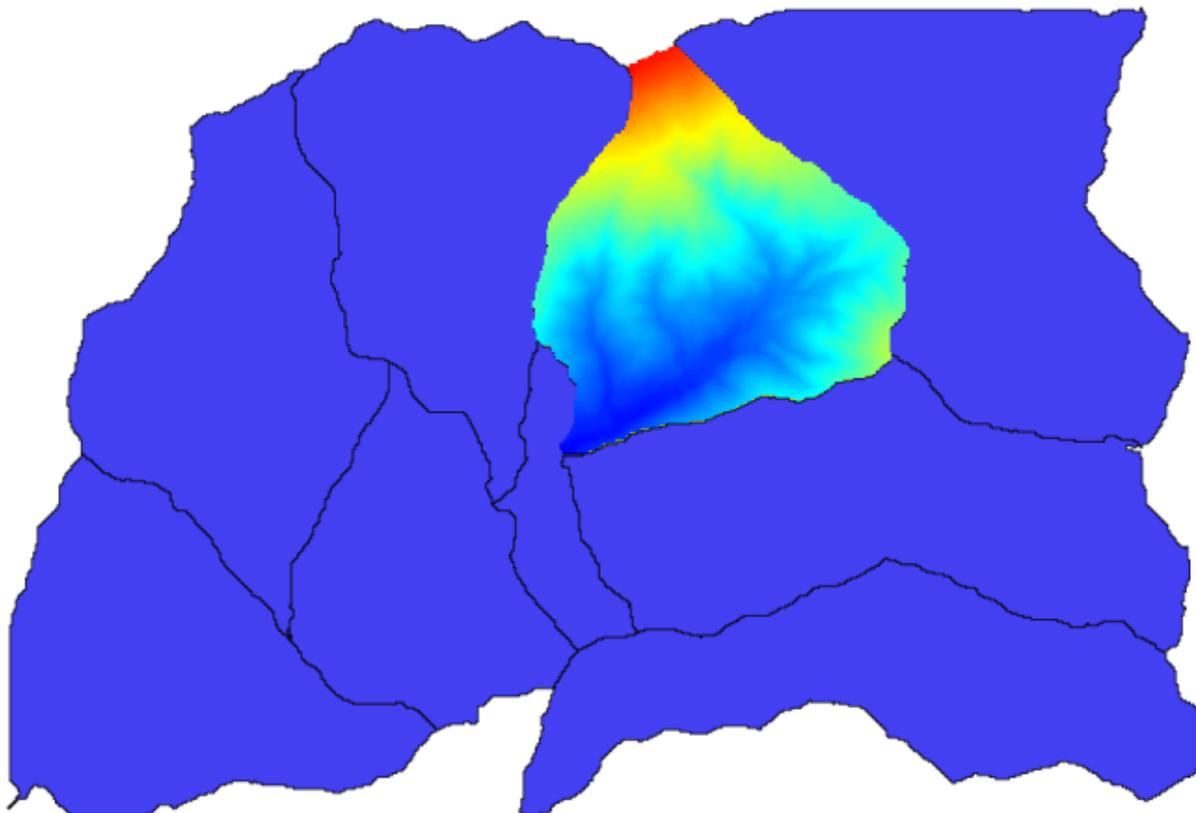
ポリゴンを選択し、



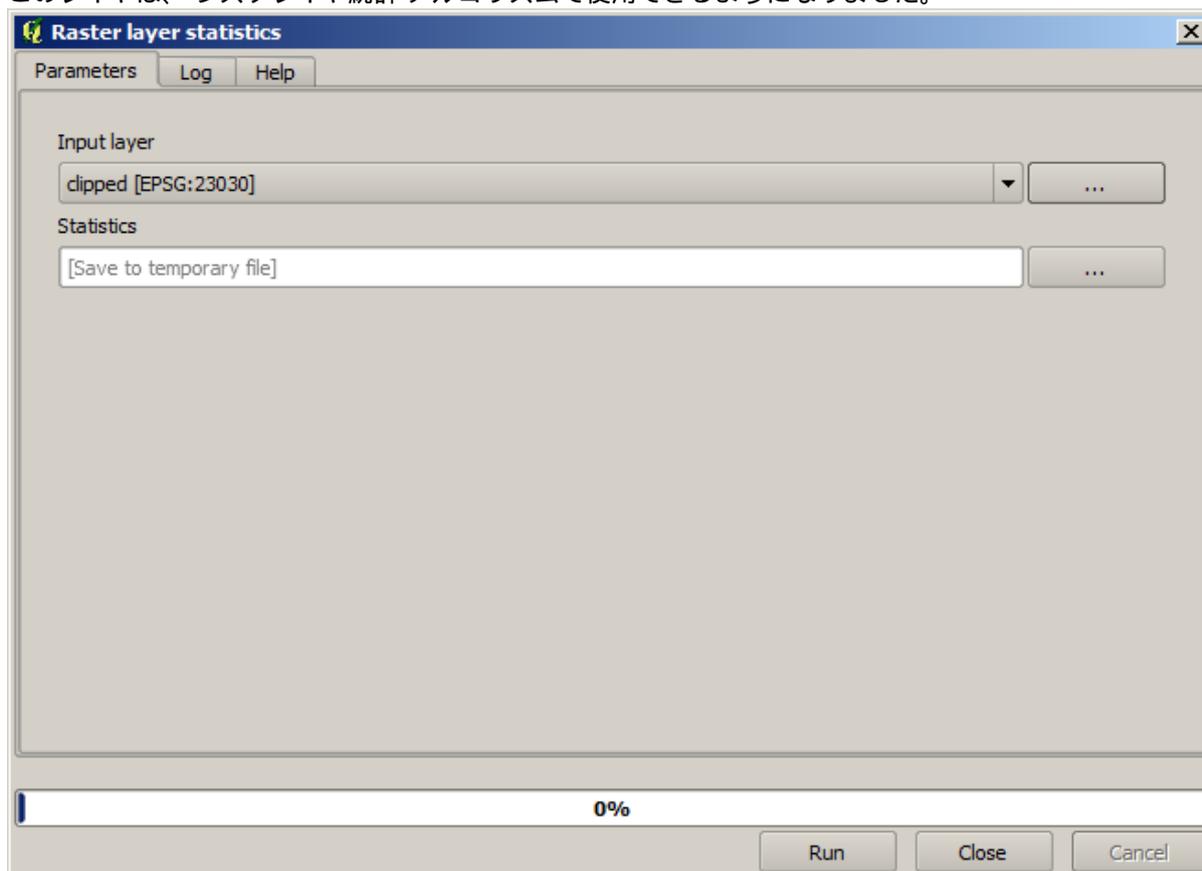
そして次のパラメータを使用してクリッピングアルゴリズムを呼び出します。



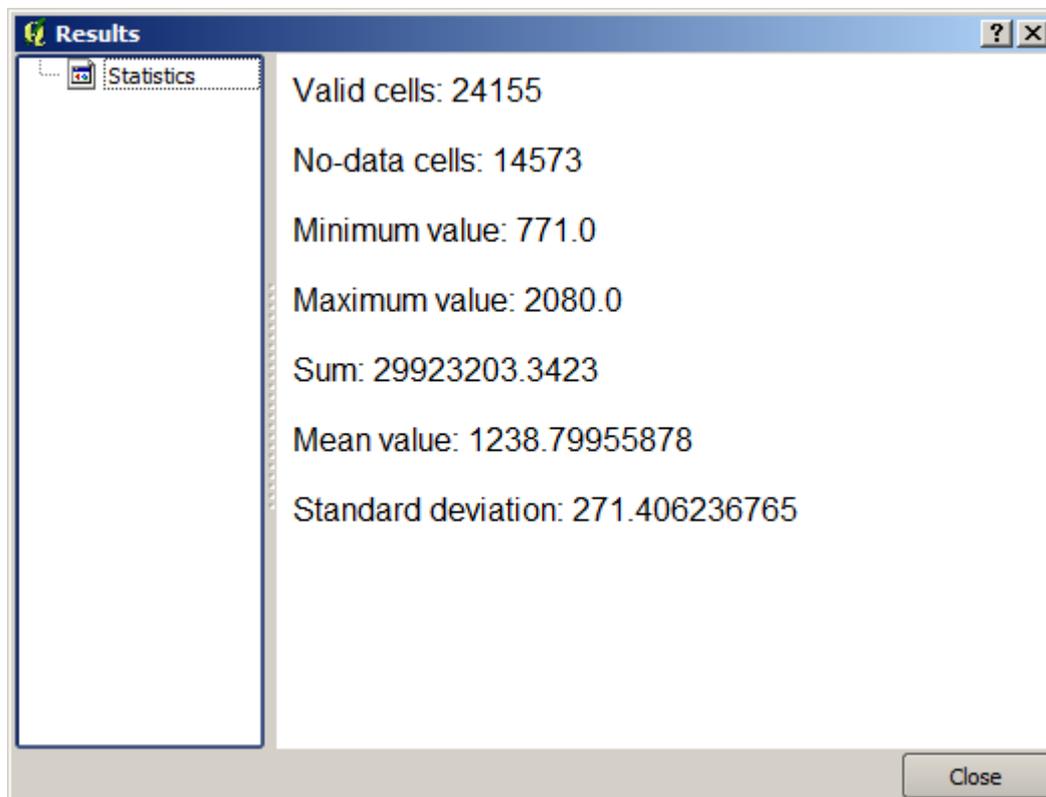
入力フィールドに選択された要素が、もちろん、クリップしたいDEMです。
このようなものが得られます。



このレイヤは、ラスタレイヤ統計 アルゴリズムで使用できるようになりました。



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及び統計計算の両方を使用し、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を助けることができる方法を見ていきます。

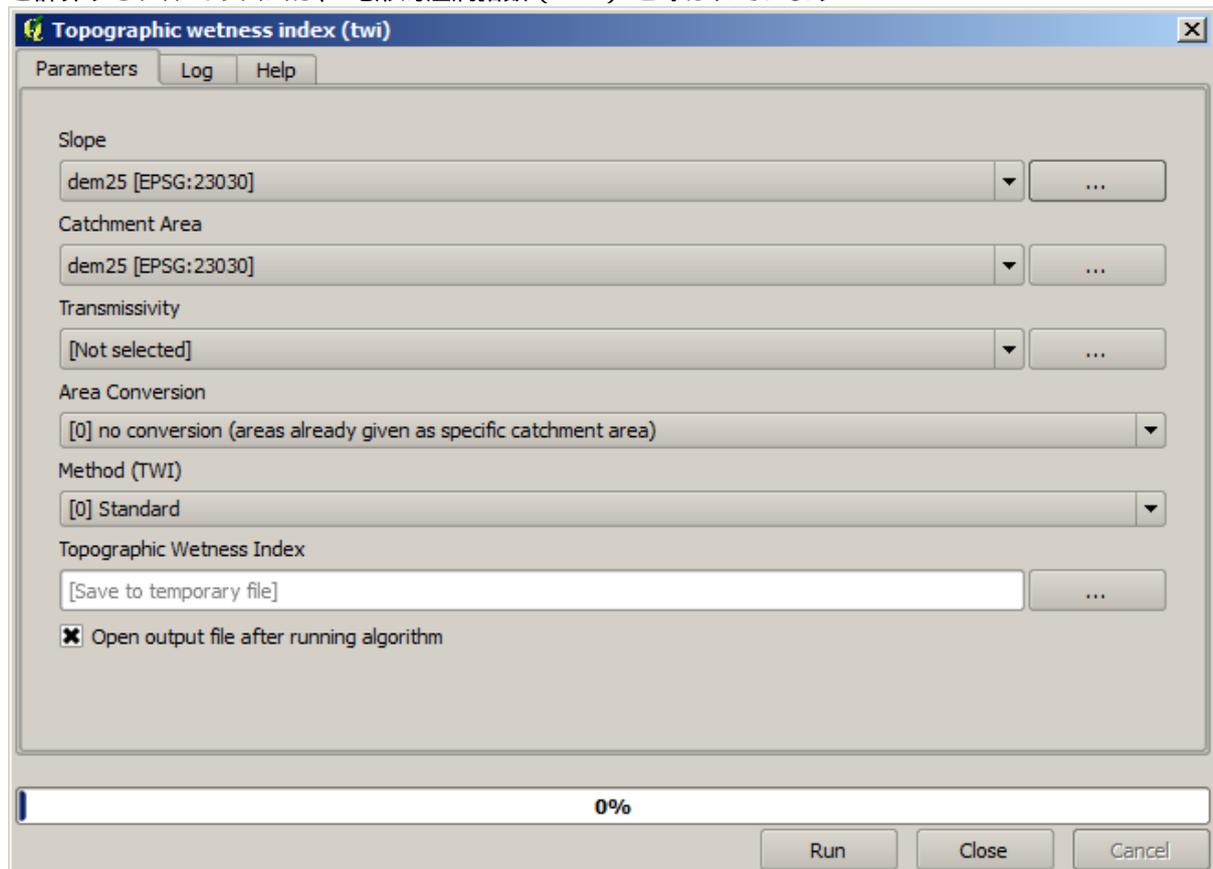
17.17 グラフィカルなモデラーで始まります

ノート: このレッスンでは、グラフィカルなモデラー、我々はワークフローを定義し、アルゴリズムのチェーンを実行するために使用できる強力なコンポーネントを使用します。

処理ツールと通常のセッションは、単一のアルゴリズムを実行している以上のものを含んでいます。通常、それらのいくつかは、結果を得るために実行され、そしてこれらのアルゴリズムのいくつかの出力は、他のもののいくつかの入力として使用されています。

グラフィカルなモデラーを使用して、そのワークフローは、このように全体のプロセスを簡素化し、それを自動化し、単一の実行に必要なすべてのアルゴリズムを実行するモデルに入れることができます。

このレッスンを開始するには、地形的湿潤指数という名前のパラメータを計算しようとしています。それを計算するアルゴリズムは、地形的湿潤指数 (TWI) と呼ばれています

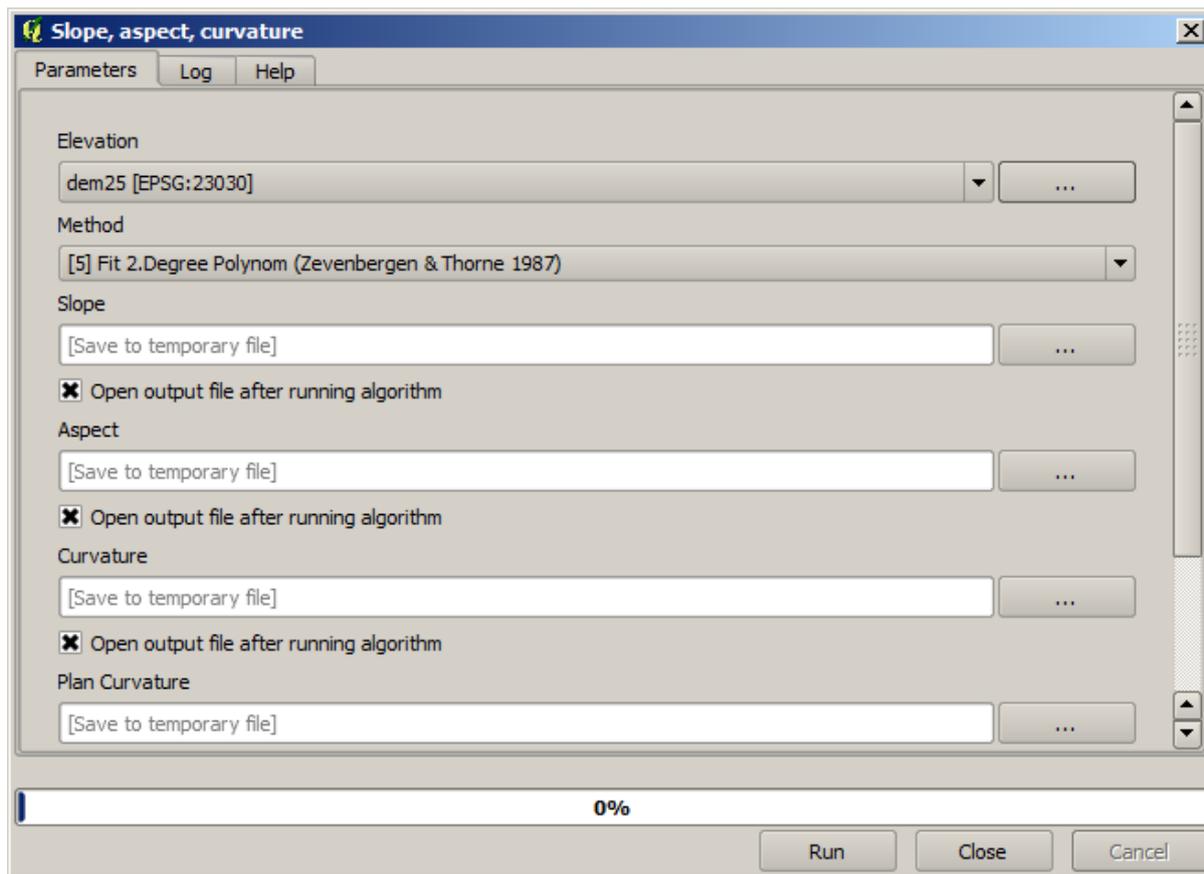


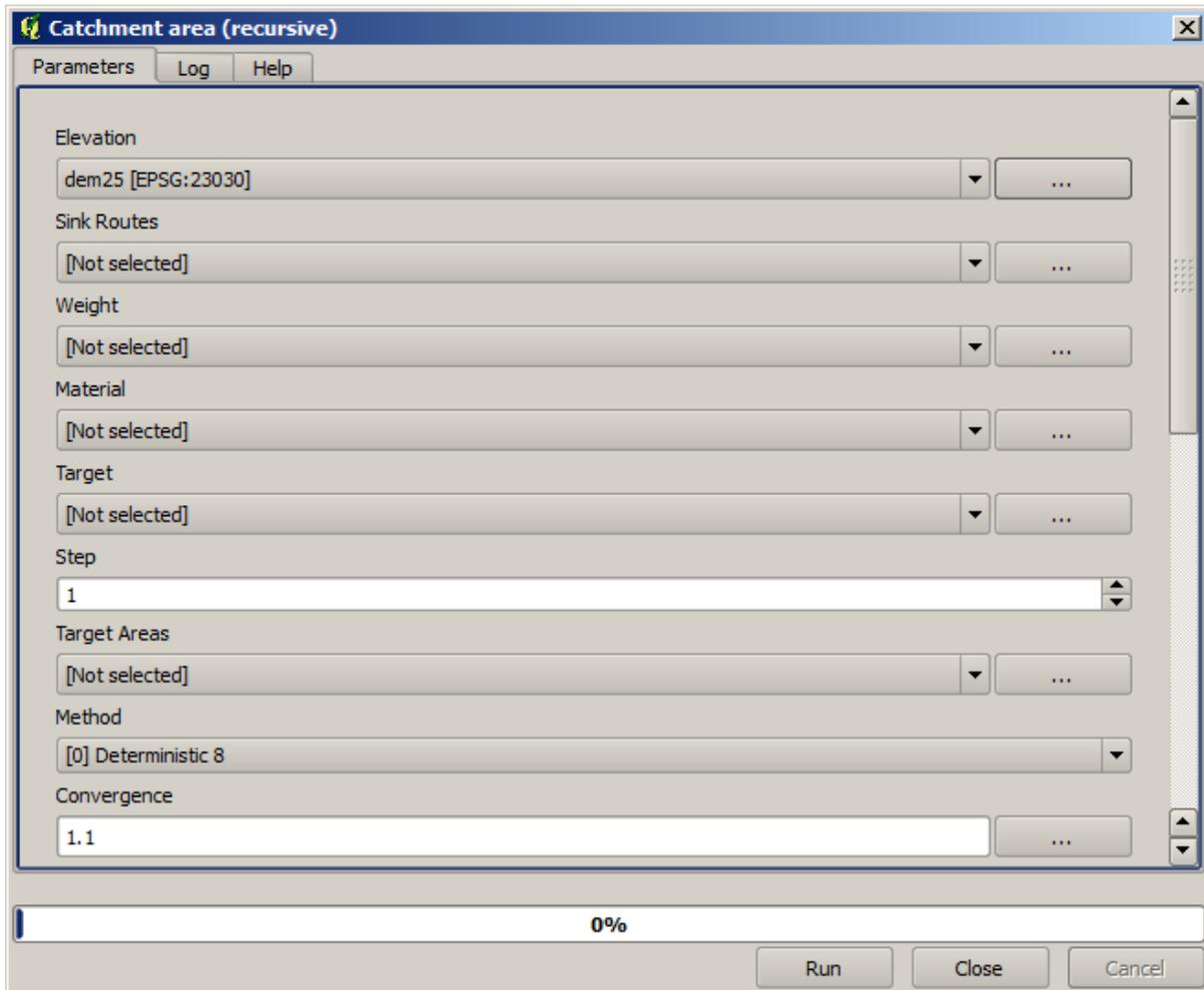
ご覧のように、2つの必須入力があります：スロープと流域面積。オプションの入力もありますが、それを使用するつもりはありませんので、それは無視できます。

このレッスンのためのデータだけ DEM が含まれているので、必要な入力のいずれかを持っていません。しかし、すでに傾きと流域面積を計算するためのアルゴリズムを見てきたことから、その DEM からそれらの両方を計算する方法を知っています。だから、最初にこれらの層を計算して、TWI アルゴリズムのためにそれらを使用できます。

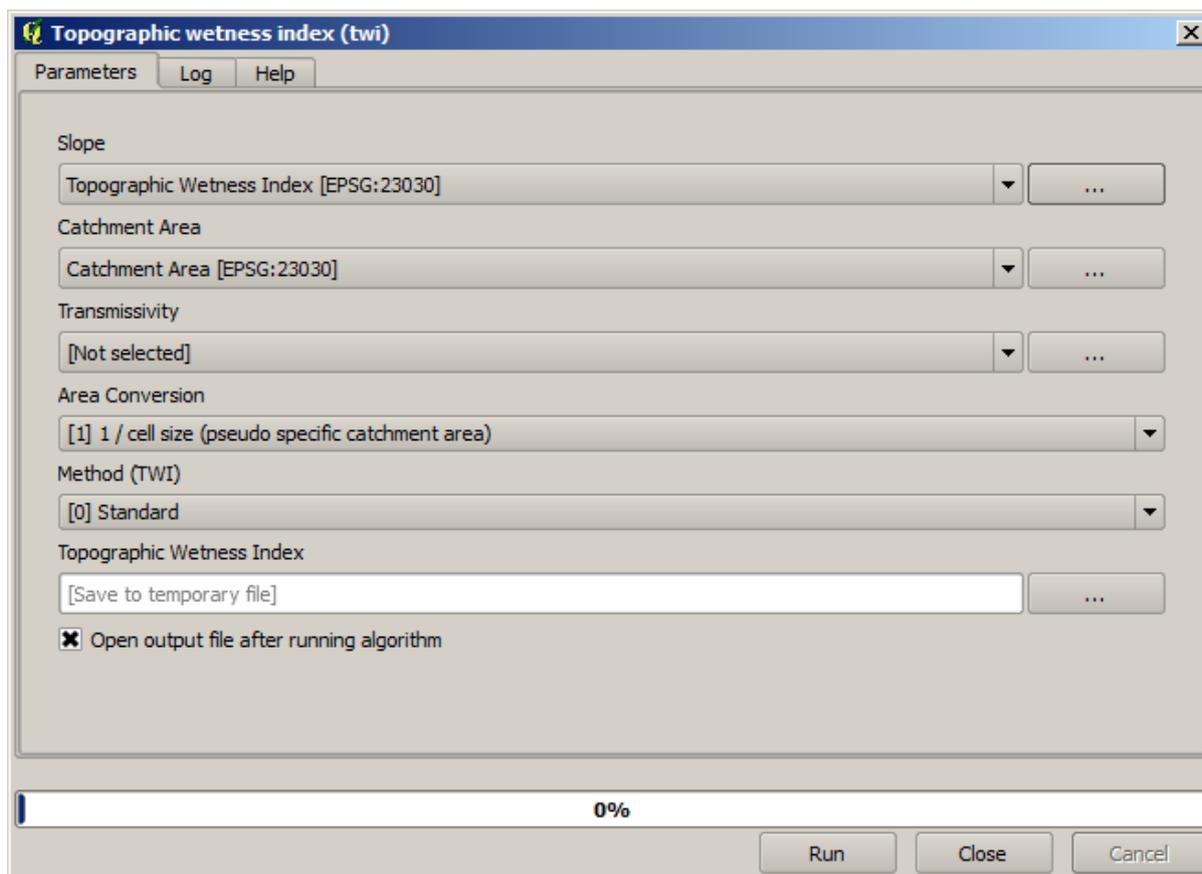
ここでは、2つの中間層を計算するために使用すべきパラメータのダイアログがあります。

ノート: 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。

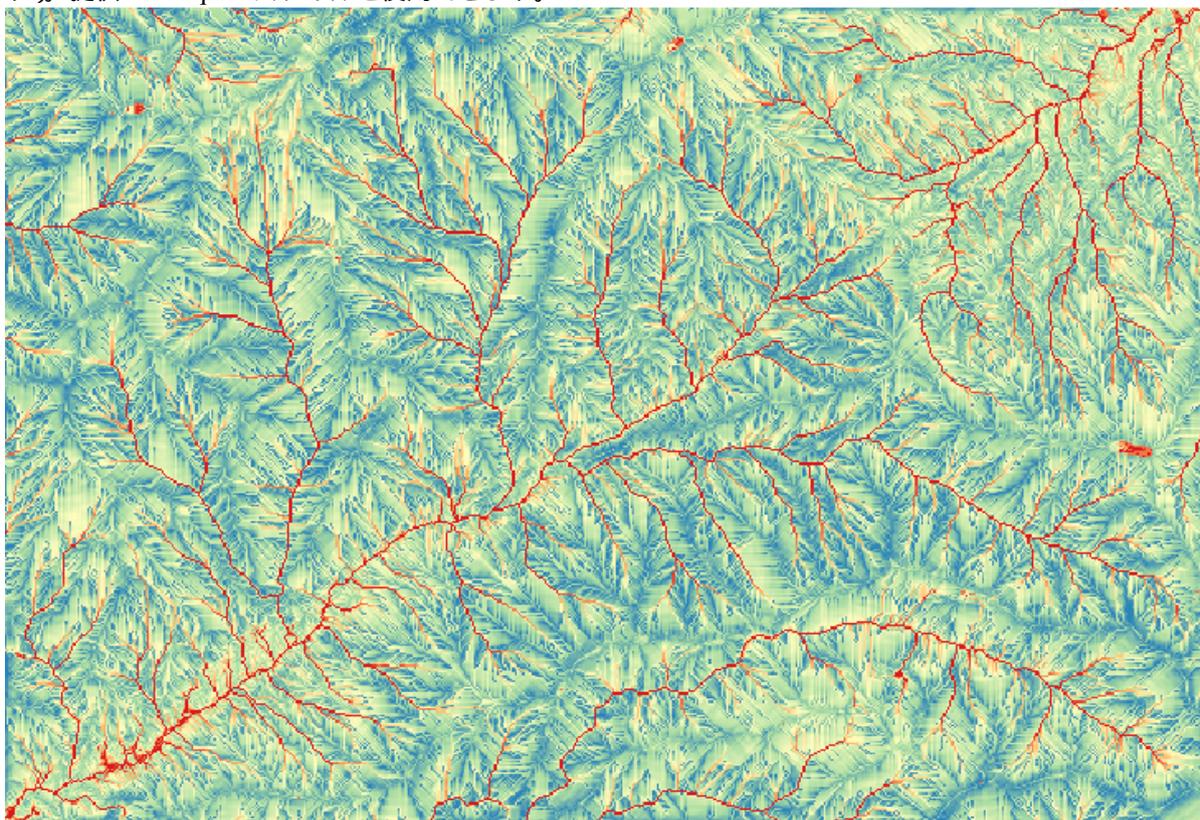




そして、これは [TWI アルゴリズムのパラメータ] ダイアログボックスを設定する必要があります。



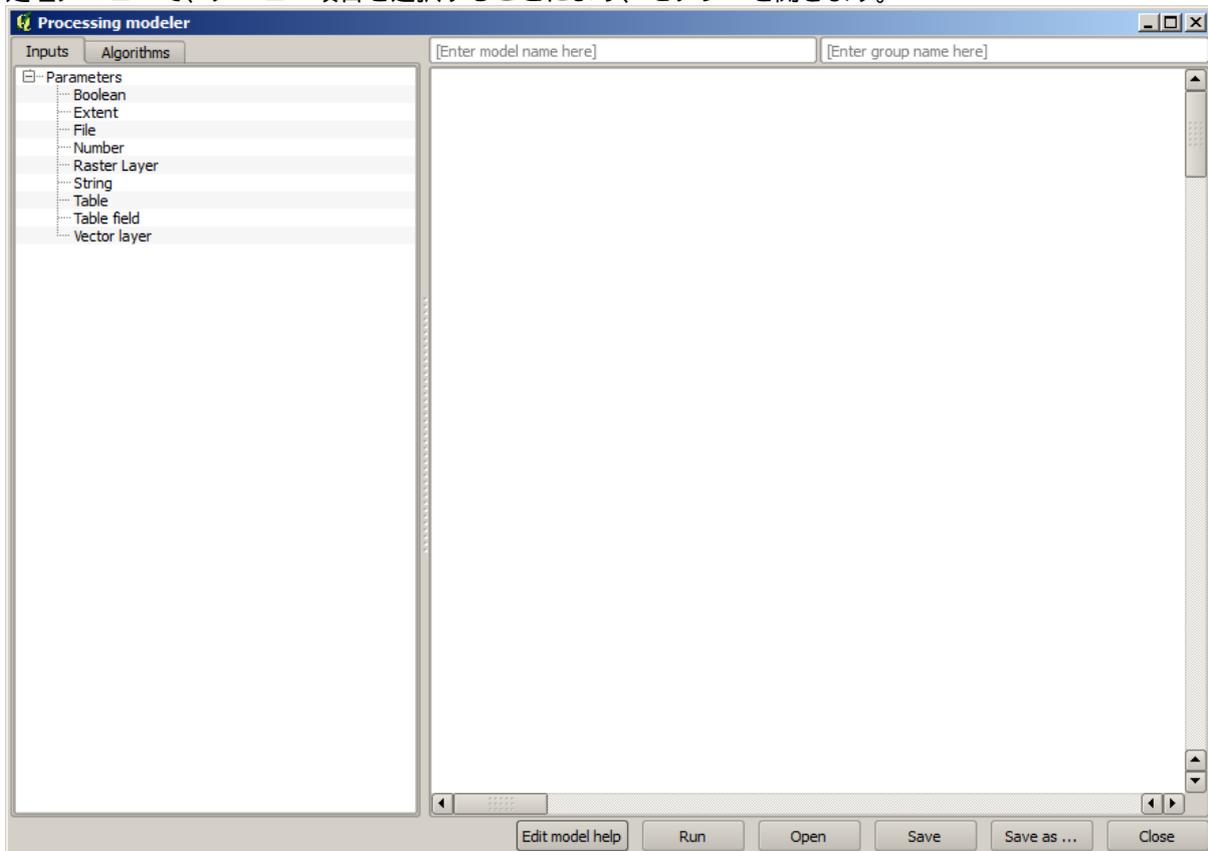
これが得られる結果です（デフォルトの singleband の疑似反転パレットがレンダリングに使用されています）。提供 “twi.qml” スタイルを使用できます。



今しようとしていることは、DEM から TWI をただ 1 つのステップで計算するアルゴリズムを作成することです。それは後に代わりに上記 3 つのものでそれを行うには、単に 1 つのステップが必要になるため、別

の DEM から TWI レイヤーを計算する必要がある場合で働くを保存します。必要とするすべてのプロセスは、ツールボックスに発見されたので、しなければならないことは、それらをラップするワークフローを定義することです。グラフィカルモデラーの出番です。

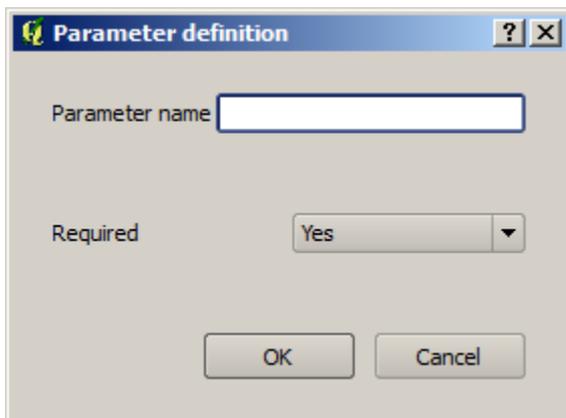
処理メニューで、メニュー項目を選択することにより、モデラーを開きます。



それが必要となる入力を設定し、それが含まれているアルゴリズムを定義する：二つのことは、モデルを作成するために必要とされます。両者を左に 2 つのタブから要素を追加することによって行われる - Modeler ウィンドウの右側：入力*と*アルゴリズム

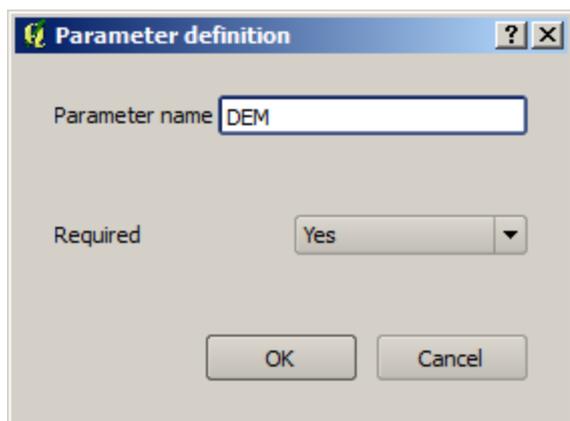
入力から始めましょう。このケースでは、追加することがあまりありません。私達はちょうど DEM をラスターレイヤーを必要とし、それは私たちの唯一の入力データとなります。

ラスターレイヤー 入力をダブルクリックし、次のダイアログが表示されます。

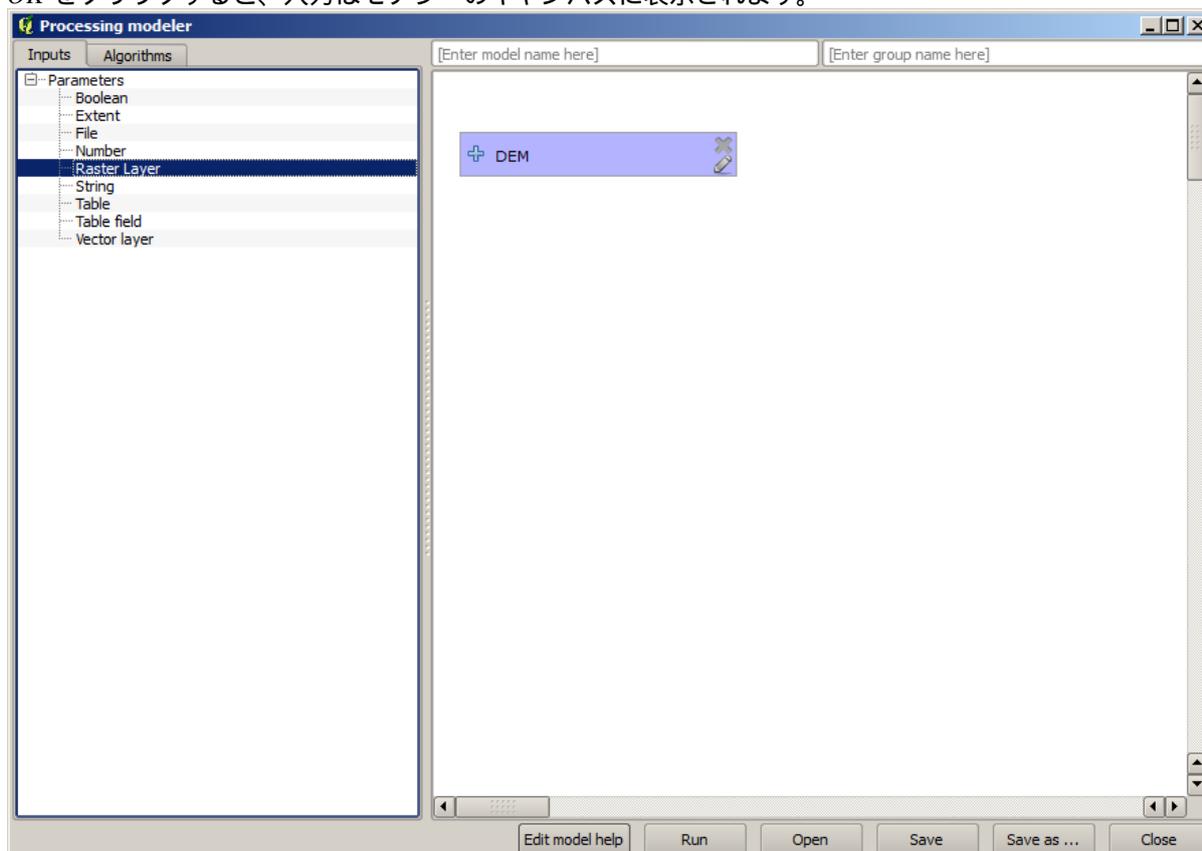


ここでは、望むの入力を定義する必要があります。このラスターレイヤーは、DEM ことを期待しているので、DEM それを呼び出します。それはそれを実行しているときに、モデルのユーザーに表示される名前です。そのレイヤーが動作することを必要とするので、必須レイヤーとして定義します。

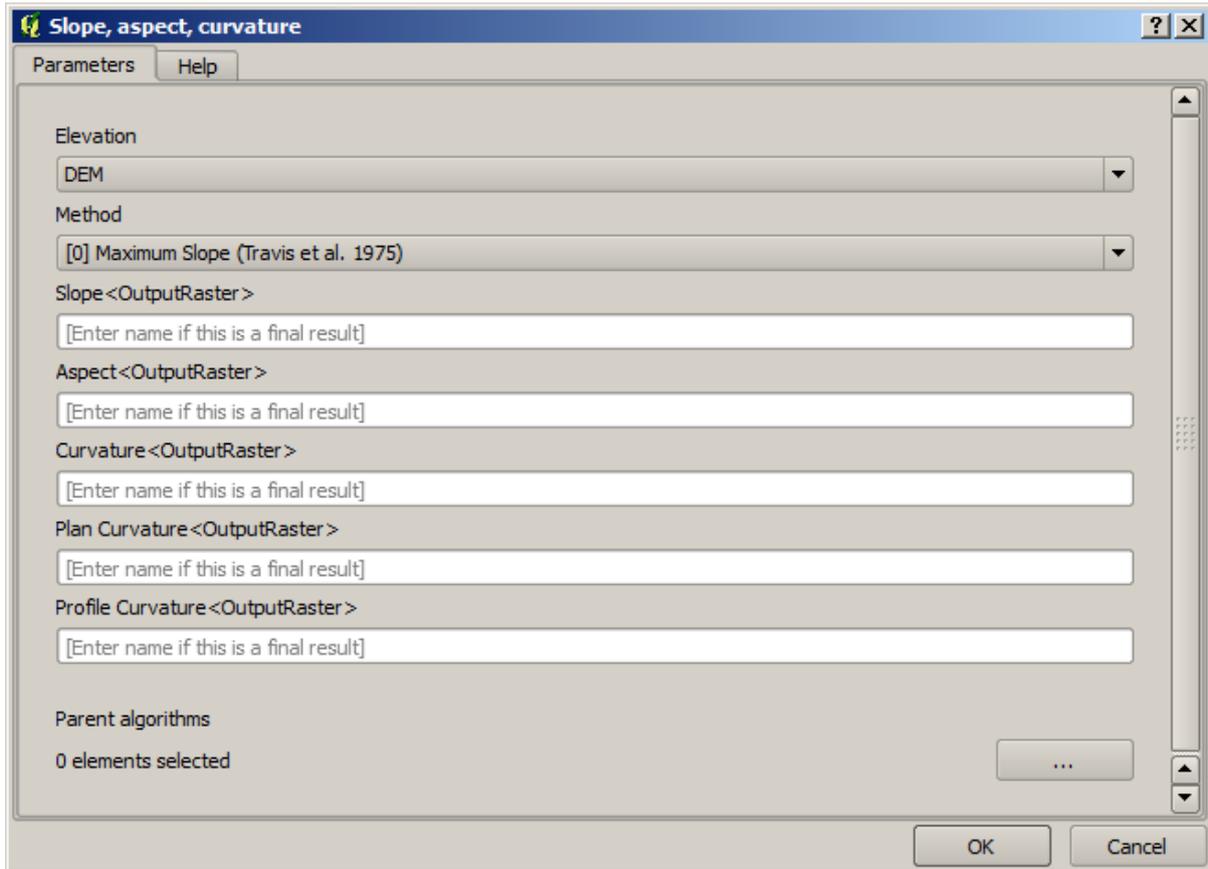
これがダイアログを設定する方法です



OK をクリックすると、入力はモデラーのキャンバスに表示されます。



それでは、[アルゴリズム] タブに移動してみましょう。我々は実行する必要がある最初のアルゴリズムは、*スロープ、側面、曲率*アルゴリズムです。ダブル、アルゴリズムのリストでそれを見つけて - それをクリックすると、以下のようなダイアログが表示されます。

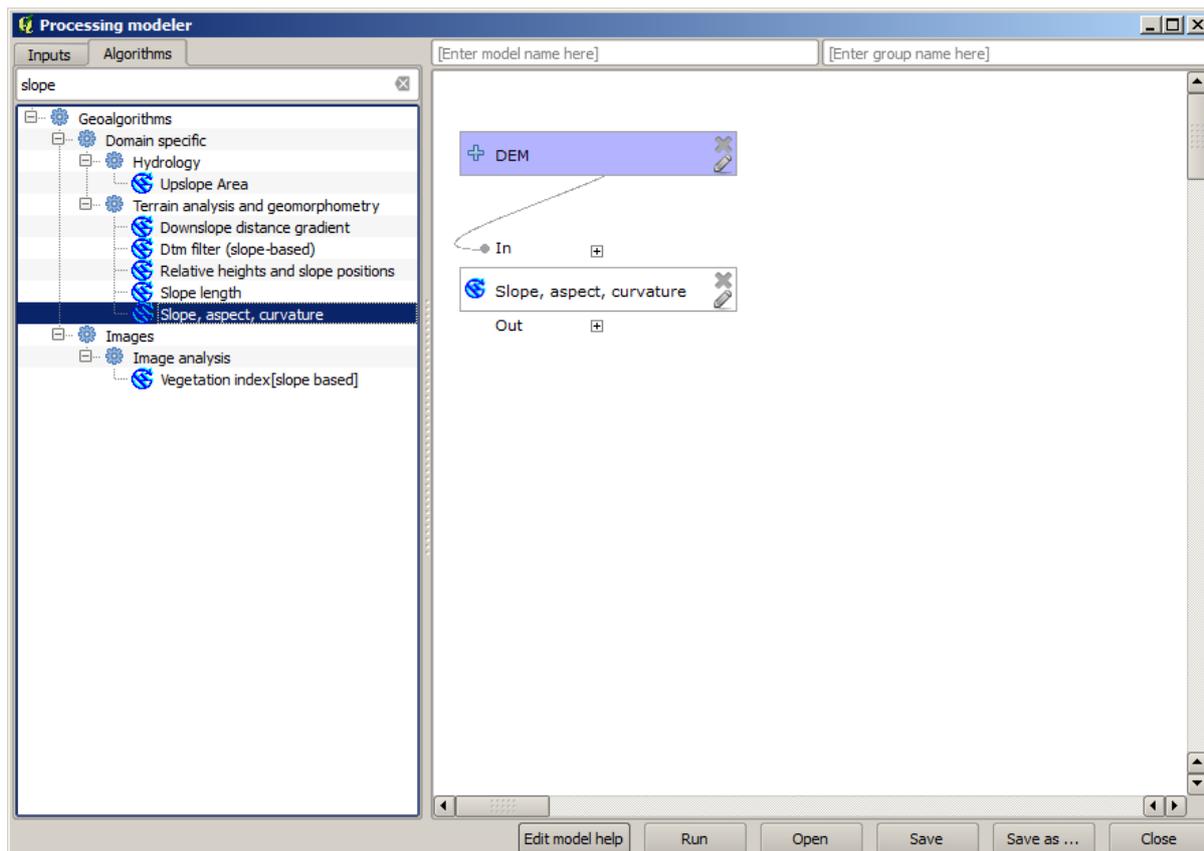


このダイアログは、ツールボックスから、アルゴリズムを実行するときに、見つけることができると非常によく似ていますが、パラメータ値として使用できる要素は、現在の QGIS プロジェクトから取られますが、モデル自体からされていません。すなわち、この場合には、すべての*標高*フィールドに利用可能なプロジェクトのラスターレイヤが、モデルで定義されたものだけを持っていない、ということの意味します。の昇格*パラメータに対応するリストに表示されますのみラスターレイヤーになります DEM の*という名前だけで 1 つのラスター入力し、追加したので。

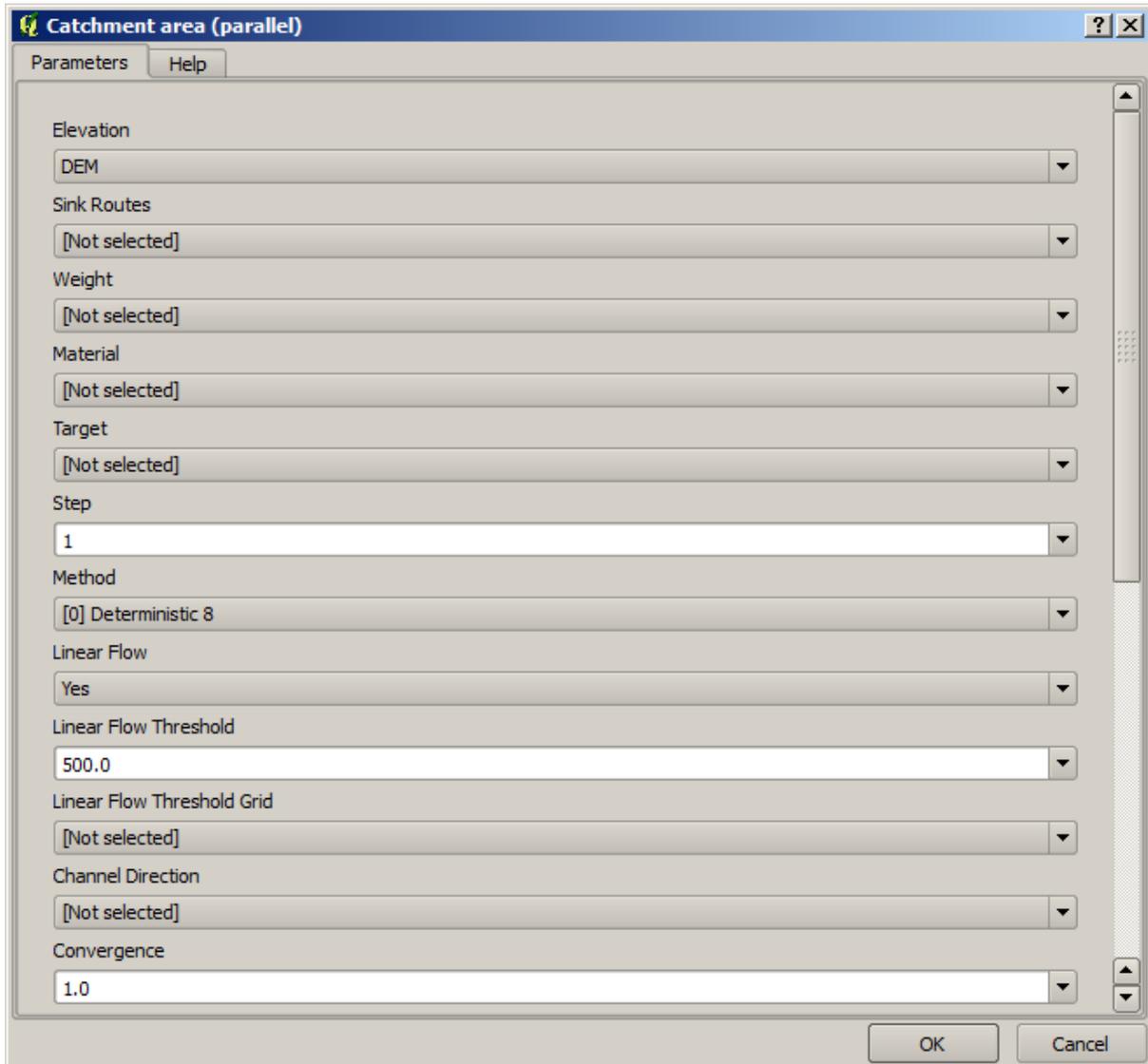
アルゴリズムは、モデルの一部として使用される場合、アルゴリズムによって生成される出力は、少々異なる方法で処理されます。代わりに、各出力を保存するファイルパスを選択するので、だけのその出力に含まは、中間レイヤー（およびモデルが実行された後、それを保存したくない）である、またはそれは、最終的なものであれば指定する必要があります。この場合には、このアルゴリズムによって生成される全てのレイヤーが中間です。それら（傾斜レイヤー）のいずれかを使用しますが、ちょうどそれが入手したい最終的な結果である TWI レイヤーを、計算する必要があるため、それを維持する必要はありません。

レイヤーは、最終的な結果ではない場合は、だけの対応するフィールドを去ら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ますパラメータ] ダイアログボックスでレイヤー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。

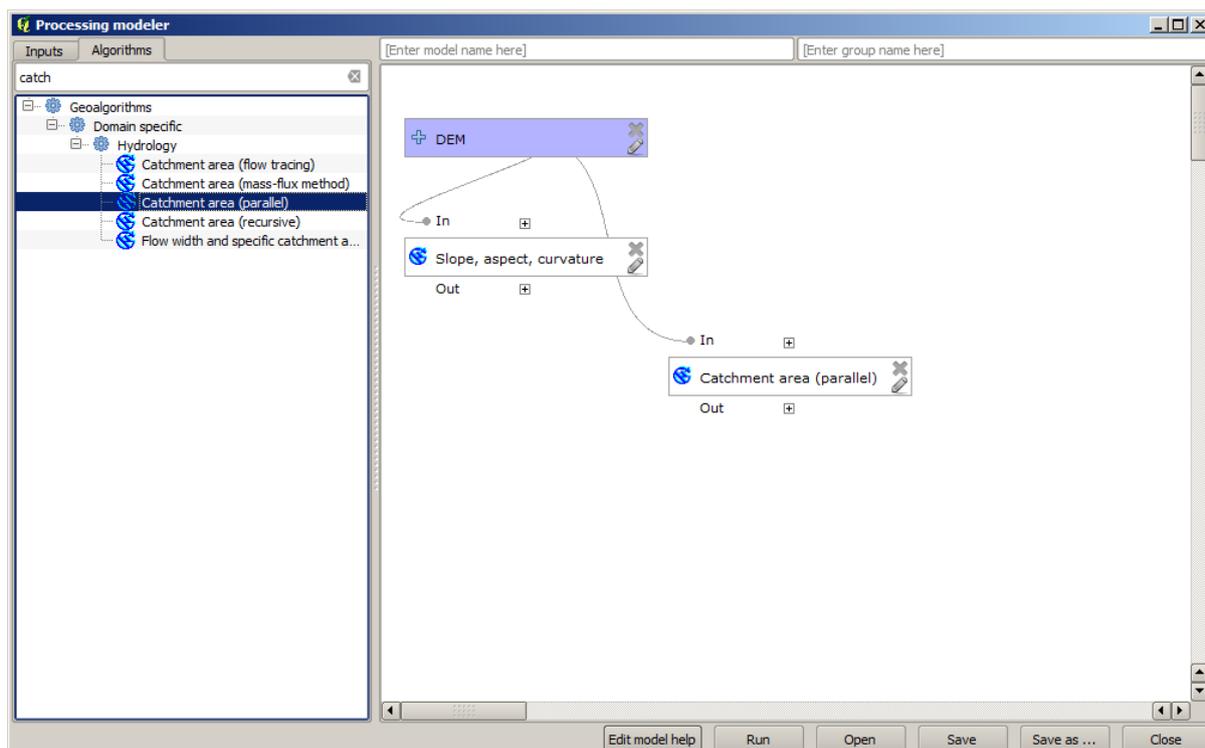
持っていないので、この最初のダイアログで選択はるかにちょうど 1 つにおけるレイヤーまたはモデル（私たちが作成した DEM の入力）ありません。OK 実際には、ダイアログのデフォルト設定では、この場合に正しいものであるので、ただ押す必要があります。これは、今モデラーキャンパスになりますものです。



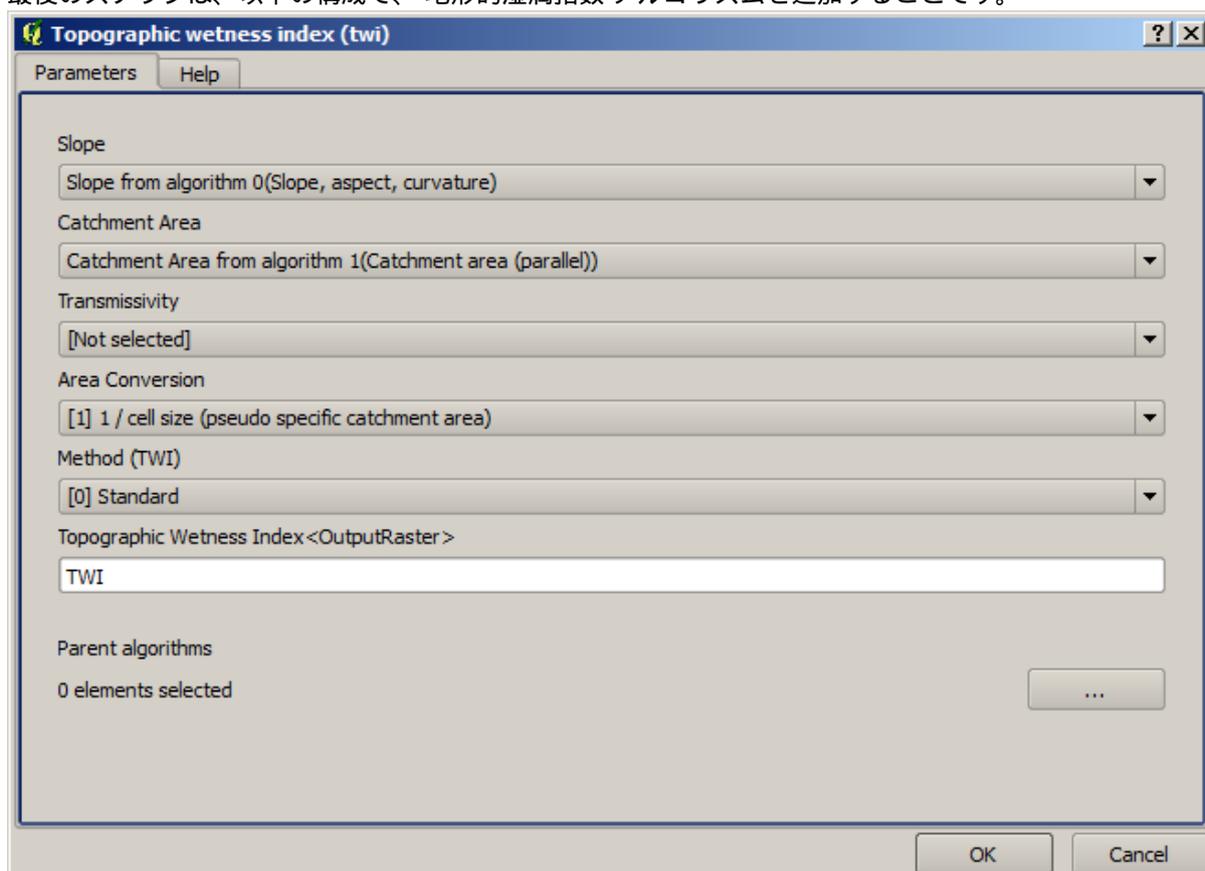
モデルに追加する必要がある第2のアルゴリズムは、集水域のアルゴリズムです。***流域面積 (Paralell)*** という名前のアルゴリズムを使用します。入力として再び DEM レイヤを使用し、それが生成 outputs のどれも最終的なものでないので、ここで対応するダイアログを埋めるために持っているかです。



今、モデルは、次のようになります。



最後のステップは、以下の構成で、地形的湿潤指数 アルゴリズムを追加することです。

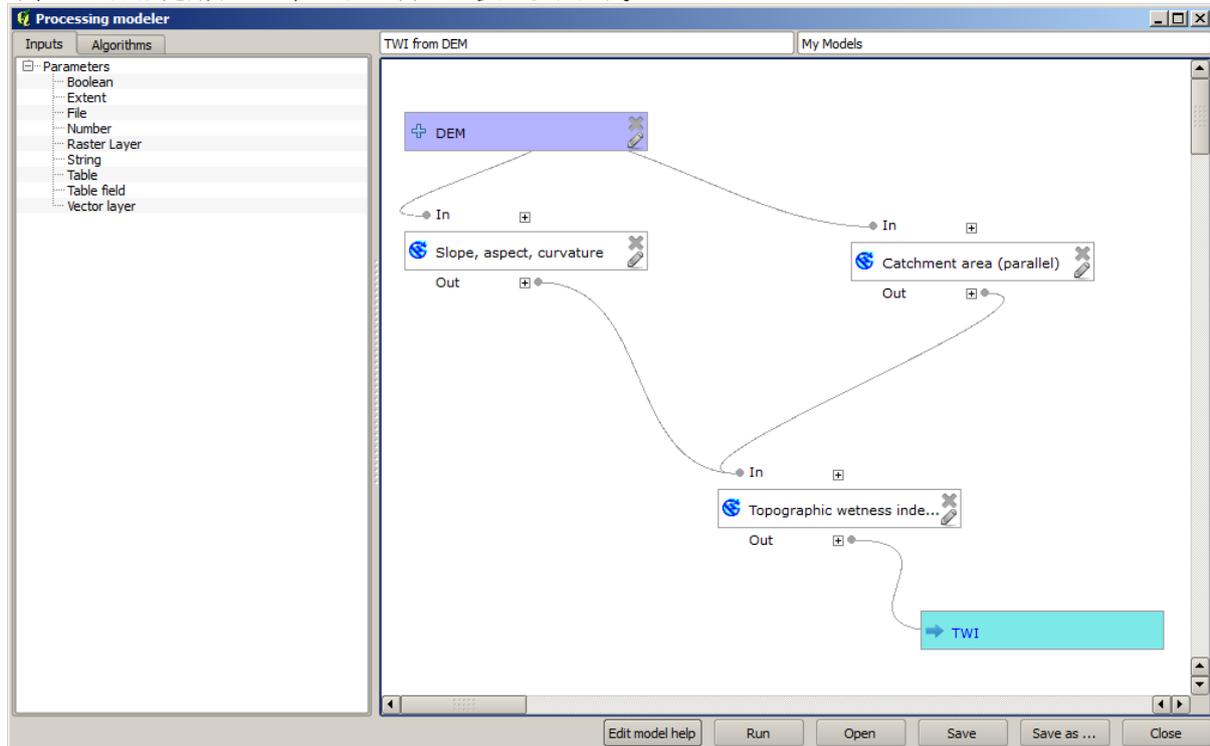


この場合、入力として DEM を使用しない、代わりに、以前に追加アルゴリズムによって計算される傾きと流域層を使用します。新しいアルゴリズムを追加すると、彼らが作り出す出力は、他のアルゴリズムのために利用可能になる、とワークフローの作成、アルゴリズムをリンクにそれらを使用します。

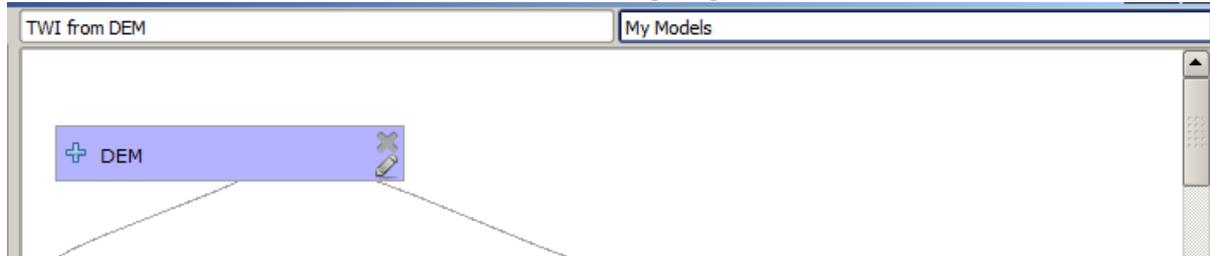
この場合、出力 TWI 層が最終層であるので、そう指示しなければなりません。対応するテキストボックス

では、この出力に示すことにし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。

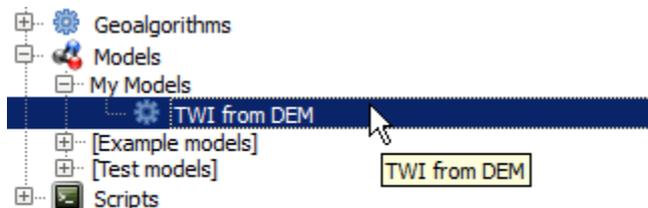


モデルウィンドウの上部に名前とグループ名を入力し、[保存] ボタンをクリックして保存します。

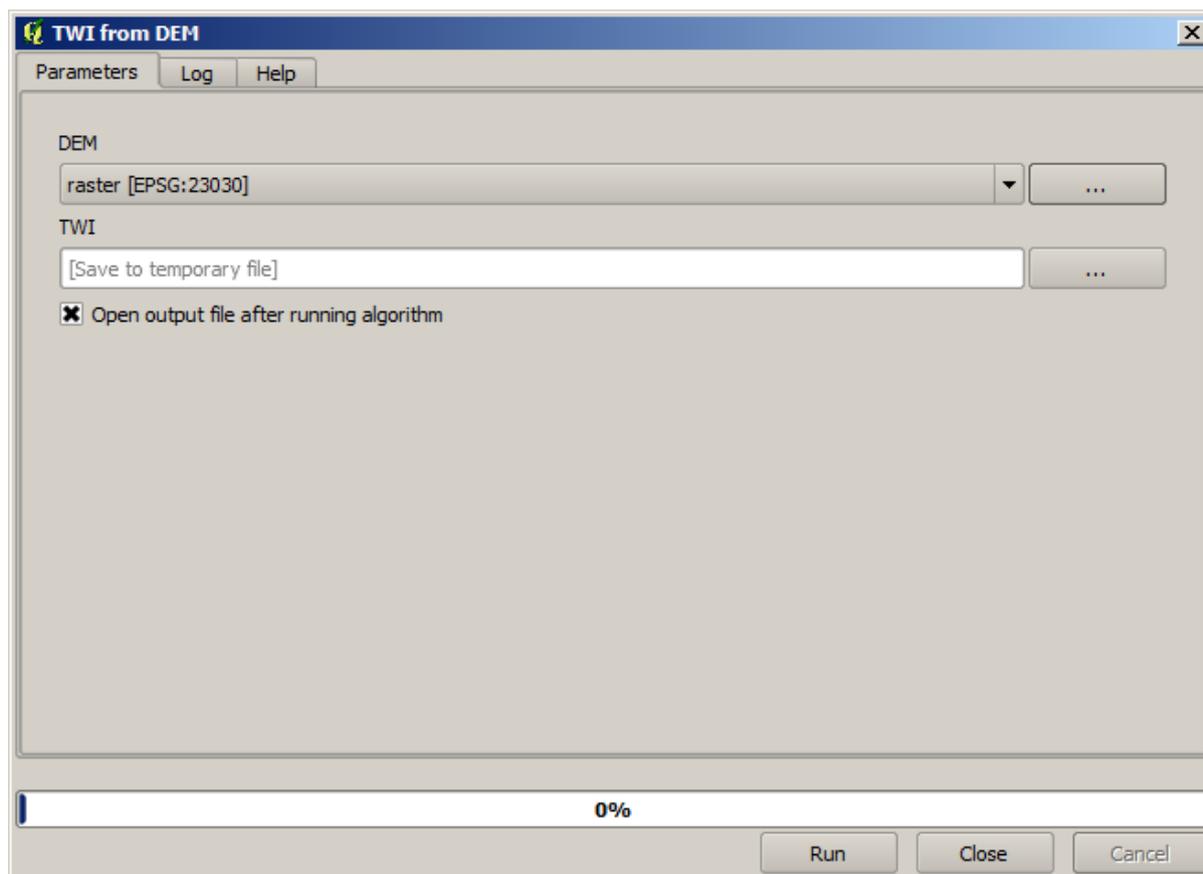


望む任意の場所にそれを保存し、後でそれを開くが、(ファイル保存ダイアログが表示されたときに表示されるフォルダである)モデルのフォルダに保存した場合、モデルも同様にツールボックスで利用できるようになります。することができます だから、そのフォルダに滞在し、好むファイル名でモデルを保存します。

今モデラーダイアログを閉じ、ツールボックスにアクセスしてください。*モデル*エントリでは、モデルを見つけるでしょう。



それをダブルクリック、ただの通常のアルゴリズムのようにそれを実行できます。



パラメータダイアログを見ることができるよう、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と共に、あなたがモデルに追加の入力を含みます。

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 層を取得します。

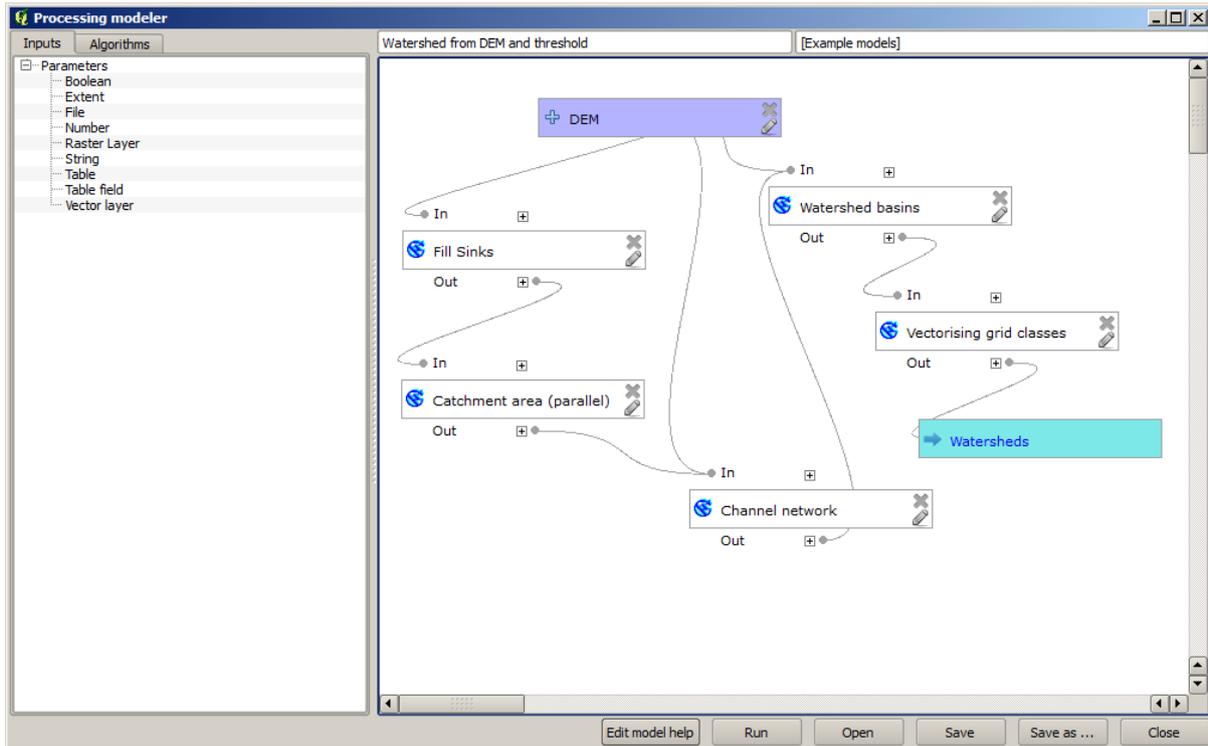
17.18 より複雑なモデル

ノート: このレッスンでは、グラフィカルなモデラーで、より複雑なモデルで動作します。

私たちは前の章で作成した最初のモデルは、ただ1つの入力と3つのアルゴリズムで、非常にシンプルなものでした。より複雑なモデルは、異なる入力の種類と、より多くのステップを含むと、作成することができます。この章のために我々は DEM としきい値に基づいて、流域のベクトル層を作成したモデルで動作します。すなわち、各シングルステップを毎回繰り返すことなく、異なる閾値に対応するいくつかのベクターレイヤを計算するために非常に有用であろう。

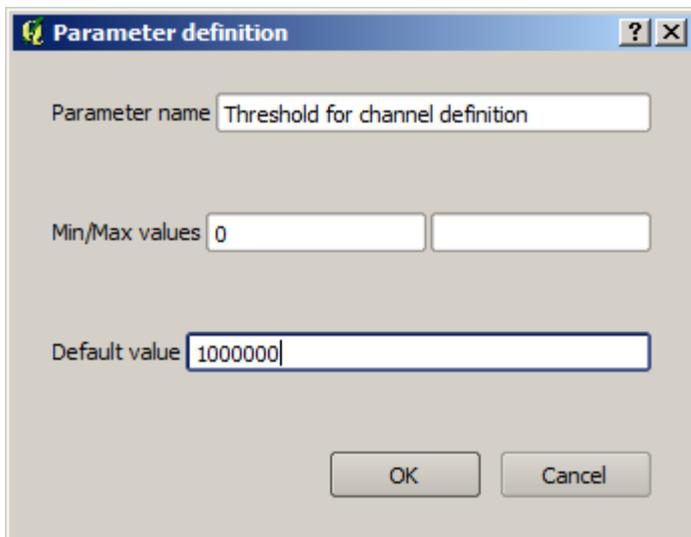
このレッスンには、モデルを作成する方法は含まれていません。既に必要な手順はおわかりでしょうし（私たちは前のレッスンでそれらを見て）、モデラーに関する基本的な考え方はすでに見てきましたので、それは御自分で試してみてください。モデルを作成しようと数分を過ごし、間違えることは心配しないでください。注意：最初に入力を追加し、それからワークフローを作成するためにそれらを使用するアルゴリズムを追加します。

完全なモデルをご自分で作成できなかった場合は、いくつかの余分な助けを必要と、このレッスンに対応するデータフォルダーは、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ります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。

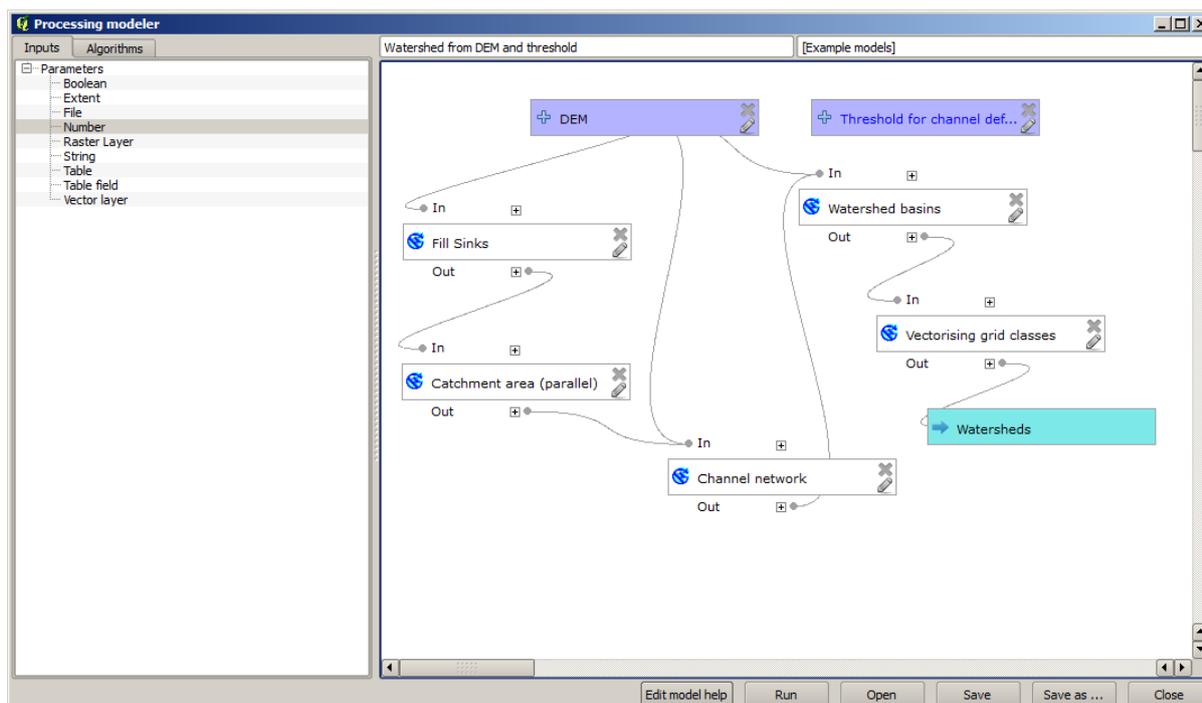


このモデルは、計算を完了するために必要なすべてのステップが含まれていますが、それはただ一つの入力があります：DEM を。すなわち、チャンネル定義のしきい値はモデルではないことがあり得るとして有用になり、固定値を使用することを意味します。私たちはモデルを編集できますので、それは問題ではありません、そしてそれはまさにこれからするであろうことです。

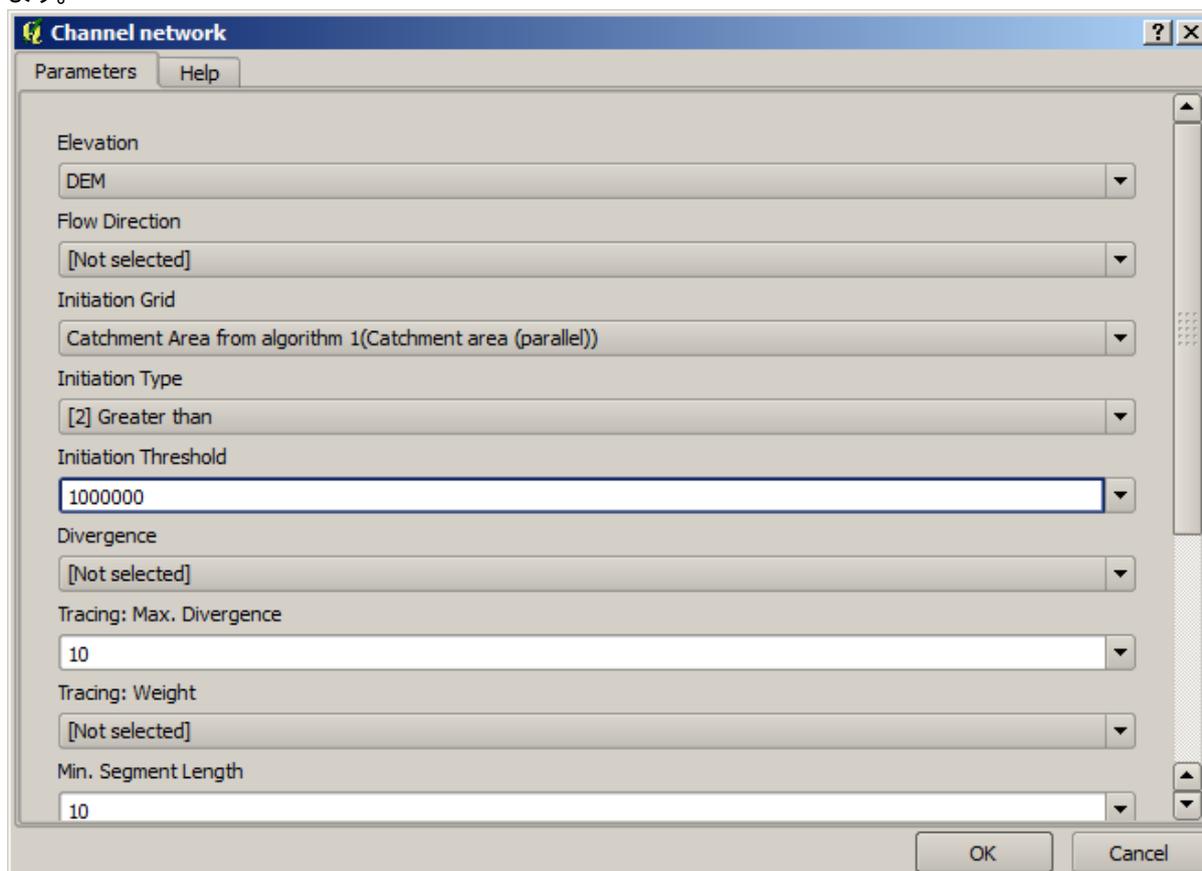
まずは、数値の入力を追加してみましょう。すなわち、このような値がモデルに含まれたアルゴリズムのいずれかで必要とされるときに使用でき、数値の入力をユーザーに要求します。***番号***入力ツリー内のエントリをクリックすると、対応するダイアログが表示されます。次のような値でそれを埋めます。



今、あなたのモデルは、次のようになります

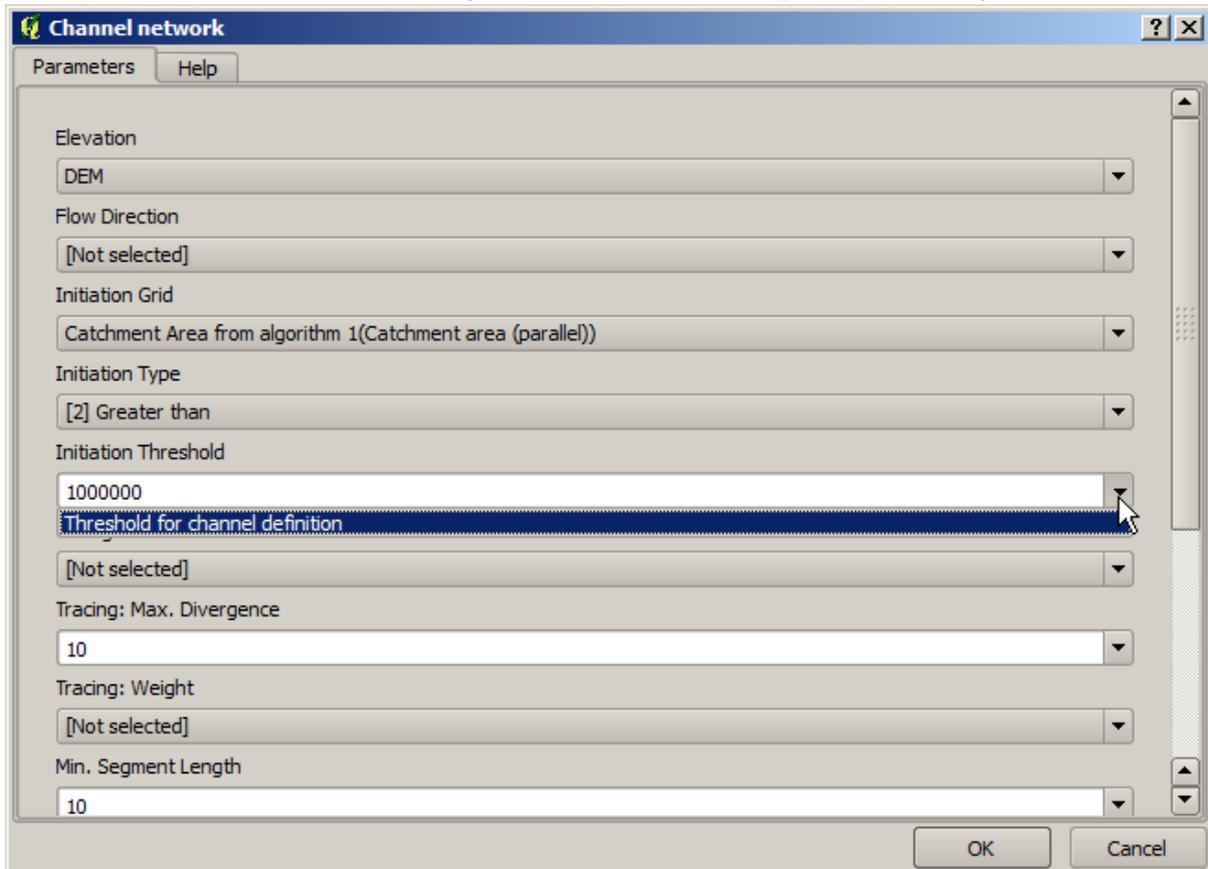


追加した入力を使用されていないので、このモデルは、実際には変更されていません。このような場合には、それを使用するアルゴリズムに*水路ネットワーク* 1 をその入力をリンクする必要があります。すでにモデラーに存在するアルゴリズムを編集するには、単にキャンバスで対応するボックスの上にペンのアイコンをクリックしてください。水路ネットワーク アルゴリズムをクリックすると、次のように表示されます。



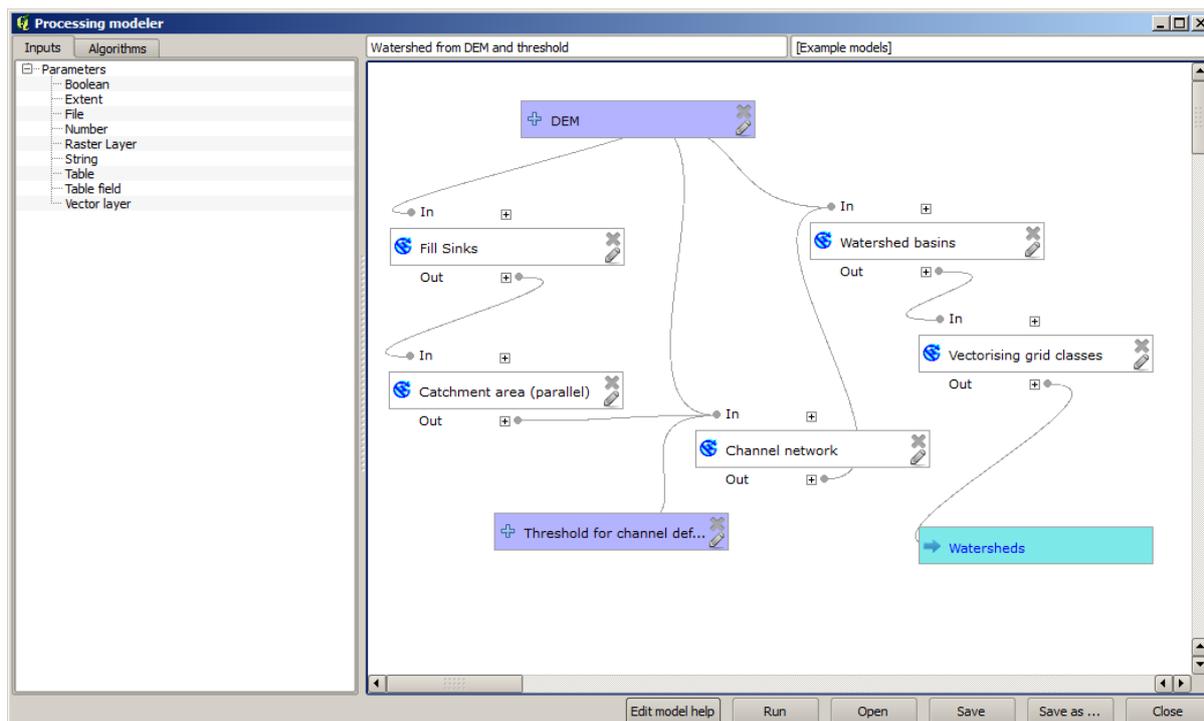
ダイアログは、アルゴリズムによって使用される現在の値で満たされています。閾値パラメータは、1,000,000の固定値を持っていることがわかります（これはアルゴリズムのデフォルト値でもあるが、他の値がそ

に置くことができる)。ただし、パラメータは、共通のテキストボックスに入力されていないことがわかりますが、オプションメニューであります。それを展開すると次のように表示されます。

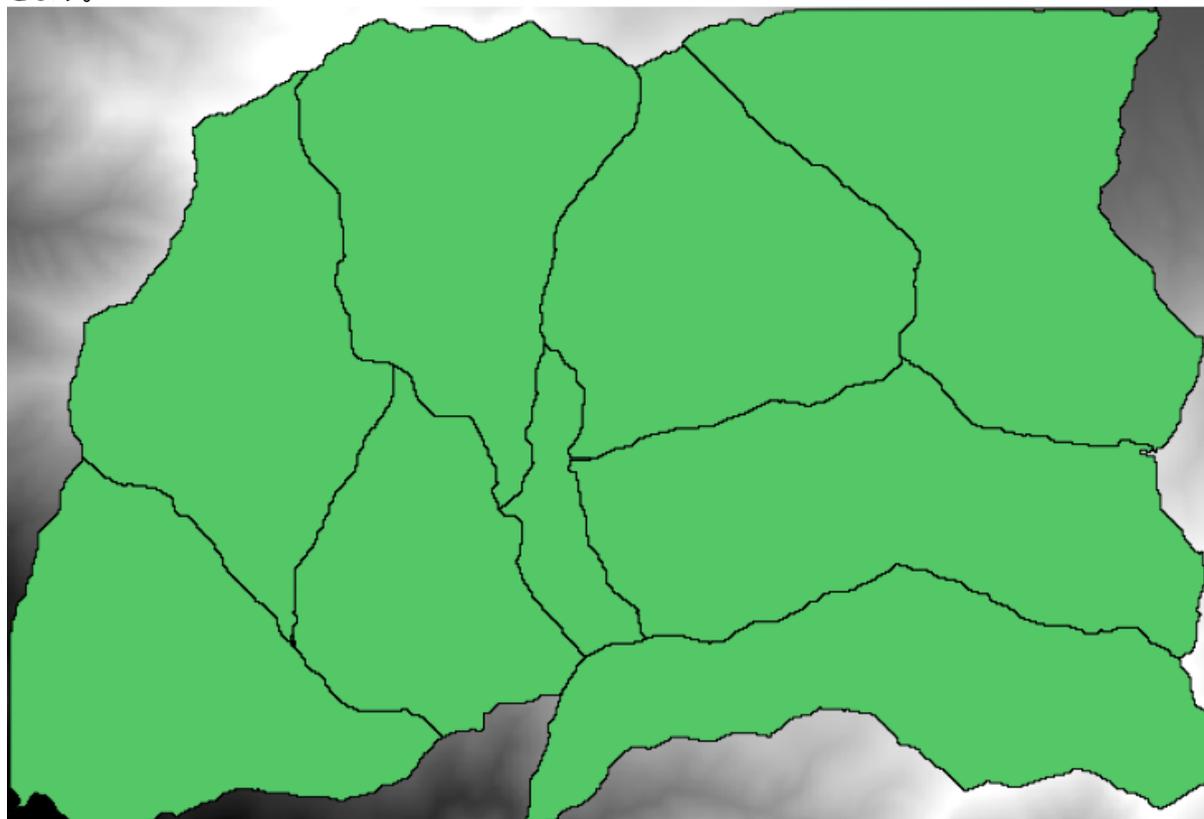


追加した入力があるところにあり、それを選択できます。モデルでアルゴリズムが数値を必要とするたびに、それをハードコーディングしそれを直接入力することも、使用可能な入力と値のいずれかを使用できます（いくつかのアルゴリズムは単一の数値を生成することを思い出してください。これについての詳細はすぐに見るでしょう）。文字列パラメータの場合、また、文字列の入力が表示され、それらのいずれかを選択、または所望の固定値を入力できるようになります。

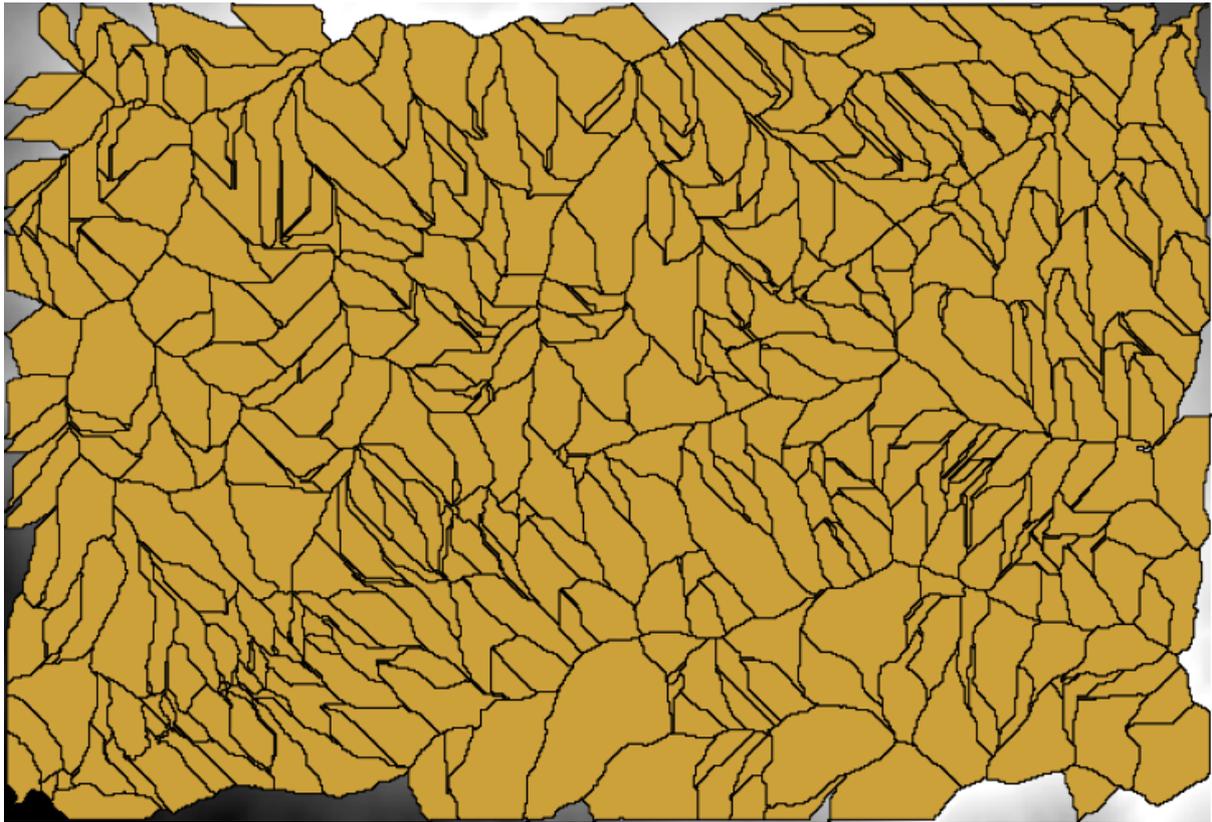
しきい値 パラメータで*しきい値* 入力を選択し、*OK* をクリックしてモデルへの変更を適用してください。今モデルのデザインは、次のようになります。



モデルが完成しました。私たちは前のレッスンで使用している DEM を使用して、それを実行しようと、異なるしきい値を持ちます。ここでは、異なる値に対して得られた結果のサンプルを持っています。あなたは、私たちは、水文解析のレッスンで得られたものですデフォルト値、のための結果と比較することができます。



しきい値= 100,000



しきい値= 1,000,000

17.19 モデラーでの数値計算

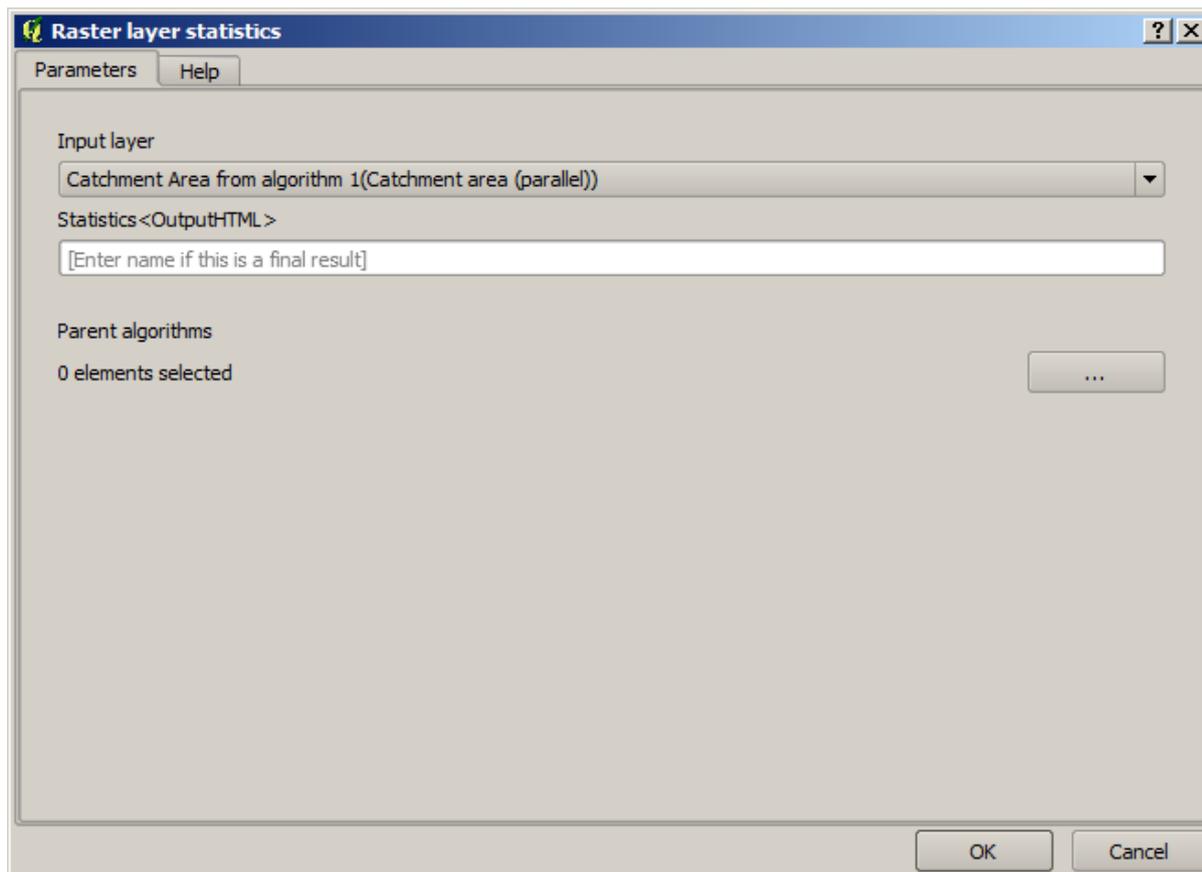
警告: すべての問題を報告してください、この章では、十分にテストされていない、注意してください。画像は表示されません

ノート: 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ようとしているので、有効なしきい値の計算を自動化できますし、ユーザーに依頼する必要はありません。それを入力してください。その値が閾値ラスタレイヤ内の変数を指しているため、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。

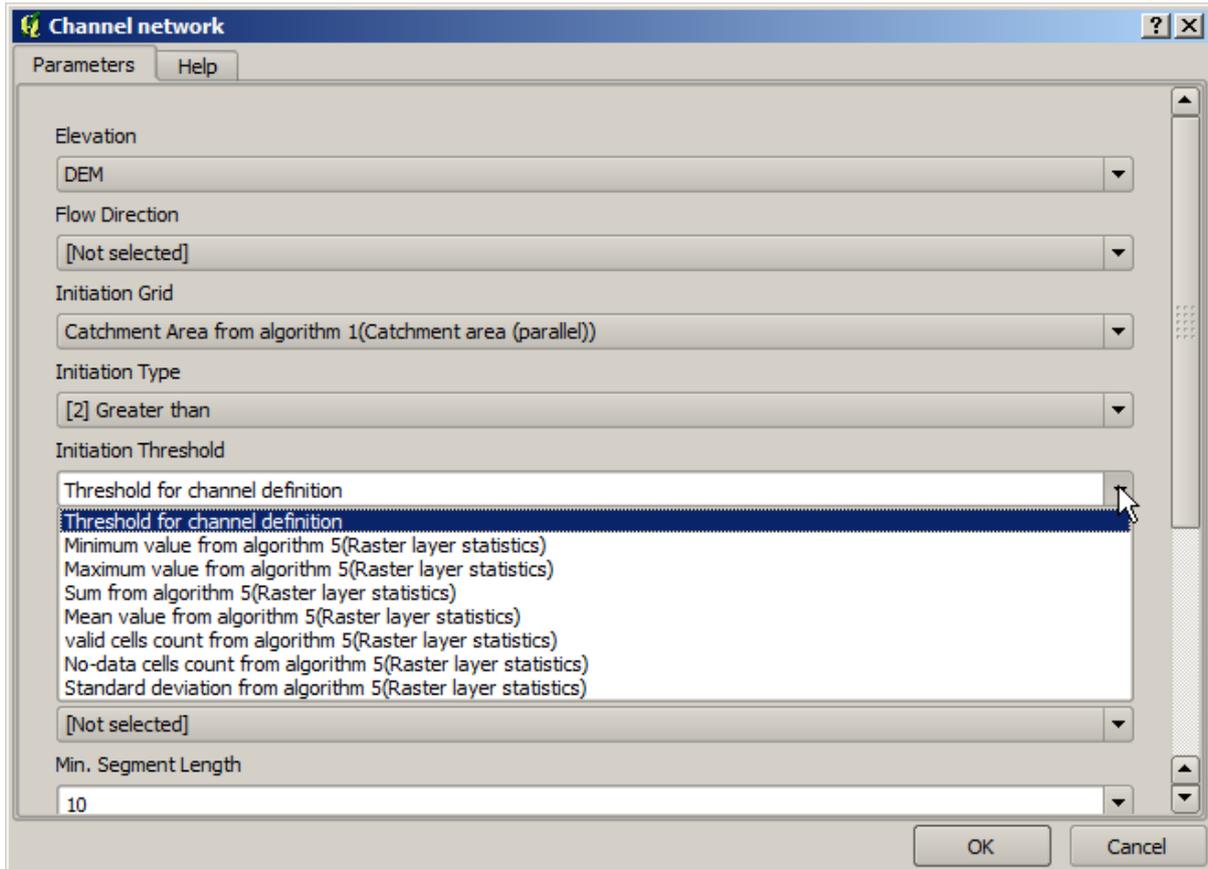
前述のモデルから開始し、以下の修正を行いましょう。

まず、ラスタレイヤ統計 アルゴリズムを使用してフロー流量蓄積レイヤの統計情報を計算します。



これは、今、他のアルゴリズムのすべての数値フィールドのために利用可能になり、統計値のセットを生成します。

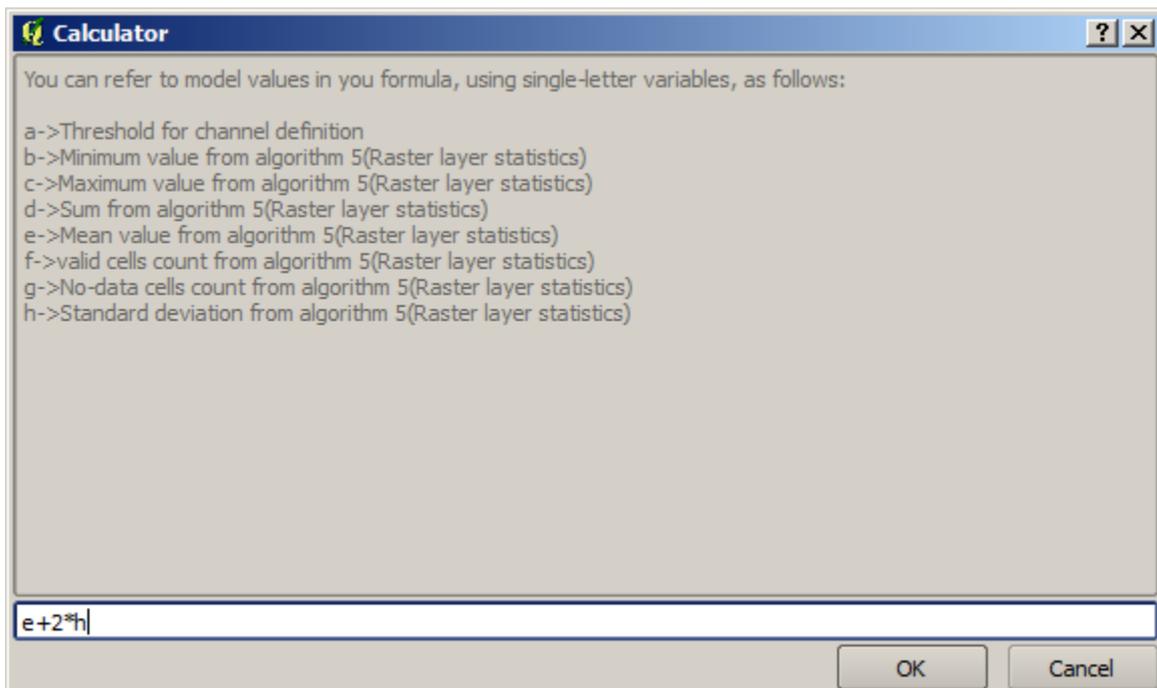
水路ネットワーク アルゴリズムを編集する場合、最後のレッスンで行ったように、離れて、追加した数値の入力から他のオプションを持っていることを、今表示されます。



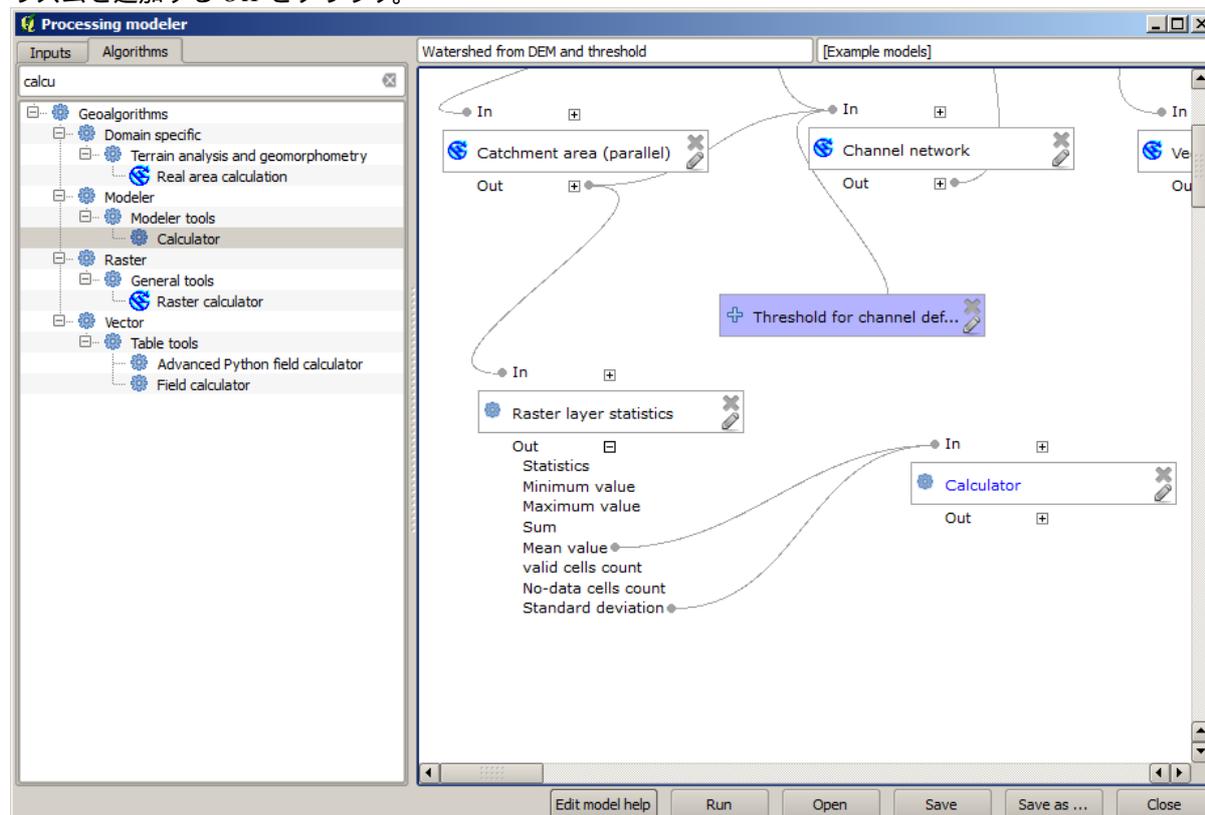
それらは非常に現実的ではない水路ネットワークになりますので、この値はいずれも、有効なしきい値として使用するのに適していません。代わりに、より良い結果を得るために、それらに基づいて、いくつかの新しいパラメータを導出できます。例えば、平均+標準偏差の2倍を使用できます。

その演算処理を追加するには、グループ Geoalgorithms /モデラー/モデラー・ツールで見つける電卓を使用できます。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。

電卓アルゴリズムのパラメータ] ダイアログボックスは次のようになります。

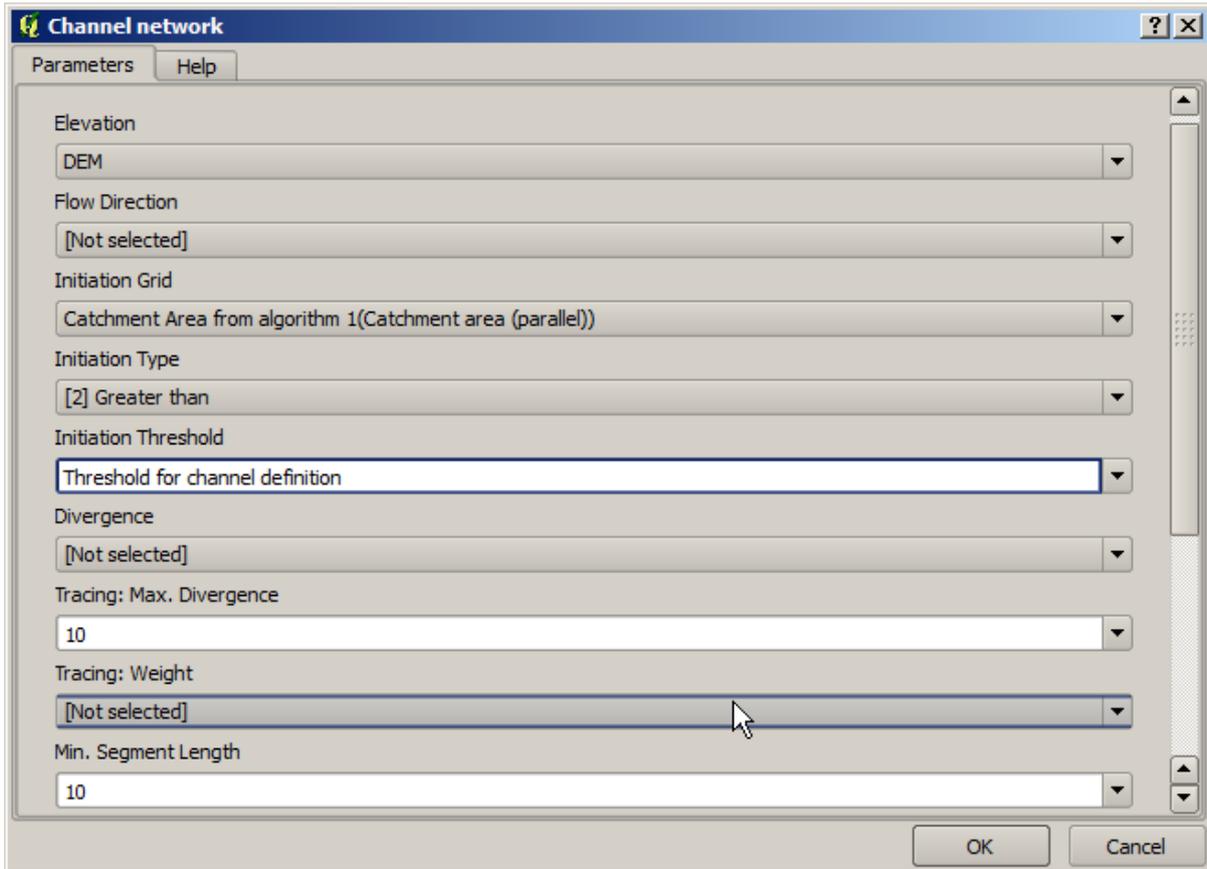


ご覧になれるように、ダイアログは、見てきた他のものと異なっているが、そこに水路ネットワーク アルゴリズムに しきい値 フィールドで使用可能だった同じ変数を持っています。上記の式を入力して、アルゴリズムを追加する *OK* をクリック。

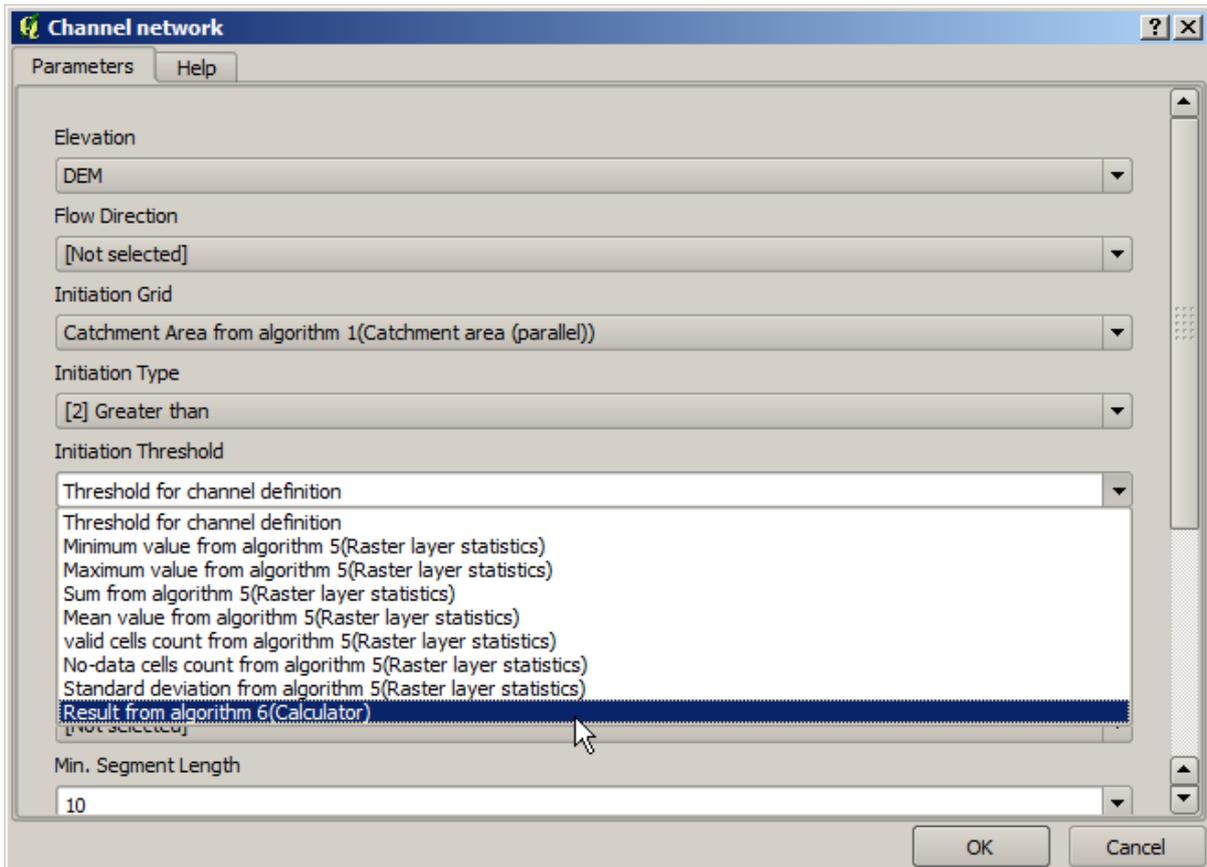


上記のように出力項目を展開すると、モデルが2つの値、つまり平均と標準偏差に接続されていることがわかります。これは、数式で使用した値です。

この新しいアルゴリズムを追加すると、新しい数値を追加します。チャンネルネットワーク アルゴリズムに再度行く場合、今しきい値 パラメータで、その値を選択できます。



OK をクリックすると、モデルは次のようになります。



モデルに追加数値入力を使用していないので、それを除去できます。それを右 - クリックして 削除を選択

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、いかなる問題を見つけたら画像とともに報告ください。

ノート: 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、単一の大きい方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。

現在のモデルに追加し、その入力として入力 DEM を使用しています。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出力のみがまだ流域のベクトルレイヤーになります。

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 TWI の値を含む新しいものを生成するために、流域ベクトルレイヤーと一緒に使用できる TWI レイヤーを持っています。

この計算は、ポリゴンでのグリッドの統計情報 アルゴリズムで使用して行われます。最終的な結果を作成するために、入力として上記のレイヤーを使用します。

警告: TODO : 画像を追加します。

*アルゴリズム*ベクタライズグリッドクラスの出力は、もともと私たちの最終的な出力だったが、今はちょうど中間結果としてそれをしたいです。それを変更するには、我々はアルゴリズムを編集する必要があります。ただ、ダブル - そのパラメータのダイアログが表示され、出力の名前を削除するには、それをクリックしてください。それがデフォルトであるようつまり、一時的な出力になります。

警告: TODO : 画像を追加します。

これは、最終的なモデルは次のようになります方法です。

警告: TODO : 画像を追加します。

ご覧のとおり、別のモデルでモデルを使用することは特別なものではありません、別のアルゴリズムを追加するだけのようにいる限り、モデルがモデルフォルダに保存され、ツールボックスで提供されているとして、それを追加できます。

17.21 データ補間

ノート: 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トレイヤを得ます。それをする前には、若干のデータ準備をしなければならず、そして補間後に結果として生じるレイヤを修正する追加の処理をいくつか加えますから、完全な分析ルーチンになるでしょう。

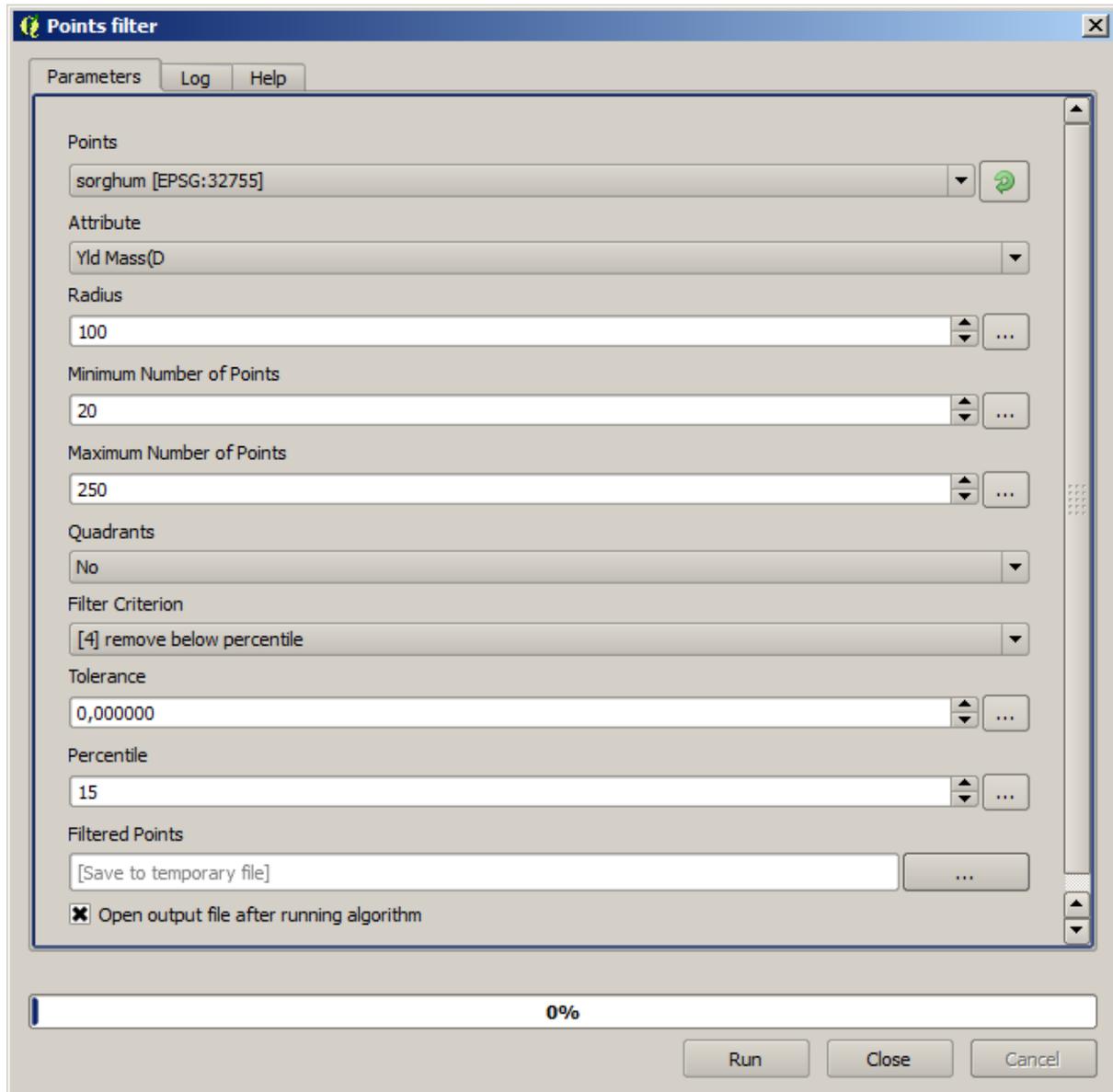
このレッスンの例となるデータを開くと、このように見えます。



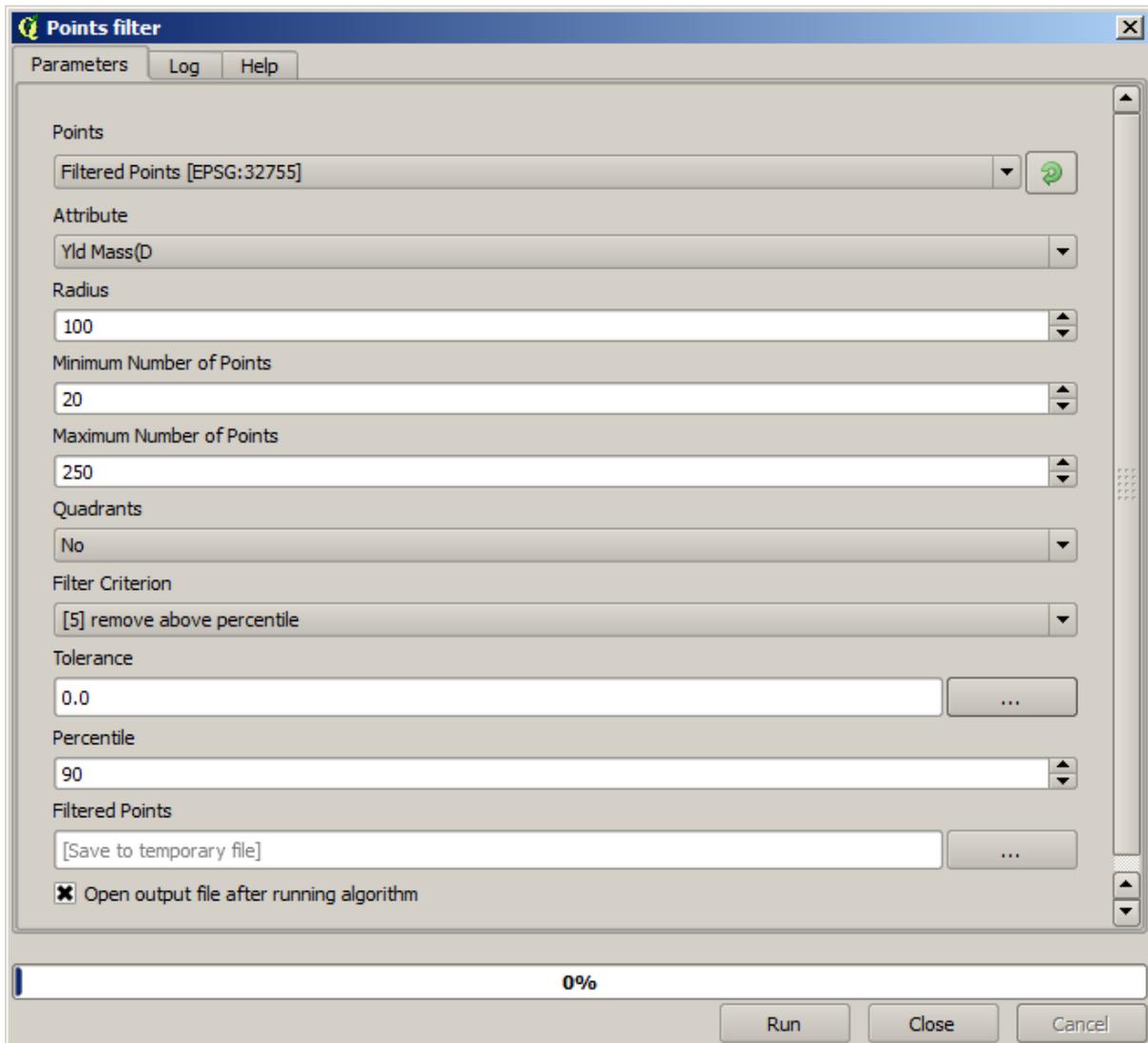
現代の収穫者によって生み出されるように、データは収穫産出高データと一致します。そして、我々は収穫産出高のラストレイヤを得るためにそれを使います。我々は、そのレイヤで更なる分析もしますが、ちょうど最も生産的な地域を簡単に特定するための背景レイヤ、更には生産性が改善されることができるとして使うつもりはありません。

最初にやることは、ポイントが冗長なポイントを含むため、レイヤをクリーンアップすることです。それが何らかの理由で変更しなければならない、または変更する場所では、これらは収穫者の動きに起因します。**Points filter**アルゴリズムは、これに役立ちます。二回それを使い、分布の上位または下位の部分の両方で外れ値とみなされるポイントを除外します。

最初の実行のため、次のパラメータ値を使います。



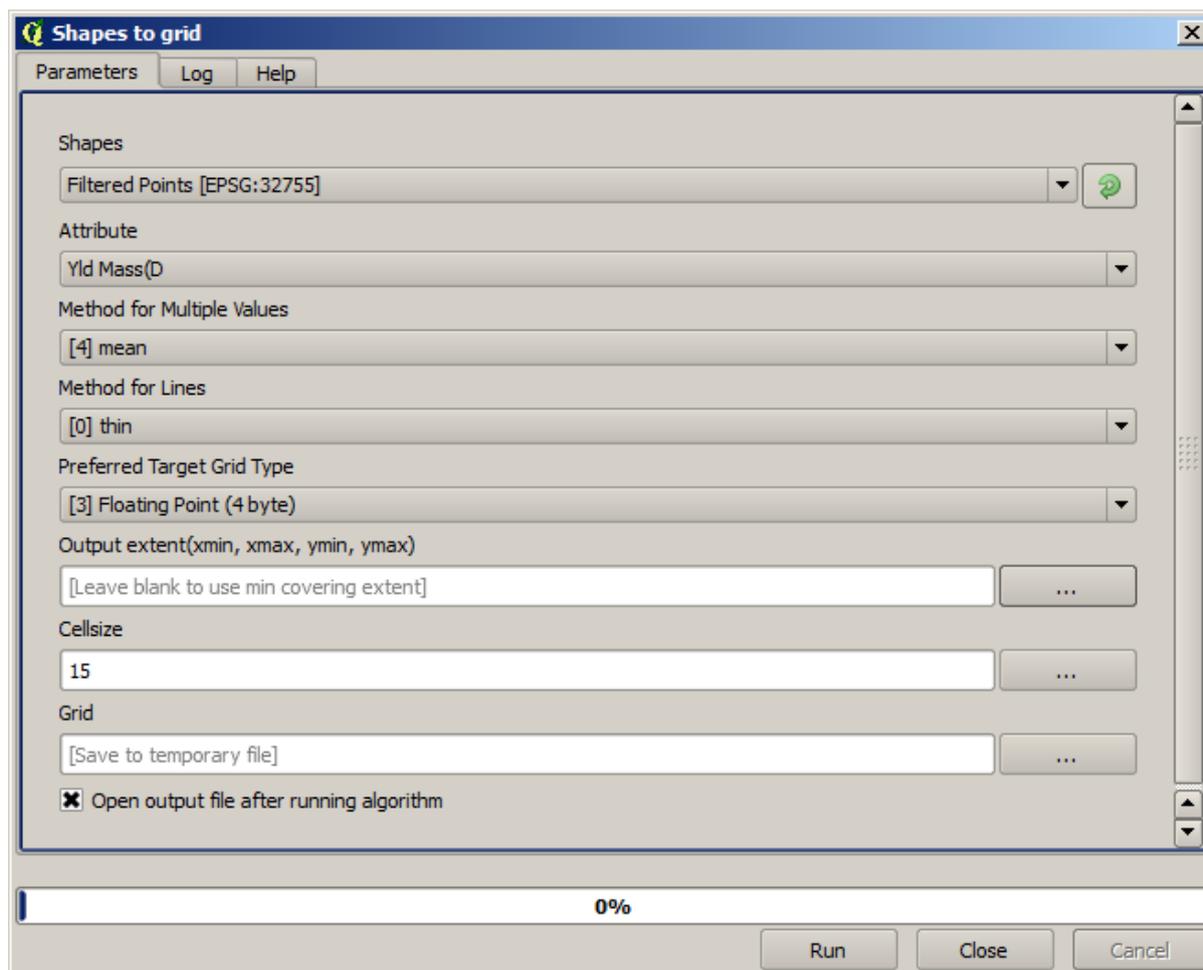
次の手順のため、以下に示す設定を使用します。



我々がオリジナルレイヤを入力して使っているが、それよりも前に実行した際の出力以外である点に注意してください。

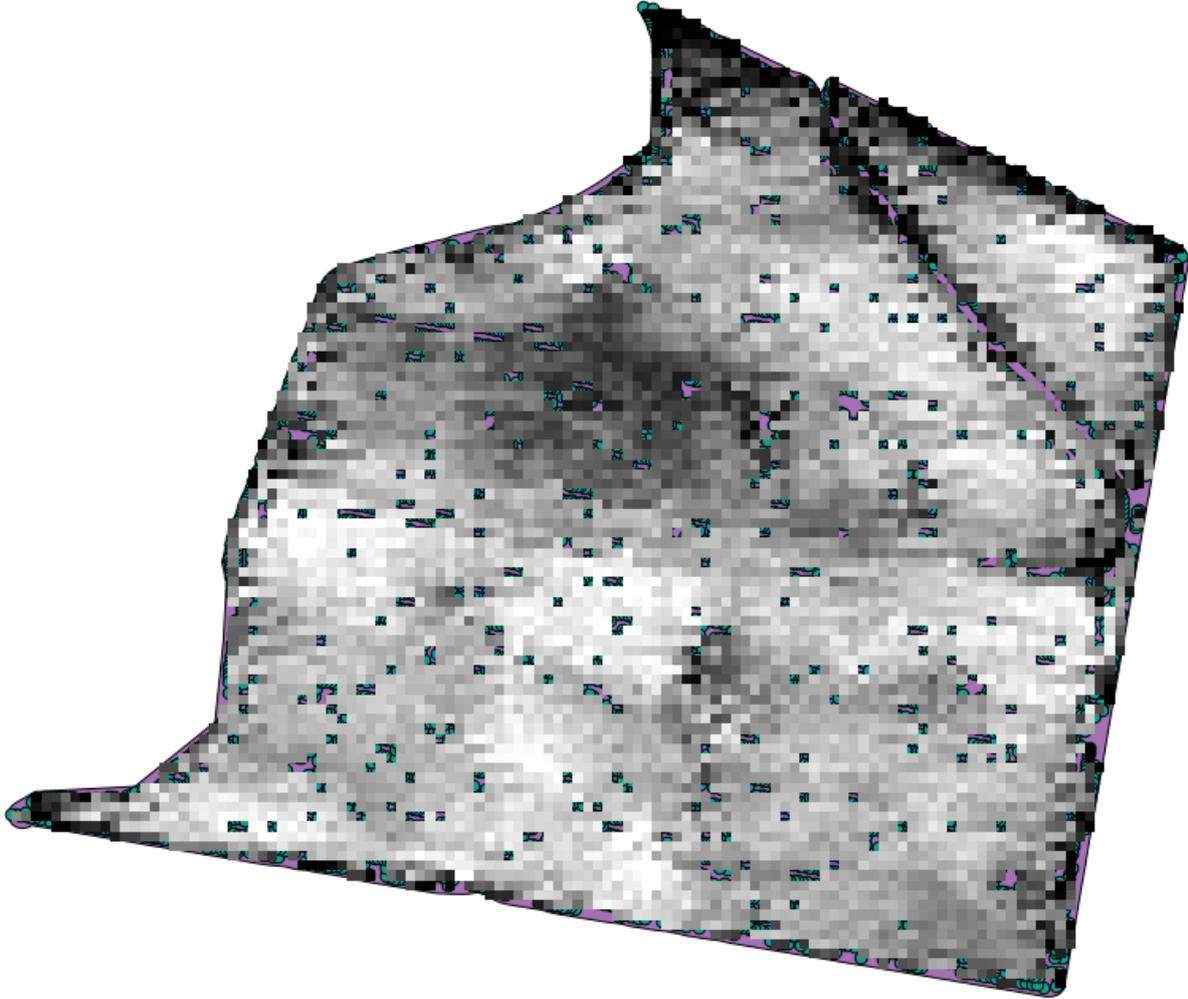
最終的なフィルタレイヤはポイントのセットが減っていますが、オリジナルものと類似して見えます。しかしより少ないポイントを含みます。それらの属性テーブルで比較することで確認することができます。

では *Rasterize* アルゴリズムを使用してレイヤーをラスタライズしましょう。

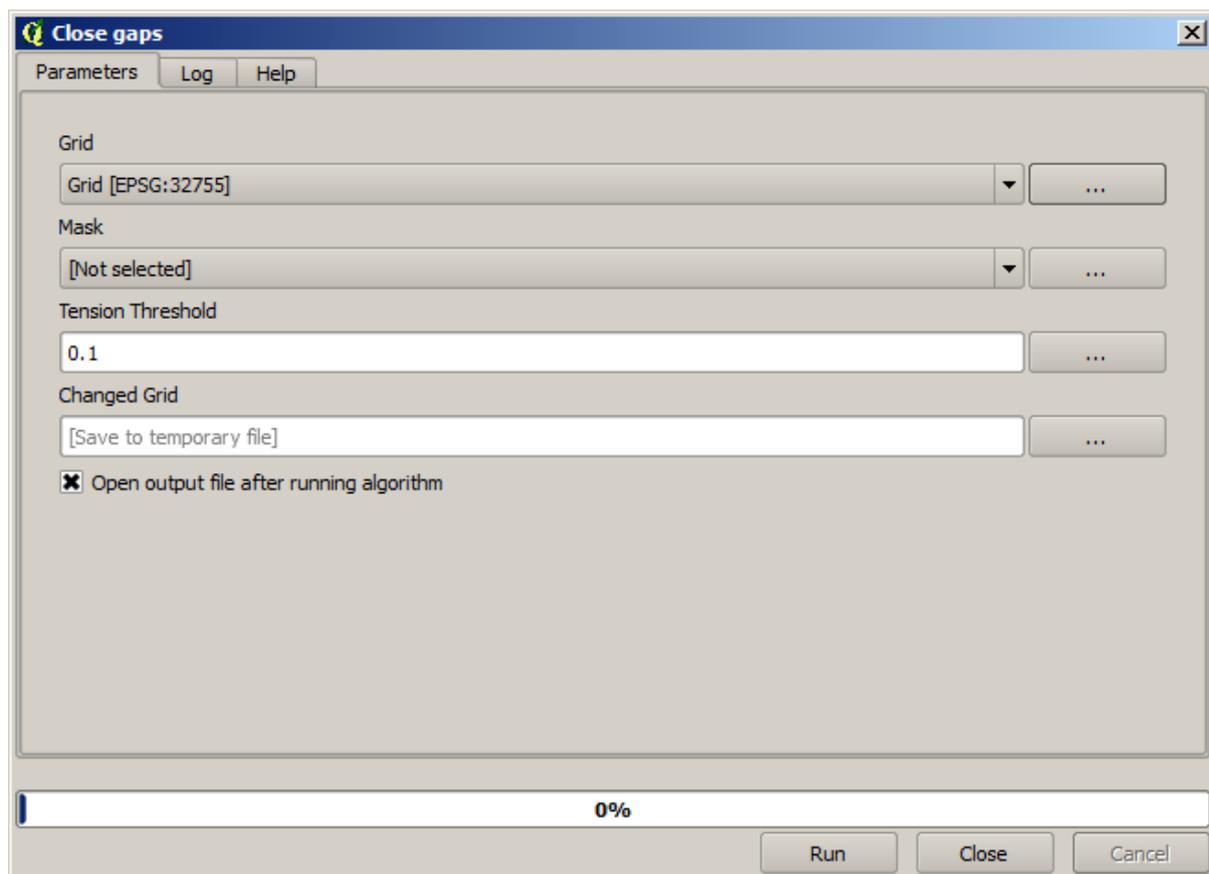


Filtered points レイヤは二番目のフィルタの1つを結果として参照します。それはアルゴリズムによって名前が指定されているため、最初のフィルタによって生成されたものと同じ名称を持ちますが、最初のものを使うべきではありません。我々が他の何かのためにそれを使っていないため、あなたは問題なく混乱を避けるプロジェクトからそれを取り除くことができ、最終的なフィルタ処理したレイヤを残すことができます。

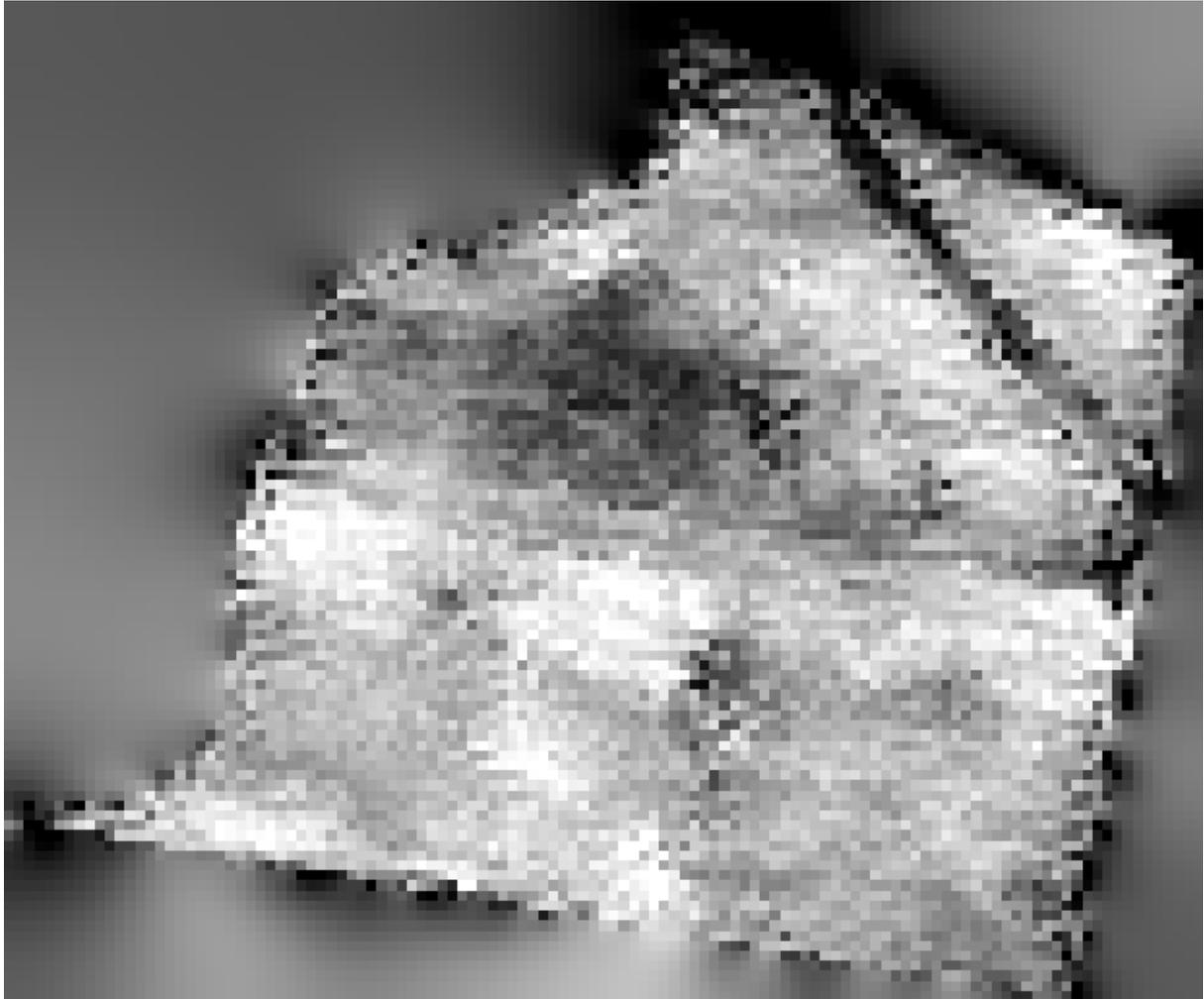
結果のラスタレイヤはこのように見えます。



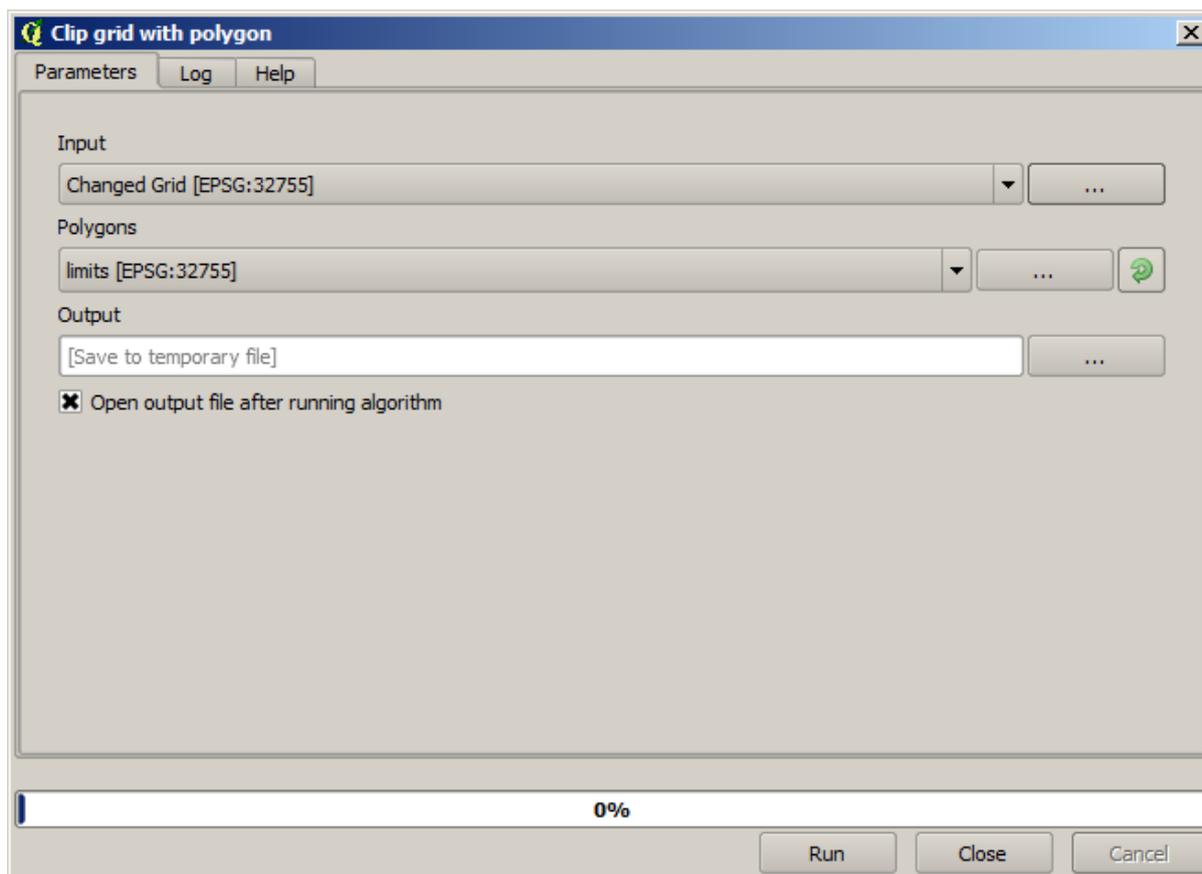
それはすでにラスタレイヤですが、一部のセルが欠落しています。ラスタライズしたベクタレイヤ由来のポイントとデータなし値を含むセルで有効な値のみを含みます。この欠損値を埋めるため、*Close gaps* アルゴリズムを使用することができます。



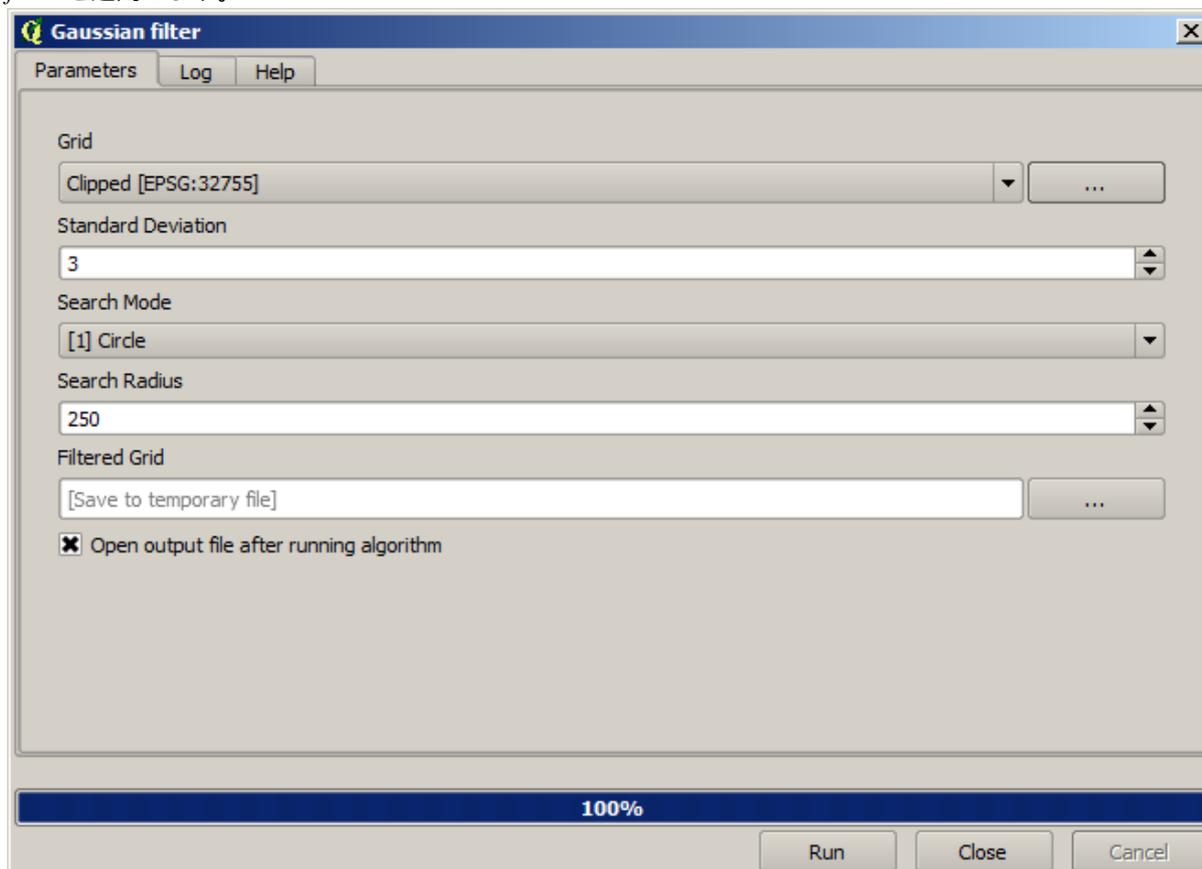
データなし値のないレイヤはこのように見えます。



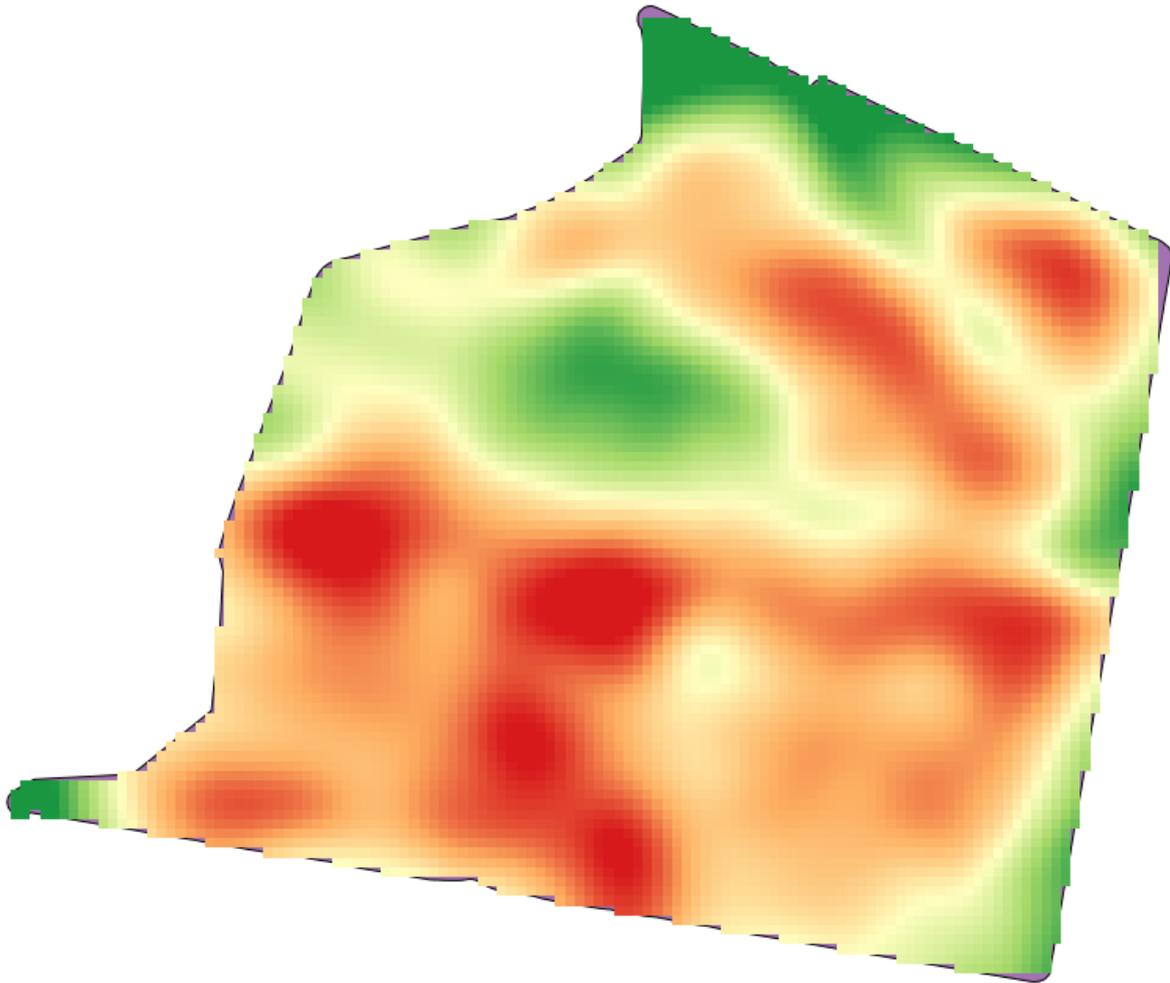
収穫産出高が計られた地域において、データによっておおわれる地域を制限するために、我々は提供された制限レイヤでラスタレイヤを切り抜くことができます。



スムーズな結果（精度は低いですがサポートレイヤとして背景にレンダリングされる）として、レイヤに *Gaussian filter* を適用します。



上記のパラメータで、次の結果が得られます。



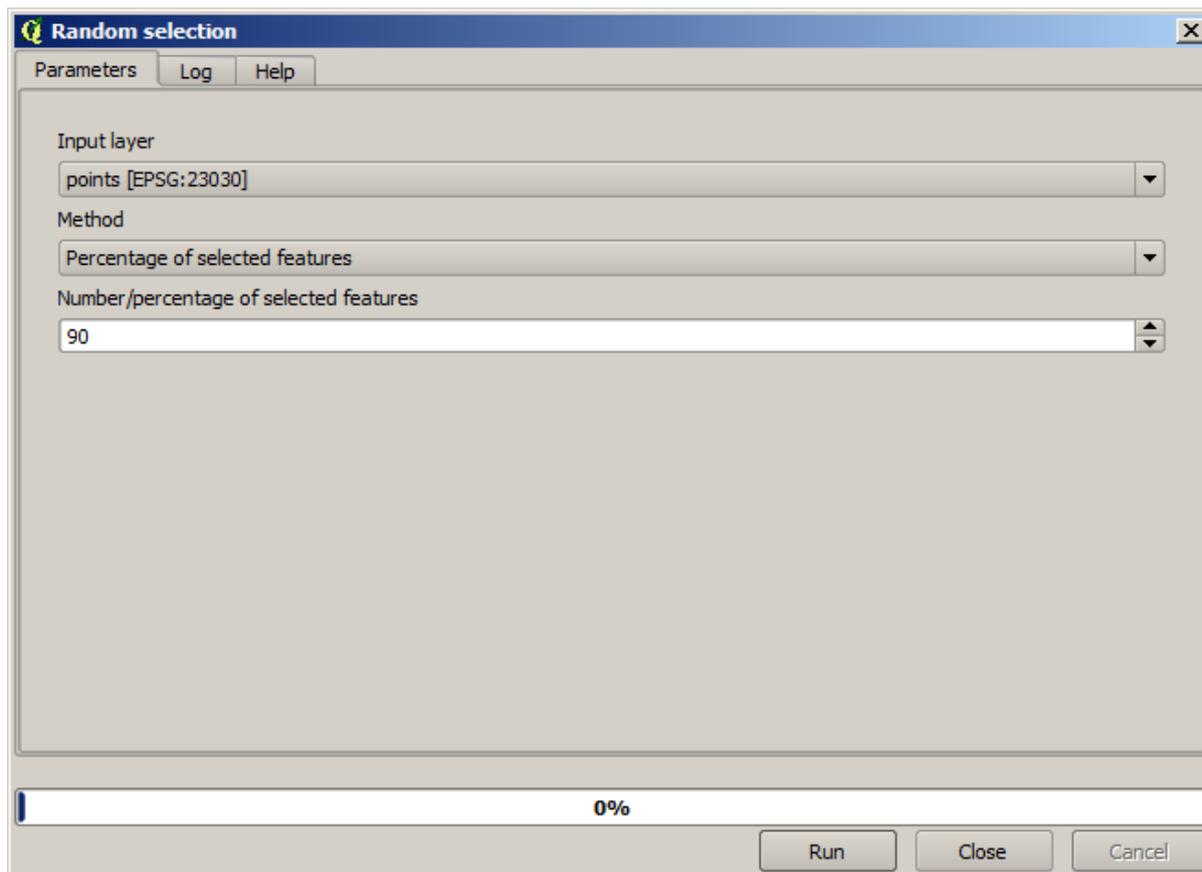
17.22 その他の補間

ノート: この章では、補間アルゴリズムが使用されている別の実用的な場合を示しています。

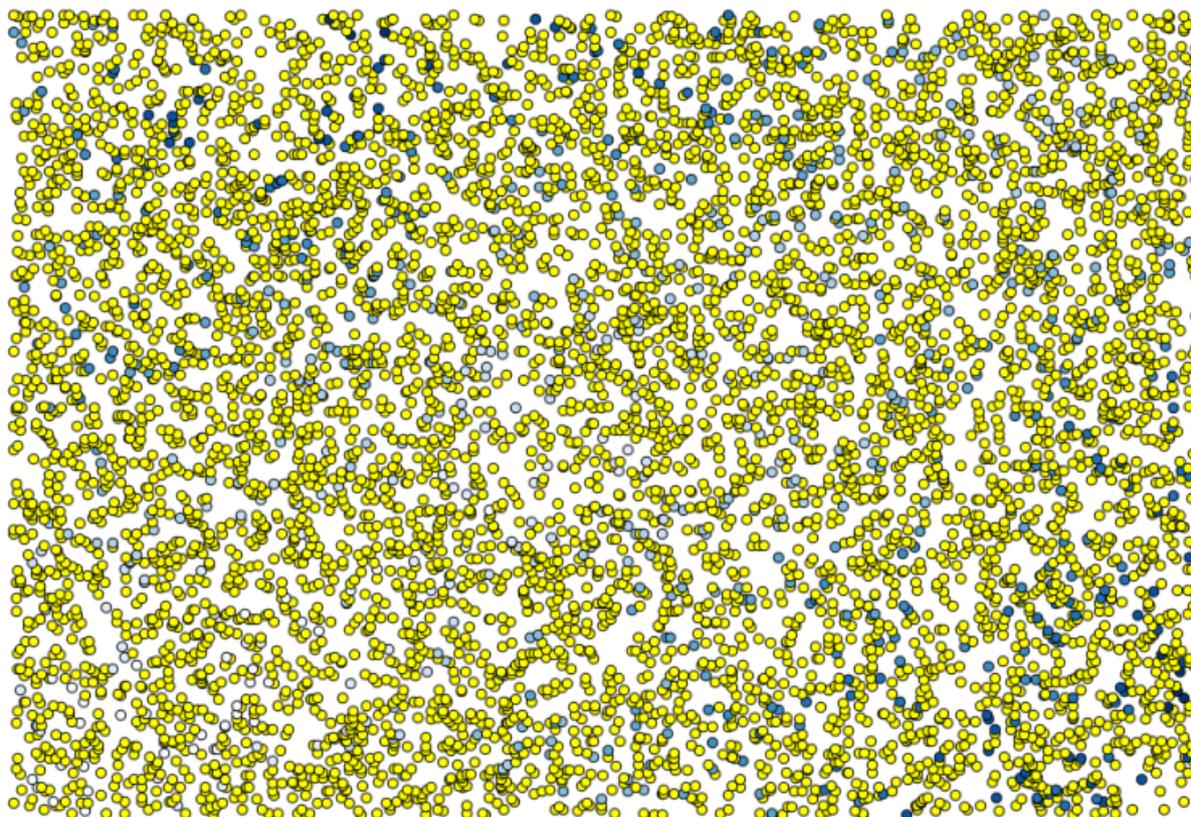
補間は一般的な技術であり、QGIS 処理フレームワークを使用して適用することができるいくつかの技術を実証するために使用することができます。このレッスンでは、すでに導入されたいくつかの補間アルゴリズムを使用しますが、別のアプローチがあります。

このレッスン用データは、標高データと、この場合も、点レイヤーを含んでいます。前のレッスンで行ったのと同じ方法で、ずっとそれを補間しようとしているが、今回は、補間処理の品質を評価するためにそれを使用するために元のデータの一部が保存されます。

まず、ポイントレイヤをラスタライズし、得られていない入力する必要があり - データセルを、しかし、レイヤー内の点のほんの一部を使用しました。10 が保存されます% of 後でチェックのポイントは、90 を持っている必要があります% o 補間のための準備がポイント F。そうするために、シェイプレイヤーをランダムにスプリット アルゴリズムを使用でき、すでに前のレッスンで使用していたが、それを行うには良い方法があります、任意の新しい中間レイヤーを作成することなく。その代わりに、ちょうど補間に使用したいポイント (90 % Fraction を選択できます) その後、アルゴリズムを実行します。すでに見てきたように、ラスタライズアルゴリズムは、これらの選択されたポイントを使用し、残りを無視します。選択は、ランダム選択 アルゴリズムを使用して行うことができます。次のパラメータを使用して、それを実行します。



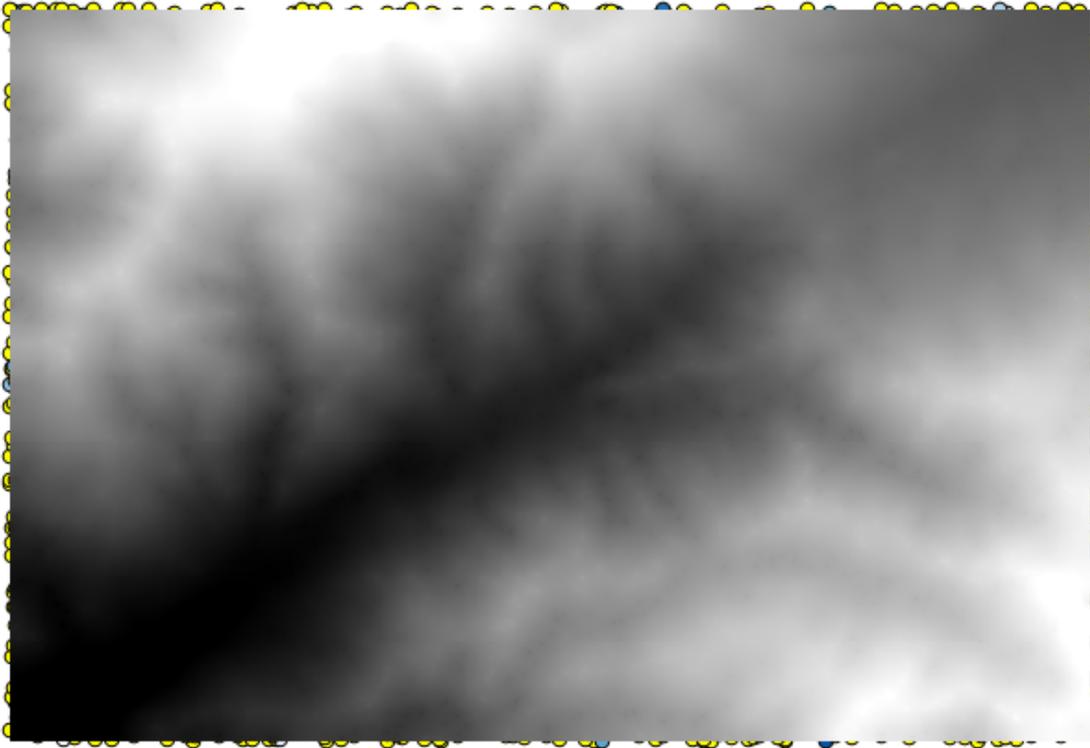
それはラスター化するレイヤー内の点の 90 % を選択します



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。

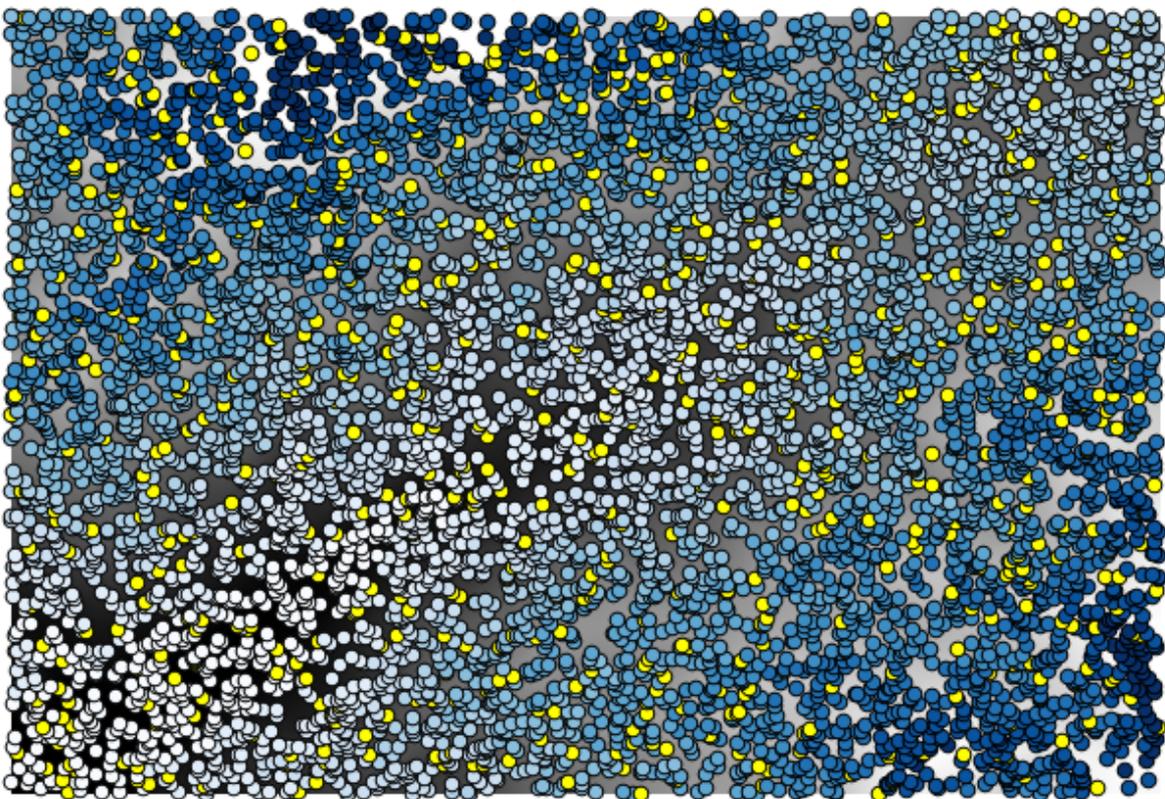
今第 1 ラスタレイヤを得るため*ラスター化*アルゴリズムを実行し、無充填する*アルゴリズム*閉じる

ギャップを実行 - データセル [セル解像度 100 メートル]。



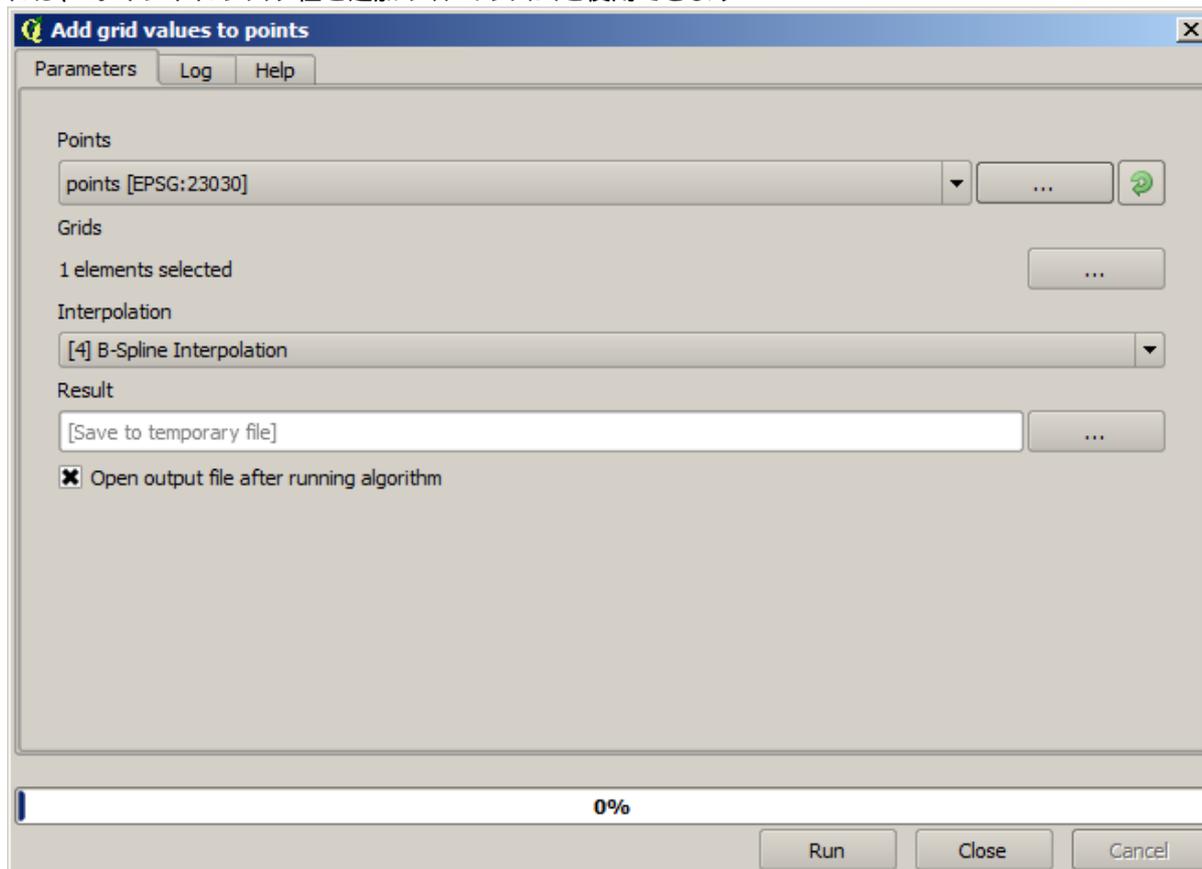
補間の品質を確認するために、今、選択されていないポイントを使用できます。この時点で、実際の標高（ポイント層における値）と補間標高（補間ラスタレイヤの値を）知っています。これらの値の差を計算することにより、2つを比較できます

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、の選択を反転してみましょう。



ポイントは、元の値ではなく、補間されたものが含まれています。新しいフィールドにそれらを追加する

には、ポイントにラスタ値を追加 アルゴリズムを使用できます



(アルゴリズムは、複数のラスタをサポートしていますが、1つだけ必要)を選択するラスタレイヤは、補間から生じる1つです。補間することを名前を変更し、そのレイヤ名は、追加するフィールドの名前に使用されるものです。

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を持っています。

Attribute table - Result :: Features total: 703, filtered: 703, selected: 0

	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

Show All Features

今、私たちは、この作業のためにフィールドの電卓を使用します。***フィールドの電卓*アルゴリズム**を開き、次のパラメータを使用して、それを実行します。

Field calculator

Parameters Log Help

Input layer: Result [EPSG:23030]

Result field name: error

Field type: Float

Field length: 10

Field precision: 5

Formula: `abs(VALUE - interpolat)`

Output layer: [Save to temporary file]

Open output file after running algorithm

0%

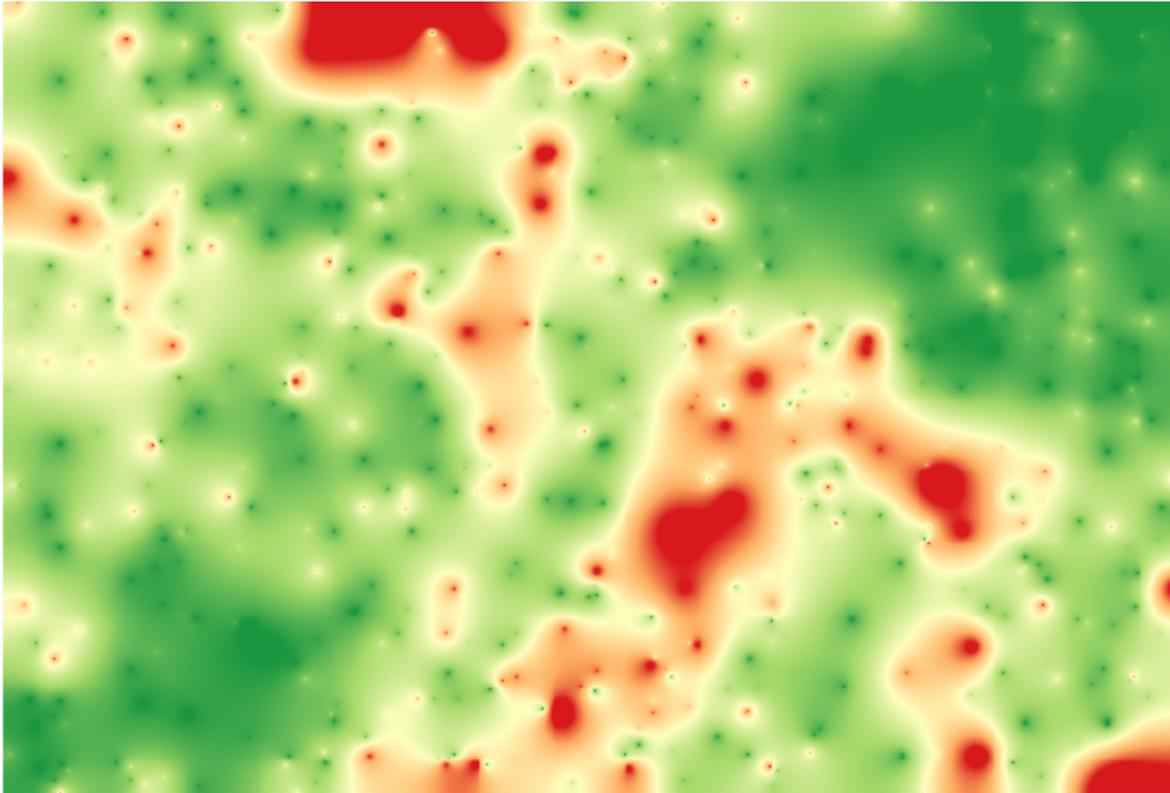
Run Close Cancel

ラスタレイヤ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には、我々は彼らのそれぞれが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を取得します。

その値に応じてその層を表現すること私たちの最大の不一致が見つかった場所の最初のアイデアを与えるだろう。

	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レイヤを取得します。



GRASS -> *v.sample* で直接にと同じ情報（原点値と補間のものとの差）を得ることもできます。

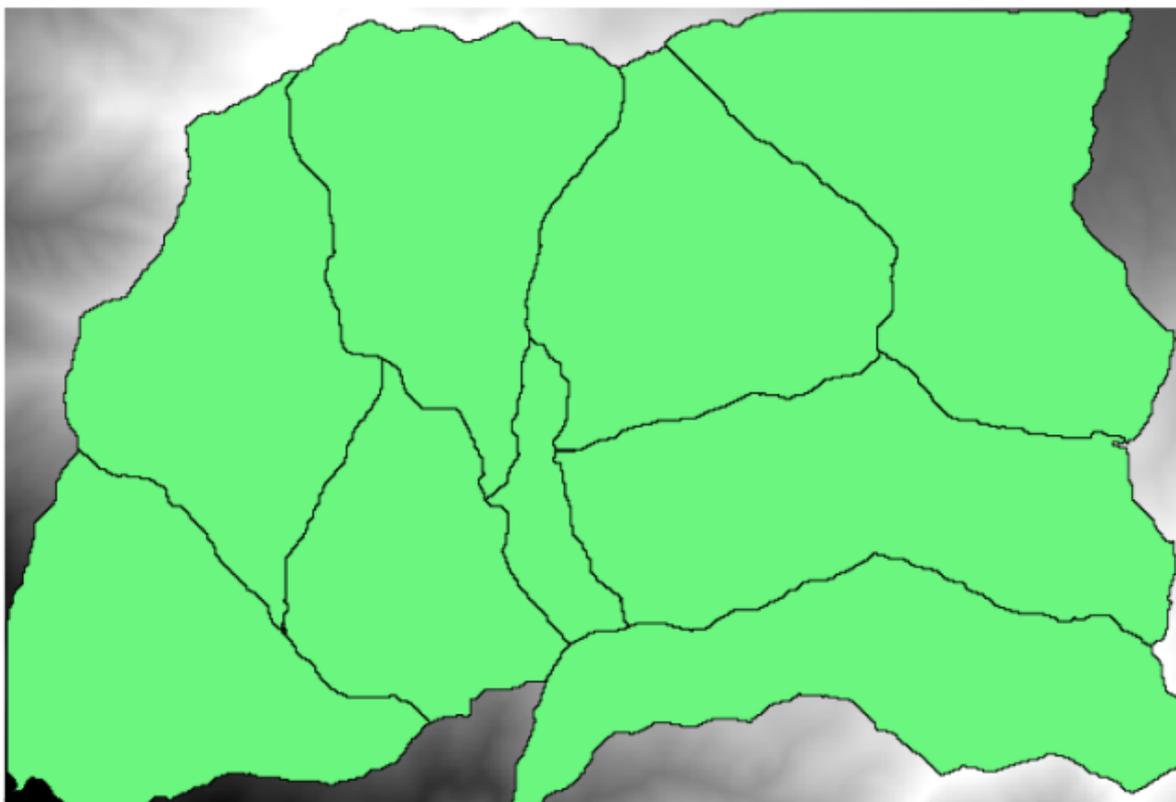
このレッスンの初めに、ランダムな選択を実行する際に導入ランダム成分があるので、結果はこれらのものと異なる場合があります。

17.23 アルゴリズムの反復実行

ノート： このレッスンでは、入力ベクトル層にフィーチャを反復繰り返しそれらを実行して、ベクトル層を使用するアルゴリズムを実行する別の方法を示しています

すでに処理タスクを自動化する一つの方法であるグラフィカルなモデラーを、知っています。しかし、いくつかの状況では、モデラーは、与えられたタスクを自動化するために必要なものではないかもしれません。アルゴリズムの反復実行：のは、これらの状況のいずれかを確認し、どのように簡単に別の機能を使用して、それを解決するためにしてみましょう。

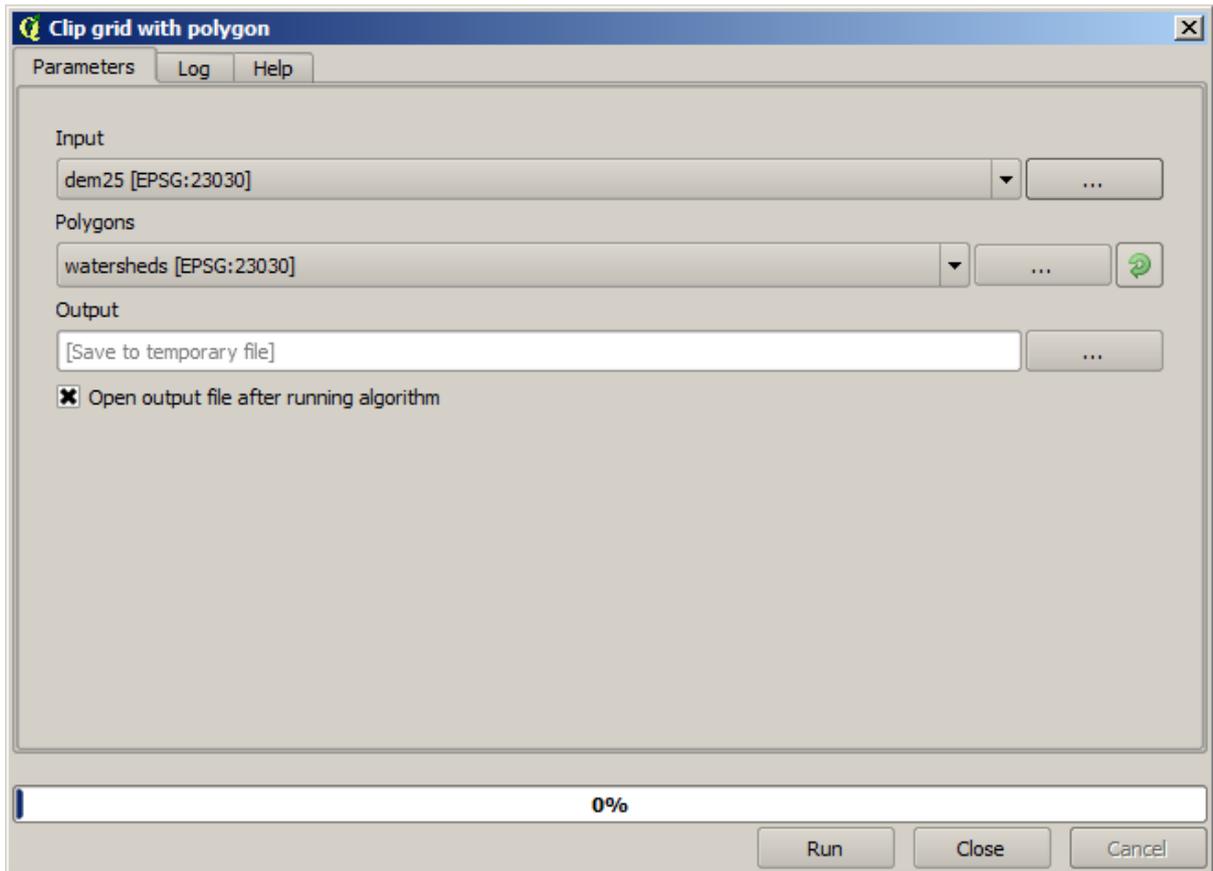
この章に対応するデータを開きます。それは次のようになります。



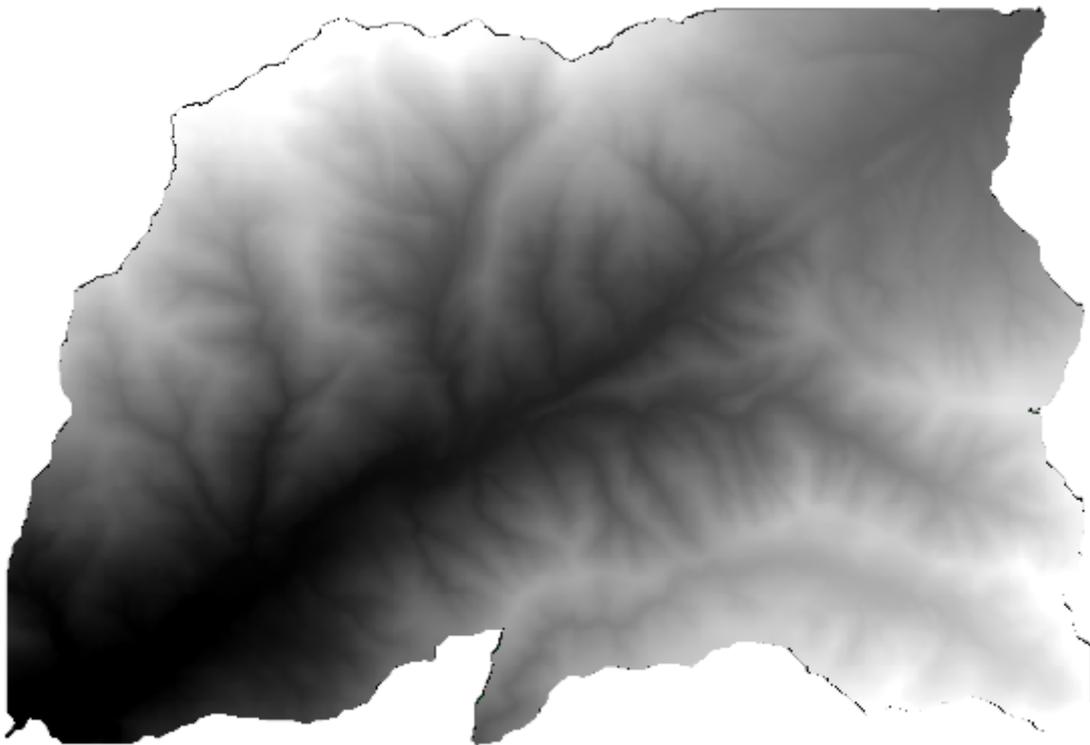
よく知られた、前の章から DEM とそれから抽出された流域のセットを認識します。それらのそれぞれは、単一の流域に対応するだけの標高データを含む、いくつかの小さなレイヤに DEM をカットする必要があることを想像してみてください。後で、そのようなその平均標高またはそれ hypsographic 曲線として、各流域に関連するいくつかのパラメータを計算したい場合には有用であろう。

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。

ポリゴンレイヤでラスタレイヤをクリッピングするために使用するアルゴリズムは、ポリゴンでラスタをクリップ と呼ばれ、以下のパラメータ] ダイアログボックスを持っています。



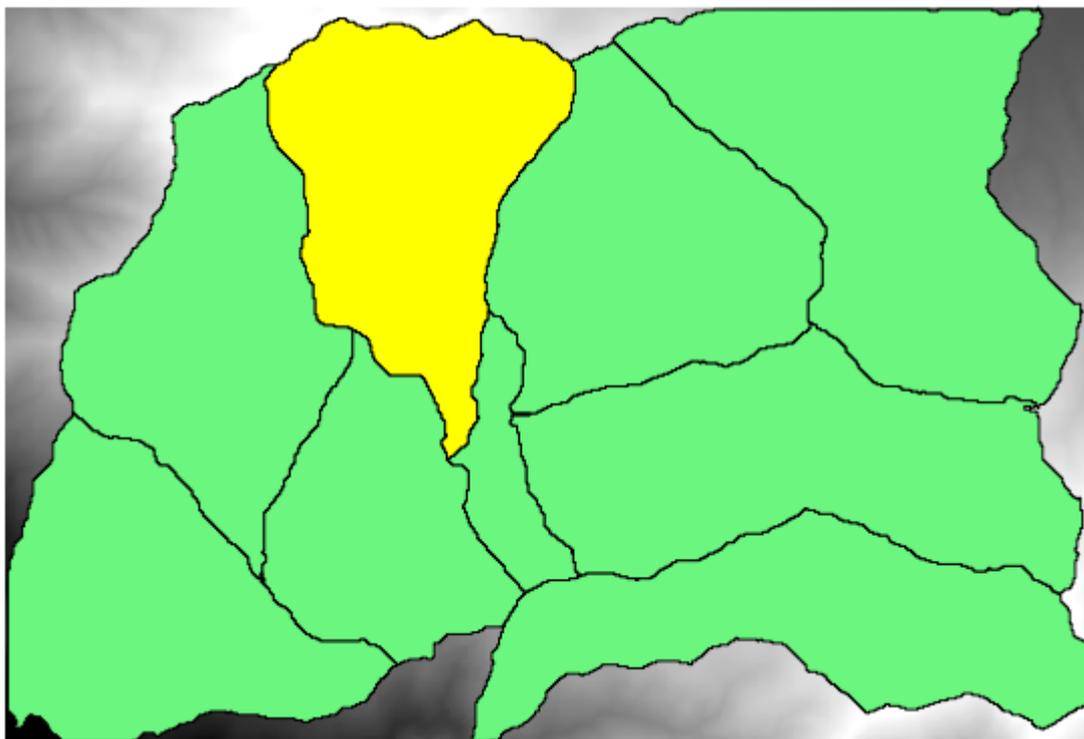
流域レイヤーと DEM 入力としてを使用してそれを実行でき、次の結果が得られます。



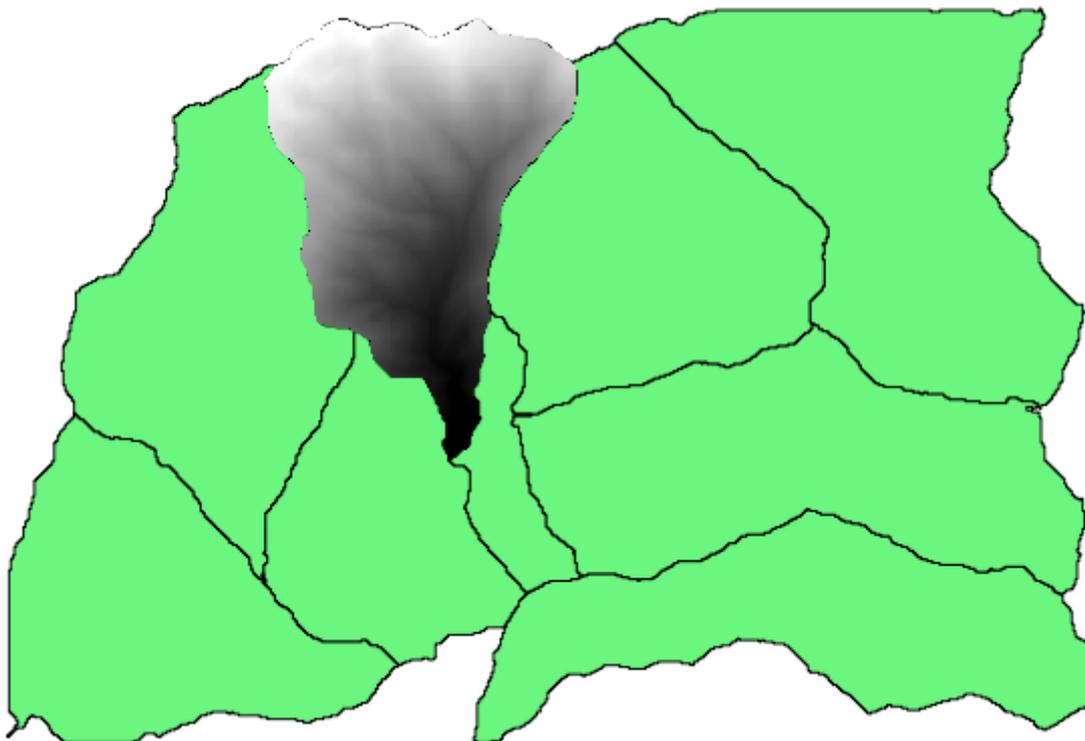
ご覧のように、すべての分水界ポリゴンで覆われた領域が使用されています。

DEM が希望流域を選択し、前に行ったようなアルゴリズムを実行することにより、単に 1 つの流域でク

リップできます。



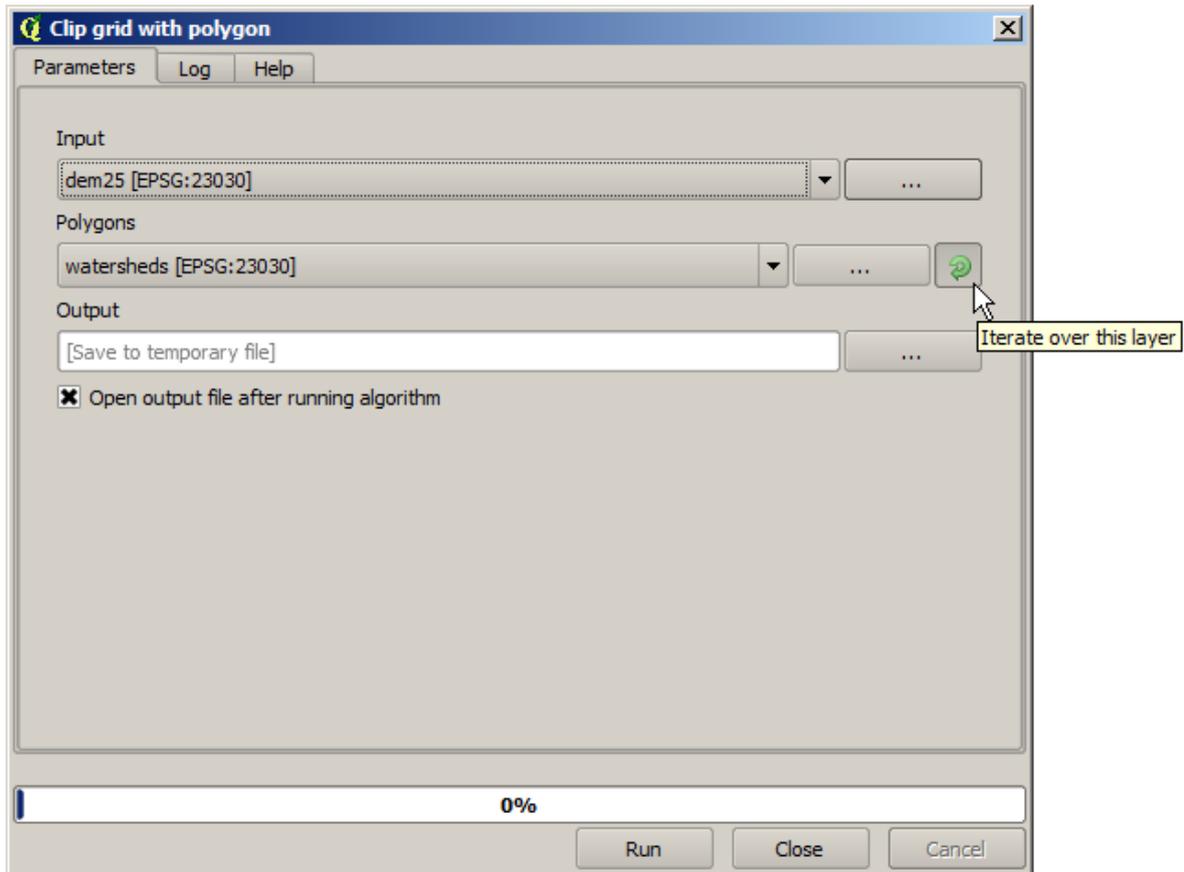
唯一の選択した機能が使用されているので、唯一の選択されたポリゴンは、ラスタレイヤーをトリミングするために使用されます



すべての流域のためにこれを行うと、探している結果を生成しますが、それはそれを行うための非常に実用的な方法のように見えません。代わりに、その選択とクロップルーチンを自動化する方法を見てみましょう。

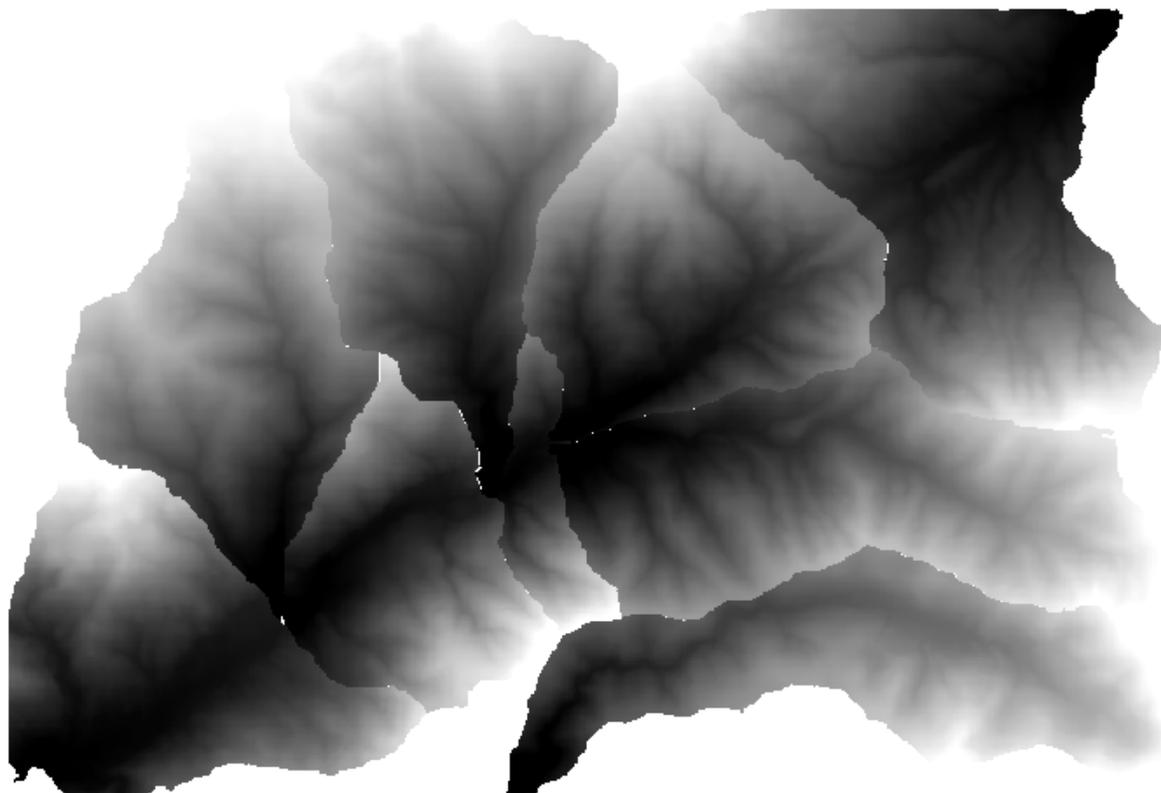
まず、以前の選択を削除するため、すべてのポリゴンが再び使用されます。今*アルゴリズムポリゴンで*

クリップラスタを開き、前と同じ入力を選択していますが、右側で見つけるボタンでこの時間をクリック
 - あなたは流域層を選択したベクトルレイヤー入力の右側。



このボタンは、それらのそれぞれが単一のポリゴンを含む、選択した入力レイヤは、その中に発見された機能な限り多くのレイヤに分割されることとなります。それによって、アルゴリズムは、それらのシングルポリゴンレイヤのそれぞれに1つずつ、繰り返し呼び出されます。結果は、代わりに、このアルゴリズムの場合にだけ1つのラスタレイヤから、ラスタレイヤのセット、アルゴリズムの実行の1つに対応するそれらのそれぞれであろう。

説明したようにクリッピングアルゴリズムを実行する場合に得られる結果です。



各レイヤ、黒と白のカラーパレット（または使用しているどのようなパレット）のために、その最小値からその最大値に、異なって調節されます。それは異なる部分と色がレイヤの間の境界に一致していないようで見ることができる理由です。値は、しかし、一致します。

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。

17.24 アルゴリズムのより多くの反復実行

ノート：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。

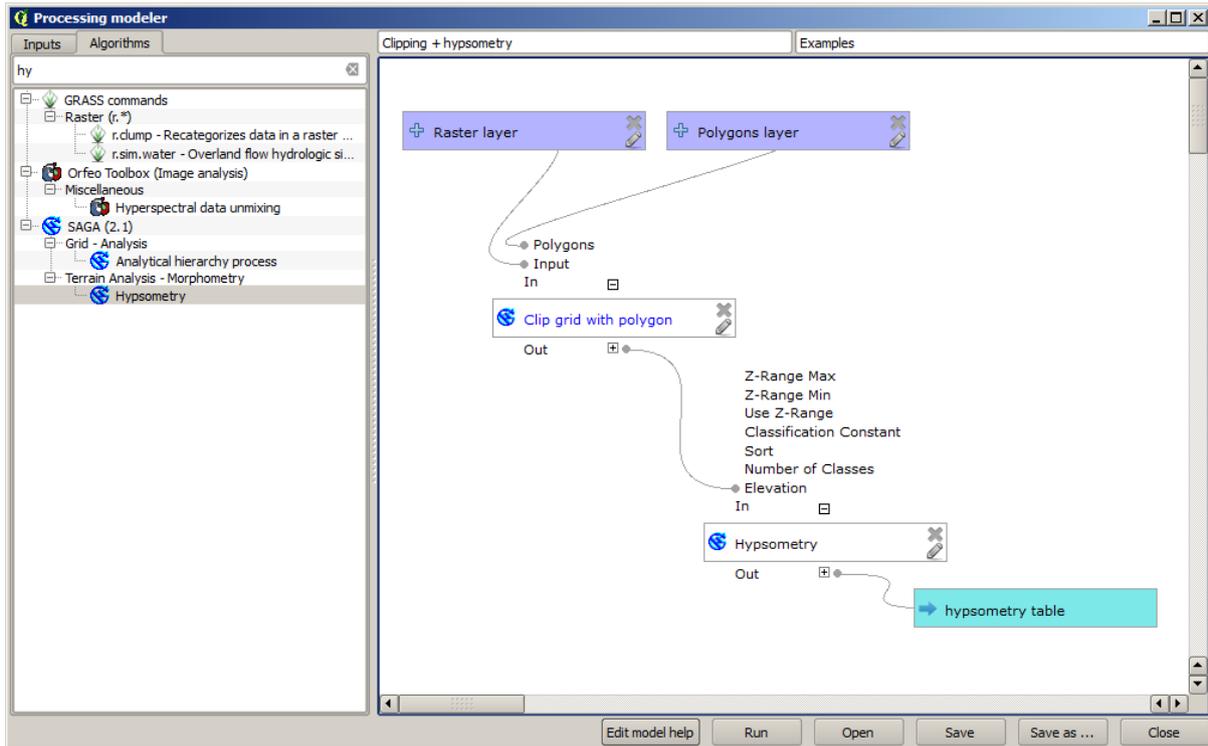
アルゴリズムの反復実行は、内蔵されたアルゴリズムに対してのみでなく、モデルなど、自作できるアルゴリズムに対しても使用可能です。より複雑な結果を簡単に得られるように、モデルとアルゴリズムの反復実行を組み合わせる方法を見ていきます。

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水深曲線を計算し、水深が流域内にどう分布するかを考察します。

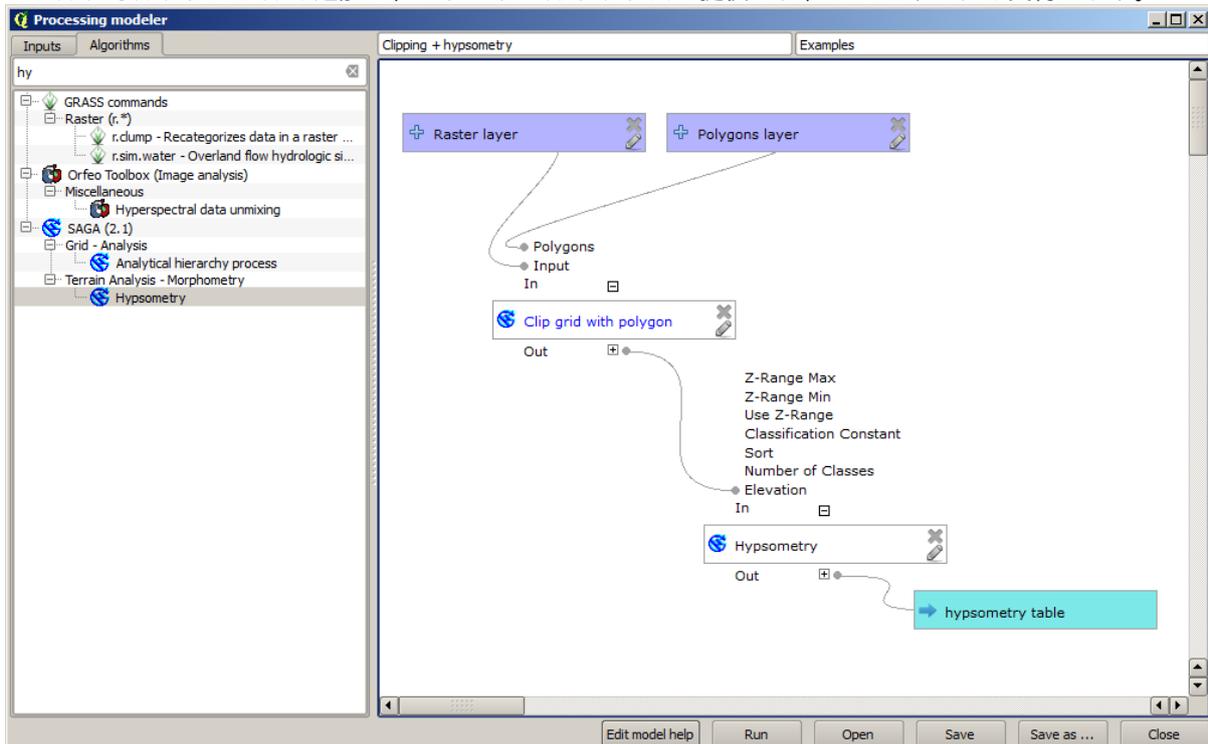
いくつかのステップ（hypsothetic 曲線を計算+クリッピング）を必要とするワークフローを持っているので、我々はモデラーに移動し、そのワークフローの対応するモデルを作成する必要があります。

既にこのレッスンのためのデータフォルダ内に作成したモデルを見つけることができますが、最初は自分で作成しようとするのが良いでしょう。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。だからこのモデルは、レイヤーは何も作成せず、単に曲線データを有するテーブルを生成します。

モデルは次のようになります。

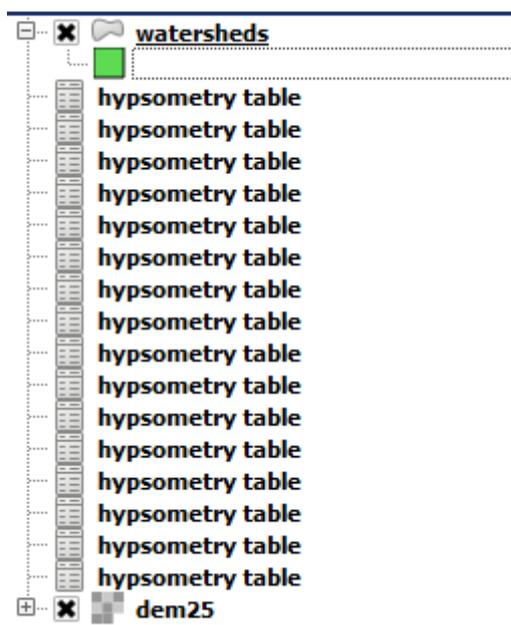


モデルフォルダにモデルを追加し、それはツールボックスで提供され、そして今それを実行します。

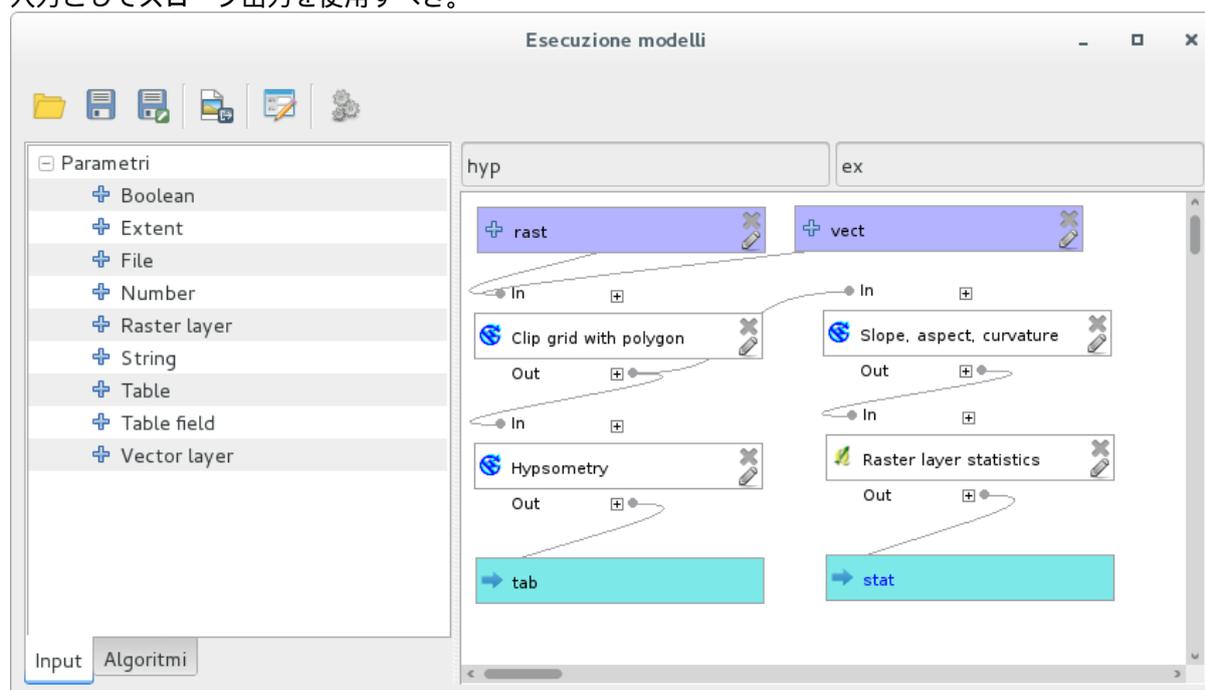


DEM や流域の流域を選択し、アルゴリズムが反復的に実行する必要があることを示すボタンをトグルすることを忘れないでください。

このアルゴリズムは、複数回実行され、対応するテーブルが作成され、あなたの QGIS プロジェクトで開きました。



モデルを拡張し、いくつかの斜面の統計を計算することにより、この例をより複雑にできます。**勾配、向き、曲率**アルゴリズムをモデルに追加し、それから**ラスタ統計**アルゴリズムを追加します、その唯一の入力としてスロープ出力を使用すべき。



今のモデルを実行すると、離れてテーブルから、あなたは統計を持つページのセットを取得します。これらのページは、結果ダイアログで利用できるようになります。

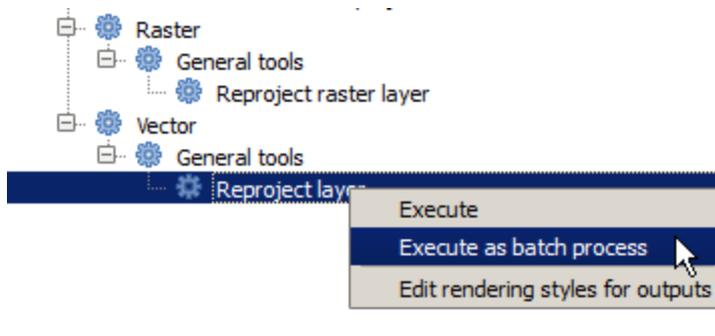
17.25 バッチ処理インターフェース

ノート: このレッスンではバッチ処理インターフェースを紹介します。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。

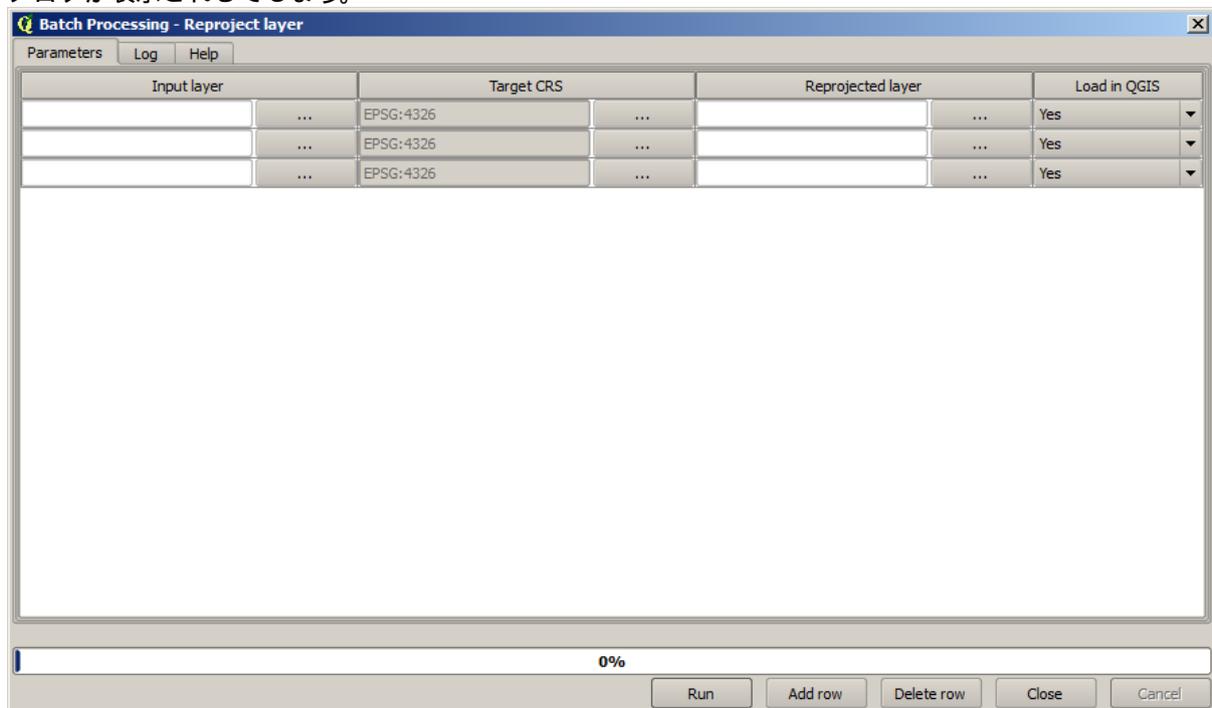
特定のアルゴリズムが異なる入力で繰り返し実行する必要があることがあります。これは、例えば、入力ファイルのセットをあるフォーマットから他のフォーマットに変換するときとか、所与の投影におけるい

いくつかのレイヤーを他の投影に変換する必要があるときです。

その場合には、ツールボックスに繰り返しアルゴリズムを呼び出すのは最良の選択肢ではありません。代わりに、バッチ処理インターフェースを使用すべきです。そうすれば与えられたアルゴリズムの複数の実行が大幅に簡略化されます。バッチプロセスとしてアルゴリズムを実行するには、ツールボックスでそれを見つけて、ダブルクリックではなく右クリックして、バッチ処理として実行を選択します。



この例では、再投影アルゴリズムを使用するので、前述したようにそれを見つけて実行します。次のダイアログが表示されるでしょう。



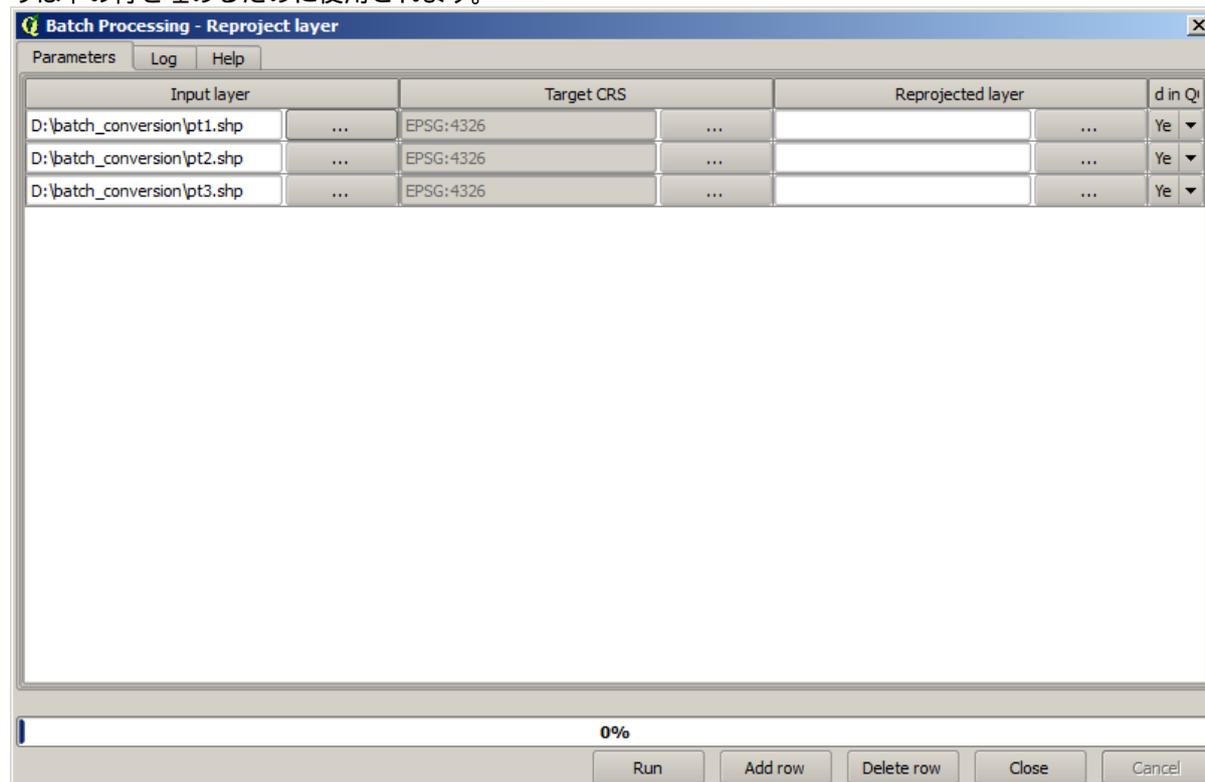
このレッスンのためのデータを見てみると、3つのシェープファイルのセットは含むが、QGIS プロジェクトファイルが含まれていないことがわかります。アルゴリズムは、バッチプロセスとして実行されると、レイヤーの入力は、現在の QGIS プロジェクトまたはファイルのいずれかから選択できるからです。それによって、例えば指定したフォルダ内のすべてのレイヤーなど、大量のレイヤーを簡単に処理できるようになります。

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。しかし水平にその行の、実行ダイアログ - 行のセルは、通常、単一のように、互いに上下に配置されていないアルゴリズムによって必要とされるパラメータに対応します。

実行するバッチ・プロセスを定義すると、対応する値を持つテーブルを充填することによって一つであり、ダイアログ自体は、この作業を容易にするためにいくつかのツールが含まれています。

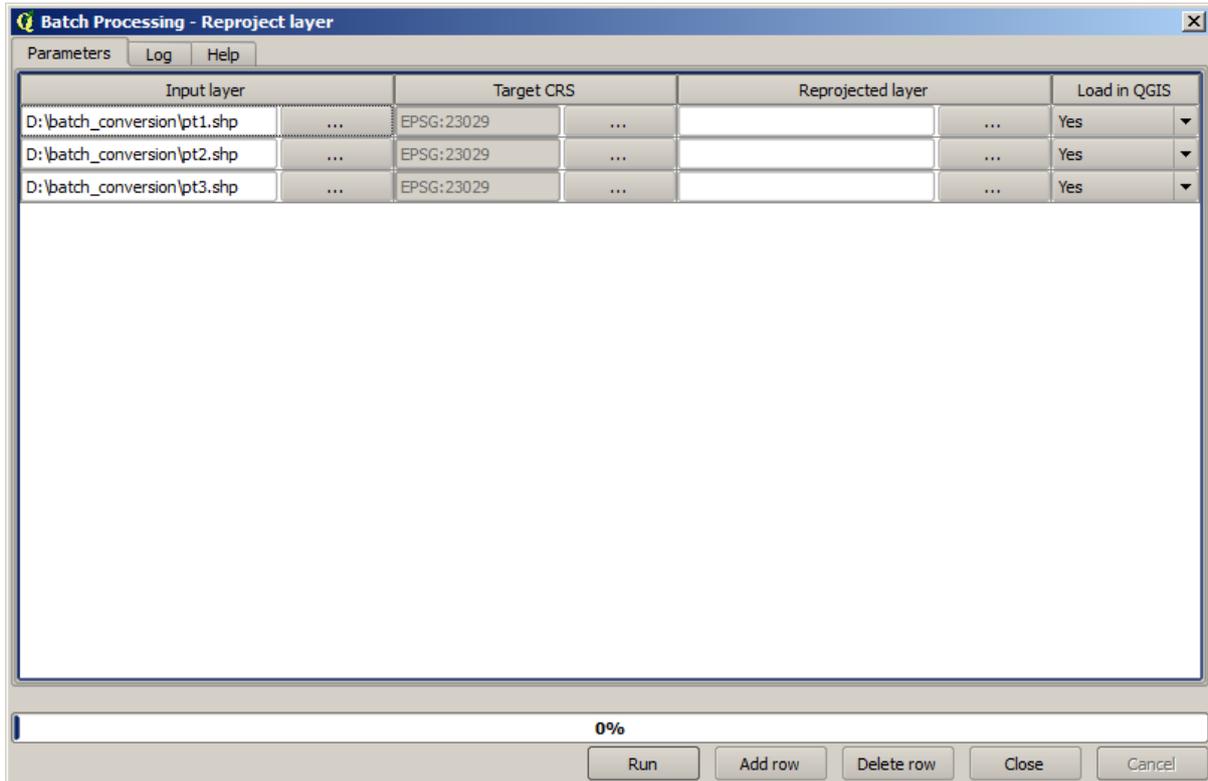
フィールドの一つ一つを入力始めましょう。入力する最初の列は、入力レイヤー列です。処理したいレイヤーのそれぞれの名前を入力する代わりに、それらのすべてを選択してダイアログが各行にひとつを入れるようにさせることができます。左上のセル内のボタンをクリックし、ポップアップするファイル選択ダイアログで、再投影する3つのファイルを選択します。行ごとにそれらの1つだけが必要とされるので、残

りは下の行を埋めるために使用されます。



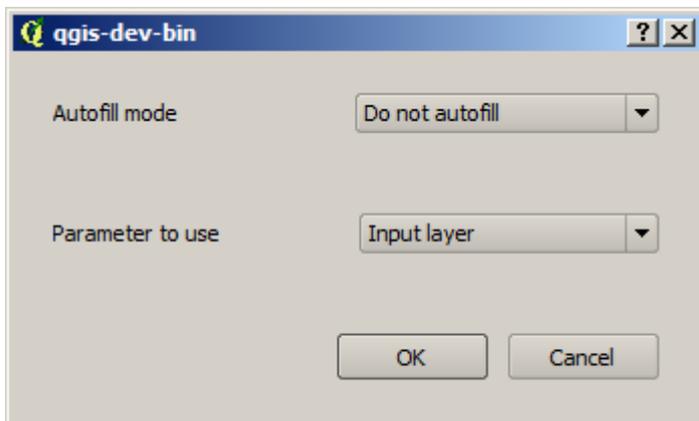
デフォルトの行数は3で、これはちょうど変換する必要があるレイヤーの数ですが、より多くのレイヤーを選択した場合、新しい行が自動的に追加されます。手動でエントリを記入したい場合は、*行を追加*ボタンを使用して複数の行を追加できます。

これらのレイヤーをすべて EPSG:23029 CRS に変換しようとしているので、2 番目のフィールドでその CRS を選択する必要があります。すべての行で同じにしたいが、各行ひとつひとつに対してそれを行う必要はありません。その代わりに、1 行目（最上の 1 つ）にその CRS を、対応するセル内のボタンを使用して設定し、列ヘッダをダブルクリックします。それにより、列のすべてのセルがトップセルの値を使用して埋められます。

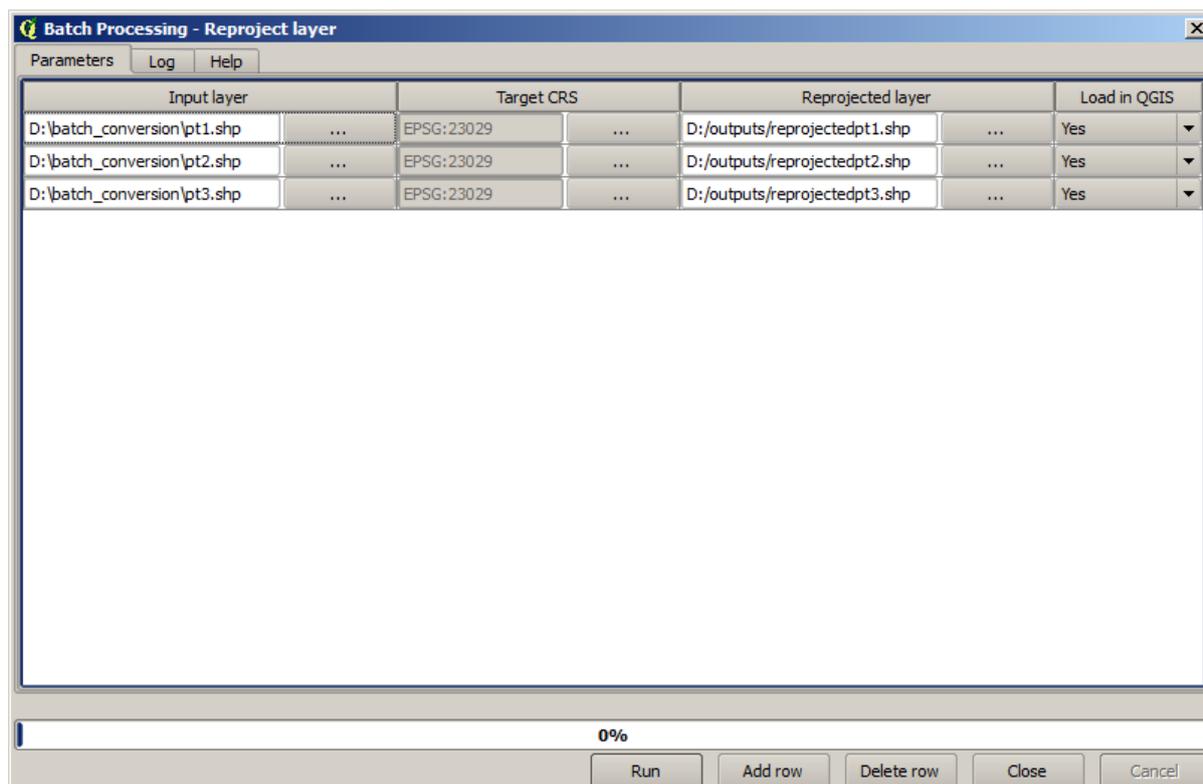


最後に、対応する再投影レイヤーが含まれています各実行のための出力ファイルを選択する必要があります。もう一度、ちょうど最初の行のためにそれをやらせます。(、例えば reprojected.shp) の上部セル内のボタンをクリックすると、出力ファイルを置きたいフォルダに、ファイル名を入力します。

さて、ファイル選択ダイアログ上で *OK* をクリックすると、ファイルは自動的にセルに書き込まれず、以下のような入力ボックスが代わりに表示されます。



最初のオプションを選択した場合のみ、現在のセルが入力されます。他のもののいずれかを選択した場合は、以下のすべての行は、与えられたパターンで埋められます。この場合は、パラメータ値で塗りつぶしオプションを選択しようとしている、と下のドロップダウンメニューで、その後入力レイヤー値。それは各出力ファイル名が異なる作り、追加したファイル名に追加する(つまり、レイヤ名です)入力レイヤーの値が発生します。バッチ処理テーブルには、次のようになります。



最後の列は、得られるレイヤーを現在の QGIS プロジェクトに追加するかどうかを設定します。このような場合には、結果を見ることができるよう、デフォルトの はい オプションのままにしておきます。

OK をクリックするとバッチ処理が実行されます。すべてがうまくいった場合は、すべてのレイヤーが処理され、3 つの新しいレイヤーが作成されているでしょう。

17.26 バッチ プロセッシング 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、いかなる問題を見つけたら画像とともに報告ください。

ノート: このレッスンではバッチ処理インター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ースで使用できます。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:イメージを追加

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:イメージを追加

おわかりのようにバッチ処理インターフェースは、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。

17.27 その他のプログラム

Module contributed by Paolo Cavallini - *Faunalia*

ノート: 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。

17.27.1 GRASS

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 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。

17.27.2 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。

これは、一緒にいくつかの必要なライブラリ (LIST) と、個別にインストールする必要があります。

処理の実装の美点は、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 (処理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。

17.27.3 OTB

(また、オルフェオツールボックスとしても知られる) OTB_は、画像処理アルゴリズムのフリーでオープンソースのライブラリです。これは OSGeo4W スタンドアロンインストーラ (NB: 32 ビットのみ) を介して Windows 上にデフォルトでインストールされます。パスは、処理中に設定する必要があります。

標準 OSGeo4W Windows インストールでは、パスは次のようになります:

```
OTB application folder    C:\OSGeo4W\apps\orfeotoolbox\applications
OTB command line tools folder C:\OSGeo4W\bin
```

Debian とデリバティブで、それは次のようになります / usr / bin

17.27.4 他

TauDEM_は、デジタル標高モデル (DEM) 水文情報の抽出および分析のためのツールのスイートです。さまざまなオペレーティングシステムの可用性が変化します。

LASTools_はレーザー測量データを処理し、分析するために、混合フリーと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の可用性が変化します。

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 の以前の一部分は、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。

以上が来ます。

17.27.5 バックエンドの間での比較

バッファとの距離

Let's load *points.shp* and type *buf* in the filter of the Toolbox, then double click on:

- 一定距離 *buffer* : 距離 10000
- 可変距離 *buffer* : 距離フィールドサイズ
- *v.buffer.distance* : 距離 10000
- *v.buffer.column* : *bufcolumn* SIZE
- *シェイプ Buffer* : 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

速度はかなり異なっているかを確認し、さまざまなオプションが用意されています。

読者のための練習 : 異なる方法間の幾何出力の違いを見つけます。

さて、ラスターバッファとの距離 :

- まず、*GRASS->v.to.rast.value* でベクトル *rivers.shp* をロードしラスターライズします。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 を持つこととなります
- 同じく、*SAGA->Shapes to Grid->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 1000,2000,3000 で *r.buffer r.grow.distance* (2 つの地図のうち 1 つめ ; SAGA ラスタで行われる場合は、2 番目は各河川に関連する領域が表示されます)

ディゾルブ

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* → *Dissolve municipalities.shp* on *PROVINCIA*
- *OGR* → *Dissolve municipalities.shp* on *PROVINCIA*
- *SAGA* - > *ポリゴン Dissolve municipalities.shp* *PROVINCIA* 上 (**NB:** 内側 *boundaries* を保持を未選択にしてください)

ノート: 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 : 異なる方法での違い (形状および属性) を見つけてください。

17.28 補間と等高線作成

Module contributed by Paolo Cavallini - *Faunalia*

ノート: 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。

17.28.1 補間

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、ベクトル “points.shp”、パラメータ “RAIN” に基づいて、すべてを使用してみましょう：

警告：すべての分析に対してセルのサイズは 500 に設定します。

- GRASS → *v.surf.rst*
- SAGA -> マルチレベル B スプライン *Interpolation*
- SAGA -> 逆距離 *Weighted* [電源に逆距離。電源：4。検索半径：グローバル。検索範囲：すべてのポイント]
- GDAL → *Grid (Inverse Distance to a power)* [Power:4]
- GDAL → *Grid (Moving average)* [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 → *Proximity*
- 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、遠くのポイントからの領域が少なく、正確な補間を持っています。

17.28.2 等高線

Various methods to draw contour lines [always step= 10] on the *stddev* raster:

- GRASS → *r.contour.step*
- GDAL → *Contour*
- SAGA → *Contour lines from grid* [**NB:** output shp is not valid, known bug]

17.29 ベクトルの簡素化とスムージング

パオロカヴァリーニで貢献モジュール - *Faunalia* <<http://www.faunalia.eu>> _

ノート：この章では、ベクトルを簡素化し、鋭い角を滑らかにする方法を示します。

時々、ベクトルの簡易版が必要になります。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簡素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。簡素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、元のベクトル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

17.30 太陽光ファームを計画

Module contributed by Paolo Cavallini - Faunalia

ノート: この章では、太陽光発電所を設置するのに適した領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。唯一の南向きの斜面を抽出するには (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。

大規模なファームを構築したいので、連続した大規模な (> 100 ヘクタール) 領域だけを選択します :

- GRASS → *r.reclass.greater*

最後に、ベクトルに変換します

- GRASS → *r.to.vect* [Feature type: area; Smooth corners: yes]

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、分析を繰り返します。

17.31 処理で R スクリプトを使用する

Module contributed by Matteo Ghetta - funded by [Scuola Superiore Sant'Anna](#)

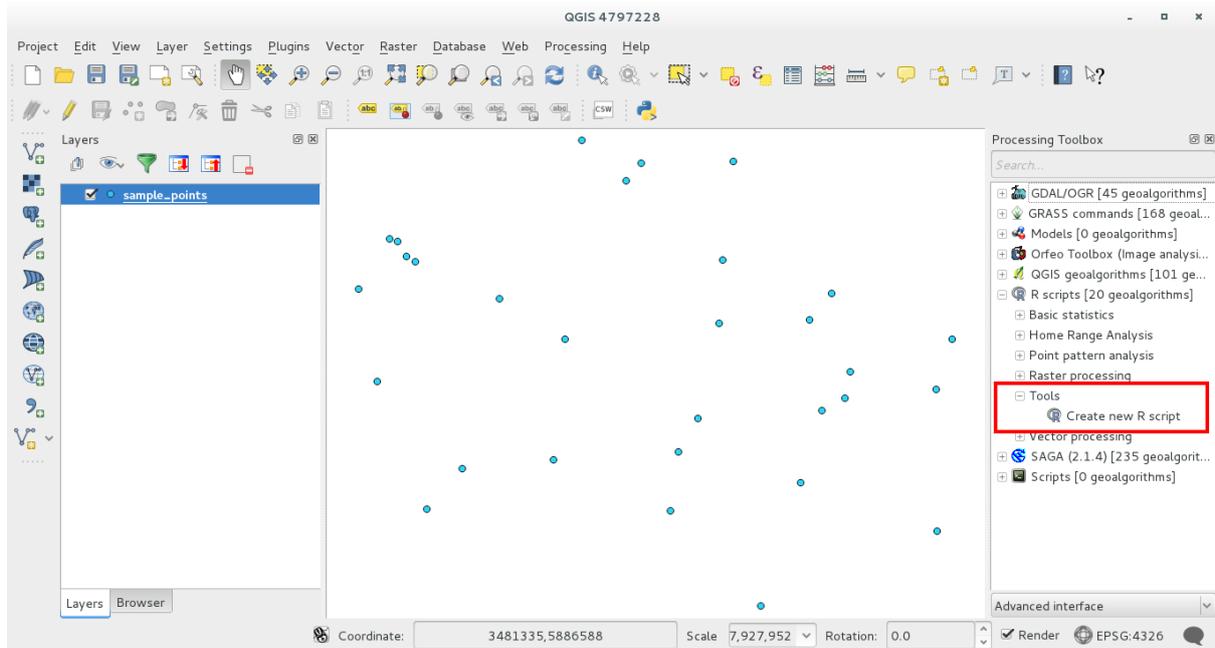
処理は QGIS 内の R スクリプトを作成して実行できます

警告: R がお使いのコンピュータにインストールされ、PATH が正しく設定される必要があります。さらに、処理は外部の R パッケージを呼び出すだけで、それらをインストールできません。だから、R に直接外部のパッケージをインストールしてください。ユーザーマニュアル中の関連する *chapter* を参照。

ノート: いくつかのパッケージ問題がある場合は、多分それは `sp`、`rgdal` と `raster` のように、処理が必要とされる 必須 パッケージが不足しているに関連しています。

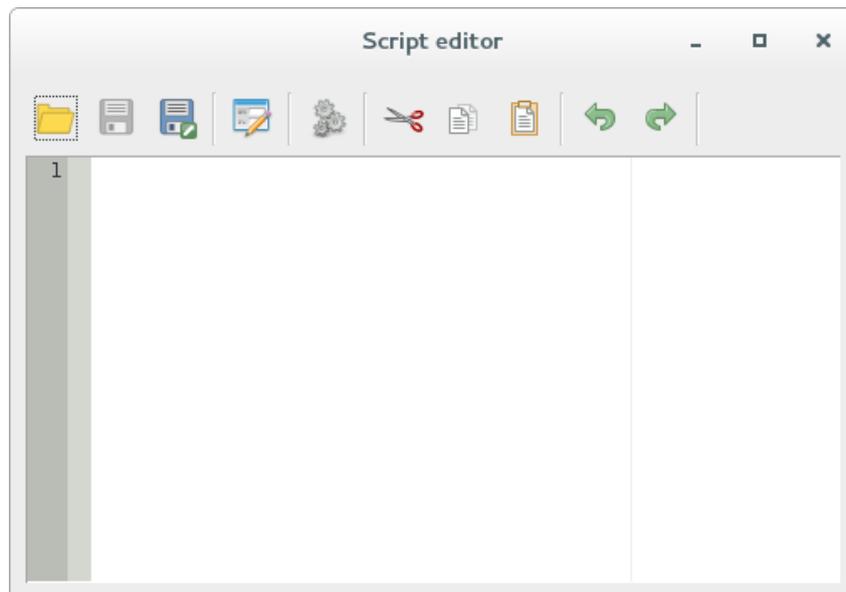
17.31.1 スクリプトの追加

スクリプトを追加するのはとても簡単です。処理ツールボックスを開き、*R -> ツール -> 新しい R の script* を作成 をクリックするだけです。



ノート: 処理の中に R が見あたらない場合は、処理 -> オプション -> Providers を有効にする必要があります

スクリプト本体を追加できる前にいくつかのパラメータを指定する必要がある スクリプトエディタウィンドウ を開きます。



17.31.2 プロットを作成します

このチュートリアルでは、ベクトル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。

r_intro.qgs QGIS プロジェクト開きます。

スクリプトのパラメータ

エディタを開いて、そ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を指定する 必要があります。

1. スクリプトを置く先となるグループの名前:

```
##plots=group
```

処理ツールボックスの プロット グループでスクリプトを見つけます。

2. (ちょうどこの例では) プロットを表示する処理を指示する必要があります:

```
##showplots
```

プロットが表示されます 処理の結果ビューア 中にこの方法で。

3. 作業しているデータの種別をして処理を伝えるためにも必要です。この例では、ベクトルレイヤのフィールドからプロットを作成したい:

```
##Layer=vector
```

処理は、入力がベクトルであることを今知っています。名前 レイヤー は重要ではありません、重要なのはベクトル パラメータです。

4. 最後に、プロットするベクトルレイヤの入力フィールドを指定する必要があります:

```
##X=Field Layer
```

だから、処理はそのフィールドレイヤが **** X **** と呼ばれていることを知っています。

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:

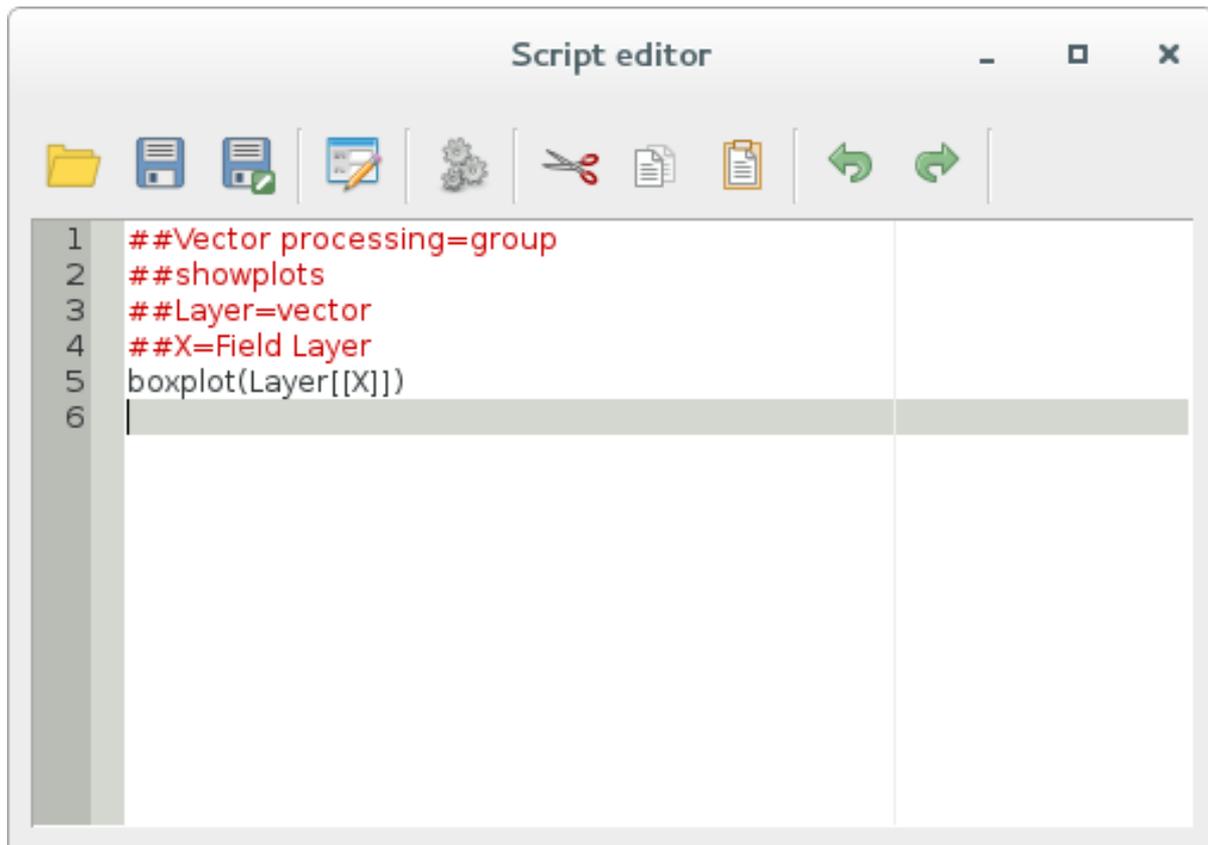
```
boxplot (Layer[[X]])
```

箱ひげは、データセットとしてレイヤをデータセットのフィールドとして X を呼び出す、R 関数自体の名前であることに注意してください。

警告: パラメータ X は、二重角括弧 [[]] の範囲内にあります

最後のスクリプトは次のようになります:

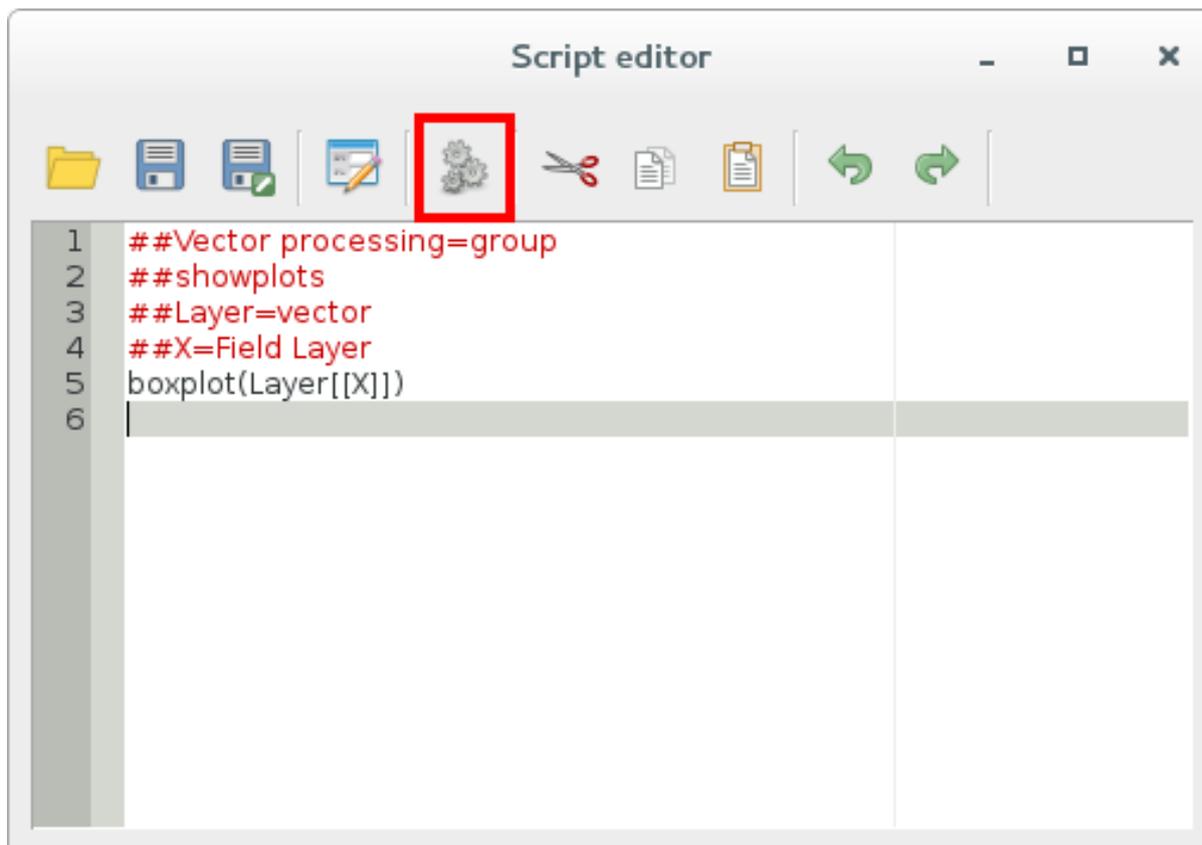
```
##Vector processing=group
##showplots
##Layer=vector
##X=Field Layer
boxplot (Layer[[X]])
```



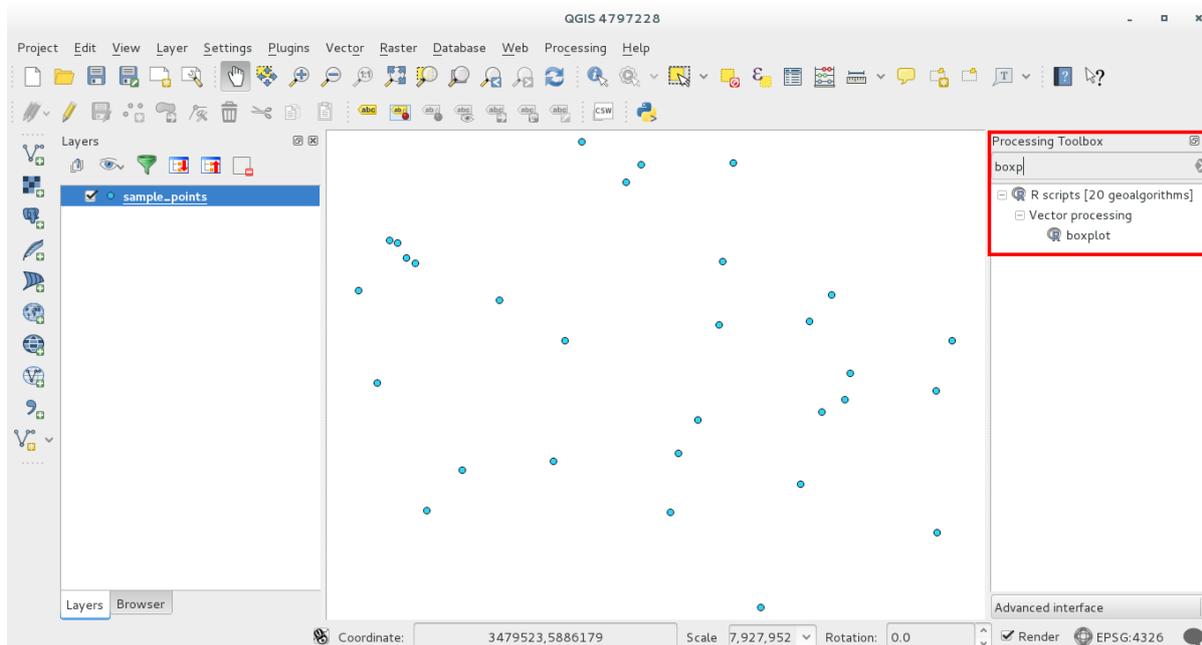
処理によって提案されたデフォルトのパスにスクリプトを保存します。選択した名前が処理ツールボックスに見つかるスクリプトの名前と同じになります。

ノート: スクリプトは他のパスに保存できますが、処理が自動的にアップロードできず、手動ですべてのスクリプトをアップロードすることが必要になります

今だけのエディタ・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。



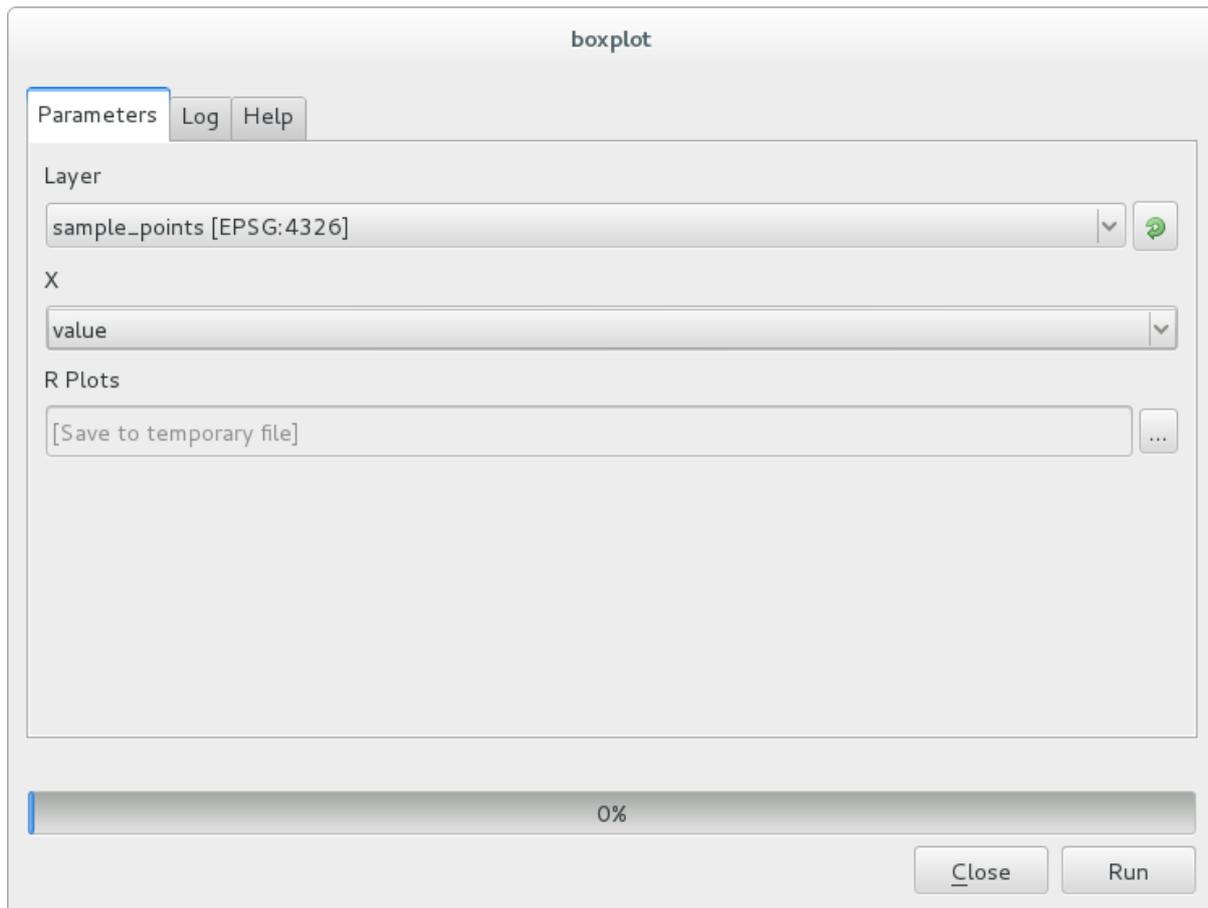
エディタウィンドウが閉じられた後、それ以外の場合は、スクリプトを見つけるために、処理のテキストボックスを使用します。



これで、処理アルゴリズムのウィンドウで必要なパラメータを記入できます：

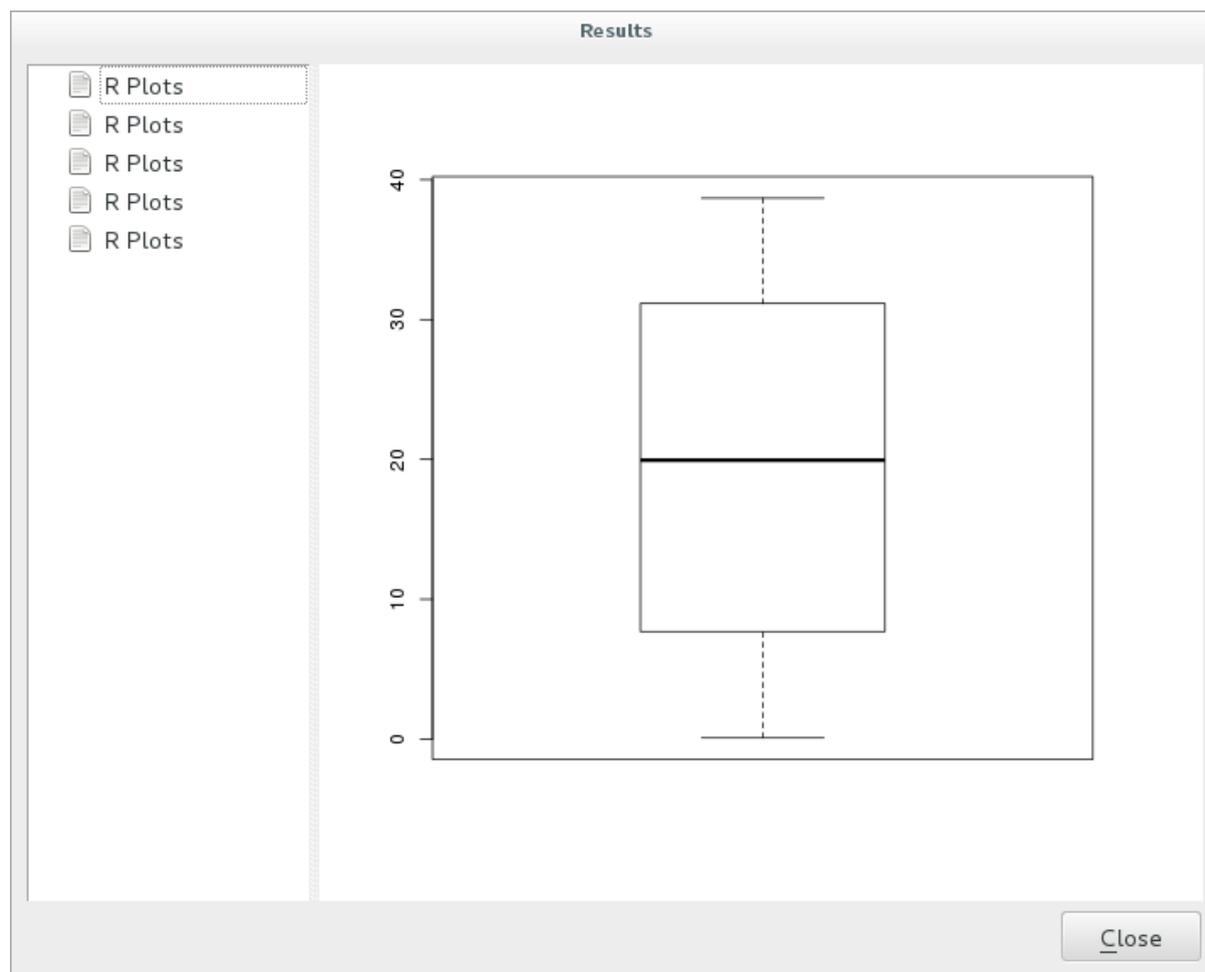
- レイヤとしてサンプル点を選択する
- X フィールドに値パラメータを記入します

[実行] をクリックしてください。



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 処理 ->結果ビューア... をクリックするだけです。

これが表示される最終的な結果です。



ノート: プロットを右クリックして画像を開き、コピーし、保存できます

17.31.3 ベクトルを作成します。

R スクリプトを使用すると、ベクトルも作成できますし、自動的に QGIS でそれをロードします。

R -> ツール -> オンライン *collection* から R スクリプトをダウンロード 次の例では、オンラインコレクションからダウンロードできます ランダムサンプリング *grid* スクリプトからとられています。

この演習の目的は、*sp* パッケージの *spsample* 機能を使用してレイヤの範囲内のランダムな点ベクトルを作成することです。

スクリプトのパラメータ

前と同じように、スクリプト本体の前いくつかのパラメータを設定する必要があります。

1. スクリプトを置く先となるグループの名前を指定します、例えば ポイントパターン分析

```
##Point pattern analysis=group
```

2. ランダムな点が含まれているレイヤを設定する:

```
##Layer=vector
```

3. 作成しようとしているポイントの数を設定する:

```
##Size=number 10
```

ノート: 10 は、デフォルト値になるだろう。この番号を変更したり、デフォルトの番号なしでパラメータを残すことができます

- 出力はベクトルレイヤで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。

- spsample 機能を実行します:

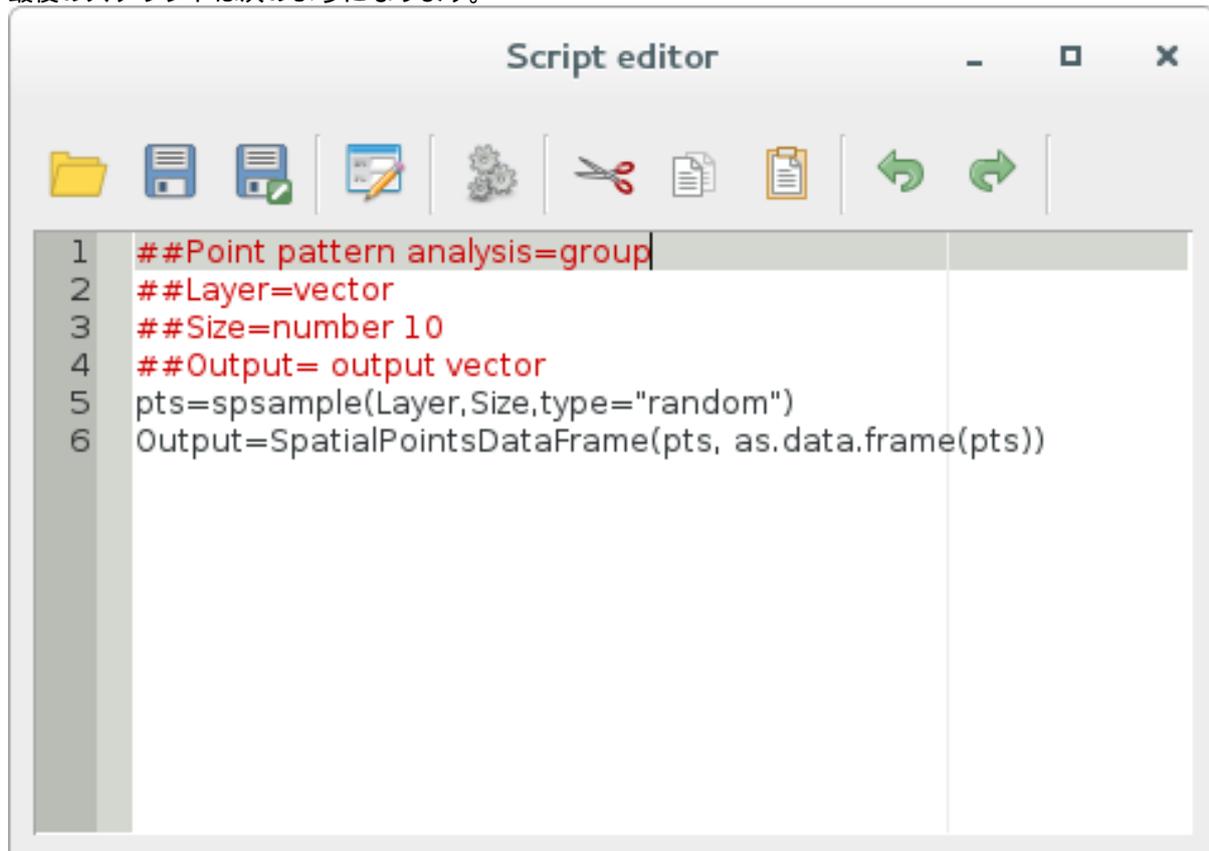
```
pts=spsample(Layer,Size,type="random")
```

このように関数は点の数は サイズ パラメータから取得され、点発生が ランダム であり、レイヤーの範囲を取り

- 出力のパラメータを含む行を書く:

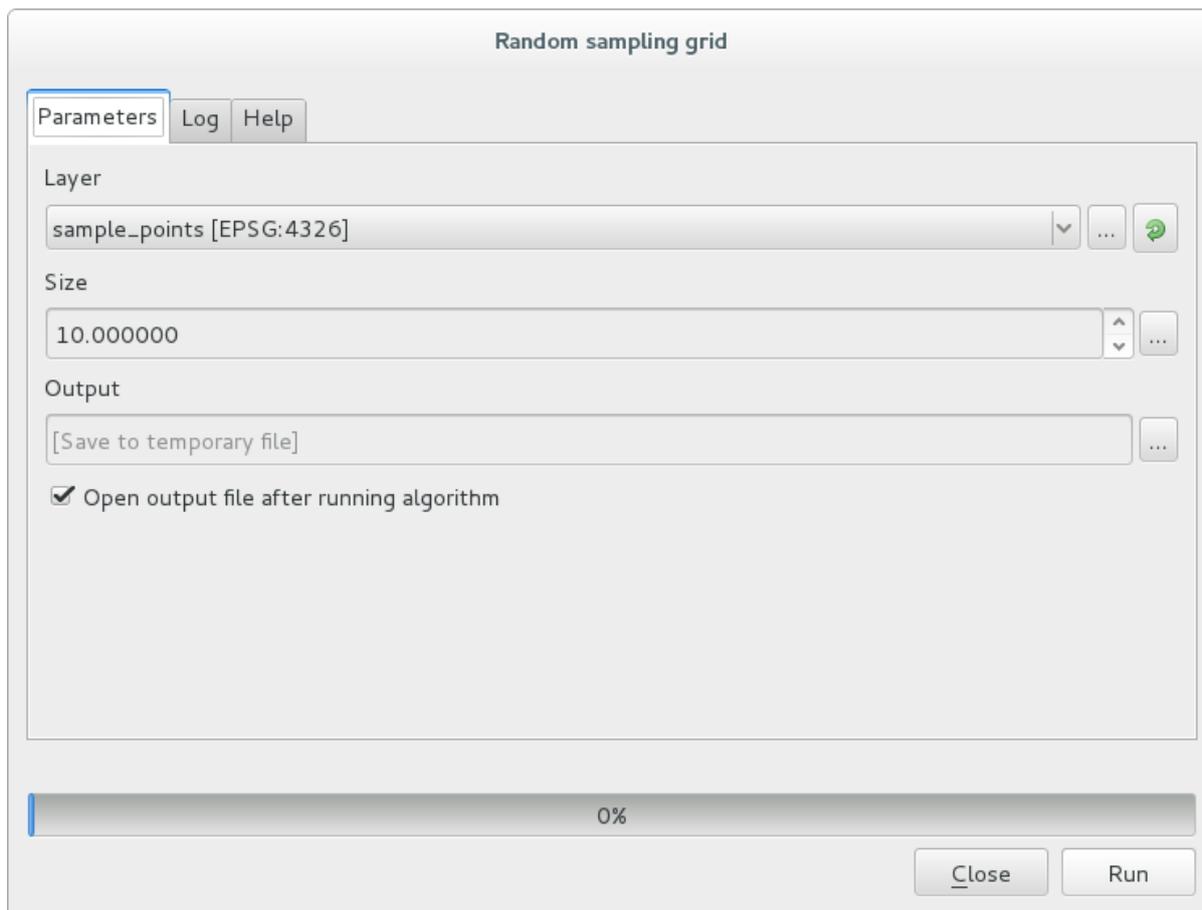
```
Output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```

最後のスクリプトは次のようになります。



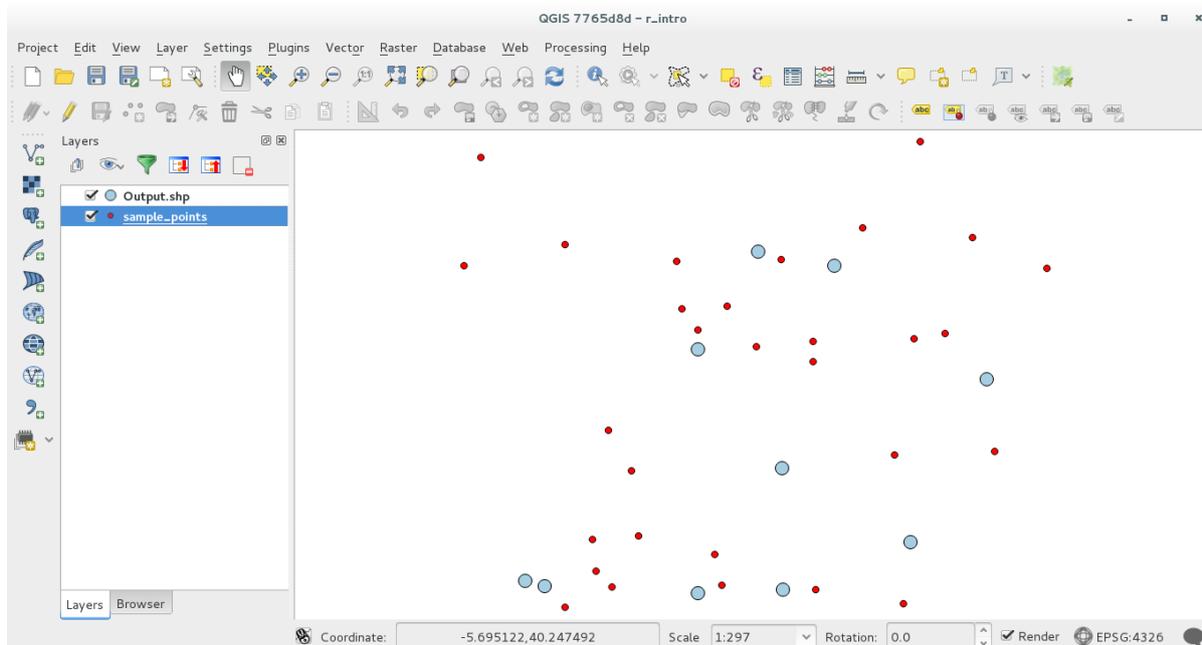
それを保存し、それを実行し、実行しているボタンをクリックします。

右のパラメータの新しいウィンドウタイプで :



し、実行をクリックしてください。

結果のポイントが、マップのキャンバスに表示されます



17.31.4 R - 処理構文

処理は、R のうち、成果を得るために、いくつかの特別な構文を使用していることに注意してください：

- > コマンドの前に、のように “> lillie.test (レイヤー [[フィールド]]) “結果は R 出力に送られるべき手段 (結果ビューア)
- + プロットした後、オーバーレイプロットを呼び出します。例えば、 “プロット (レイヤ [X]、レイヤ [Y]) + abline (H =平均 (レイヤ [X])) “

17.32 処理スクリプトで R の構文

Module contributed by Matteo Ghetta - funded by [Scuola Superiore Sant'Anna](#)

[処理] で R スクリプトを記述するために採用されている構文はかなりトリッキーである可能性があります。各スクリプトは “##“ で始まる 入力 と 出力 で始まります。

17.32.1 入力

teh の入力を指定する前に、スクリプトが置かれるているアルゴリズムのグループを設定することもできます。グループがすでに存在する場合は、アルゴリズムが自動的に作成される他、他の新しいグループに追加されます。

1. group creation, ##My Group=group

それから、すべての入力タイプと最終的には追加のパラメータを指定する必要があります。異なる入力にすることができます。

1. ベクトル、 “ ##層= vector“
2. ベクトルフィールド、 “ ## F=フィールド Layer“ (レイヤは、入力レイヤの名前です)
3. テーブル、 “ ##層= raster“
4. number, ##Num = number
5. string, ##Str = string
6. boolean, ##Bol = boolean

また、必要なすべてのパラメータを使用してドロップダウンメニューを持つことができます。項目は “半列に分離されなければなりません; “ :

7. ##type=selection point;lines;point+lines

17.32.2 出力

入力のように、各出力は、スクリプトの先頭で定義する必要があります。

1. ベクター、 “ ##の出力=出力 vector“
2. ラスタ、 “ ##の出力=出力 raster“
3. テーブル、 “ ##の出力=出力 TABLE“
4. プロット、 “ ## showplots“
5. 表示したいだけ入れ*結果ビューアー*の R 出力、 **スクリプト**内部の “> “ **前**出力

17.32.3 スクリプト本体

スクリプト本体は R 形式の構文に従い、ログ パネルは何かスクリプトで間違っていた場合に助けになります。

スクリプトでは、すべての追加のライブラリをロードする必要があることを忘れないでください

```
library(sp)
```

ベクトル出力の例

入力レイヤの範囲からランダムな点を作成するオンラインコレクションからのアルゴリズムを見てみましょう:

```
##Point pattern analysis=group
##Layer=vector
##Size=number 10
##Output= output vector
library(sp)
pts=spsample(Layer,Size,type="random")
Output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```

そして、線を介して取得します :

1. ポイントパターン分析 は、アルゴリズムのグループ
2. “Layer“ は入力 ベクトル レイヤ
3. “Size“ は 10 のデフォルト値を持つ 数値 パラメータ、
4. Output はアルゴリズムによって作成される ベクトル レイヤ、
5. “ライブラリー (sp)“ は sp ライブラリをロード (既にコンピュータにインストールされている必要があります、そのインストールは R 内で されている必要があります)
6. sp ライブラリの spsample 関数を呼び出し、上記で定義されたすべての入力に渡します
7. “SpatialPointsDataFrame“ 機能で出力ベクトルを作成します

それでおしまい! QGIS 凡例中に持っているベクタレイヤでアルゴリズムを実行し、多数のランダムポイントを選択するだけで、QGIS 地図キャンパスにそれらが得られるでしょう。

ラスタ出力の例

次のスクリプトは、基本的な通常のクリギングを実行し、補間値のラスタ地図を作成します:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##Output=output raster
require("automap")
require("sp")
require("raster")
table=as.data.frame(Layer)
coordinates(table)= ~coords.x1+coords.x2
c = Layer[[Field]]
kriging_result = autoKrige(c~1, table)
prediction = raster(kriging_result$krige_output)
Output<-prediction
```

入力中のベクトルおよびそのフィールドからアルゴリズムは、automap の R パッケージの autoKrige 関数を使用し、最初にクリギングモデルを計算し、次にラスタを作成します。

ラスタは、ラスタ R パッケージの raster 機能を使用して作成されます。

テーブル出力の例

出力は、テーブルファイル (CSV) になるようにのが “概要 Statistics“ アルゴリズムを編集してみましょう。スクリプトの本文は以下の通りです:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##Stat=Output table
Summary_statistics<-data.frame(rbind(
sum(Layer[[Field]]),
length(Layer[[Field]]),
length(unique(Layer[[Field]])),
min(Layer[[Field]]),
max(Layer[[Field]]),
max(Layer[[Field]]-min(Layer[[Field]]),
mean(Layer[[Field]]),
median(Layer[[Field]]),
sd(Layer[[Field]]),row.names=c("Sum:", "Count:", "Unique values:", "Minimum value:", "Maximum value:
colnames(Summary_statistics)<-c(Field)
Stat<-Summary_statistics
```

3行目は入力にベクトル場を指定し、4行目は出力テーブルであるべきであるアルゴリズムを伝えます。最後の行は、スクリプトで作成された “Stat” オブジェクトを取得し、“csv” テーブルに変換します。

コンソール出力の例

前の例を取り、テーブルを作成する代わりに、結果ビューアで結果を印刷できます:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
Summary_statistics<-data.frame(rbind(
sum(Layer[[Field]]),
length(Layer[[Field]]),
length(unique(Layer[[Field]])),
min(Layer[[Field]]),
max(Layer[[Field]]),
max(Layer[[Field]]-min(Layer[[Field]]),
mean(Layer[[Field]]),
median(Layer[[Field]]),
sd(Layer[[Field]]),row.names=c("Sum:", "Count:", "Unique values:", "Minimum value:", "Maximum value:
colnames(Summary_statistics)<-c(Field)
>Summary_statistics
```

スクリプトはわずか2箇所編集する以外は上記とまったく同じです。

1. 出力はもはや指定されない (第4行は除去されました)
2. 最後の行は、“>” “それは結果ビューアでオブジェクトを印刷するための処理を指示することから始まります

プロットと例

プロットを作成することは非常に簡単です。次のスクリプトに示すように、“## showplots” パラメータを使用する必要があります:

```
##Basic statistics=group
##Layer=vector
##Field=Field Layer
##showplots
qqnorm(Layer[[Field]])
qqline(Layer[[Field]])
```

スクリプトは入力にベクトルレイヤのフィールドを取り、分布の正規性をテストするために QQ プロットを作成します。

プロットは自動的に処理の結果ビューアーに追加されます。

17.33 処理のための R 構文の概要表

Module contributed by Matteo Ghetta - funded by [Scuola Superiore Sant'Anna](#)

処理は、スクリプト本体で使うことができる、多くの異なった入力と出力パラメータを可能にします。これは概要表です：

17.33.1 入力パラメータ

パラメータ	構文の例	オブジェクトを返します
ベクタ	Layer = vector	SpatialDataFrame object, default object of <code>rgdal</code> package
ベクトルポイント	レイヤー=ベクトルポイント	SpatialPointDataFrame object, default object of <code>rgdal</code> package
ベクトルライン	Layer = vector line	SpatialLineDataFrame object, default object of <code>rgdal</code> package
vector polygon	Layer = vector polygon	SpatialPolygonsDataFrame object, default object of <code>rgdal</code> package
multiple vector table	Layer = multiple vector Layer = table	SpatialDataFrame objects, default object of <code>rgdal</code> package dataframe conversion from csv, default object of <code>read.csv</code> function
field	Field = Field Layer	name of the Field selected, e.g. "Area"
raster	Layer = raster	RasterBrick object, default object of <code>raster</code> package
multiple raster number	Layer = multiple raster	RasterBrick objects, default object of <code>raster</code> package
string	N = number	integer or floating number chosen
longstring	S = string	string added in the box
selection	LS = longstring S = selection	string added in the box, could be longer than the normal string string of the selected item chosen in the dropdown menu
crs	first;second;third C = crs	string of the resulting CRS chosen, in the format: "EPSG:4326"
範囲	E = extent	Extent object of the <code>raster</code> package, you can extract values as <code>E@xmin</code>
point	P = point	地図をクリックすると点の座標が得られます
ファイル	F = file	path of the file chosen, e.g. "/home/matteo/file.txt"
folder	F = folder	path of the folder chosen, e.g. "/home/matteo/Downloads"

入力のいずれかがオプションにもでき、それはこのパラメータを無視するようにスクリプトに指示するための便利な方法を持っていることを意味します。

入力をオプションとして設定するためには、その入力の前に文字列 `optional` を追加する必要があります。例えば:

```
##Layer = vector
##Field1 = Field Layer
##Field2 = optional Field Layer
```

17.33.2 Output parameters

出力パラメータは、スクリプトの先頭で与えた入力名を取ると、したいオブジェクトを書き込みます。

パラメータ	構文の例
ベクタ	Output = output vector
raster	Output = output raster
table	Output = output table
ファイル	Output = output file

ノート: プロット入力タイプのために、処理結果ビューアー から直接 “png”としてプロットを保存できません。また、アルゴリズムのインターフェイスから直接プロットを保存するように選択できません

17.33.3 Examples

入力と出力のパラメータすべてをより理解するためには、[R Syntax chapter](#) をご覧ください。

17.34 地滑りを予測

Module contributed by Paolo Cavallini - [Faunalia](#)

ノート: この章では、地滑りの可能性を予測するために単純化しすぎの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、興味のある読者は、出力の差分を計算できます)。

- GRASS → *r.slope*
- SAGA → *Slope, Aspect, Curvature*
- GDAL *Slope*

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斜面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を使用します) (降雨*スロープ) / 100 としましょう。

- SAGA → ラスター計算機 *rain, slope: (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 (関心のパラメータは最大と平均)。

Chapter 18

Module: QGIS での空間データベースの利用

このモジュールでは、クエリによって分析を実行するだけでなく、データベース内のデータを管理、表示、そして操作するために QGIS で空間データベースを使用する方法について学習します。我々は主に PostgreSQL と PostGIS (前のセクションで説明) を使用しますが、spatialite を含む他の空間データベースの実装でも同じ概念が適用可能です。

18.1 Lesson: QGIS ブラウザーにおけるデータベースについて

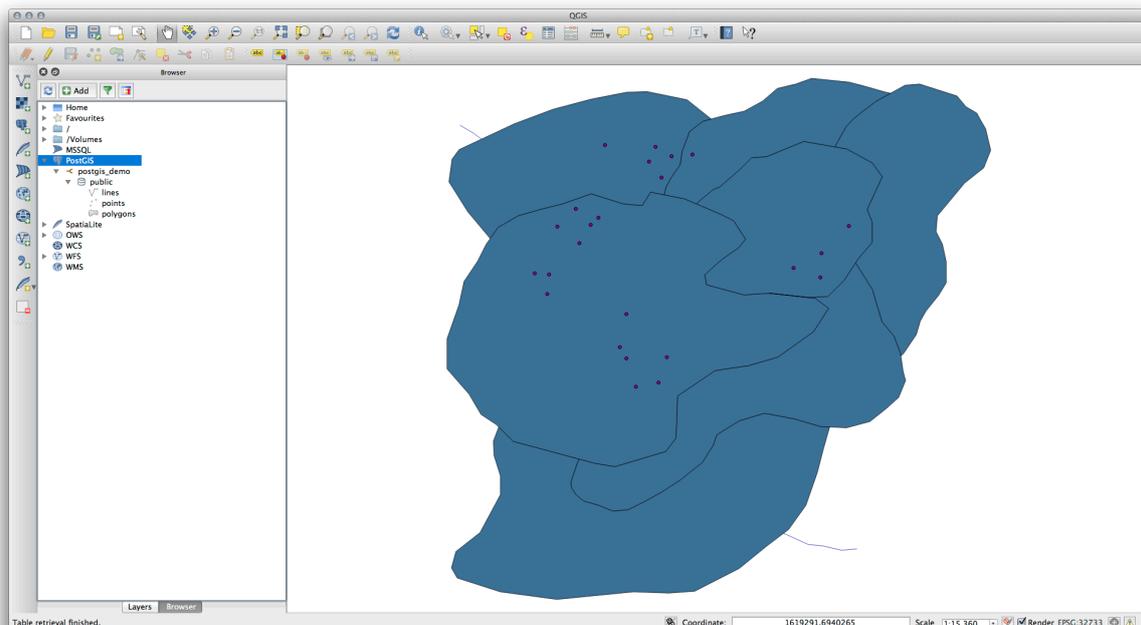
前の 2 つのモジュールでは、基本的な概念、特徴や機能リレーショナルデータベースだけでなく、私たちは、保存、管理、クエリとリレーショナルデータベース内の空間データを操作できる機能拡張を見ました。このセクションでは、効果的に QGIS で空間データベースを使用する方法に深く潜ります。

この章のゴール QGIS のブラウザー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ぶ。

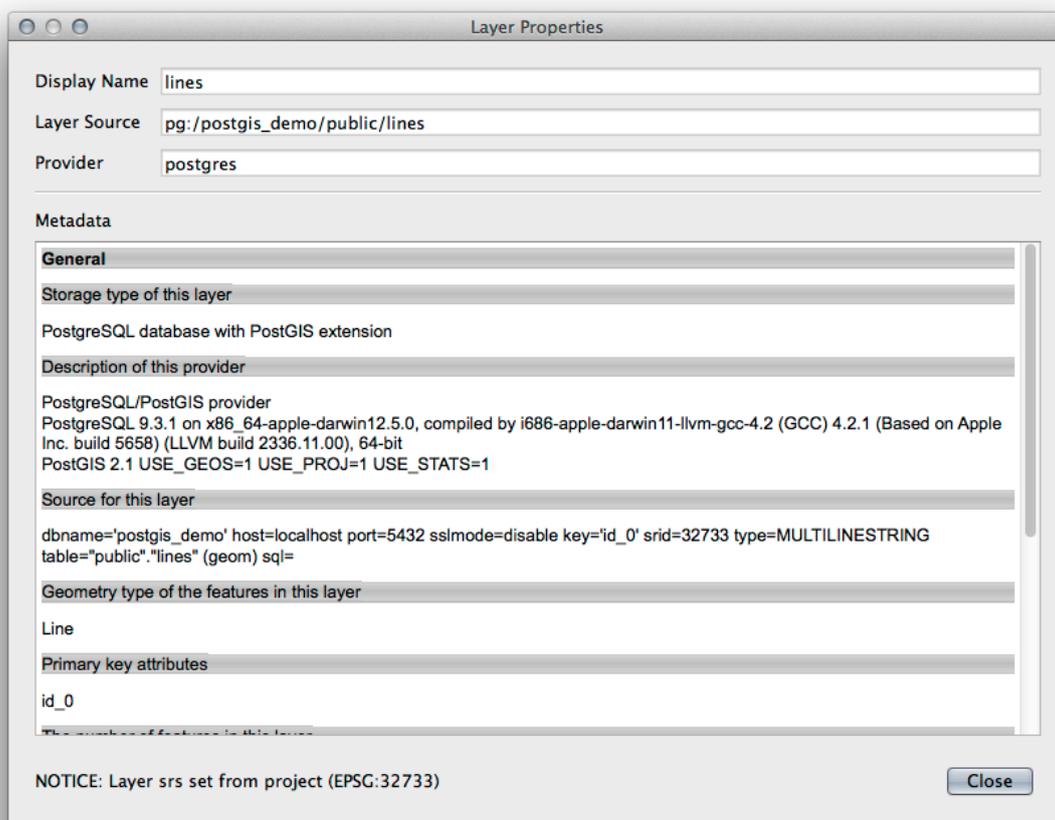
18.1.1 Follow Along: ブラウザを使用して QGIS にデータベーステーブルを追加します

すでに QGIS レイヤー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 で新しい空のマップを開きます。
- レイヤーパネルの下部にある *Browser* タブをクリックしてブラウザを開きます
- 木の PostGIS の部分を開きます。前に設定した接続が可能なはずです (ブラウザウィンドウの上部にある [更新] ボタンをクリックする必要があるかもしれません)。



- 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、それが地図キャンパスに追加されます。
- 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。レイヤのプロパティを見るために *Properties* 項目をクリックしてください。



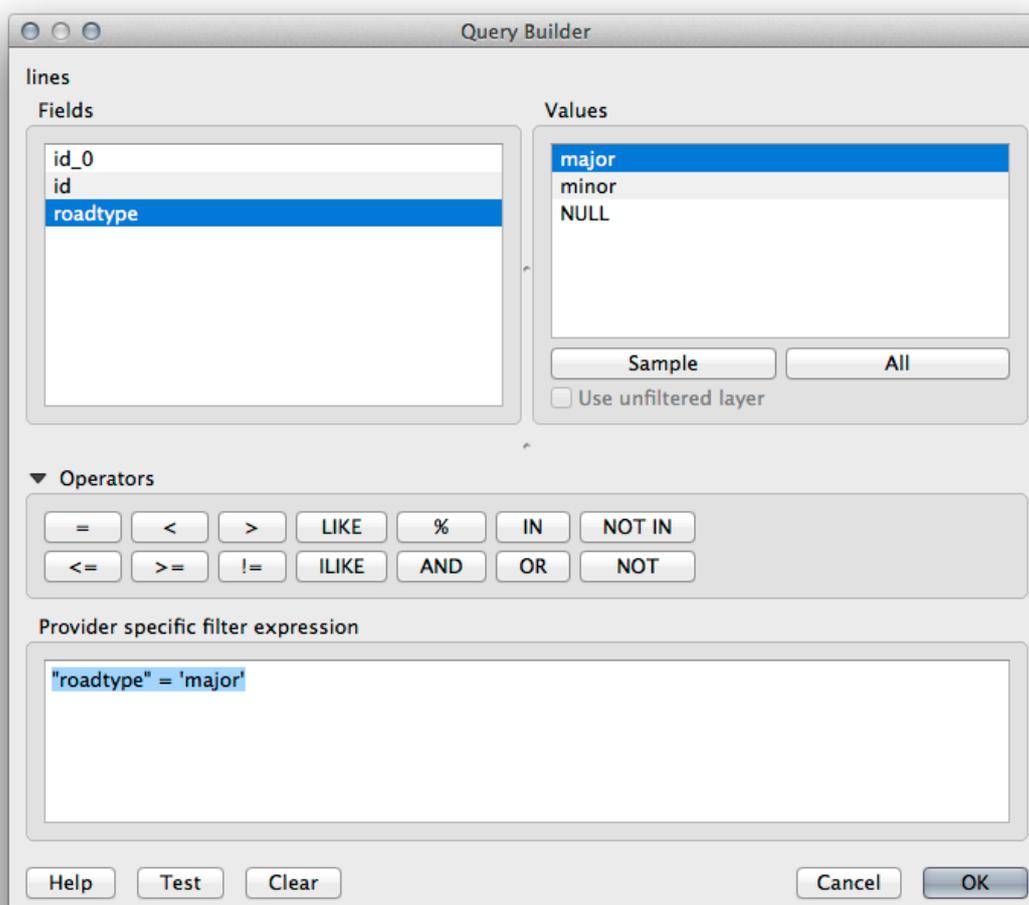
ノート: もちろん、自分のワークステーションへの外部サーバーでホストされているの PostGIS データベースに接続するにも、このインターフェイスを使用できます。ツリー内の PostGIS のエントリを右クリックすると、新しい接続の接続 PARAMETERS を指定できます。

18.1.2 Follow Along: レイヤーとして、レコードのフィルタセットを追加します

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マップを開始
- *PostGIS* レイヤーを追加 ボタンをクリックするか、メニューから レイヤー ->*PostGIS* レイヤーを追加 を選択します。
- 表示される *PostGIS* のテーブルを追加 ダイアログで、 `postgis_demo` 接続に接続します。
- 公共 `schema` を展開し、以前に使用していた 3 つのテーブルを見つける必要があります。
- `lines` レイヤーをクリックして選択しますが、それを追加する代わりに、 *Filter* を設定 ボタンをクリックして、クエリ *Builder* ダイアログを起動してください。
- 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。
- ツリー内で *lines* レイヤーの名前を *roads_primary* に変更します。

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 In Conclusion

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 What's Next?

次はデータベース管理タスクのより完全なセットのために QGIS での DB マネージャインターフェースで作業する方法について説明します。

18.2 Lesson: Using DB Manager to work with Spatial Databases in QGIS

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 DB Manager 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。

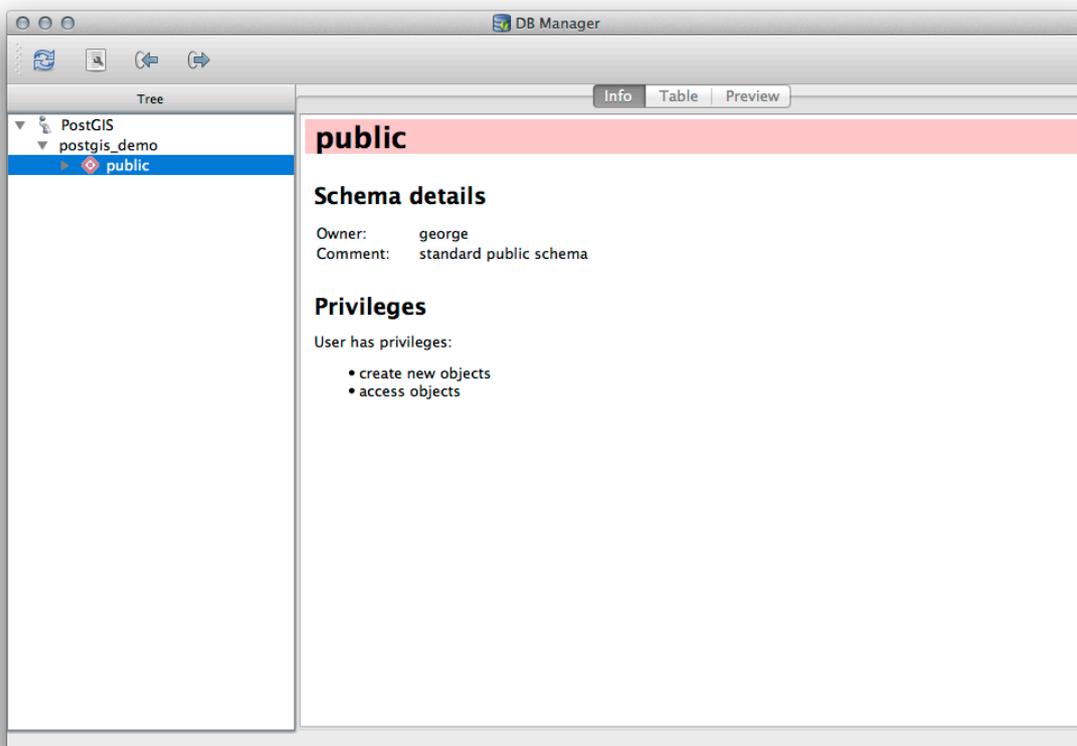
18.2.1 Follow Along: DB Manager での PostGIS データベースの管理

最初に DB マネージャのインター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。



すでに設定した以前の接続が見えているはずであり、前のセクションで作業したテーブルを見るために myPG セクションとその public スキーマを展開できるはずです。

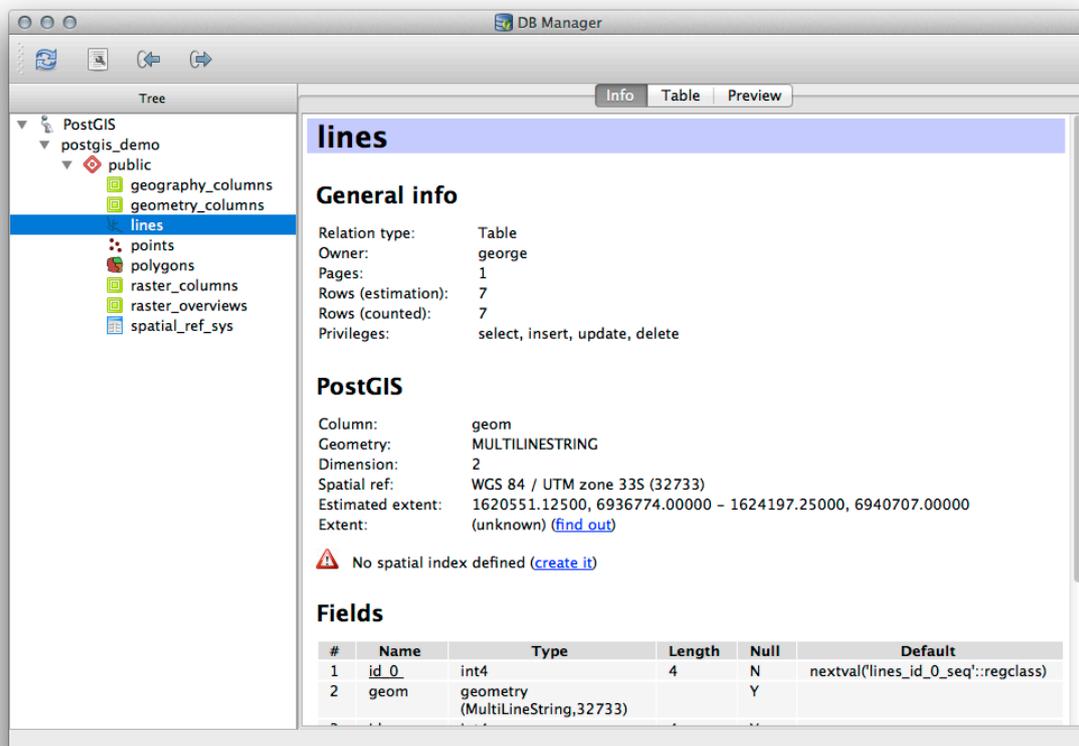
気づくかもしれない最初の事は、今データベースに含まれるスキーマに関するいくつかのメタデータを見ることができることです。



スキーマは、データテーブルおよび PostgreSQL データベース内の他のオブジェクトと権限および他の制約のコンテナをグループ化する方法です。PostgreSQL のスキーマを管理することはこのマニュアルの範囲外ですが、それらについての詳細な情報は 'スキーマに関する PostgreSQL のドキュメント <<http://www.postgresql.org/docs/9.1/static/ddl-schemas.html>> ' _に見つけることができます。DB Manager を使用して新しいスキーマを作成できますが、それらを効果的に管理するには pgAdmin III のようなツールやコマンドライン・インタフェースを使用する必要があります。

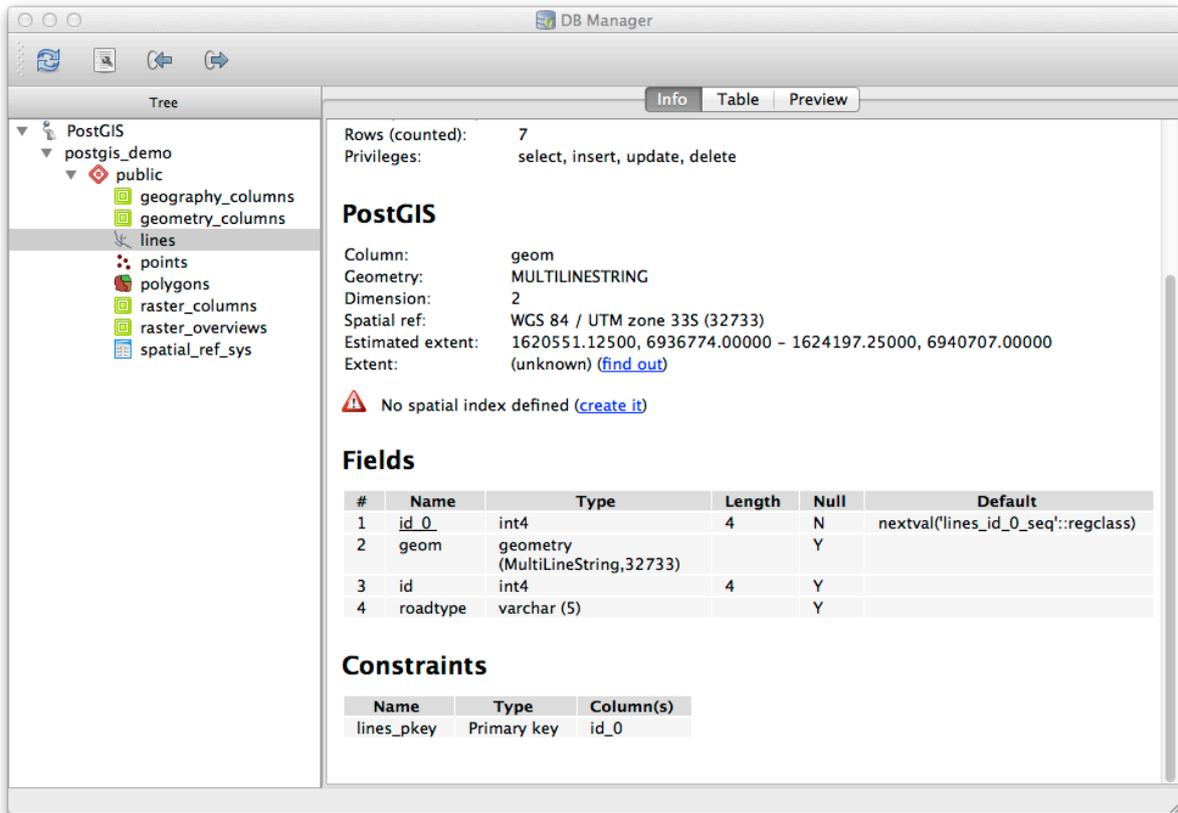
DB Manager は、データベース内のテーブルを管理するために使用できます。すでに、コマンドライン上でテーブルを作成し、管理するためのさまざまな方法を見て、今は DB マネージャでこれを行う方法を見てみましょう。

まず、ツリー内の名前をクリックし *Info* タブの中を見ることで、テーブルのメタデータをただ見ることは有用です。

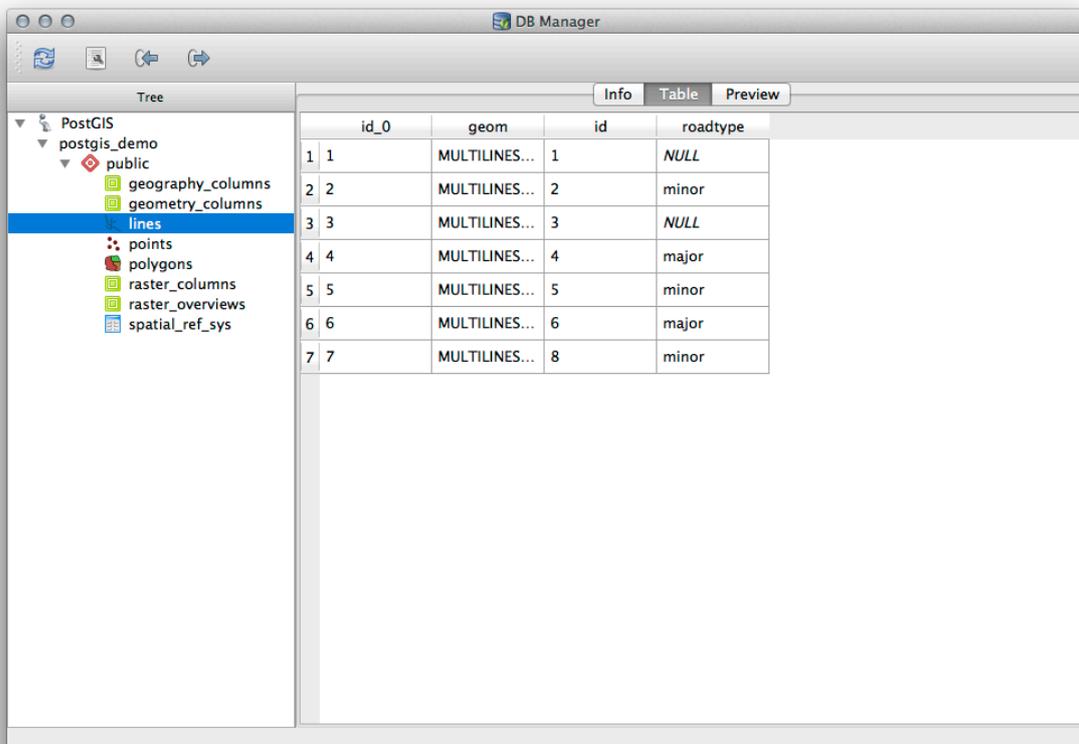


このパネルでは、テーブルについての一般 *Info* も、幾何学と空間参照システムに関して PostGIS の拡張子が維持している情報を見ることができます。

Info 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、*fields*、*Constraints*、*Indexes* についての詳細な情報を見ることができます。



単にレイヤー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 Manager を使用することも非常に便利。 *Table* タブを選択して、データを閲覧できます。

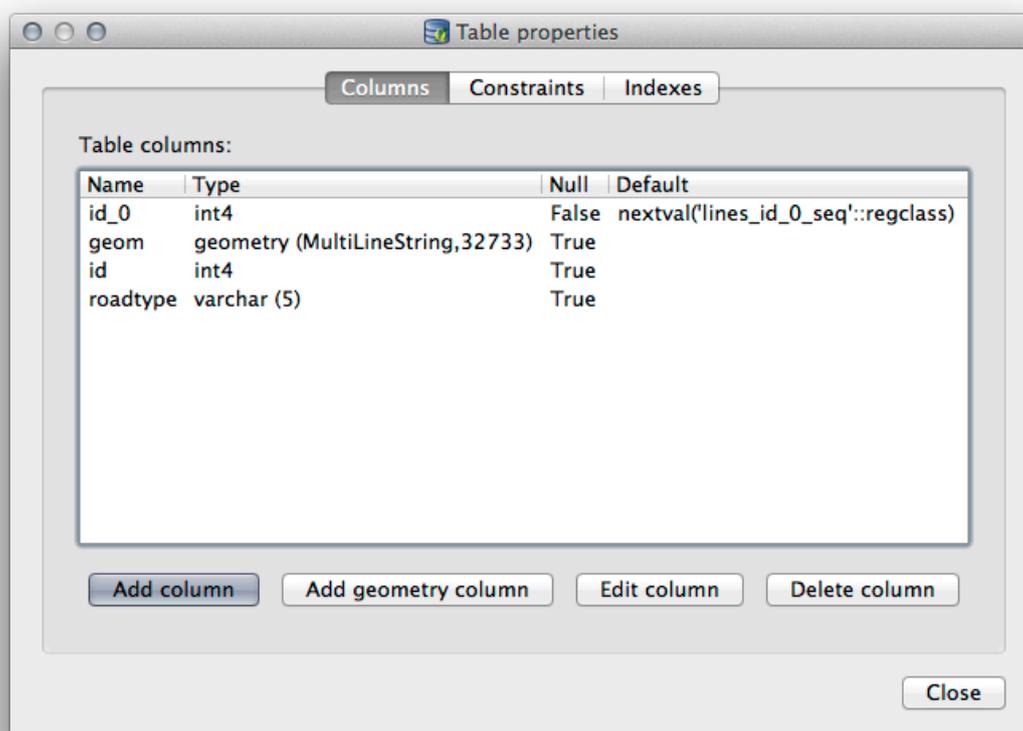


地図プレビューでレイヤーのデータが表示される *Preview* タブもあります。

ツリー中のレイヤーを右クリックし、キャンバスに追加 をクリックすると地図にこのレイヤーが追加されます。

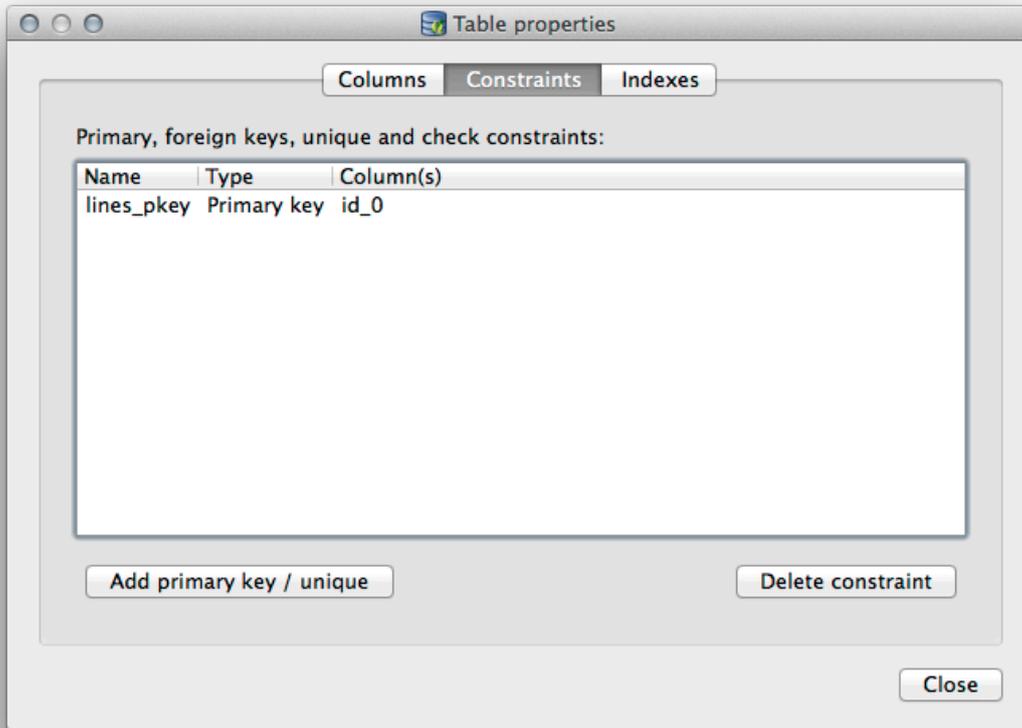
これまでのところは唯一のデータベースのスキーマとテーブルとそのメタデータを閲覧されているが、おそらく追加の列を追加するテーブルを変更したい場合はどう? DB Manager は、直接これを行うことができます。

- ツリーで、編集するテーブルを選択します
- メニューから 表->表を編集 を選択し 表のプロパティ ダイアログを開きます。

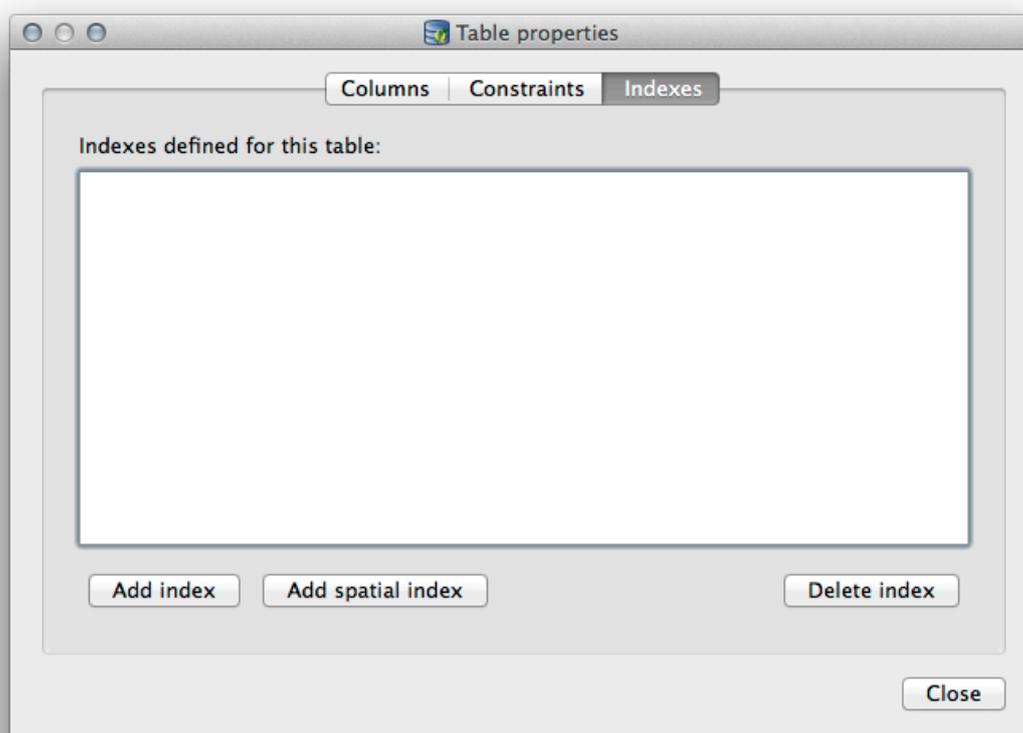


列を追加するには、このダイアログを使用してジオメトリ列を追加し、既存の列を編集したり、列を完全に削除できます。

Constraints 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。



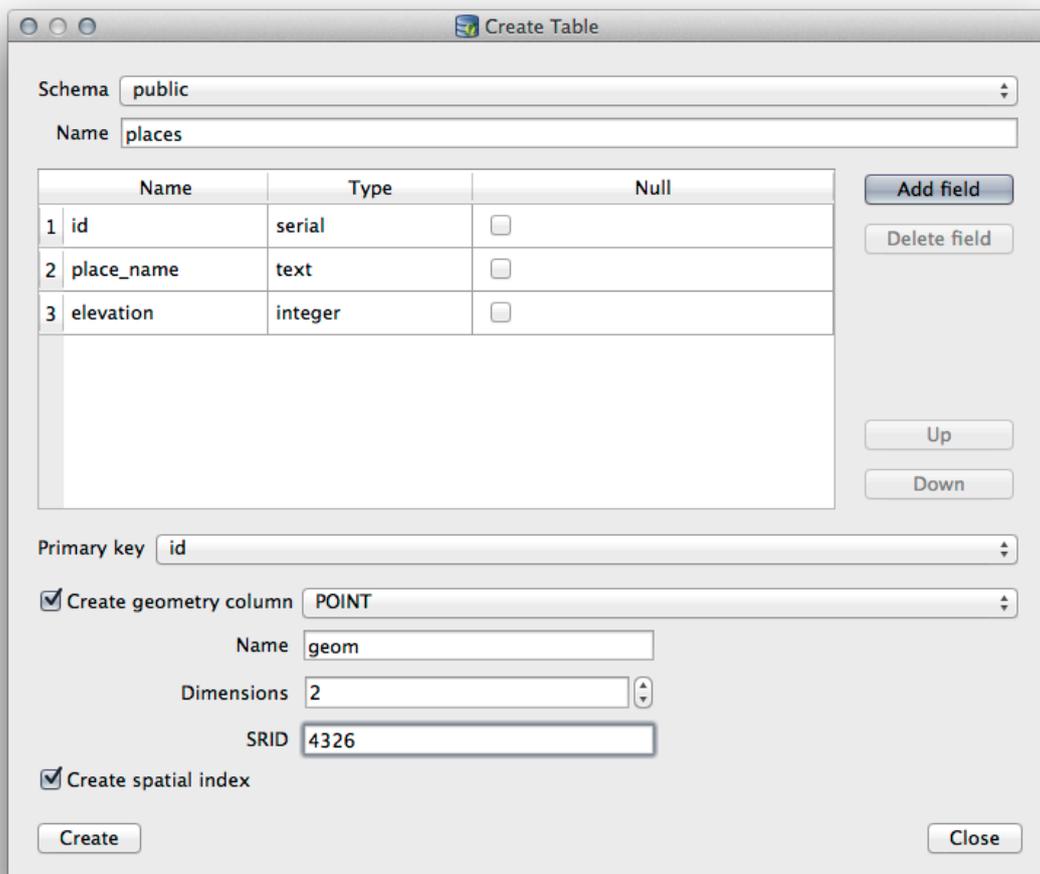
Indexes タブは、空間インデックスと通常のインデックスの両方を追加および削除するために使用できます。



18.2.2 Follow Along: 新しいテーブルを作成する

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。

- 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、すでにデータベース内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。
- メニューから 表 -> Table を作成 を選択し、[表の作成] ダイアログを開きます。
- デフォルトの Public スキーマを使用し、テーブル places に名前を付けてください。
- id、place_name、および elevation フィールドを以下に示すように追加します
- 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てください。
- ジオメトリ COLUMN を作成 の横にあるチェックボックスをクリックし、それが POINT のタイプと設定されていることを確認し、名前は geom のまま、SRID として 4326 を指定します。
- 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。



- 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、*Close* をクリックして「表を作成」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。ここからレイヤーのメニューで 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 Follow Along: 基本的なデータベース管理

DB マネージャによって、いくつかの基本的なデータベース管理タスクを実行できるようにもなります。それは確かにより完全なデータベース管理ツールに代わるものではありませんが、それはデータベースを維持するために使用でき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でき、頻繁に変更されているテーブルは、もはや PostgreSQL で必要とされていないレコードの残党を中心に残して終わることがあります。VACUUM コマンドでは、圧縮し、オプションのパフォーマンス向上のために、テーブルを分析するためのガベージコレクションのようなものを行ってくれます。

DB Manager 内から `VACUUM ANALYZE` コマンドを実行する方法を見てみましょう。

- DB マネージャツリーでテーブルのいずれかを選択します。
- メニューから 表 ->実行してくださいバキューム *Analyze* 選択します。

それでおしまい！PostgreSQL は操作を実行します。テーブルの大きさに応じて、これは完了するまでに時間がかかる場合があります。

‘ANALYZE VACUUM についての PostgreSQL のドキュメント<<http://www.postgresql.org/docs/9.1/static/sql-vacuum.html>> ‘_ANALYZE VACUUM の処理に関する詳細な情報を見つけることができます。

18.2.4 Follow Along: DB Manager と SQL クエリの実行

DB Manager 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。すでにこの種の機能見てきた *Browser* パネルを、しかし、DB マネージャで再びここでそれを見てみましょう。

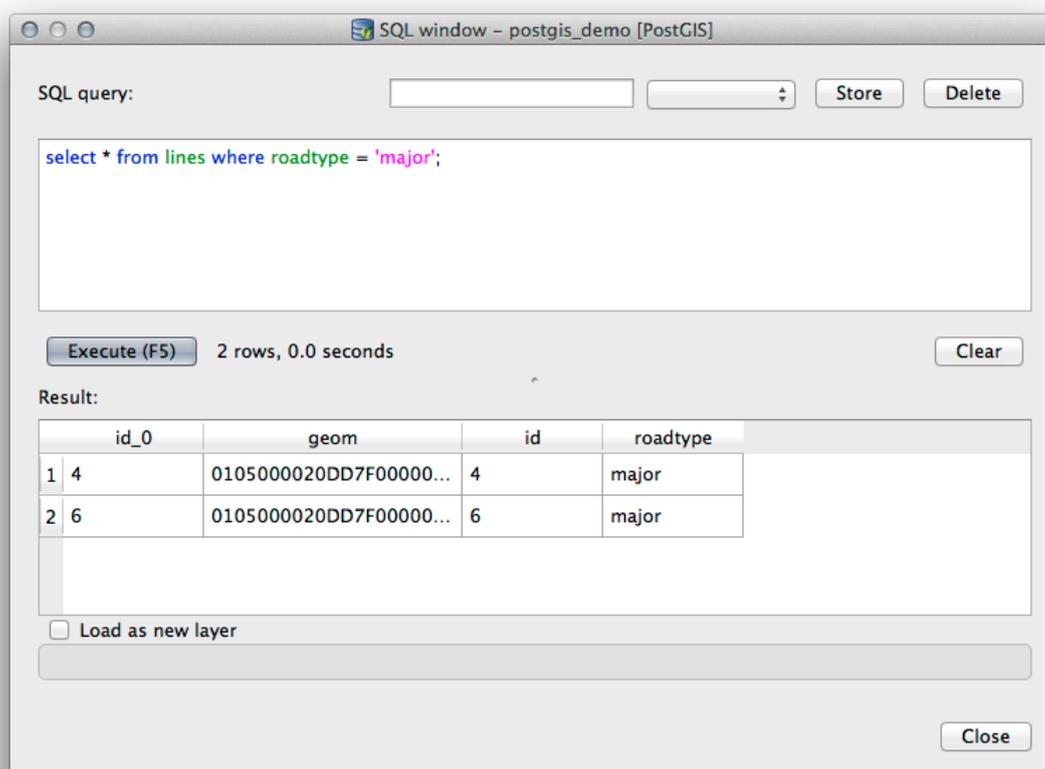
- ツリー内で *lines* テーブルを選択してください。
- DB Manager ツールバーで *SQL* の *window* ボタンを選択します。



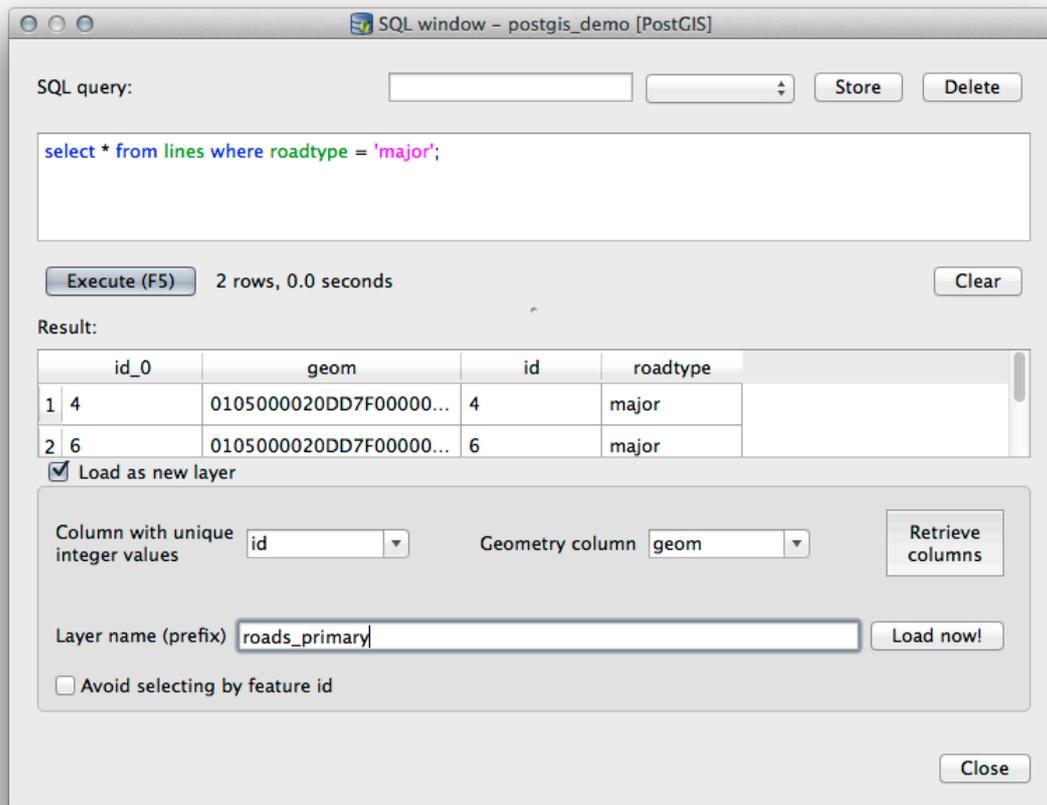
- 次の *SQL* クエリ を与えられたスペースに構成します

```
select * from lines where roadtype = 'major';
```

- クエリを実行するには、実行 (*F5*) ボタンをクリックします。
- *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。



- 新しい *layer* としてロード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。
- *id* 列を一意の整数値を持つ列として、*geom* 列をジオメトリ *COLUMN* として選択します。
- レイヤ名 (接頭辞) を *roads_primary* と入力します。
- 今すぐロード! をクリックして地図に新しいレイヤーとして結果をロードします。



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。

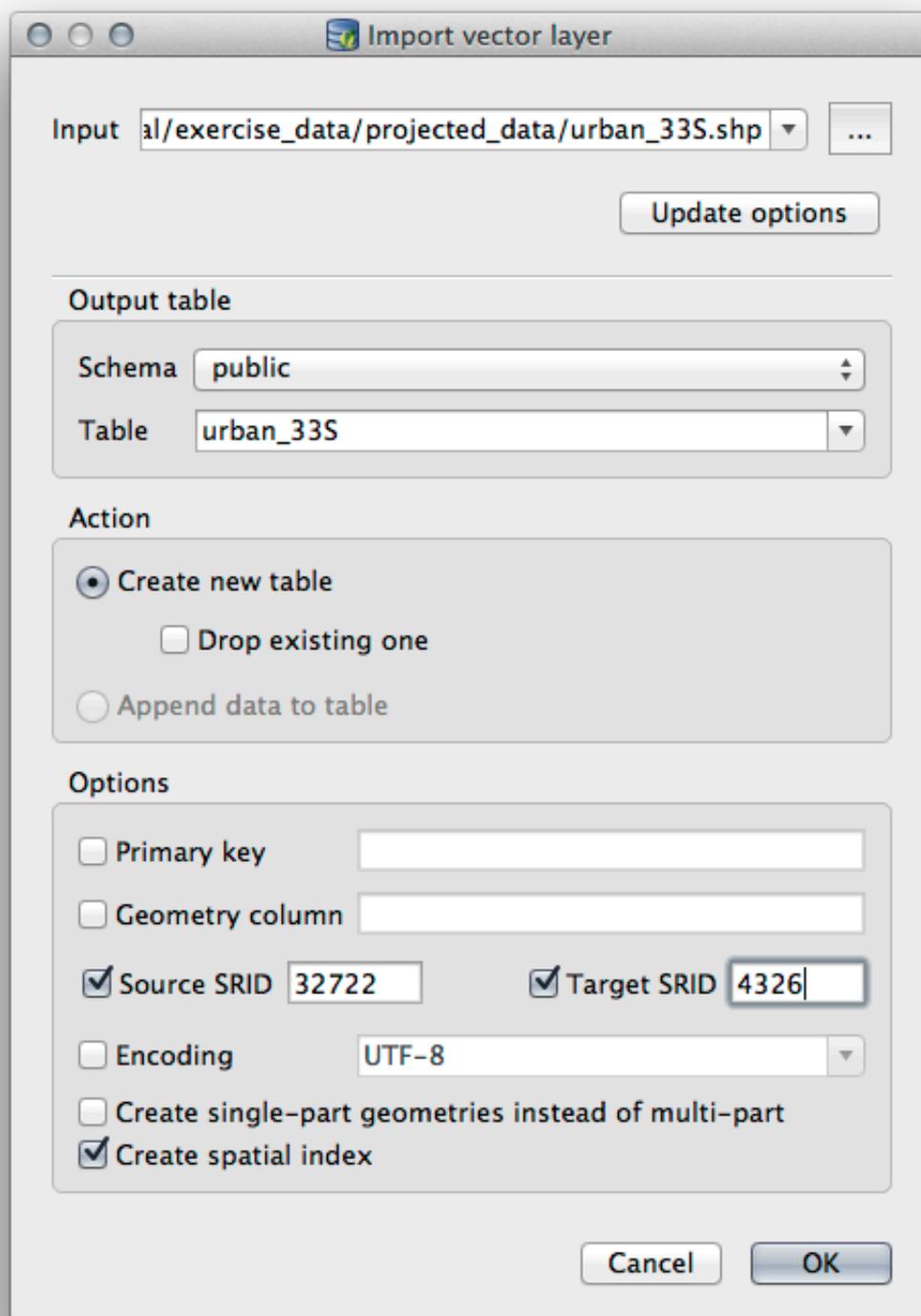
18.2.5 DB マネージャを使用したデータベースへのデータのインポート

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読み込みする方法はすでに見てきており、また SPIT プラグインを使用する方法も見ってきましたので、今は読み込みを行うために DB Manager を使用する方法を学ぶことができます。

- [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。

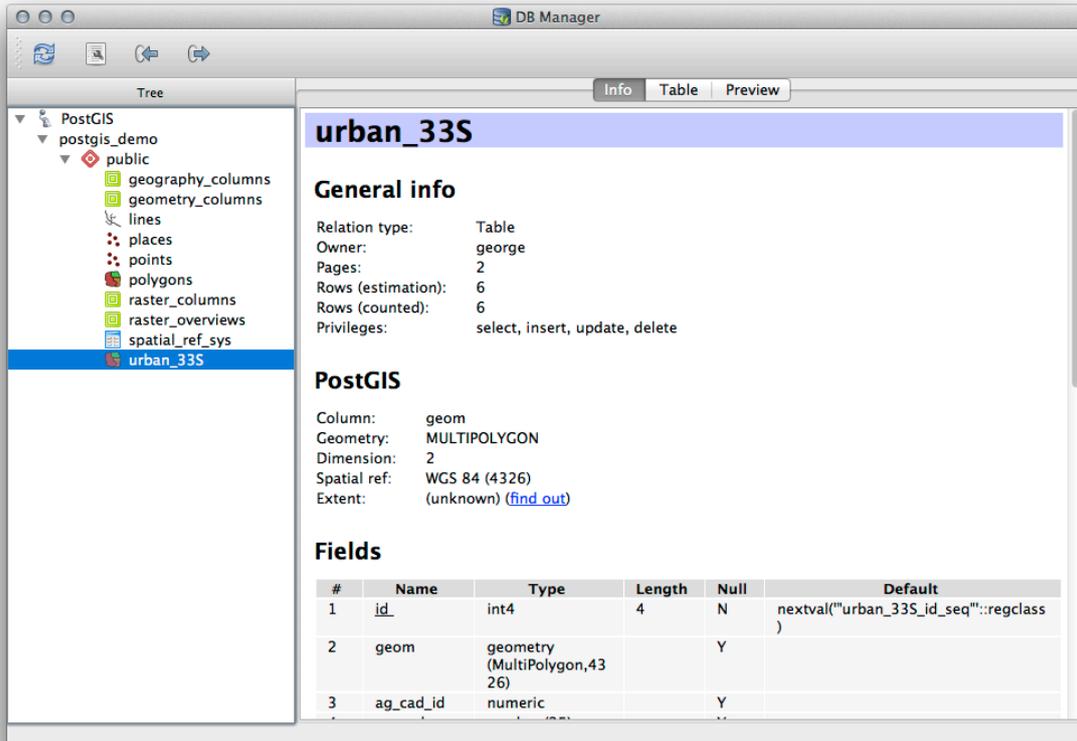


- `exercise_data / projected_data` から `urban_33S.shp` ファイルを入力データセットとして選択します。
- フォームの値の一部を事前に埋めるために、更新 *Options* ボタンをクリックしてください。
- 新しい *table* を作成 オプションが選択されていることを確認してください
- 元 *SRID* を 32722、先 *SRID* を 4326 として指定します。
- 空間 *Index* を作成 の横にあるチェックボックスをオンにします。
- *OK* をクリックして読み込みを実行します。



- 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
- DB マネージャのツールバー上の *Refresh* ボタンをクリックしてください。

今の木でそれをクリックすることで、データベース内のテーブルを調べることができます。空間 *REF* は WGS 84 (4326) のように表示されているデータは、そのチェックすることによって再投影されていることを確認し

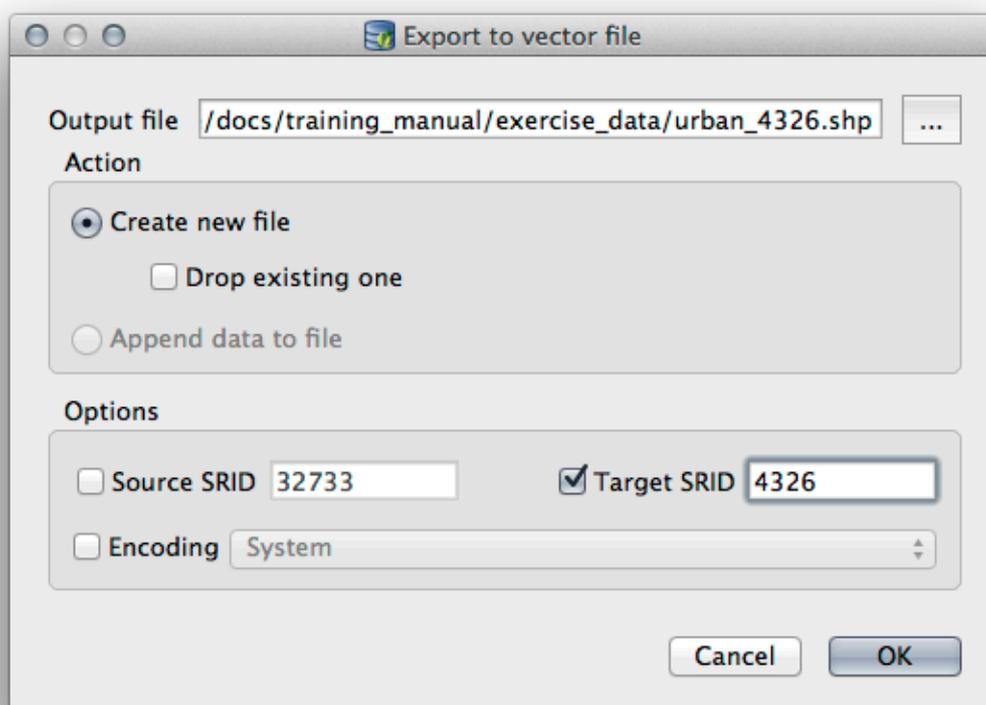


右ツリーと選択して、テーブルをクリックして *Canvas* に追加する地図内のレイヤとしてテーブルを追加します。

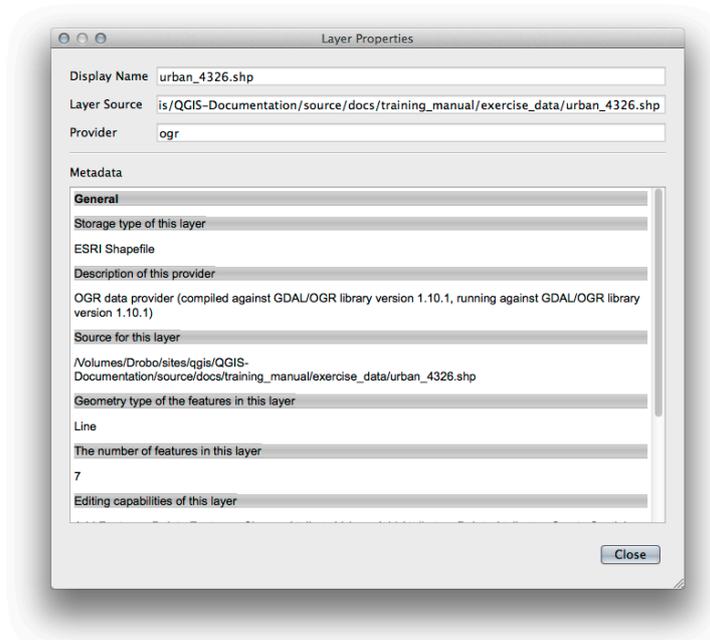
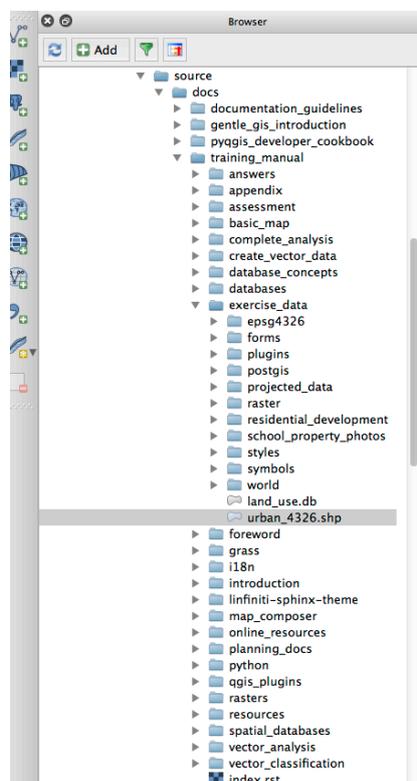
18.2.6 DB マネージャを使用したデータベースからのデータの書き出し

もちろん、DB Manager は空間データベースからデータを書き出すために使用できます。そこで、それがどのように行われるかを見てみましょう。

- ツリーで *lines* レイヤーを選択し、ツールバー上の *File* に書出 ボタンをクリック ベクトルファイルを書き出し ダイアログを開きます。
- `:guilabel:' ... ボタンがクリック` `:guilabel:'出力 file とデータを保存` 選択し の `exercise_data` ディレクトリとして `urban_4326`。
- ターゲット *SRID* を 4326 と設定します。
- *OK* をクリックして書き出しを初期化します。



- 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 In Conclusion

今まで、QGIS で空間データベースを管理するための DB マネージャインターフェースを使用して、データに対して SQL クエリを実行する方法、およびデータを読み込んだり書き出しする方法を見てきました。

18.2.8 What's Next?

次は *spatialite* データベースでこれらの同じ技術の多くを使用する方法を見ていきます。

18.3 Lesson: QGIS における SpatiaLite データベースの操作

PostGIS は一般に、同時に複数のユーザに空間データベース機能を提供するためにサーバー上で使用されますが、QGIS では、空間データベース全体を単一のファイル格納するための軽量かつポータブルな方法である *spatialite* と呼ばれるファイル形式の使用もサポートしています。もちろん、空間データベースのこれらの 2 種類は異なる目的のために使用されるべきですが、同じ基本原則とテクニック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と連携して提供される機能を探ってみましょう。

このレッスンの目標：QGIS ブラウザのインターフェイスを使用して SpatiaLite データベースを利用する方法を学ぶ。

18.3.1 Follow Along: ブラウザで SpatiaLite データベースを作成する

ブラウザのパネルを使って新しい SpatiaLite データベースを作成し、QGIS で使用するための準備をすることができます。

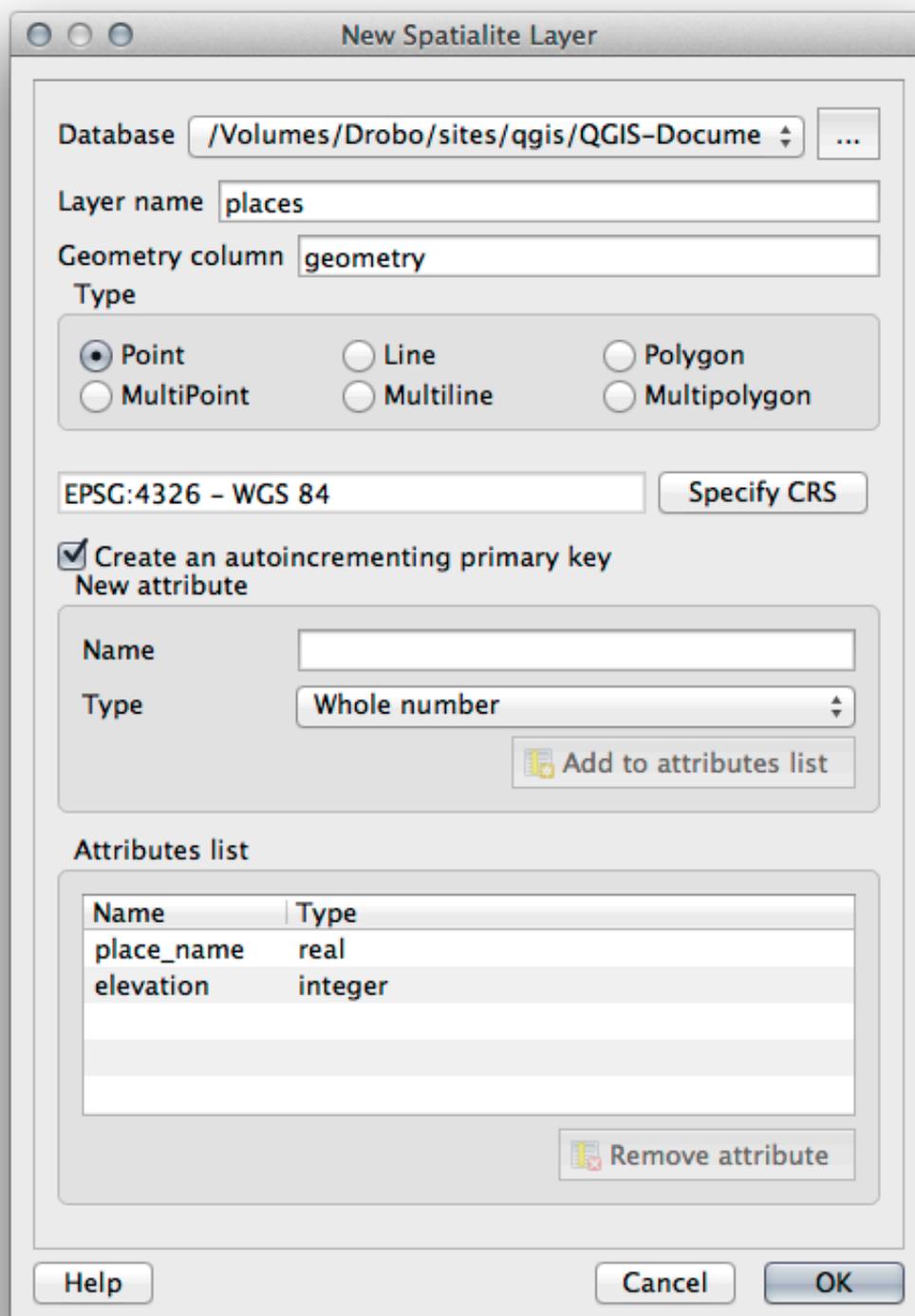
- *Spatialite* エントリを右クリックし、*Create Database* を選択します。
- *qgis-s1.db* という名前前でファイルシステム上に保存します。
- 再びブラウザツリー内の *Spatialite* を右クリックし、*New Connection* を選択して先ほど作成したファイルを探して開きます。

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。

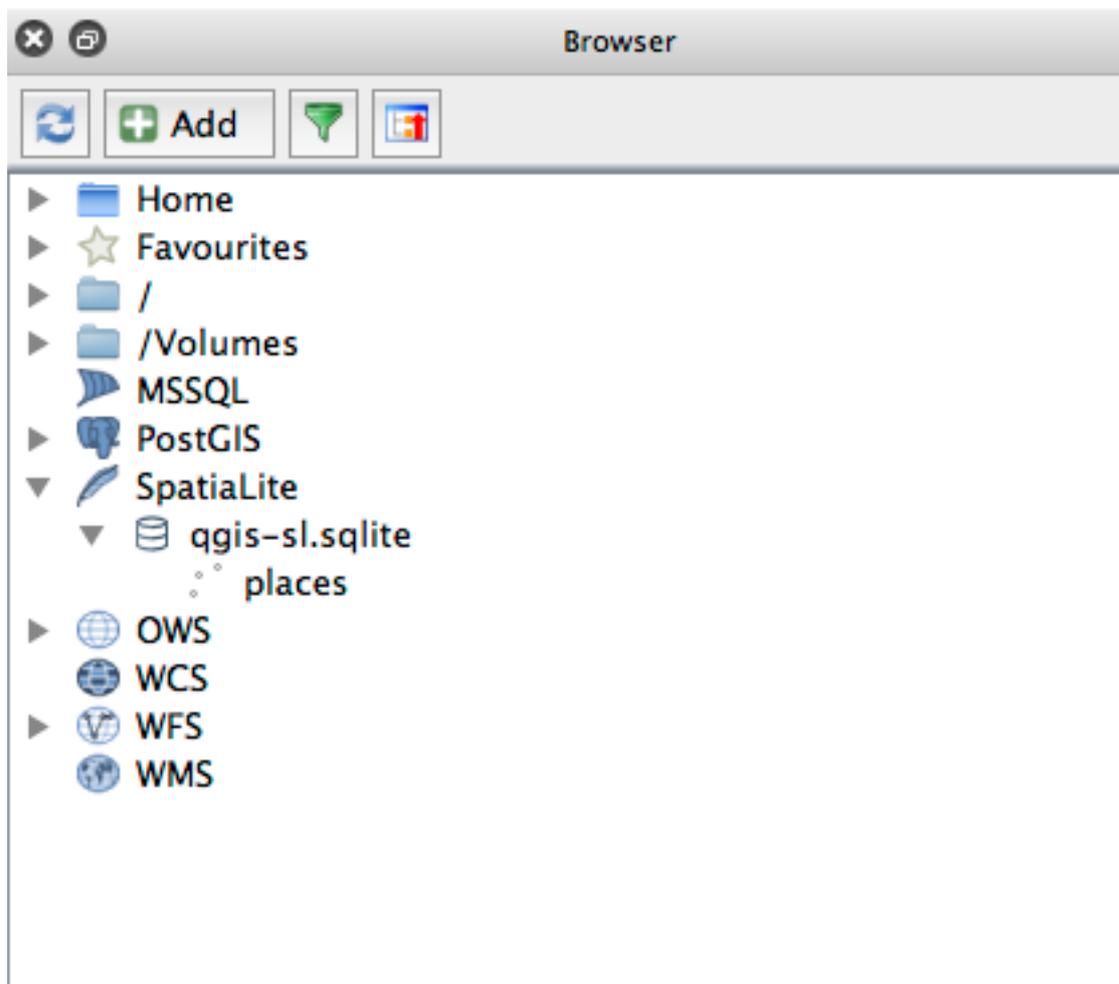
- 新しいレイヤーを作成するためのボタンを見つけ、新しい新しい *Spatialite* レイヤーを作成するために、ドロップダウンを使用するか、*-->新規 ->新規 Spatialite Layer* を選択します。



- 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。
- レイヤーに *places* という名前を付けてください。
- 次に *Create an auto-incrementing primary key* のチェックボックスを選択します。
- 下のような 2 つの属性を追加
- *OK* をクリックしてテーブルを作成します。



- ブラウザ上部にある更新ボタンをクリックすると, places テーブルの一覧が表示されます.



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。

DB Manager を利用してデータベースにデータをインポートする方法を学びました。新しい SpatiaLite データベースにデータをインポートするためにこの手法を用いることができます。

18.3.2 In Conclusion

ここでは SpatiaLite データベースを作成し、そこにテーブルを追加して QGIS のレイヤとしてテーブルを利用する方法を見てきました。

Chapter 19

Module: インタフェース

19.1 概要

このコースで我々が扱うのはこのとおりです：

- What is python? * Hello world
- プログラムロジック * python の空白 * 変数の宣言 * 式 * ループ * if..then..else * 関数の宣言 * 関数のドキュメント化
- Python のデータ型 (動的型指定、厳密な型指定) * 文字型、整数型、浮動小数点型 * 辞書型 * リスト型 * タプル * 文字型の書式 * リスト内包書式
- 内観 * 省略可能な引数と名前付き引数 * type, str, dir 関数 * getattr 関数 * ラムダ関数 * __doc__
- オブジェクト*モジュールのインポート*インポート検索パス*の定義クラス*クラスの初期化 (コンストラクタ) *自己*クラスのインスタンス化*ガベージコレクション*インスタンス変数 (クラスメンバー) * (サポートされていません) *クラスの属性 (静的クラス変数) のオーバーロードメソッド*プライベート関数 (クラス)(モジュール) *プライベートクラスメソッド* (クラスの) プライベート属性
- 例外* try...except * try...except...else * try...except...finally
- 読み取りテキストファイルはテキストファイルに*ファイルパス操作 (os モジュール) *分割・パスを書く*ディレクトリリスト/グロブ* * IO ファイル

19.2 Lesson: Python の基礎

このレッスンでは、Python の基本を紹介します。あなたが他の言語 (Java や C ++、VB など) でプログラムされている場合は、それが他の言語、特にコードフォーマットの要件の観点から、作業方法に多少違いがありますが、Python を学ぶことは非常に簡単かつスムーズであることがわかるでしょう。

19.2.1 Follow Along: Hello World

python.org から python をインストールし、端末あるいはコマンドウィンドウを開き、python プロンプトを開始します：

```
timlinux@ultrabook:~/dev/cpp/QGIS-Training-Manual/python$ python
```

開始するとこのようなメッセージが表示されるでしょう：

```
Python 2.7.3 (default, Aug 1 2012, 05:14:39)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

ここで下に示すように、コマンドプロンプトで `print 'Hello World'` とタイプします。

```
>>> print 'Hello World'
```

コマンドを実行すると、python は反応します:

```
Hello World
>>>
```

おめでとうございます、初めて python アプリケーションを書きました！

ノート: You can escape from the python prompt by pressing `ctrl-D` or by typing `quit()` and then pressing Enter.

19.2.2 Follow Along: ファイルからのコマンド実行

当然、インタラクティブに python コマンドをタイプするのに使用が限定されるため、python コマンドは `.py` ファイルとして保存してから実行するのが一般的です。例えば、`hello_world.py` と呼ばれるテキストファイルにこの行を保存します。

```
print 'Hello World'
```

ノート: 慣例により、スペースやハイフンを含むファイル名を使用して Python コードを保存するのは避けましょう。

これで、コマンドプロンプトから以下のように入力することで、プログラムを実行することができます:

```
python hello_world.py
```

19.2.3 |FA|ファイルでのインタプリタの定義

ファイルを直接実行することができれば、より便利になります。ファイルの先頭にインタプリタ注釈を追加することにより、Linux、および Mac OSX でこれを行うことができます:

```
#!/usr/bin/python
print 'Hello World'
```

また、以下のようにして、ファイルを実行可能にする必要があります:

```
chmod +x hello_world.py
```

これで、以下のようにしてファイルを実行することができるようになりました:

```
./hello_world.py
```

ノート: これを行うと、(訳注: 環境によって python のパスが異なるため) 異なるオペレーティングシステム間での移植性を損なう可能性があります。

Chapter 20

付録：このマニュアルに貢献する

このコースに材料を追加するには、この付録のガイドラインに従う必要があります。明確化のためを除き、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できるようにするためです。

20.1 リソースのダウンロード

この文書のソースは *GitHub* の <https://github.com/qgis/QGIS-Documentation> 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 <https://github.com/>](https://github.com)参照してください。

20.2 マニュアルの形式

このマニュアルは、[‘スフィンクス<http://sphinx.pocoo.org/>](http://sphinx.pocoo.org/)、[‘reStructuredText の使用 Python のドキュメントジェネレータ<http://docutils.sourceforge.net/rst.html>](http://docutils.sourceforge.net/rst.html)、[_ マークアップ言語](#)を使用して書かれています。これらのツールを使用する方法については、それぞれのサイトからご利用可能です。

20.3 モジュールの追加

- 新しいモジュールを追加するには、最初に新しいモジュール名を持つ新しいディレクトリ（*QGIS-トレーニングマニュアル* ディレクトリのトップレベルのすぐ下）を作成します。
- 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、このファイルは空白のままにします。
- 最上位ディレクトリの下での `index.rst` ファイルを開きます。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、各ディレクトリ内のインデックスファイルにトップレベルのインデックスファイルを指示します。それらがリストされている順序は、文書中での順序を決定します。

- モジュールを表示したい場所に、このリストに、`/index` を続けて、新しいモジュール（すなわち、新しいディレクトリを付けた名前）の名前を追加します。

- 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。
- 新しいモジュールの独自のインデックスファイルを開きます ([module name]/index.rst)。
- ページの上部に沿って、80 個のアスタリスクの行を書く (*)。これは、モジュールの見出しを表します。
- これに続けて |MOD| (「モジュール」の略)、モジュールの名前が続くマークアップフレーズを含む行でください。
- 80 個のアスタリスクの別の行でこれをオフに終了します。
- この下に空白行を残します。
- モジュールの目的と内容を説明する短いパラグラフを書きます。
- 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、計画された授業の名前です。

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。

```
*****
|MOD| Module Name
*****
```

Short paragraph describing the module.

```
.. toctree::
    :maxdepth: 2

    lesson1
    lesson2
```

20.4 レッソンの追加

新規または既存のモジュールにレッスンを追加:

- モジュールディレクトリを開く
- 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。
- 計画された授業の名前が、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てください。
- 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。
- このファイルにモジュールの index.rst ファイルで与えられ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。

ノート: 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。

- レッソンの書き込みを開始するには、マークアップのフレーズ |LS| を書き、その後ろにレッスン名を書きます。
- 次の行では、80 の等号 (=) の行を書き込む。

- この後に空行を残します。
- 授業の意図された目的の短い説明を書きます。
- 主題への一般的な導入を含みます。例については、このマニュアルでは、既存のレッスンを参照してください。
- この下には、このフレーズから始まる、新しい段落を開始:

```
**The goal for this lesson:**
```

- 簡単に言えば、このレッスンを完了することを意図し成果を説明します。
- 1つのまたは2つの文にレッスンの目標を記述することができない場合は、複数のレッスンにかかわらアップ対象を壊すことを検討してください。

各レッスンには、次の対処されることになり、複数のセクションに細分化されます。

20.5 セクションの追加

セクションは2種類あります。「この通りに従ってください」と「自分でやってみよう」。

- 「この通りに従ってください」セクションには、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方向の詳細なセットです。これは通常、クリックごとのクリックスクリーンショットが散在し、できるだけ明瞭として指示を与えることによって行われます。
- 「自分でやってみよう」セクションでは、読者自身で試して短い割り当てを与えます。これは通常表示または割り当てを完了するために、可能な場合に予想される結果を表示する方法を説明しますドキュメントの最後に解答用紙のエントリに関連付けられています。

すべてのセクションでは、難易度が付属しています。簡単なセクションは |basic|、適度は |moderate|、および上級は |hard| で表されます。

20.5.1 「一緒に従う」セクションを追加

- (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。
- |FA| を書き込み、その後スペースを入れてください (“一緒に従う”のため)。
- (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)、別のスペースを残して、セクションの名前を書きます。
- 次の行では、80 マイナス/ダッシュの線を書き込む (-)。テキストエディタが長いダッシュまたは他の文字でデフォルトマイナス/ダッシュ文字に置き換えられないことを確認してください。
- 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、立証されるように手続き上の詳細 (クリックでクリック) 指示を与えます。
- 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。
- 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。

20.5.2 「自分でやってみよう」セクションを追加

- (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。
- スペースを残し、その後 |TY| と書き込みます (“自分でやってみよう”のために)。
- 次の行では、80 マイナス/ダッシュの線を書き込む (-)。テキストエディタが長いダッシュまたは他の文字でデフォルトマイナス/ダッシュ文字に置き換えられないことを確認してください。

- 読者が完了し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てください。
- プレーンテキスト記述が明確でない場合は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割り当てを完了する方法についての答えを提供したいと思うでしょう。そのためには、解答用紙にエントリを追加する必要があります。

- まず、回答に一意の名前を決定します。理想的には、この名前は、レッスンの名前とインクリメント番号が含まれます。
- この回答へのリンクを作成:

```
:ref: `Check your results <answer-name>`
```

- 解答用紙を開きます (answers/answers.rst)。
- この行を書き込むことによって、「自分を試してみてください」セクションへのリンクを作成します。

```
.. _answer-name:
```

- 必要な場合、リンクや画像を使用して、割り当てを完了する方法の手順を書きます。
- それを終了するには、この行を書き込むことによって、バック「自分自身を試してみてください」セクションへのリンクが含まれています。

```
:ref: `Back to text <backlink-answer-name>`
```

- このリンクを動作させるために、「自分を試してみてください」セクションに、見出しの上に次の行を追加します。

```
.. _backlink-answer-name:
```

文書を作成しながら、それ以外の場合はエラーが発生する可能性があります。上に示したこれらの線のそれぞれは、その上下の空白行を持たなければならないことに注意してください。

20.6 結論を追加

- フレーズを書き、レッスンを終了するには `|IC|` 80 マイナス/ハイフンの新しい行に続いて、“結論に”のために (-)。レッスンでカバーされている概念を説明、レッスンの結論を書きます。

20.7 また、読取部を追加します。

- このセクションは追加分です。
- 80 マイナス/ハイフンの新しい行に続いて“さらに読書”のためのフレーズ `FR` を書く (-)。
- 適切な外部のウェブサイトへのリンクが含まれています。

20.8 何の次のセクションを追加します。

- フレーズ `|WN|` を書く 80 マイナス/ハイフンの新しい行に続いて、“次に”(-)。
- このレッスンは、次のレッスンまたはモジュールのための学生を準備している方法について説明します。
- 必要であれば、それは新しいレッスンを参照するように、前のレッスンの「次は」のセクションを変更することを忘れないでください。これは既存の授業の間に、または既存のレッスン後に新しいレッスンを挿入した場合に必要となります。

20.9 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。

20.9.1 新しい概念

- 新しい概念を説明している場合は、アスタリスク (*) で囲むことでイタリック体で新しい概念の名前を記述する必要があります。

This sample text shows how to introduce a **new concept**.

20.9.2 強調

- 新しい概念ではない重要な用語を強調するために、二重のアスタリスク (**) で囲むことで太字の用語を記述します。
- これは控えめに使用!あまり使用しすぎると、読者には叫んでたり恩着せつつあるように見えるかも。

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.****

20.9.3 画像

- 画像を追加する場合は、フォルダ `_static/lesson_name/` に保存します。
- このような文書でそれを含める:

```
.. image:: /static/training_manual/lesson_name/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。

20.9.4 内部リンク

- リンクのアンカーを作成するには、リンクを::を指すようにしたい場所の上に次の行を書きます

```
.. _link-name:
```

- リンクを作成するには、この行を追加します:

```
:ref:`Descriptive link text <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。

20.9.5 外部リンク

- 外部リンクを作成するには、このようにそれを書き出す:

```
`Descriptive link text <link-url>`_
```

- 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。

20.9.6 等幅のテキストを使用して

- ユーザーが入力する必要があるテキスト、パス名、またはテーブルやカラム名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 `text` で記述する必要があります。例えば ::

```
Enter the following path in the text box: :kbd:`path/to/file`.
```

20.9.7 ラベリング GUI アイテム

- 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.
```

- ボタンをクリックし、ユーザを必要とせずに、ツールの名前を言及している場合にも適用されます。

20.9.8 メニューの選択

- メニューを介してユーザーを導く場合は、`メニュー -> 選択 -> format` 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`.
```

20.9.9 ノートを追加する

- 簡単に授業の流れの一部を行うことができない余分な詳細を説明するテキストに注釈が必要になる場合があります。これは、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.]
```

20.9.10 後援/原作者ノートを追加します

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、あなたは自分の好きなショート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の読者に通知しなければならないと、彼ら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、それは自分の会社の広告ではないかもしれません。

自身の容量のモジュール、レッスンまたはセクションを記述するために志願し、ないスポンサーを代表している場合は、あなたが執筆モジュール、レッスンやセクションの見出しの下、原作者のノートを含むことができます。これは形を取る必要がありますの `kbd`:`` : ‘この [モジュール/レッスン/セクション] [著者名] .’ このような詳細は、の「協力者」セクションに追加されるさらに、テキスト、連絡先などを追加しないでくださいで貢献しましたはじめは、一部 (複数可) の名前 (複数可) と一緒にあなたが追加しました。あなただけの機能強化、修正および/または追加を行った場合は、編集者として自分自身をリストします。

20.10 ありがとうございます！

このプロジェクトに貢献してくれてありがとう！そうすることで、ユーザーに QGIS は、よりアクセスしやすく、全体として QGIS プロジェクトに値を追加しています。

Chapter 21

回答シート

21.1 Results For 最初のレイヤを追加

21.1.1 準備

道路をシンボル化した多数のラインが見えます。このラインは基本地図を作るのにちょうど今ロードしたベクターレイヤ内にあります。

[テキストに戻る](#)

21.2 Results For インタフェースのあらまし

21.2.1 あらまし (パート 1)

バックインタフェースのレイアウトを示す画像を参照し、画面要素の名称と機能を覚えていることを確認してください。

[テキストに戻る](#)

21.2.2 あらまし (パート 2)

1. 名前を付けて保存
2. レイヤにズーム
3. ヘルプ
4. 描画 オン/オフ
5. ラインを計測

[テキストに戻る](#)

21.3 Results For ベクターデータで作業

21.3.1 Shapefiles

あなたのマップには5つのレイヤができます:

- 場所
- 水系
- 建物
- 河川 および
- 道路.

[テキストに戻る](#)

21.3.2 データベース

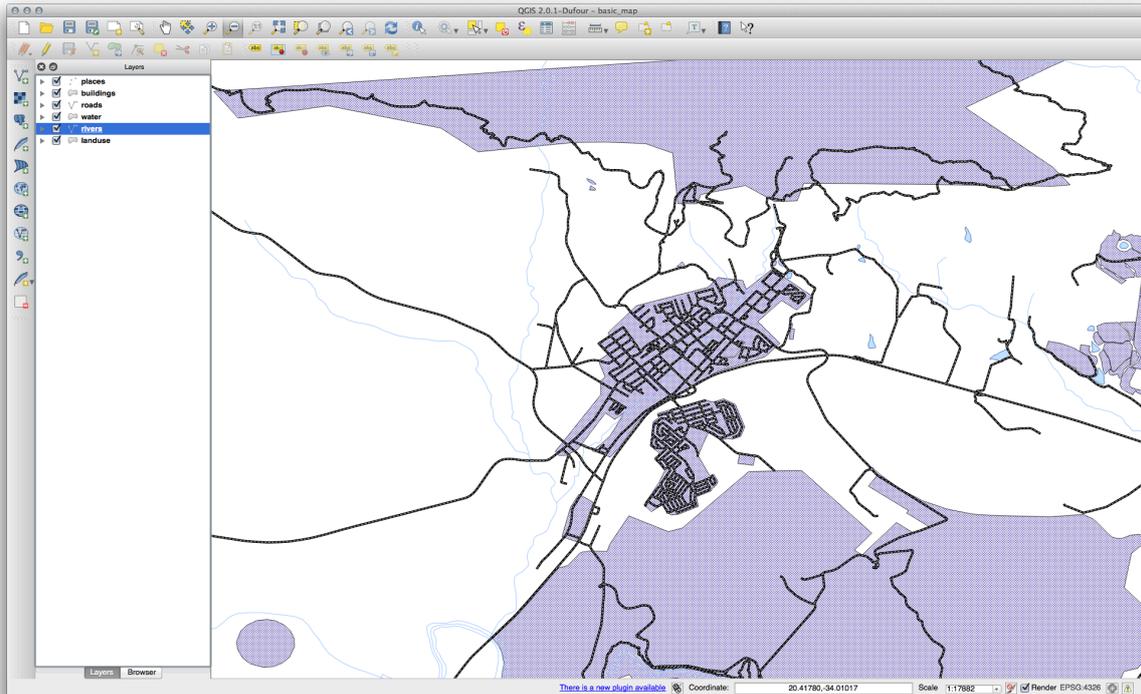
すべてのベクターレイヤはマップにロードしておく必要があります。まだ見た目は素敵ではないかもしれませんが（配色は後ほど見直します）。

[テキストに戻る](#)

21.4 Results For シンボル

21.4.1 色

- 色が期待通りに変わっているか確認してください。
- 現時点では 水系 レイヤを変えるだけで十分です。下記はその例ですが、選んだ色によっては違って見えるかもしれません。

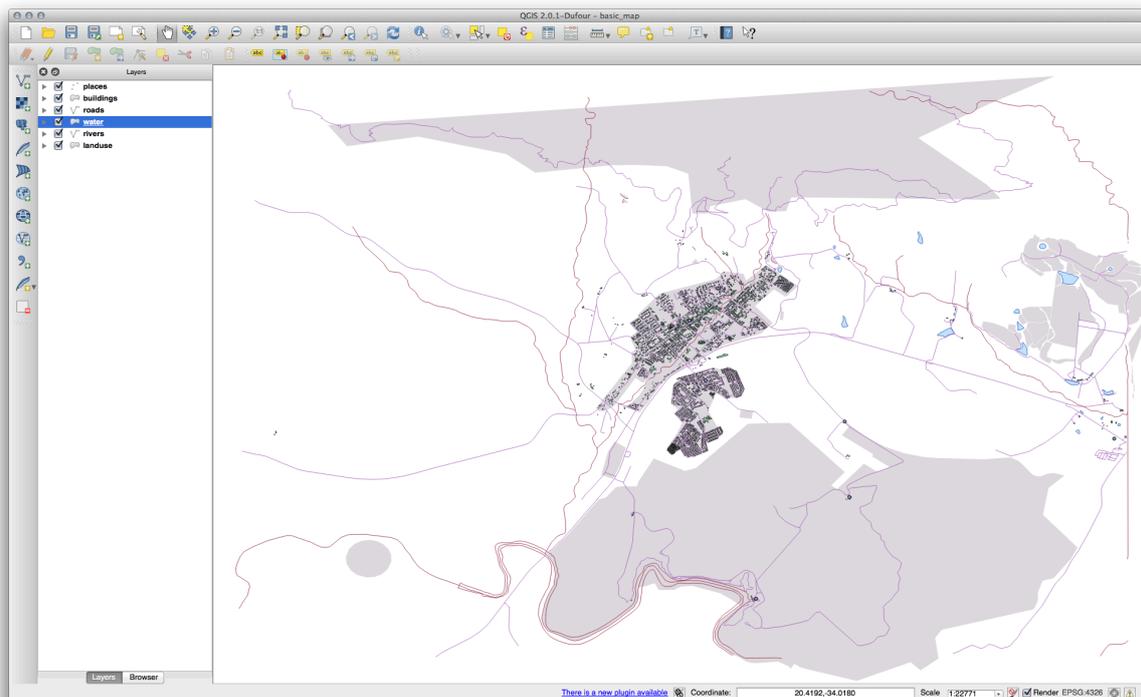


ノート: 他のレイヤに煩わされずに一度にひとつのレイヤだけで作業したい場合、レイヤー一覧内のその名前の隣にあるチェックボックス内をクリックしてレイヤを非表示にできます。ボックスが空の場合にレイヤは非表示です。

テキストに戻る

21.4.2 シンボルの構造

これであなたのマップはこのように見えていると思います:



あなたが初心者レベルのユーザであれば、ここで止めた方が良いでしょう。

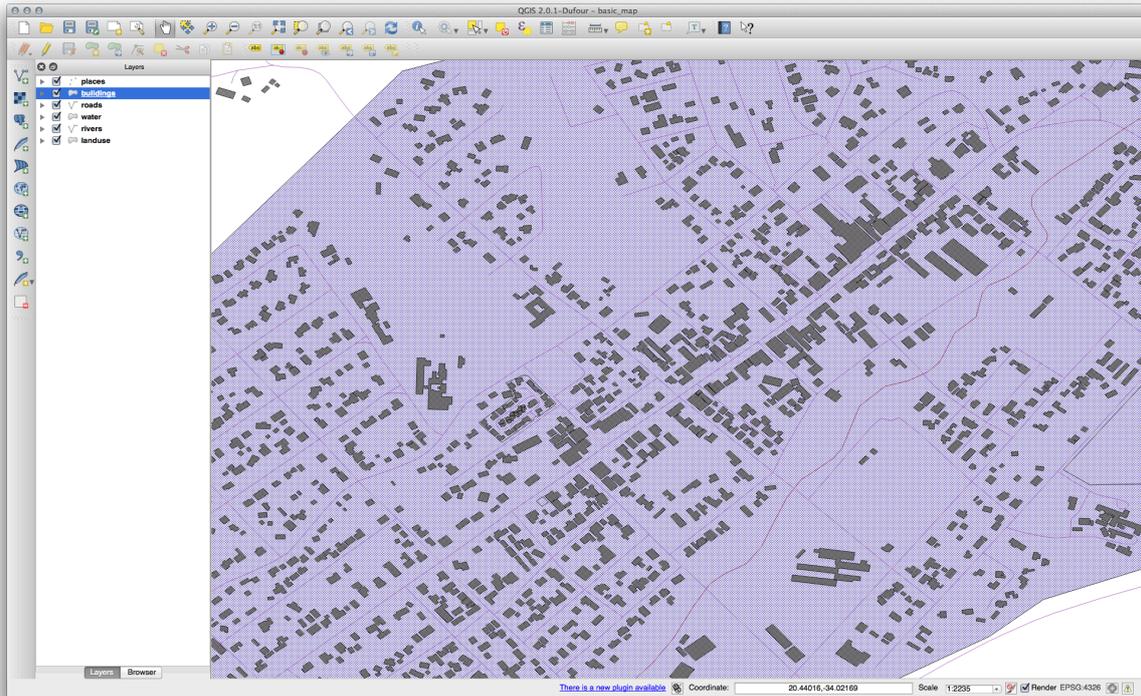
- 上のメソッドを使って残りのレイヤすべての色とスタイルを変更します。
- オブジェクトにはできるだけ本来の色を使うようにしてください。たとえば、道路は赤や青ではなく、グレイまたは黒であるべきです。
- 別のポリゴンの塗りつぶしスタイル ボーダー *Style* 設定も遠慮なく試してください。

テキストに戻る

21.4.3 シンボルレイヤ

- 建物 レイヤは好きなようにカスタマイズできますが、マップ上で離れている異なるレイヤに容易に伝えられるものでなければなりません。

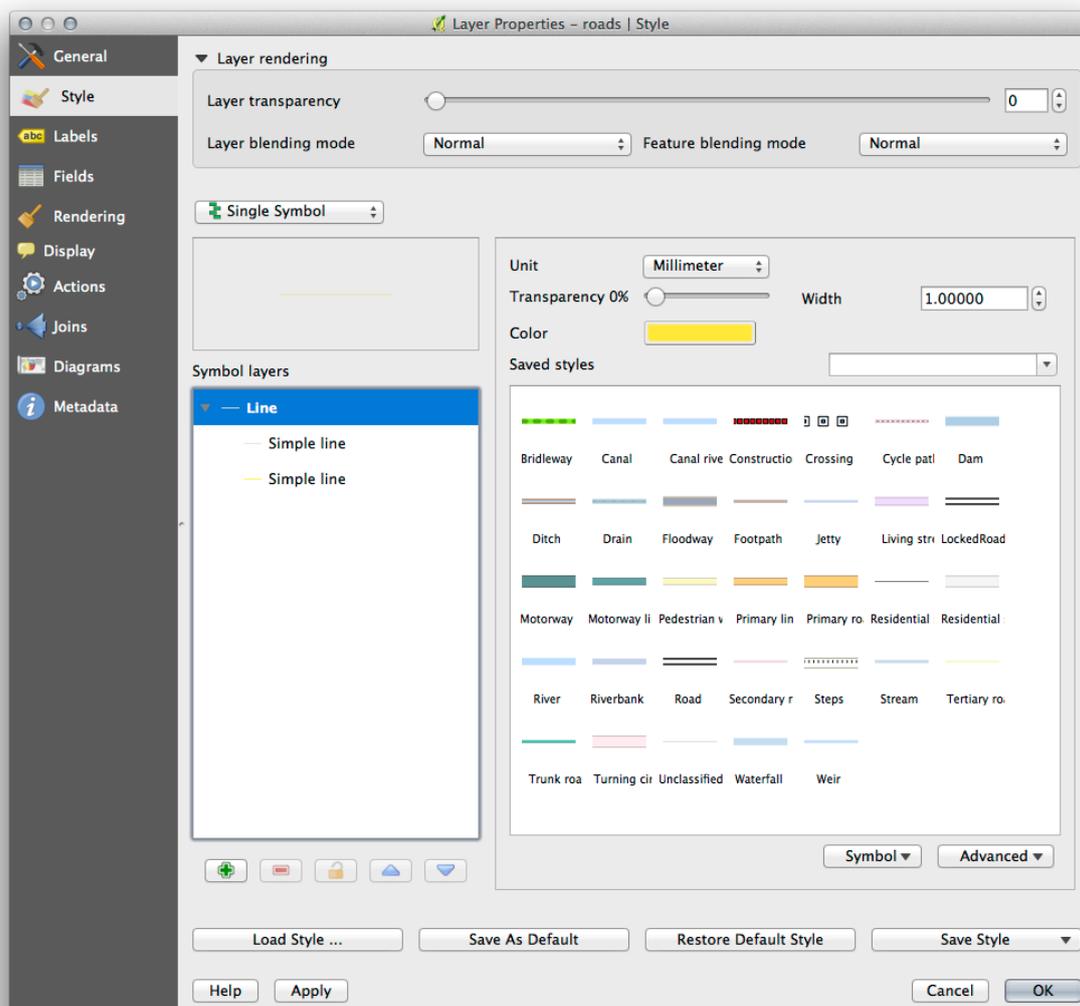
こちらがサンプルです:



テキストに戻る

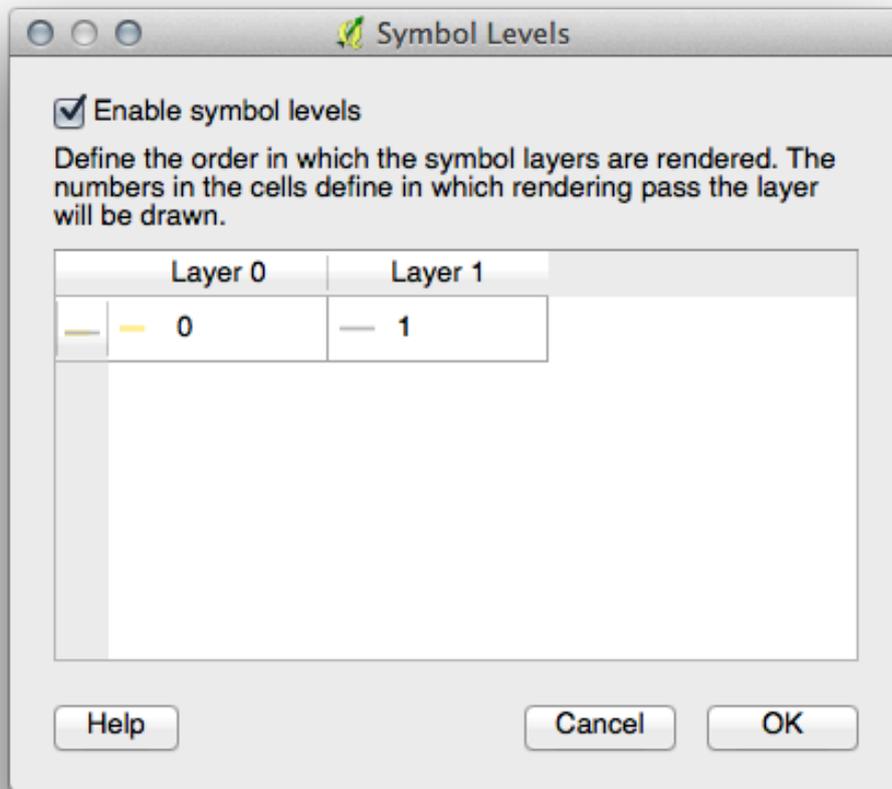
21.4.4 シンボルのレベル

必要なシンボルを作るには 2 つのシンボルレイヤが必要です:

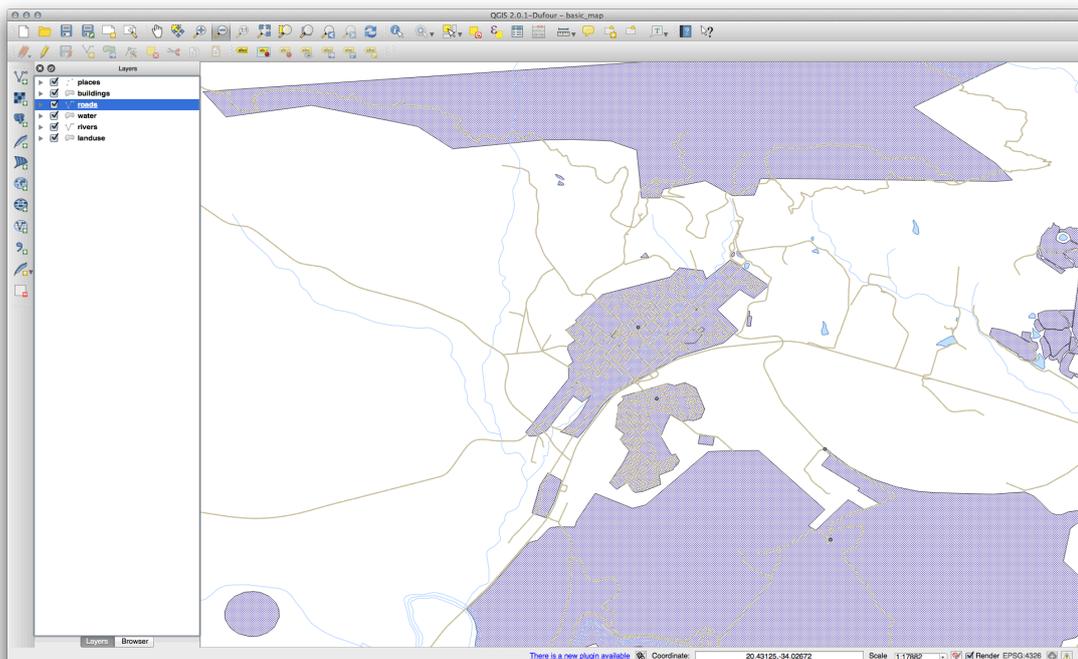


いちばん下のシンボルレイヤは幅広の黄色の実線です。その先頭にはやや狭いグレーの実線があります。

- もしあなたのシンボルレイヤが上記に似てはいるが欲しい結果が得られていない場合は、あなたのシンボルレベルが次のように見えるかどうかチェックしてください:



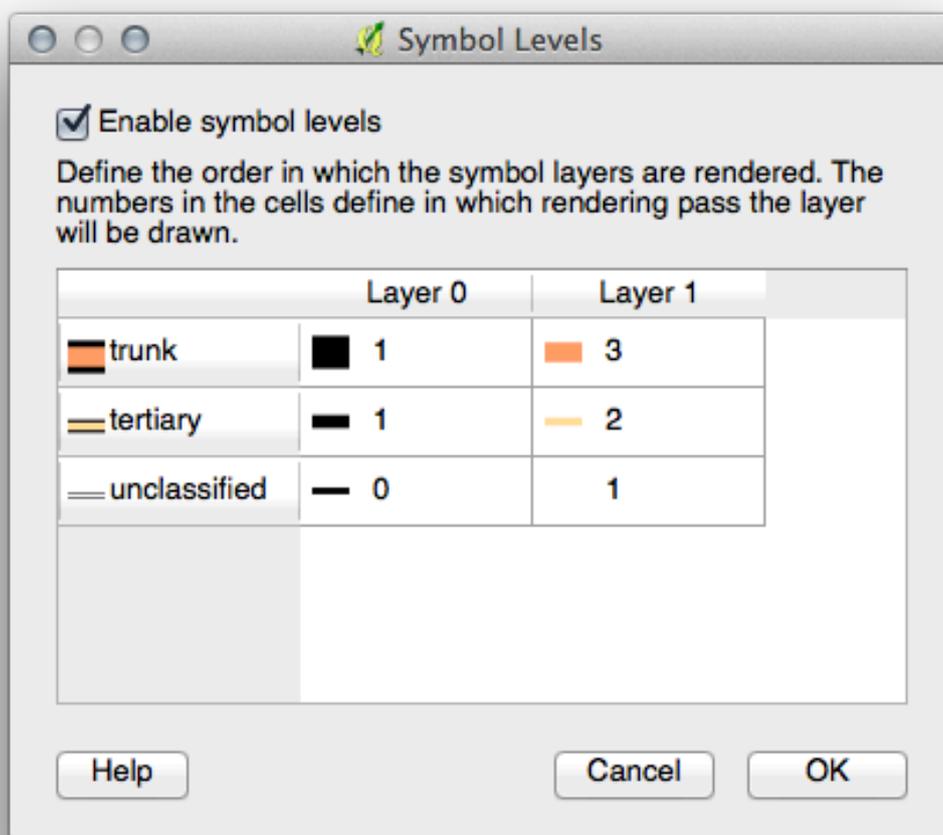
- これであなたのマップは次のように見えるようになったはずです:



[テキストに戻る](#)

21.4.5 シンボルレベル

- あなたのシンボルレベルをこれらの値に合わせてください:



- 値をいろいろ試して様々な結果を取得してみてください。
- 次の演習を続ける前にオリジナルのマップを再度オープンしてください。

[テキストに戻る](#)

21.5 Results For 属性データ

21.5.1 属性データ

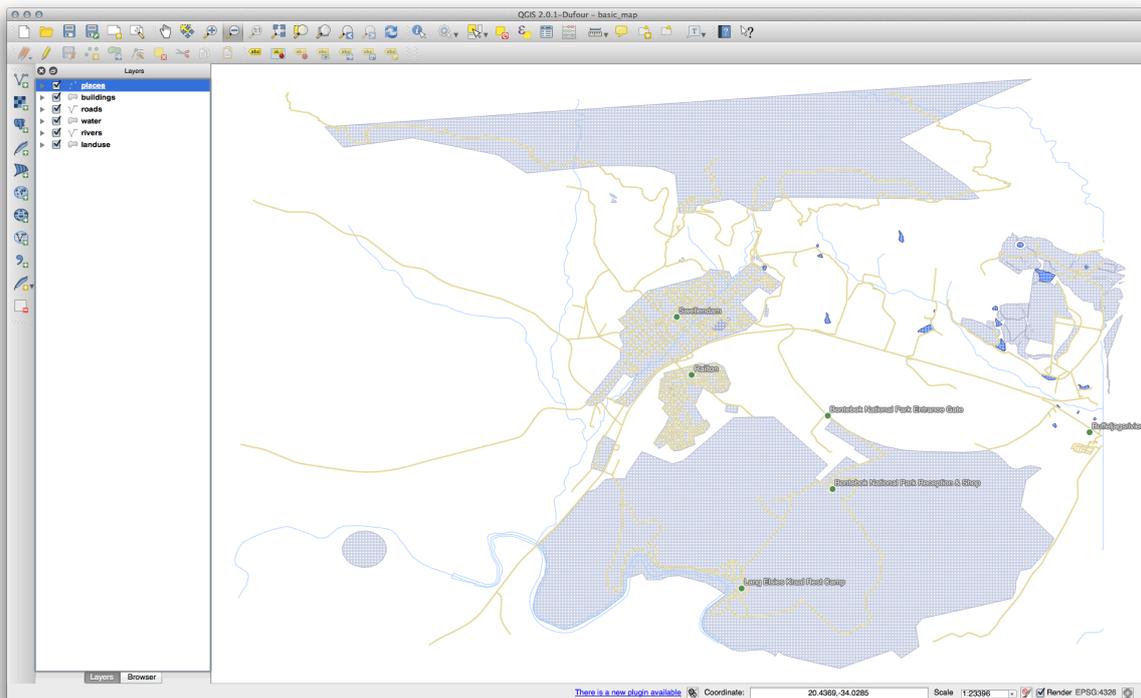
NAME フィールドはラベルとして表示するのに最も便利です。そのすべての値は各オブジェクトについて一意であり、*NULL* 値が含まれている可能性が非常に低いからです。データにいくつかの *NULL* の値が含まれている場合、場所のほとんどが名前を持っている限り心配いりません。

テキストに戻る

21.6 Results For ラベルツール

21.6.1 ラベルのカスタマイズ (パート 1)

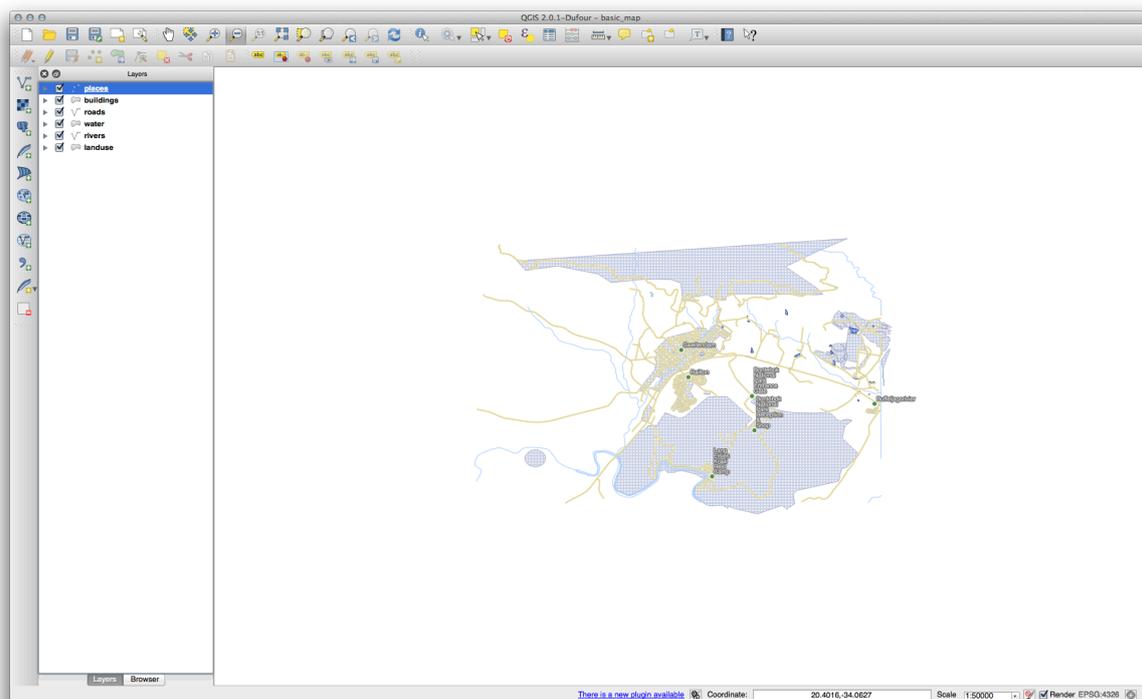
今、地図はマーカ・ポイントを示さなければならず、ラベルは 2.0 mm だけオフセットされなければなりません：マーカとラベルのスタイルは、両方が地図上ではっきりと見えることを可能にする必要があります。



テキストに戻る

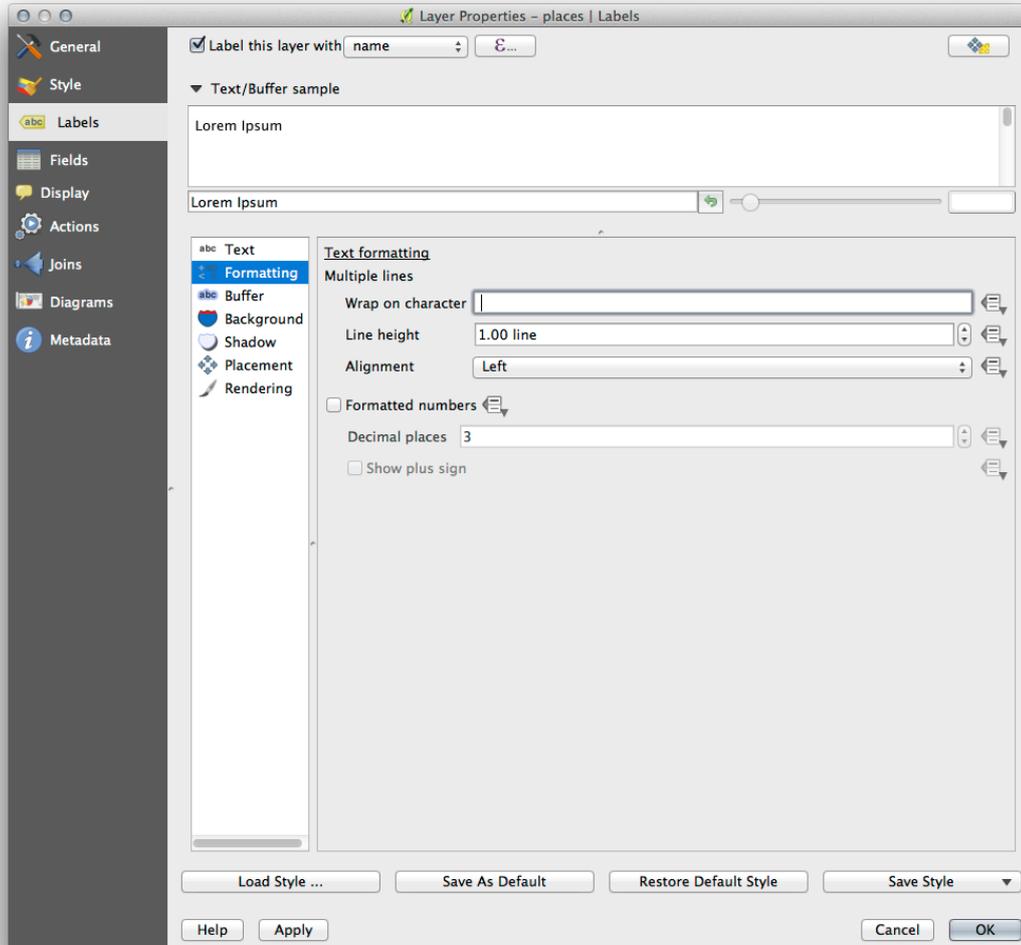
21.6.2 ラベルのカスタマイズ (パート 2)

一つの可能な解決策は、この最終製品があります。



この結果に到着するには：

- フォントサイズ 10、ラベル距離 1.5 mm、シンボル幅とシンボルサイズ 3.0 mm を使用します。
- さらに、この例では *character* ラベルをラップ オプションを使用します。

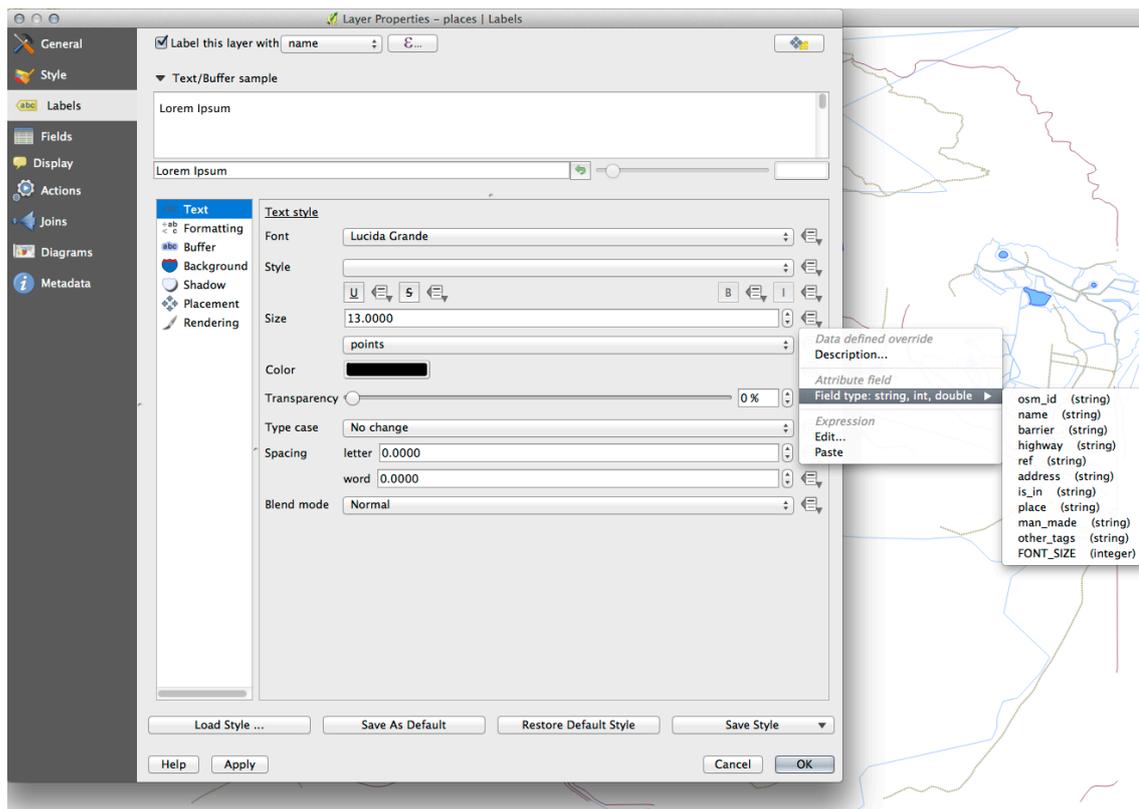


- このフィールドに `space` と入力し *Apply* をクリックして、同じ効果を達成します。この場合には、地名の一部は非常に長く、その結果複数の行を持つ名前が、非常にユーザーフレンドリーではありません。この設定が自分の地図にとってより適切と見つかるかもしれません。

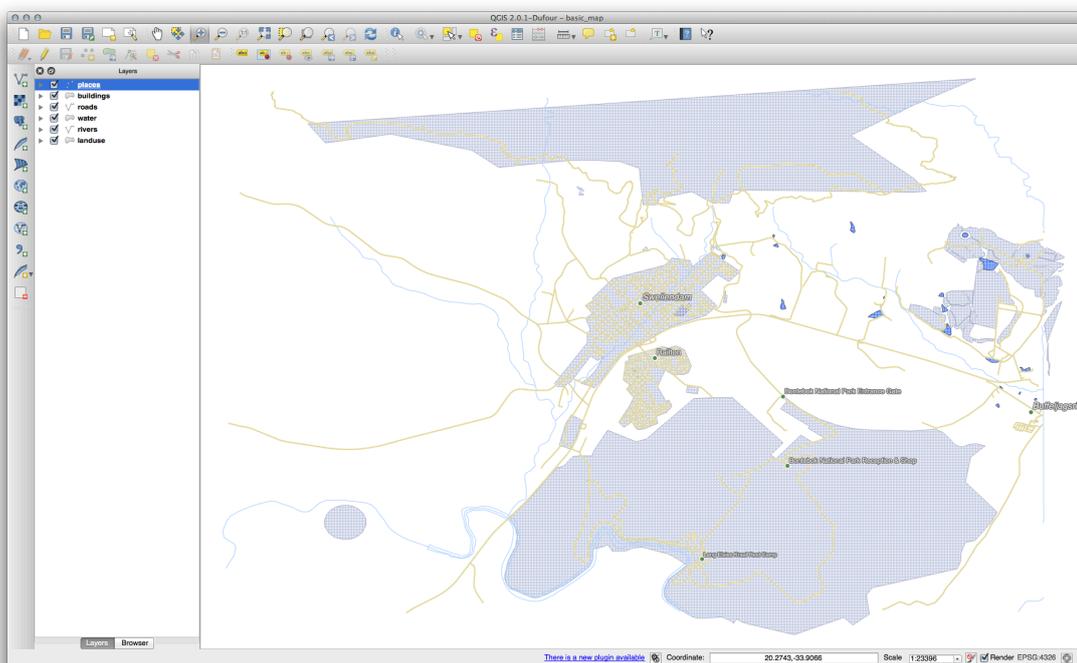
テキストに戻る

21.6.3 データ定義された設定を使用して

- まだ編集モードで、`FONT_SIZE` 値をお好きなように設定してください。例では、町に 16、郊外に 14、地域に 12、集落に 10 を使用しています。
- 編集モードを抜ける前に忘れずに変更を保存してください。
- *places* レイヤーに対する *Text* 書式設定オプションに戻り、フォントサイズのデータの上書きドロップダウンの属性フィールド 中で `FONT_SIZE` を選択：



結果は、上記の値を使用している場合、このようになります。

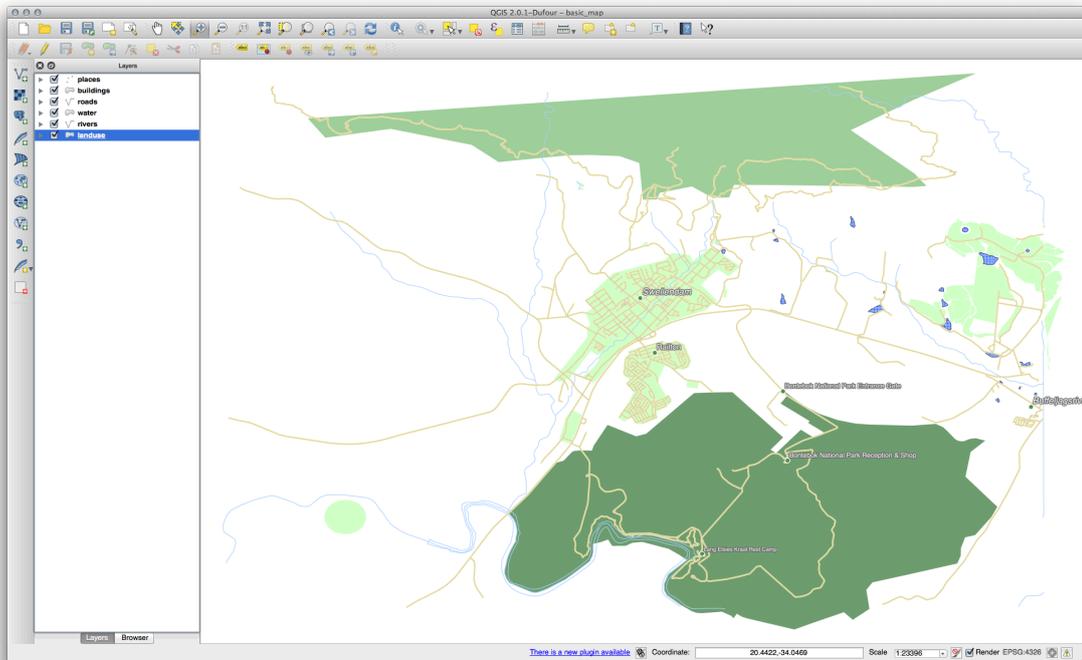


テキストに戻る

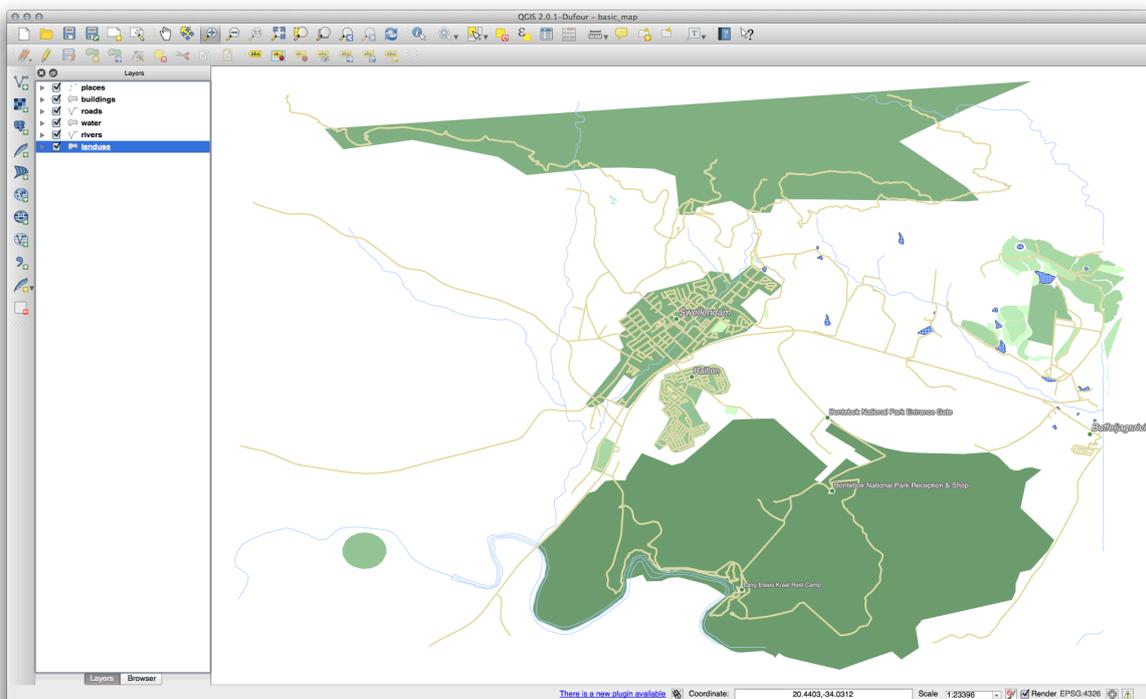
21.7 Results For 分類

21.7.1 分類を改善

- 国境を取り除くために、レッスンの最初の練習と同様の方法を使用します。



使用した設定は同じでなくてよいですが、値は *Classes = 6* とモード = 自然なブレイク (ジェンクス) で (そしてもちろん同じ色を使用して) 地図は次のようになります。

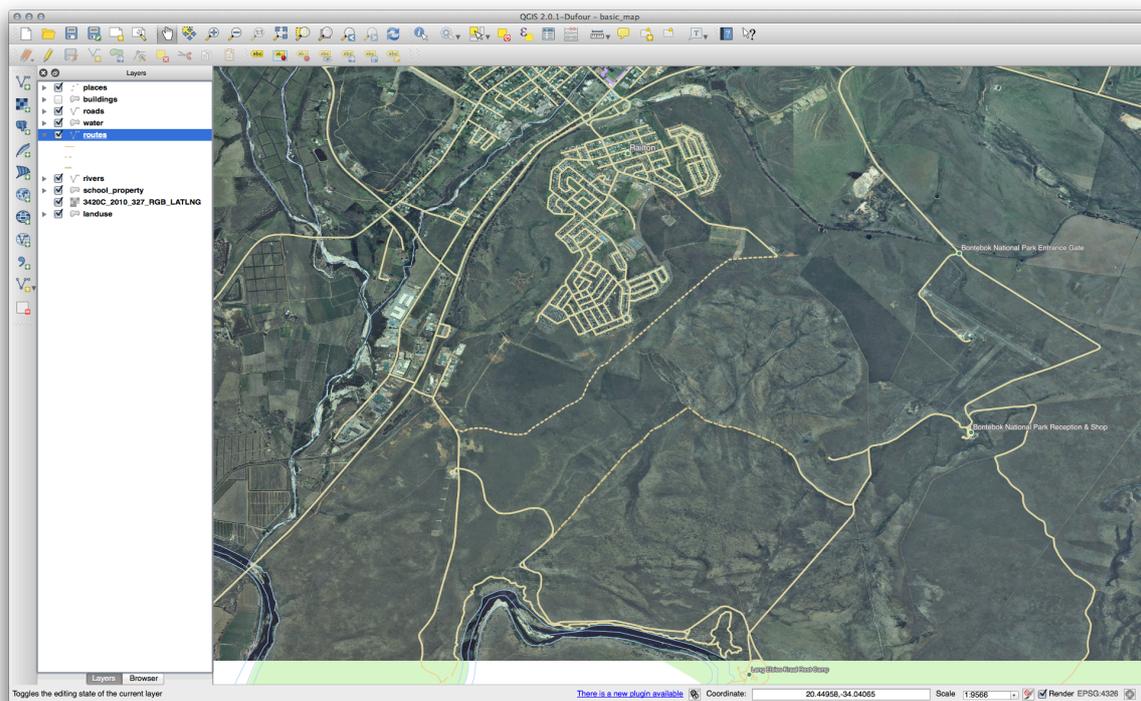


[テキストに戻る](#)

21.8 Results For 新しいベクターデータセットの作成

21.8.1 デジタイジング

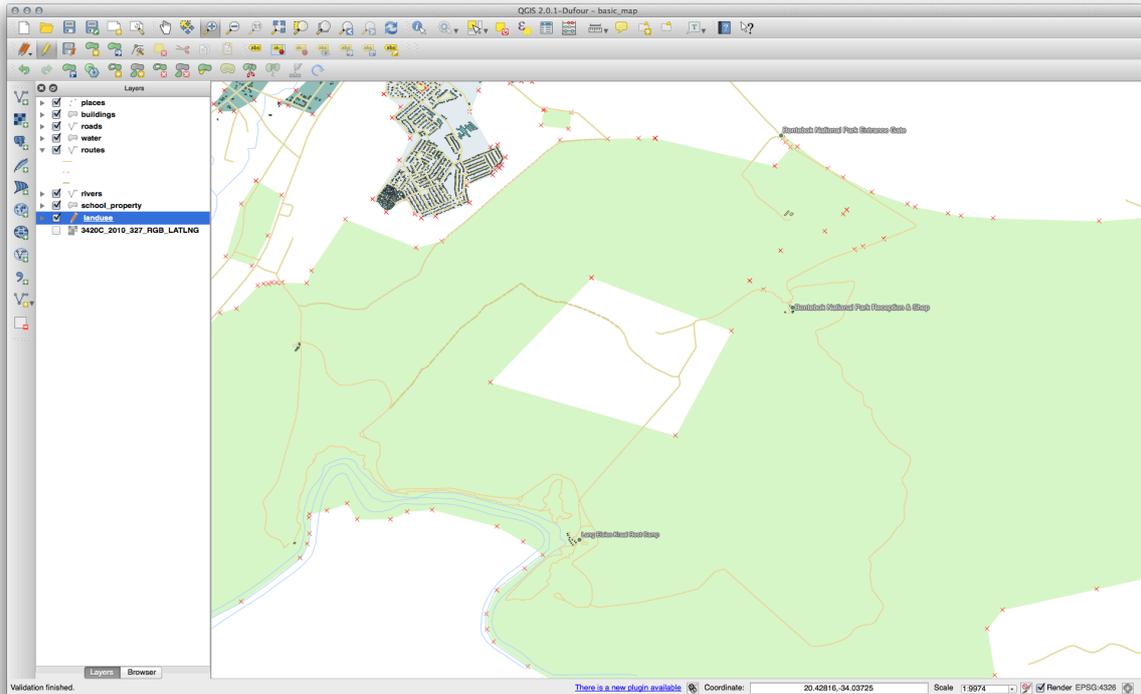
記号は重要ではありませんが、結果はこのような多かれ少なかれなります。



[テキストに戻る](#)

21.8.2 トポロジー: リングツールを追加

正確な形状は重要ではありませんが、あなたの地物の中央には穴が空くことになります。こちらのよう

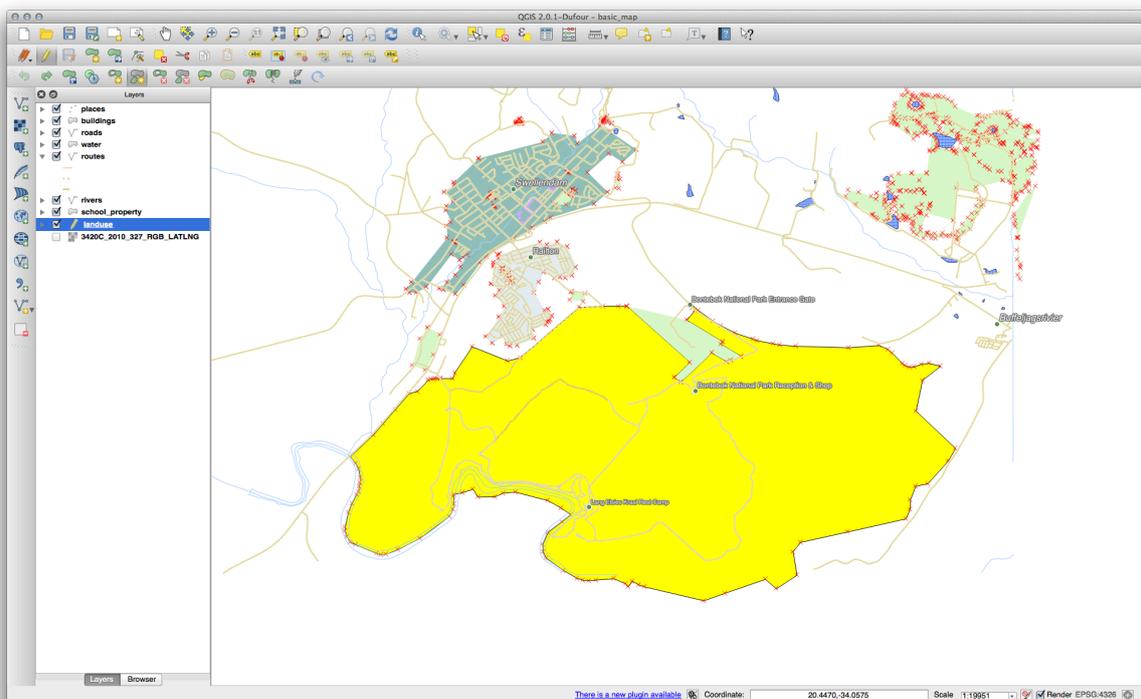


- 次のツールのための演習を続行する前に編集を取り消します。

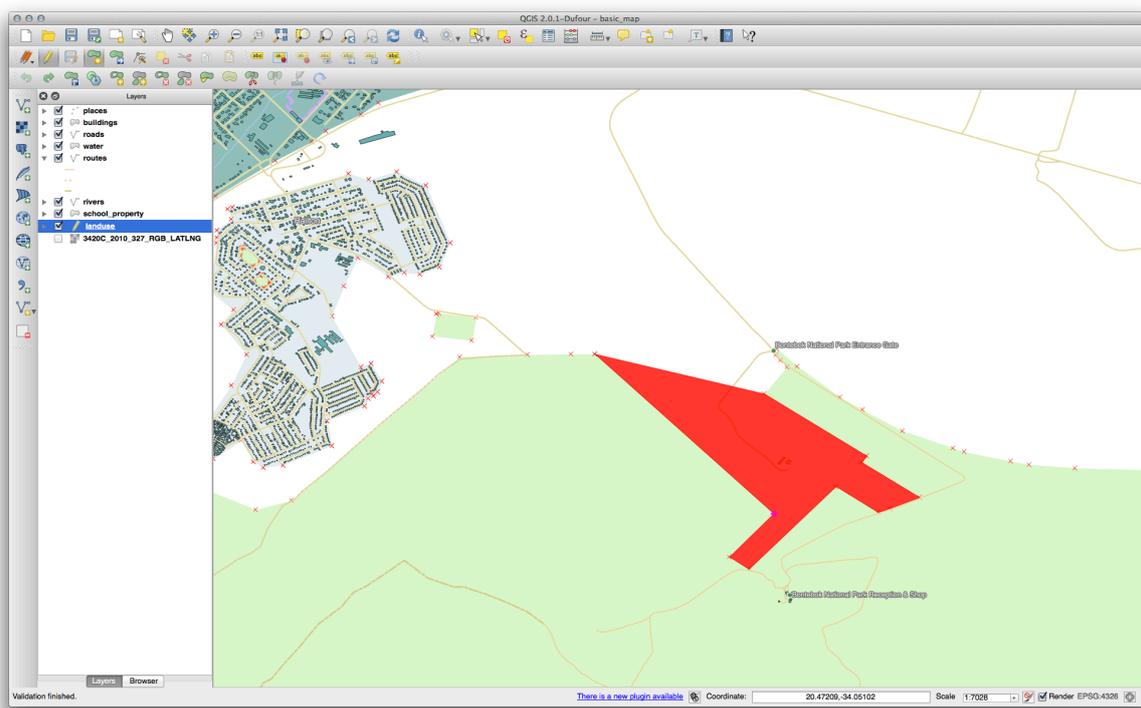
テキストに戻る

21.8.3 *Topology*: パートツールを追加

- 最初に Bontebok National Park を選択します:



- 新しいパートを追加:



- 次のツールのための演習を続行する前に編集を取り消します。

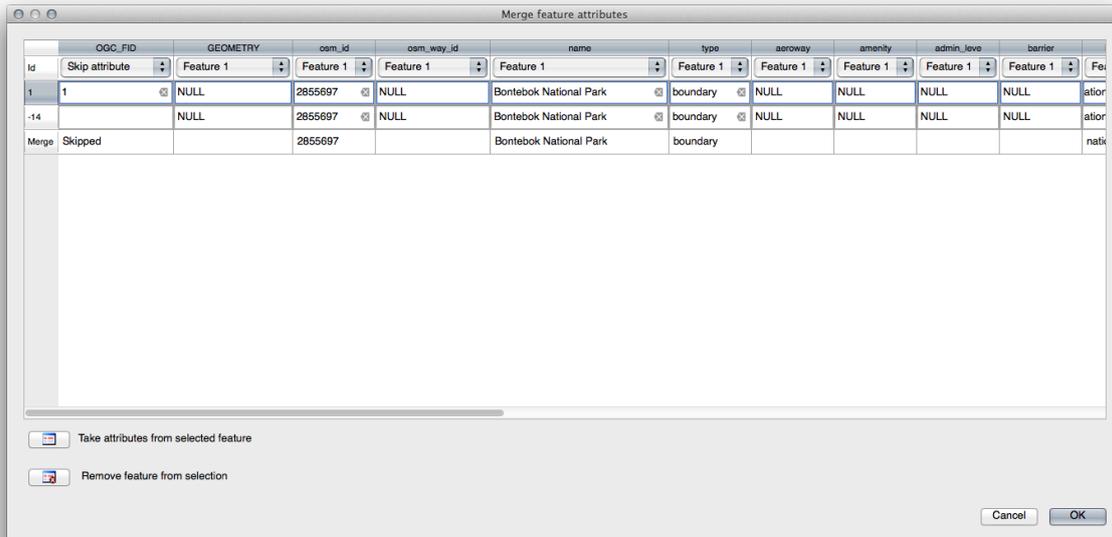
テキストに戻る

21.8.4 地物をマージ

- 選択した地物のマージ ツールを使う際には、最初にマージしたいポリゴンを両方選んでください。
- 1 の属性の *OGC_FID* を持つ地物をソースとして使用します (ダイアログでそのエントリをクリックし、それから 選択地物から属性を取る ボタンをクリックしてください):

ノート:

別のデータセットを使用している場合、それはあなたの可能性が高いです 元々のポリゴンの *OGC_FID* は 1 にはならないでしょう。 *OGC_FID* を持っている地物だけを選択してください。



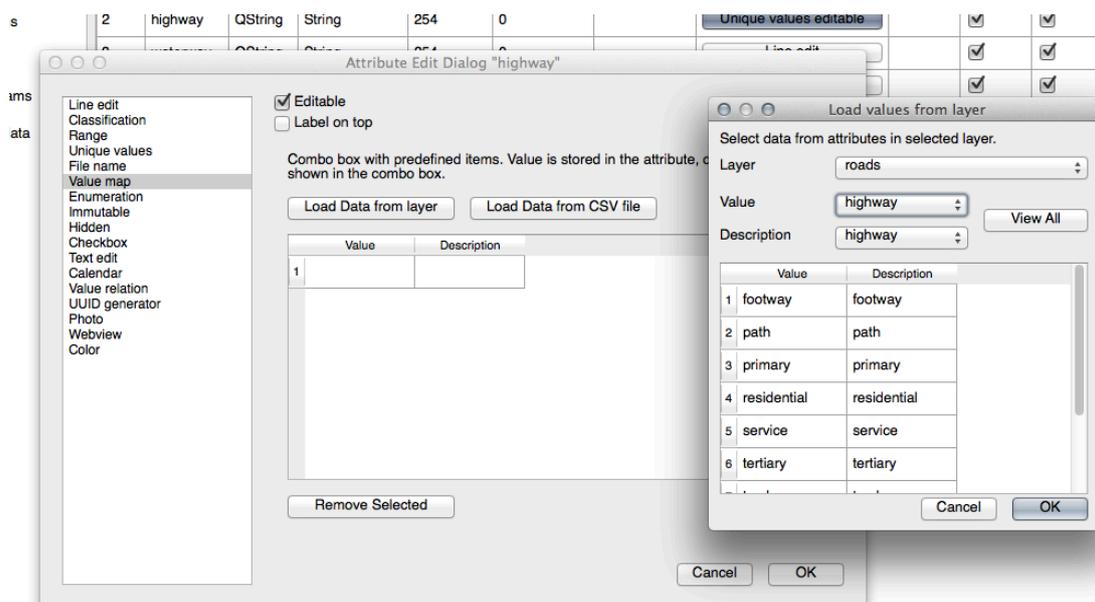
ノート: 選択地物の属性をマージ ツールを使用すると、ジオメトリは別々のまま、それらに同じ属性を与えます。

テキストに戻る

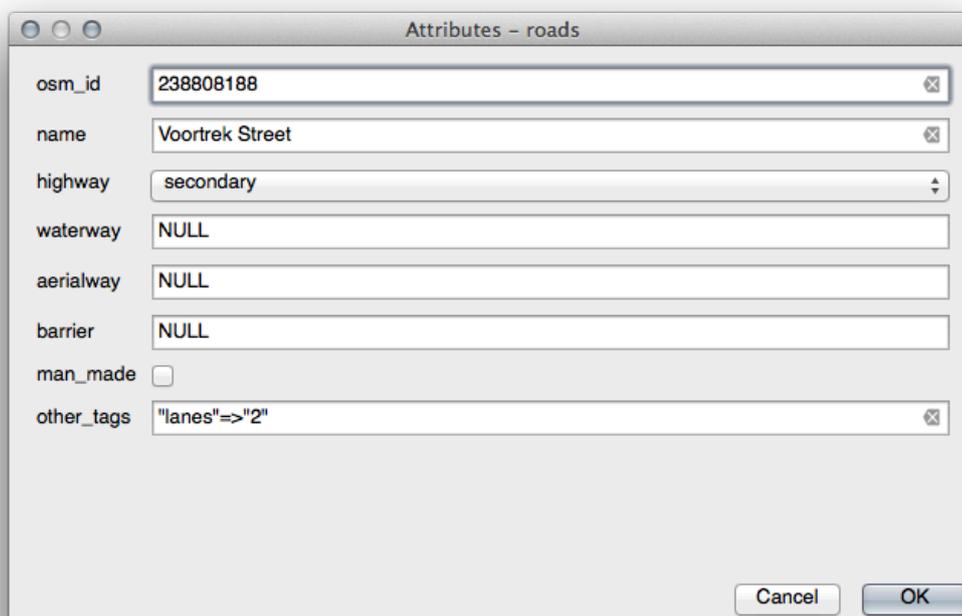
21.8.5 フォーム

type について、道路がなりうるタイプの量は明らかに限られており、そしてこのレイヤーのための属性テーブルをチェックすると、それらが事前定義されていることがわかります。

- ウィジェットをバリューマップにセットしてレイヤからデータをロードをクリックしてください。
- *Label* ドロップダウンで *:guilabel: 'roads* を、値と *Description* オプション両方のために *highway* を選択します:



- *Ok* を 3 回クリックしてください。
- 今路上で `:guiabel:'Identify'` ツールを使用している場合は、編集モードがアクティブな間、出てくるダイアログは次のようになります。



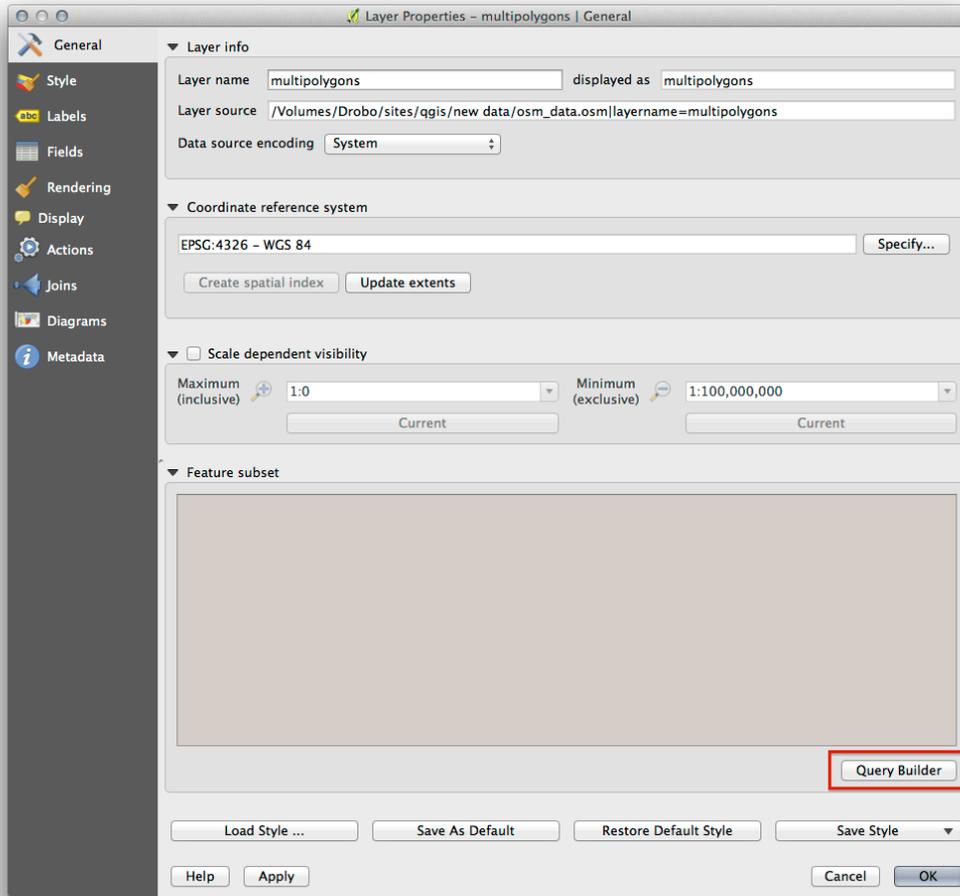
テキストに戻る

21.9 Results For ベクター分析

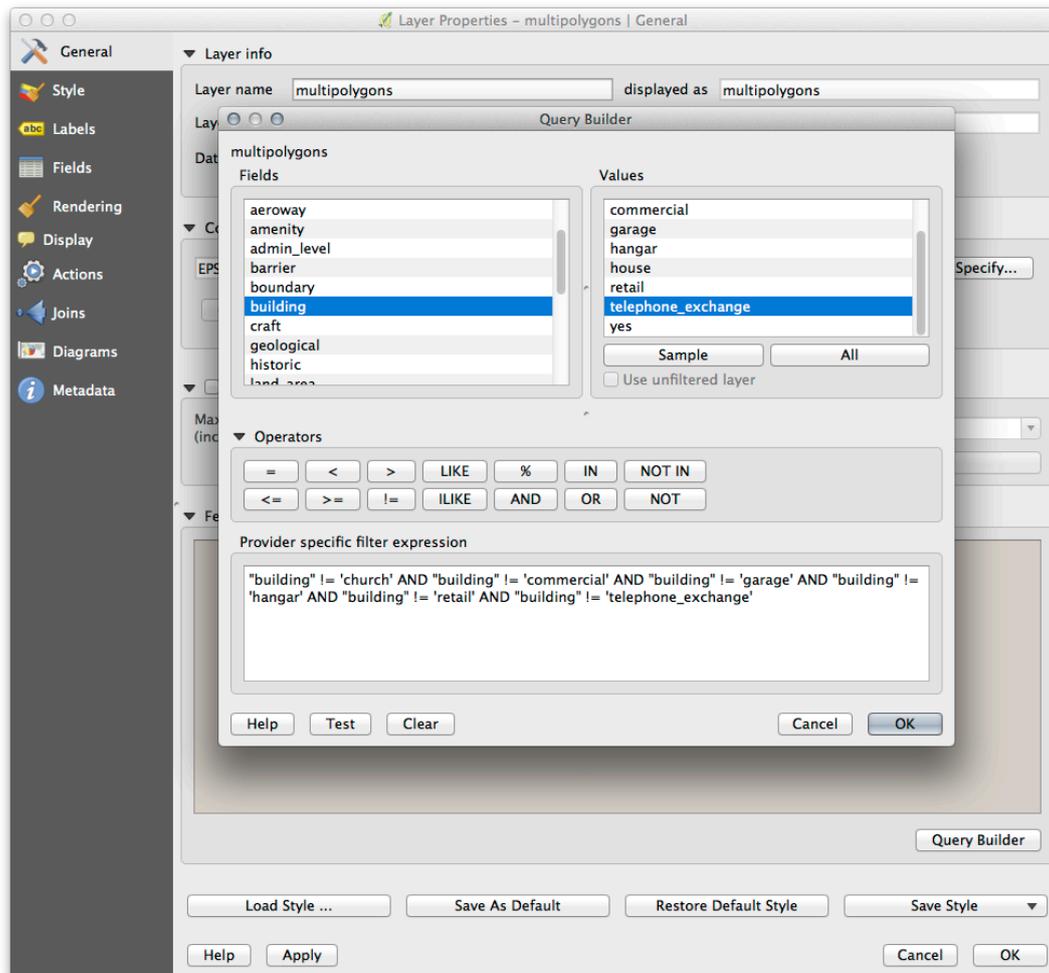
21.9.1 OSM データから自分用のレイヤを抽出

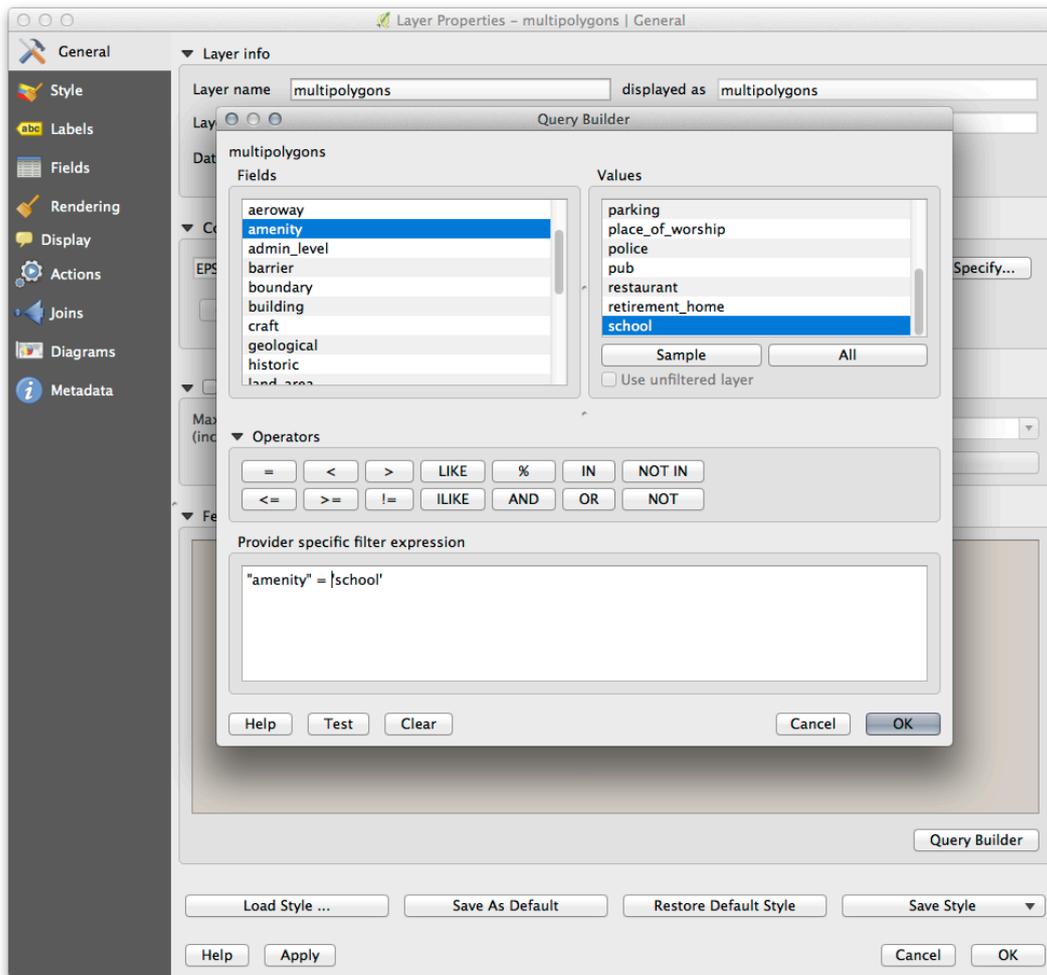
この演習の目的のために、私たちが興味を持っている OSM レイヤーは `multipolygons` と `:kbd:'lines'` です。 `multipolygons` レイヤーには、 `houses`、 `schools` と `restaurants` レイヤーを生産するために必要なデータが含まれています。 `lines` レイヤーには道路データセットが含まれています。

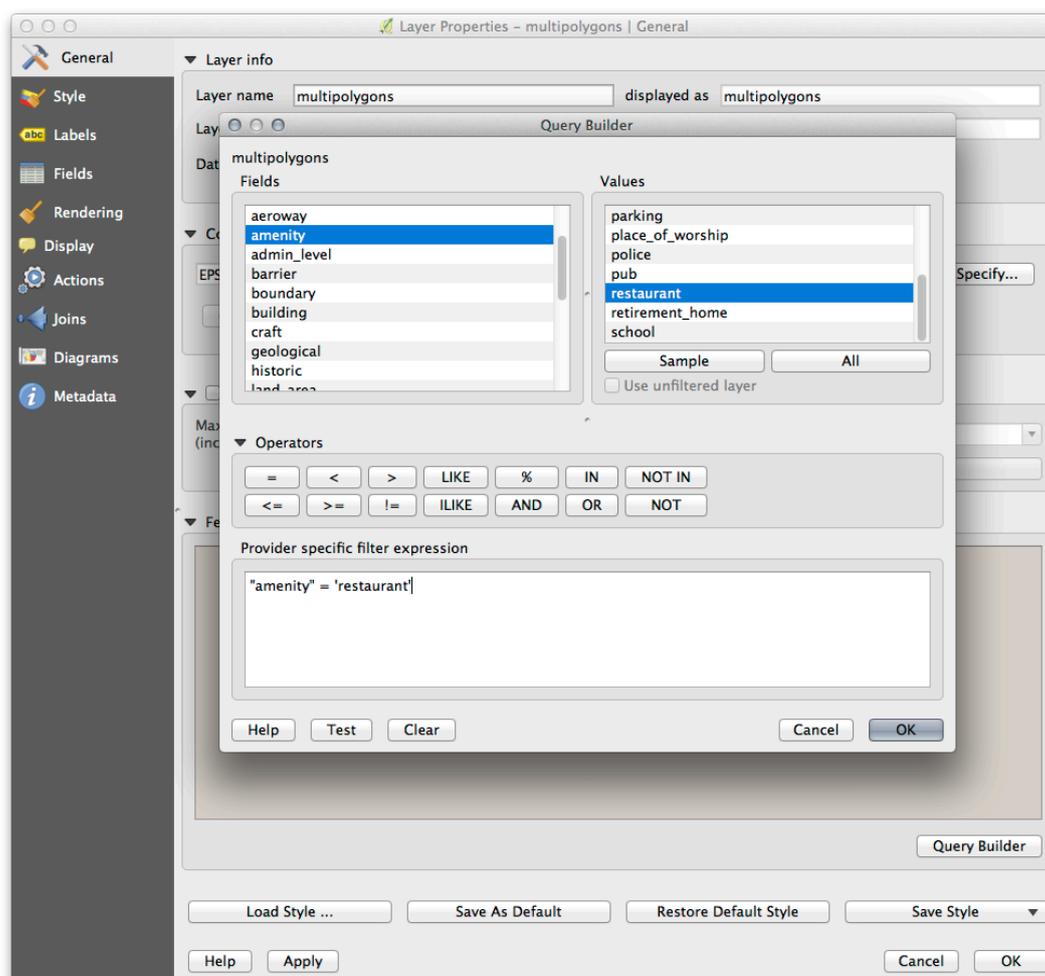
クエリビルダー はレイヤプロパティにあります:



multipolygons レイヤーに対して クエリ *Builder* を使用し、houses、schools、restaurants と residential レイヤーのために以下のクエリを作成します：







各クエリを入力したら、OK をクリックしてください。地図が更新されて選択したデータのみが表示されることがわかります。OSM のデータセットから再び multipolygons データを使用する必要があるので、この時点では、次のいずれかの方法を使用できます。

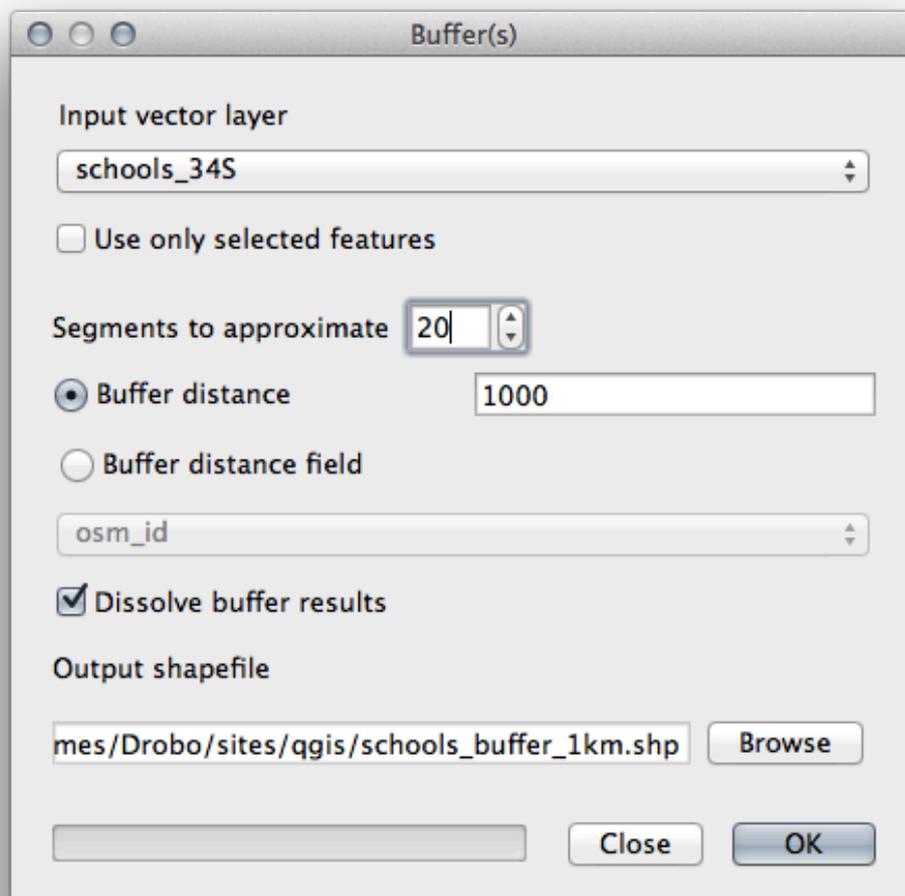
- フィルタされた OSM レイヤーの名前を変更し、osm_data.osm からレイヤーを再インポートします、または
- フィルタレイヤーを複製し、コピーの名前を変更し、クエリをクリアし、クエリ Builder 中で新しいクエリを作成します。

ノート: OSM の building フィールドの値は house 値ではありますが、お住まいの地域のカバレッジは - 私たちのように - 完全ではないかもしれません。テスト領域ではそれゆえ、house 以外と定義されているすべての建物を*除外* することがより正確です。house など明確な意味を持っていない他のすべての値 yes を単にとして定義されている建物を含めることを決定できます。

OSM の lines レイヤーに対して、このクエリを構築し、roads レイヤーを作成します :

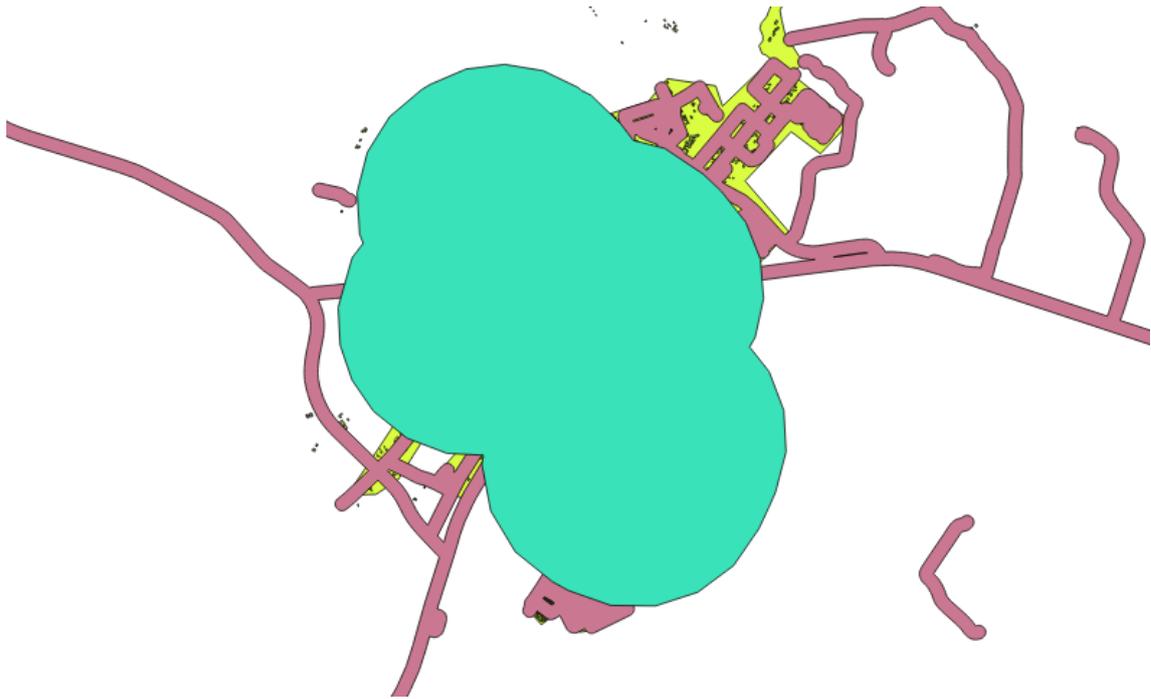
21.9.2 高校からの距離

- あなたのバッファダイアログはこのように見えるはずです:



バッファ距離は 1000 メーターです (すなわち 1 キロメートル)。

- セグメントを *approximate* 値は 20 に設定されます。これはオプションですが、出力バッファがよりスムーズに見えるのでお勧めです。これと比較してみてください:



これに :



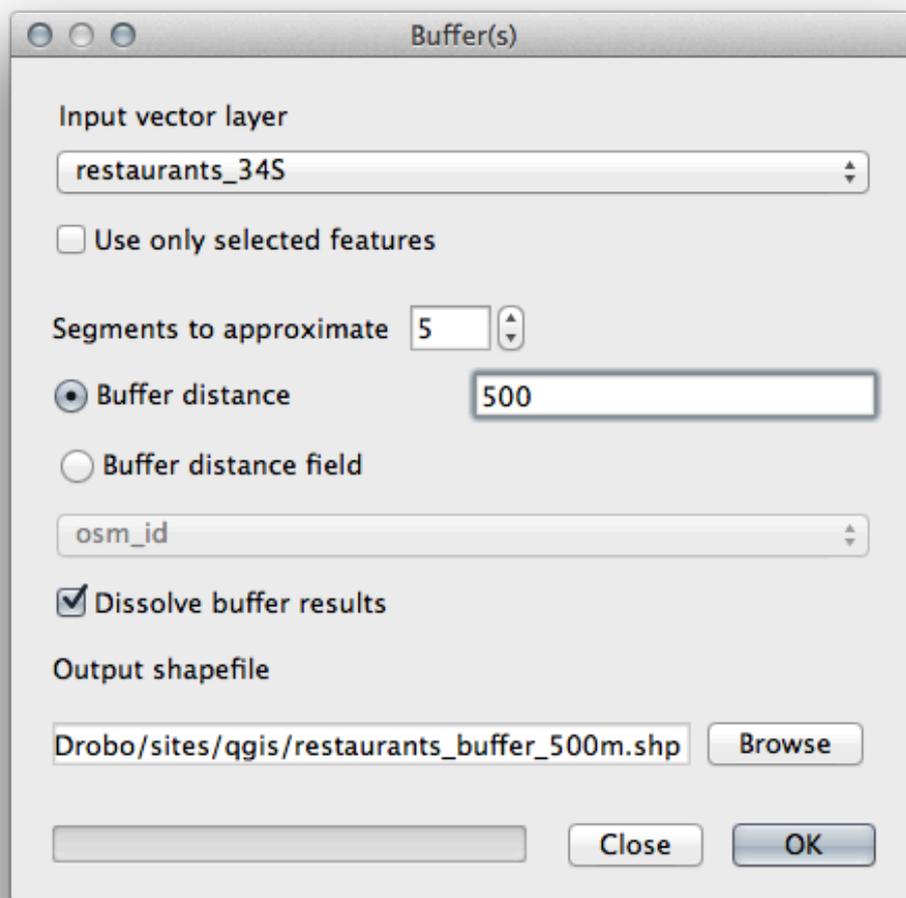
第1の画像は、セグメント *approximate* 値が `5` に設定されたバッファを、第2の画像は値が `20` に設定されたバッファを示しています。この例では違いは微妙ですが、より高い値を持つほどバッファのエッジがより滑らかであることがわかります。

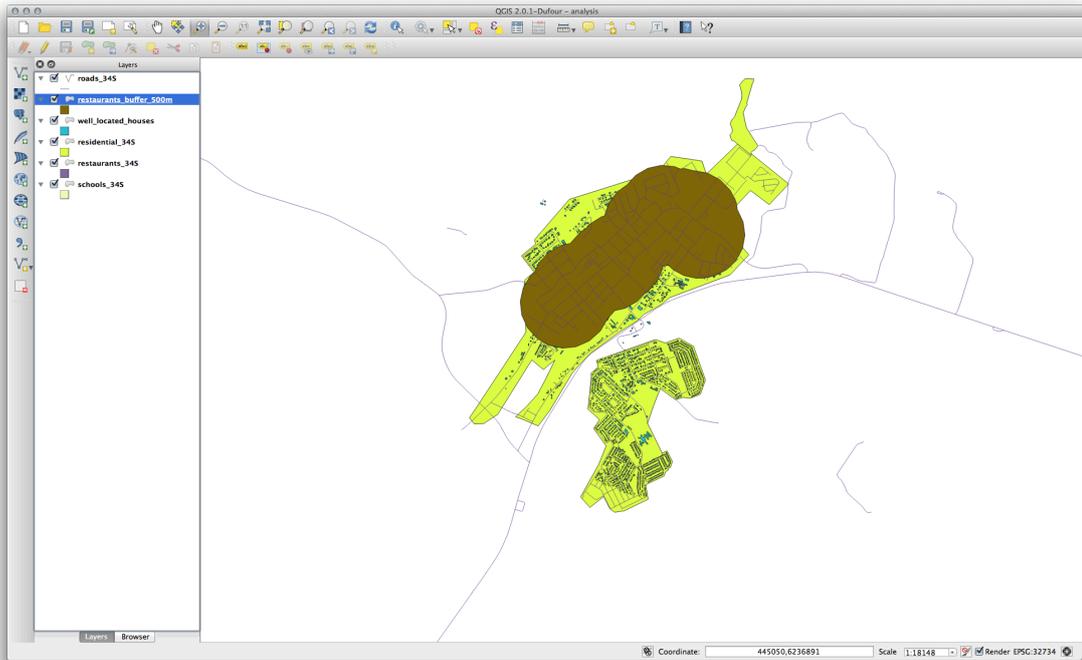
[テキストへ戻る](#)

21.9.3 レストランからの距離

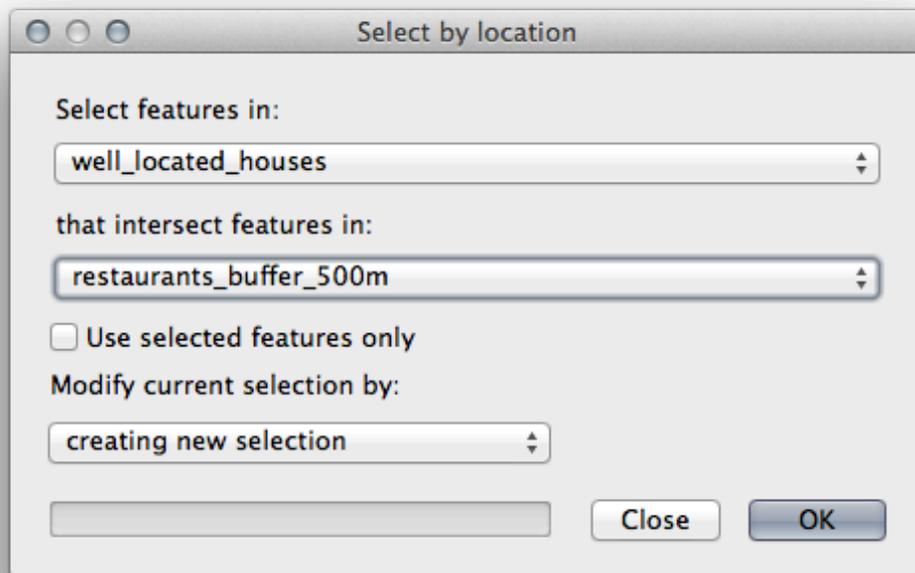
新しい `houses_restaurants_500m` レイヤを作成するために、2ステップの手順を実施します:

- 最初に、レストランの周囲に 500m のバッファを作成し、マップにレイヤを追加します:

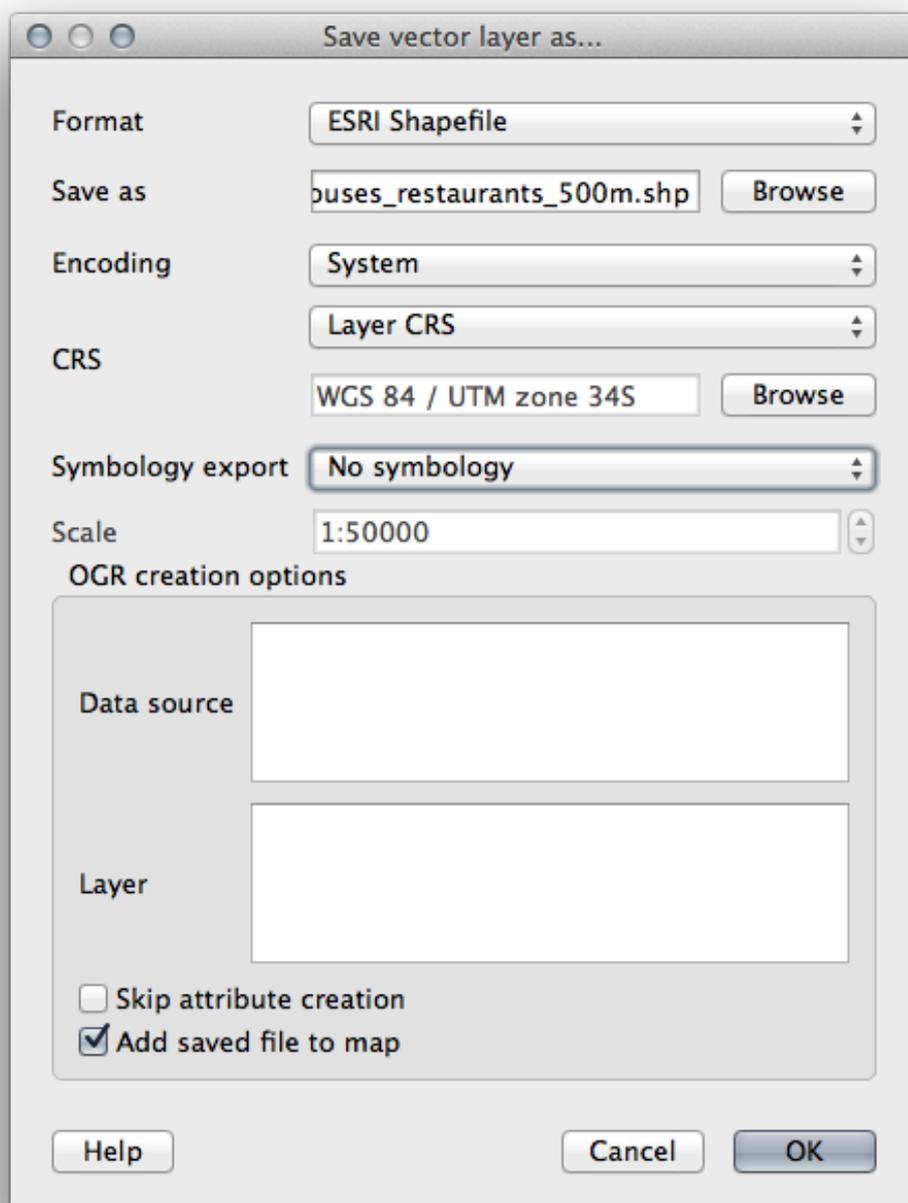




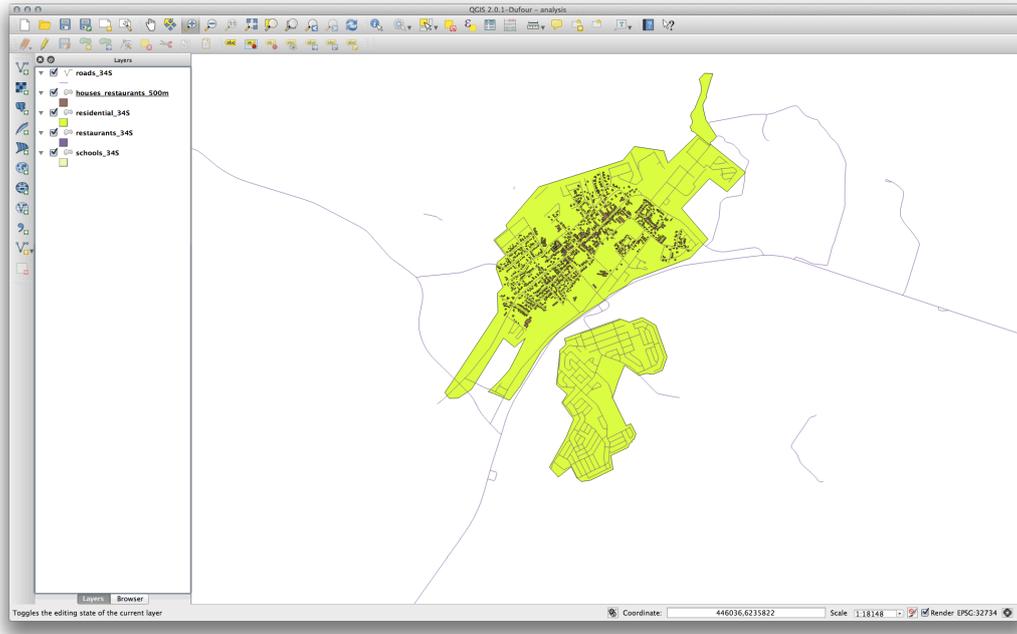
- 次に、そのバッファ領域内の建物を選択します:



- そして選んだものを新しい houses_restaurants_500m レイヤに保存します:



これであなたのマップには道路から 50m 以内、学校から 1km 以内、そしてレストランから 500m 以内の建物だけが表示されます:

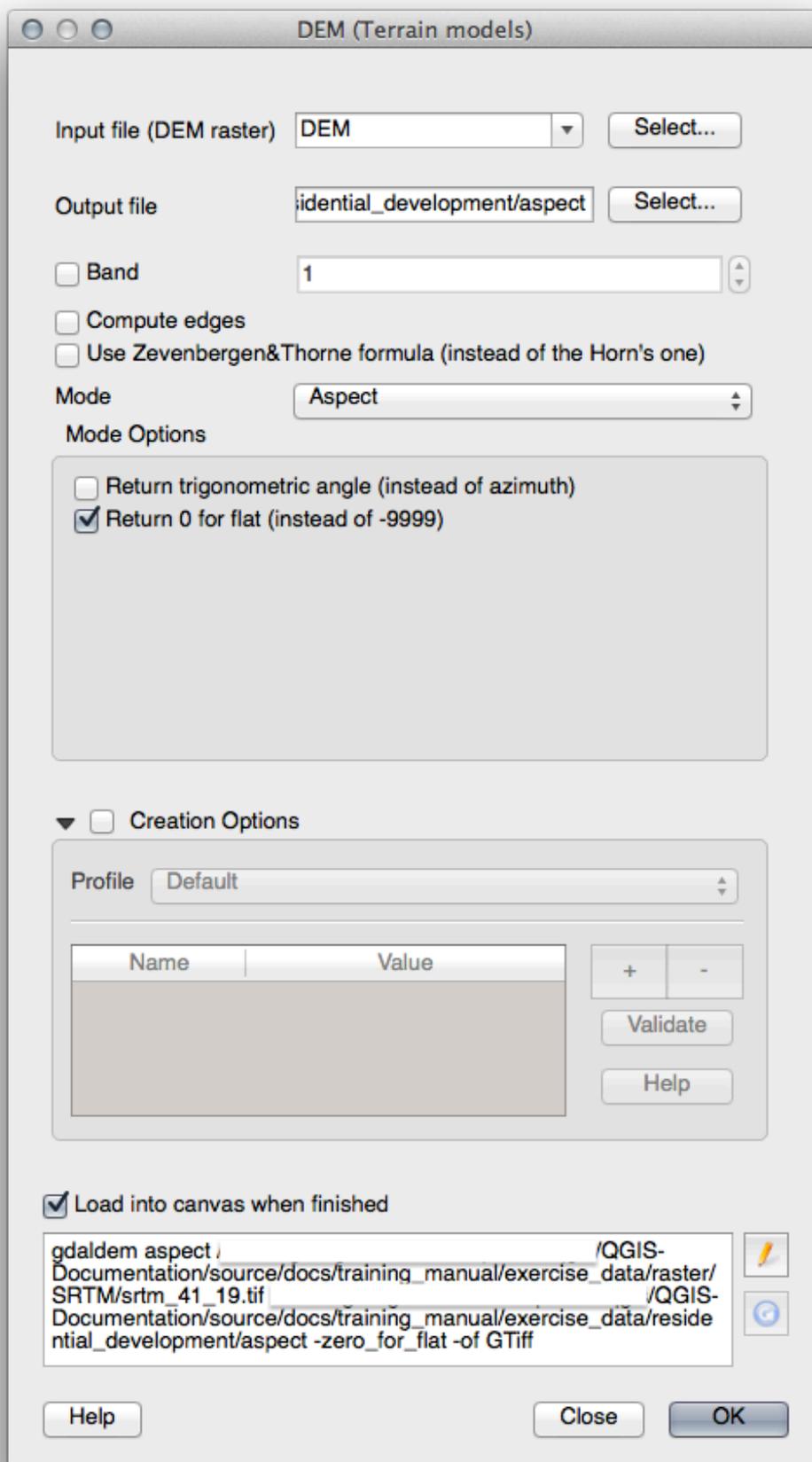


テキストに戻る

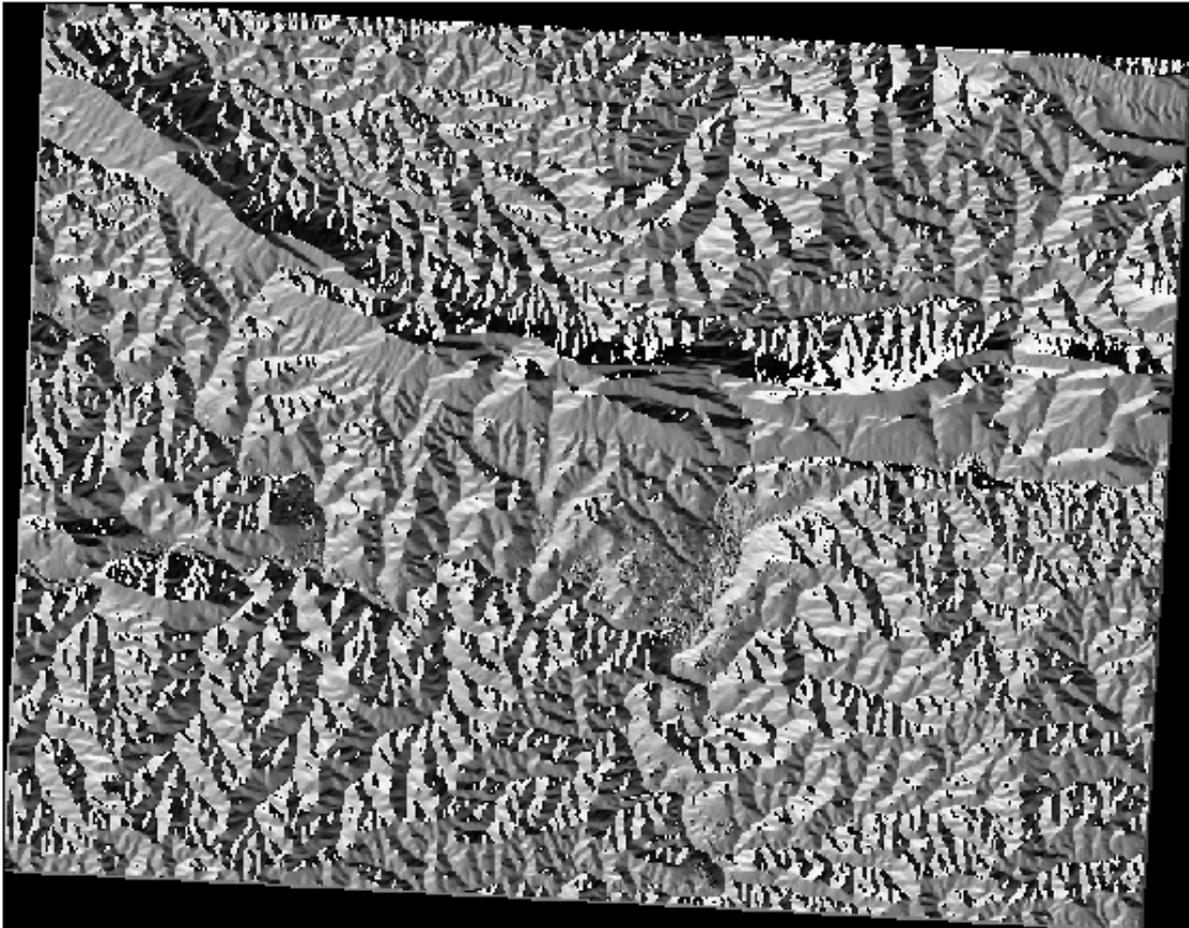
21.10 Results For ラスタ分析

21.10.1 角度を計算

- このように :guilabel:DEM (地形解析) ダイアログアップを設定します



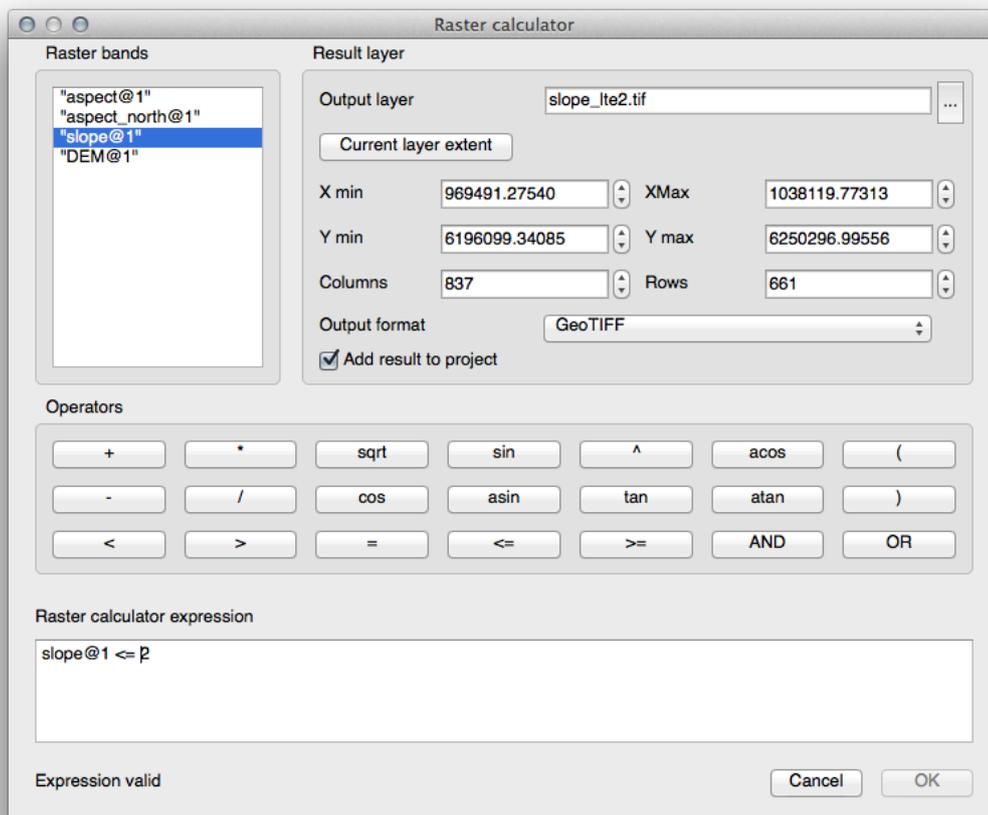
結果:



[テキストに戻る](#)

21.10.2 傾斜を計算 (2度と5度未満)

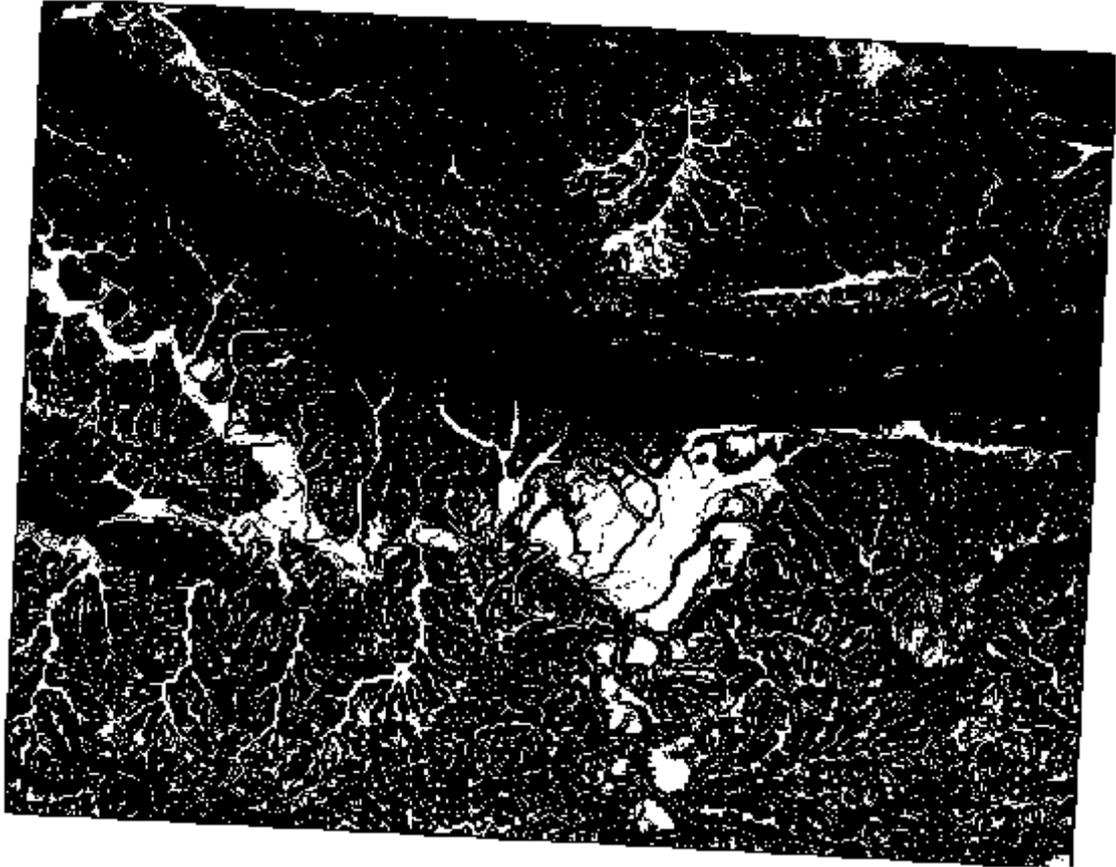
- ラスター計算機 ダイアログをこのように設定します。



- 5 度のバージョンの場合は、式とファイル名の中の 2 を 5 で置き換えてください。

結果:

- 2 度:



- 5 度:



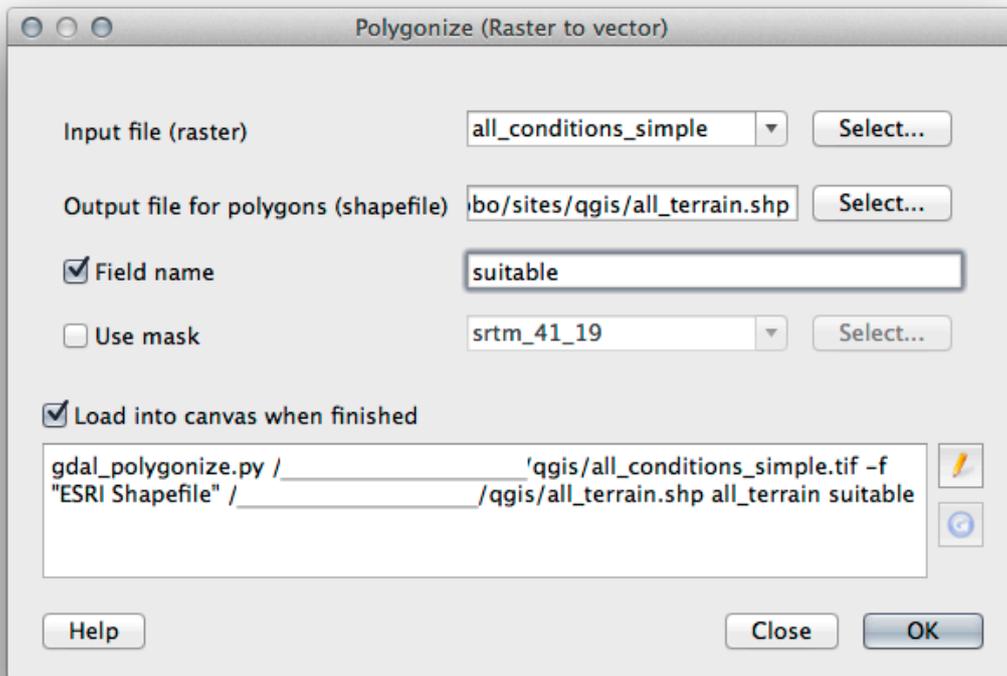
[テキストに戻る](#)

21.11 Results For 分析を完了させる

21.11.1 ラスタからベクタ

- レイヤーリスト中の `all_terrain` レイヤー上で右クリックすることで `:guilabel:` 'クエリ Builder' を開き、一般 タブを選択します。
- 次に、「適切な」= 1 クエリを構築します。
- `OK` をクリックしてこの条件が満たされていないすべてのポリゴンをフィルタリングします。

オリジナルのラスタ上で閲覧するとその領域は完全にオーバーラップされるはずです:



- レイヤーリスト中の *all_terrain* レイヤー上で右クリックし名前を付けて保存... を選択することでこのレイヤーを保存できます。その後は指示に従って続けます。

[テキストに戻る](#)

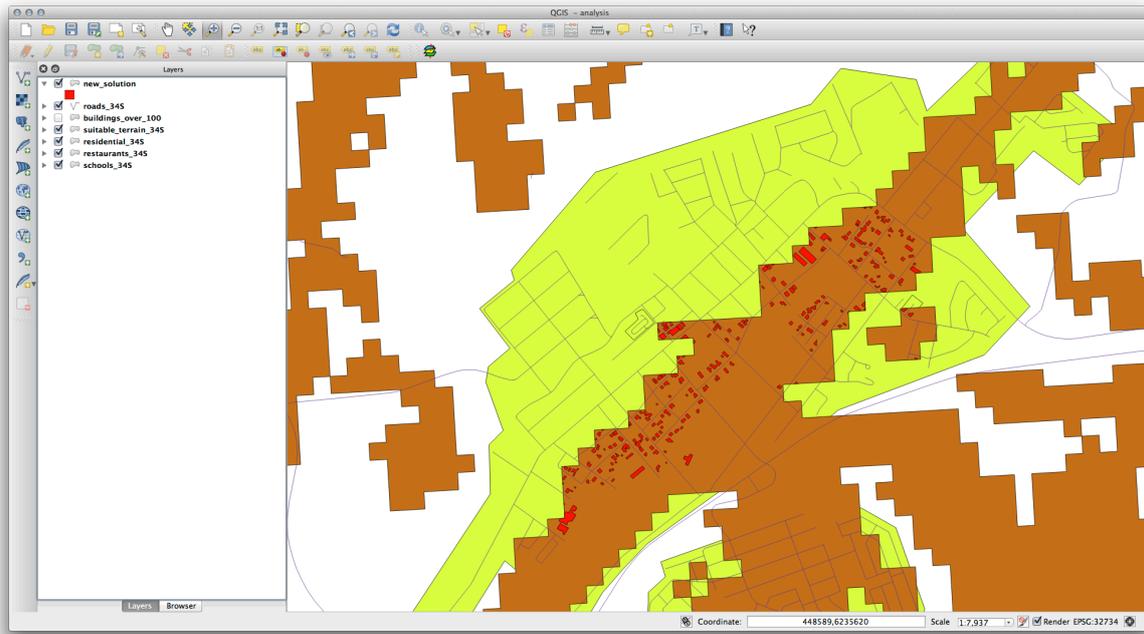
21.11.2 結果を精査

Intersect ツールによって `:kbd: new_solution` レイヤー中の建物の一部が「スライス」されることがあることに気づくことがあります。これは、建物の一部のみ - それゆえ資産の一部のみ - が適した地形の上にあることを示しています。したがって、賢明にデータセットから、これらの建物を排除できます

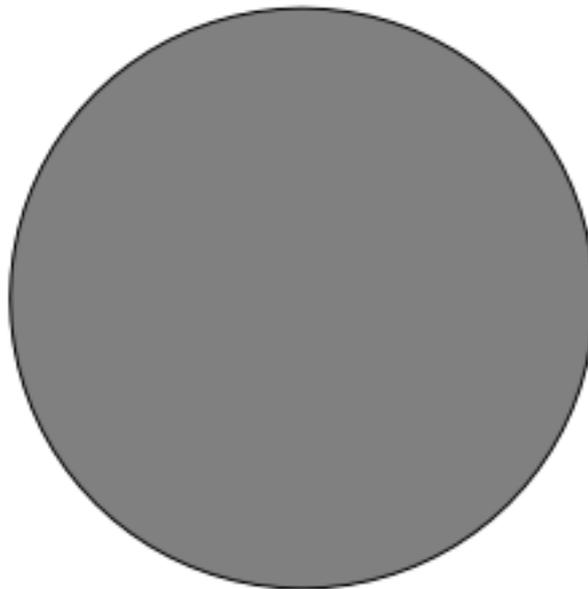
[テキストに戻る](#)

21.11.3 分析を改善する

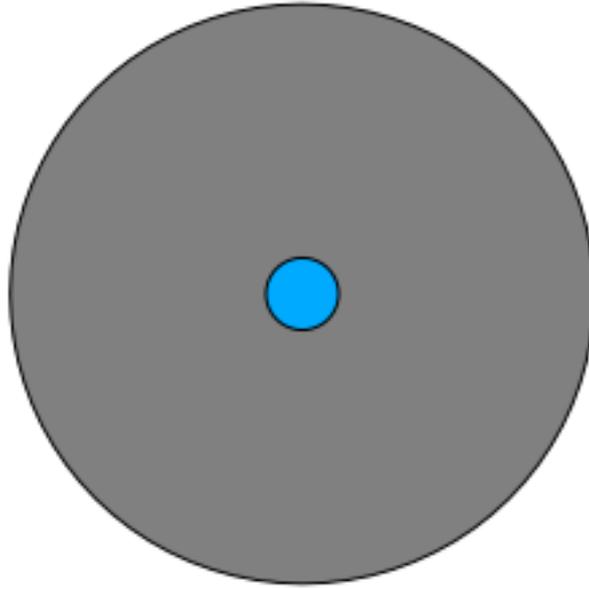
現時点ではあなたの分析は次のように見えるはずですが:



全ての方向に 100 メートルのための連続円形領域を考えます。



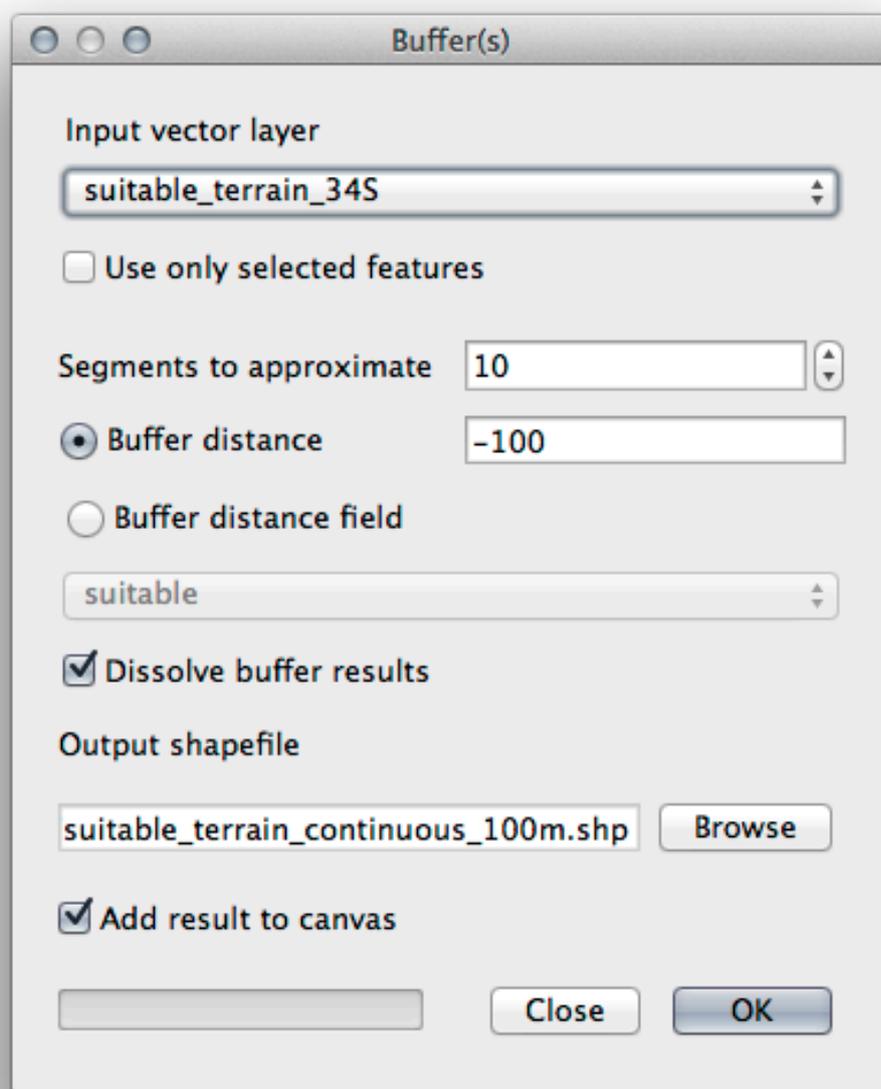
それは半径 100 メートルより大きい場合、(すべての方向から)、その大きさから 100m 減算して、それが途中で放置された部分になります。



そのため、既存の *suitable_terrain* ベクトルレイヤー上で 100 メートルの*内部バッファ*を実行できます。バッファ機能の出力においては、元のレイヤーのどんな残りも、100 メートルを超えて適した地形がある領域を表すことになります。

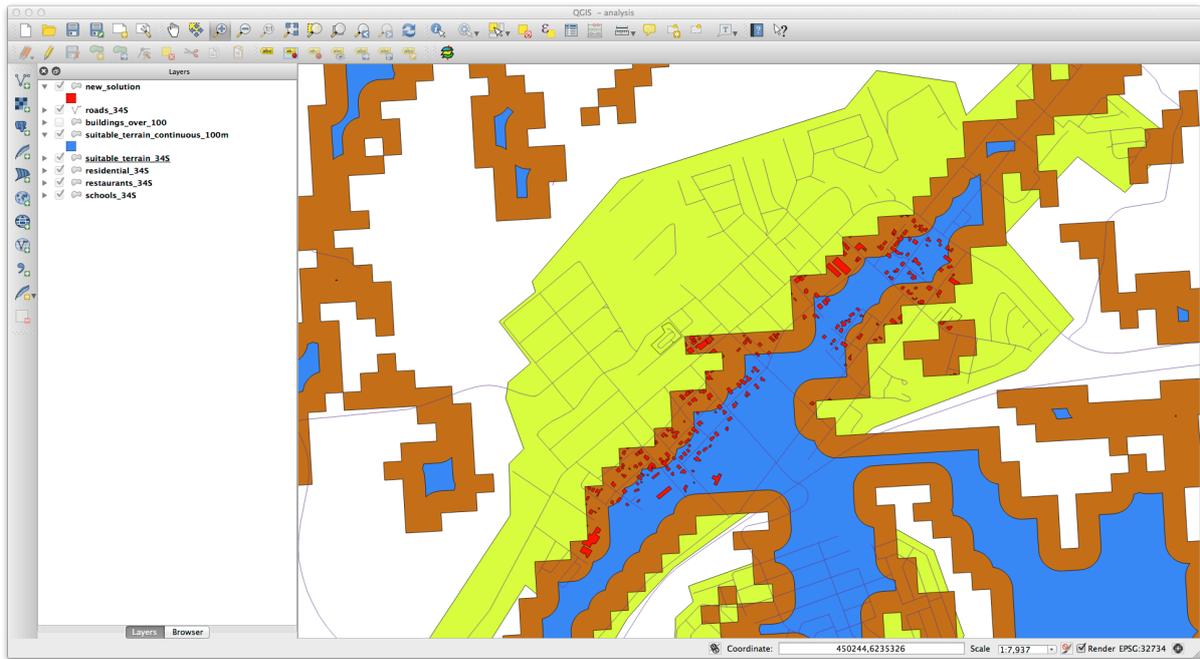
証明するために：

- ベクトル -> ジオプロセッシングツール -> バッファ に行き、バッファダイアログを開きます。
- このようにセットアップします：

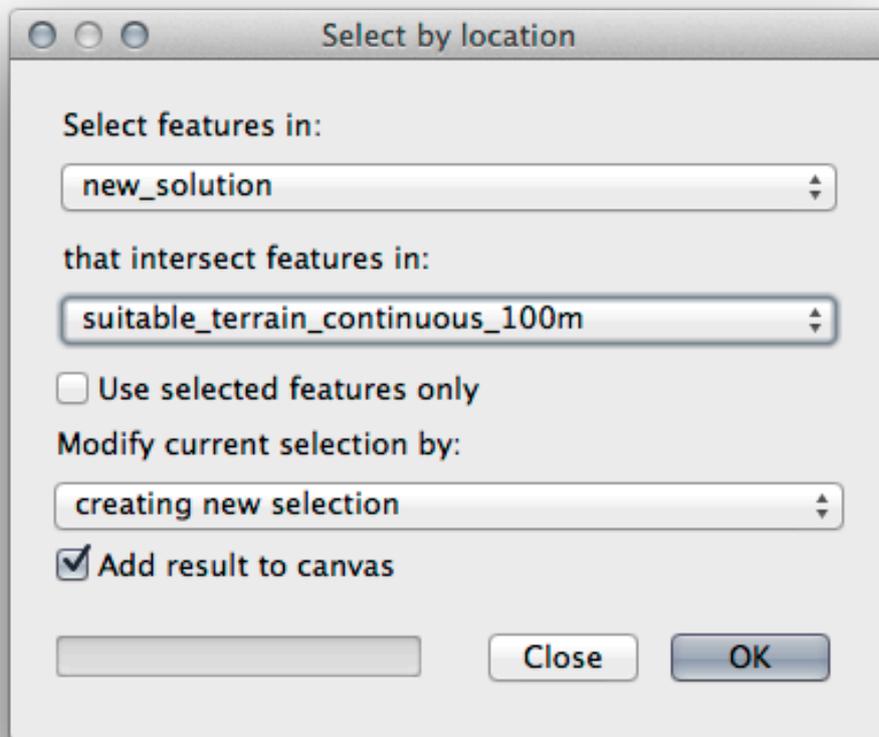


- 10 のセグメントで -100 のバッファ距離で *suitable_terrain* レイヤーを使用します。(地図が投影 CRS を使用しているため、距離は自動的にメートル単位です。)
- `exercise_data / residential_development /` と `suitable_terrain_continuous100m.shp` で出力を保存。
- 必要に応じて、あなたのオリジナルの *suitable_terrain* レイヤの上に新しいレイヤを移動してください。

作業結果は次のように見えるはずでず:



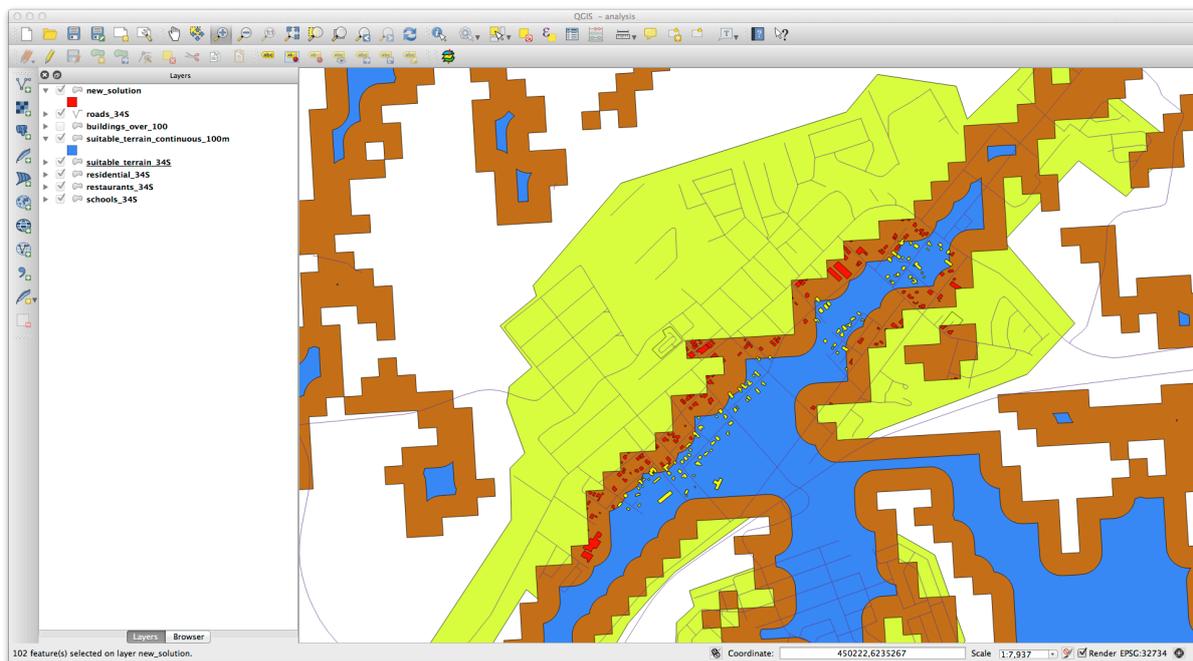
- 今 *Location* で選択 ツール (ベクトル -> 研究のツール -> *location* により選択) を使用します。
- このようにセットアップします:



- :guilabel:' suitable_terrain_continuous100m.shp' の中の地物に交差する *new_solution* 中の地物を選択し

ます。

結果はこちらです:



黄色の建物が選択されています。建物の一部は、新しい `suitable_terrain_continuous100m` レイヤー外に一部が落ちるものの、それらは元の `suitable_terrain` レイヤー範囲内に十分にあり、したがって私たちの要件のすべてを満たしています。

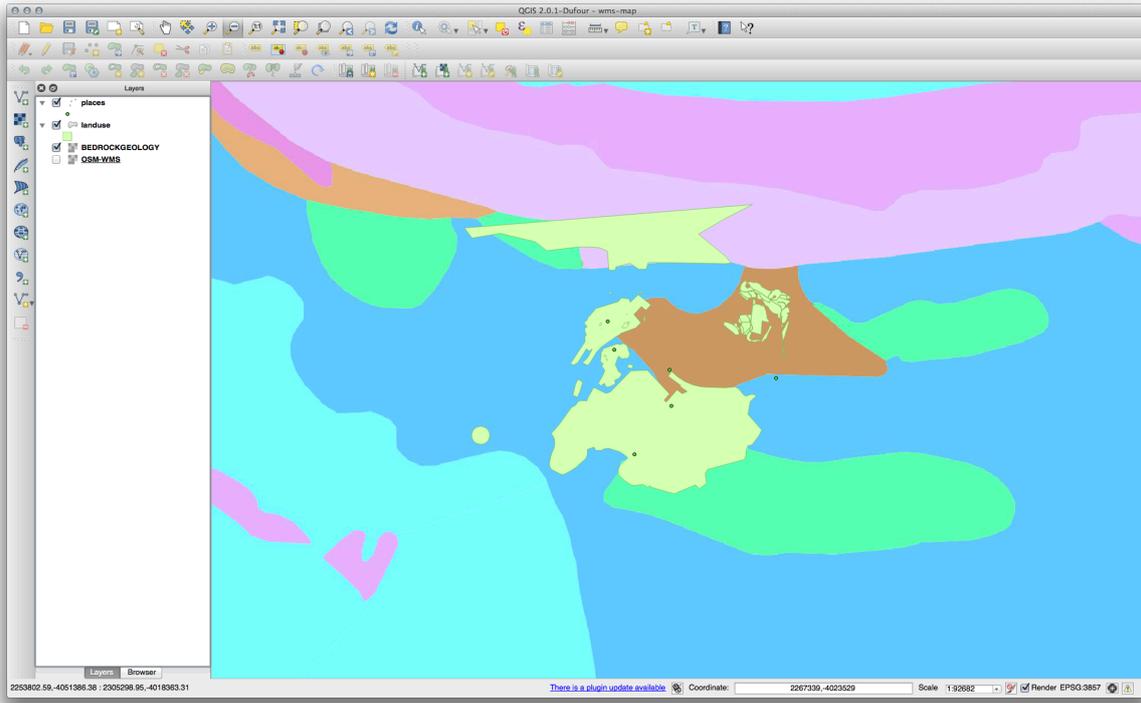
- 選択を `exercise_data/residential_development/` 下に `final_answer.shp` として保存してください。

[テキストに戻る](#)

21.12 Results For WMS

21.12.1 別の WMS レイヤを追加します

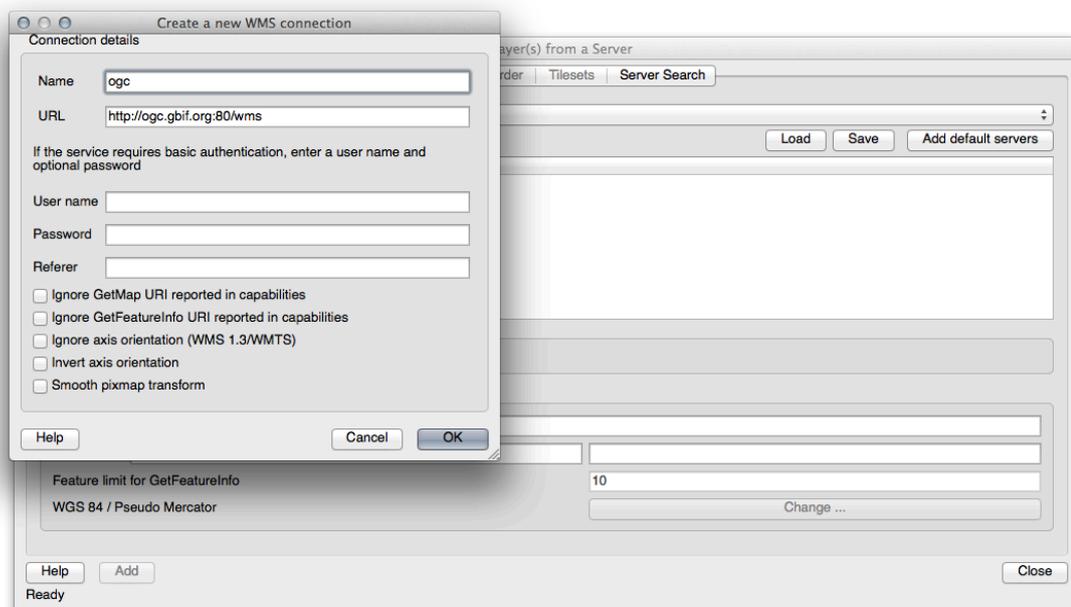
地図は次のようになります (レイヤーを整理し直す必要があるかもしれません):

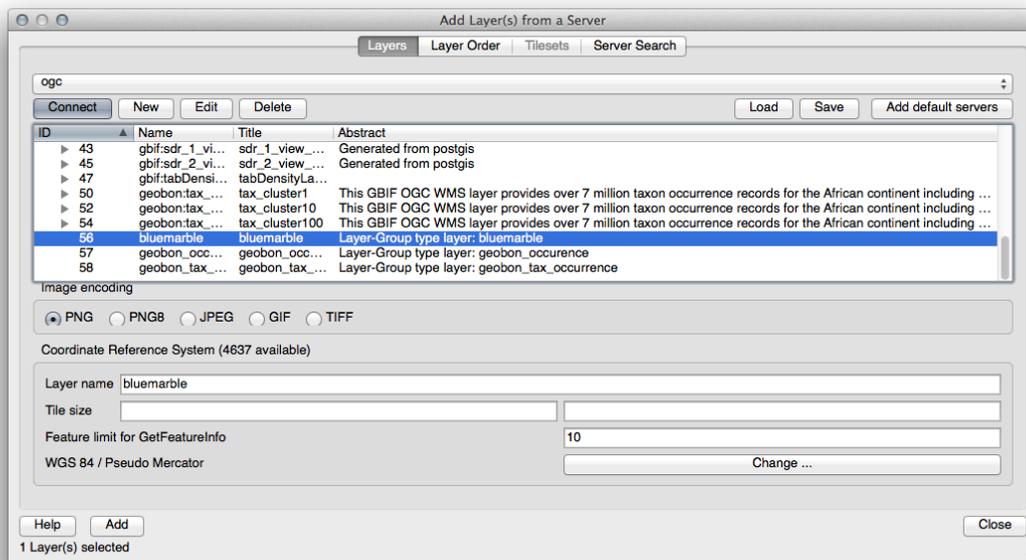


テキストに戻る

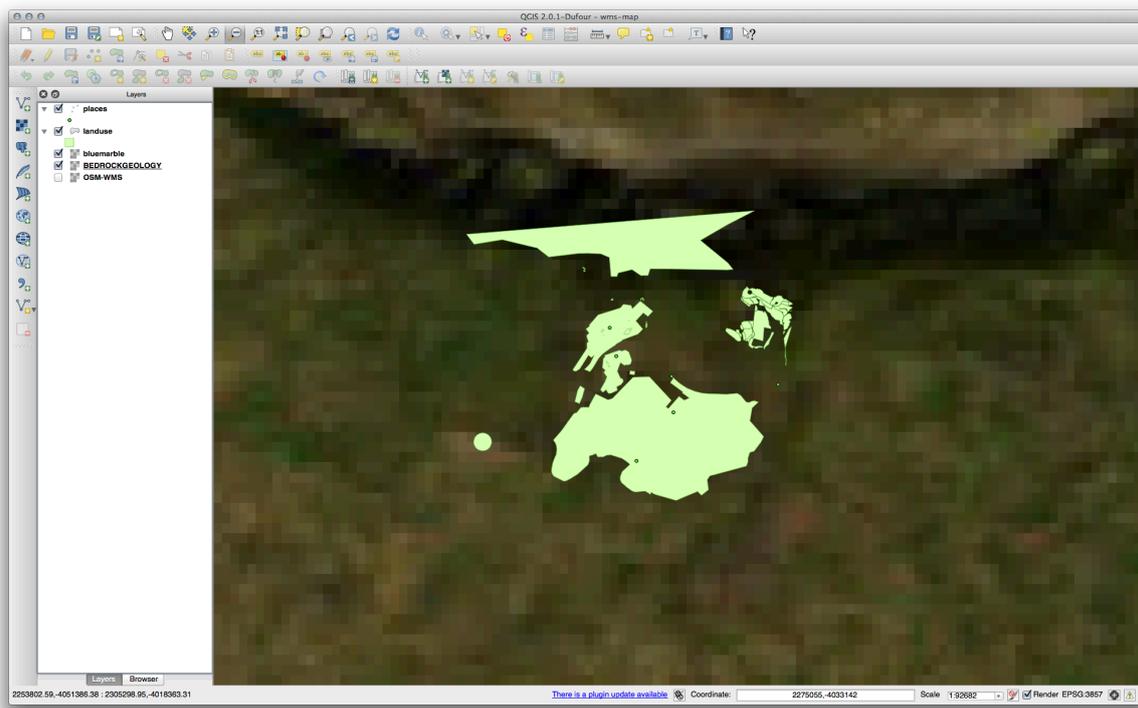
21.12.2 新しい WMS サーバの追加

- 新しいサーバーとそのサーバー上でホストされているように、適切なレイヤーを追加する前と同じアプローチを使用します。





- Swellendam 領域にズームインした場合、このデータセットは低解像度を持っていることに気づくでしょう：



したがって、現在の地図にこのデータを使用しない方が良いでしょう。ブルーマーブルデータは、グローバルまたは全国規模の方が適しています。

テキストに戻る

21.12.3 WMSサーバの検索

多くの WMS サーバが常に利用可能ではないことに気づくことがあります。これは、時には、時にはそれが永続的で、一時的なものです。執筆の時点で稼働していた WMS サーバの例は http://apps1.gdr.nrcan.gc.ca/cgi-bin/worldmin_en-ca_ows で世界鉱物鉱床 WMS です。これは、手数料を必要とせず、アクセス制限もなく、全世界的です。したがって、まさに要件を満たしています。これは単なる例であること、しかし、覚えておいてください。から選択する多くの他の WMS サーバがあります。

[テキストに戻る](#)

21.13 Results For データベースの概念

21.13.1 アドレステーブルのプロパティ

私たちの理論上のアドレステーブルのために、私たちは、次のプロパティを保存したい場合があります:

```
House Number
Street Name
Suburb Name
City Name
Postcode
Country
```

アドレスオブジェクトを表すために、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現するために列を作成し、SQL 準拠し、おそらく短縮名とそれらに名前を付けるでしょう:

```
house_number
street_name
suburb
city
postcode
country
```

[テキストに戻る](#)

21.13.2 人々表を正規化します

people テーブルの主要な問題は、人の住所全体を含む単一の住所フィールドが存在することです。以前のレッスンでは、私たちの理論 *address* テーブルを考える、私たちは住所が多くの異なる特性で構成されていることを知っています。すべてのこれらのプロパティを1つのフィールド内に格納することにより、データを更新して照会することがはるかに困難にします。そこで、様々なプロパティに住所フィールドを分割する必要があります。これは、次のような構造を持つテーブルを与えるだろう:

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

ノート: 次のセクションでは、さらに当社のデータベースの構造を改善するために、この例で使うことができ、外部キーの関係について学びます。

[テキストに戻る](#)

21.13.3 人々表のさらなる正規化

当社 'people' テーブルには、現在このようになります:

id	name	house_no	street_id	phone_no
1	Horst Duster	4	1	072 121 122

street_id 列は 'people' オブジェクトと関連 'street' オブジェクト、'streets' テーブルにある、の間の「1 対多」関係を表します。

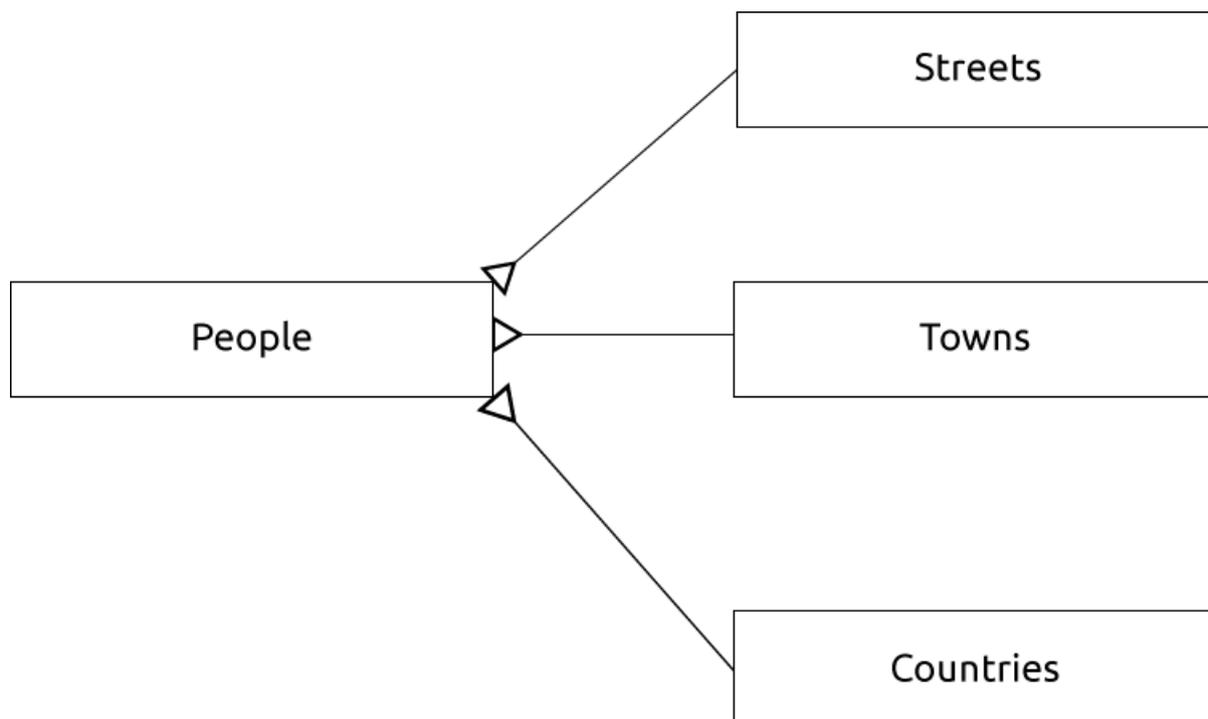
テーブルをさらに正規化する一つの方法は、名前のフィールドを姓と名に分割することです:

id	first_name	last_name	house_no	street_id	phone_no
1	Horst	Duster	4	1	072 121 122

また、町名や都市名と国に対して別々のテーブルを作成し、「1 対多」関係を介して私たちの people テーブルにそれらをリンクできます:

id	first_name	last_name	house_no	street_id	town_id	country_id
1	Horst	Duster	4	1	2	1

ER ダイアグラムは次のようになり、これを表現します:



[テキストに戻る](#)

21.13.4 人々表を作成します

正しい人のテーブルを作成するために必要な SQL です:

```

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は (`\d people` と入力します) 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)

ノート: 説明のために、FKEY 制約は意図的に省略しています。

[テキストへ戻る](#)

21.13.5 DROP コマンド

people テーブルは *streets* テーブルへの外部キー制約があるため、DROP コマンドは、このケースでは動作しない理由があります。これは、*streets* テーブルをドロップする (または削除する)、存在しない *streets* データを参照して *people* テーブルを離れることを意味します。

ノート: それは力 `CASCADE` コマンドを使用して削除する *streets* テーブルに可能であるが、これはまた *people* と *streets* テーブルへの関係を持っていた他のテーブルを削除します。注意して使用してください!

[テキストへ戻る](#)

21.13.6 新しい街路を挿入します

使用する必要がある SQL コマンドは、(選択した名前を持つ通りの名前を置き換えできます)、このようになります:

```
insert into streets (name) values ('Low Road');
```

[テキストへ戻る](#)

21.13.7 外部キー関係に新しい人を追加

ここでは、正しい SQL ステートメントがある:

```
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');
```

再び路上のテーブルを (以前のように `select` 文を使用して) 見ると、主要道路 エントリのための `id` は 2 であることがわかるでしょう。

上で単に数 2 以を入力することができた理由です。私たちが上記のエントリで完全に書き出された 主要道路 を見ていないにもかかわらず、データベースはそれを 2 の `street_id` 値を持つものと関連付けできるようにします。

ノート: すでに新しい `street` オブジェクトを追加している場合、新しい主要道路の ID は 2 でなく 3 であることを見つけるかもしれません

[テキストへ戻る](#)

21.13.8 戻りストリート名

ここでは、使用すべき正しい SQL ステートメントがある:

```
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)
```

ノート: テーブル名をフィールド名の接頭辞にしていることに気づくでしょう (例えば `people.name` と `streets.name`)。フィールド名があいまいな (つまり、データベース内のすべてのテーブル間で一意ではない) 時はいつでもこれが行われる必要があります。

[テキストへ戻る](#)

21.14 Results For 空間問合せ

21.14.1 空間クエリで使用される単位

レイヤーが使用している CRS これは、その単位は度であることを意味する地理 CRS、ある WGS 84 であるため、例えば、クエリで使用される単位は、度です。投影 CRS は、UTM 投影のように、メートルです。

クエリを書くときはレイヤーの CRS がどの単位かを知る必要があることを忘れないでください。これにより期待する結果が返されるクエリを記述できるでしょう。

[テキストへ戻る](#)

21.14.2 空間索引の作成

```
CREATE INDEX cities_geo_idx
ON cities
USING gist (the_geom);
```

[テキストへ戻る](#)

21.15 Results For ジオメトリ建設

21.15.1 ラインストリングを作成します

```
alter table streets add column the_geom geometry;
alter table streets add constraint streets_geom_point_chk check
    (st_geometrytype(the_geom) = 'ST_LineString'::text OR the_geom IS NULL);
insert into geometry_columns values ('','public','streets','the_geom',2,4326,
    'LINESTRING');
create index streets_geo_idx
    on streets
    using gist
    (the_geom);
```

[テキストへ戻る](#)

21.15.2 表のリンク

```
delete from people;
alter table people add column city_id int not null references cities(id);
```

(QGIS の都市をキャプチャ)

```
insert into people (name,house_no, street_id, phone_no, city_id, the_geom)
    values ('Faulty Towers',
        34,
        3,
        '072 812 31 28',
        1,
        'SRID=4326;POINT(33 33)');
```

```
insert into people (name,house_no, street_id, phone_no, city_id, the_geom)
    values ('IP Knightly',
        32,
        1,
        '071 812 31 28',
        1,F
        'SRID=4326;POINT(32 -34)');
```

```
insert into people (name,house_no, street_id, phone_no, city_id, the_geom)
    values ('Rusty Bedsprings',
        39,
        1,
        '071 822 31 28',
        1,
        'SRID=4326;POINT(34 -34)');
```

次のエラーメッセージを取得している場合：

```
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".
```

それは都市テーブルのポリゴンを作成して実験しながら、それらのいくつかを削除してやり直していなければならないことを意味します。都市テーブルのエントリをチェックし、いずれが存在する:kbd:id を使用するだけです。

[テキストに戻る](#)

21.16 Results For 簡易機能モデル

21.16.1 表を移入

```
create table cities (id serial not null primary key,
                    name varchar(50),
                    the_geom geometry not null);
alter table cities
add constraint cities_geom_point_chk
check (st_geometrytype(the_geom) = 'ST_Polygon'::text );
```

[テキストに戻る](#)

21.16.2 * GEOMETRY_COLUMNS 表を移入*

```
insert into geometry_columns values
('','public','cities','the_geom',2,4326,'POLYGON');
```

[テキストに戻る](#)

21.16.3 ジオメトリの追加

```
select people.name,
       streets.name as street_name,
       st_astext(people.the_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、私たちの制限ではデータベースへの null の追加を認めています。

[テキストに戻る](#)

Chapter 22

索引とテーブル

- *genindex*
- *modindex*
- *search*